



Det här verket har digitaliserats vid Göteborgs universitetsbibliotek och är fritt att använda. Alla tryckta texter är OCR-tolkade till maskinläsbar text. Det betyder att du kan söka och kopiera texten från dokumentet. Vissa äldre dokument med dåligt tryck kan vara svåra att OCR-tolka korrekt vilket medför att den OCR-tolkade texten kan innehålla fel och därför bör man visuellt jämföra med verkets bilder för att avgöra vad som är riktigt.

This work has been digitized at Gothenburg University Library and is free to use. All printed texts have been OCR-processed and converted to machine readable text. This means that you can search and copy text from the document. Some early printed books are hard to OCR-process correctly and the text may contain errors, so one should always visually compare it with the images to determine what is correct.



Rapport

R137:1983

# Datorn för byggnadskonstruktionsarbete

— metoder, möjligheter och risker

Anders Eriksson

Leif Andersson

Jan Ludvigsson

R  
Adt

INSTITUTET FÖR BYGGDOKUMENTATION	
Accnr	Plac Ser

Byggtforskningsrådet



R137:1983

DATORN FÖR BYGGNADSKONSTRUKTIONSARBETE

- metoder, möjligheter och risker

Anders Eriksson  
Leif Andersson  
Jan Ludvigsson

Denna rapport hänför sig till forskningsanslag  
820325-4 från Statens råd för byggnadsforskning  
till Institutionen för brobyggnad, Tekniska  
högskolan, Stockholm.

I Byggforskningsrådets rapportserie redovisar forskaren sitt anslagsprojekt. Publiceringen innebär inte att rådet tagit ställning till åsikter, slutsatser och resultat.

R137:1983

ISBN 91-540-4025-6

Statens råd för byggnadsforskning, Stockholm

LiberTryck Stockholm 1983

## FÖRORD

Föreliggande rapport hänför sig till ett projekt benämnt "Tillförlitlighet hos datorberäkningar i byggnadskonstruktionsarbetet", utfört vid institutionen för Brobyggnad vid Kungliga Tekniska Högskolan i Stockholm, och med stöd från Statens Råd för Byggnadsforskning.

Arbetet bygger på det BFR-stödda projektet "Tillförlitlighet och kommunikation i en datoriserad byggbransch", redovisat i rapport R35:1982, utfört i samarbete med avdelningarna för Byggnadsmekanik och Bärande konstruktioner vid Tekniska Högskolan i Lund.

Forskningen har bedrivits parallellt med projekt vid de ovan nämnda avdelningarna vid LTH.

I rapport R35:1982 har sökt identifieras de möjligheter och risker som finns med ett ökat datoranvändande inom byggbranschen. I föreliggande projekt har intresset främst fokuserats på de risker som en långt driven datorisering av konstruktionsprocessen kan medföra.

De mest påtagliga riskerna vid datoranvändning grundas på brister i kommunikation och information. Botemedel mot dessa problem kan främst sökas i utbildning och dokumentation.

Rapportens huvudinriktning är att ge en översikt och en beskrivning av de metoder som används i dagens datorprogram. En fortsatt forskning skisseras också med avsikt att öka tillförlitligheten genom bland annat krav på beskrivning av program och beräkningsmetoder.

Målgruppen för läroboksdelen är främst de personer som har erfarenhet av byggnadskonstruktionsarbete, men med liten tidigare kontakt med datorberäkningar, vilket innebär att, förutom konstruktörer, även till exempel byggnadsinspektörer bör kunna ha nytta av den.

Rapporten skall i första hand ses som ett grupparbete mellan de tre författarna. Arbetet har dock uppdelats så att en person haft huvudansvaret för de olika kapitlen. Så har Leif Andersson främst arbetet med kapitlen 1, 9 och 11, medan Jan Ludvigsson ansvarat för kapitlen 3 och 7. För övriga kapitel och för den slutliga sammanställningen av rapporten svarar Anders Eriksson.

Rapporten har till största delen framställts med hjälp av textbehandlingssystemet EGG i NORD-100 datorn Valle vid Sektionen för Väg- och Vattenbyggnad vid Kungliga Tekniska Högskolan, kompletterad med maskinskrift utförd av Maj-Britt Eriksson och Suzanne Sköld.



En stor del av figurerna har ritats av Jan Ludvigsson.

Till dem som på detta konkreta sätt medverkat till rapportens tillkomst framför vi vårt tack.

Vi vill även framföra vårt tack till dem vid institutionerna på LTH och KTH, som bidragit genom stimulerande diskussioner.

Sist men inte minst är vi tacksamma för den hjälp vi fått från representanter för olika företag.

Stockholm, Maj 1983.

Anders Eriksson  
Leif Andersson  
Jan Ludvigsson

## INNEHÅLL

1.	INTRODUKTION .....	9
1.1	Datorprojektering .....	10
1.1.1	Tänkt verklighet	
1.1.2	Modell	
1.1.3	Indata	
1.1.4	Beräkning	
1.1.5	Utdata	
1.1.6	Redovisning	
1.1.7	Verklighet	
1.2	Personalkategorier i datorprojekteringen ..	17
1.2.1	Arkitekt	
1.2.2	Konstruktör	
1.2.3	Granskare	
1.2.4	Programmerare	
1.3	Nya och alternativa former av dator- projektering .....	19
2.	HÅRD- OCH MJUKVARA .....	21
2.1	Maskinvaran .....	21
2.2	Persondatorn .....	22
2.2.1	Processorn	
2.2.2	Primärminne	
2.2.3	Yttre minnen	
2.2.4	Utmatningsenheter	
2.2.5	Ett litet datorsystem	
2.3	Fleranvändarsystem .....	26
2.3.1	Anslutningsformer	
2.3.2	Operativsystem	
2.3.3	Kundidentifikation	
2.4	Representation av data .....	28
2.4.1	Text	
2.4.2	Instruktioner	
2.4.3	Heltal	
2.4.4	Flyttal	
2.4.5	Noggrannhet	
2.4.6	Inmatning av tal	
2.5	Programspråk .....	31
2.6	Filer .....	32
2.7	Olika körningsformer .....	33
3.	TILLGÄNGLIGA PROGRAM .....	37
3.1	Tillgänglighetsformer .....	37
3.1.1	Servicebyrå med fullservice	
3.1.2	Serviceföretag med kompletta arbetsplatser	
3.1.3	Terminal uppkopplad till stordator	
3.1.4	Egen dator	
3.1.5	Inköp eller inhyrning av program	
3.2	Information om datorprogram .....	42
3.2.1	Målgrupp och avsikt	
3.2.2	Innehåll	
3.2.3	Dokumentation	
3.3	Svenska programinventeringar .....	45
3.3.1	Byggforskningsrådet, 1972	
3.3.2	Byggdok, 1974	
3.3.3	Fredriksson & Mackerle, 1975	

3.3.4	STU, 1977	
3.3.5	Byggmästaren, 1977	
3.3.6	Svensk Byggtjänst, 1982	
3.3.7	ByggInfo, 1982	
3.4	Internationell utblick .....	52
3.4.1	Finland	
3.4.2	Norge	
3.4.3	England	
3.4.4	Väst-Tyskland	
3.4.5	USA	
4.	GRUNDERNA FÖR STATISKA BERÄKNINGSMETODER ..	57
4.1	Beskrivning av rörelser .....	57
4.2	Upplagsförhållanden och randvillkor .....	60
4.3	Utnyttjande av symmetri och repetition ....	62
4.3.1	Symmetri	
4.3.2	Repetition	
4.3.3	Antimetriska laster	
4.4	Kraftmetod och förskjutningsmetod, principer .....	65
4.4.1	Kraftmetoden	
4.4.2	Förskjutningsmetoden	
4.5	Kraftmetoden .....	71
4.5.1	Kontinuerlig balk, metod 1	
4.5.2	Kontinuerlig balk, metod 2	
4.5.3	Ramar	
4.6	Förskjutningsmetoden .....	75
4.6.1	Exempel på ramberäkning	
4.6.2	Elementarfall	
4.7	Jämförelse mellan metoderna .....	80
5.	FINITA ELEMENT METODEN .....	83
5.1	Elementanalys, balkstrukturer .....	84
5.2	Strukturanalys .....	90
5.3	Randvillkor, laster och lösning .....	94
5.4	Behandling av resultat .....	97
5.5	Beräkningsgång i FE-analys .....	99
5.6	Andra problemtyper .....	101
5.6.1	Teoretisk bakgrund	
5.6.2	Analys av skivproblem	
5.6.3	Analogier mellan olika problem	
5.7	Beräkningstekniska aspekter på FEM .....	109
5.8	Icke-linjära beräkningar .....	114
6.	PROGRAM FÖR ANALYS .....	115
6.1	Balkstrukturer .....	115
6.1.1	Kraftmetoder	
6.1.2	Förskjutningsmetoder	
6.1.3	Finite Element metoder	
6.1.4	Utbredda belastningar	
6.1.5	Variierande böjstyvheter	
6.1.6	Normalkrafters inverkan	
6.1.7	Dynamik	
6.1.8	Stångsystem	
6.2	Skivstrukturer .....	128
6.2.1	Typfall	
6.2.2	Olika elementtyper	
6.2.3	Isoparametriska element	
6.2.4	Resultatpresentation	



6.2.5	Lokala effekter	
6.2.6	Substrukturer och superelement	
6.3	Plattstrukturer .....	139
6.3.1	FEM för plattanalys	
6.3.2	Alternativa formuleringar	
6.3.3	Utbredda laster	
6.3.4	Ortotropa och förstyvade plattor	
6.3.5	Fourieranalys av plattor	
6.3.6	Brottlinjeteori för plattor	
6.4	Andra strukturer .....	148
6.4.1	Skalstrukturer	
6.4.2	Tre-dimensionella strukturer	
<b>7.</b>	<b>PROGRAM FÖR DIMENSIONERING .....</b>	<b>151</b>
7.1	Krav på strukturer .....	153
7.2	Materialanpassade program .....	157
7.2.1	Betong	
7.2.2	Stål	
7.2.3	Trä	
<b>8.</b>	<b>TEXTBEHANDLING OCH DATABASER .....</b>	<b>175</b>
8.1	Textbehandling .....	175
8.2	Databaser .....	177
8.2.1	Informationssökning	
8.2.2	Beräkningsdata	
8.3	Projekt databaser .....	182
<b>9.</b>	<b>PROGRAM FÖR DESIGN OCH RITNING .....</b>	<b>185</b>
9.1	Inledning och begreppsförklaringar .....	185
9.2	Existerande system .....	186
9.2.1	Generella system	
9.2.2	Speciella system	
9.2.3	Ritsystem	
9.3	Marknadsöversikt .....	196
9.3.1	Internationell utblick	
9.3.2	Svenska marknaden	
9.4	Användbarheten av CAD, en diskussion .....	199
9.4.1	Ritningsframställning	
9.4.2	Ekonomi och lönsamhet	
<b>10.</b>	<b>RISKER MED DATORANVÄNDNING .....</b>	<b>207</b>
10.1	Konstruktörens arbetssituation .....	207
10.1.1	Före datorisering	
10.1.2	Under datorisering	
10.1.3	Efter datorisering	
10.1.4	Utvecklingslinjer	
10.2	Informationsflöden .....	212
10.3	Felkällor .....	214
10.3.1	I programmeringsskedet	
10.3.2	I indata-skedet	
10.3.3	I beräkningsskedet	
10.3.4	I resultat-skedet	
10.4	Åtgärder mot felkällorna .....	227
10.4.1	I programmeringsskedet	
10.4.2	Vid programbeskrivandet	
10.4.3	Vid programanvändandet	
10.5	Målsättningar för fortsatt forskning .....	235

11.	TOTAL-SYSTEM, DISKUSSION OM EN VISION .....	237
11.1	Visioner .....	237
11.2	Diskussion om totalsystem .....	239
11.2.1	Behov och önskemål	
11.2.2	Existerande system	
11.2.3	Svårigheter med totalsystem	
11.2.4	Besluts- och kontrollmöjligheter	
11.3	Alternativa ideer .....	248
12.	LITTERATUR FÖR FORTSATT STUDIER .....	251
A.	APPENDIX - Matriser och ekvationssystem ...	257
A.1	Matriser	
A.2	Räkneregler för matriser	
A.3	Linjära ekvationssystem	
A.4	Lösning av ekvationssystem	

## 1 INTRODUKTION

I detta första introducerande kapitel skall vi tillåta oss att ge läsaren vår personliga syn på datorsituationen av idag. Vi kommer att vara, åtminstone delvis, kontroversiella och utnyttja forskarnas rättighet att vara "uppnosiga". Anledningen till detta är att vi i vissa avseenden upplever situationen som otillfredsställande och utvecklingen som litet planlös.

Under de senaste åren har datorer och datorprogram blivit allt mer vanliga hjälpmedel i konstruktionsarbetet. Antalet konstruktörer som i någon form kommer i kontakt med datorberäkningar ökar ständigt. Denna utveckling kommer sannolikt att fortsätta ett stort antal år framöver. Detta innebär att alla konstruktionskontor och konstruktörer snart är mer eller mindre beroende av olika datorhjälpmedel.

Trots denna intensiva utveckling är det sparsamt med litteratur som ser datoriseringen från konstruktörens sida. Det finns hyllmeter med litteratur riktad till programmerare men även riktad till konstruktörer som har datormetoder till specialitet. Till exempel, rörande FEM (Finita Element Metoder) finns det åtskilligt med rapporter som beskriver beräkningsmetodikens detaljer. De är dessutom oftast riktade till dem som kan tänkas skriva egna program. Litteratur som endast ger en metodförståelse och en övergripande bild av FEM saknas däremot nästan helt och hållet, se dock kapitel 12.

Detta innebär att kunskapen om datorer och datorprogram bitvis är bristfällig hos de flesta konstruktörer. Dessutom har det inte funnits, förrän helt nyligen, kurser om detta i utbildningen av ingenjörer och civilingenjörer.

Eftersom de datorbaserade hjälpmedlen ökar i användning samtidigt som kunskapen om dessa är begränsad finns det stora risker förknippade med användningen av dessa. I senare kapitel (10-11) kommer vissa risker och osäkra punkter att diskuteras mera ingående och strukturerat. Vi vill dock redan på ett tidigt stadium peka på några av dessa.

Eftersom vi ännu inte förutsätter några direkta dator-kunskaper är detta kapitel skrivet i en något "lättsam" form för att undvika alltför många specialtermer. Kapitellet skall därför främst ses som en inblick i ett komplicerat och relativt utforskat område. Det är dessutom avsett som en aptitretare för de efterföljande mera teoretiskt inriktade kapitlen.



## 1.1 Datorprojektering

En inblick i datoranvändning skall ges i form av ett enkelt exempel. Detta exempel beskriver en idag vanlig situation vid konstruktionsberäkningar med hjälp av dator. FIG 1.1 ger en bild av konstruktionsprocessens olika beståndsdelar och skall användas som diskussionsunderlag.

Projekteringen har uppdelats i 7 steg och förutsätter en situation där själva konstruktionsberäkningen utförs med hjälp av något datorprogram. Om konstruktionen, som till exempel i FIG 1.1, är en industrihall används lämpligen något ramverksprogram som hjälp vid dimensioneringen.

Givetvis är en verklig situation mycket mer komplex än vad FIG 1.1 antyder, men om man inriktar sig främst på datorprogrammets roll i konstruktionsarbetet så beskriver figuren projekteringsgången i stora drag. I kapitel 10 kommer dessutom FIG 1.1 att förfinas och vissa delar av den att detaljstuderas.

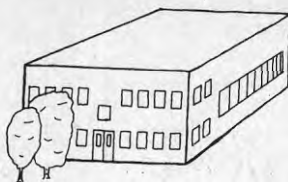
Om datorn används mer aktivt i projekteringen blir situationen något annorlunda. Vi skall beröra en sådan situation i slutet av detta kapitel, avsnitt 1.3. Detta är dock inte så vanligt idag vilket medför att vi riktar huvudintresset mot "vanliga" beräkningsprogram såsom visas i FIG 1.1.

Det kommer att förutsättas att för den i FIG 1.1 skisserade projekteringsgången datorprogrammen inte skrivs av konstruktören utan av en programmerare. Det stadium då programmeraren och konstruktören var en och samma person har därmed passerats. Idag och i framtiden kommer konstruktören själv endast att skriva små och enkla program, om några alls.

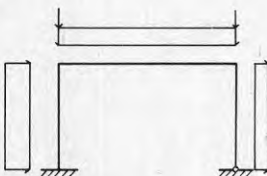
I vart och ett av de 7 stegen finns det en chans att något fel begås. Inom varje steg måste därmed eventuella felkällor identifieras och kontrolleras. Olika personalkategorier, t ex arkitekter och konstruktörer, medverkar i och är ansvariga för olika steg. De talar inte alltid "samma språk" vilket kan leda till missförstånd. Dessa brister i kommunikation gäller inom varje steg men framförallt mellan de olika stegen. Detta eftersom det är åtskilligt med information som skall överföras mellan de olika projekteringsstegen. Viktigt är att endast den för steget relevanta informationen överförs. Annars finns risken att man missar viktig information då den totala informationsmängden blir för stor.

Varje steg bör innehålla kontroll- och beslutspunkter. Kontrollpunkterna kan vara olika i form och omfattning, t ex kontroll av rimligheten hos det beräknade momentet i en bestämd punkt eller kontroll av en armeringsspecifikation. Med beslutspunkter menas situationer där beslut tas som bestämmer och begränsar

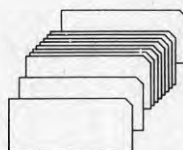
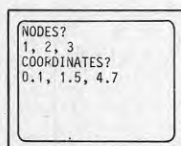
Tänkt verklighet



Modell



Indata

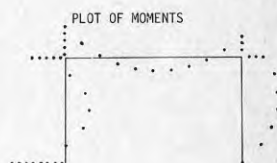
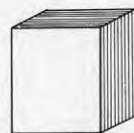
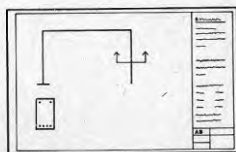


Beräkning

```
0101101110010110
1000101101100011
0110001101110101
0001001110100111
0111010011010100
```

Utdata

```
NODE DISPLACEMENTS
X Y Z XX
0.100 0.247 -0.163 ...
```

Resultat-  
redovisning

Verklighet

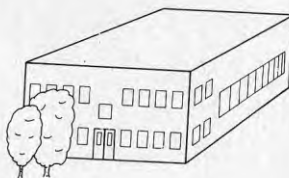


FIG 1.1 Successiva steg i en projektering där datorn används i konstruktionsarbetet

beräkningens omfattning. Det kan t ex vara att efter en enkel skiss besluta om konstruktionsmaterial och sektionens utformning. Det kan även vara mindre komplexa beslut till exempel val av armeringskvalitet.

Varje steg innehåller ett stort antal kontroll- och beslutspunkter och utfallet av dessa har en avgörande inverkan på fortsättningen av projektet. Det är sålunda viktigt att kontrollen utförs av tillräckligt kompetenta personer och att besluten tas av "rätt" person. Med detta avses en person som kan se det aktuella beslutet i sitt hela perspektiv och kunna uppskatta beslutets inverkan på projektets fortsättning.

Vi ska utgående från FIG 1.1 analysera varje delsteg och kortfattat diskutera hur projekteringen går ifrån tänkt verklighet till allt enklare och mer renodlade element för att sedan byggas upp igen till en allt mer komplicerad verklighet.

### 1.1.1 Tänkt verklighet

Låt oss börja med steget tänkt verklighet. Det är här projektets förutsättningar, restriktioner och begränsningar bestäms. Alternativa konstruktionstyper skisseras och diskuteras och i ett tidigt stadium allokeras personella resurser för det fortsatta projektarbetet.

Resultatet av detta förarbete ligger till grund för konstruktionsarbetet och åskådliggörs ofta med en arkitektskiss. I detta steg bestäms också, klart uttalat eller genom andra faktorer, många parametrar som till exempel höjder, bredder och spännvidder.

### 1.1.2 Modell

Utifrån skissen skall en beräkningsmodell uppställas och det är då som konstruktören på allvar kommer in i projektet. Detta steg är avgörande för att "verkligheten" skall fungera som den "tänkta verkligheten". Det är mycket viktigt att modellen och förutsättningarna beskrivs ingående för att underlätta nästa steg, indata, men även för att underlätta granskningen. Är modellen väl beskriven blir framtida ändringar enklare att utföra.

Vi använder ordet modell i vid bemärkelse. Översättningen av skissen till en streckfigur med leder, inspanningar o dyl ingår i detta steg. Men vi avser även sådana aspekter såsom val av typ av finita element eller om normalkraftens arbete skall beaktas eller ej.

Att på detta sätt inkludera metodval och angreppssätt i beräkningsmodelleringen innebär att modellsteget



innehåller ett stort antal och mycket viktiga besluts- punkter. Ett olämpligt val av en modellaspekt, t ex upplagsförhållanden, kan medföra att hela dimensioneringen blir felaktig. Några aspekter som måste beaktas är nedbrytning, geometri, randvillkor, teori och lastmodell. Med nedbrytning menas att den 3-dimensionella verkligheten indelas i mindre delelement, t ex ramar eller balkar, som sedan analyseras var för sig.

En del modellaspekters inverkan på beräkningsresultatet är relativt väl kända, medan andra är mer osäkra. Följaktligen utförs ibland modelleringen mer rutinmässigt än förståndsmässigt. I vilket fall som helst så finns det få klara riktlinjer för vissa av modell- aspekterna.

På grundval av modellen görs efterföljande beräkningar. Vissa lastfall och situationer behandlas förenklat och överslagsmässigt. Det är då vanligt att man ytterligare förenklar modellen. Detta kräver en erfaren konstruktör med känsla för problemet.

Modellen kan även behöva förenklas eller modifieras för att möjliggöra en utförlig analys. De första datorprogrammen för konstruktionsberäkningar var begränsat generella och krävde ett visst modellutseende. Dagens program är ofta betydligt mer flexibla. Till exempel om vi återigen tar ramen i FIG 1.1 som exempel så tillåter dagens ramprogram ofta fjädrande stöd och elastiska inspänningar. Detta innebär att man idag inte behöver förenkla och modifiera modellerna lika mycket som förr.

### 1.1.3 Indata

Att överföra modellen till datorprogrammet kräver att man känner till hur programmet vill ha och tolkar indata. För stora generella program kan indatagivandet vara omständigt och besvärligt. Om man känner dessa program väl kan man dock med lite knep och knåp beskriva nästan "vilken modell som helst". Detta kräver dock stor erfarenhet, och vana vid det aktuella programmet.

Det har emellertid lagts ner mycket arbete under de senaste åren på att förenkla indatahanteringen. Detta kan åskådliggöras som i FIG 1.2 som visar gårdagens sätt att beskriva indata jämfört med dagens sätt. Det finns dock ännu idag kvar många program som använder hålkort och där indata beskrivs nästan uteslutande med siffror i bestämda positioner. Likaledes fanns det även för några år sedan några program där indata skrivs mera läsbart. Med detta menas att text kopplas ihop med siffrorna, se FIG 1.2.

Naturligt nog görs det nu mindre fel i indataskrivningen och det är dessutom enklare att kontrollera att

man skrivit rätt. Ofta får man nu en direkt tolkning av varje indatarad som då antingen accepteras eller måste korrigeras. Tidigare var det vanligen så att alla indataraderna måste ges innan dessa tolkades av programmet, jämför avsnitt 2.7.

I en del av dagens program, företrädelsetvis de små och enkla programmen, består indatabeskrivningen av en dialog mellan datorn och användaren. Användaren svarar då på frågor som datorn ställer. För något större program är dialogen ofta kompletterad med speciella kommandon. Detta innebär att det krävs en mer aktiv medverkan av användaren.

Även om indata nu är enklare att skriva, krävs så gott som alltid av den som bestämmer indata, lämpligen konstruktören, att denne har läst en program- och indatabeskrivning. Detta medium är, förutom frågor, kontroller och felmeddelanden i programmet, ofta programmerarens enda möjlighet till kommunikation med konstruktören, se kapitel 10.

Det är först under senare hälften av 70-talet som någon betydande del av programmerarnas insatser varit inriktade på indatahantering. Detta innebär att många av dagens program- och indatabeskrivningar är ofullständiga och klumpiga, från användarens sida sett. Dessutom tillhör ju programmeraren ofta en annan yrkeskategori än användaren och talar därmed ett annat "språk" vilket innebär en stor risk för missförstånd och kommunikationsproblem.



((((NODNR FÖR FÖRSTA NODPUNKTEN PÅ LINJEN:	<u>1</u>
KOORDINATER X OCH Y:	<u>0,0.0</u>
ANTAL NODPUNKTER PÅ LINJEN:	<u>5</u>
DIFFERANSEN MELLAN NODNR:	<u>1</u>
DIFFERANSEN MELLAN X OCH Y KOORD:	<u>1,0</u>
ANTAL FG I NODEN:	<u>))2</u>

FIG 1.2 Olika former för indatagivning

Det är vidare väldigt svårt för en programmerare att på förhand förutse alla ideer som kretsar i huvudet på en användare. Kontaktytan mellan programmeraren och konstruktören, användaren, är som tidigare indikerats väldigt begränsad. Att man nu lägger ned stora insatser på att strukturera och undersöka indatahantering är ur tillförlitlighetssynpunkt glädjande.

#### 1.1.4 Beräkning

Själva datorberäkningen sker vanligtvis utan aktiv medverkan av konstruktören. Allt arbete i detta skede sker huvudsakligen i datorn. De fel som kan uppkomma i beräkningsskedet är oftast beroende på programmeringen. Anledningen till att vi tar upp programmeringsaspekter i denna bok för konstruktörer är dels att utvecklingen inom detta område har gått snabbt framåt och dels att det för en konstruktör är nödvändigt att veta vad man kan kräva av programmen.

Först skall konstateras att intresset för "god programmering" har vuxit på bekostnad av effektiv programmering. En effektiv programmering var ibland önskvärd förr, vid begränsad datorkapacitet. Med god programmering avses en programmering som är strukturerad och tydligt skriven. Programmen bör innehålla ett stort antal förtydligande kommentarer.

Skillnaden mellan en för problemet absolut nödvändig programmering och en god programmering är att en god programmering blir betydligt mer voluminös men trots detta mer läsbar och på grund av detta mer tillförlitlig. Kontroller, av t ex orimliga värden, är mycket önskvärda bland annat för att hjälpa konstruktören. Det finns flera krav som bör vara uppfyllda för en god programmering som t ex modularisering, indataeko osv. Dessa aspekter diskuteras något ytterligare i kapitlen 10 och 11 där kommunikationsproblemen mellan olika intressenter analyseras med avsikt att finna olika tänkbara felkällor i datorhanteringen.

Vi skall lämna avsnittet om programmering med att nämna att programmeringen skrivs i något datorspråk som FORTRAN eller liknande. Detta språk skall sedan omvandlas till instruktioner som datorn skall utföra. I och med detta berörs ytterligare en yrkesgrupp, de som konstruerar datorn och kompilatorerna. Det är dock sällsynt att fel i projekteringen orsakas av något fel på denna nivå. Noteras kan dock den för konstruktören obegripliga beskrivningen av konstruktionen, i form av en lång följd binära tal, till exempel:

01001000100011111001010101.....

### 1.1.5 Utdata

Beräkningsresultaten presenteras i nästa steg, utdata. Dessa resultat levererades förr som tjocka, ibland extremt tjocka datalistor. Dessa listor fyllda med siffror och åter siffror har fått många att skaka på huvudet och en del att sucka beundransfullt.

Oläsbarheten av dessa gjorde att man har satsat mer och mer på att förbättra utdataredovisningen. Förutom tydligare, bl a mer text, och selektiv utmatning av spänningar och moment eller dylikt, så har man satsat på grafisk redovisning. Talesättet att en bild säger mer än tusen sidor har myntats.

Den grafiska redovisningen används som kontroll av rimliga värden och den används för att ge en översiktlig bild av beräknat beteende. I många fall är det tillräckligt att detta utförs på det vanliga datalistepappret med ungefärliga värden utritade med stjärnor eller liknande. Numera är det också vanligt att noggrannare bilder uppritas på ritfilm av speciella ritmaskiner. Detta område, redovisning av utdata, utvecklas för närvarande mycket hastigt och är ett måste för en vidare spridning av datorkraften.

### 1.1.6 Redovisning

Alla beräkningsresultat och överväganden måste redovisas i slutliga handlingar. Dessa handlingar består bl a av ritningar, som i och för sig kan vara datorritade, och en beräkningsbeskrivning. I beskrivningen dokumenteras något om de datorprogram som har använts i projekteringen. Lämpligen ges en kort beskrivning av de metoder programmen bygger på och en hänvisning till utförligare programbeskrivningar.

Handlingarna är viktiga för att underlätta granskning. En beräkningsredovisning med resultat i form av några siffror skrivna av en programmerbar kalkylator utan redovisande av program, modell eller förutsättningar är givetvis helt oacceptabelt, men förekommer tyvärr likväl.

### 1.1.7 Verklighet

Sista steget, där konstruktören är mera passiv, är skapandet av verkligheten enligt ritningar och beskrivningar. Har allt fungerat väl, så får verkligheten samma funktion och samma utseende som arkitektens tänkta verklighet.

Utseende av och funktion hos "verkligheten", dvs resultatet av alla insatser, bör tas tillvara för en erfarenhetsåterföring. Speciellt viktigt är att de

misstag som gjorts utreds och att den nyvunna kunskapen kommer framtida projekt tillgodo.

## 1.2 Personalkategorier i datorprojekteringen

Som har framgått av beskrivningen ovan så har denna huvudsakligen inriktats på fyra olika personalkategorier : Arkitekter, Konstruktörer, Granskare och Programmerare. I FIG 1.3 redovisas de steg som de olika kategorierna medverkar i eller har stort intresse av.

### 1.2.1 Arkitekt

Arkitektens roll i projekteringen är huvudsakligen att skapa en väl fungerande och estetiskt tilltalande konstruktion. Eftersom det har förutsatts i FIG 1.1 att CAD inte används (jämför avsnitt 1.3) så blir därmed kopplingen arkitekt-datorberäkning ringa.

Den grad av datorisering som datorprogram för konstruktionsberäkningar har medfört påverkar inte direkt arkitektens arbetssituation. Emellertid borde det nu vara enklare att hållfasthetsberäkna mer komplicerade konstruktioner. Detta innebär att arkitektens frihet, i fråga om konstruktiv utformning, har vuxit något.

### 1.2.2 Konstruktör

Eftersom konstruktörer är den främsta målgruppen för denna rapport så har i beskrivningen ovan om datorprojektering konstruktörens roll redan berörts vid ett flertal tillfällen. De främsta skillnaderna mellan datorprojektering och traditionell projektering utan dator är att i datorprojekteringen konstruktören inte är verksam i beräkningssteget och att programmerarens roll har vuxit.

	ARKITTEKT	KONSTRUKTÖR	PROGRAMMERARE	GRANSKARE
TÄNKT VERKLIGHET	X	(X)		X
MODELL		X		X
INDATA		X	X	
BERÄKNING			X	
UTDATA		X	X	(X)
RES. REDOVISNING		X		X
VERKLIGHET	X	X		X

FIG 1.3 Olika intressenter i konstruktionsprocessen



Även om konstruktören nu inte är aktiv i beräkningssteget så är det ändå konstruktören som är ansvarig för resultatens riktighet. Detta är mycket viktigt att komma ihåg och det skall även diskuteras nedan, avsnitt 1.3.

Det är därmed viktigt att konstruktören inte blint litar på datorberäkningen utan att man alltid utför jämförande överslagsberäkningar. Erfarenheten visar dessutom att, om resultaten från en överslagsberäkning och från en datorberäkning väsentligt skiljer sig, det vanligtvis är något fel i datorberäkningen. Det kan då t ex vara fel i indata, fel användningsområde för datorprogrammet eller felprogrammering.

### 1.2.3 Granskare

I FIG 1.3 avser kategorien granskare främst granskare i form av en kontrollerande myndighet eller en kontroll utförd av någon utomstående firma. Om vi även tar hänsyn till interngranskning så bör givetvis även stegen indata och utdata infogas.

I den fortsatta diskussionen om granskningens roll skall endast externgranskningen beröras.

Det bör påpekas att en sådan granskning inte bör vara speciellt intresserad av själva detaljberäkningen. Intresset bör istället inriktas på modellens utseende och på att resultaten är rimliga.

Att granskningens huvuduppgift bör vara detta i stället för en siffergranskning är inget nytt och det gäller inte bara för datorberäkningar. Däremot har frågan aktualiserats i och med användandet av datorprogram i konstruktionsberäkningarna. Det är ju nästan omöjligt att siffergranska datorberäkningar. Dessutom borde gälla, om programmet är riktigt, att om beräkningsmodellen är lämpligt formulerad så blir också resultatet riktigt.

Om modellen är redovisad kan enkla kontroller och överslagsberäkningar utföras för hand eller med hjälp av något enkelt datorprogram.

Vid större och mer komplicerade beräkningar är oberoende granskningsberäkningar ett intressant alternativ. Detta innebär att de som granskar använder sina egna rutiner och sina egna datorprogram. På detta sätt minskas risken för att både konstruktör och granskare gör samma tankefel. Dessutom behöver granskarna inte ideligen lära sig, och sätta sig in i, nya datorprogram. Att ha en tillräcklig kunskap om t ex alla FE-program som finns på den svenska marknaden är redan idag ouppnåeligt för en enskild människa.

### 1.2.4 Programmerare

I FIG 1.3 skall noteras att programmeraren är ovetande om aktuell verklighet och om aktuell modell. Detta innebär att stora krav ställs på denne att i förhand ha en god uppfattning om i framtiden tänkbara modeller.

Programmerarna har traditionellt sett inte medverkat i konstruktionsarbetet och i samband därmed är deras inflytande på projekteringsarbetet inte så väl utforskat. Deras insatser är dock mycket avgörande för både resultatet och effektiviteten av konstruktionsberäkningarna. Att det inte forskats mer om programmerarnas roll är faktiskt något förvånande.

### 1.3 Nya och alternativa former av datorprojektering

Totalintegrerade system, där extremfallet skulle kunna vara kravspecifikation in - ritning och beskrivning ut, har tillsammans med CAD blivit dagens mode inom datorbranschen. Stora resurser och mycket kapital används för närvarande inom dessa områden.

Med CAD (Computer Aided Design) avser vi en relativ strikt tolkning av de ingående orden. Vi anser inte att programsystem som endast används för att framställa ritningar ingår i begreppet CAD. Tonvikten borde snarare ligga på ordet "design". Inom datorbranschen finns annars ett stort antal olika "definitioner". På svenska används ibland istället ordet datorstöd. Dessutom finns det i samband med CAD ett stort antal ytterligare "mystiska" beteckningar som till exempel CAM, CADD o s v, vilka ytterligare något diskuteras i kapitel 9.

Givetvis kan CAD vara till stor nytta inom många områden men det är definitivt inget undermedel. Vi skall inte närmare diskutera detta här men vill varna för reklamens budskap om : flexibilitet, kreativitet och ekonomi.

Om man använder sig av totalsystem (en alternativ benämning på totalintegrerade system) blir projekteringsgången annorlunda än den i avsnitt 1.1 beskrivna. Den väsentligaste skillnaden består i att man vidgar programmerarens insatser. Därför överförs allt fler besluts- och kontrollpunkter på programmeraren och datorn genom att konstruktörens kontrollmöjligheter mellan stegen försvinner.

Ansvaret får dock fortfarande helt och hållet tas av konstruktören. Detta är värt att hålla i minnet när man diskuterar datorsystemens förträfflighet.

Totalsystem innebär alltså att stora, kanske för stora, krav ställs på programmeraren. Dels måste denne vara

en god programmerare. Systemen blir nämligen, av nödvändighet, stora om de skall vara flexibla och användbara, vilket kräver en strukturerad och genomtänkt programmering. Det finns åtskilliga exempel på stora system som övergivits för att programmet varit oläsbart och rörigt. Dessutom måste programmeraren vara en god konstruktör. Konstruktörskänslan krävs för att förutse lämpliga alternativ och att göra optimala val samt att anpassa indata och utdata till konstruktörens önskemål. Att skriva system som upplevs som flexibla och inte håller sig till ett begränsat antal standardlösningar är mycket svårt.

I detta avsnitt har vi skilt på program för ritningsframställning, design och på totalintegrerade system. Dessa delar kommer att behandlas utförligare i kapitel 9 och i kapitel 11.

Denna rapport lägger tonvikten på rena beräkningsprogram. Dock har i detta kapitel även andra delar som berör datoriseringen behandlats. Likaledes kommer även i fortsättningen delar som inte har något direkt samband med beräkningsprogram, t ex texthantering, att genomgå. Denna genomgång blir dock bitvis ganska ytlig, men är nödvändig för att man skall få en korrekt bild av dagens datoriseringsprocess. Dessutom kan förhoppningsvis genomgången medverka till att dagens "mystiska skimmer" kring datorerna försvinner.

Det är viktigt att vi, när vi i fortsättningen diskuterar olika typer av datorprogram, har kedjan mellan "tänkt verklighet" och "verklighet" i bakhuvudet. De delar som i fortsättningen skall diskuteras är de delar där kommunikationen konstruktör - programmerare - granskare är mest framträdande, dvs stegen modell, indata, beräkning och utdata i FIG 1.1 .

Innan detta behandlas utförligare måste dock en översiktlig genomgång av datorer och kringutrustning göras.

## 2 HÅRD- OCH MJUKVARA

I det första kapitlet har vi diskuterat ett tänkt exempel på en datorberäkning för en byggnadskonstruktion. Trots att exemplet beskrevs ganska ytligt kunde inte vissa hänvisningar undgås till ett antal begrepp som flitigt används inom datorbranschen. För att söka förklara dessa skall i detta kapitel ett antal termer behandlas som har att göra med datorarbetet, begrepp som ofta dyker upp i resonemang med datorfolk och som lätt skapar onödiga bekymmer för den som inte är så insatt i terminologin.

I första delen av detta kapitel skall maskinvaran (hårdvaran med en försvenskning av den engelska beteckningen) det vill säga datorn och dess kringutrustning beskrivas. I de senare avsnitten skall ett antal begrepp, som snarare skall betraktas som hörande till programvarusidan (mjukvara), diskuteras.

För att inte detta skall leda alltför långt, skall vi hålla beskrivningen på en tämligen elementär nivå, den som har intressen som sträcker sig längre hänvisas till speciallitteraturen på området, se avsnitt 12.2.

### 2.1 Maskinvaran

Den vanligaste indelningsgrunden för datorer är stor-, mini- och mikrodatorer eller, det senaste tillskottet i benämningväg, superdatorer. Begreppen försöker, med ett enda ord, säga något om datorns storlek och kapacitet. Begreppen är dock ganska luddiga, och har olika innebörd för olika personer och i olika sammanhang.

Med den avsikt som detta kapitel har är dock en annan indelningsgrund mera ändamålsenlig, eftersom den är baserad på datorns funktion ur användarens synvinkel. Med detta som bakgrund vill vi alltså hellre dela in datorerna i två andra grupper - dels den (lilla) dator som en person i taget kan använda, dels den (stora) dator där flera delar de tillgängliga resurserna.

I denna genomgång av ett datorsystems beståndsdelar kan det vara lämpligt att börja med att betrakta den lilla datorn, som ju ofta innehåller ett absolut minimum av komponenter. I den stora datorn tillkommer sedan en mängd olika komponenter för speciella tillämpningar och funktioner. Det krävs också en stor mängd speciell programvara för att kunna hantera flera olika användare samtidigt.

En reservation skall dock genast läggas in. Med hänsyn till den oerhört snabba utvecklingen inom datorområdet kommer de uppgifter om kapaciteter och dylikt som ges

snart att vara inaktuella, även om de försöker återge förhållandena under 1983.

## 2.2 Persondatorn

För att beskriva vad som är absolut nödvändigt i fråga om utrustning i en dator skall alltså först persondatorn behandlas. Eftersom denna alltid hanteras av endast en person i taget, så blir ju de olika komponenternas funktioner mera renodlade.

Från användarens synpunkt består persondatorn främst av två enheter, en tangentbordsdel och en bildskärm. All kommunikation med datorn sker via tangentbordet, och användaren är hela tiden i full kontakt med datorn och dess funktioner.

### 2.2.1 Processorn

Den centrala och viktigaste, om än inte den till volymen största, delen av datorn är dess hjärna, processorn (CPU eller Central Processing Unit på engelska). Processorn innehåller bland annat en aritmetisk-logisk enhet som utför de beräkningar och den databehandling som behövs i en datorkörning, och register i vilka vissa data kan hållas. Processorn kan bara utföra mycket enkla operationer i sig själv, addera/subtrahera och jämföra tal samt utföra vissa logiska funktioner. Dessa enkla operationer utförs dock mycket snabbt. Processorn i en typisk persondator adderar exempelvis två heltal (med belopp mindre än ca 32000) på - i storleksordningen - 5 mikrosekunder (förutsatt att de två talen tidigare placerats i processorns register).

### 2.2.2 Primärminne

Processorn kan endast behandla ett visst antal data i taget, nämligen så många som ryms i dess register. En mindre processor kanske har åtta stycken åtta-bitsregister, medan en större har flera. För att ha tillgång till alla data som åtgår i en viss beräkning måste därför datorn även bestå av ett minne. Den del av minnet som processorn har direkt tillgång till kallas primärminne eller snabbminne (den gamla benämningen kärnminne lever även kvar, även om primärminnena nuförtiden ej är uppbyggda av ferrit-kärnor).

Från och till primärminnet kan processorn utbyta data på någon eller några mikrosekunder, och primärminnet kan därför ses som en utbyggnad av processorn. I primärminnet lagras såväl de tal som ingår i beräkningen som de instruktioner som processorn skall utföra, d v s programmet kodat i en form som processorn kan tol-



ka, allt i form av binära tal. En del om hur denna lagring går till beskrivs i avsnitt 2.4.

Primärminnet kan inte vara hur stort som helst, dels på grund av att denna minnesform är förhållandevis dyr, dels beroende på att processorn inte kan hantera hur stora minnen som helst. För många små datorer (de som utnyttjar 16-bits processorer) är primärminnets storlek av denna anledning begränsat till 64 kilobyte. (Vi bortser här från möjligheterna att genom avancerad programmering samtidigt hantera flera olika "banker" av denna storlek.)

Den använda enheten, kilobyte, är ett mått på informationsmängd och 1 kilobyte = 1024 bytes motsvarar 1024 tecken (bokstav/siffra e dyl, jämför avsnitt 2.4.1). Om primärminnet skulle användas för lagring av information i form av en text skulle alltså den ovan nämnda primärminnesstorleken räcka till ca 25 sidor bestående av 50 rader om 50 tecken på varje. Alternativt räcker utrymmet till lagring av ca 30 000 heltal eller ca 12 000 flyttal (jämför avsnitt 2.4.3 och 2.4.4).

Av det totala maximala utrymmet om 64 kilobyte är dock endast en del tillgängligt för användaren, då en stor del (ofta upp mot hälften) används för datorns interna arbete, bland annat en viss del ROM-minne ("Read only") för exempelvis datorns BASIC-system. En tillgänglig primärminneskapacitet i form av RAM-minne ("Random access") om 16 eller 32 kilobyte är därför det vanliga i de persondatorer som finns på marknaden.

Observera dock att detta, som gäller 1983, förmodligen inte längre är sant år 1985 då introducerandet av 32-bits mikroprocessorer och de ständigt sjunkande priserna på elektronikkomponenter kan förväntas helt ha förändrat förhållandena.

Primärminnet i en dator har dock en annan, och kanske inte lika lättövertärlig, begränsning. Den del av datorns primärminne som används för lagring av det egna programmet och dess data försvinner nämligen då datorn stängs av. På grund av detta, och på grund av den även på sikt- begränsade primärminnestillgången finns alltså behov av någon annan form av minne för datorn, där bestående lagring i litet större skala kan ske.

### 2.2.3 Yttre minnen

För persondatorn finns två principiellt olika typer av yttre minnen, magnetband och magnetiserade skivor.

Den billigaste, men minst behändiga, formen är magnetbandet, för persondatorn ofta av samma form som musik-kassetter. Fördelen med dessa är att de är billiga och enkla att använda samt att banden är relativt tåliga mot yttre påverkan.

Banden har dock en uppenbar olägenhet i sin långsamhet. Att hitta ett visst program som finns lagrat på bandet och läsa in det i datorn kan ta flera minuter, för vissa datortyper.

Riskerna att information försvinner till följd av miss-tag eller tekniska fel vid in- och utläsning är också stora. Detta gäller speciellt för vissa typer av smådatorer, där datorn inte har någon avancerad kontakt med bandenheten, och därför till exempel inte märker om något band ej finns ilagt då ett försök till utläsning till bandet göres. Vissa datortyper gör dock en kontroll av den information som läses ut varför dessa risker ej är lika stora.

Om man ser till att man alltid har två kopior på sina program sparade, kan band av denna form ändå vara en acceptabel lösning då det gäller permanent lagring av program. För lagring av stora datamängder är de dock inte lämpliga. Detta gäller speciellt för datorer med den ovan nämnda dåliga kontrollen över dataöverföringen.

En bättre, men också betydligt dyrare, typ av yttre minne är skivminnet, för den lilla datorn oftast i form av en "floppy-disc".

Denna typ av yttre minne har klara fördelar jämfört med band bland annat på grund av sin snabbhet, som gör att ett nytt program läses in på, i storleksordningen, ett par sekunder. Datorn har också normalt en betydligt bättre kontroll på in- och utläsningarna till dessa enheter, vilket medför att riskerna att information förstörs vid användandet är avsevärt mindre. Det medför också att ett skivminne är lämpligare att använda för lagring av data för ett program, om primärminnet inte är tillräckligt stort.

Minnes-skivor av typen "floppy-disc" är löstagbara ur skivminnesenheten, och ser ut ungefär som en liten, mjuk grammofonskiva inlagd i ett pappersfodral. Minneskapaciteten är av storleksordningen 100-400 kbyte. (40 a 160 A4-sidor text).

Under den senaste tiden har små datorer även försetts med "riktiga" skivminnesenheter av samma typ som för stora datorer. Skivorna för dessa enheter är inte löstagbara i samma utsträckning som de tidigare nämnda (även om man kan byta dem med en viss arbetsinsats). Lagringskapaciteten för en enhet rör sig om 5-250 Mbyte (2000 a 100 000 A4-sidor)

#### 2.2.4 Utmatningsenheter

Som tidigare nämnts består persondatorn oftast av ett tangentbord från vilket information matas in till processorn, och en bildskärm där resultaten presenteras.

Då man önskar köra litet större program blir det emellertid snart obekvämt att få ut alla resultat på bildskärmen, och ur dessa sortera fram de intressanta. Bättre är då att till datorn koppla en skrivare (printer) där väsentliga resultat kan fås ut på papper. Sådana skrivare finns i många olika utformningar, och därför olika prislägen. Den generella sanningen är att dyra skrivare skriver snyggare bokstäver, och/eller skriver snabbare än billiga.

Kapaciteten för en måttligt dyr skrivare är 30 - 200 tecken per sekund medan en stor radskrivare klarar upp till 30 rader om 130 tecken per sekund.

En annan indelningsgrund för skrivare är sättet på vilket de framställer tecknen. En skönskrivare har till exempel typer på samma sätt som en skrivmaskin, medan matrissskrivaren bygger upp tecknen genom att sätta ett antal punkter på pappret. Detta innebär i och för sig att matrissskrivaren, med rätt programmering, kan fås att skriva vilka tecken som helst (smala, breda, grekiska o s v), men också att den skrivna texten blir suddigare och tecknen mindre välformade. Radskrivare skriver visserligen med typer men å andra sidan blir raderna inte helt raka, varför texten ändå inte blir perfekt.

En annan intressant utmatningsenhet är plottern, där man i bildform kan presentera de resultat som datorn beräknat. I många fall är denna redovisningsform klart överlägsen den skrivna texten, och - om inte annat - ett utmärkt komplement till resultat i sifferform, se också avsnitt 6.2.4. En enkel utrustning är inte heller speciellt dyr.

Bilder kan också produceras på grafiska terminaler där, förutom vanlig text, även kurvor och annan grafisk information kan presenteras med acceptabel noggrannhet. Dessa terminaler är dock fortfarande mycket dyra.

I samband med detta skall digitaliseringsbord beröras. Dessa ger en möjlighet att med hjälp av en ljuspenna överföra en bild eller ritning till siffer-data som datorn kan hantera.

### 2.2.5 Ett litet datorsystem

I de tidigare avsnitten har beståndsdelarna i ett litet datorsystem beskrivits. I FIG 2.1 visas nu ett diagram över hur hela systemet kan vara sammansatt. Ett system av denna typ är, i princip, tillräckligt omfattande för att kunna hantera de flesta vanliga beräkningar som görs på ett konstruktionskontor.

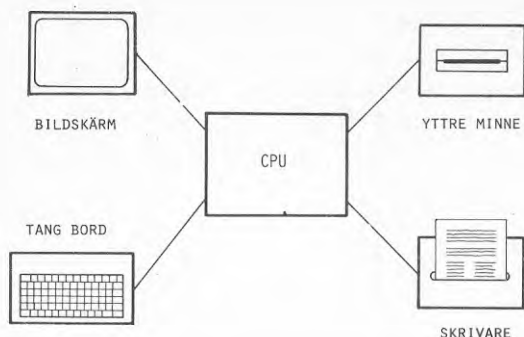


FIG 2.1 Ett enkelt datorsystem.

### 2.3 Fleranvändarsystem

I föregående avsnitt har beståndsdelarna i ett persondatorsystem behandlats. Samma komponenter ingår även i de större datorerna, även om de i dessa fall är snabbare och har betydligt större kapacitet. Dessutom finns ofta mera speciella komponenter som hålremsstan-sar, magnetbandsstationer samt olika typer av skrivare och ritmaskiner.

Det som dock i detta sammanhang främst är intressant i jämförelse med persondatorerna är möjligheterna för flera användare att dela datorns resurser, och de ändringar i arbetssätt som detta medför.

#### 2.3.1 Anslutningsformer

När det gällde persondatorn så var användaren via bildskärm och tangentbord hela tiden i fullständig, såväl fysisk som logisk, kontakt med datorn och dess processor. Detta gäller inte för fleranvändarsystemet. Normalt sitter man i stället vid en av de terminaler som är anslutna till datorn. Om man vill kunna använda datorn från en avlägsen plats kan anslutningen ske med hjälp av telefonnätet. Terminalen är i så fall kopplad till en telefon via ett modem, som omvandlar de elektriska signaler som datorn och terminalen utväxlar, till signaler som kan befordras över telenätet.

Om man har en terminal och ett modem kan man på detta sätt, genom att slå olika telefonnummer, nå ett stort antal datorer.

En viss nackdel med denna typ av telefonkopplade förbindelser kan sägas vara den hastighet i överföringen



som kan åstadkommas. Med den billigaste typen av modem är denna nämligen begränsad till 300 baud (motsvarar ungefär 30 tecken/sekund), vilket är förhållandevis långsamt vid överföring av exempelvis mer omfattande resultatutskrifter. En dyrare typ av modem kan ge en hastighet av 1200 baud med utnyttjande av vanliga telefonlinjer, medan man för att uppnå 9600 baud i överföringen måste ha en fast ledning till datorn.

### 2.3.2 Operativsystem

Även i de stora datorerna finns (normalt) bara en processorenhet. Detta innebär att då flera användare samtidigt är uppkopplade mot datorn, de på något sätt måste dela på processorns kapacitet. Delandet sker ofta med hjälp av så kallad "time-sharing" vilket innebär att processorn arbetar med en användares problem i taget, under korta tidsrymder. Då denna tid är slut sparas de aktuella resultaten undan, och en annan användare får sig litet tid tilldelad. På detta sätt cirkulerar processorn mellan de olika användarnas beräkningar, och eftersom varje intervall är så kort, upplever samtliga användare att de har full tillgång till datorns uppmärksamhet.

Fördelningen av tiden mellan olika användare sköts av operativsystemet som kan sägas vara ett överordnat program som hela tiden körs i datorn. Förutom att fördela tiden mellan olika användare har det även till uppgift att övervaka all verksamhet i datorn och kontrollera vilka enheter som är lediga och så vidare.

### 2.3.3 Kundidentifikationer

I och med att flera användare gemensamt använder en anläggning aktualiseras frågan om hur kostnaderna skall betalas. Det gäller alltså för operativsystemet i datorn att hålla reda på hur mycket av de olika resurserna - processortid, utskrifter, uppkopplad tid o s v - de olika användarna nyttjar.

För att en sådan debitering skall kunna genomföras, måste varje användare vara känd av datorn (eller tillhöra en grupp som är känd). För den skull får varje (grupp av) användare i ett datorsystem en kundidentifikation i form av endera ett namn eller en kombination av bokstäver och siffror. Denna identifikation måste alltid ges då körning mot datorn påbörjas. För att höja säkerheten mot obehörigt utnyttjande kompletteras ofta denna identifikation med någon hemlig lösenkod som måste ges exempelvis vid inloggning av en terminal.

I operativsystemet finns också andra spärrar mot intrång. Exempelvis skyddas oftast filer i datorns skiv-



minne med hjälp av koder och identifikationer med avsikt att omöjliggöra för andra än den som skapat filen att läsa dess innehåll (eller åtminstone att förstöra det).

## 2.4 Representation av data

Som tidigare nämnts är datorns processor endast kapabel att hantera information i form av binär kod. Detta gäller för såväl det program som körs som de (siffer-) data som programmet använder.

Den binära informationen kan, ur mänsklig synvinkel, ses som en följd av 0:or och 1:or men är fysiskt sett olika spänningsnivåer.

Den minsta informationsmängden som förekommer i datorn kan alltså betraktas som en binär siffra (=bit, från binary digit, men numera även med svenskt uttal).

För att förenkla hanteringen i, och beskrivning av, datorn vill man dock gärna använda sig av ett litet större begrepp än dessa "enhets-informationer". Därför talar man, när man skall beskriva minneskapacitet o dyl, i stället oftast om en "byte" (eng uttal) som omfattar åtta binära siffror. Normalt är detta också den minsta informationsmängd som datorn kan flytta mellan register och minnen.

Förutom dessa två allmänt använda begrepp används dessutom ofta beteckningen ord (word). Denna informationsmängd är dock varierande för olika datorfabrikat - exempelvis 16 bitar för NORD-datorer och 32 bitar för IBM.

### 2.4.1 Text

Informationsmängden i en byte (8 binära siffror) räcker till att bilda 256 olika tal (0-255 med normal representation). Med tanke på att man vid dataöverföring o dyl ofta vill använda en binär siffra för kontrolländamål (paritetskontroll) räcker därför en byte lagom till för att representera våra bokstäver (stora och små), siffror och övriga tecken. Detta görs genom en standardiserad kodning där varje bokstav representeras av ett tal.

Tyvärre finns det flera olika sådana system. Åtminstone två är vanliga nämligen EBCDIC (används av bl a IBM) och ASCII (t ex i DEC-datorer och i de flesta mikrodata-torer). Som exempel kan nämnas att i ASCII-systemet tecknet "A" representeras av talet 65 (binärt: 01000001), tecknet "g" av 103 (01100111) och "1" av 49 (00110001). I EBCDIC fås helt andra tal.



Eftersom representationen är olika i olika datorer kan ingen exakt gräns för noggrannheten ges. Som tumregel gäller dock att taldelen  $p$  kan bestämmas med 6 decimalers noggrannhet då 32 bitar används och med 8-9 då 48 används för representationen av flyttal, och att denna relativa noggrannhet även gäller för det aktuella flyttalet.

Detta innebär att, i en dator där 32 bitar används för representationen av flyttal, talen  $0.1234556 \cdot 10^{12}$  och  $0.1234564 \cdot 10^{12}$  kommer att kunna representeras på samma sätt. Skillnaden kan, om man betraktar detta enda tal, synas obetydlig. Faktum är dock att denna noggrannhet i representationen ofta är för dålig när man skall lösa de stora ekvationssystem som exempelvis Finita Element metoden ger upphov till. För att klara detta problem finns i större datorer en möjlighet att använda så kallad dubbel precision för flyttal. Med denna variant får man, till kostnad av en betydligt längre beräkningstid, en mycket noggrannare representation av talen. Denna möjlighet finns dock ej i de flesta persondatorer.

#### 2.4.6 Inmatning av tal

Som ovan visats behandlar processorn fyra olika slag av information, nämligen programinstruktioner, text, heltal och flyttal. Samtliga dessa typer lagras i form av binära tal och det är endast sammanhanget som avgör hur datorn uppfattar en viss byte i primärminnet.

Den normale datoranvändaren behöver emellertid inte känna till så mycket om detta. Förutom den begränsade noggrannheten i representationen av flyttal är det dock en sak som bör noteras.

Eftersom datorn skiljer mellan heltal och flyttal måste man även som användare skilja mellan dessa typer vid inmatning av data till en del program. Det gäller alltså att förstå skillnaden mellan heltalet 5 (t ex antalet fack i en kontinuerlig balk) och flyttalet 5.0 (längden för ett av facken). För många äldre program måste denna skillnad markeras i indata genom att talen skrives som "5" resp "5.0" (eventuellt räcker "5.").

Man måste även vid formuleringen av indata för olika program ta reda på hur data skall formateras. För äldre program, åtminstone de som är skrivna i FORTRAN, är indata kolumnberoende vilket innebär att varje sifferuppgift måste hamna exakt i de kolumner som programmeraren specificerat för att inte alla indata skall tolkas på fel sätt. Denna restriktion borde emellertid inte längre behövas då de flesta datorer numera har så kallat fritt format i inmatningen, vilket innebär att olika sifferuppgifter kan åtskiljas med, exempelvis, komma-tecken.

Observera också att de flesta (alla?) datorer använder decimalpunkt och inte decimalkomma.

## 2.5 Programspråk

Processorn arbetar, som tidigare nämnts, endast med mycket elementära operationer. Att formulera sin beräkningsgång i dessa termer är närmast omöjligt för någorlunda komplexa problem, även om det i princip skulle vara möjligt att endast mata in binära sifferkoder direkt i maskinens primärminne, och starta exekvering (körning) av ett sådant program. ( Nästan denna typ används dock för vissa delar av program där det är viktigt att åstadkomma maximal beräkningshastighet, med hjälp av s k maskinnära programmering, eller assembler-programmering)

För den skull har man skapat programspråk för formulering av beräkningsalgoritmer. Med hjälp av dessa kan man skapa sitt program med hjälp av vissa bestämda formuleringar där engelska ord är vanliga ingredienser. Med hjälp av speciella program i datorn översätts där efter dessa, någorlunda människo-orienterade formuleringar till de sifferkoder processorn begriper.

Det enklaste programspråket är BASIC. Detta språk förekommer, tyvärr i ganska skiftande utformningar, i alla datorer, och har hittills varit det enda tillgängliga språket i persondatorer. Ett enkelt exempel på BASIC-programmering är

```
10 LET A=3
20 LET B=2
30 LET C=A+B
40 PRINT "A+B=",C
50 STOP
```

vars innebörd rätt tydligt framgår av texten.

För program av denna svårighetsgrad är valet av programspråk oväsentligt, och BASIC är lämpligt p g a sin lätthet att hantera. I mera komplicerade beräkningar blir dock BASIC för ineffektivt för att vara ett lämpligt språk.

Det särklassigt vanligaste språket i tekniska tillämpningar är FORTRAN. Fördelarna med detta språk är, förutom att det är välkänt, att det ger effektiva program, att det ger goda möjligheter till snygga resultatutskrifter och att språket är - åtminstone i en grundversion - standardiserat, så att program relativt lätt kan flyttas mellan olika datorer. Nackdelen med språket sägs ofta vara svårigheten att strukturera program på ett bra sätt, vilket gör programmen svårlästa. Det förekommer också ett antal dialekter i språket, i form

av olika finesser som, utanför den standardiserade grunden, finns tillgängliga i olika datorer. Språket har också vissa brister då det gäller att hantera text-information. Språket finns numera tillgängligt även på vissa persondatorer.

För administrativa program i, främst, stora datorer är språket COBOL nästan totalt dominerande.

En mängd andra programspråk existerar vid sidan av dessa. De har dock haft svårt att få någon riktig genomslagskraft, utom för speciella tillämpningar. De två nya språken PASCAL och ADA tros dock, av många, kunna hota de ovan nämnda då det gäller utbredning. Speciellt tros detta gälla ADA som tillkommit som ett försök att standardisera all databehandling inom USA:s förvaltning.

Det program som på detta sätt byggs upp skall som ovan nämnts översättas till maskinkod. Detta görs med hjälp av en kompilator, vilket är ett program som översätter exempelvis ett FORTRAN-program till den följd av binära tal som datorn kan hantera. (Denna beskrivning gäller normalt inte för ett BASIC-program som i stället byggs upp interaktivt med hjälp av ett interpretator-program).

För att ur denna binära kod få fram ett färdigt körbart program skall den av programmeraren skapade, nu kompilerade, koden kompletteras med ett antal systemrutiner. Dessa innefattar hanteringen av in- och utmatning, koder för hur standardfunktioner som sinus och cosinus bildas samt en del överordnad kod. De steg som gör detta kallas länkning och laddning.

## 2.6 Filer

Begreppet filer har använts vid flera tillfällen redan. Vad innebär då detta begrepp? En fil är generellt en samling information som lagras på ett eller annat sätt. I datorsammanhang tänker man kanske främst på skivminnesfiler, ävenom såväl en trave hålkort som en inspelning på en kassetten kan betecknas som filer. Filen är oftast ett medel att spara information mellan olika körningar, men kan också vara av temporär natur så att information sparas och återhämtas i samma datorkörning.

Den sparade informationen kan vara i olika form. Exempelvis lagras program ofta som filer uttryckta i klarspråk, det vill säga som en följd satser i FORTRAN eller något annat programspråk. Program kan emellertid även lagras i sin kompilerade och laddade version, det vill säga den till binär-kod översatta formen, vilket möjliggör snabbare körningar. (Det är oftast i denna form man kommer i kontakt med färdiga program.) Slutligen kan även siffer- eller textdata lagras i olika filer till exempel i datorns skivminne.



En användning av filer som är mycket vanlig är skapandet av indatafilerna för olika program. Dessa utgörs av ett antal rader med kommandon och sifferdata till något program man önskar köra. Även om man gör sin körning från terminalen så beskrivs dessa indata ofta som kortbilder, d v s rader om maximalt 80 tecken.

Filerna kan skapas endera av olika program som skriver ut data eller, då det gäller exempelvis program, med hjälp av så kallade "editor"-program, med vilka man från sin terminal kan bygga upp en text och sedan spara den som en fil.

Arbetet med editorerna styrs med hjälp av speciella "språk". Tråkigt nog är dessa inte alls standardiserade mellan olika datorer, utan varje dator har sina egenheter i detta avseende, vilket kan leda till problem om man växelvis arbetar med flera datorer.

Speciellt till stora Finita Element program har speciella program utvecklats för indatagenerering, man talar om pre-processorer. Dessa bygger ofta på att användaren i dialog, eller med hjälp av hårkorset i en grafisk bildskärm, får specificera sin beräkningsmodell, varefter datorn omsätter denna information till de exakta indata som det aktuella programmet vill få. Dessa indataprogram behöver inte kopplas till ett speciellt beräkningsprogram utan kan vara generellt användbara. Den individuella anknypningen till ett bestämt program fås då genom omformning av den databas som indataprogrammet genererat till en indatafil.

Filer, databaser och editorer (textbehandlingsprogram) behandlas utförligare i kapitel 8,

## 2.7 Olika körningsformer

I de tidigare avsnitten har, mer eller mindre klart utsagt, förutsatts att arbetet med datorn sker från en terminal, det vill säga att man har direkt kontakt med datorn. Detta möjliggör ett interaktivt arbetssätt där man styr sitt arbete med hjälp av datorns reaktioner på den gjorda inmatningen. Detta arbetssätt har utvecklats relativt snabbt under senare år.

Tidigare har arbetet baserats på kortbuntar. Man har preparerat en hel kortbunt i en följd och sedan kört in den som en "batch"-körning i datorn, ett begrepp som innebär att jobbet körs som en helhet utan möjlighet till ingripande från användaren. Om något då varit fel på det första hålkortet, så har ändå hela kortbuntens behandlings och de resultat som kommit ut varit helt felaktiga.

Utvecklingen av terminaler och datorernas terminalhantering har alltså möjliggjort ett betydligt bättre arbetssätt. Fördelarna är allra störst vid uttestning

av program och beräkningsmodeller där det är av stort värde att kunna avbryta och ändra om något verkar gå fel. Normalt kostar dock detta interaktiva arbetssätt betydligt mer än batch-körningar, dels till följd av att timkostnaden för en inloggad terminal är hög, dels till följd av att man för batch-körningar kan acceptera en något lägre prioritet för bearbetningen och därför använda spilltid i datorn.

För körning av stora program är batch-körningar fortfarande dominerande, även om de "fysiska" kortbuntarna i många fall ersatts med indatafiler i kortbilsformat. I det senare fallet kan alltså uppbyggandet av indata ske interaktivt, medan själva beräkningen sker i batch-miljö.

Många större beräkningsprogram har under senare år omformats till att arbeta interaktivt genom att den tidigare använda kortbilsinmatningen ersatts med en dialog, där datorn ställer frågor (i mer eller mindre klartext) då den önskar inmatning av vissa data. Detta är ett mycket användarvänligt sätt att arbeta, och kan spara tid genom att risken för fel i indata minskas. För stora, generella, program kan dock detta arbetssätt bli mycket arbetsamt då man i varje läge kan ha många tänkbara alternativ att välja mellan, varav kanske endast ett litet antal skulle kunna vara relevanta för det aktuella problemet.

Vartefter utvecklingen av texthanteringen i de vanliga programspråken gått framåt har också intresset riktats på möjligheterna att kommandostyra programmen. Iden bygger på att användaren, med kommandon i klartext, skall kunna styra programexekveringen till vissa relevanta delar av programmet. När man kommit till rätt del kan resten av indatagivningen ske i dialogform.

För att detta skall fungera måste såväl datorn som användaren i varje läge ha (eller kunna få) klart för sig vilka alternativ till kommandon som erbjuds. Den aktuella listan av giltiga kommandon brukar ofta kallas meny.

Ett exempel på kommandostyrning är det monitor-program man träffar på då man loggar in sin terminal till datorn. Man har då valet att ge kommandon av typen att starta editorprogram, kompilator eller laddar-program, eller till exempel att beordra fram en uppgift om vad klockan är.

En ytterligare förenkling av indatahanteringen kan i vissa fall nås genom användning av så kallade macros. Dessa kan ses som en sammanfattning av ett antal kommandon i en viss ordning, exempelvis en samling av kommandon som tillsammans utgör en vanlig typ av körning. På detta sätt kan användaren alltså ytterligare minska den behövliga inmatningen.

Hur man på ett optimalt sätt skall utforma kommunikationen mellan dator och användare har länge varit ett eftersatt område för utveckling, då all möda lagts på utveckling av alltmer avancerade beräkningsmetoder. Som skall visas i kapitel 10 kan dock denna aspekt ha ett stort intresse ur tillförlitlighetssynpunkt, och behovet av forskning och utveckling inom detta delområde är därför stort.

Efter att nu ha behandlat litet allmänna datorkunskaper i ett kapitel skall vi övergå till att ge en beskrivning av programvara för tillämpningar inom byggbranschen. Vi återkommer dock till den allmänna datororienteringen i kapitlen 8 och 9 som skall behandla några mera avancerade datortillämpningar i samband med framställning av texter och ritningar.



### 3 TILLGÄNGLIGA PROGRAM

När vi vet litet mera om hård- och mjukvaran är det dags att överväga på vilket sätt vi skall föra in data-tekniken i det egna företaget. Det fordras omfattande studier av företagets organisation, för att de fördelar man eftersträvar skall uppnås. Ett misstag kan betyda att datorresursen inte används på ett sätt som är till gagn för företaget.

Företagets datamognad bör analyseras, det vill säga den allmänna kunskapsnivån och attityden till datorstöd bland den berörda personalen. Om man i förhållande till datamognaden har en för högt ställd ambitionsnivå, finns risker att datorresursen endast utnyttjas av ett fåtal anställda i företaget, och dessutom till låg grad. Detta innebär naturligtvis att effektiviteten blir låg och att de mål och förväntningar man hade, kanske aldrig infrias.

Vidare är det angeläget att man kartlägger företagets arbetsmetoder och utreder hur, och i vilken mån en datorisering kommer att förändra dessa.

Varje företag och dess situation är unik och kräver en individuell analys. Att i detalj gå in på denna skulle leda för långt i detta sammanhang. Därför lämnar vi åt var och en att själv analysera det egna företagets situation, och nöjer oss med att ge några allmänna synpunkter.

I detta kapitel skall vi fortsättningsvis diskutera tillgänglighetsformerna för hård- och mjukvara, behandla programbeskrivningar och dessas innehåll samt presentera några svenska programinventeringar. Därefter skall vi ge exempel på vad som görs utanför Sveriges gränser i fråga om sammanställningar av tillgängliga program o dyl.

#### 3.1 Tillgänglighetsformer

I detta avsnitt skall vi diskutera hur man kan få tillgång till datorresurser. Dels kommer vi att behandla olika sätt att utnyttja servicebyråernas olika möjligheter dels alternativet egen dator.

##### 3.1.1 Servicebyrå med fullservice

Vi nämnde inledningsvis i detta kapitel att man måste beakta en mängd olika faktorer, innan man kan ta beslut om en satsning på datoranvändning. För att minska de initiella kostnaderna kan man anlita en servicebyrå med fullservice. Detta innebär att det är servicebyrån



som, i uppdragsform, sköter hela arbetet med datorn utgående från vissa grunddata som konstruktören måste ange, till exempel grundmått, materialmodeller o dyl.

Med ett sådant arrangemang behåller man en stor del av sin handlingsfrihet för framtiden, eftersom man inte bundit något kapital i datorutrustning eller programvara. Samtidigt drar man fördel av det relativt stora programutbud man erbjuder och av att beräkningarna i många fall utförs på stora datorer, vilket ger större möjligheter att prova olika resurser.

Lösningen innebär att konstruktören själv inte kommer i direkt kontakt med själva datorberäkningen, annat än via en modellbeskrivning och resultaten. Hans del i detta skede består i att förbereda och förse "beräkningsspecialisten" med de data som är nödvändiga för att specialisten skall kunna skapa en riktig beräkningsmodell. Servicebyrån sköter sedan, förutom själva skapandet av beräkningsmodellen, alla administrativa göromål inför datorberäkningen, t ex ifyllandet av indatablanketter eller kanske vanligare inskrivandet av dessa direkt via terminal.

Från det att man lämnar ifrån sig ett uppdrag till det att man erhåller resultatet kan det förflyta en till flera dagar. Detta kan i vissa tillämpningar vara en besvärande lång svarstid.

En annan nackdel med denna lösning är att konstruktören tappar kontrollen över själva beräkningen. Fel kan då insmyga sig vid indatagenereringen, p g a att "beräkningsspecialisten" inte förstår de byggnadstekniska antaganden och förutsättningar som gäller.

För att konstruktören skall kunna ta ansvar för den utförda datorberäkningens resultat är det således viktigt att inte enbart kontrollera resultatet av beräkningen, utan också de indata och den beräkningsmodell den grundar sig på.

### 3.1.2 Serviceföretag med kompletta arbetsplatser

Vissa serviceföretag och konsulter erbjuder en något annorlunda arbetsform, en komplett utrustad arbetsplats på serviceföretagets kontor. Arbetsplatsen är utrustad med alla de faciliteter man kan behöva, terminaler, skrivare, grafiska skärmar, plotter, etc. Kundföretaget får mot en överenskommen hyra, själv arbeta med den tillgängliga maskin- och programvaran.

Möjlighet till konsultationer erbjuds vanligen. Ofta är det enbart datatekniska problem som man kan hjälpa till med, men på vissa serviceföretag kan man även få hjälp med byggnadstekniska frågor.

Man arbetar vanligen vid en terminal som står i direkt kontakt med datorn. Detta gör det möjligt att arbeta interaktivt.

Kundföretagets vinst med detta arrangemang blir att man slipper binda kapital i egen datorutrustning, samtidigt som man vinner värdefulla erfarenheter av datorprojektering. Erfarenheter som senare kan visa sig vara av stort värde vid en eventuell framtida investering i egen datorkraft.

Det finns dock nackdelar med denna lösning. En av dessa är att det ofta inte är möjligt att låta all personal komma i direktkontakt med datorbearbetningen, vilket kan försvåra överväganden om att utöka datorverksamheten.

För att det över huvud taget skall vara möjligt att arbeta på detta sätt, måste serviceföretaget ligga inom en nära omgivning. Dessutom krävs en viss planering av verksamheten för att undvika onödigt resande. Detta kan dock kanske accepteras om databehandlingsmängden är måttlig.

Denna form av uthyrning av kompletta arbetsplatser, är inte särskilt vanligt förekommande ännu. Vissa anser dock att det är en modell för framtiden. De små företagen har svårt att hävda sig på den kapitalintensiva datormarknaden och riskerar därför att slås ut. Med denna lösning har de dock möjlighet att, till en rimlig kostnad, få tillgång till den moderna datortekniken och därmed kunna konkurrera med de stora företagen som har egna datorer.

### 3.1.3 Terminal uppkopplad till stordator

Om man inte tycker att lösningarna som behandlats i avsnitten 3.1.1 och 3.1.2 uppfyller de krav man ställer på flexibilitet och smidighet i datoranvändandet, finns ytterligare en lösning där man utnyttjar serviceföretagets datorresurser.

Man kan antingen köpa eller hyra en egen terminal (eller en mikrodator med möjlighet att användas som terminal) och koppla upp den mot en dator, placerad på servicebolagets kontor. Anslutningsformerna diskuteras i avsnitt 2.3.1.

Allt efter det att de egna behoven ändras kan man komplettera terminalen med den nödvändiga kringutrustningen. Med denna lösning kan man således utforma en arbetsplats som svarar mot det egna företagets speciella behov, samtidigt som man slipper de riktigt stora kapitalkostnader det innebär att anskaffa en egen dator och programvara till den.

En stor fördel framför de i avsnitten 3.1.1 och 3.1.2 behandlade lösningarna, är att datorkraften kommer närmare användaren. Personalen kommer oftare i kontakt med den och kan lära sig att använda den succesivt. En annan påtaglig fördel är att man kan använda resursen när det passar en själv, man får inga onödiga väntetider. Dessutom finns möjligheten att komma i kontakt med fler datorer via telenätet. På så sätt blir inte företaget bundet till att använda ett företags programvara utan får tillgång till ett mycket bredare sortiment av program. Hur man kan hitta vad som finns tillgängligt i form av programvara skall behandlas i avsnitt 3.3.

#### 3.1.4 Egen dator

Om man vill gå ytterligare ett steg längre kan man skaffa egen dator. För den enskilde konstruktören skiljer sig inte denna lösning avsevärt från den som behandlades i det föregående avsnittet, vad gäller själva beräkningsförfarandet. Skillnaderna uppstår vid anslutningen. När man arbetade med serviceföretagets dator skedde detta via telenätet, vilket medför att överföringshastigheten kan bli relativt låg jämfört med den som kan uppnås med en dator placerad på det egna kontoret. I övrigt är rutinerna likartade.

Som vi berört lite i kapitel 2.1, finns på marknaden ett utbud av datorer som spänner från mikro-, via minidatorer till de modernaste super-datorerna. Beroende på vilken man väljer på denna skala innebär det mer eller mindre omfattande investeringskostnader. Trenden är för närvarande att priserna på hårdvaran sjunker, medan de stiger på mjukvara. Denna trend kan förväntas hålla i sig.

Till de initiella kostnaderna måste man också lägga framtida kostnader, t ex för underhåll dels av maskin- och dels av programvaran, samt eventuella investeringar i nya maskiner. Man har således kostnader som sträcker sig långt in i framtiden och är därför svåra att överblicka. Man drabbas också av nackdelen att inte lika lätt få tillgång till nyutvecklade programvara.

Mikrodatorerna säljes ofta i "paket", där både mjuk och hårdvara ingår. Mini- och stordatorer saluförs oftare komponentvis. Det vill säga, man kan plocka ihop ett system bestående av t ex terminaler, plotter, bandstation, skivminne, etc. Detta kräver att man besitter goda kunskaper på området. Har man inte dessa är det lämpligt att anlita konsult.

Datorresursen kan utformas så att den har endast de för uppgiften nödvändiga komponenterna. Men komponentuppbyggnaden gör det också möjligt att komplettera med nya komponenter när behov för detta uppstått. Detta gör datorn mera flexibel och den kan på så vis anpassas till framtida behov.

Bland de fördelar man uppnår med en mini- eller stor-dator framför en mikrodator är naturligtvis större kapacitet. Denna större kapacitet kan dels utnyttjas för att lösa större problem, men också till att använda en mera användarvänlig programmering. Det senare innebär att man anpassar kommunikationen med datorn till det mänskliga språket. Man har också bättre möjligheter att ansluta diverse kringutrustning till datorn.

En annan påtaglig fördel är att utbudet av tillämpningsprogram är större än för smådatorerna, och att man får ett fleranvändarsystem, med alla de fördelar som detta medför.

### 3.1.5 Inköp eller inhyrning av program

En del kommentarer till smådatorbranschens utbud av programvara är på sin plats. Om man studerar de programförteckningar som företagen som säljer smådatorer tillhandahåller, märker man relativt snart att branschen som sådan inte är, i någon större utsträckning, inriktad på byggföretagen. Tekniska beräkningsprogram finns, men huvuddelen av utbudet behandlar andra områden. Vanligtvis brukar en programförteckning innehålla några program för materialadministration, textbehandlingssystem, order- och kundreskontra, lönerutiner och en del program för projektledning, uppföljning och så några byggnadstekniska beräkningsprogram.

En anledning till att det finns relativt få beräkningsprogram, kan vara att man på smådatorföretagen ofta saknar egen personal med teknisk utbildning. Man anlitar istället en fristående konsult, som gör beräkningsprogram på beställning av ett kundföretag.

En annan anledning till bristen på bygginriktade beräkningsprogram för smådatorer kan vara att beräkningarna kräver stor minneskapacitet. Detta kapacitetsbehov har inte kunnat tillgodoses, men inom en inte alltför avlägsen framtid kan nog denna begränsning vara undanröjd med dagens snabba utvecklingstakt.

De system som datorförsäljarna tillhandahåller säljs vanligen som standardprogram. Detta har naturligtvis sina fördelar i att man kan erbjuda programvara till ett relativt lågt pris. Företaget blir tvunget att anpassa sig till detta program, och hurvida detta innebär positiva konsekvenser eller inte måste analyseras.

Det är också ganska vanligt förekommande att fristående programmeringskonsulter säljer egna program genom smådatorföretagen. Denna konstruktion kan vara till nackdel för ett kundföretag som vill få ett fel i ett program rättat eller behöver hjälp vid handhavandet av programmet. Risker finns då att kunden blir hänvisad mellan de olika konsulterna och på så vis inte får den hjälp som behövs.



För de stora datorcentralerna och vid användning av en egen stordator är dessa problem mindre, då det nämligen för stora datorer finns ett relativt stort utbud av bra program för olika tillämpningar. I samband med en större datoranläggning är det också fullt möjligt att lägga in dessa goda, men tyvärr ofta ganska dyra program.

Om man inte vill göra ett eget program finns det alltså goda möjligheter att köpa eller hyra ett. Hur man finner ett lämpligt sådant program skall diskuteras i avsnitt 3.3, men dessförinnan skall vi titta lite närmare på vad en programbeskrivning är och vilka krav man kan ställa på den.

### 3.2 Information om datorprogram

När datorn introducerades på konstruktionskontoren i slutet av femtio-talet var det vanligen användaren, konstruktören, som själv tog fram den programvara som behövdes för de beräkningar som skulle utföras. Detta sätt att arbeta, programmerare och användare i en och samma person, ställde inga eller små krav på dokumentation av programmen. Användaren var specialist på det egna programmet.

Sedan dess har både mjukvaran och speciellt hårdvaran undergått en explosionsartad utveckling. Priserna har i takt med utvecklingen sjunkit på hårdvaran medan mjukvaran blivit allt dyrare. Detta har inneburit att förhållandet mellan vad som investeras i hårdvara och mjukvara för tekniska beräkningar förändrats, så att numera betydligt större andel läggs på programvara.

Detta förhållande har betytt att det blir allt vanligare att man hyr eller köper programvara. Det är för dyrt att bekosta egen utveckling. Detta har i sin tur skapat ett behov av att beskriva program på ett sådant sätt att utomstående kan använda dem.

En fullständig programbeskrivning i alla aspekter blir i regel tämligen omfattande. Det är därför lämpligt att viss information bryts ut och presenteras i en programinformation. Syftet med denna skall vara att ge en utomstående en kort men ändå så fullständig information, att han kan avgöra om ett program förtjänar ytterligare undersökning eller om sökandet skall fortsätta.

När man med hjälp av programinformationen funnit ett lämpligt program har man sedan behov att få reda på hur programmet kan användas. I det läget är man hjälpt av en programdokumentation som mer i detalj beskriver programmets förutsättningar och begränsningar.



### 3.2.1 Målgrupp och avsikt

Den kategori man först tänker på är kanske konstruktörerna. Det är ju dessa som vet vilket problem som skall lösas och vilken beräkningsmetod som är lämplig för detta. Dessa kunskaper ligger till grund för de beräkningstekniska krav som ställs på programmet.

Men det är inte enbart dessa som konstruktören måste beakta innan själva beräkningen sättes igång. Det är nödvändigt att man även tar hänsyn till ekonomiska aspekter. Det är kanske billigare att utföra beräkningen för hand. Svårigheten ligger ofta i att bestämma priset för en datorberäkning. För detta behöver den ovane datoranvändaren hjälp, t ex i form av en uppgift i programbeskrivningen som anger en ungefärlig beräkningskostnad för en typisk beräkning.

När man väl hittat ett program som stämmer in på de facktekniska kraven gäller det att ta reda på om programmet, utan större förändringar, kan användas på den tillgängliga datorn. Det är inte troligt att en sådan bedömning kan göras av konstruktören själv, utan han kanske måste ha en datorkunnig till hjälp. Här har vi då ännu en grupp som kan tänkas vara hjälpta av en programinformation.

En tredje grupp som kan behöva en programinformation är granskarna. Det kan t ex vara en myndighetsperson på någon lokal byggnadsnämnd, som vill kontrollera att konstruktören har använt ett program som är applicerbart på beräkningen. Det kan också tänkas att han själv vill hitta ett annat program att använda för att jämföra resultaten.

### 3.2.2 Innehåll

Med ledning av vad vi sagt ovan kan tre huvuddelar i en beskrivning vara lämpliga: fackteknisk, datateknisk och en ekonomisk/administrativ del. Med dessa tre som utgångspunkt skall vi ge några kommentarer till de programkataloger som presenteras i avsnitt 3.3.

Om man ser på de tre första programkatalogerna, publicerade mellan 1972 och 1975, kan man finna rubriker som går att föra in under fack- respektive datateknisk beskrivning. Ingen av dem har dock några uppgifter om ekonomin. I de två senaste finns dock rubriker som behandlar denna del. Tyvärr tvingas man konstatera att det påfallande ofta saknas information under dessa. Även om man inte presenterar ett pris, borde ett exempel på t ex använd CPU-tid för ett specificerat problem kunna ges. Detta underlättar för användaren att uppskatta en ungefärlig kostnad för en körning.

Vi har i avsnittets inledning påpekat vikten av att program är väl dokumenterade. Detta avgör ett programs

användbarhet för en utomstående användare. Därför kan det tyckas vara lämpligt om det i informationen anges hur programmet är dokumenterat. I de flesta av inventeringarna saknas sådan information. Tyvärr kan man i Svensk Byggtjänsts inventering från 1982 (jämför avsnitt 3.3.6) finna påståenden i den facktekniska beskrivningen som "God dokumentation", för att sedan under "Systembeskrivning" och "Användarhandbok" finna att sådana inte finns.

Som en sammanfattning av de redovisade inventeringarna kan sägas att programinformationen ofta är så kortfattad att det är svårt, ibland omöjligt, att dra slutsatser om programmets lämplighet för ett visst ändamål.

### 3.2.3 Dokumentation

Som vi nämnde redan i inledningen till detta avsnitt, har inte dokumentationen av datorprogrammen undergått samma snabba utveckling som datatekniken i övrigt. Det är först på senare år man börjat inse betydelsen av god dokumentation. Främst utvecklingen mot att man köper och hyr program har bidragit till detta. En annan orsak är att man uppmärksammat faran med att endast en enda eller kanske ett par personer på företaget "kan" programmen. Om någon av dessa slutar, finns risken att investeringar i ett programsystem går till spillo. Att i efterhand försöka sätta sig in i hur ett illa dokumenterat program fungerar, kan bli en mycket tidskrävande process och kostar därmed mycket pengar. Följden kan då bli att man istället "skrotar" programmet.

Anledningen till att dokumentationen blivit eftersatt, kan förmodligen vara att dokumentationsarbetet har setts som ett nödvändigt ont. Man skjuter därför på detta i det längsta och i många fall gör man den i efterhand. Eftersom programmeringsarbetet för ett större program kan sträcka sig över flera år hinner man glömma en hel del innan det är dags att dokumentera vad man gjort. Risken finns då att man glömt tankegången bakom och att dokumentationen blir ofullständig. I stället bör man lägga upp en plan för dokumentationen redan innan man börjar arbeta med programmeringen. Revidering sker sedan under arbetets gång.

Dokumentationen skall inte enbart ge användaren information om hur programmet skall användas. Här skall också finnas sådana uppgifter att en konstruktör kan förstå den beräkningsgång som programmet följer. Förutsättningar, begränsningar, utförlig beskrivning av beräkningsmodellen är andra uppgifter som bör finnas med i en beräkningsteknisk dokumentation.

Vidare skall dokumentationen vända sig till andra programmerare som skall rätta fel eller göra ändringar i programmet. För detta ändamål måste en utförlig systembeskrivning finnas, som förklarar den filosofi den ur-

sprunglige programmeraren haft.

För att minska risken att allvarliga misstag begås, på grund av en oklar eller ofullständig dokumentation, bör man ta fram riktlinjer på hur dessa skall vara beskaffade. I Canada har man utarbetat en (icke bindande) rekommendation för detta - någon sådan finns däremot inte i Sverige. Varje enskild programmerare har sin egen uppfattning. Detta förhållande gör det svårt att för en utomstående hitta den relevanta information som man behöver.

Det finns således ett både stort men också akut behov av forskning på detta område - ett behov som betydligt utförligare kommer att diskuteras i kapitel 10.

### 3.3 Svenska programinventeringar

I detta avsnitt skall en del svenska programinventeringar presenteras. Rubrikerna syftar på utgivare och det år de är publicerade.

#### 3.3.1 Byggforskningsrådet, 1972

Byggforskningsrådet bildade 1970 en ADB-grupp. Denna fick som sin första uppgift att sammanställa en inventering av datorprogram med anknytning till byggbranschen. Inventeringen publicerades 1972 under namnet "Byggnadstekniska dataprogram", B3:1972.

Uppgifterna i inventeringen grundar sig på en rundfråging man lät göra till ett stort antal företag samt till högskolor och statliga verk.

Avsikten var att förmedla kontakt mellan programägare och användare av datorprogram. Kontakten har förmedlats genom att man för varje program har angivit programägare med adress och telefonnummer. Man genomförde inte någon värdering eller kontroll av programmen.

Inventeringen omfattar cirka 800 program, uppdelade på 31 för byggbranschen specifika ämnesområden, t ex balk, betong, stål, dynamik, datalagring, etc.

Programsammanställningen är uppställd i tabellform, se utdraget i FIG 3.1.

Följande tabellhuvud används:

- 1) Löpnummer
- 2) Programnamn
- 3) Funktion
- 4) Programspråk
- 5) Dator (erforderlig minneskapacitet etc)
- 6) Upplysning om programmet lämnas av.

767	"KANI"	Flexibelt ramberäkningsprogram				Kjessler och Mannerstråle AB, M. Widerberg, Box 27131, 102 52 Stockholm 08/22 42 00
806	BK 305 RAM	Beräknar böjmoment och tvärkrafter för kontinuerliga balkar och envåningsramar	Fortran	55K		Brokonsult AB, L. Pålsson Vretnavägen 8, 171 54 Solna 08/98 17 45
808	BK 307 REKT TVÄRSN -SKEV- BÖJNING	Beräknar maximala betong- och armeringsspänningar i ett rektangulärt betongvärsnitt utsatt för skev böjning och normalkraft, samt beräknar enligt betongbestämmelserna tillåten normalkraft	Fortran	55K		dito
820	B:101	Enkelsymmetrisk II-sektion. Sektionsdata	Basic	Bull-GE		AIB, A. Wannerberg Box 5511, 114 85 Stockholm 08/63 00 20

### FIG 3.1 Utdrag ur Byggeforskningsrådets inventering

Under punkt 4 och speciellt under punkt 5 saknas ofta upplysningar.

Varje program beskrivs mycket kortfattat. Man får endast upplysning om att ett program existerar, samt att det utför vissa beräkningar. Upplysningar om vilken beräkningsmodell som används eller om vilka begränsningar som programmet har finns endast för vissa program. Vilka indata som programmet kräver samt vilka utdata som genereras framgår i regel ej.

Uppgifterna om själva programmen är väl kortfattade. De är så knapphändiga att det är svårt att fastställa om ett program är lämpligt eller inte för det problem man vill lösa.

#### 3.3.2 Byggdok, 1974

Denna inventering, benämnd Byggnadstekniska Datorprogram, är en uppföljning av Byggeforskningsrådets utgåva B3:1972. Inventerings- och sammanställningsarbetet har utförts av Byggeforskningsrådets ADB-grupp, men Byggdok är ansvariga för utgivningen.

Den omarbetade upplagan innehåller uppgifter om ca 900 program, uppdelade på 30 ämnesområden. Jämfört med BFR's utgåva är 40 % av programmen nytillkomna. Presentationerna är utformade efter samma mönster.

Rapporten är mera komplett vad avser uppgifter om programspråk och erforderlig datorutrustning. Man har dessutom markerat program som har tillkommit med hjälp av byggeforskningsanslag. Dessa är allmänt tillgängliga att nyttjas fritt av envar.

Sist i rapporten finns en förteckning över andra kända programkataloger från andra branscher och länder.

### 3.3.3 Fredriksson & Mackerle, 1975

Rapporten, med namnet "Structural Mechanics Finite Element Computer Programs - Surveys and Availability", är utgiven vid Linköpings tekniska högskola i september 1975 och är reviderad ett år senare. Ytterligare en utgåva med samma titel utkom 1980.

Enbart Finita Element program behandlas. Man har dels sådana program som är tillgängliga i Sverige, dels sådana program som är tillgängliga utomlands. Var programmen är tillgängliga kan man utläsa av den tabellframställning som inleder rapporten.

Man har indelat programmen i fem områden, (linjära elastostatiska, icke-linjära statiska, linjära dynamiska, icke-linjära dynamiska och värmeledning). Ett och samma program kan förekomma under en eller flera rubriker.

Rapporten är sorterad i bokstavsordning efter programmens namn. Programmen, och deras funktion, presenteras i tabellform, FIG 3.2.

Bland de egenskaper/funktioner som redovisas finns uppgifter om programmets karaktär, analysområden, beräkningsförutsättningar (lastegenskaper, elementtyper, etc), uppgifter om handhavandet av programmet samt om de krav som ställs på hårdvaran.

Sammanställningen omfattar cirka 270 program varav cirka en tredjedel är tillgängliga i Sverige. Uppgifterna om varje program är relativt fullständiga. Att samma program förekommer på fyra olika sidor, gör dock informationen lite svåröverskådlig.

Sist i rapporten redogörs för hur man kan erhålla mera information om programmen.

FE- PROGRAM CHARACTERISTIC	pr. name												
	1 GAP	2 GASP-2	6 GENESYS	3: GENFEM	4 GIFTS	5 GOLTA	7 GWU-FAP	8 GEOFEM	9 GNATS	10 GPLA	11 GENFEM-D		
GENERAL STRUCTURES			●	●	●								
STRUCTURAL MEMBERS	●	●	●	●	●	●	●	●					
THICK SHELLS				●									
THIN SHELLS				●						●			
SHIP STRUCTURES				●	●								
BUILDING AND CIVIL ENG.			●				●		●				
PIPING SYSTEMS													
OTHER								●					

FIG 3.2 Utdrag ur Fredriksson & Mackerle



### 3.3.4 STU, 1977

Rapporten är utgiven under rubriken "Finita element-metoden: Användningsområden, organisation av programbibliotek samt behov av samverkande utvecklingsinsatser".

Denna rapport passar egentligen inte in under kapitelrubriken, men vi tycker att initiativet är så intressant att det motiverar att det tas upp här.

Projektet som finansierades av STU, har haft som mål att kartlägga behovet av FEM-program och gemensamma utvecklingsinsatser.

Rapporten bygger på en intervjuundersökning av utvalda företag och institutioner. Man valde ut sådana där man visste att FEM-program fanns. Inga mindre eller medelstora företag intervjuades. Man ansåg att dessas synpunkter och behov framkom ändå, genom att de anlitar de stora företagen för sina FE-körningar.

Inga enskilda program redovisas i rapporten. I stället redogörs allmänt för de olika företagens och institutionernas FE-kapacitet, önskemål och utvecklingstenden- ser.

Författarna har som bilaga till rapporten lagt fram egna funderingar om hur ett nationellt programsystem för FEM bör se ut. Eftersom utveckling av FEM-program- vara är en mycket tidskrävande och kostsam process anser man att gemensamma utvecklingsinsatser behövs. Man har i intervjuundersökningen mött i stort sett positiva reaktioner på denna ide.

Gruppens tankar har vidareutvecklats i ett nytt STU- projekt, "Samordnat specifikationsprojekt inom FEM-områ- det".

Man har i detta studerat hur befintlig och nyutvecklad programvara bättre kan nyttiggöras och spridas, genom hjälpmedel som gör det lättare att kombinera och modi- fiera programmen. På så sätt kan man använda existeran- de programvara i nya tillämpningar och stimulera ut- byte av program mellan användare av olika kategorier.

Man har funnit att databastekniken är lämplig för det- ta ändamål. Denna teknik och STU-projektets slutsatser behandlas vidare i kapitel 8.

### 3.3.5 Byggmästaren, 1977

Arbetsgruppen för "Use of Computers in Structural Engi- neering" inom IABSE (International Association for Bridge and Structural Engineering) rekommenderade sina medlemmar att verka för en publicering av programresu- meer i lokala tekniska tidskrifter, liksom den man gjort

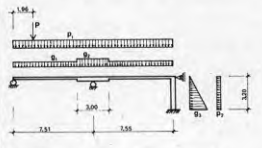

i en väst-tysk tidskrift "Bauwirtschaft", se avsnitt 3.4.4. För att främja ett internationellt utbyte i framtiden, rekommenderade man att så nära som möjligt söka efterlikna de resumer som presenterats i Bauwirtschaft.

I Sverige togs idén upp av tidskriften Byggmästaren (BM) på initiativ av Åke Bengtsson som svensk representant i IABSE-gruppen. Den första resumen publicerades 1977 i Byggmästarens decembernummer.

Man hade i princip tre målgrupper:

- ingenjörskåren i stort, som en orientering om utvecklingstendenser,
- ingenjörer med aktuella beräkningsproblem,
- företag med egna datorer som önskar förvärva tillgång till programvara, än så länge i relativt begränsad omfattning i Sverige, har internationellt en växande betydelse.

De programinnehavare som önskade få sina program publicerade, uppmanades att från BM rekvirera den standardiserade blankett som skulle användas. Sedan man returnerat den ifyllda blanketten fick man, mot en kostnad av 550 kronor, resumen införd i BM.

<b>Byggmästaren</b>		Byggmästarens Hönummer: 1
<b>PROGRAMRESUME • Software-Information</b>		Publiceringsmånad: December 1977
<b>0.1 Titel:</b> Snölastberäkning för kontinuerliga balkar och isolerande snöbeläggningar		
<b>0.2 Programnamn:</b> NADB-2200 Ramberäkning II		
<b>0.3 Programförfattare:</b> Ingenjörfirma Nordisk ADB AB		
<b>1. Fackteknisk beskrivning</b>		
1.1 Beräkningsuppgifter	1.4 Indata	1.7 Tillämpningsexempel
1.2 Beräkningsmetod	1.5 Utdata	1.8 Användardokumentation
1.3 Förutsättningar	1.6 Kopplingar	
1.1 Beräkning av moment- och tvärfkraftsverkningslinjer till följd av vind- och lastvirkas eller lastfaktors virkliga laster. 1.2 Informationsmetod. 1.3 Programmet tillåter godtycklig hänsynsvariation längs elementen samt variabla insparningsförhållanden vid utlägg. Lasterna kan vara utbredda med varierande intensitet eller koncentrerade som punktlastar eller moment. 1.4 Indata består av följande data: balkens systemgeometri, tvärsnittsdata längs elementen samt laster. 1.5 Dataavläs och resultat trycks i format sådant som A4. Resultatet redovisas i bestämda punkter i valbar redovisningsform. Momentavläsning kan även redovisas grafiskt som radskrivningsutskrift. 1.6 Resultatet utgörs av en utdata för efterbearbetning med program NADB-2210 Remberäkning (dimensionering av stödramar) i separat utgåva Statens Beredningskommittés publikation KI-1954. 1.7 Exempel: Belysning med förstrålningsgips; ytterväggs påverkan av jordtryck.		
		
		
1.8 Användarbeskrivning		

BYGGMÄSTAREN programresumé		Program: NADB-2200 Ramberäkning II	
<b>2. Datateknisk beskrivning</b>			
2.1 Första versionsdatum: 1972-11-14	2.2 Aktuell vers.-nummer: (ver. 0)	2.3 Aktuell vers.-datum: 1977-11-02	
2.4 Programmeringspråk: Fortran IV		2.5 Programlängd (statement): Ca 3600 statement	
2.6 Körminnesbehov: 30 K ord	2.7 Arbetslätt: Inga		
2.8 Indata-medium: Kort eller remsa			
2.9 Resultat-medium: Godtycklig skrivare (min. 72 tecken/bredd)			
2.10 Bearbetningsmjöd: Batch eller remote batch			
2.11 Installation: anläggning: UNIVAC Honeywell Bull	modell: 1100-serien HS000	exekutivsystem: Exec-8 Mark III	antal installationer: 2 2
2.12 Datateknisk dokumentation: Programdokumentation			
<b>3. Marknadsinformation</b>			
3.1 Programägare (agent):		Ingenjörfirma Nordisk ADB AB Box 40198, 103 44 Stockholm, Tf: 08-24 63 35	
3.2 Servisinställ: Enligt 0.1			
3.3 Köppris: Enligt avtal	3.4 Hyppris: Enligt avtal	3.5 Installationskostnad: Enligt avtal	3.6 Utbildningskostnad: Enligt avtal
		3.7 Underhållskostnad: Enligt avtal	
3.8 Övrig			
<b>Fackområde</b>			
Byggnadsstatik			
<b>Nyckelord</b>			
Snölastar, momentberäkning, balk, ram			

FIG 3.3 Programresumé 1 Byggmästaren

Det visade sig dock att intresset både från programägare och läsare inte motsvarade de förväntningar man haft. Efter ett års provotid och 12 publicerade programbeskrivningar, lades verksamheten ned. Anledningen härtill var att:

- det visade sig vara svårt att få in resumeerna i rätt tid och i föreskriven form. Allt för mycket tid och pengar gick åt till att redigera dem.
- intresset motsvarade inte på långa vägar kostnaderna.
- konsulterna visade av ekonomiska skäl ett svalt intresse för att sälja enbart programdelen.

Programresumeernas utseende framgår av FIG 3.3 . Den är indelad i fyra delar: blanketthuvud, fackteknisk beskrivning, datateknisk beskrivning och marknadsinformation.

I blanketthuvudet redovisas titel, programnamn och programförfattare.

Den facktekniska beskrivningen redogör för programmets beräkningsuppgift, beräkningsmetod, förutsättningar, in- och utdata samt kopplingar med andra program. Dessutom finns här ett beräkningsexempel med figur och en beskrivning av tillgänglig dokumentation.

Under punkten datateknisk beskrivning framgår första versionsdatum, aktuellt versionsdatum, programmeringsspråk, erforderlig kapacitet på datorn och dess kringutrustning samt en del datatekniska arbetsbeskrivningar.

Marknadsinformationen redovisar programägare, service-ställen och priser. Sist i programresumen anges fackområde och nyckelord för sökning av programmet.

### 3.3.6 Svensk Byggtjänst, 1982

Svensk Byggtjänst lät under hösten 1981 göra en inventering av de på marknaden tillgängliga datorprogrammen. Med tillgängliga menar man sådana program som man kan köpa eller på annat sätt komma åt som "konsument". Man har tagit intryck av BM's Dokumentationsbank och BFR's ADB-grupps arbete.

Man gick tillväga så att man lät skicka ut en enkät till sådana företag där man kände till att det fanns ADB-kapacitet. Företagens namn fick man ur ett eget företagsregister. Man sände inte ut några påminnelser till företag som inte svarade, med undantag från ett par som bedömdes vara av speciellt stort intresse.

En del av svaren man fick ansågs vara så oklara, att man lät en datakonsult tolka svaren.

Syftet med Marknadsguiden var att informera konsulter och entreprenörer om programmens existens, och därmed undvika dubbelarbete med programutveckling.

Cirka 1800 företag kontaktades - av dessa fick man svar från cirka 150. Denna låga siffra trodde man på Byggtjänst berodde på att man i företagen inte visste vilken policy man skulle ha. "Skall vi eller skall vi inte gå ut med det här programsystemet?" Man valde i många fall den enklaste lösningen: att inte delta i guiden.

Viss kritik har riktats mot Marknadsguiden och dess utformning. Bland annat har man ansett att den borde vara komplett, dvs även program som inte är allmänt tillgängliga borde ingå. Andra har påpekat att de administrativa programmen är allt för mycket tagna "rakt in", utan att ifrågasätta om de är användbara för byggbranschen. Kritik har också riktats mot presentationen av programmen. Man anser att det bör framgå om uppgift saknas om en funktion eller om den inte erbjuds av företaget. Ett utdrag visas i FIG 3.4.

Andra reaktioner är att de som inte svarat, i efterhand har skickat "långa listor" på program som man vill ha med i nästa guide.

I 1982 års Marknadsguide har cirka 400 program presenterats, från ett hundratal företag. Programmen är indelade i fyra produktområden, nämligen

**Systembeskrivning:** Nej.  
**Användarhandbok:** Ja.  
**Kostnader:** Anslutning 20 000 kronor.

#### Godtycklig plan ram I

**Beskrivning:** Programmet behandlar plana stångbärverk med godtycklig geometri och belastning enligt systematiserad förskjutningsmetod. Standardversionen av programmet tillåter stångbärverk bestående av olika material (E-moduler). Stångbärverkets stänger kan ha sprängvis föränderlig sektion och vara ledade eller böjstyvt anslutna i knutar.

**Företag:** Skandinavisk Byggdata.  
**Kontaktman:** Villy Rasmussen, 031-12 76 55.  
**Årtal för driftsättning:** 1975-82.

**Antal användare:** 15.  
**Antal körtillfällen per år:** -.  
**Tillhandahållandesätt:** Egen dator.  
**Programmeringsspråk:** Basic-2.  
**Teknisk miljö:** Dator: Wang 2200. Operativsystem: Wang. Databashanteringsystem: -. Internminne: min 32 kB. Skivminne: 1-80 MB. Linjeprotokoll: -.

**Leveransform:** Källkod.  
**Systembeskrivning:** Ja, 4 sidor.  
**Användarhandbok:** Ja, 10 sidor.  
**Kostnader:** Anslutning 15 000 kronor.

#### NADB-2203 Ramberäkning II

**Beskrivning:** Beräkning av moment och avskärningskrafter för kontinuerliga balkar och envändningsramar med oförskjutbara knutpunkter. Balkar och pelare kan ha varierande tröghetsmoment längs elementen. Programmet medger beräkning med utbredda laster med varierande lastintensitet, punktlast och punktmoment. Lasterna kan vara vilande eller rörliga.

**Företag:** Nordisk ADB AB.  
**Kontaktman:** Bo E Ahlgren, 08-24 63 35.

**Årtal för driftsättning:** 1972.  
**Antal användare:** 20.  
**Antal körtillfällen per år:** 100.  
**Tillhandahållandesätt:** Servicebyrå och egen dator.  
**Programmeringsspråk:** Fortran IV.  
**Teknisk miljö:** Uppgifter saknas.  
**Leveransform:** Objektкод.  
**Systembeskrivning:** Nej.  
**Användarhandbok:** Ja, 28 sidor.  
**Kostnader:** Ej angivet.

#### Ram

**Beskrivning:** Ramberäkningar enligt finita elementmetoden.  
**Företag:** A-Data AB.  
**Kontaktman:** Christer Hällman, 08-81 00 60.

FIG 3.4 Utdrag ur Svensk Byggtjänsts Marknadsguide

- 1) Byggnadstekniska datorprogram,
- 2) Administrativa system,
- 3) CAD/CAM system,
- 4) Databaser.

De byggnadstekniska programmen är i sin tur indelade i 27 ämnesområden. Dessa områden är i stort sett samma som i BFR's rapport B3:1972. De administrativa programmen indelas i 4 grupper beroende på vilket driftsätt systemen är tillgängliga på.

Eftersom man tycker sig ha mött en positiv reaktion och man anser att ett behov av ett sådant informationsmedium såsom Marknadsguiden finns, har man för avsikt att årligen uppdatera guiden. Nästa utgåva beräknas komma ut i maj 1983.

### 3.3.7 ByggInfo, 1982

Bygginfo har sammanställt en informationspärm om datoranvändning i byggbranschen. Pärmens innehåller information om företag i databranschen och deras datorprogram. Den är avsedd som ett hjälpmedel för dels informationsingenjörer från Bygginfo, dels de företag som är anslutna till informations servicen.

Data-Guiden är inte någon egentlig programinventering, snarare en allmän informationspärm om ett antal datorföretag. Vi har ändå valt att presentera den här bland inventeringarna, eftersom den innehåller en viss, av Bygginfo utvald, information om datorprogram.

Man har presenterat företagen indelade i tre grupper: CAD-system, Serviceföretag och Informationsföretag.

Bygginfo har valt ut företag som man anser vara av intresse för kunderna. Det är alltså inte fråga om någon fullständig förteckning utan ett urval av datorföretag.

Pärmens är uppbyggd med ett lösbladssystem, för att man lätt skall kunna hålla den aktuell. Företagens programutbud presenteras oftast med reklambroschyrer, vissa företag har dock förutom detta bilagt en kort allmän presentation av det egna företaget.

## 3.4 Internationell utblick .

Vi skall i detta avsnitt kort beröra vad som hänt och händer utanför Sveriges gränser. Det skall dock understyrkas att vårt urval av projekt inte gör anspråk på att vara representativt eller fullständigt. Syftet är att se de svenska initiativen i fråga om programinformation och programdokumentation i ett internationellt perspektiv.



### 3.4.1 Finland

TEKLA är ett programvaruhus som också fungerar som ADB-konsult. Det bildades av en grupp ledande konsultfirmor i Finland 1966. Eftersom de ledande firmorna i branschen deltog i projektet kom TEKLA att få stor betydelse för utvecklingen av byggnadstekniska datorprogram. TEKLA var ett, internationellt sett, ganska tidigt initiativ och beskrevs vid IABSE:s kongress om datorer i Bergamo 1978.

Syftet med initiativet var att ta fram byggnadsteknisk programvara. Ett antal programmerare samlades under TEKLA's vingar för att dra upp riktlinjerna för utvecklingen av denna programvara. Det hela resulterade i att man några år senare kunde presentera ett programutbud av hög teknisk kvalitet. Man hade med ett fungerande lagarbete utnyttjat erfarenheter för att lösa de gemensamma problemen.

Programmen blev tillgängliga via uppkopplade terminaler. Detta fick som följd att de små persondatorernas intåg i Finland försenades i förhållande till övriga världen.

I och med att personer med kunskaper både från programmering och byggnadskonstruktion fanns i organisationen, kunde man undvika de kommunikationsproblem som vanligtvis uppstår dessa grupper emellan.

### 3.4.2 Norge

I Norge startades 1978 ett projekt, Profil, för att skaffa fram tillgängliga datorprogram för byggnadsbranschen. Initiativtagare var SINTEF, forskningsinstitutet vid Norges Tekniska Högskola i Trondheim. Tanken var att verksamheten skulle grundas på aktivt medlemskap och att verksamheten helt skulle tas över av medlemmarna efter viss tid.

Medlemmarna får kostnadsfritt publicera beskrivningar av egna program eller sådana som på annat sätt är tillgängliga hos medlemmen. Ett villkor för publicering är att programmen är, mot skälig kostnad, tillgängliga för samtliga medlemmar av Profil. I skrivande stund upptar katalogen 142 program, uppdelade i 12 grupper. Programmen presenteras på standardblanketter med upplysning om vad programmet gör, vilken metod som används, programstorlek, pris och av programmeraren subjektivt bedömd kvalitet.

Parallellt med detta ger man ut en informationsbulletin med bidrag från medlemmar och redaktion. Avsikten med denna är att verka som ett generellt forum för byggnadsbranschens databehandling. Här finns också möjlighet för medlemmarna att kostnadsfritt presentera egna produkter och program. Den utkommer oregelbundet,

cirka 2-3 gånger per år.

Verksamheten finansieras helt med hjälp av medlemsavgifter.

### 3.4.3 England

Construction Industry Computing Association, CICA, är en medlemsägd serviceorganisation. Verksamheten finansieras med hjälp av medlemsavgifter. Varje fullvärdig medlem har rösträtt vid val av organisationens styrelse. Sammansättningen bestäms dels av medlemmarna dels av myndigheter som utser egna representanter.

Organisationens målsättning är att:

- uppmuntra ett förnuftigt användande av datorer i byggnadsbranschen.
- förmedla erfarenheter och kontakter medlemmarna emellan.
- ge kostnadsfri rådgivning till sina medlemmar.

Förutom denna service, ger man ut en lista över offentligt tillgängliga program och ibland även en värdering av dessa ur användarens synpunkt. Genom kopplingen till myndigheterna försöker man stimulera programutvecklingen att svara mot medlemmarnas behov.

"Computer programs for the building industry" är en programkatalog som ges ut av arkitektfirman Hutton+Rostron. Verksamheten drivs affärsmässigt genom försäljning av katalogen till ett pris av 25 pund.

Drygt 1000 program finns upptagna (prognos för tredje upplagan, mer än 1200). Informationen om programmen består av en kort beskrivning, namn och adress till programägare, tillgänglighet, programmeringsspråk, dator och eventuella referenser. För ytterligare information hänvisas till ägaren/försäljaren av programmet.

Nytt från och med den tredje upplagan är en sammanställning av tillbehör och utrustning. Denna säljs separat.

### 3.4.4 Väst-Tyskland

Idén bakom Byggmästarens programresumee, se avsnitt 3.3.5, kom ursprungligen från den tyska tidskriften Bauwirtschaft, på initiativ från IABSE. Ett formulär hade tagits fram för att ge en koncentrerad programbeskrivning på två A4-sidor. Med hjälp av detta skulle en kund kunna välja ut sådana program som förtjänade ytterligare intresse och få upplysning om var mera information kunde inhämtas.

Under projektets gång, som började 1975, hann man med att publicera drygt 300 s k datablad. Det kostade inte programförsäljaren något att publicera programbeskrivningar i tidningen, man räknade med att företagen skulle öka sin annonsering i stället. Så blev dock inte fallet.

Anledningen till att man lade ned verksamheten var att intresset, som hade förväntats vara stort, inte motsvarade kostnaderna. Anledningar till detta svala intresse trodde man bland annat var:

- "mänskliga konkurrensfaktorn" (att köpa skulle minska den egna expertstatusen på företaget),
- att man av gammal vana behåller etablerade kontakter,
- de som söker något nytt vänder sig direkt till kända programleverantörer.

Rechen- und Entwicklungsinstitut für EDV im Bauwesen, RIB, är ett programvaruhus i Stuttgart. Det bildades 1961, med uppgift att förse byggnadsbranschen med programvara. Institutet är helt fristående utan privata vinstintressen. Bakom står intressenter för näringsliv och det allmänna som är inblandade i byggprocessen.

Verksamheten finansierades till en början av forskningsministeriet i Bonn. Institutet drivs affärsmässigt med försäljning och uthyrning av program, och är numera självbärande.

Man erbjuder ett standard sortiment som i stort sett täcker in hela byggprocessen. Cirka ett 90-tal program erbjuds i programkatalogen. Men det finns även möjlighet att beställa specialutveckling av program.

### 3.4.5 USA

ACI, American Concrete Institute, efterlyste i en artikel i ACI-Journal, datorprogram för analys och dimensionering av betongkonstruktioner. Året var 1975. Syftet med detta var att bereda medlemmarna i ACI tillgång till existerande program på området. Det man var ute efter var små, portabla applikationsprogram för allt ifrån vanliga balkar och plattor till konstruktioner av kompositbetong.

Man uttryckte minimikrav i en programspecifikation. Syftet med detta var:

- att etablera en standard mot vilken existerande program kunde jämföras,
- att hjälpa programförfattare att definiera viktiga programstrukturer,

-att bidra till en förbättring av befintliga program för att därmed höja värdet av datorbearbetningarna.

Man uppmanade ägare av program att sända in en kort beskrivning av sina program. Kontroll utfördes huruvida programmen uppfyllde de i specifikationen uppställda kraven. Man kontrollerade bland annat att en utförlig användarinstruktion fanns, att hjälp kunde erbjudas användaren t ex per telefon, att använda ACI-normer angivits och att antingen en utförlig systembeskrivning eller motsvarande service erbjöds.

Om programmen verkade uppfylla de uppställda kraven, infortrade man ytterligare information om programmet (dokumentationer, tillgänglighetsform, etc). Klarade programmen också denna granskning, publicerades resultatet i ACI-Journal. De som fanns vara ofullständiga på någon punkt fick möjlighet att antingen komplettera eller att dra tillbaka sitt program.

Eftersom ACI inte genomförde någon teknisk programtest, frantog man sig allt ansvar för programmets riktighet.

En beskrivning av verksamheten ges i ACI-Journal, maj 1976 och juli 1980.

## 4 GRUNDERNA FÖR STATISKA BERÄKNINGSMETODER

I detta kapitel skall beskrivas två grundläggande metoder för statistisk analys, nämligen kraftmetoden och förskjutningsmetoden. Dessa två tillvägagångssätt kan ses som bakgrunden för ett stort antal välkända beräkningsmetoder för ramar och andra balkstrukturer. Förskjutningsmetodikerna bildar också grundval för Finita Element metoden som kommer att diskuteras i kapitel 5. Beskrivningen inriktas helt på statistiskt obestämda system, då det främst är för dessa som datorbaserade beräkningsmetoder kan bli aktuella.

I kapitlet kommer främst stångsystem (fackverk) och balksystem (kontinuerliga balkar, ramar) att behandlas. När den allmänna metodiken som demonstreras är klarlagd bör generaliseringen till andra typer av strukturer inte vara alltför svår.

De metoder och samband som demonstreras är formulerade med tanke på stora problem. Detta kan medföra att metoderna i vissa fall förefaller onödigt komplicerade för de små problem som, för överskådlighetens skull, visas i exemplen.

Innan genomgången av teorierna påbörjas skall ett par termer definieras, för undvikande av missförstånd:

Med struktur avses i den fortsatta behandlingen ett helt statistiskt system som skall analyseras. Detta kan till exempel utgöras av en takstol, en ram eller till och med en hel byggnadsstomme.

Med element avses de enstaka delar som tillsammans bygger upp strukturen.

Observera vidare att denna definition är tvetydig såtillvida att elementet stång kan bygga upp strukturen takstol, men att elementet takstol kan användas i strukturen husstomme. Termerna är alltså kopplade till vilken modelleringsnivå man befinner sig på. Som tidigare visats så är ju just nedbrytning av en hel struktur till mindre och hanterbara delar ett viktigt steg i förberedelserna för datorberäkningar.

### 4.1 Beskrivning av rörelser

De metoder som skall behandlas kan tillämpas på statistiskt obestämda strukturer. Definitionen av denna term är att de uppträdande krafterna i den analyserade strukturen inte kan lösas enbart med hjälp av jämviktsbetraktelser, utan att rörelserna i strukturen måste utnyttjas. Alltså måste man ha ett sätt att beskriva de rörelser som strukturen genomgår. För detta skall begreppet frihetsgrad utnyttjas.



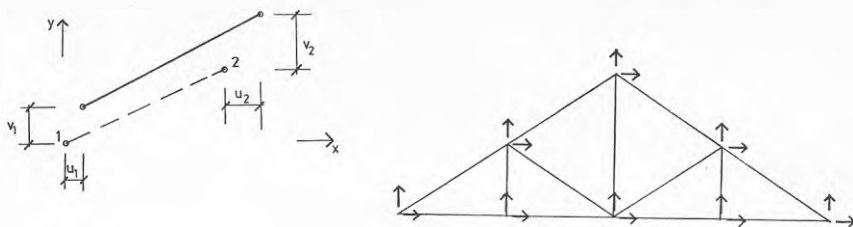


FIG 4.1 Frihetsgrader hos stång resp fackverk.

Som ett exempel på detta kan en enstaka stång (till exempel i det plana fackverket i FIG 4.1) tas. Vid en belastning av fackverket kommer stångens ändpunkter att röra sig på något sätt i  $x$ - $y$  planet. Hur kan dessa rörelser beskrivas?

Eftersom stången, till skillnad från en balk definitionsmässigt, inte antas kunna ta upp böjning kan stångens läge efter påläggande av belastningen beskrivas med hjälp av fyra parametrar nämligen  $x$ -förskjutningarna  $u_1$  och  $u_2$  samt  $y$ -förskjutningarna  $v_1$  och  $v_2$ . Stångelementet kan därför sägas ha fyra frihetsgrader, eftersom värdena på dessa fyra parametrar fullständigt och entydigt beskriver stångelementets rörelser (med de gjorda antagandena).

Genom en förlängning av detta resonemang inses att hela det plana fackverkets rörelsetillstånd kan beskrivas med hjälp av två frihetsgrader i var och en av strukturens knutpunkter, se FIG 4.1.

I ovanstående stycke har talats om elementets och strukturens rörelser. En viss språkförbistring råder i detta sammanhang, då man för motsvarande kvantiteter ofta använder ordet deformationer. I detta kapitel kommer dock endast uttrycken rörelse och förskjutning att användas. Begreppet deformation skall i stället förbehållas de förändringar av ett elements (eller en hel strukturs) form som är en följd av förskjutningarna. Exempelvis kan gälla för stångelementet i FIG 4.2 att dess förskjutningar i  $x$ -led är  $u_1$  resp  $u_2$ . Endast om  $u_1 \neq u_2$  är stången deformerad - förlängd om  $u_2$  är större än  $u_1$  och förkortad om motsatsen gäller.

För ett balkelement, exempelvis ingående i en ram, är det inte tillräckligt att definiera ändpunkternas  $x$ - resp  $y$ -förskjutningar för att hela förskjutningstillståndet skall vara känt. För att beskrivningen skall vara entydig måste man även införa exempelvis vinkeländringarna vid ändarna,  $\theta_1$  och  $\theta_2$ , som frihetsgrader,

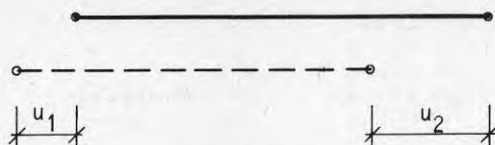


FIG 4.2 Förskjutningar resp deformation i stångelement.

jämför FIG 4.3. Detta medför att för en plan ram uppbyggd av balkelement tre frihetsgrader krävs i varje knutpunkt, FIG 4.3b. Detta är dessutom endast helt sant för fall där dels böjstyvheten är konstant över hela elementlängden, dels endast punktlaster angriper strukturen.

Som framgått av beskrivningen ovan är antalet frihetsgrader i en struktur beroende av dels vilken typ av struktur man behandlar, dels vilka antaganden man gör om de ingående elementens verkningsätt.

Ovan har begreppet frihetsgrad kopplats till de förskjutningar en struktur (eller en del därav) kan genomgå. Genom studie av FIG 4.1 inses också att de definierade frihetsgraderna i fackverket motsvarar de kraftkomponenter man normalt talar om för denna typ av bärverk. Det plana fackverkets knutpunkter kan ju endast belastas med punktkrafter, uppdelade i en x-komponent och en y-komponent.

Som sammanfattning kan frihetsgrader definieras som, endera de förskjutningar som entydigt definierar strukturens läge, eller de kraftkomponenter som definierar de på strukturen verkande krafterna, i båda fallen med hänsyn till antaganden om strukturens verkningsätt.

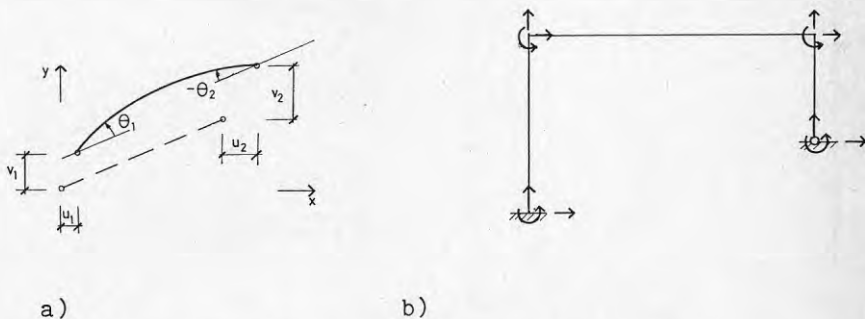


FIG 4.3 Frihetsgrader hos balkelement resp ram.

## 4.2 Upplagsförhållanden och randvillkor

När fackverket i FIG 4.1 diskuterades i föregående avsnitt så definierades två frihetsgrader i var och en av fackverkets knutpunkter. Ingen skillnad gjordes därvid för de upplag som ju strukturen måste ha, utan dessa behandlades på samma sätt. Men frihetsgraderna var ju även ett uttryck för vilka krafter som kunde påläggas strukturen. Från vanliga jämviktsbetraktelser inses att samtliga krafter på strukturen inte kan väljas godtyckligt, utan att åtminstone tre (för det plana fallet) av dem måste räknas fram ur de övriga för att jämvikt säkert skall kunna råda.

Ofta garanteras jämvikten genom att tre upplagsreaktioner specificeras, exempelvis för den plana ramen i FIG 4.1 genom att en knutpunkt fixeras mot rörelser i såväl  $x$ - som  $y$ -led och en knutpunkt mot rörelser i  $y$ -led, se FIG 4.4. I dessa frihetsgrader skall förskjutningen alltså vara noll, vilket egentligen strider mot den andra halvan av definitionen av frihetsgrader som de förskjutningar som behövs för en entydig beskrivning av strukturens läge. I praktisk hantering brukar man ändå ta med dessa när man definierar strukturens frihetsgrader. I ett senare skede under beräkningen införas i stället som ett extra villkor att förskjutningen i några frihetsgrader skall vara noll. Viktigt att notera är emellertid att man i de frihetsgrader som fixeras mot förskjutning får en, i förväg, okänd reaktionskraft - man kan inte bestämma såväl förskjutningen som kraftens värde i några frihetsgrad.

Att förskjutningen i en viss frihetsgrad skall vara noll (eller ha ett annat fixt värde) i en struktur är ett randvillkor för problemet. Dessa villkor är nödvändiga för att fixera strukturens läge i rymden (eller, i detta fall, i planet). De är endast ett alternativt sätt att uttrycka de gängse jämviktsuttrycken (projektionsekvationer och momentjämviktsekvationer) i en form som i stället talar om förskjutningar.

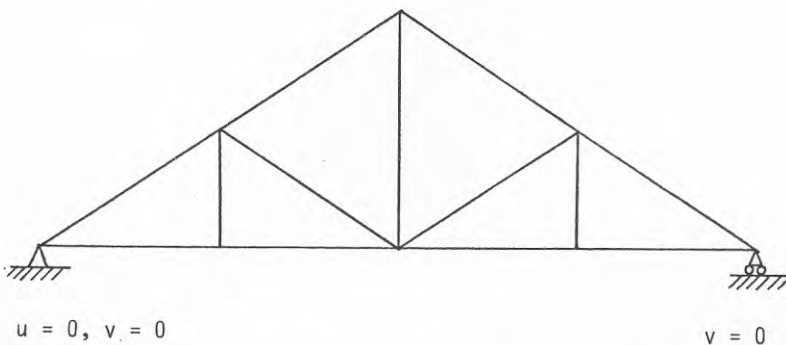


FIG 4.4 Upplagsförhållanden för plant fackverk.

För en struktur finns alltså ett minsta antal förskjutningar som måste fixeras (tre för en plan struktur, sex för en rymdstruktur). Om flera fixeras fås ett statiskt obestämt system. Detta leder inte till några extra problem då förskjutningmetoden, se avsnitt 4.6, används. Om däremot kraftmetoden används leder dessa övertaliga randvillkor till formulering av ett ekvationssystem med upplagsreaktionerna som obekanta.

Som kommer att framgå senare resulterar förskjutningsmetoden i ett ekvationssystem med förskjutningarna i frihetsgraderna som obekanta. En följd av detta är att man, speciellt då man skall lösa problemet för hand, har ett intresse av att reducera antalet obekanta så långt som möjligt.

Det finns, förutom borteliminering av upplagsreaktionerna, andra sätt att reducera antalet "egentliga" frihetsgrader (de man beaktar vid lösning) för att minska antalet obekanta i det ekvationssystem man formulerar. Dessa är alldeles nödvändiga vid handräkning, men behöver ej utnyttjas vid datoranalys.

I FIG 4.5 ges exempel på en plan tvåvåningsram. Var och en av strukturens sex knutpunkter skulle egentligen föras med tre frihetsgrader, det vill säga totalt 18. I knutpunkt 1 är dock förskjutningarna i samtliga tre frihetsgrader fixerade till värdet noll, till följd av upplaget. I knutpunkt 4 är, av samma skäl, förskjutningar i x- resp y-led förhindrade, däremot kan balkelementet invid knutpunkten rotera. Efter borttagandet av de fem upplagsreaktionerna kvarstår 13 frihetsgrader.

Även ett ekvationssystem om 13 obekanta är dock i största laget att lösa för hand, varför ytterligare förenklingar behöver vidtas. Nästa förenkling som kan göras är att försumma axialdeformationerna i balkelementen. Detta kan man göra eftersom det normalt åtgår en betydligt större kraft att förlänga balken med ett visst mått än att böja ut balkens ände ett lika stort

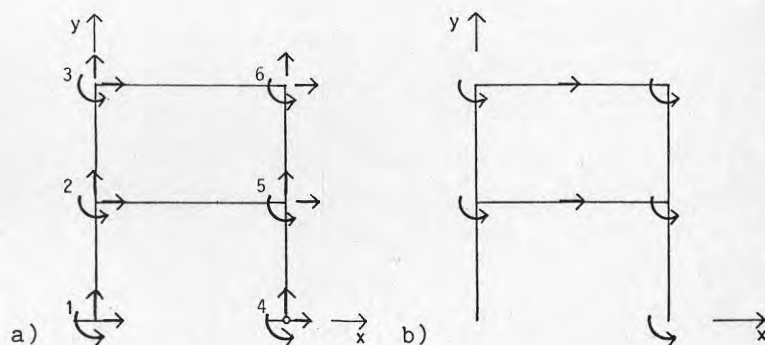


FIG 4.5 Frihetsgrader i plan tvåvåningsram.

mått. Därför kan man ofta bortse från den förkortning/förlängning som sker av balkelementen i en ram. Detta medför, i FIG 4.5, att dels  $v_2=v_3=v_1=0$ , där fixeringen i upplaget utnyttjas. På samma sätt fås att  $v_5=v_6=v_4=0$ . Eftersom inte heller de horisontella balkdelarna antas längdändras kommer vidare att gälla att  $u_2=u_5$  och att  $u_3=u_6$ .

Om dessa förenklingar införs inses att hela ramens förskjutningstillstånd nu kan beskrivas med hjälp av sju parametrar, nämligen fem stycken knutpunktsrotationer och förskjutningen i x-led av de två bjälklagsnivåerna, jämför FIG 4.5b.

Storleken på ekvationssystemet kan faktiskt reduceras ytterligare något. Man kan nämligen, genom att hela tiden hålla i minnet att momentet i knutpunkt 4 är noll, ta bort även rotationsfrihetsgraden i knutpunkt 4, och sluta med ett ekvationssystem om sex obekanta, jämför också avsnitt 4.6 där ett liknande fall löses.

### 4.3 Utnyttjande av symmetri och repetition

Ofta kan en modell av en struktur man önskar beräkna förenklas genom utnyttjande av, till exempel, symmetriegenskaper i problemet. Förutsättningarna för detta är dock att såväl strukturen som de pålagda lasterna och upplagsförhållandena är symmetriska.

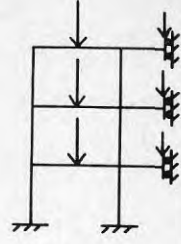
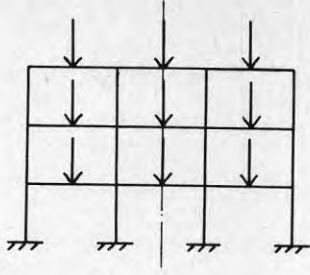
I de fall detta kan utnyttjas innebär det ofta en betydande minskning av det erforderliga räknearbetet, genom att den betraktade strukturen endast är en halva (eller en fjärdedel, eller någon annan andel) av den ursprungliga. För att i analysen ta in effekterna av de delar som inte modelleras skall den betraktade delstrukturen förses med vissa randvillkor i de gjorda snitten. I FIG 4.6 visas några exempel på hur symmetri- och repetitionsegenskaper hos problem kan utnyttjas. Observera dock att de inte utan vidare kan utnyttjas för alla lastfall på de visade strukturerna.

#### 4.3.1 Symmetri

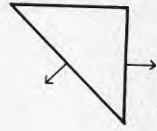
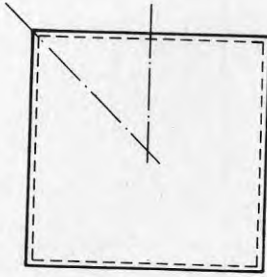
Randvillkoren i ett symmetrisnitt bestäms av att även den deformerade geometrin skall vara symmetrisk. Detta innebär för ramen i FIG 4.6a att momentstyva rullager måste införas i symmetrisnittet, med avsikt att rambalkarna skall få en horisontell tangent i mittpunkten. Ur symmetrin inses vidare att i rambalkarnas mittpunkter förskjutningen i x-led måste vara noll. Däremot kan dessa punkter förskjutas fritt i y-led.



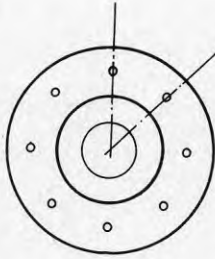
a)



b)



c)



d)

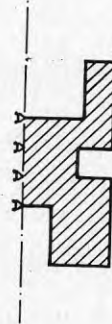
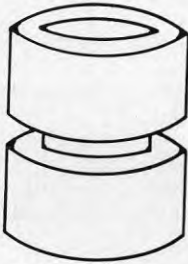


FIG 4.6 Exempel på strukturer som kan förenklas.

Även de givna lasterna kan behöva ändras då symmetri-egenskaper utnyttjas. Ur FIG 4.6a framgår att, då endast halva strukturen betraktas, endast halva de vertikala lasterna i symmetrisnittet skall tas med. Om vi dessutom haft vertikala stänger i mittsnittet skulle deras areor ha halverats.

I FIG 4.6b visas en kvadratisk platta utsatt för en jämnt utbredd last. I detta fall består plattan av åtta stycken likadana delar (sånär som på riktningarna). Detta kan utnyttjas för att minska den betraktade strukturen vid en analys med Finita Element metoden (se avsnitt 6.3). Endast en åttondel behöver beaktas, om som randvillkor införes att plattans lutning i en riktning vinkelrätt mot den sneda begränsningslinjen skall vara noll, och att samma förhållande gäller i det andra utnyttjade snittet.

#### 4.3.2 Repetition

I samband med runda strukturer kan också analysen ofta förenklas. Detta gäller bland annat för strukturen i FIG 4.6c, där man har en viss repetition i strukturens utseende.

Detta kan utnyttjas i analysen, om man skär ut en sektor som motsvarar den repeterande andelen av strukturen. I sektorgränserna måste sedan vissa randvillkor införas, exempelvis att punkter på sektorgränsen endast kan röra sig längs den radie som begränsar sektorn, ej vinkelrätt däremot, se FIG 4.6c.

Fallet i FIG 4.6d med en axisymmetrisk struktur medger också vissa förenklingar. Den egentligen tre-dimensionella strukturen kan nämligen, som vidare beskrivs i avsnitt 6.2, behandlas som en tvådimensionell skiva om riktiga förskjutningssamband ansättes.

#### 4.3.3 Antimetriska laster

I diskussionen om symmetriska förhållanden ovan förut-sattes att såväl den aktuella strukturen som dess laster och randvillkor var symmetriska.

Om dock en symmetrisk struktur utsätts för en antimetrisk belastning kan även detta utnyttjas i arbetsbesparande syfte. Detta medför att randvillkor, skilda från dem som införts vid den symmetriska belastningen, måste anges. Som exempel visas FIG 4.7 där en ram påverkas av antisymmetriska punktlaster. Analysen kan även i detta fall begränsas till att omfatta ena halvdelen. Som randvillkor måste då införas ett villkor att vertikalförskjutningen i symmetrisnittet skall vara noll, och - eftersom en inflektionspunkt fås - att momentet skall vara noll i samma punkt.

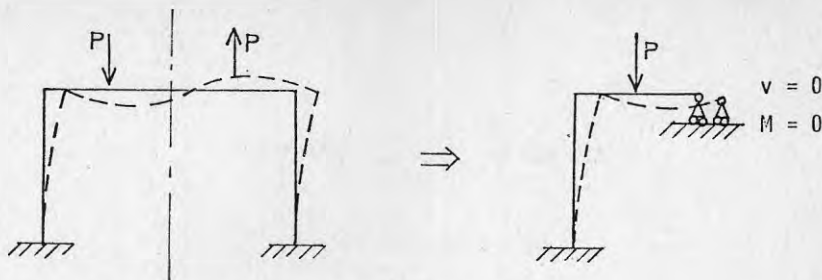


FIG 4.7 Ram med antimetrisk belastning.

Även för en godtycklig belastning på en symmetrisk struktur är detta användbart. Det gäller ju nämligen att varje belastning kan delas i två delar, den ena symmetrisk och den andra antimetrisk. Sålunda räcker det att endast ena halvdel av strukturen betraktas, men i stället fås två olika lastfall, med inbördes olika randvillkor. Resultaten från de två lastfallen kan sedan adderas enligt superpositionsprincipen.

#### 4.4 Kraftmetod och förskjutningsmetod, principer

Som en introduktion till de mer noggranna beskrivningarna av de två beräkningsmetoderna i avsnitt 4.5 resp 4.6 skall nu ett relativt enkelt exempel genomräknas med hjälp av dels kraftmetoden, dels förskjutningsmetoden. Den struktur som skall behandlas är stångsystemet i FIG 4.8.

##### 4.4.1 Kraftmetoden

När kraftmetoden skall användas måste man först bestämma hur mångfalt statiskt obestämd strukturen är. I detta fall finns två övertaliga stänger, och systemet är alltså tvåfalt statiskt obestämt.

Nästa steg är att snitta upp strukturen på ett sådant sätt att det som kvarstår är ett statiskt bestämt system. I detta fall kan detta, till exempel, ske genom att två stänger skärs av invid sina upplag, såsom visas i FIG 4.9a. Denna struktur är lätt att analysera vad gäller stångkrafter och förskjutningar.

För att den förenklade strukturen skall överensstämma med den verkliga måste man nu i de punkter där man skurit av stängerna införa krafter (motsvarande stångkrafterna i dessa stänger). Som positiv riktning för dessa väljs krafter som motsvarar drag i stängerna. Målet för analysen är nu att välja dessa krafter till en sådan storlek att det glapp som uppstått i de snittade stängerna precis går ihop.

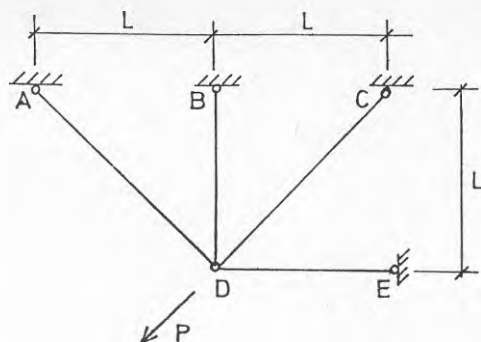


FIG 4.8 Exempel på stängsystem

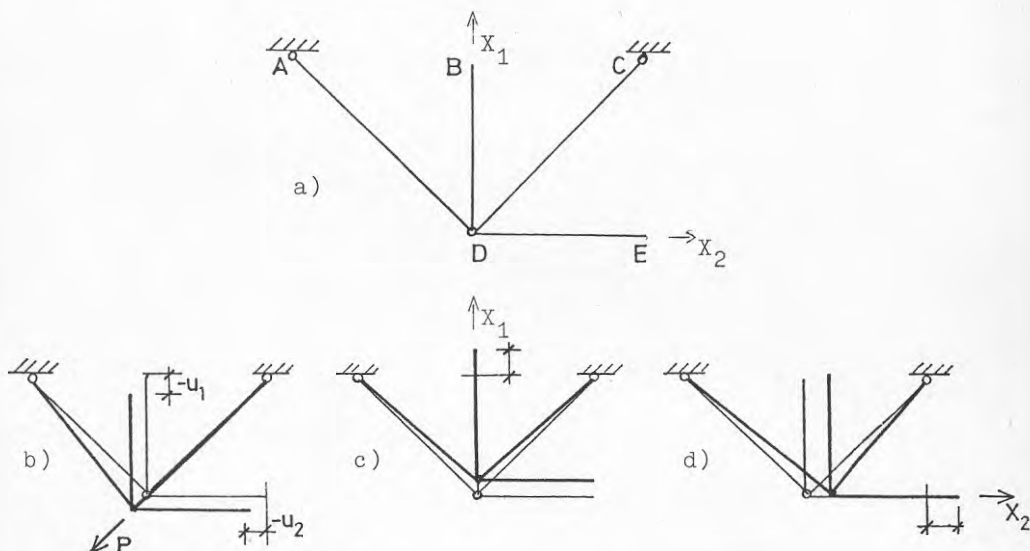


FIG 4.9 Analys med kraftmetoden av stängsystemet i FIG 4.8.

Förutom att analysera det statiskt bestämda hjälpbärverket för de verkande yttre lasterna, FIG 4.9b, skall vi även analysera det med hänsyn till dessa statiskt obestämda krafter. Vi beräknar därför ytterligare två lastfall. I vardera sätts den ena av våra okända krafter till ett värde 1, medan den andra är noll. I FIG 4.9c-d visas dessa fall. Som framgår av figurerna ger vart och ett av fallen vissa förskjutningar i de två snitten, parallellt med de obekanta krafterna. Dessa skall nu med hjälp av att man väljer värden på de okända krafterna  $X_1$  och  $X_2$  adderas till noll.

Två villkor uppstår alltså, ett för var och en av våra snittade stänger. Och eftersom det finns två okända (valbara) krafter att använda, kommer man att få ett lösbart ekvationssystem med de två statistiskt obestämda krafterna som obekanta.

Låt oss genomföra beräkningen. Först skall alltså det statistiskt bestämda stångsystemet beräknas med de verkliga lasterna pålagda. Ur FIG 4.9b fås att stångkrafterna blir

$$\begin{aligned} S_{AD} &= 0 \\ S_{CD} &= P \quad (\text{dragen}) \end{aligned}$$

På grund av dessa krafter förlängs stängen CD med en kvantitet

$$\delta_{CD} = \frac{P L \sqrt{2}}{EA}$$

Denna förlängning medför att förskjutningarna i de fiktiva snitten blir - observera att stängerna BD och ED endast flyttas-

$$\begin{aligned} u_1(q) &= - \frac{P L}{E A} \\ u_2(q) &= - \frac{P L}{E A} \end{aligned}$$

där  $q$  betecknar inverkan av "verkliga" laster.

Det andra fallet som skall analyseras är det med kraften  $X_1=1$  som verkar i snitt nummer 1. Observera att kraften väljs positiv åt samma håll som betecknats som positiv för förskjutningen  $u_1(q)$ . Detta lastfall ger förskjutningar enligt FIG 4.9c. Stångkrafterna i det statistiskt bestämda systemet för detta lastfall blir

$$S_{AD} = S_{CD} = - \frac{X_1}{\sqrt{2}}$$

Förkortningen av dessa stänger ger en förskjutning uppåt av knutpunkt D, vilken tillsammans med förlängningen av stängen BD ger en förskjutning uppåt av snitt 1 med värdet

$$u_1(X_1) = \sqrt{2} \frac{X_1 L}{EA} + \frac{X_1 L}{EA} = \frac{X_1 L}{EA} (\sqrt{2} + 1)$$

I snitt 2 fås ingen förskjutning av detta lastfall

$$u_2(X_1) = 0.$$



Motsvarande beräkningar för fallet  $X_2 = 1$ ,  $X_1 = 0$  ger

$$u_1(X_2) = 0$$

$$u_2(X_2) = \sqrt{2} \frac{X_2 L}{EA} + \frac{X_2 L}{EA} = \frac{X_2 L}{EA} (\sqrt{2} + 1)$$

Vi är nu mogna att söka kombinera de tre fallen i FIG 4.9b-d på ett sådant sätt att strukturen i FIG 4.8 åstadkommes. Vi får de totala förskjutningarna i de två snitten genom superponering av de tre delfallen

$$u_1 = u_1(q) + u_1(X_1) + u_1(X_2)$$

$$u_2 = u_2(q) + u_2(X_1) + u_2(X_2)$$

Eftersom förskjutningarna i de fiktiva snitten måste vara lika med noll ger ovanstående ekvationer två villkor för bestämning av de obekanta krafterna. I detta fall blir beräkningen enkel. Man får

$$X_1 = X_2 = \frac{P}{\sqrt{2}+1}$$

När sedan dessa två stångkrafter är kända kan övriga stångkrafter beräknas med hjälp av jämviktsekvationer. Mycket av detta arbete finns ju dessutom redan gjort. I de tre delfallen har ju beräknats, för exempelvis stångkraften i stång AD,

$$S_{AD} = S_{AD}(q) + S_{AD}(X_1) + S_{AD}(X_2)$$

som nu kan adderas. Detsamma gäller för förskjutningen av punkt D i vertikalled, då vi har beräknat detta värde för vart och ett av de tre delfallen, och nu kan addera dem då de verkliga värdena på X-krafterna beräknats.

#### 4.4.2 Förskjutningsmetoden

Det första steget i en analys med förskjutningsmetoden är att definiera de behövliga frihetsgraderna för den aktuella strukturen. Detta måste göras med hänsyn till strukturens verkningssätt och de belastningar som skall läggas på, och frihetsgraderna måste dessutom väljas så att strukturens rörelsemöjligheter kan beskrivas på ett entydigt sätt.

Det aktuella stångsystemet har egentligen 10 frihetsgrader, nämligen en förskjutning i x-led och en i y-led för var och en av de 5 knutpunkterna. Med hänsyn till upplagsförhållandena kvarstår dock endast 2 egentliga frihetsgrader, nämligen de i knutpunkt D. Vi betecknar denna punkts rörelse i x-led med  $r_1$  och dess

rörelse i y-led med  $r_2$ .

Till de obekanta förskjutningarna i frihetsgraderna hör enligt avsnitt 4.1 krafter som definierar det krafttillstånd som verkar på strukturen. Vi betecknar krafterna som svarar mot förskjutningarna  $r_1$  och  $r_2$  med  $R_1$  resp  $R_2$ . Figur 4.10a visar de krafter och förskjutningar i strukturen som skall beaktas.

Den grundläggande tanken i förskjutningsmetoden är nu att analysera vilka krafter som behövs för att åstadkomma vissa förskjutningstillstånd i strukturen, och att sedan superponera dessa förskjutningstillstånd på ett sätt som svarar mot de pålagda lasterna. Det som skall lösas är ett ekvationssystem med de två okända förskjutningarna som obekanta.

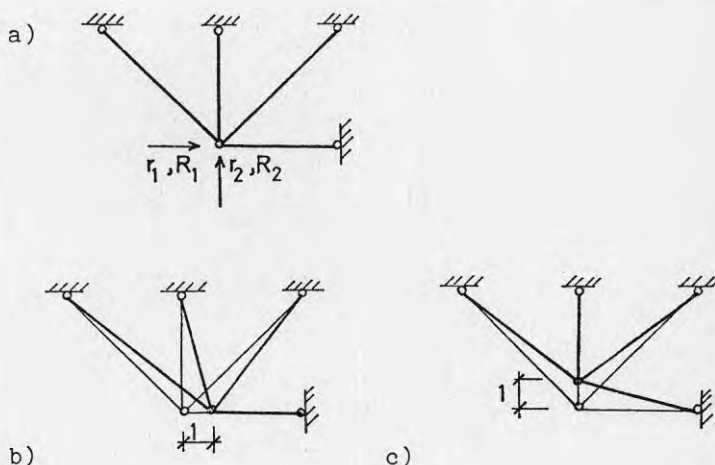


FIG 4.10 Förskjutningsmetoden för stångsystemet i FIG 4.8.

Det första förskjutningstillstånd som skall analyseras kännetecknas av att förskjutningen  $r_1$  väljs till 1 medan  $r_2 = 0$ , se FIG 4.10b. För att deformera strukturen till detta tillstånd åtgår vissa krafter. Beräkna därför töjningarna och krafterna i de olika stängerna till följd av den påtvingade förskjutningen.

$$\delta_{AD} = \frac{r_1}{\sqrt{2}} \qquad S_{AD} = + \frac{EA}{2L} r_1$$

$$\delta_{BD} = 0 \qquad S_{BD} = 0$$

$$\delta_{CD} = -\frac{r_1}{\sqrt{2}} \quad S_{CD} = -\frac{EA}{2L} r_1$$

$$\delta_{DE} = -r_1 \quad S_{DE} = -\frac{EA}{L} r_1$$

Dessa krafter uppstår alltså i stängerna då stängssystemet hålls deformerat. De skall nu överföras till krafterna i de definierade frihetsgraderna, eftersom det endast är dessa som skall verka på systemet. Med hänsyn till riktningarna och lutningarna på stångkrafterna fås kraftsummorna

$$R_1 = \frac{S_{AD}}{\sqrt{2}} - \frac{S_{CD}}{\sqrt{2}} - S_{DE} = \frac{EA}{L} \left( \frac{1}{\sqrt{2}} + 1 \right) r_1$$

$$R_2 = -\frac{S_{AD}}{\sqrt{2}} - \frac{S_{CD}}{\sqrt{2}} - S_{BD} = 0$$

På motsvarande sätt fås för fall nummer 2, där  $r_1 = 0$  och  $r_2 = 1$ , jämför FIG 4.10c.

$$\delta_{AD} = -\frac{r_2}{\sqrt{2}} \quad S_{AD} = -\frac{EA}{2L} r_2$$

$$\delta_{BD} = -r_2 \quad S_{BD} = -\frac{EA}{L} r_2$$

$$\delta_{CD} = -\frac{r_2}{\sqrt{2}} \quad S_{CD} = -\frac{EA}{2L} r_2$$

$$\delta_{DE} = 0 \quad S_{DE} = 0$$

$$R_1 = \frac{S_{AD}}{\sqrt{2}} - \frac{S_{CD}}{\sqrt{2}} - S_{DE} = 0$$

$$R_2 = -\frac{S_{AD}}{\sqrt{2}} - \frac{S_{CD}}{\sqrt{2}} - S_{BD} = \frac{EA}{L} \left( \frac{1}{\sqrt{2}} + 1 \right) r_2$$

Genom superponering av dessa fall kan man, med olika värden på de två förskjutningarna i frihetsgraderna, beräkna vilka krafter som åtgår för att hålla strukturen deformerad. Eftersom vi har två parametrar, förskjutningarna, kan vi genom ett visst val av dessa få fram vilken kombination av R-krafter som helst, genom lösning av ekvationssystemet

$$R_1 = \frac{EA}{L} \left( \frac{1}{\sqrt{2}} + 1 \right) r_1 + 0 r_2$$

$$R_2 = 0 r_1 + \frac{EA}{L} \left( \frac{1}{\sqrt{2}} + 1 \right) r_2$$

I det aktuella lastfallet skall lasterna vara, FIG 4.8 och 4.10a,

$$R_1 = R_2 = - \frac{P}{\sqrt{2}}$$

varför lösningen fås som

$$r_1 = r_2 = - \frac{1}{\sqrt{2}+1} \frac{PL}{EA}$$

Med dessa förskjutningar kända kan sedan stångkrafterna och upplagsreaktionerna lätt beräknas ur de uttryck som härleddes ovan. Liksom var fallet vid användning av kraftmetoden kan de totala snittkrafterna och förskjutningarna fås genom addition av redan betraktade delfall.

#### 4.5 Kraftmetoden

I avsnitt 4.4.1 löstes ett enkelt exempel med hjälp av kraftmetoden. I det fallet blev lösningen enkel, eftersom vi fick en ekvation för var och en av de statistiskt obestämda stångkrafterna. Detta är dock inte det normala, utan vanligtvis fås i stället ett linjärt ekvationssystem för lösning av dessa kvantiteter. Den generella metodiken för denna typ av problem framgår tydligare av en analys av den kontinuerliga balken i FIG 4.11.

##### 4.5.1 Kontinuerlig balk, metod 1

I detta fall väljs de tre mittersta stödreaktionerna som statistiskt obestämda kvantiteter, och det hjälpbärverk som skall behandlas blir en fritt upplagd tvåstödsbalk. Med den angivna belastningen fås nedböjningar vid de borttagna stöden. Dessa kan betecknas  $v_1(q)$ ,  $v_2(q)$  resp  $v_3(q)$ , där index anger punktens nummer och  $(q)$  anger att nedböjningarna avser den yttre pålagda lasten. Dessa värden beräknas relativt enkelt med hjälp av handboksformler.

De på detta sätt erhållna nedböjningarna skall återställas med hjälp av upplagsreaktionerna  $X_1$ ,  $X_2$  och  $X_3$  i de inre stöden. För fallet  $X_1 = 1$  och  $X_2 = X_3 = 0$  fås fallet i FIG 4.11c. Nedböjningarna i stödpunkterna av dessa punktlaster är också enkla att beräkna, de får värdena  $f_{11}$ ,  $f_{21}$  och  $f_{31}$  där första index avser den betraktade punkten och andra index den belastade punkten. Observera också att en teckenregel måste följas benhårt. I det aktuella fallet sätts alla beräknade nedböjningar positiva då de sammanfaller med de an-

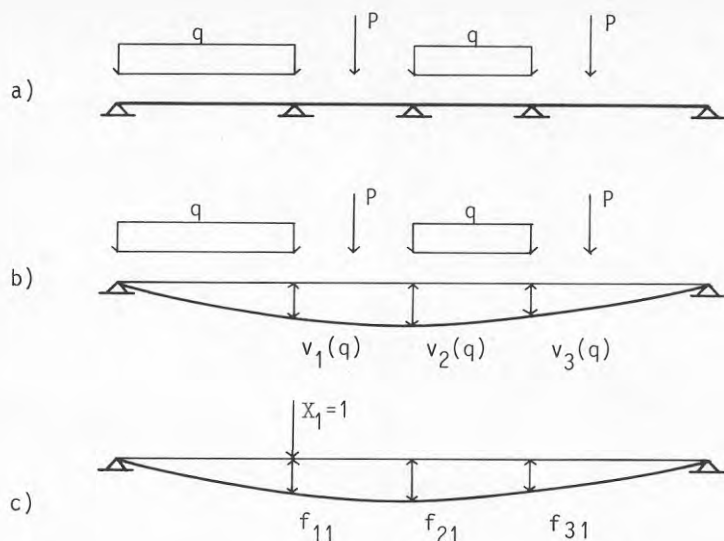


FIG 4.11 Kontinuerlig balk som analyseras med kraftmetoden.

sätta positiva riktningarna för de statistiskt obestämda krafterna, och negativa annars.

Om även det andra fallet med  $X_2 = 1$  och  $X_1 = X_3 = 0$  analyseras fås i stället nedböjningarna  $f_{12}$ ,  $f_{22}$  och  $f_{32}$  i de aktuella punkterna.

Motsvarande beräkningar genomförs även för det tredje fallet, där  $X_3$  ges värdet 1 - resultaten blir tre nedböjningar  $f_{13}$ ,  $f_{23}$  och  $f_{33}$ .

Med godtyckliga värden på de tre  $X$ -krafterna tillsammans med den yttre verkande lasten fås för nedböjningen i stöd 1 (parallell med  $X_1$ ), eftersom superpositionsprincipen antas gälla,

$$v_1 = v_1(q) + f_{11}X_1 + f_{12}X_2 + f_{13}X_3$$

och analogt, för de två andra stöden

$$\begin{aligned} v_2 &= v_2(q) + f_{21}X_1 + f_{22}X_2 + f_{23}X_3 \\ v_3 &= v_3(q) + f_{31}X_1 + f_{32}X_2 + f_{33}X_3 \end{aligned}$$



För att bestämma  $X$ -krafterna används villkoret att nedböjningarna i stöden skall vara noll. Detta ger ett ekvationssystem

$$f_{11} X_1 + f_{12} X_2 + f_{13} X_3 + v_1(q) = 0$$

$$f_{21} X_1 + f_{22} X_2 + f_{23} X_3 + v_2(q) = 0$$

$$f_{31} X_1 + f_{32} X_2 + f_{33} X_3 + v_3(q) = 0$$

som kan lösas. I matrisform kan detta skrivas (jämför Appendix)

$$(F) \bar{x} + \bar{v}(q) = \bar{v}$$

Detta är den generella formuleringen av kraftmetoden. I denna är (F) strukturens flexibilitetsmatris som är en (n/n) matris där n är antalet statistiskt obestämda kvantiteter i problemet. Vektorn  $\bar{v}$  innehåller de förskjutningar som önskas i de snitt som görs för att åstadkomma den statistiskt bestämda strukturen (generellt sett behöver dessa inte vara noll, även om detta är det vanligaste). Eftersom denna vektor är känd och vektorn  $\bar{v}(q)$  kan beräknas ur de yttre lasterna så ger det ovanstående matrissambandet ett linjärt ekvationssystem för lösning av de statistiskt obestämda kvantiteterna i vektorn  $\bar{x}$ .

Som framgår av exemplet i det föregående avsnittet är talen i flexibilitetsmatrisen beroende av strukturens geometri. För en kontinuerlig balk exempelvis bildas termerna ur fackens längder och deras böjstyvheter. Man noterar också att termerna i flexibiliteten skall vara av dimensionen (generaliserad) rörelse dividerad med (generaliserad) kraft. Generaliseringen innebär att även rotationer och moment kan vara aktuella.

#### 4.5.2 Kontinuerlig balk, metod 2

I det ovan skisserade exemplet togs upplagskrafter som statistiskt obestämda kvantiteter, och de mot dessa svarande förskjutningarna utnyttjades för bestämningen. Som alternativ kan i stället moment antas som statistiskt obestämda. I så fall kommer villkor att behöva formuleras för de mot momenten svarande rotationerna eller vinkeländringarna. För den kontinuerliga balken i föregående avsnitt skulle i stället en uppsnittning av balken över stöden kunna göras som i FIG 4.12, vilket ger fyra stycken fritt upplagda enfacksbalkar som statistiskt bestämda hjälpbärverk.

För dessa enfacksbalkar kan uttryck för vinkeländringarna i stöden formuleras med hjälp av handböcker. Exempelvis gäller för facket mellan A och B att

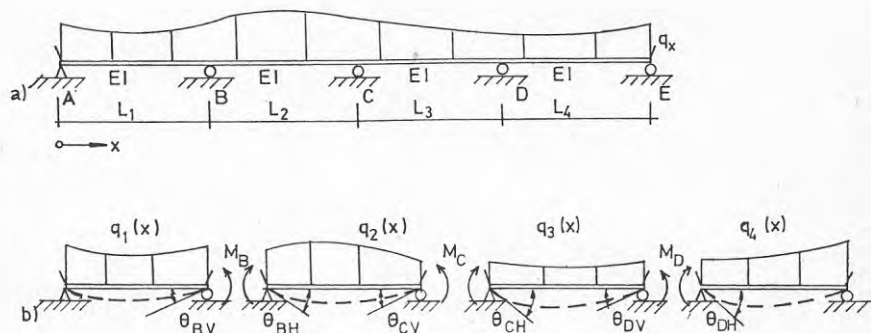


FIG 4.12 Alternativ snittning av kontinuerlig balk.

$$\theta_{Bv} = \frac{M_B L_1}{3EI} + \theta_{Bv}^q$$

För facket BC fås också

$$\theta_{Bh} = \frac{M_B L_2}{3EI} + \frac{M_C L_2}{6EI} + \theta_{Bh}^q$$

Det geometriska villkor som sedan kan utnyttjas är att kontinuitet i lutningen måste gälla, vilket kan formuleras som

$$\theta_{Bv} + \theta_{Bh} = 0$$

Med uttrycken ovan fås ur detta

$$M_B \left( \frac{L_1}{3EI} + \frac{L_2}{3EI} \right) + M_C \frac{L_2}{6EI} + (\theta_{Bv}^q + \theta_{Bh}^q) = 0$$

Över stöden C och D fås motsvarande samband:

$$M_B \frac{L_2}{6EI} + M_C \left( \frac{L_2}{3EI} + \frac{L_3}{3EI} \right) + M_D \frac{L_3}{6EI} + (\theta_{Cv}^q + \theta_{Ch}^q) = 0$$

$$M_C \frac{L_3}{6EI} + M_D \left( \frac{L_3}{3EI} + \frac{L_4}{3EI} \right) + (\theta_{Dv}^q + \theta_{Dh}^q) = 0$$

Ovanstående tre villkor ger de nödvändiga ekvationerna för bestämning av de tre okända stödmomenten. De överensstämmer med den allmänna formen för kraftmetoden.

Vi ser, om ekvationssystemet skrivs i matrisform att flexibilitetsmatrisen består av konstanter av formen

$$\frac{L}{3EI} \quad \text{och} \quad \frac{L}{6EI}$$

för de olika facken. Detta motsvarar konstanterna

$$\alpha_a^0, \quad \alpha_b^0 \quad \text{och} \quad \beta_{ab}^0$$

som ges i exempelvis Handboken BYGG, vilket kan utnyttjas då balken har voter eller dylikt.

Motsvarigheten till det härledda ekvationssystemet kan ställas upp för en kontinuerlig balk i ett godtyckligt antal fack. Metoden går i Handboken BYGG under beteckningen Clapeyron's ekvation eller tre-momentekvationen. Anledningen till det senare namnet är att man i varje ekvation enligt ovan får in tre obekanta stödmoment, nämligen det som råder i den betraktade punkten själv och momenten i de närmast belägna punkterna på båda sidorna.

#### 4.5.3 Ramar

Den ovan visade metodiken kan även användas för ramar, som då skärs upp i hörnen. Hörnmomenten kan så ansättas som statiskt obestämda kvantiteter och villkor ställas upp för kontinuiteten i vinkeländringarna. Denna metodik går oftast under namnet vinkeländringsmetod, och kan användas för alla ramar där ramen ej kan sidoförskjutas.

#### 4.6 Förskjutningsmetoden

Som framgick av det tidigare genomgångna exemplet är det förskjutningarna i vissa, definierade, frihetsgrader som väljs som obekanta i förskjutningsmetoden. Första steget i en analys är därför att definiera den aktuella strukturens frihetsgrader, vilka vi betecknar

$$r_1, r_2, \dots, r_n$$

Dessa måste väljas så att strukturens rörelsemöjligheter beskrivs entydigt, inom ramen för gjorda antaganden. Eftersom det är förskjutningarna som skall lösas ur ett ekvationssystem, finns det all anledning att med hjälp av förslagen i avsnitt 4.2 och 4.3 söka reducera antalet frihetsgrader så långt som möjligt.

Till de obekanta knutpunktsförskjutningarna hör knutpunktskrafter som vi betecknar  $R_1, R_2, \dots, R_n$ , för att markera att de är associerade med förskjutningarna. För knutpunktskraft  $R_k$  gäller således att den har samma läge och riktning som förskjutningen  $r_k$ . Om  $r_i$  är en translation är  $R_i$  en kraft och om  $r_j$  är en rotation är  $R_j$  ett moment.

Knutpunktskrafterna är givna, yttre laster som verkar i knutpunkterna (med specialfall att lasten är noll). Vi kan inte utan vidare betrakta några andra lasttyper, utan laster som förekommer mellan knutpunkterna måste representeras med hjälp av ekvivalenta knutpunktslaster vilkas värde kan bestämmas med hjälp av superpositionsprincipen. Hur man kan gå tillväga i sådana fall visas senare, tills vidare behandlas endast strukturer som belastas i knutpunkterna.

Målsättningen med den förskjutningsbaserade analysen är att formulera ett samband mellan knutpunktskrafter och förskjutningar av formen

$$\begin{aligned} K_{11} r_1 + K_{12} r_2 + \dots + K_{1n} r_n &= R_1 \\ K_{21} r_1 + K_{22} r_2 + \dots + K_{2n} r_n &= R_2 \\ \dots & \dots \\ K_{n1} r_1 + K_{n2} r_2 + \dots + K_{nn} r_n &= R_n \end{aligned}$$

eller, i matrisform

$$(K) \bar{r} = \bar{R}$$

I detta samband kallas (K) strukturens styvhetsmatris,  $\bar{r}$  förskjutningsvektorn och  $\bar{R}$  lastvektorn. Styvhetsmatrisen beskriver strukturens egenskaper och är oberoende (i det linjära fallet) av såväl de verkande lasterna som de uppnådda förskjutningarna. Notera även likheten med uttrycket för en fjäder, där fjäderns styvhet definieras av sambandet:

$$\text{fjäderstyvhet} * \text{förlängning} = \text{kraft.}$$

Fysikaliskt sett kan termerna i styvhetsmatrisen tolkas som krafter vid vissa "enhetsförskjutningar". Om nämligen förskjutningskomponenten  $r_j$  ges värdet 1 och alla andra förskjutningar värdet 0 blir, om matrismultiplikationen ovan utförs, kraftvektorn  $\bar{R}$  lika med kolumn j i styvhetsmatrisen. Alltså kan komponent  $k_{ij}$  i styvhetsmatrisen tolkas som den kraft i frihetsgrad nr i som måste verka för att strukturen skall kunna hållas deformerad i ett läge där förskjutningskomponent nr j givits värdet ett och alla andra förskjutningar värdet noll, jämför exemplet i avsnitt 4.4.2.

Denna definition kan användas för identifiering av talen i styvhetsmatrisen. Frihetsgraderna ges då i tur och ordning förskjutningen 1 och de knutpunktskrafter som måste påföras strukturen för att hålla den i detta föreskrivna deformerade tillstånd beräknas. Denna beräkning kan för en ramstruktur, där böjstyvheten inom varje balkdel är konstant, ske med utnyttjande av endast fyra olika elementarfall, se avsnitt 4.6.2.

Denna metod att bestämma styvhetsmatrisen används endast för enkla system som kan räknas för hand, men är i gengäld mycket användbar i sådana situationer. Den kan även användas för kontroll av resultaten från datorberäkningar.

#### 4.6.1 Exempel på ramberäkning

Som ett exempel på metodiken skall ramen i FIG 4.13 analyseras. Ramen skulle, med 4 knutpunkter, egentligen försees med 12 frihetsgrader. Av dem försvinner emellertid 5 vid upplagen, jämför diskussionen om FIG 4.5 i avsnitt 4.2. Om även i detta fall axialdeformationer i balkelementen försummas kan även förskjutningarna  $v_B$  och  $v_C$  tas bort ur analysen. Av samma skäl kan förskjutningarna  $u_B$  och  $u_C$  ersättas med en gemensam frihetsgrad för horisontalförskjutningen i rambalken. Med hänsyn till leden i D behöver inte heller rotationen  $\theta_D$  medtas. Kvar blir alltså endast tre frihetsgrader enligt FIG 4.13b.

Från den givna systemskissen i FIG 4.13a och frihetsgraderna i 4.13b inses att för det aktuella lastfallet de pålagda yttre lasterna beskrivs med lastvektorn

$$\bar{R} = \begin{matrix} \downarrow \\ \downarrow \\ \downarrow \end{matrix} \begin{matrix} R_1 \\ R_2 \\ R_3 \end{matrix} = \begin{matrix} \downarrow \\ \downarrow \\ \downarrow \end{matrix} \begin{matrix} P \\ 0 \\ 0 \end{matrix} \begin{matrix} \downarrow \\ \downarrow \\ \downarrow \end{matrix}$$

Styvhetsmatrisen, som är av storleken (3/3), kan nu bestämmas genom att frihetsgraderna i tur och ordning ges förskjutningen 1, medan övriga frihetsgrader hålls fast. Dessa tre enhetsförskjutningar framgår av delfigurerna c-e.

Det första fallet, som skall användas för bestämning av första kolumnen i styvhetsmatrisen, fås med  $r_1 = 1$ ,  $r_2 = 0$  och  $r_3 = 0$ .

Ur delfigur c inses att för detta fall åtgår en horisontell kraft  $R_1$ , som alltså blir matriskomponent  $K_{11}$  med en storlek



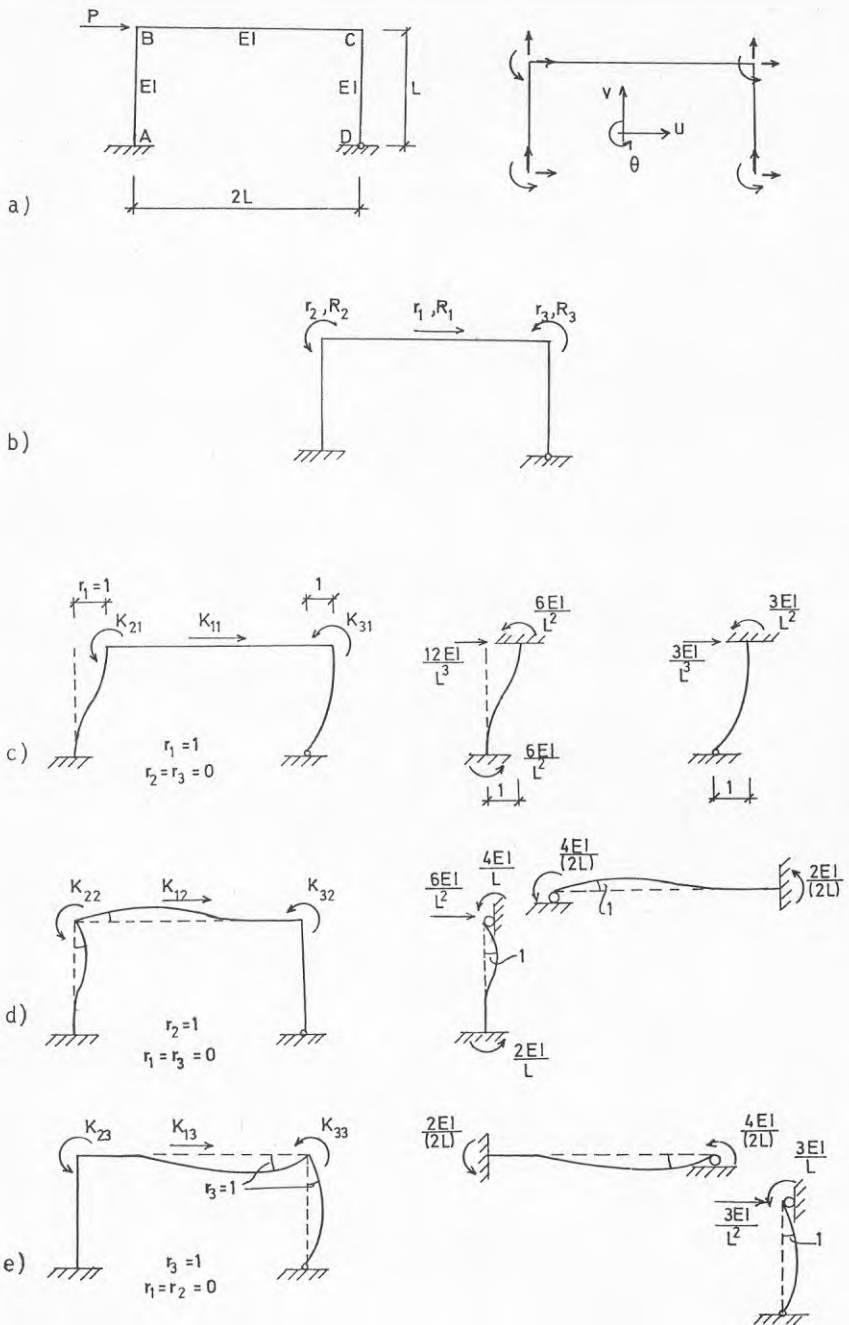


FIG 4.13 Ram analyserad med förskjutningsmetodik

$$K_{11} = \frac{12EI}{L^3} + \frac{3EI}{L^3} = \frac{15EI}{L^3}$$

ett moment i B ( $R_2$ ) med värdet

$$K_{21} = \frac{6EI}{L^2}$$

och ett moment i C ( $R_3$ )

$$K_{31} = \frac{3EI}{L^2}$$

Ur samma delfigur fås även, vilket värde dock inte behövs för den egentliga analysen.

$$M_A(r_1=1) = \frac{6EI}{L^2}$$

Analysen fortsätter sedan med delfallen i figurerna 4.13d och e, vilka ger värden till kolumnerna 2 och 3 i styvhetsmatrisen. Det färdiga resultatet blir

$$(K) = \begin{bmatrix} \frac{12EI}{L^3} + \frac{3EI}{L^3} & \frac{6EI}{L^2} & \frac{3EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} + \frac{4EI}{(2L)} & \frac{2EI}{(2L)} \\ \frac{3EI}{L^2} & \frac{2EI}{(2L)} & \frac{4EI}{(2L)} + \frac{3EI}{L} \end{bmatrix}$$

Dessutom fås, på köpet, vid analys av delfigurerna d och e

$$M_A(r_2=1) = \frac{2EI}{L}$$

$$M_A(r_3=1) = 0$$

Den beräknade styvhetsmatrisen är symmetrisk, det vill säga att varje tal på den högra, övre delen har sin motsvarighet på den diagonalt stående platsen i den nedre, vänstra halvan. Detta kan användas som kontroll på att beräkningarna är riktiga, då symmetri måste råda i styvhetsmatrisen för ett linjärt fall. Med styvhetsmatrisen ( $K$ ) och lastvektorn  $\bar{R}$  kända återstår nu att lösa ett ekvationssystem för att få fram de aktuella förskjutningarna. Man får för det aktuella lastfallet

$$\bar{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} 0.1224 L \\ -0.1139 \\ -0.0506 \end{bmatrix} \frac{PL^2}{EI}$$

Genom superponering av de tre delfallen i beräknade proportioner fås vidare

$$M_A = \frac{6EI}{L^2} 0.1224L \frac{PL^2}{EI} + \frac{2EI}{L} (-0.1139) \frac{PL^2}{EI} + 0 =$$

$$= 0.5066 PL$$

Övriga moment fås också lätt ur de beräknade förskjutningarna, genom utnyttjande av de gjorda delanalyserna.

#### 4.6.2 Elementarfall

Liksom vid användningen av kraftmetoden har strukturen delats upp i enklare delar som relativt lätt kan analyseras. Ur bidragen från de olika delementen byggs sedan ett ekvationssystem upp vilket ger lösningen för hela strukturen.

Vid användningen av kraftmetoden var det som behövdes, utböjningar eller vinkeländringar för fritt upplagda balkar utsatta för vissa laster. I detta fall är det i stället krafter svarande mot vissa förskjutningar som behövs. Som framgår av exemplet ovan är det för denna typ av balkstruktur endast fyra olika elementarfall som kan behöva betraktas. Dessa återges i FIG 4.14, tillsammans med de aktuella krafterna för vart och ett av fallen.

#### 4.7 Jämförelse mellan metoderna

Allmänt gäller för kraftmetoden att man på ett någorlunda enkelt sätt måste kunna skapa flexibilitetsmatrisen, och vektorn innehållande förskjutningar svarande mot de verkliga lasterna på det statiskt bestämda systemet. Detta arbete, liksom lösandet av ekvationssystemet kan utföras med dator. Det är emellertid svårt att generalisera kraftmetoden så att ett program klarar olika former av exempelvis ramar, då oftast en speciell kunskap krävs om det aktuella problemet och dess randvillkor och dylikt. Denna begränsning gäller dock inte för Clapeyrons ekvation som lätt kan programmeras att klara olika varianter av kontinuerliga balkar. Speciella förhållanden som voter och dylikt kan också lätt läggas in i ett sådant program. För dessa fall måste dock numerisk integration av utböjningskurvan tillgripas då stödvinkeländringarna dels för angripande punktmoment  $M=1$  i ena balkänden, dels för de yttre verkande lasterna, skall bestämmas.

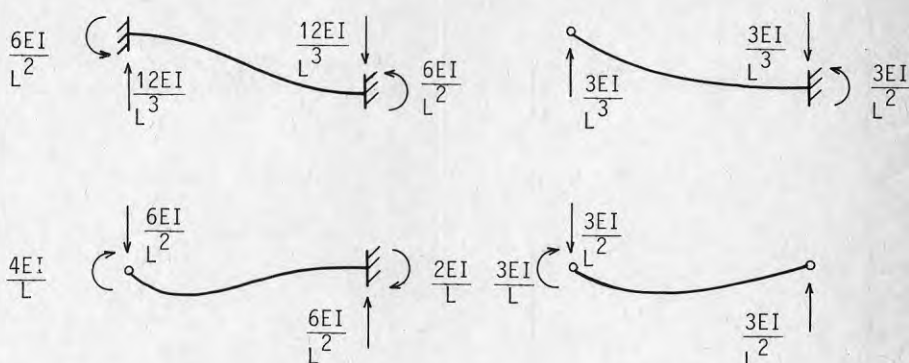


FIG 4.14 Elementarfall för analys av balkstruktur med förskjutningsmetodik.

Kraftmetodsbaserade beräkningar har sin största fördel i att de är lätta att förstå och tillämpa, framförallt för små problem. För att få en rimlig arbetsmängd är det dock viktigt att strukturen frigörs på ett sådant sätt att beräkningen av de behövliga förskjutningarna blir så enkel som möjligt. Det enklaste sättet vid beräkning av kontinuerliga balkar och ramar torde vara att konsekvent frigöra strukturen i enskilda statistiskt bestämda balkdelar, med införande av obekanta moment i balkändarna.

En annan fördel med kraftmetoden är att olika lasttyper lätt kan behandlas. För förskjutningsmetoden är man, som framgått av tidigare avsnitt, tvungen att begränsa sig till krafter som verkar i de definierade frihetsgraderna, vilket innebär att en jämnt utbredd belastning på en balk inte, utan konstgrepp, kan påföras. Denna begränsning finns inte då kraftmetoden används.

Som ovan nämnts är dock metoden svår att programmera för generella fall, vilket innebär att metoden inte är ett realistiskt alternativ för stora strukturer som är mångfalt statistiskt obestämda.

Möjligheterna att lyckas med förskjutningsmetodiken är beroende av på vilket sätt strukturen kan delas in, och på hur frihetsgraderna väljs. Ju enklare deelementen görs, ju enklare blir analysen av deelementen, men samtidigt växer storleken på ekvationssystemet.

Som sammanfattning kan sägas att förskjutningsmetoden är något mera komplicerad än kraftmetoden för enkla strukturer. Visserligen är de enstaka delfall som skall analyseras lättare, men svårigheterna uppstår då mera komplicerade lastfall skall behandlas.

Som skall visas i nästa kapitel har dock förskjutningsmetoden den stora fördelen att den relativt enkelt kan systematiseras på ett sådant sätt att en dator kan utföra det mesta arbetet. Generellt sett är därför förskjutningsmetoden att föredra som grund för skapandet av flexibla och allmängiltiga program för statistisk analys.



## 5 FINITA ELEMENT METODEN

I förra kapitlet gavs en grund för förskjutningsmetoden. I de två exempel som gavs framgick beräkningsgången för de fall då beräkningarna utförs för hand.

Det första steget bestod i att bestämma strukturens frihetsgrader, vilka var beroende av såväl strukturens geometri som de antaganden man gjort om strukturens verkningssätt, samt de laster man önskade påföra.

Nästa steg var att reducera bort de frihetsgrader som ej behövdes i analysen, på grund av att förskjutningen eller eventuellt kraftkomponenten var känd. Efter detta kvarstod strukturens egentliga frihetsgrader.

I dessa frihetsgrader pålades, i tur och ordning, förskjutningar med värdet 1, och de krafter som erfordrades för att hålla strukturen deformerad på detta sätt beräknades. Detta gav komponenterna i modellens styvhetsmatris.

När styvhetsmatrisen och vektorn med de yttre lasterna etablerats kunde förskjutningarna i frihetsgraderna bestämmas genom lösning av ett ekvationssystem.

Angreppssättet ovan är visserligen, åtminstone i princip, tillämpligt för alla typer av problem, men det är uppenbart att bildandet av styvhetsmatrisen för mera komplicerade strukturer blir mycket tidskrävande och arbetsamt. Det är lika uppenbart att den visade metodiken inte, i den form den visats, kan helt systematiseras och överlämnas till en dator. För att kunna skapa ett mer generellt datorprogram måste ett mer systematiskt angreppssätt väljas.

Vi skall i detta kapitel behandla Finita Element metoden, som är en mycket generell beräkningsmetod för beräkning av, bland annat, olika typer av statiska problem. För att, inom det begränsade utrymme som står till buds, göra den allmänna metodiken förståelig skall vi emellertid först behandla ett mellansteg, och behandla balkstrukturer med något som kan kallas en "elementbaserad förskjutningsmetod". Den metodik som detta leder fram till är egentligen densamma som en "riktig" Finit Element analys av samma fall skulle ge. Vi vill emellertid ha en speciell beteckning för denna beräkningsmetod av ett speciellt skäl. Skälet är att man i det diskuterade balkfallet får resultat som exakt överensstämmer med de enligt elasticitetsteorin beräknade, medan - och detta är viktigt - Finita Element beräkningar är i grunden approximativa, det vill säga att resultaten endast är begränsat riktiga.

Användbarheten hos de resultat som beräknas med hjälp av ett Finit Element program beror bland annat på vilken elementindelning som göres, och resultaten kan bli

mycket dåliga med fel antaganden. Några sådana problem har vi däremot inte för de balkelement som först skall studeras, åtminstone inte så länge som vi endast behandlar koncentrerade laster i frihetsgraderna.

### 5.1 Elementanalys, balkstrukturer

Vi såg i kapitel 4 hur analysen av en balkstruktur med förskjutningsmetodik kunde utföras med hjälp av fyra olika elementarfall. Förutsättningarna var då endast att vi kände varje balkelements upplagsförhållanden. Den analys som gjordes var på detta sätt starkt kopplad till den aktuella strukturen. Metodiken kan emellertid ganska lätt generaliseras till att behandla alla typer av balkstrukturer. För att åstadkomma detta skall vi dock helt vända på de tidigare nyttjade begreppen.

I den tidigare analysen utgick vi från den hela strukturen, definierade dess frihetsgrader med hänsyn till de aktuella randvillkoren, och skar sedan sönder strukturen i hanterbara element för vilka vi analyserade de nödvändiga krafterna. Dessa adderades sedan till en styvhet för hela strukturen.

I den elementbaserade metodiken utgår vi i stället från ett enda element. Om detta görs tillräckligt generellt kan sedan olika "varianter" av det användas för att bygga upp den aktuella struktur som behandlas. Det framgår ju nämligen ganska klart att vissa likheter i styvhetsbegrepp måste finnas mellan de olika markerade elementen i FIG 5.1.

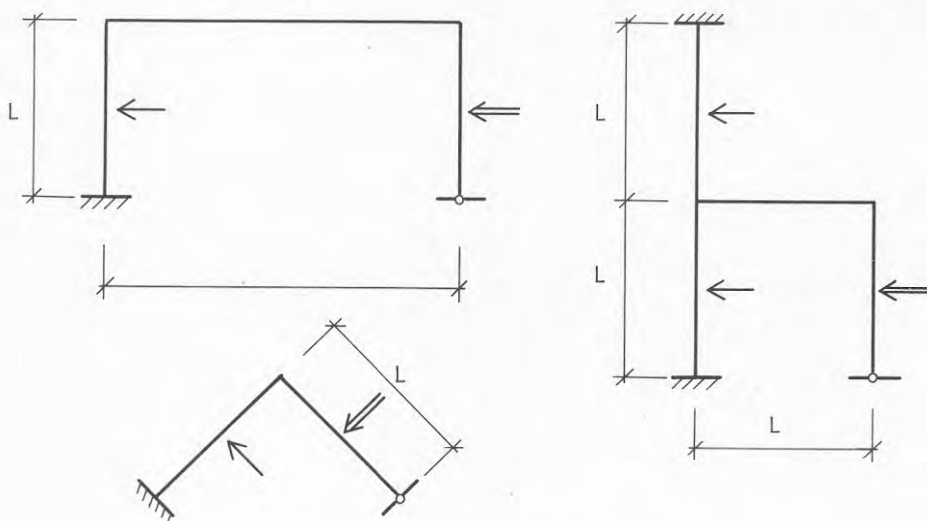


FIG 5.1 Likheter i styvhet för olika strukturer.

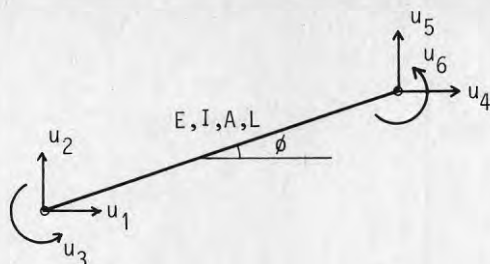


FIG 5.2 Generellt balkelement.

Hur skall då ett generellt balkelement se ut? Uppenbarligen kan man inte förutsätta någonting om vilka randvillkor som gäller för någon av elementets ändpunkter (noder, med FE-termer). Dessutom måste elementet kunna vara orienterat hur som helst i planet, och dessutom ha godtyckliga värden på tröghetsmoment, tvärsnittsarea och längd. Ett generellt balkelement kan alltså beskrivas med hjälp av FIG 5.2.

För detta kan en elementstyvhet härledas med samma grundläggande metodik som användes i kapitel 4. För att något renodla analysen skall dock först ett fall med balkelementet liggande parallellt med x-axeln behandlas.

För elementanalysen införs lokala frihetsgrader i elementet, som i ett senare skede skall omvandlas till strukturens frihetsgrader. För att kunna modellera alla randförhållanden ges elementets ändpunkter full rörelsefrihet med hjälp av tre frihetsgrader - två translationer och en rotation - enligt FIG 5.3.

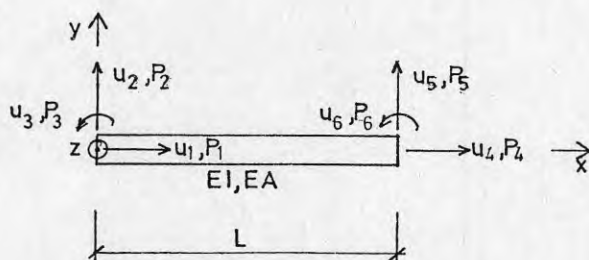


FIG 5.3 Balkelement med sex frihetsgrader.

För att nu härleda elementets styvhetsgenskaper ges först frihetsgrad 1 en förskjutning med storleken 1 i positiv riktning, medan alla övriga frihetsgrader ges förskjutningarna noll. Detta deformationstillstånd motsvarar en sammantryckning av elementet, jämför FIG 5.4a. Den tryckkraft som åtgår för detta är

$$F = \frac{EA}{L}$$

För att uppnå denna tryckkraft skall, med hänvisning till FIG 5.3, gälla att

$$\begin{aligned} P_1 &= -P_4 = F \\ P_2 &= P_3 = P_5 = P_6 = 0 \end{aligned}$$

Dessa värden utgör alltså första kolumnen i elementstyvhetsmatrisen. För att få den andra kolumnen analyseras fallet i FIG 5.4b. Detta motsvarar en stödsjunkning för en fast inspänd balk, och de krafter i frihetsgraderna som åtgår för att hålla detta deformerade läge kan fås ur tabellverk som

$$\begin{aligned} P_2 &= -P_5 = 12 \frac{EI}{L^3} \\ P_3 &= P_6 = 6 \frac{EI}{L^2} \\ P_1 &= P_4 = 0 \end{aligned}$$

På motsvarande sätt analyseras fallet i FIG 5.4c, varvid man får de nödvändiga krafterna som

$$\begin{aligned} P_2 &= -P_5 = 6 \frac{EI}{L^2} \\ P_3 &= 4 \frac{EI}{L} \\ P_6 &= 2 \frac{EI}{L} \\ P_1 &= P_4 = 0 \end{aligned}$$

Motsvarande analyser genomförs även för delfallen d-f i FIG 5.4 där förskjutningar ansätts i den högra noden. Efter dessa kan en styvhetsmatris för elementet etableras som

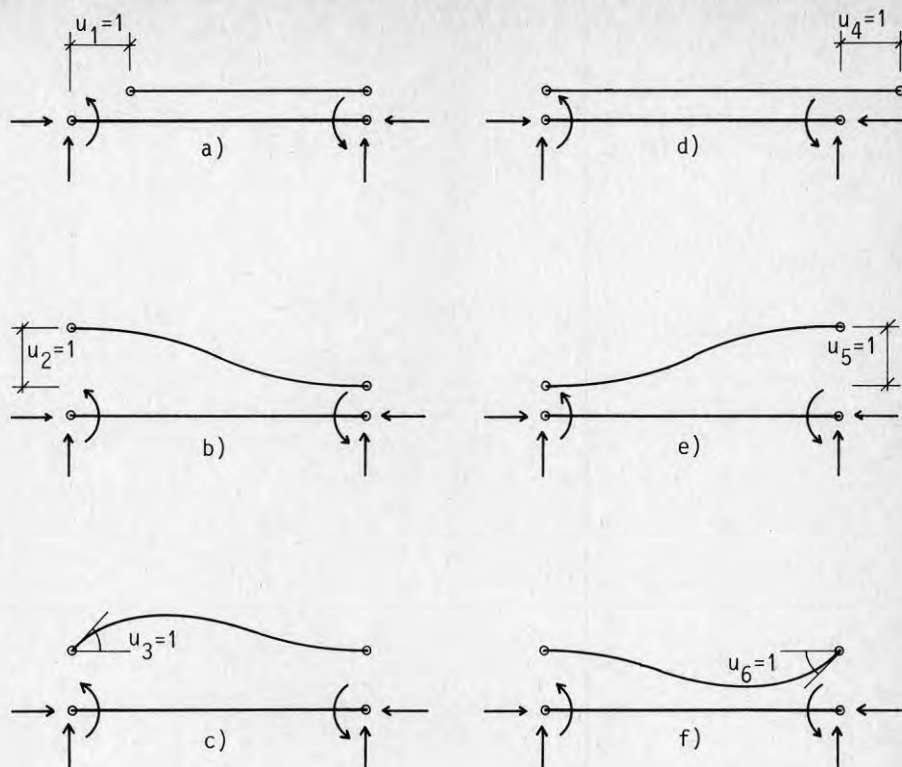


FIG 5.4 Enhetsförskjutningar i balkelement

$$(k^0) = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & 12\frac{EI}{L^3} & 6\frac{EI}{L^2} & 0 & -12\frac{EI}{L^3} & 6\frac{EI}{L^2} \\ 0 & 6\frac{EI}{L^2} & 4\frac{EI}{L} & 0 & -6\frac{EI}{L^2} & 2\frac{EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -12\frac{EI}{L^3} & -6\frac{EI}{L^2} & 0 & 12\frac{EI}{L^3} & -6\frac{EI}{L^2} \\ 0 & 6\frac{EI}{L^2} & 2\frac{EI}{L} & 0 & -6\frac{EI}{L^2} & 4\frac{EI}{L} \end{bmatrix}$$

Denna matris relaterar då krafterna i elementets frihetsgrader till förskjutningarna enligt ett samband

$$\bar{P} = (k^0) \bar{u}$$

där



$$\bar{p} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{bmatrix} \quad \bar{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix}$$

Elementstyhetsmatrisen visar ett samband mellan de krafter och förskjutningar som råder i ett element, för varje förskjutningstillstånd. Betrakta exempelvis den horisontella balken i ramen i FIG 4.13. Förskjutningstillståndet i delfigur d motsvarar då  $u_3=1$  och e motsvarar  $u_6=1$ . Resultaten överensstämmer väl med tanke på att längden för balken i FIG 4.13 är  $2L$ .

Man kan även kontrollera att elementet är i jämvikt för alla förskjutningstillstånd, vilket dock inte är speciellt uppseendeväckande med den härledning som gjorts. En projektionsekvation i x-led ger exempelvis

$$P_1 + P_4 = \left( \frac{EA}{L} u_1 - \frac{EA}{L} u_4 \right) + \left( -\frac{EA}{L} u_1 + \frac{EA}{L} u_4 \right) = 0$$

och motsvarande kan visas för en projektionsekvation i y-led, och en momentekvation.

För att kunna använda denna elementstyhetsmatris till något nyttigt måste vi emellertid ta bort den restriktion vi infört, och låta elementet bilda en vinkel med x-axeln. För det lutande elementet skall frihetsgrader definieras parallella med de globala koordinataxlarna x och y enligt FIG 5.5a, med avsikten att olika element lätt skall kunna adderas. I delfigur b har frihetsgrader i elementets eget koordinatsystem använts. Uttryckt i dessa frihetsgrader måste det nyss härledda sambandet gälla, d v s

$$\begin{bmatrix} P_1^0 \\ P_2^0 \\ P_3^0 \\ P_4^0 \\ P_5^0 \\ P_6^0 \end{bmatrix} = (k^0) \begin{bmatrix} u_1^0 \\ u_2^0 \\ u_3^0 \\ u_4^0 \\ u_5^0 \\ u_6^0 \end{bmatrix}$$

$$\bar{p}^0 = (k^0) \bar{u}^0$$

Men de lokala förskjutningarna i vektorn  $\bar{u}^0$  kan uttryckas i de globala

$$u_1^0 = u_1 \cos \theta + u_2 \sin \theta$$

$$u_2^0 = -u_1 \sin \theta + u_2 \cos \theta$$

$$u_3^0 = u_3$$

$$u_4^0 = u_4 \cos \theta + u_5 \sin \theta$$

$$u_5^0 = -u_4 \sin \theta + u_5 \cos \theta$$

$$u_6^0 = u_6$$

Dessa uttryck kan sammanfattas som en matrismultiplikation

$$\bar{u}^0 = (T) \bar{u}$$

På motsvarande sätt härledes att för de globala och lokala krafterna gäller

$$\bar{P} = (T)^T \bar{P}^0$$

där matrisen (T) transponerats. Som framgår av härledningen ovan innehåller transformationsmatrisen (T) termer som består av cosinus och sinus för elementets lutningsvinkel. Som slutresultat fås ett uttryck för krafterna i de globala frihetsgraderna

$$\bar{P} = (T)^T \bar{P}^0 = (T)^T (k^0) \bar{u}^0 = (T)^T (k^0) (T) \bar{u}$$

som alltså ger sambandet mellan krafterna och förskjutningarna i de globala frihetsgraderna.

Härledningen av elementstyvhetsmatrisen har nu kommit så långt att uttrycken är beroende av de parametrar som antogs definiera elementets egenskaper, vilket är slutpunkten för elementanalysen. För att fortsätta analysen av en hel ram måste nu en vidare vy anläggas och hela strukturen betraktas.

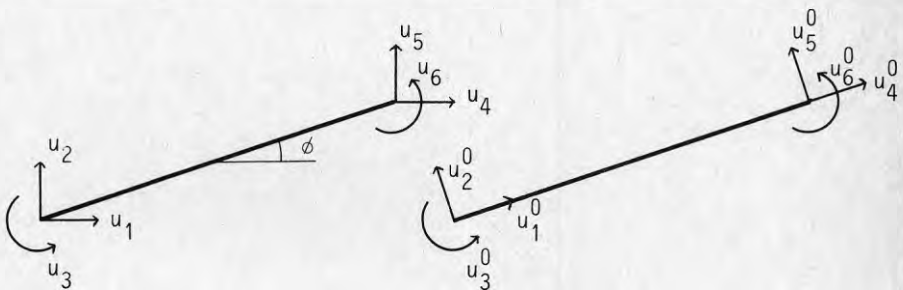


FIG 5.5 Frihetsgrader för lutande balkelement

## 5.2 Strukturanalys

I avsnitt 5.1 har ett samband mellan krafter och förskjutningar för ett godtyckligt balkelement härletts. Detta samband uppfyllde kraven på jämvikt inom elementet. För att nu "bygga upp" en större struktur med hjälp av denna typ av enkla element skall jämviktsvillkor och kontinuitetsvillkor utnyttjas. Avsikten med denna strukturanalys är att formulera ett samband mellan krafter och förskjutningar i strukturens frihetsgrader, av den principiella formen

$$\bar{R} = (K) \bar{r}$$

vilket alltså överensstämmer med uttrycken som härledes i avsnitt 4.6.

I stället för att ha fått fram delarna genom uppskärning av den totala strukturen skall olika element av standardtyp nu tvingas ihop så att de tillsammans bildar den betraktade strukturen. Till att börja med tas inte någon hänsyn till de olika elementens upplagsförhållanden. Alltså fortsätter analysen med tre frihetsgrader per nod.

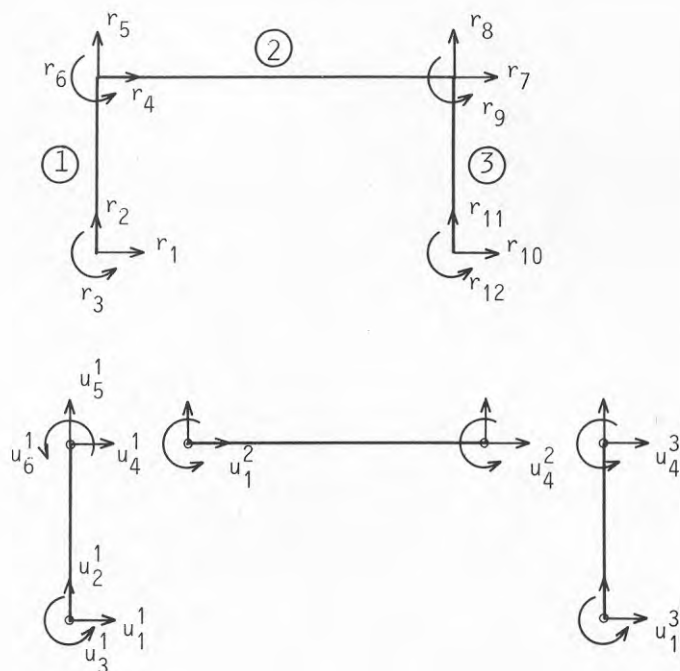


FIG 5.6 Ram för analys med elementbaserad metodik

Som ovan nämndes måste nu villkor om jämvikt och kontinuitet införas (det är dessa som definierar uppbyggnaden). Som ett exempel skall ramen i avsnitt 4.6.1 analyseras igen, FIG 4.13 och 5.6. Ramen har nu till att börja med 12 frihetsgrader.

Börja med att betrakta det vänstra rambenet. Efter jämförelse med avsnitt 5.1 inses att elementstyhetsmatrisen för detta element kan fås med hjälp av avsnitt 5.1 (vinkeln=90). Med de aktuella värdena på parametrarna fås ett styvhets samband

$$\bar{p}^1 = (k^1) \bar{u}^1$$

där  $\bar{p}^1$  och  $\bar{u}^1$  är elementkrafter respektive elementförskjutningar för element 1. På motsvarande sätt fås för elementen 2 och 3

$$\bar{p}^2 = (k^2) \bar{u}^2 \qquad \bar{p}^3 = (k^3) \bar{u}^3$$

För att elementen tillsammans skall bygga upp den aktuella strukturen skall kontinuitetskraven införas. Ur FIG 5.6 fås genom jämförelse av elementens lokala frihetsgradsnummer och strukturens globala att

$$\begin{array}{ll} u_1^1 = & r_1 \\ u_1^1 = & r_2 \\ u_3^1 = & r_3 \\ u_5^1 = u_2^2 = & r_5 \\ u_2^2 = u_4^3 = & r_7 \\ u_4^2 = u_6^3 = & r_9 \\ u_3^3 = & r_{11} \\ u_2^3 = & r_{12} \end{array} \qquad \begin{array}{ll} u_2^1 = & r_2 \\ u_1^1 = u_2^2 = & r_4 \\ u_6^2 = u_3^3 = & r_6 \\ u_5^2 = u_5^3 = & r_8 \\ u_1^3 = & r_{10} \\ u_3^3 = & r_{12} \end{array}$$

Dessa villkor kan utnyttjas för att i elementstyhetsrelationen för element 1 "byta ut" förskjutningarna i  $\bar{u}^1$  enligt

$$\begin{array}{|c|} \hline P_1^1 \\ \hline P_2^1 \\ \hline P_3^1 \\ \hline P_4^1 \\ \hline P_5^1 \\ \hline P_6^1 \\ \hline \end{array} = (k^1) \begin{array}{|c|} \hline r_1 \\ \hline r_2 \\ \hline r_3 \\ \hline r_4 \\ \hline r_5 \\ \hline r_6 \\ \hline \end{array}$$

och motsvarande för element 2 och 3. På detta sätt relateras krafterna som verkar i de olika elementen till de förskjutningar som definierar hela strukturens förskjutningstillstånd. Dessutom fås ju också att till exempel  $u_4^1 = u_1^2$ , vilket är ett villkor för att ramhörnet skall "gå ihop".

Med detta är alltså kontinuitetsvillkoret uppfyllt, och det som återstår är att kräva jämvikt i strukturen. Vi har tidigare konstaterat att varje element är i jämvikt för varje förskjutningstillstånd och skall nu se till att alla kopplingspunkter också är det. Betrakta först kraften i frihetsgrad 4 ( $R_4$ ).

Enligt definitionen är till exempel kraftkomponenten  $P_4^1$  den kraft i lokal frihetsgrad 4 som går åt för att hålla element 1 i ett deformerat läge, beskrivet av vektorn  $\bar{u}^1$  eller, enligt ovan, vissa komponenter av vektorn  $\bar{r}$ . På motsvarande sätt är  $P_1^2$  en kraft som åtgår för att hålla element 2 i samma läge.

Ur FIG 5.6 framgår då, eftersom kraftkomponent  $R_4$  skall vara den kraft som åtgår för att hålla hela strukturen deformerad, att denna kraft är summan av de två elementkrafterna, det vill säga att

$$R_4 = P_4^1 + P_1^2$$

ett kraftbidrag från vardera av de två anslutande elementen, jämför också additionen av erforderliga krafter i avsnitt 4.6.1, FIG 4.13.

Genom en upprepning av detta resonemang fås för de olika kraftkomponenterna

$$\begin{array}{ll} P_1^1 & = R_1 \\ P_3^1 & = R_3 \\ P_5^1 + P_2^2 & = R_5 \\ P_4^2 + P_4^3 & = R_7 \\ P_6^2 + P_6^3 & = R_9 \\ P_2^3 & = R_{11} \end{array} \qquad \begin{array}{ll} P_2^1 & = R_2 \\ P_4^1 + P_1^2 & = R_4 \\ P_6^1 + P_3^2 & = R_6 \\ P_5^2 + P_5^3 & = R_8 \\ P_1^3 & = R_{10} \\ P_3^3 & = R_{12} \end{array}$$

ett samband som har vissa likheter med kontinuitetsvillkoren ovan.

För att nu etablera det totala kraftsambandet skall ett litet trick användas. Skriv först den totala kraftvektorn  $\bar{R}$  som



$$\begin{array}{c}
 \begin{array}{|c|} \hline R_1 \\ \hline R_2 \\ \hline R_3 \\ \hline R_4 \\ \hline R_5 \\ \hline R_6 \\ \hline R_7 \\ \hline R_8 \\ \hline R_9 \\ \hline R_{10} \\ \hline R_{11} \\ \hline R_{12} \\ \hline \end{array}
 =
 \begin{array}{|c|} \hline P_1^1 \\ \hline P_2^1 \\ \hline P_3^1 \\ \hline P_4^1 \\ \hline P_5^1 \\ \hline P_6^1 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline \end{array}
 +
 \begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline P_1^2 \\ \hline P_2^2 \\ \hline P_3^2 \\ \hline P_4^2 \\ \hline P_5^2 \\ \hline P_6^2 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline \end{array}
 +
 \begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline P_4^3 \\ \hline P_5^3 \\ \hline P_6^3 \\ \hline P_1^3 \\ \hline P_2^3 \\ \hline P_3^3 \\ \hline \end{array}
 \end{array}$$

där tre "expanderade" element-kraftvektorer adderas till en struktur-kraftvektor. Den andra vektorn i högerledet skrivs nu med hjälp av elementets styvhetsrelation som

$$\begin{array}{c}
 \begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline P_1^2 \\ \hline \vdots \\ \hline P_6^2 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline \end{array}
 =
 \begin{array}{|c|} \hline 0 \ 0 \ 0 \\ \hline 0 \ 0 \ 0 \\ \hline 0 \ 0 \ 0 \\ \hline \vdots \\ \hline 0 \\ \hline \end{array}
 \begin{array}{|c|} \hline k_{11}^2 \ \dots \ k_{16}^2 \\ \hline \vdots \\ \hline k_{11}^2 \ \dots \ k_{16}^2 \\ \hline \end{array}
 \begin{array}{|c|} \hline 0 \\ \hline 0 \ 0 \ 0 \\ \hline 0 \ 0 \ 0 \\ \hline 0 \ 0 \ 0 \\ \hline \end{array}
 \begin{array}{|c|} \hline r_1 \\ \hline r_2 \\ \hline r_3 \\ \hline r_4 \\ \hline \vdots \\ \hline r_9 \\ \hline r_{10} \\ \hline r_{11} \\ \hline r_{12} \\ \hline \end{array}
 \end{array}$$

där alltså styvhets sambandet för element 2 uttrycks med matriser av dimensionen 12. Man ser hur innehållet i första raden av den lilla elementstyvhetsmatrisen hamnar på fjärde raden i detta expanderade globala styvhetsuttryck. Detta är en följd av kontinuitetsvillkorets utsaga att, för element 2, den lokala frihetsgrad 1 skall vara lika med den globala nummer 4. Man ser vidare att även elementstyvhetsens kolumn 1 hamnar i kolumn 4 av strukturens styvhetsmatris.

På samma sätt fås elementstyvhetsens övriga innehåll:

rad (kolumn) 2	ger bidrag till rad (kolumn)	5
3		6
4		7
5		8
6		9

Innebörden av detta är att kontinuitetsvillkoret givit en översättningstabell mellan lokala och globala frihetsgradsnummer.

Genomförs detta resonemang även för elementen 1 och 3 kan additionen av de tre elementkraftvektorerna ske. Om de expanderade elementstyvhetsmatriserna kallas

$(k_e^1)$ ,  $(k_e^2)$  och  $(k_e^3)$  fås att

$$\bar{R} = ( (k_e^1) + (k_e^2) + (k_e^3) ) \bar{r} = (K) \bar{r}$$

Den totala styvhetsmatrisen byggs alltså upp som summan av de expanderade elementstyvhetsmatriserna. Genom utvecklande av de tre elementstyvhetsmatriserna inses att efter additionen raderna (kolumnerna) 1-3 i strukturstyvhetsmatrisen (K) kommer att bestå enbart av tal hämtade från elementstyvhetsmatrisen för element 1, medan raderna (kolumnerna) 4-6 består av summan av tal från såväl  $(k_e^1)$  som  $(k_e^2)$ . Rader och kolumner svarande mot en frihetsgrad där två eller flera element kopplas samman kommer alltså att utgöras av summor av bidrag från de anslutande elementen.

Förfarandet har så här långt varit mycket systematiskt, och därför lätt att programmera. Det uttryck för elementstyvhetsmatrisen som härleddes i förra avsnittet kan ju bestämmas direkt ur vissa parametrar, och det enda som ytterligare behövs är att man, för varje element, etablerar tabellen över de globala frihetsgradsnummer som svarar mot de lokala.

Man skall dock märka att denna systematisering skett till en kostnad av att den framtagna styvhetsmatrisen är av dimensionen (12/12) i stället för den (3/3) matris som beräknades i exemplet i avsnitt 4.6.1. Detta är delvis en följd av att ingen reduktion av antalet frihetsgrader gjorts med hänsyn till de aktuella randvillkoren. Den beskrivning av strukturen som definieras av den nu härledda styvhetsmatrisen är dock också något noggrannare genom att normalkrafterna i balkelementen beaktats.

### 5.3 Randvillkor, laster och lösning

I de tidigare avsnitten i detta kapitel har visats hur strukturstyvhetsmatrisen för en balkstruktur kan byg-

gas upp. Den matris som skapats tar ingen hänsyn till vare sig de laster som skall läggas på strukturen eller de upplagsförhållanden som gäller. Som tidigare visats så innehåller matrisen ett fullt antal frihetsgrader.

Med ett definierat förskjutningstillstånd karakteriserat av vektorn  $\bar{r}$  fås de erforderliga krafterna i strukturen (vektorn  $\bar{R}$ ) genom matrismultiplikationen

$$\bar{R} = (K) \bar{r}$$

Men i de allra flesta fallen är det ju det motsatta sambandet vi önskar. För vissa pålagda laster vill vi ha fram de resulterande förskjutningarna. Formellt förefaller det vara ett enkelt problem, då det nämligen borde gälla att förskjutningsvektorn är lika med inversen av styvhetsmatrisen gånger lastvektorn. Denna beräkning låter sig emellertid ej utföras, eftersom strukturstyvhetsmatrisen i den form den beräknats är singulär (den har en rangdefekt av ordningen 3 för de behandlade fallen). Detta är dock inte bara ett "matematiskt" problem utan speglar en fysikalisk sanning. Antag nämligen att man beräknat ett visst krafttillstånd svarande mot ett visst förskjutningstillstånd. Man kan ju sedan förskjuta hela strukturen en viss sträcka i (till exempel) x-led, det vill säga öka förskjutningen i alla frihetsgrader i x-riktningen med en viss kvantitet. Uppenbarligen måste samma krafttillstånd råda i strukturen även efter denna stelkroppsförskjutning. Rangdefekten på tre i strukturstyvhetsmatrisen svarar exakt mot de tre upplagskrafter vi (minst) måste ta med i analysen. Som diskuterades i avsnitt 4.2 innebär detta att (minst) tre frihetsgrader skall ges ett fixt förskjutningsvärde (ofta lika med noll).

Om exemplet i FIG 5.6 åter betraktas så påverkar upplagsförhållandena strukturen på så sätt att i punkt A såväl förskjutningar som rotation är förhindrade, det vill säga att  $r_1 = r_2 = r_3 = 0$ . I punkt D är förskjutningarna förhindrade, alltså är  $r_{10} = r_{11} = 0$ . Dessa randvillkor måste föras in i strukturstyvhetsmatrisen i samband med lösningen av styvhetsrelationen för ett givet lastfall.

De yttre laster som svarar mot det definierade lastfallet är, med hänvisning till FIG 5.6

$$\begin{aligned} R_4 &= P \\ R_5 &= R_6 = R_7 = R_8 = R_9 = R_{12} = 0 \end{aligned}$$

De krafter som råder i de fixerade frihetsgraderna är däremot (åtminstone generellt sett) okända. Som påpekades i avsnitt 4.2 så måste ju dessa upplagskrafter få ett värde som är en funktion av de pålagda lasterna.

Analysen måste alltså fortsätta med antagandet att dessa upplagskrafter är okända.

Betrakta nu, exempelvis, den fjärde ekvationen i styvhetssambandet. Den utsäger att

$$K_{41}r_1 + K_{42}r_2 + K_{43}r_3 + K_{44}r_4 + \dots + K_{412}r_{12} = R_4$$

Men vissa av de ingående förskjutningskomponenterna är ju inte okända utan har vissa bestämda värden. De kan då tas bort som obekanta i ekvationen och föras över till högerledet, enligt

$$\begin{aligned} & K_{44}r_4 + K_{45}r_5 + \dots + K_{49}r_9 + K_{412}r_{12} = \\ & = R_4 - K_{41}r_1 - K_{42}r_2 - K_{43}r_3 - K_{410}r_{10} - K_{411}r_{11} = \\ & = R_4 \end{aligned}$$

På motsvarande sätt kan även de övriga ekvationerna där R-kraften är känd behandlas. En analys av de ekvationer som kvarstår ger vid handen att man nu har kvar sju ekvationer i de sju obekanta förskjutningskomponenterna. Det system som på detta sätt åstadkommit är inte bara lösbart utan även positivt definit (Appendix) vilket innebär att problemet är snällt ur lösningssynpunkt. Om man betraktar det färdiga ekvationssystem som nu skall lösas ser man att skillnaden mot den ursprungliga styvhetsrelationen är att rader och kolumner svarande mot fixerade förskjutningskomponenter strukits bort. Dessutom har högerledets kraftkomponenter modifierats med hänsyn till de kända förskjutningarna, även om detta i det betraktade fallet där alla fixeringar skett till värdet noll inte föranlett några förändringar.

Det skapade ekvationssystemet kan lösas med hjälp av de standardrutiner som finns tillgängliga, och lösningen är de sju okända förskjutningskomponenterna. Om dessa placeras in på sina platser i vektorn  $\bar{r}$  och denna kompletteras med de fem förhandsbestämda förskjutningarna, är vektorn  $\bar{r}$  fullständigt definierad.

Med den beräknade förskjutningsvektorn  $\bar{r}$  kan sedan den tillhörande kraftvektorn  $\bar{R}$  beräknas. Eftersom detta använts som villkor kommer krafterna i de icke-fixerade frihetsgraderna att få de värden som angivits ovan (P i frihetsgrad 4, noll i övriga). För de fixerade frihetsgraderna beräknas också vissa kraftvärden. Dessa är de reaktionskrafter som åtgår för att fixera strukturen i det definierade läget.

Som en sammanfattning av diskussionen om randvillkor kan sägas att kraftvektorn är entydigt definierad av förskjutningsvektorn, men att motsatsen inte gäller.

Vidare har av exemplet framgått att det inte gör någon skillnad om strukturen är statistiskt obestämd, det enda väsentliga är att (för det plana fallet) minst tre förskjutningar fixeras på ett sådant sätt att stelkroppsförflyttningar är förhindrade.

#### 5.4 Behandling av resultat

Den lösning av strukturens styvhetsrelation som fås ger alltså förskjutningarna i strukturens frihetsgrader. Normalt är det dock inte dessa man som konstruktör är mest intresserad av, utan det man vill ha är krafter och moment i olika punkter i strukturen.

För att få fram dessa återvänder man till elementnivån. Med hjälp av den tabell över globala frihetsgradsnummer som kontinuitetsvillkoren gav upphov till kan man nu plocka fram ett visst elements förskjutningar  $\bar{u}^i$  ur strukturförskjutningsvektorn  $\bar{r}$ .

När elementets förskjutningsvektor är känd kan sedan elementets styvhetsrelation användas för att beräkna de krafter som verkar i elementets frihetsgrader. Man får med hjälp av sambanden i avsnitt 5.1 först förskjutningarna i de lokala frihetsgradernas riktning som (jfr FIG 5.5)

$$\bar{u}^0 = (T) \bar{u}^i$$

och sedan krafterna i elementets lokala frihetsgrader som

$$\bar{P}^0 = (k^0) \bar{u}^0$$

Eftersom kraftkomponenterna i vektorn  $\bar{P}^0$  utgörs av dels punktkrafter, dels moment i elementets ändpunkter är det sedan ur dessa lätt att beräkna de aktuella spänningarna i varje punkt inom elementet. Man noterar också att dessa sex kraftkomponenter är de enda som kan definieras för elementet, med den valda representationen. Detta innebär till exempel att momentet varierar linjärt mellan de två ändvärdena, vilket ju också är det riktiga förhållandet då endast punktkrafter och punktmoment antas belasta strukturen. Vad som dock är värre är att det även sätter en gräns för hur andra belastningstillstånd kan återges. I avsnitt 6.1.4 kommer till exempel metoder att representera utbredda laster på balkstrukturer att behandlas. Ur ovanstående framgår emellertid att momentfördelningen i ett balk-element alltid måste vara linjär, så hur man än betar sig kommer momentkurvan under en utbredd last att beskrivas med någon form av polygon. Det enda man kan göra är att försöka få fram en polygon som så nära som möjligt återger den krökta kurvan.



Den närmast till hands liggande, om än inte den bästa, metoden att representera, till exempel, en fritt upplagd balk utsatt för en jämnt fördelad belastning är att dela in balkens längd i ett antal delar, och i varje kopplingspunkt låta en punktkraft verka. I FIG 5.7 finns en sådan modell visad med 4 delar. I var och en av de inre knutpunkterna skall en punktkraft med storleken  $(qL/4)$  verka. De beräknade momentvärdena för detta fall överensstämmer med de riktiga i fjärdedelspunkterna, men mellan dessa avviker den beräknade momentkurvan från den sanna parabeln. Ju finare indelning som görs desto bättre kommer anpassningen att bli. Att välja hur man skall göra är därför en avvägning av behovet av noggrannhet mot kostnaden för lösning av ett system med fler frihetsgrader.

När det gäller nedböjningen av balken påverkas den i större grad än momentfördelningen av den indelning av balken som görs. I TABELL 5.1 redovisas, för olika antal delelement, den beräknade mittnedböjningen ( $w_p$ ) för en balk enligt FIG 5.7 ( $n$ =antalet delelement)

Man ser tydligt av tabellen hur snabbt noggrannheten i den beräknade mittnedböjningen förbättras då antalet delelement ökas. Man bör dock komma ihåg att för detta fall den beräkningstid som åtgår i datorn växer mer än linjärt med antalet noder. I kapitel 6 kommer en annan metodik för representation av jämnt utbredda laster att diskuteras, en metod som, även om den utan en noggrannare analys kan upplevas som "fusk" ger goda resultat.

TABELL 5.1 Nedböjning för fritt upplagd balk belastad med jämnt fördelad last  $q$ , representerad med hjälp av punktlaster av storlek  $(qL/n)$ .  
(multiplikator  $(qL^4/EI)$  ).

$n$	$w_p$	FEL (%)
2	0.0104167	20.0
4	0.0123698	5.00
8	0.0128581	1.25
16	0.0129801	0.31
32	0.0130106	0.08
Exakt $(\frac{5}{384}) = 0.0130208$		

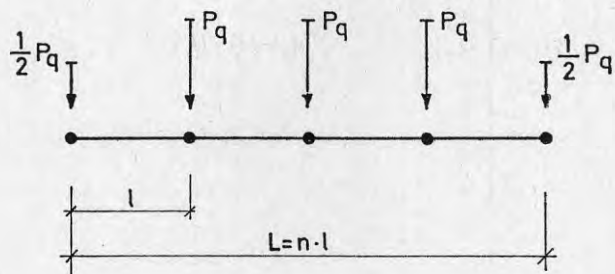


FIG 5.7 Representation av jämnt fördelad last med koncentrerade laster. ( $n$ =antalet delningar)

### 5.5 Beräkningsgång i FE-analys

I de tidigare avsnitten i detta kapitel har beräkningsgången för balkstrukturer med hjälp av elementbaserad förskjutningsmetod beskrivits. I en praktisk analys av denna typ av problem, och i andra typer av Finit Element analys kan följande steg urskiljas:

- 1) Problemdefinition. Välj lämpligt grundelement, identifiera element, nodpunkter och frihetsgrader samt ange problemets randvillkor och bestäm givna yttre laster.
- 2) Elementanalys. Bilda elementstyvhetmatriser för ingående element. Transformera till det globala koordinatsystemets riktningar.
- 3) Strukturanalys. Sätt in elementstyvheterna i jämviktssambanden för noderna, vilket resulterar i en strukturstyvhetmatris.
- 4) Ekvationslösning. Lös ekvationssystemet med beaktande av de ansatta randvillkoren.
- 5) Resultatbehandling. Bestäm elementkrafter och eventuellt spänningar ur de beräknade förskjutningarna och elementstyvhetssambanden.
- 6) Kontroll. Kontrollera lösningens rimlighet bland annat med avseende på:
  - Är givna randvillkor uppfyllda.
  - Är jämvikten uppfylld (kraftsummor lika med noll)
  - Är erhållna förskjutningar rimliga

Av dessa sex punkter är punkterna 1) och 6) direkt probleberoende och måste göras manuellt. De övriga stegen 2) - 5) kan däremot utföras helt automatiskt av datorn. Detta är just fördelen med den elementbaserade metodiken, och vi skall i nästa avsnitt visa hur exakt samma grundläggande tillvägagångssätt kan tillämpas på såväl andra statistiska system ( skivor, plattor ) som på helt andra problemtyper som värmeflöden, grundvattenströmning o dyl.

Efter att manuellt ha gjort det grundläggande steg 1) måste man överföra viss information till datorn, för att den skall kunna göra analysen. För de flesta Finita Element beräkningar kan denna information om en struktur man önskar analysera delas upp i fem delar, nämligen

- a) Nodinformation
- b) Elementinformation
- c) Materialinformation
- d) Randvillkor
- e) Belastningar

även om informationsdelarna ofta är mer eller mindre sammanflätade. I det fall som behandlades i föregående avsnitt fördes de olika delarna in på följande sätt.

Nodinformationen var dels det underförstådda antagandet att varje nod försågs med tre frihetsgrader. I detta fall behövde vi inte explicit ge de olika nodpunkternas koordinater (som man ofta måste i verkliga program) men denna information finns likväl där i form av längder och vinklar för de olika balkelementen.

Elementinformationen är i detta fall upplysningen att vi använt exakt denna typ av balkelement (andra är tänkbara). Till denna grupp kan också föras informationen om elementens topologi, det vill säga den tabell över globala frihetsgradsnummer, svarande mot de lokala frihetsgradsnumren 1 - 6, som behövdes vid skapandet av strukturstyvhetsmatrisen. I ett verkligt program talar man dock snarast om de två noder som elementet "sammanbinder" och låter programmet beräkna de sex frihetsgradsnumren.

Den materialinformation som behövs är uppgifter om balkarnas tvärsnittsareor och deras tröghetsmoment samt den elasticitetsmodul som antas gälla. Med den formulering som valdes måste också balkelementens längder och lutningsvinklar definieras. Om dock i stället de olika nodpunkternas koordinater givits hade dessa uppgifter kunnat beräknas av datorn.

Informationen om de aktuella randvillkoren var uppgifterna om vilka frihetsgrader som var fixerade. I detta speciella fall var alla de givna frihetsgradernas förskjutningar lika med noll. Andra randvillkor kan dock tänkas för balkstrukturer. Exempelvis kan en stödsjunkning med ett visst värde införas på samma sätt som

nollförskjutningen - enda skillnaden är att högerledet i ekvationssystemet måste manipuleras. Även elastiska upplag, där upplagsreaktionen är proportionell mot rörelsen kan i många program modelleras med hjälp av anbringande av fjädrar med vissa egenskaper.

Belastningarna var i detta fall enbart punktkrafter i någon av strukturens frihetsgrader. Man kan emellertid även tänka sig fördelade belastningar, punktlaster mellan knutpunkterna eller exempelvis temperaturskillnader mellan balkelementets två sidor.

## 5.6 Andra problemtyper

Som förutskickats i tidigare avsnitt kan den elementbaserade metodiken användas för andra typer av problem än analys av balkstrukturer. För dessa måste dock normalt en regelrätt Finit Element analys göras. Vi vill återigen passa på att påpeka att detta innebär att de lösningar som beräknas är approximativa, det vill säga begränsat noggranna.

### 5.6.1 Teoretisk bakgrund

Vad är då Finita Element metoden? Teoretiskt sett är det en generell, matrisbaserad metod för lösning av differentialekvationer eller system av sådana. De matriser som definierar lösningen kan, ur en matematikers synvinkel, härledas direkt ur den differentialekvation som gäller, genom ansatser av vissa approximativa funktionsuttryck. I de fall då de sökta storheterna är så konkreta som krafter och deformationer kan dock uttrycken lättare formuleras (och förstås) med hjälp av vanliga statistiska samband. I denna bok kommer inget annat än denna mera direkta metodik att nyttjas. För den som är intresserad av den bakomliggande metodiken hänvisas till Strang & Fix eller Bathe & Wilson.

I avsnitt 5.6.2 kommer den mer generella metodiken att skisseras, tillämpad på skivproblem. Motsvarande metodik för härledning av balkelementets styvhetsgenskaper kommer också något att beröras i avsnitt 6.1.4 där utbredda belastningar behandlas.

### 5.6.2 Analys av skivproblem

I den schematiserade gången för en elementbaserad analys som skisserades i det föregående avsnittet är det främst punkt 2), det vill säga elementanalysen som förändras då man i stället för en balkstruktur vill behandla till exempel ett skivproblem.

Studera FIG 5.8 och antag att spänningstillståndet i plåten, utsatt för en dragkraft, söks. Efter att, i enlighet med avsnitt 4.3, ha utnyttjat symmetrin till att ta ut endast den ena fjärdedelen av plåten kan ett antal element skapas genom en tänkt uppskärning av plåten, enligt delfigur b. Observera den finare indelning som görs i den nedre delen, med avsikt att få en acceptabel beskrivning av spänningstillståndet i denna del där spänningsgradienterna kan förväntas vara stora. Den skapade beräkningsmodellen består nu av 181 stycken triangulära skivor, och den innehåller 112 stycken noder i skärningspunkterna av de tänkta skärningslinjerna.

Förskjutningarna i skivan antas nu kunna beskrivas med hjälp av två frihetsgrader i varje nodpunkt, det vill säga totalt 224 frihetsgrader. De element som då uppstår blir, som visas i delfigur c, triangulära skivor med en nod i varje hörn, vardera med två frihetsgrader.

Att analysera de krafter som åtgår för att förskjuta, exempelvis nod 1 ett mått lika med ett i x-riktningen är dock inte så lätt. För denna typ av problem måste därför ett speciellt tillvägagångssätt användas. Mycket kortfattat kan man säga att man beskriver förskjutningarna i x-led och y-led inom det betraktade elementet med ett polynom i punktens koordinater x och y som

$$u(x,y) = A_0 + A_1 x + A_2 y$$

$$v(x,y) = B_0 + B_1 x + B_2 y$$

Om man därefter utnyttjar villkoret att de beräknade förskjutningarna skall överensstämma med de verkliga i de tre hörnpunkterna kan konstanterna  $A_0$ ,  $A_2$  och  $B_0$ ,  $B_2$  bestämmas, som funktion av frihetsgradernas förskjutningar.

Ur den gjorda förskjutningsansatsen kan töjningarna sedan beräknas. Med uttrycken enligt ovan fås

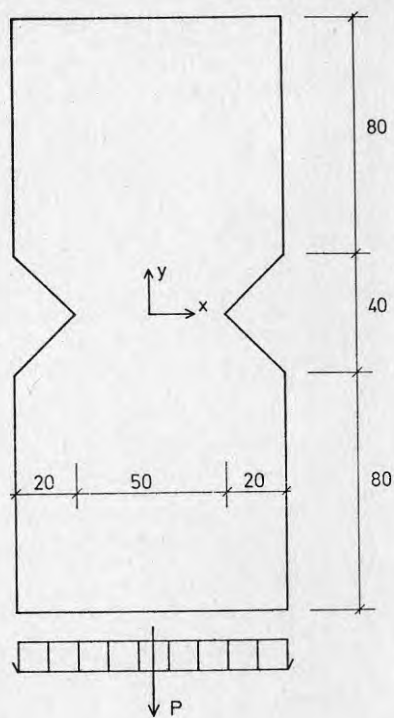
$$\epsilon_x = \frac{\partial u(x,y)}{\partial x} = A_1$$

$$\epsilon_y = \frac{\partial v(x,y)}{\partial y} = B_2$$

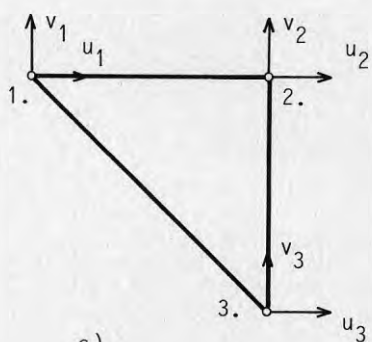
$$\gamma_{xy} = \frac{\partial u(x,y)}{\partial y} + \frac{\partial v(x,y)}{\partial x} = A_2 + B_1$$

Med kännedom om det aktuella materialets elasticitetsmodul och dess Poisson-tal kan ur dessa töjningar spänningarna inom elementet beräknas, då i det aktuella fallet ett plant spänningstillstånd (avsnitt 6.2.1) kan antas föreligga. När spänningarna är kända kan det inre arbetet av den gjorda förskjutningen beräknas. Om vidare detta inre arbete jämförs med det yttre arbetet av de erforderliga krafterna fås ett styvhetssamband som relaterar de sex krafterna i elementets frihetsgrader till motsvarande förskjutningar.





a)



c)

b)

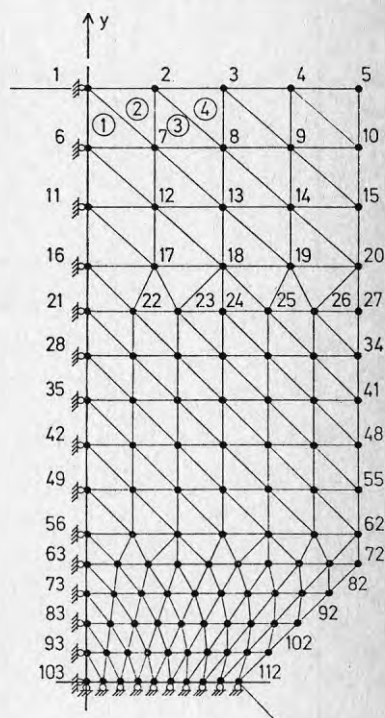


FIG 5.8 FE-analys av dragen plåt

När väl elementstyvhetismatrisen är etablerad fortskri-  
der beräkningen på precis samma sätt som för balkstruk-  
turen. Genom att en översättningstabell skapas mel-  
lan elementfrihetsgrader och strukturfrihetsgrader ut-  
nyttjas kontinuitets- och jämviktsvillkor i noderna  
till att skapa en styvhetsmatris för hela strukturen.

De randvillkor som skall införas i samband med lösning-  
en av styvhetsrelationen fås för det aktuella fallet  
ur FIG 5.8b. I detta fall skall införas att noderna i  
symmetrilinjerna inte får röra sig vinkelrätt mot des-  
sa. Alltså skall noderna 103-112 fixeras mot rörelser  
i y-led och noderna 1,6,...,103 mot rörelser i x-led.  
I den styvhetsrelation som formuleras skall vidare las-  
ter i y-led införas i noderna 1 - 5. För att efterlik-  
na den givna jämna dragspänningen införes krafter av  
relativ storlek 1 i noderna 2-4 och krafter = 0.5 i  
noderna 1 och 5.

Efter att det uppställda ekvationssystemet lösts och  
därmed förskjutningarna i frihetsgraderna beräknats  
kan spänningarna i de olika elementen beräknas. Detta  
kan göras med de samband som användes vid bildandet av  
styvhetsmatrisen. Ur de olika nodernas koordinater och  
de beräknade förskjutningarna i elementets frihetsgra-  
der är konstanterna  $A_0 - A_2$  och  $B_0 - B_2$  definierade,  
och därefter kan töjningarna beräknas. Som framgår av  
formlerna är dessa töjningar konstanta inom varje ele-  
ment, varför denna elementtyp fått beteckningen CST  
( Constant Strain Triangle ).

En mängd andra typer av element för skivberäkningar  
kan också konstrueras. Alla olika typer har olika egen-  
skaper i fråga om noggrannhet. Några olika typer visas  
i avsnitt 6.2.2, FIG 6.6.

Det som kan skilja olika elementtyper, och det som ut-  
gör approximationen i beräkningen är den förskjutnings-  
ansats som gjorts. Vid härledningen av elementstyvhe-  
ten har nämligen antagits att förskjutningarna  $u$  och  $v$   
inom elementet varierar linjärt mellan nodvärdena. Det-  
ta är uppenbarligen inte sant, speciellt inte i de  
fall där spänningarna varierar snabbt. I detta fall  
har just denna typ av skivelement dålig noggrannhet.

En intressant fråga i detta sammanhang är då varför de  
balkelement som användes i tidigare avsnitt ger exakta  
resultat som helt överensstämmer med dem som ges av  
elasticitetsteorin, åtminstone då de endast utsätts  
för koncentrerade nodkrafter och nodmoment. Detta kan  
visas ha att göra med den förskjutningsrepresentation  
som antagandena utgör. För varje element har ju anta-  
gits att böjdeformationen beskrivs av dels förskjut-  
ningarna  $r_2$  och  $r_5$ , dels ändrotationerna  $r_3$  och  $r_6$   
(jämför FIG 5.3). Med fyra parametrar kan ett tredje-  
gradspolynom bestämmas som överensstämmer med nodvär-  
dena. Och eftersom, med de antaganden som gjorts, mo-  
mentfördelningen inom elementet varierar linjärt, in-

ses att utböjningskurvan är ett tredjegradspolynom (två integrationer av elastiska linjens ekvation med linjär momentfördelning). Den gjorda ansatsen överensstämmer alltså exakt med den sanna utböjningsfördelningen, varför resultaten överensstämmer. Detta gäller dock normalt inte för andra element.

### 5.6.3 Analogier mellan olika problem

Som tidigare nämnts är FE-metodiken ett generellt tillvägagångssätt för lösning av fysikaliska problem, formulerade med hjälp av endera en differentialekvation eller ett funktions samband (till exempel något energiuttryck). Detta innebär att flera olika typer av problem beräkningsmässigt sammanfaller, då de har ett gemensamt styrande matematiskt samband. Betrakta till exempel det enkla fjädersystemet i FIG 5.9. Detta kan lätt analyseras genom betraktande av jämvikten i kopplingspunkterna. Definiera först systemets frihetsgrader som förskjutningarna av kopplingspunkterna. Om fjäderkonstanterna för de resp fjädrarna antas vara  $k_{01}$ ,  $k_{12}$  resp  $k_{23}$  kan fjäderkrafterna i FIG 5.9b skrivas som

$$\begin{aligned} P_{0h} &= k_{01} (u_0 - u_1) & P_{1v} &= k_{01} (u_1 - u_0) \\ P_{1h} &= k_{12} (u_1 - u_2) & P_{2v} &= k_{12} (u_2 - u_1) \\ P_{2h} &= k_{23} (u_2 - u_3) & P_{3v} &= k_{23} (u_3 - u_2) \end{aligned}$$

Jämvikten i systemets frihetsgrader ger sambanden mellan förskjutningar  $u$  och krafter  $R$  som

$$\begin{aligned} R_0 &= P_{0h} &= k_{01} u_0 - k_{01} u_1 \\ R_1 &= P_{1v} + P_{1h} &= -k_{01} u_0 + (k_{01} + k_{12}) u_1 - k_{12} u_2 \\ R_2 &= P_{2v} + P_{2h} &= -k_{12} u_1 + (k_{12} + k_{23}) u_2 - k_{23} u_3 \\ R_3 &= P_{3v} &= -k_{23} u_2 + k_{23} u_3 \end{aligned}$$

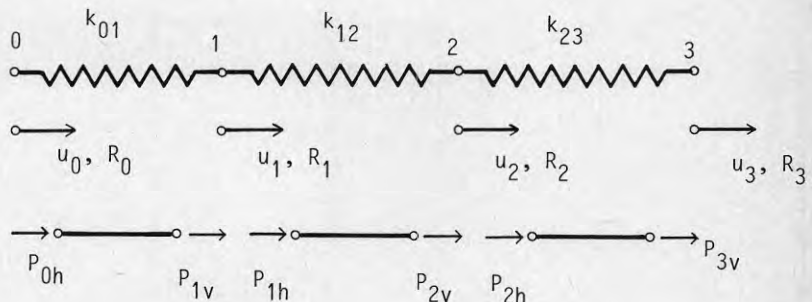


FIG 5.9 Fjädersystem för demonstration av analogier

vilket är ett styvhetssamband för fjädersystemet, analogt med de tidigare härledda styvhetsmatriserna. I detta fall var förskjutningarna i kopplingspunkterna de obekanta och styvhetsmatrisen ges genom en "addition" av fjäderstyvheter.

Betrakta nu som en jämförelse väggtvärsnittet i FIG 5.10. Antag att temperaturerna i skiktgränserna är  $T_0$ ,  $T_1$ ,  $T_2$  och  $T_3$ . Om dessa temperaturer skiljer sig kommer ett temperaturflöde att ske genom väggen. Till exempel kommer från skiktgräns 0 till skiktgräns 1 att flöda, räknat positivt från 0 till 1,

$$q_{01} = k_{01} ( T_0 - T_1 )$$

och, analogt

$$q_{12} = k_{12} ( T_1 - T_2 )$$

$$q_{23} = k_{23} ( T_2 - T_3 )$$

Om nu stationära förhållanden uppnåtts måste flödet in till punkt 1 vara lika med flödet ut från punkt 1, vilket kan uttryckas som

$$q_{12} - q_{01} = 0 .$$

Om motsvarande formuleras för de övriga punkterna fås ett ekvationssystem, där  $Q_0$  och  $Q_3$  är flödena av värmeenergi in till punkt 0 resp ut från från punkt 3.

$$k_{01} T_0 - k_{01} T_1 = Q_0$$

$$-k_{01} T_0 + ( k_{01} + k_{12} ) T_1 - k_{12} T_2 = 0$$

$$-k_{12} T_1 + ( k_{12} + k_{23} ) T_2 - k_{23} T_3 = 0$$

$$-k_{23} T_2 + k_{23} T_3 = Q_3$$

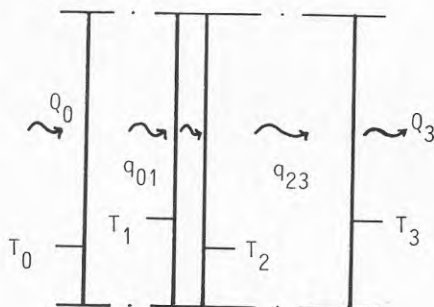


FIG 5.10 Temperaturfördelning i vägg

Detta ekvationssystem är detsamma som gällde för fjäderssystemet, sånär som på de använda beteckningarna. Man ser hur fjäderstyvheterna utbyts mot värmeomgångskoefficienter, förskjutningarna mot temperaturer och krafterna mot värmeflöden. Man kan dock se att de vanligaste randvillkoren skiljer sig. I fjäderfallet är väl det vanligaste att en frihetsgrad hålls fast och en annan belastas, varefter man söker de inre knutpunkternas rörelser. I temperaturfallet däremot är snarare de två ytornas temperaturer givna, varur man önskar beräkna värmeflödet, och eventuellt temperaturerna i de inre skiktgränserna.

Att man får exakt samma uttryck är en följd av att de båda betraktade fysikaliska fenomenen styrs av samma differentialekvation. Det finns dessutom ytterligare fall som ger upphov till samma beräkningar, till exempel grundvattenströmning, elkretsar eller laminär vattenströmning. Alla dessa fenomen kan alltså beräknas med hjälp av samma teorier och med samma program.

De två exemplen som visats var lätta att härleda med hjälp av vanliga samband. De skulle dock även ha kunnat tas fram med hjälp av en generell FE-metodik. Vad man gör, matematiskt sett är att dela in längsaxeln i problemet i ett antal delar, element. Inom dessa approximerar man det sökta värdet, som i de demonstrerade fallen är förskjutningen resp temperaturen, med hjälp av approximativa funktioner. I de aktuella fallen är dessa funktioner av utseendet enligt FIG 5.11, det vill säga funktioner som är styckvis linjära. De har dessutom egenskapen att de är noll utanför ett begränsat område, och att de är ett i en punkt inom detta område. Vi kan definiera lika många sådana funktioner som vi har noder, d v s elementgränser. Vad beräkningen sedan går ut på är att till var och en av dessa funktioner finna multiplikatorer som gör att summan av de definierade "bas"-funktionerna (multiplikerade med sina resp konstanter) så nära som möjligt ansluter till den exakta lösningen till den gällande differentialekvationen.

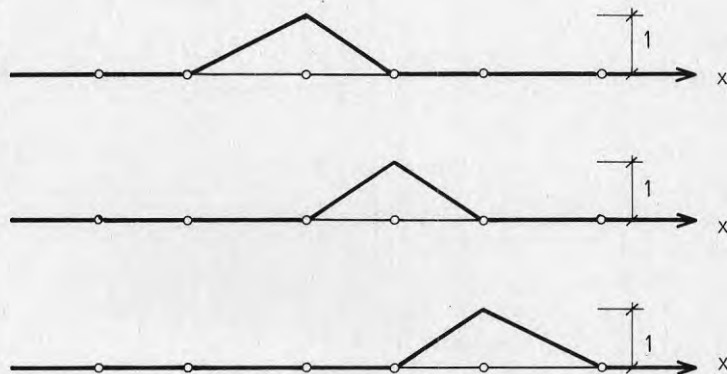


FIG 5.11 Enkla basfunktioner vid FE-analys



Man inser också ur de definierade basfunktionernas utseende att de multiplikatorer som beräknas kommer att motsvara de sökta värdena i noderna. Detta är emellertid endast en följd av de valda funktionernas egenskaper, och inget generellt krav för metoden.

Vidare inses att, eftersom var och en av de definierade basfunktionerna är linjär inom varje element, den beräknade approximerande lösningen kommer att vara linjär inom elementen, och alltså kommer att vara en linjär interpolation mellan de beräknade nodvärdena.

I de behandlade fallen är detta sant, lösningen skall variera rätlinjigt mellan nodvärdena. Detta beror på att differentialekvationen är homogen. Betrakta till exempel ett fjäderelement. Detta motsvarar ju även en rak stång. Med en konstant axialkraft  $F$  verkande i stången får man ju en töjning i längsled

$$\epsilon_x = \frac{du}{dx} = \frac{F}{EA}$$

vilket innebär att derivatan av  $u$  är konstant. Jämför nu FIG 5.12 där en stång är utsatt för en fördelad last  $g$ , exempelvis av egentyngd. Jämfört med ekvationen ovan som ger

$$\frac{d^2u}{dx^2} = 0$$

får man i stället, eftersom axialkraften förändras

$$\frac{d^2u}{dx^2} = -\frac{g}{AE}$$

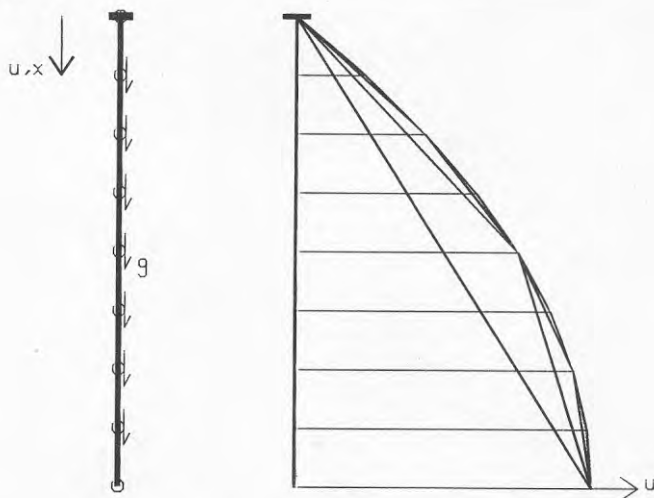


FIG 5.12 Lösning av icke-homogen ekvation (Förskjutningar i stång utsatt för en jämnt fördelad dragkraft)

som alltså ger en krökt lösningskurva. Även denna kan dock approximativt lösas med hjälp av basfunktionerna i FIG 5.11. I FIG 5.12 visas de lösningar man får genom att dela in stångens längd i 1, 2, 4 resp 8 delar och, som jämförelse den exakta lösningen. Man ser i detta fall hur lösningen allt bättre överensstämmer med den korrekta, men att lösningen fortfarande innehåller ett relativt stort fel mellan noderna.

## 5.7 Beräkningstekniska aspekter på FEM

I de tidigare avsnitten av detta kapitel har vi försökt beskriva FE-metodiken på ett sätt som skall vara begripligt för den som har sina främsta erfarenheter från konstruktörsarbete och i mindre omfattning från programmering. Teorierna har då grundats på jämvikts- och kontinuitetssamband, i stället för på minimering av funktionaler och liknande, som är det mera matematiskt inriktade tillvägagångssättet.

Innan vi, i nästa kapitel, skall börja betrakta praktiska tillämpningar av teorierna måste dock diskussionen utökas till att omfatta några mer matematiskt inriktade begrepp som är förknippade med FE beräkningar.

Det första delområde vi vill ta upp är det ekvationssystem som genereras. Som ju redan framgått bygger ju hela metodiken på att etablera ett styvhetssamband mellan vissa, för problemet väsentliga, variabler - i det statiska fallet normalt krafter och förskjutningar. Efter införandet av problemets randvillkor fick vi ett ekvationssystem som skall lösas. Detta lösande av ett linjärt ekvationssystem är så centralt i alla FE beräkningar att den föranleder en speciell diskussion.

Eftersom, för stora strukturer med många element, detta lösande står för den helt dominerande delen av den datortid som åtgår, är det också väsentligt ur effektivitetssynpunkt (och därigenom också för kostnaderna) att ekvationssystem kan hanteras på ett bra sätt.

En liten, allmän diskussion om linjära ekvationssystem finns i Appendix. Vi skall här främst inrikta oss på några speciella egenskaper hos de system som skapas, vilka är betydelsefulla ur effektivitetssynpunkt.

För det första är, vilket exempelvis framgår av de lösta exemplen i kapitel 4 och 5, en styvhetsmatris för en struktur alltid symmetrisk (åtminstone i ett linjärt fall). Detta, vilket kan visas vara en följd av Maxwells sats, innebär att för varje värde på  $i$  och  $j$  det  $i$  strukturstyvhetsmatrisen ( $K$ ) gäller att komponent  $(i,j)$  är lika med komponent  $(j,i)$ .

För det andra kan matrisen fås bandad, vilket innebär att alla komponenter med större avstånd till huvuddiagonalen än ett visst värde -  $p$  - är noll. Som framgår

av FIG 5.8 så kopplas ju till exempel nod 87 endast ihop med noderna 76, 77, 86, 88, 97 och 98, vilket innebär (jämför härledningarna) att endast 14 tal kan finnas i vardera av raderna 173 och 174 (hörande till nod 87) i styvhetsmatrisen. Det första av dessa befinner sig uppenbarligen i kolumn 151 (nod 76 x-riktning) och det sista i kolumn 196 (nod 98 y). För denna rad har vi alltså en bredd av  $196-151+1$ , d v s 46 komponenter. Med definitionen ovan sägs dock bandbredden vara  $p=174-151=23$ . Om detta gäller för alla rader i strukturstyvhetsmatrisen sägs  $p$  vara systemmatrisens ( $K$ ) bandbredd.

Uppenbarligen är värdet på bandbredden  $p$  beroende på efter vilket system vi numrerat strukturens noder. Man inser ur FIG 5.8 att om numreringen i stället skett i vertikal led skulle bandbredden ha blivit större. Grundregeln för nodnumrering är alltså att numrera i strukturens kortare riktning, där kortare skall tolkas som riktningen med det minsta antalet noder.

Att det är fördelaktigt att få bandbredden så liten som möjligt inses av att man, om man är ute efter att spara plats, kan låta bli att i datorns minne lagra alla de nollor som man vet finns utanför bandet. I stället för att, för fallet i FIG 5.8, lagra hela strukturstyvhetsmatrisen med sina  $224*224$  komponenter räcker det att skapa utrymme för  $224*(23+1)$  om man samtidigt utnyttjar kunskapen att systemmatrisen är symmetrisk. Besparingen i lagringsutrymme är i detta fall 89.3%, vilket i många fall kan vara skillnaden som gör ett problem åtkomligt för analys. Även beräkningstiden som åtgår sjunker, trots att metodiken för lösningen blir något mera komplicerad. Grovt kan anges att i det aktuella fallet beräkningstiden minskar med cirka 96%.

Även begreppen "bandbreddsoptimering", "skyline-lagring" och vågfrontslösning har att göra med metoder att minska erforderlig datorkapacitet.

Den etablerade strukturstyvhetsmatrisen har ytterligare en egenskap, som är värd att känna till, även om den inte direkt ger möjlighet till några besparingar i fråga om lagringsutrymme eller beräkningstid. Genom att betrakta uttrycken för den inre upplagrade energin i strukturen för olika förskjutningstillstånd kan man nämligen visa att styvhetsmatrisen är positivt definit. Praktiskt sett innebär detta att det ekvationsystem som ställs upp är välkonditionerat, det vill säga att man har gott hopp om att lösningen till systemet inte skall behöva innehålla några större fel. Detta är dock en sanning med viss modifikation, då vissa typer av problem ändå kan medföra bristande noggrannhet i lösningen.

Egenskapen kan dessutom ha ett visst intresse för den som skriver egna program, eftersom man i subrutinbiblioteken vid datorcentralerna ofta kan finna rutiner som är speciellt avpassade för denna typ av system.

Att matrisen är positivt definit innebär matematiskt att samtliga egenvärden till matrisen är större än noll, vilket i sin tur innebär att problemet är någorlunda snällt. Hur snällt det är, och ett mått på hur bra den beräknade lösningen kan bli kan bestämmas med hjälp av ett så kallat konditionstal för systemmatrisen. Detta, vilket kan definieras som kvoten mellan matrisens största resp dess minsta egenvärden ger ett mått på den förväntade precisionen i den beräknade lösningen. Om kvoten är mindre än, säg, 1000 kan man hoppas på en bra lösning på problemet, om den blir större har man anledning vara litet mer tveksam.

Det är dock endast i speciella fall man har anledning att betrakta lösningsnoggrannheten i en FE beräkning. Detta behandlas något i kapitel 10.

En annan mycket viktig del av en FE beräkning, även om den inte framgår lika tydligt ur de hittills gjorda härledningarna består i numerisk integration. Som skisserades i avsnittet om skivberäkningar så bildas ju styvhetssambanden för ett element, vid användning av en riktig FE metodik, genom att den inre, upplagrade energin beräknades för vissa enkla deformationstillstånd, kännetecknade av vissa funktionsansatser. En central del i detta etablerande av elementstyvhetsmatrisen är en integration av vissa derivator av de ansatta funktionerna över elementets volym. Dessa kan normalt inte beräknas med hjälp av analytiska metoder, utan man måste tillgripa en så kallad numerisk integration. En sådan metod bygger på att den funktion som skall integreras utvärderas i vissa bestämda punkter. De beräknade funktionsvärdena multipliceras med en för punkten bestämd vikt och de framkomna talen adderas till en summa, som med viss noggrannhet representerar den avsedda integralen.

Den allmänna metodiken för numerisk integration kan visas med ett par enkla metoder, där den integral som skall bestämmas är integralen av funktionen  $F(x)$  över intervallet  $(-h \leq x \leq h)$ .

$$\int F(x) dx = h ( F(-h) + F(h) )$$

$$\int F(x) dx = \frac{h}{3} ( F(-h) + 4F(0) + F(h) )$$

vilka ger det exakta svaret för polynom av låga gradtal, och som approximerar högre ordningens funktioner relativt väl. Högre noggrannhet i integralberäkningarna kan man dock få genom användning av så kallad Gaussintegration (Gauss-kvadratur). I dessa metoder används i stället optimalt valda punkter, vilket innebär att de använda koordinaterna får "ojämna" värden, vilket dock inte spelar någon roll för datorn. De vanligast använda metoderna för Gauss-integration är 2- resp 3-punkts. Formlerna för dessa ges av

$$\int F(x)dx = h \left( F\left(-\frac{h}{\sqrt{3}}\right) + F\left(+\frac{h}{\sqrt{3}}\right) \right)$$

$$\int F(x)dx = \frac{h}{9} (5F(-0.77460\dots) + 8F(0) + 5F(+0.77460\dots))$$

Även högre ordningens integrationsapproximationer kan härledas. Fördelen med detta är att noggrannare resultat kan erhållas, men till kostnaden av att flera funktionsvärden måste utvärderas. Det finns också en viss gräns för varje polynom där det exakta svaret uppnås, och medtagande av flera punkter alltså inte ger bättre resultat. För de vanligaste elementtyperna för balk-, skiv- eller plattanalys är denna gräns 2, 3 eller i vissa fall 4.

För flerdimensionella områden, exempelvis rektanglar kan motsvarande ideer användas. För rektangeln

$$(-h \leq x \leq h, -k \leq y \leq k)$$

kan exempelvis integralen av funktionen  $F(x,y)$  beräknas med hjälp av approximationen (2\*2 Gauss-integration)

$$\int F(x,y)dx dy = hk(F(-g,-g) + F(-g,g) + F(g,-g) + F(g,g))$$

där  $g = \frac{1}{\sqrt{3}}$

och på motsvarande sätt för 3\*3 integration, vilket kan åskådliggöras med FIG 5.13 där de medtagna punkterna och deras respektive vikter visas.

Grundprincipen för val av integrationsordning är alltså att välja så många punkter som möjligt. Detta störs emellertid av vissa problem som kan inträffa i vissa speciella fall. Till exempel så kommer i avsnitt 6.3 att visas att problem uppstår vid användning av allt-

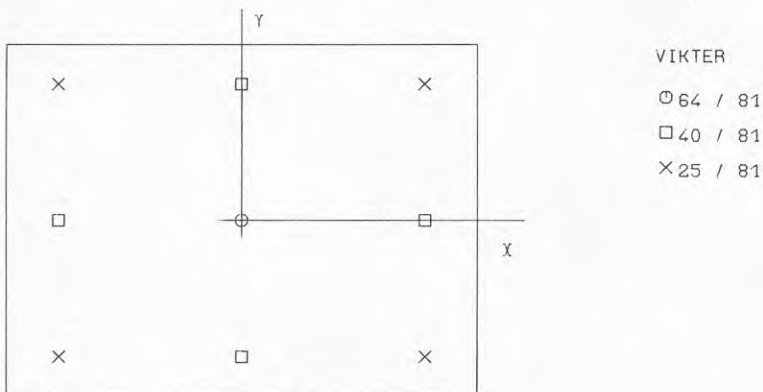


FIG 5.13 Integrationspunkter för 3\*3 Gaussintegration



för noggrann integration i samband med plattelement baserade på den så kallade Mindlin's platteori. I dessa fall krävs nämligen en viss brist på noggrannhet i integrationen för att lösningen skall bli optimalt riktig.

En annan aspekt på FE beräkningar bör också diskuteras i detta allmänna sammanhang, nämligen möjligheterna att blanda olika elementtyper. Rent principiellt borde det, om man ser styvhetsrelationen som enbart en formulering av jämviktssambanden, vara möjligt att helt fritt blanda olika elementtyper om de olika typerna bara beskrivs med hjälp av samma nodfrihetsgrader. Detta skulle innebära att olika typer av balk- och skivelement skulle vara kombinerbara.

I verkligheten ställer dock ofta denna typ av blandningar till problem. Problemen är en följd av de olika förskjutningsansatser som görs vid härledningen av elementstyvhetsrelationerna. Betrakta till exempel balk- och skivelementet i FIG 5.14a. Visserligen skulle nodjämvikten för de två ingående noderna kunna formuleras, genom betraktande av de två elementens lokala styvhetsmatriser, men vi har då infört ett fel i kontinuiteten mellan förskjutningarna. Med de två förskjutnings- och de två rotationsparametrarna beskrivs nämligen balkelementets förskjutningar som ett tredjegradspolynom, medan skivelementet - som endast utnyttjar förskjutningsfrihetsgraden - beskrivs som en rät linje mellan nodvärdena. Vi har alltså ett "glapp" mellan de två elementen i alla punkter utom nodpunkterna. Om vi nu tänkt oss att balken och skivan skulle samverka fullständigt motsvarar detta glapp ett fel i våra antaganden, som kan ha ganska stor betydelse för lösningens noggrannhet.

Samma fenomen uppträder vid försök att kombinera olika typer av skivelement, till exempel 4- och 8-nods elementen i FIG 5.14b. I detta fall beskrivs 8-nodselementets förskjutningar i gränslinjen med hjälp av en andrags parabel, medan 4-nodselementen har en linjär förskjutningsansats. Följden blir ett visst glapp i området mellan nodpunkterna som leder till felaktigheter i lösningen.

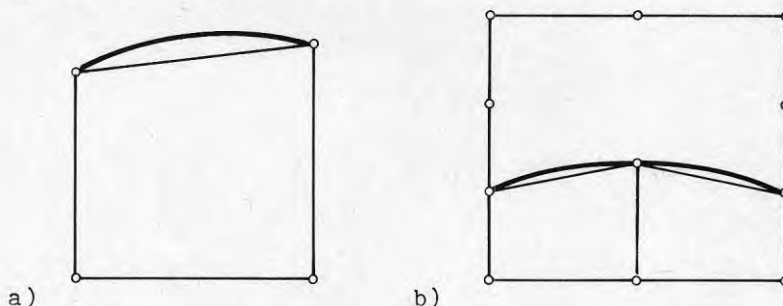


FIG 5.14 Glapp vid kombination av olika element

En kombination som ofta används för att på ett någorlunda enkelt sätt representera olika villkor är att man lägger in fiktiva fjädrar mellan olika noder. Genom att ge vissa egenskaper åt dessa fjädrar kan olika randvillkor, och olika icke-linjariteter ofta införas.

Ett program som ger möjlighet till användning av flera olika elementtyper bör kontrollera de elementkombinationer som används, och åtminstone ge en varning om inkompatibla typer definierats.

### 5.8 Icke-linjära beräkningar

I många fall har man anledning att ta hänsyn till olika icke-linjariteter i en strukturs beteende. Dessa kan vara av två olika slag, dels geometriska icke-linjariteter exempelvis hänsyn till stora axialkrafter i balkelement - jämför avsnitt 6.1.6 - dels icke-linjariteter i materialbeteendet, till exempel i form av uppsprickning eller plasticeringar - jämför avsnitt 6.1.5 .

Båda dessa typer av effekter kan behandlas med hjälp av Finita Element beräkningar. Den effekt som icke-linjariteten har på de använda uttrycken är ofta en förändring av styvhetsuttrycken. I stället för att, i de linjära fallet, sambandet mellan krafter och förskjutningar kan skrivas

$$\bar{R} = (K) \bar{r}$$

kommer nu styvhetsmatrisen (eller eventuellt lasterna) att vara beroende av de uppnådda förskjutningarna i princip enligt

$$\bar{R} = (K(\bar{r})) \bar{r}$$

Detta innebär att styvhetsmatrisen (K) inte kan bildas utan att lösningen är känd. För att lösa denna typ av problem krävs därför någon form av iterativt förfarande. Detta innebär att man från någon ursprunglig gissning  $\bar{r}_0$  bildar en styvhetsmatris ( $K(\bar{r}_0)$ ). Med hjälp av denna kan en ny lösning  $r_1$  beräknas. Förfarandet upprepas till dess acceptabel noggrannhet uppnåtts.

Noggrannheten i en viss, beräknad lösning, kan bedömas med hjälp av någon norm för residualkrafterna, det vill säga skillnaden mellan högerledets och vänsterledets kraftvektorer i styvhets sambandet ovan.

Generellt måste man också säga att en mycket god kunskap om det aktuella problemet erfordras för att man skall kunna få ut användbara resultat ur en icke-linjär FE beräkning.

## 6 PROGRAM FÖR ANALYS

I detta kapitel och det följande, skall vi behandla en del praktiska aspekter på beräkningar med de teorier som diskuterats i kapitlen 4 och 5. Avsikten med detta är dels att visa vilka metoder som är praktiskt gångbara för olika typer av problem, men även att kortfattat söka förklara en mängd termer och beteckningar som ofta används i programbeskrivningar o dyl.

Avsikten med denna genomgång är inte att leda den nye datoranvändaren in i komplicerade beräkningar, utan i första hand att söka klargöra vad olika teorier, beräkningsmetoder och dylikt innebär. Vår egen erfarenhet av till exempel dynamiska och icke-linjära beräkningar säger oss att dessa lämpligen bör behandlas av specialister. Det är en illusion att tro att man kan lösa mer komplicerade problem med datorn än utan den, även om program som tar hänsyn till ett stort antal olika effekter finns allmänt tillgängliga. Att man trots detta inte skall tro på någon större framgång om man försöker sig på några mera speciella tillämpningar, beror på att komplicerade datorprogram ofta är känsliga och svårhanterliga samt ofta innehåller svårgenomsådliga begränsningar och förutsättningar.

I genomgången skall vi skilja på program för analys och för dimensionering. Med termen analys avser vi ett program som, utgående från vissa geometriska data samt laster och randvillkor kan beräkna vissa snittkrafter i olika punkter i strukturen. För att programmet skall betraktas som ett dimensioneringsprogram krävs dessutom att de beräknade resultaten på något sätt jämförs med vissa uppställda krav, på exempelvis maximala accepterade spännings- eller deformationsvärden. Dimensioneringsprogrammen är alltså kopplade till vissa, explicita eller implicita, krav, till exempel normer, medan ett analysprogram endast redovisar uppnådda värden, utan hänsyn till om, till exempel, de beräknade spänningarna är 10000 eller 0.00001.

I detta kapitel kommer endast program för analys att behandlas, medan dimensioneringsprogram diskuteras i kapitel 7. Vi skall i detta kapitel behandla de vanligaste strukturtyperna i olika avsnitt.

### 6.1 Balkstrukturer

I kapitlen 4 och 5 har vi diskuterat ett par grundläggande beräkningsmetoder för balkstrukturer, det vill säga kontinuerliga balkar och ramar. Det visades att för fallet med kontinuerliga balkar såväl kraftmetodbaserade som förskjutningsmetodbaserade beräkningar

kunde väljas, medan man för ramar hade de största möjligheterna att lyckas med en förskjutningsbaserad metodik. En slutsats drogs därför att förskjutningsmetoden var mer lämplig vid skapande av generell användbara program, på grund av att kraftmetoden kräver en större kunskap om det aktuella problemets upplagsförhållanden och dylikt.

För kontinuerliga balkar är dock denna generella sanning inte helt tillämplig. Skall man nämligen skriva ett program för enbart analys av kontinuerliga balkar är det snarare lämpligast att tillämpa någon av de två metoderna som diskuterades i avsnitten 4.5.1 och 4.5.2.

Ofta vill man dock inte begränsa sig så utan göra ett program som är användbart för alla typer av balkstrukturer, och där en kontinuerlig balk betraktas som ett specialfall. I detta fall är förskjutningsmetodiken, till exempel i form av en Finit Element analys, att föredra.

#### 6.1.1 Kraftmetoder

Två alternativa, kraftmetodbaserade beräkningsmetoder för kontinuerliga balkar har presenterats i avsnitt 4.5. I detta fall är denna metodik att föredra framför en förskjutningsbaserad beräkning. Denna rekommendation är speciellt giltig vid komplicerade last eller upplagsförhållanden.

Kraftmetoden har också klara fördelar då det gäller att behandla speciella balkutformningar, till exempel vid vutförsedda balkar och balkar på elastiska stöd. För detta krävs dock en numerisk integration av utböjningarna och vinkeländringarna i balkdelarna för att etablera den aktuella flexibilitetsmatrisen, vilket dock i och för sig är en lätt uppgift för datorn.

Tack vare att antalet statistiskt obestämda kvantiteter ofta är mindre än det antal frihetsgrader som måste medtas blir också beräkningsarbetet vid en kraftmetodbaserad beräkningsgång mindre, vilket i sin tur medför en lägre kostnad.

I många fall kan också kraftmetoden vara ett intressant alternativ vid mera komplicerade balkstrukturer. Detta gäller till exempel då man relativt enkelt kan utföra en del av analysen manuellt, och sedan använda ett standardprogram för hantering av de matriser som bildas.

### 6.1.2 Förskjutningsmetoder

För handräkning har ett par förskjutningsbaserade beräkningsexempel visats i kapitel 4. Liksom för kraftmetoden är det svårt att skriva ett generellt program för analys av balkstrukturer baserad på denna rena förskjutningsmetodik. Problemen är de samma, nämligen krav på en noggrann kännedom om den aktuella strukturens upplagsförhållanden och dylikt. För att kunna skapa ett mer generellt program för allmänna ramar bör snarare en elementbaserad teknik enligt kapitel 5 tillämpas.

### 6.1.3 Finita Element metoder

En elementbaserad teknik för beräkning av balkstrukturer har redan behandlats i kapitel 5. Den formulering som togs fram överensstämmer helt med den som fås med en direkt Finit Element härledning för det använda balkelementet. Detta element, med sex frihetsgrader enligt FIG 5.2, används också i de flesta tillgängliga FE program. Det finns dock andra möjligheter till styvhetsformuleringar. Till exempel kan element skapas som har tre stycken noder. Härledningen för ett sådant element kan inte ske med den enkla metoden från kapitel 5, utan man måste i stället tillgripa den metodik som skisseras i avsnitt 6.1.4.

En alternativ formulering av balkstyvheten förekommer i vissa fall, då hänsyn tas till skjuvdeformationer i balkelementet. Man brukar i så fall tala om en balkteori enligt Timoshenko, i stället för den vanliga "Euler-Bernoulli"-teorin.

I denna formulering kan samma elementutseende som ovan användas. Man kompletterar bara härledningen med ett hänsynstagande till de uppstående tvärkraftsdeformationerna i elementet. En beskrivning av denna elementtyp finns till exempel i Lorentsen, se avsnitt 12.4.

För denna tillämpning kan också ett principiellt annat element användas än det som visades i FIG 5.2. Även om detta element, utvecklat av Hughes och beskrivet i bland annat Owen & Hinton (jfr avsnitt 12.5), ser likadant ut har nodparametern  $\theta$  en annan innebörd. I FIG 5.2 var denna lika med lutningen av balkens medellinje i förhållande till elementets ursprungliga läge. I Hughes formulering är  $\theta$  i stället lutningen av balkens tvärsnittsplan, jämför FIG 6.1. Som framgår av figuren kommer denna vinkel att skilja sig från medellinjens lutning med en kvantitet beroende på skjuvspänningen i punkten. Förutsättningen för teorin är att tvärsnitt, ursprungligen plana och vinkelräta mot medellinjen, förblir plana, men inte nödvändigtvis vinkelräta mot den deformerade medellinjen.



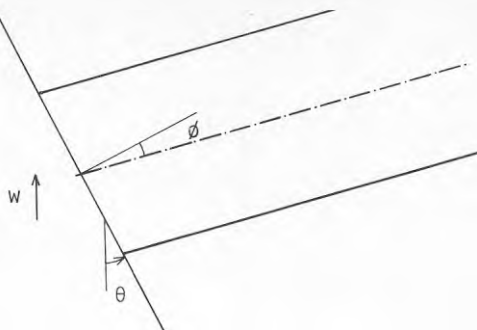


FIG 6.1 Alternativ definition av vinkeln  $\theta$  i samband med Timoshenko-teori

Vi betecknar i fortsättningen detta element som Timoshenko-elementet.

Vinkeln  $\phi$  i FIG 6.1 svarar mot en skjuvdeformation. Det inre arbetet av denna kan tas med i uttrycken då styvhetsrelationen för elementet härleds, varvid en annan styvhetsmatris erhålls.

I FE-formuleringen beskrivs såväl utböjningen  $w$  som tvärsnittslutningen  $\theta$  genom en linjärinterpolation mellan nodvärdena. Det bör då observeras att detta innebär en klart sämre representation av utböjningen än den som erhöles med elementet i kapitel 5, då ett tredjegradspolynom för utböjningen definierades av nodparametrarna. En motsvarande försämring av momentrepresentationen i elementet fås också. I stället för den linjära momentfördelning som gällde för det vanliga balk-elementet, fås nu i stället ett element där momentet är konstant över hela elementets längd.

I FIG 6.2 jämförs den beräknade momentfördelningen för en enkel kontinuerlig balk, som analyserats med de två metoderna. Det framgår att man, trots flera element får en sämre representation av momentfördelningen då Timoshenkoelementet används. Notera även att man, då metoden i avsnitt 6.1.4 använts för representation av den utbreddalasten, får exakt rätta momentvärden i noderna då det vanliga elementet används.

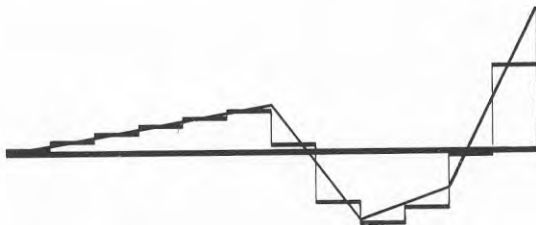


FIG 6.2 Jämförelse av momentvärden för olika element

Endast mycket speciella fall motiverar alltså ett användande av Timoshenko-balkelementet. Den enda gången då skjuvdeformationerna kan ha någon betydelse i jämförelse med böjdeformationerna är då balkarna är mycket skjuvveka till exempel då en ramverksbalk behandlas. Även i vissa fall då man behandlar fiktiva balkelement, till exempel representerande någon samverkanstruktur kan skjuvningen bli intressant.

Då programmet kan ta hänsyn till skjuvdeformationer i balkelementen måste en skjuvstyvhet för balktvärsnittet definieras. Denna kan ofta skrivas som

$$(GA)^* = k G A$$

där  $G$  och  $A$  är balkens skjuvmodul respektive tvärsnittsarea. Faktorn  $k$  är en konstant som är beroende av tvärsnittets utformning, och för vanliga balktyper har ett värde av cirka 0.7, vilket också vissa program använder som inbyggd förutsättning. I de fall då tvärkraftsdeformationen över huvud taget är intressant, det vill säga för fiktiva, skjuvveka balkar kan dock skjuvstyvheten stå i en helt annan relation till tvärsnittsarean.

#### 6.1.4 Utbredda belastningar

I kapitel 5 har en mindre diskussion ägnats åt problemen att påföra fördelade laster på balkelementen vid en FE analys. Egentligen kan ju nämligen dessa endast belastas med punktkrafter och punktmoment i strukturens nodpunkter. I avsnitt 5.4 löstes detta med hjälp av att den verkliga balken delades in i flera element, där kopplingspunkterna belastades med punktlaster svarande mot de olika elementens andelar av den utbredda lasten, ett förfarande som kunde visas ge relativt goda resultat.

Denna metod leder dock till att ett stort antal element erfordras, vilket kan vara besvärande. För att på ett bättre sätt representera de utbredda lasterna krävs emellertid ett annat tillvägagångssätt.

För att härleda detta måste man dock angripa formuleringen av balkelementet på ett mera teoretiskt sätt. Med denna avsikt skall utböjningen av balkelementet beskrivas på ett sätt, analogt med det som skisserades för skivelementet i avsnitt 5.6.2. Om, för detta ändamål, balkelementet antas beläget längs  $x$ -axeln, kan utböjningen av balken beskrivas som

$$w = A_0 + A_1 x + A_2 x^2 + A_3 x^3$$

Om vidare en lokal koordinat  $\xi$  införes, som

$$\xi = \frac{(x-x_c)}{(L/2)}$$

där L är avståndet mellan de två nodpunkterna för elementet, och  $x_c$  mittpunkten mellan dem kan uttrycket för utböjningen skrivas

$$\begin{aligned} w &= w_1 \frac{1}{4} (2 - 3\xi + \xi^3) + \\ &+ w_2 \frac{1}{4} (2 + 3\xi - \xi^3) + \\ &+ \theta_1 \frac{L}{8} (1 - \xi - \xi^2 + \xi^3) + \\ &+ \theta_2 \frac{L}{8} (-1 - \xi + \xi^2 + \xi^3) . \\ &= \bar{N}^T \bar{u} = \bar{u}^T \bar{N} \end{aligned}$$

med beteckningar enligt FIG 6.3.

Genom derivering två gånger av detta uttryck kan ett uttryck för krökningen fås, vilket i matrisform uttrycks som

$$K = \frac{d^2 w}{dx^2} = \frac{4}{L^2} \begin{bmatrix} \frac{3}{2} \xi \\ -2 + 6\xi \\ -\frac{3}{2} \xi \\ +2 + 6\xi \end{bmatrix}^T \begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{bmatrix}$$

eller

$$K = (B) \bar{u}$$

Det som nu skall användas är en formulering av det inre arbetet av krökningar och moment. Eftersom det gäller att momentet M i en punkt är integralen över tvärsnittsarean

$$M = \int E z^2 K dz$$

fås för det inre arbetet  $W_i$  (E=elasticitetsmodulen, z=avståndet från medellinjen)

$$W_i = \frac{1}{2} \int (K E z^2 K) dvol$$

Om matrissambandet ovan utnyttjas kan detta skrivas

$$W_i = \frac{1}{2} \int \bar{u} ( (B)^T E z^2 (B) ) \bar{u} \, dvol$$

Eftersom vektorerna  $\bar{u}$ , vid denna integration, skall betraktas som konstanter, får man ett uttryck för det inre arbetet

$$W_i = \frac{1}{2} \bar{u} (k) \bar{u}$$

där elementstyvhetsmatrisen  $(k)$  kan bestämmas genom integrering av de definierade funktionerna. Den på detta sätt erhållna elementstyvheten överensstämmer helt med den matris som härleddes i avsnitt 5.1.

Det som nu genomförts är en Finit Element inriktad metod för härledning av elementstyvhetsmatrisen för balk-elementet, till skillnad från den statiska metoden i kapitel 5. Om den nya metodiken skall fullföljas skall nu användas som villkor att det inre arbetet skall vara lika med det yttre arbetet av pålagda laster. Eftersom man, då endast koncentrerade laster i de fyra definierade frihetsgraderna beaktas, kan skriva det yttre arbetet som, jämför FIG 6.3,

$$W_y = \frac{1}{2} ( P_1 w_1 + P_2 w_2 + M_1 \theta_1 + M_2 \theta_2 ) = \frac{1}{2} \bar{u}^T \bar{P}$$

fås jämviktsvillkoret som

$$\frac{1}{2} \bar{u}^T (k) \bar{u} = \frac{1}{2} \bar{u}^T \bar{P}$$

Detta gällde alltså för koncentrerade nodkrafter. För utbredda laster med intensiteten  $q(x)$  kan det yttre arbetet i stället formuleras som

$$W_y = \frac{1}{2} \int q(x) w \, dx$$

Om det ansatta sambandet för utböjningen  $w$  införs fås därefter

$$W_y = \frac{1}{2} \int ( q(x) \bar{N}^T \bar{u} ) \, dx = \frac{1}{2} \bar{u}^T \int ( q(x) \bar{N} ) \, dx$$

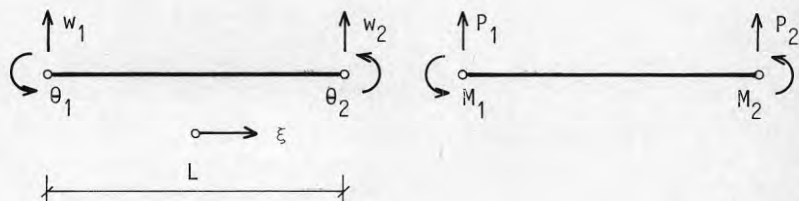


FIG 6.3 Beteckningar för härledning av balkelement

Efter jämförelse med villkoret för jämvikt inses att den utbredda lasten kommer att ersättas med "ekvivalenta nodlaster" där

$$P_1 = \int ( q(x) N_1(x) ) dx$$

och så vidare.

Om detta uttryck utvärderas för fallet med en jämnt utbredd last  $q(x)=q$  fås

$$P_1 = \int \left( \frac{q}{4} ( 2 - 3\xi + \xi^3 ) \frac{L}{2} d\xi \right) = \frac{qL}{2}$$

$$P_2 = \int \left( \frac{q}{4} ( 2 + 3\xi - \xi^3 ) \frac{L}{2} d\xi \right) = \frac{qL}{2}$$

$$M_1 = \int \left( q \frac{L}{8} ( 1 - \xi - \xi^2 + \xi^3 ) \frac{L}{2} d\xi \right) = + \frac{qL^2}{12}$$

$$M_2 = \int \left( q \frac{L}{8} ( -1 - \xi + \xi^2 + \xi^3 ) \frac{L}{2} d\xi \right) = - \frac{qL^2}{12}$$

Dessa värden, tillsammans med de ekvivalenta nodlasterna för några andra enkla lastfall, finns redovisade i FIG 6.4.

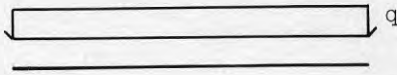
### 6.1.5 Varierande böjstyvheter

I en verklig struktur har man ofta varierande böjstyvheter för de balkar som ingår. Om de olika elementen har olika värden ställer inte detta till något problem i de härledningarna som gjorts. Om däremot böjstyvheten varierar inom elementen, till exempel på grund av att voter lagts in, kan härledningssättet i kapitel 5 försvåras. Däremot kan man, med den ansats som gjorts i detta kapitel i samband med de utbredda lasterna, lätt ta med en sådan effekt.

Eftersom i härledningen ingår en integration över balkens längd av produkten av vissa funktioner multiplicerad med böjstyvhetsvärdet kan ett varierande värde på EI relativt enkelt infogas, åtminstone om balkstyvheten integreras numeriskt, med ett någorlunda stort antal punkter. Observera dock att det, om voter skall beaktas, inte alls är tillräckligt med de 2 eller 3 punkter på längden som normalt används, utan att antalet snarare måste vara uppemot 20.

En fiktivt varierande böjstyvhet får man om man för en betongbalk beaktar den uppsprickning som sker i den dragna delen av tvärsnittet. Ett sådant hänsynstagande skulle egentligen behöva göras för att återge betongens speciella egenskaper, se vidare diskussionen i avsnitt 7.2.1. I och med ett sådant betraktelsesätt får man dock en icke-linjär beräkning eftersom det uppspruckna området är okänt före beräkningen. Följden blir att ett kostsamt, iterativt förfarande - åtmin-



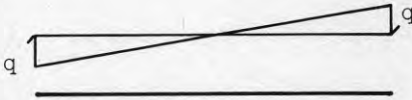


$$P_1 = -\frac{qL}{2}$$

$$P_2 = P_1$$

$$M_1 = -\frac{qL^2}{12}$$

$$M_2 = -M_1$$

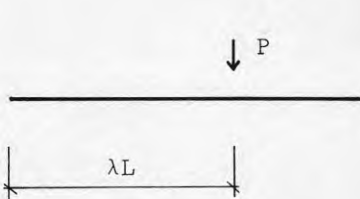


$$P_1 = \frac{qL}{5}$$

$$P_2 = -P_1$$

$$M_1 = \frac{qL^2}{60}$$

$$M_2 = M_1$$



$$P_1 = P(-1 + 3\lambda^2 - 2\lambda^3)$$

$$P_2 = P(-3\lambda^2 + 2\lambda^3)$$

$$M_1 = PL(-\lambda + 2\lambda^2 - \lambda^3)$$

$$M_2 = PL(\lambda^2 - \lambda^3)$$



$$P_1 = \frac{6M}{L}(\lambda - \lambda^2)$$

$$P_2 = \frac{6M}{L}(-\lambda + \lambda^2)$$

$$M_1 = M(-1 + 4\lambda - 3\lambda^2)$$

$$M_2 = M(2\lambda - 3\lambda^2)$$

FIG 6.4 Ekvivalenta nodlaster för balkelement

stone i princip - måste tillgripas, där en lösning ger en böjstyvhetsfördelning för en förnyad gissning.

Motsvarande kan göras för stål, och andra liknande material. I detta fall är det en successiv plasticering av balktvärsnittet som leder till icke-linjariteten i beräkningen. Om en sådan metodik används målmedvetet fås en beräkningsmetod som nära överensstämmer med en ramberäkning enligt gränslastteorin. Vissa program finns också tillgängliga för en sådan analys.

### 6.1.6 Normalkrafters inverkan

I diskussionen av balkelementen har vi hittills förutsatt att normalkrafterna och böjmomenten är oberoende av varandra, det vill säga att ingen koppling finns mellan dessa grundläggande verkningssätt hos balken. Det är dock känt att för en struktur, motståndet mot böjning avtar om stora tryckkrafter verkar i balken, ett fenomen som så småningom leder till knäckning av balken om tryckkraften ökas över en viss gräns. På motsvarande sätt gäller att en dragkraft i balken minskar dess utböjningar för transversella belastningar.

Denna effekt kan tas med i en FE-analys av en balkstruktur, om beräkningen görs efter den så kallade "andra ordningens teori". Denna teori som tar hänsyn till deflektionsmomenten i balkelementet leder till en böjstyvhetsuttryck för balkelementet som är beroende av den i elementet verkande axialkraften.

Hänsynstagandet till axialkrafterna förs in via styvhetstermerna. Om nämligen för balkelementet gäller att, med den verkande tryckkraften  $P$ ,

$$\alpha = \frac{P l^2}{\pi^2 EI}$$

ersätts böjdelen (rader och kolumner 2,3,5 och 6) av elementstyvhetsmatrisen i avsnitt 5.1 med följande

$$(k_\alpha) = \frac{EI}{l^3} \begin{pmatrix} K'_T(\alpha) & C'_T(\alpha)l & -K'_T(\alpha) & C'_T(\alpha)l \\ C'_T(\alpha)l & K'(\alpha)l^2 & -C'_T(\alpha)l & C'(\alpha)l^2 \\ -K'_T(\alpha) & -C'_T(\alpha)l & K'_T(\alpha)l^2 & -C'_T(\alpha)l \\ C'_T(\alpha)l & C'(\alpha)l^2 & -C'_T(\alpha)l & K'(\alpha)l^2 \end{pmatrix}$$

För de ingående variablerna kan exakta uttryck, lämpade för datorns arbete, härledas som funktion av variabeln  $\alpha$ , vilken alltså anger kvoten mellan den aktuella tryckkraften och elementets knäckningslast enligt Eulers andra fall. För att få en ungefärlig uppfattning om värdenas variation ges TABELL 6.1, där värdena hämtats från Edlund, Wahlström & Åkesson: "Flexibiliteter och styvheter hos prismatiska balkpelare med konstant axialkraft", Göteborg 1967.

Liksom i fallet med materialberoende icke-linjariteter i det förra avsnittet, blir det i samband med andra ordningens teori alltid frågan om en iterativ beräkning. Från en linjär, vanlig lösning fås vissa preliminära axialkrafter i elementen. Dessa används sedan för att ombilda styvhetsmatrisen, varefter en ny, och avvikande, lösning fås. På detta sätt får beräkningen fortsättas med successivt nya lösningar, som förhoppningsvis kommer allt närmare sanningen, jämför avsnitt 5.8 .

TABELL 6.1 Böjstyvhetsmatrisens variation med den aktuella tryckkraften i ett balkelement

$\alpha$	$K'_T(\alpha)$	$K'(\alpha)$	$C'_T(\alpha)$	$C'(\alpha)$
0	12.00	4.00	6.00	2.00
0.5	6.04	3.29	5.48	2.19
1.0	0	2.47	4.94	2.47
2.0	-12.40	0.14	3.67	3.52
3.0	-25.42	-5.03	2.09	7.12

Villkoret för att det iterativa förfarandet skall konvergera mot en lösning är att strukturen är stabil för den aktuella lastnivån, det vill säga att strukturens knäckningslast ej uppnåtts. Detta kan användas för att bestämma säkerhetsfaktorn mot ett stabilitetsbrott, om de verkliga lasterna på strukturen multipliceras med allt större lastfaktorer. Från och med en viss lastnivå kommer man att få en icke positivt definit strukturstyvhetsmatris, vilket alltså visar att strukturen är instabil för denna lastnivå.

Värt att notera är att man med denna beräkning får värden på hela strukturens knäckningslaster, till skillnad från i en vanlig manuell analys där man betraktar en balkdel i taget. Detta medför att man på ett korrekt sätt tar hänsyn till den elastiska inspänning som fås mellan olika ramdelar, men också att man tar hänsyn till lasternas inbördes placering och deras relativa storlekar.

Ur stabilitetsberäkningen kan man också oftast utläsa den knäckningsfigur som svarar mot de olika knäckningslastnivåerna. I detta sammanhang är en plottad bild av dessa knäckningsmoder närmast ovärderlig, då det gäller att söka bedöma utknäckningens effekter, och hur de eventuellt kan bemästras.

Generellt gäller för stabilitetsberäkningar med FE metodik att man kan få fram ett lika stort antal knäckningslastnivåer, som man har frihetsgrader i strukturen. Ofta saknar dock ett stort antal av dessa verklig fysikalisk betydelse. Om man dock begränsar sig till att betrakta ett antal av de lägsta nivåerna kan dessa antas relativt väl återge verkligheten.

En svårighet i samband med beräkningar med andra ordningens teori kan vara att använda de erhållna resultaten på rätt sätt. De nu giltiga normerna ger inget entydigt svar på denna frågeställning. Eftersom dock denna beräkningsmetodik skall ta hänsyn till axialkrafternas inverkan på böjstyvheten, och därmed innefatta deflektionsmomenten i samma beräkning, bör dock det riktiga vara att jämföra de momentvärden, och därav följande spänningar, som erhålls med normvärdena för en pelare utan knäckningsrisk, det vill säga med ett slankhetsvärde  $(l/i)=0$ .

### 6.1.7 Dynamik

En annan effekt i ramar som ofta behöver analyseras är ramens dynamiska egenskaper. Detta kan relativt enkelt göras med Finita Element och liknande metoder.

I princip kan de dynamiska beräkningar som kan vara intressanta delas in i två olika kategorier. Den första är en beräkning av balkstrukturens egenfrekvenser. Med detta vill man spåra de resonansfrekvenser som ramen har, och att söka få dessa att avvika från de frekvenser som vind och andra belastningar kan tänkas ge.

Den andra, och mer komplicerade, dynamiska beräkningen är en transient-beräkning där man följer en viss tidsberoende belastnings påverkan på strukturen och dess olika delar, under en viss tidsrymd. Denna typ av beräkningar är främst intressanta för snabba förlopp till exempel stötblastningar, och i samband med jordbävningssdimensionering, där man vill utnyttja strukturens förmåga att under kort tid bära stora belastningar.

I samband med dynamiska beräkningar blir två begrepp, förutom strukturens styvhet aktuella, nämligen strukturens massmatris och dess dämpmatris.

Mycket kortfattat kan man säga att massmatrisen beskriver fördelningen av strukturens massa - och eventuella pålagda punktmassor - på de olika frihetsgraderna. På samma sätt som utbredda laster kan massan behandlas på två sätt. Det enklaste är att koncentrera elementens massor till noderna, det mera komplicerade att fördela dem med hänsyn till basfunktionernas utseende. Man talar i de två fallen om koncentrerad resp konsistent massmatris. Den senare ger, till en ofta betydligt högre beräkningskostnad, i de flesta fall noggrannare resultat.

När det gäller dämpmatrisen är den ofta svår att skapa på grund av att kunskaperna om olika materials dämpningsegenskaper är dåliga. I praktiska tillämpningar används ofta, i brist på bättre antaganden, så kallad Rayleigh-dämpning varvid dämpmatrisen (C) skrivs som

$$(C) = a (M) + b (K)$$

det vill säga en linjärkombination av strukturens massmatris (M) och dess styvhetsmatris (K), ett antagande som har fördelar ur beräkningssynpunkt.

Ur strikt matematisk synvinkel har beräkningen av egenfrekvenser för en balkstruktur en stark koppling till knäckningsberäkningar. Detta är en följd av att det i båda fallen är egenvärden kopplade till strukturens styvhetsmatris som söks. Därför gäller i det dynamiska fallet samma begränsningar av antalet resonansfrekvenser som fås, och deras noggrannhet som i knäckningsfallet. De lägsta egenfrekvenserna blir därför även i det-

ta fall relativt noggrant bestämda, medan högre frekvenser kan missas helt, eller grovt missbedömas. Rekommendationen att låta rita upp resultatet, det vill säga de beräknade svängningsmoderna, är också fullt giltig.

### 6.1.8 Stångsystem

Mycket nära besläktade med balkstrukturerna är stångsystemen. Den enda skillnaden är strängt taget den definitionsmässiga att stängerna antas ej kunna ta upp böjning. Som diskuterats redan i kapitel 4 kan element för stånganalys lätt skapas, om stängerna förses med två frihetsgrader i vardera ändpunkten.

I många praktiska sammanhang har man dock behov av att kombinera balkar och stänger, exempelvis vid analys av hallar där en viss del platsgjuts, och alltså får momentstyva knutpunkter, medan andra delar byggs ut med hjälp av pendelpelare. I detta fall kan inte ett program för stångsystem användas eftersom vissa delar skall kunna ta moment. Man måste alltså, om man inte har tillgång till ett program som klarar båda elementtyperna, använda ett balkprogram och sedan införa som bivillkor att vissa element skall vara momentfria. I många program kan detta ganska lätt införas, genom att man, i princip, förser elementets ena eller båda ändar med leder. Matematiskt innebär detta att man i elementstyvheten "manipulerar" in ett villkor att momentet i änden skall vara noll.

Vid användningen av detta måste man dock vara försiktig, och se till att man inte inför detta för alla balkändar som ansluter till en knutpunkt. I så fall kommer nämligen rotationsfrihetsgraden  $\theta$  i denna nod att lämnas odefinierad, eftersom manipulationen innebär att elementets styvhet för rotation i den aktuella noden blir noll, och styvhetsmatrisen singular. Om momentfrihet skall införas för alla balkändar invid en knutpunkt måste man dock ändå lämna momentstyvhet för ett element (varken mer eller mindre). Om inte ett koncentrerat moment angriper i denna nod så kommer ju även det oförändrade elementet att få ändmomentet noll på grund av villkoret om jämvikt.

I fall då man behandlar fackverk kan man få problem med anslutningen av de olika balkdelarna. I verkligheten kommer ju inte systemlinjerna för till exempel understrången och två diagonaler att skäras i en punkt. Detta kan lösas på ett par olika sätt. Det enklaste är att något jämka de två diagonalerna så att de möts i en gemensam punkt, en förändring av verkligheten som inte bör ha någon större inverkan på de beräknade snittkrafterna. Ett sätt som däremot inte bör användas är att definiera ett mycket kort balkelement i understråmen mellan de två teoretiska skärningspunkterna. Detta lilla element kommer nämligen att, i så fall, få så



avvikande styvhetssegenskaper att noggrannheten i beräkningen kan förstöras. Vissa program tillåter också en annan lösning, då man kan komplettera villkoret att de tre balkdelarna möts, med att införa en viss excentricitet ( eller "offset" ) för diagonalerna, vilket tar hänsyn till den ofullkomliga skärningen.

## 6.2 Skivstrukturer

Beräkning av två-dimensionella strukturer i form av skivor var en av de allra första tillämpningarna av Finita Element metoder, och fortfarande idag finns inte just några andra metoder som kan konkurrera för praktisk tillämpning. I detta avsnitt kommer vi därför att helt koncentrera oss på sådana metoder, och de överläggningar användande av dem kan leda till. Vi vill också, redan från början, upprepa att i och med att man överger balkstrukturerna, Finita Elementberäkningar är approximativa, och att man därför inte kan lita blint på alla resultat man får fram ur datorn.

### 6.2.1 Typfall

Under beteckningen "skivstrukturer" har vi, även om beteckningen därmed blir litet tveksam, velat inkludera några olika fall, som beräkningsmässigt ligger mycket nära beräkningen av till exempel väggskivor.

De program som finns tillgängliga för denna typ av beräkningar kan nämligen oftast hantera tre olika typer av problem, nämligen

- Plana spänningstillstånd
- Plana töjningstillstånd
- Axisymmetriska tillstånd

Det plana spänningstillståndet är de förhållanden som kan antas gälla i en väggskiva. Tillståndet kännetecknas av att, om skivans plan är beläget i x-y planet, spänningarna vinkelrätt mot planet, det vill säga spänningsskomponenterna  $\sigma_z$ ,  $\tau_{xz}$  och  $\tau_{yz}$  är noll. Detta medför att skivan vid belastning i planet kommer att, i viss mån, tjockleksförändras på grund av tvärkontraktionseffekten.

Om vi i stället betraktar ett tvärsnitt av en lång struktur med konstant sektion, till exempel ett rör utsatt för inre övertryck eller en lång jorddamm, kan man uppenbarligen inte få någon sådan kontraktionseffekt utan tjockleken av den utskurna, betraktade skivan måste hållas konstant. Detta fall som kännetecknas av att töjningen vinkelrätt mot det betraktade planet,  $\epsilon_z$ , är noll kallas ett plant töjningstillstånd. På grund av den förhindrade kontraktionseffekten kommer

man i detta fall att få spänningar som verkar vinkelrätt mot det betraktade tvärsnittet, det vill säga att  $\sigma_z \neq 0$ .

De två ovan skisserade fallen är ganska lika. Däremot är det tredje delfallet, nämligen det axisymmetriska, påtagligt annorlunda. Då emellertid beräkningarna kommer att bli mycket lika finns det anledning att ta upp detta fall här. De strukturer som kan behandlas är sådana som är rotationssymmetriska runt en längsaxel. Förutom att strukturen skall vara rotations-symmetrisk måste även samtliga laster och randvillkor vara sådana. Om samtliga dessa är oberoende av vinkelkoordinaten kan den axisymmetriska metodiken användas. En förutsättning är alltså att, för det betraktade lastfallet, symmetriaxeln förblir rak och att ingen vridning av strukturen äger rum.

I FIG 6.5 visas tre exempel på de olika delfallen.

### 6.2.2 Olika elementtyper

Redan i kapitel 5 har den enklaste formen av element för plana problem diskuterats, nämligen ett trendodigt,

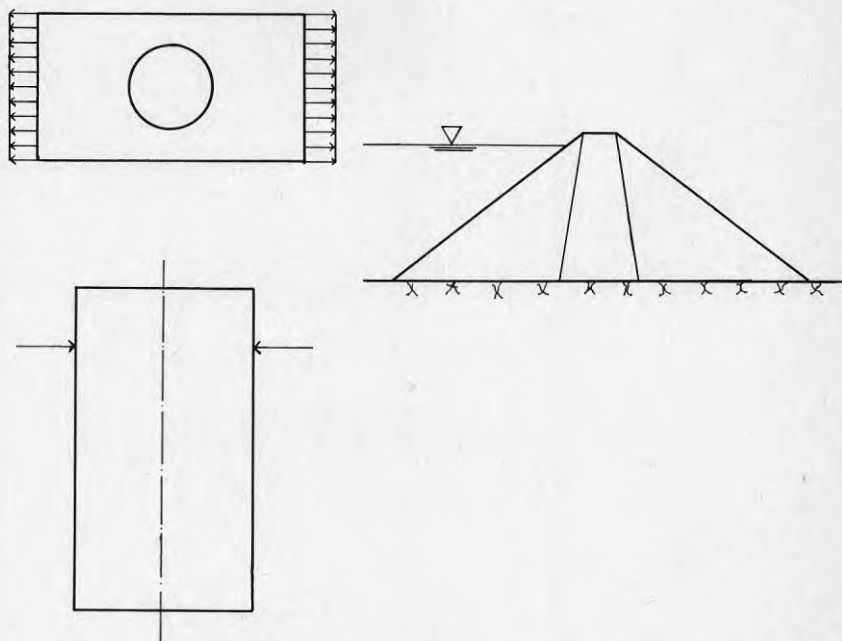


FIG 6.5 Exempel på plant spännings-, plant töjnings- och axisymmetriskt tillstånd

triangulärt Constant Strain element. Denna elementtyp var en av de första som användes, och den används fortfarande i viss utsträckning, även om andra elementtyper alltmer tar över.

Det triangulära elementet har emellertid en låg noggrannhet, på grund av att man i härledningen är tvungen att anta konstanta spänningar inom hela elementet. För att få noggrannare resultat måste bättre, och därmed mer komplicerade, elementtyper nyttjas.

Nästa steg i komplexitet kan sägas intas av ett fyrenodigt, rektangulärt plant element, FIG 6.6. Med hjälp av detta fås en, jämfört med CST-elementet, bättre representation av en ojämn spänningsfördelning, men fortfarande är möjligheterna att återge olika spänningsfördelningar relativt begränsade. Ett annat problem med dessa element är deras dåliga anpassbarhet till olika geometriska former, då man endast kan återge strukturer med räta vinklar, och någorlunda jämna mått.

Detta element, som ofta betecknas LSR (Linear Strain Rectangle) har en spänningsfördelning som varierar linjärt inom elementet.

Elementtypens begränsningar i fråga om att återge olika spänningsförhållanden leder till att man måste vara ganska försiktig vid användning av elementet. Så måste man till exempel se till att de olika elementen i en betraktad struktur ges någorlunda regelbundna mått. Man bör således inte låta kvoten mellan elementens kantlängder överstiga 2 a 2.5. Med mera långsträckta element försämras resultatets noggrannhet snabbt.

Man kan ganska lätt, successivt förbättra elementets noggrannhet genom att förse elementen med flera noder. Så visas i FIG 6.6 förbättrade element med 8 och 9 noder, av så kallad Serendipity- respektive Lagrange-typ. Den generella sanningen är att elementen, i sig själva, blir mer noggranna ju fler noder man ansätter. Dock

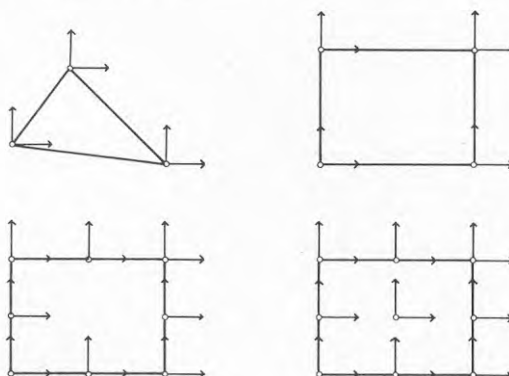


FIG 6.6 Olika element för analys av plana fall

skall man minnas att ju mer komplicerade element man använder, ju längre blir också, med givet antal element, den beräkningstid som åtgår i datorn.

Den ökade tidåtgången måste sättas i relation till den ökade noggrannhet som kan nås med de mer förfinade elementen. I TABELL 6.2 redovisas en undersökning som gjorts i samband med ett examensarbete vid institutet för Brobyggnad, med avsikt att kartlägga olika elementtypers effektivitet i förhållande till deras förbrukning av datortid. Ur tabellen framgår att man, för testexemplet i FIG 6.7, får bättre lösningar ju noggrannare element man tar, men också att den erforderliga beräkningstiden växer kraftigt.

Slutsatsen av den gjorda jämförelsen är att de mera komplicerade elementen har en fördel ur noggrannhets-synpunkt jämfört med de enklare. För det aktuella test-fallet förefaller det som det 8-nodiga elementet ger

TABELL 6.2 Jämförelse av olika elementtypers effektivitet för ett testproblem enl FIG 6.7  
Elementtyper enl FIG 6.6

Element	Noddelning	Resultat	Tidåtgång
3	1	5161	6
	2	8221	17
	4	11256	71
4	1	6664	6
	2	10355	24
	4	12644	101
8	2	11865	16
	4	13371	141
9	2	12049	21
	4	13597	210
	8	13928	2489

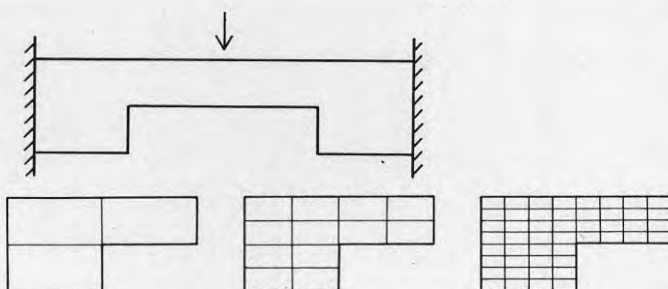


FIG 6.7 Testexempel för elementjämförelse  
(Noddelningar 1, 2 och 4)

de bästa resultaten i förhållande den erforderliga beräkningstiden. Detta resultat överensstämmer också väl med resultat redovisade i litteraturen.

Vid praktisk användning måste man dock komma ihåg att den insparade kostnaden för datortid lätt äts upp av en högre kostnad för skapandet av de indata som behövs för en körning, då detta indatagivande normalt blir mer ansträngande ju noggrannare elementet är, åtminstone om indata skall skapas manuellt. Vid användning av preprocessorer leder dock användning av högre ordningens element snarast till en besparing av tid.

Den utveckling som sker på FE området tycks också i allt större omfattning favorisera dessa, mer komplicerade element. Detta gäller kanske speciellt i samband med det isoparametriska begreppet som diskuteras i nästa avsnitt.

I samband med olika former av icke-linjära problem har dock tre- och fyrnodselementen vissa fördelar. Detta är en följd av de last- och randvillkorseffekter som diskuteras i avsnitt 6.2.5.

I de visade elementen har det förutsatts att elementen har två frihetsgrader i varje nod, nämligen en förskjutning i vardera x- och y-riktningen. Detta är dock inte nödvändigt utan man kan även använda andra frihetsgrader. Till exempel används i vissa fall någon förskjutningsderivata, till exempel  $\frac{\partial v}{\partial x}$  eller  $\frac{\partial u}{\partial y}$  som nodparameter för att åstadkomma vissa speciella effekter. Generellt sett är detta dock ingen riktigt bra lösning, även om den kan vara användbar till exempel då skivelement skall kombineras med balkelement.

### 6.2.3 Isoparametriska element

I det förra avsnittet nämndes att den stora nackdelen med rektangulära element är svårigheterna att representera oregelbundna geometrier, och att variera elementnätets finhet - och därmed dess möjlighet att återge spänningsgradienter på ett riktigt sätt - i olika delar av en struktur. Det finns dock möjlighet att, åtminstone i viss mån, övervinna denna svaghet med hjälp av så kallade isoparametriska element.

Innebörden i detta begrepp är ett element där man definierar en koordinattransformation mellan ett lokalt, rektangulärt koordinatsystem och det globala x-y systemet. På detta sätt kan man återge andra geometriska former än de rektangulära. Till exempel kan man med hjälp av ett isoparametriskt 4-nodselement modellera en nästan godtycklig fyrhörning.

Representationen av elementets utseende blir av motsvarande ordning som representationen av förskjutningarna



inom elementet. För det fyrnodiga elementet innebär det att sidorna måste bilda räta linjer mellan de givna hörnpunkterna.

Motsvarande representation kan också göras för ett åtta-nods-element, med utgångspunkt från det rektangulära elementet i FIG 6.6. Liksom förskjutningsvariationen inom elementet kommer representationen av elementets kanter att ges av en parabel som sammanbinder de givna nodpunkterna, se FIG 6.8.

Med hjälp av de isoparametriska elementen har man relativt goda möjligheter att återge mera komplicerade geometriska förhållanden på ett acceptabelt sätt. För att elementen skall ge goda resultat får man dock inte driva distortionen av den ursprungliga rektangeln alltför långt. De bästa resultaten får man om vinklarna är någorlunda räta, sidorna är ungefär lika långa och eventuella mittsidsnoder ligger ungefär mitt emellan de motsvarande hörnnoderna. Vissa, definitiva begränsningar för dessa förhållanden finns diskuterade i bland annat Zienkiewicz, se avsnitt 12.5 .

För de rektangulära elementen kan elementstyvhetmatrisen härledas analytiskt genom integration av de funktioner som anger töjningarna i elementet. I och med den koordinattransformation som erfordras för de isoparametriska elementen försvinner möjligheten till exakt integration. I stället måste någon av de numeriska integrationsmetoderna som diskuterades i avsnitt 5.7 tillgripas. I kommersiellt tillgängliga program används oftast så kallad Gauss-integration över ett punktgitte inom elementet. Den generella sanningen att resultaten blir bättre ju fler punkter som medtas gäller fortfarande. För de fyr-nodiga elementen ernås dock ingen ytterligare noggrannhet om mer än  $2 \times 2$  Gausspunkter används. Motsvarande gräns för åtta- och nio-nods-element är  $3 \times 3$  punkter.

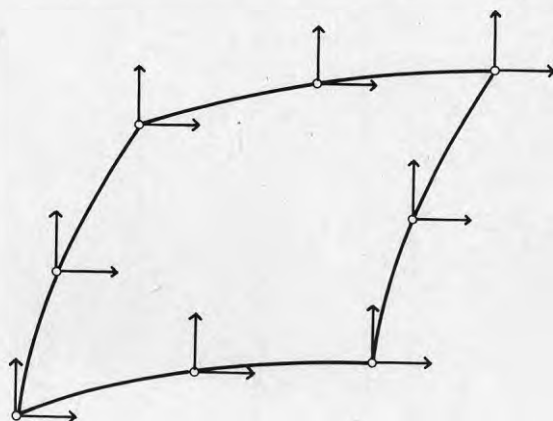


FIG 6.8 Isoparametriskt åtta-nods skivelement, byggande på åtta-nods rektangeln i FIG 6.6

## 6.2.4 Resultatpresentation

För beräkningarna av olika balkstrukturer som diskuterades i avsnitt 6.1, var det inte så svårt att presentera resultaten av beräkningarna. Förutom förskjutningsvärdena i noderna bestod resultaten endast av värden på snittstorheterna normalkraft, böjmoment och eventuellt tvärkraft i elementen.

Om man däremot behandlar skivproblem kan resultaten bli mera svårtolkade. De grunddata som beräknas är i detta fall, förutom förskjutningsvärdena, töjnings- och spänningsvärden i ett antal punkter inom elementet. Eftersom man till exempel för ett fyrnodselement använder fyra integrationspunkter får man ofta de resulterande spänningarna utskrivna i dessa fyra punkter. Med 200 element och 3 redovisade spänningskomponenter blir detta totalt 2400 spänningsvärden, och åtminstone 10 å 12 sidor utskrifter.

Det finns två argument för att redovisa spänningsvärden i just de Gauss-punkter som använts för integrationen av styvhetsuttrycken. Förutom att viss information redan finns tillgänglig för dessa punkter, kan också spänningsvärdena här förväntas bättre representera den riktiga spänningsfördelningen i strukturen. Tyvärr får man dock på detta sätt spänningsvärden beräknade i relativt ointressanta punkter, till exempel i punkter på ett litet avstånd från ytorna på en balk. Olika metoder finns också att vidarebearbeta de primärt redovisade spänningsvärdena. Så redovisas i vissa fall beräknade medelvärden för endera de olika elementen, eller för varje nod. Man skall då bara komma ihåg att dessa måste ge en sämre noggrannhet än de ursprungliga värdena.

På grund av de stora mängderna data kommer en grafisk presentation av de beräknade resultaten att vara mycket värdefull vid en utvärdering. I stället för att leta i de långa tabellerna med spänningsvärden kan man med ett snabbt ögonkast få en överblick över vilka spänningsnivåer som gäller i olika delar av strukturen.

Ett flertal olika former av grafisk presentation av spänningar förekommer. Den vanligaste, och enklaste, är att pilar ritas, svarande mot de olika spänningskomponenternas storlek i de olika punkterna, med en viss konvention för att skilja tryck- från dragspänningar.

För att bedöma ett visst lastfalls påverkan på strukturen kan det dock kanske vara viktigare att granska huvudspänningarnas storlek i olika punkter, det vill säga största drag resp tryckspänningarna och deras riktningar. En sådan figur visar på ett mycket tydligt sätt det kraftspel som finns inom strukturen.

Om man är mer intresserad av storleken än riktningarna av de största spänningarna, kan ofta en annan presentation väljas, nämligen en så kallad "iso-stress" plott-

ning. I denna kommer man att få nivålinjer som sammanbinder punkter i strukturen som har samma spänningsvärdet. Denna plottning kan ofta göras för vilken som helst av de beräknade spänningskomponenterna eller för endera av de två huvudspänningarna.

De uppritade isostresslinjerna kommer normalt att ges med relativt låg noggrannhet. Beroende på hur linjerna är beräknade kommer nämligen de i metodiken inbyggda approximationerna att ge upphov till diskontinuerliga spänningsnivåer eller andra approximationsproblem. En viss skepsis till de givna resultaten är därför befogad.

I FIG 6.9 och 6.10 visas dessa olika former av grafisk resultatpresentation för ett fall, där en väggskiva med ett dörrhål påverkas av en skjuvkraft och en utbredd vertikal belastning. Plottningen har producerats av programmet V-FEM. Tyvärr kan i denna bok inte återges de färger som i programmet används för att skilja drag- från tryckspänningar.

#### 6.2.5 Lokala effekter

I väggskivan som redovisas i FIG 6.9 och 6.10 fås stora spänningskoncentrationer dels vid dörrhålets övre högra hörn, dels vid införingen av den koncentrerade lasten. För att återge dessa någorlunda exakt vill man gärna öka antalet element inom de intressanta regionerna. Om man gör det kommer man dock att finna att spänningarna invid hörnet blir allt större ju finare indelning som görs. Förklaringen till detta är att hörnet utgör en så kallad singulär punkt som enligt elasticitetsteorin måste få oändligt stora spänningar. Detta är dock naturligtvis inte sant i verkligheten utan materialet kommer att ytterst lokalt påverkas så att spänningarna minskas, endera genom uppsprickning eller för andra material genom plasticering.

Svårigheten ligger alltså i att för praktiskt bruk på ett rimligt sätt kunna utvärdera dessa lokala fenomen inverkan, och att ta hand om de spänningskoncentrationer som uppstår.

Vissa svårigheter uppstår ofta vid ränderna av en analyserad skivstruktur. Detta har då att göra med endera påförandet av utbredda laster eller upplagsförhållanden och dylikt.

När det gäller en fördelad last längs en elementkant måste den, liksom i fallet med balkelementen påföras som koncentrerade nodlaster. För det rektangulära fyrenodselementet kan detta göras med hjälp av lika stora punktkrafter i de båda noderna längs sidan.

För såväl högre ordningens som isoparametriska element blir dock förfarandet något mera komplicerat. För här-

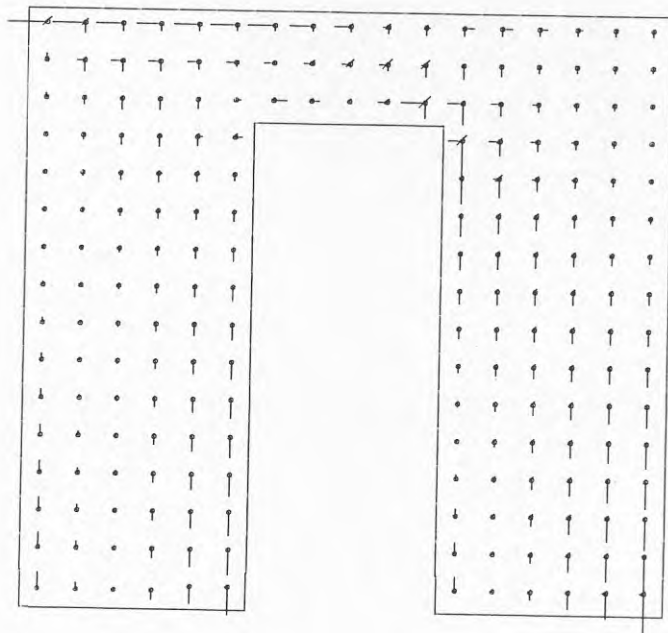
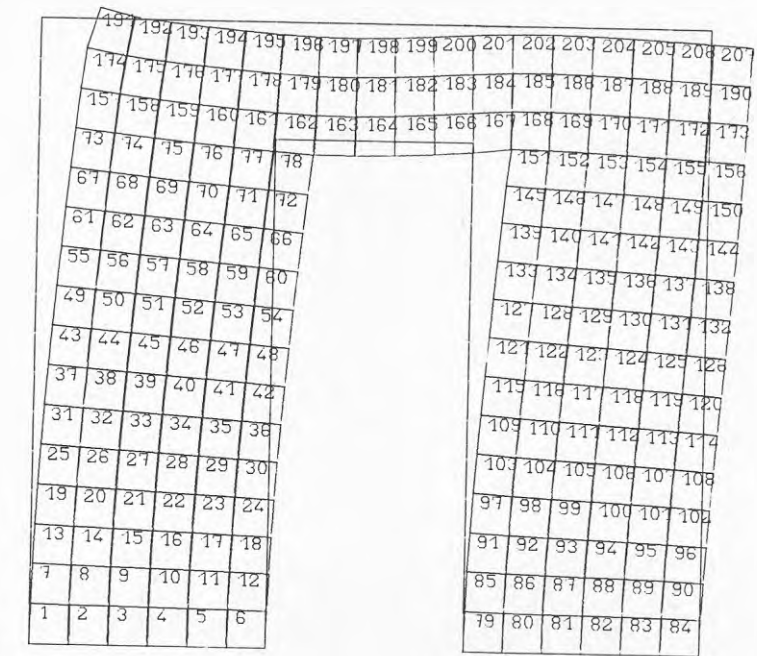


FIG 6.9 Förskjutningar och spänningar i analyserad väggskiva. (Producerade av programmet V-FEM)

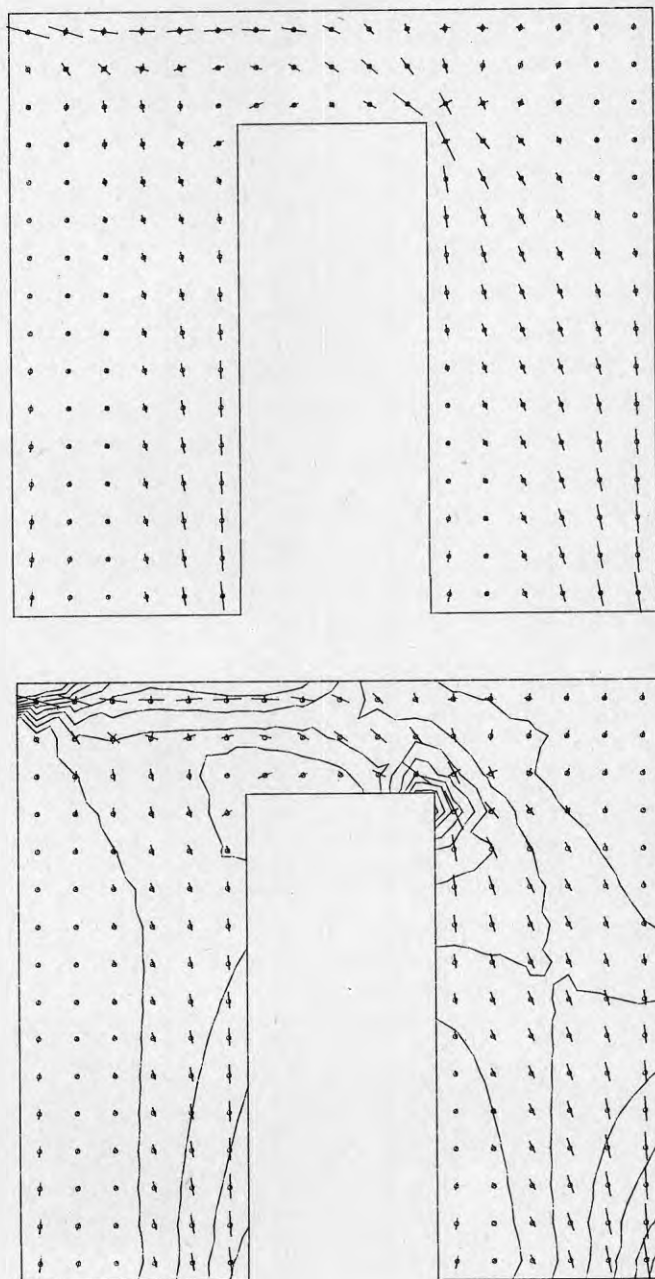


FIG 6.10 Huvudspänningar och isostresser i analyserad väggskiva. (Producerade av programmet V-FEM)



ledningen av de mot den utbredda belastningen svarande nodlasterna skall en metodik analog med den i avsnitt 6.1.4 tillämpas, det vill säga en integration av de olika basfunktionerna längs den aktuella randen. Detta kommer att leda till en fördelning som kan förefalla omotiverad. Så kommer man till exempel, för ett rektangulärt åtta eller nio noders element enligt FIG 6.6 att få en relativ fördelning på de tre noderna längs kanten av  $1 - 4 - 1$ .

Motsvarande effekt kommer också fram för upplagsreaktionerna i en rörelseförhindrad rand.

För vissa skivberäkningar, till exempel fallet i FIG 4.6c kommer också speciella randvillkor att behöva användas. Villkoret ur systemskissen är att noderna på den radie som skär ut den betraktade sektorn enbart får röra sig i radiell led. Denna typ av sneda randvillkor kan många större program hantera.

### 6.2.6 Substrukturer och superelement

Av exemplet i FIG 6.9 framgår att en relativt fin elementindelning behövs för att på ett någorlunda riktigt sätt återge förhållandena kring till exempel ett dörrhål. Antag nu att den visade väggdelen ingår i en stabiliserande vägg i ett 16-våningshus. Med tre i bredd och 16 på höjden skulle den totala väggskivan få lov att representeras med totalt över 12000 element och med över 10000 noder, det vill säga ett ekvationssystem med över 20000 obekanta.

Då ett system av denna storleksordning blir mycket dyrt att lösa vill man gärna finna något sätt att minska storleken på problemet utan att för det minska lösningens noggrannhet. En möjlighet som ibland kan utnyttjas för att åstadkomma detta är användning av så kallade superelement.

De superelement som det kan bli tal om är till exempel av den form som exemplet i FIG 6.9 visar. Grundtanken med metoden är att söka finna ett styvhetsuttryck för en sådan väggdel, uttryckt endast i de frihetsgrader som finns längs väggdelens ränder, det vill säga att man "kondenserar" bort beroendet av de inre noderna i väggdelen.

Efter att ha byggt upp styvhetsmatrisen för superelementet analogt med första steget av en analys av väggskivan i sig själv, kan man - om man förutsätter att de inre noderna är obelastade - genom delinverteringar av styvhetsmatrisen få fram ett styvhetsuttryck som bara berör randnoderna. Det man då fått fram kan sägas vara en superelementstyvhetsmatris för väggdelselementet, med hänsynen till de inre nodernas rörelser i förhållande till de yttre inbyggd. Denna styvhet kan sedan kombineras med de övriga superelementens till att

bygga upp hela strukturens styvhetsmatris, som nu kommer att vara betydligt mindre än den skulle ha varit om hela strukturen modellerats med enkla element.

Med denna styvhetsmatris och de yttre belastningarna givna kan strukturens förskjutningar beräknas. Efter detta återvänder man till superelementnivån, där de olika grundelementens förskjutningar kan återbildas, varur man lätt härleder spänningar och dylikt.

Det man egentligen gjort är att man infört en mellan-nivå mellan elementanalysen och strukturanalysen i det renodlade schemat i kapitel 5. Vissa program kan hantera denna typ av superelementbildning, eller substrukturering som den också kallas. Vinsterna för stora strukturer kan i vissa fall bli betydande, speciellt om superelementen repeteras som i det här skissade exemplet. Om så inte är fallet kan substruktureringen ändå vara motiverad genom att möjliggöra lösning av problem som med normal analys skulle vara för stora.

### 6.3 Plattstrukturer

Beräkning av plattstrukturer är en viktig del av en byggnadskonstruktion. Det är då i de flesta fall fråga om armerade betongplattor. För denna strukturtyp används normalt handbokslösningar av standardfall för analysen.

Med analys skall vi i detta avsnitt avse en beräkning där ett linjärt, elastiskt materialbeteende förutsätts, ett antagande som man enligt normerna får göra för analys av betongkonstruktioner, även om detta inte - vilket vidare diskuteras i avsnitt 7.2.1 - är sant för en betongstruktur.

Ett mellanting mellan analys och dimensionering utgörs av gränslastbetraktelser, där för plattor främst den vanligt använda brottlinjeteorin är aktuell. Denna metod kommer att något behandlas i avsnitt 6.3.6.

Plattor kan datorberäknas med flera olika metoder. Till exempel kan olika typer av FE approximationer användas. Som alternativ kan nämnas beräkning enligt brottlinjeteori eller med hjälp av ansatsfunktioner i form av trigonometriska serier.

Av de nämnda metoderna är otvivelaktigt FE metodiken den som är lättast att hantera i den meningen att man kan använda generella program, utan några speciella problem. Metoden har också klara fördelar då det gäller att ta hänsyn till oregelbundna plattutformningar, håltagningar och dylikt, samt då man vill ta hänsyn till olika former av förstyvningar.

Finite Element metodiken har dock också nackdelar vid plattanlys. Dessa har främst att göra med noggrannhe-

ten i lösningen. En plattas verkliga egenskaper vid böjning kan nämligen inte så lätt överföras till platt-elementet. Detta innebär att man även för det enklast tänkbara fallet, det vill säga en rektangulär fritt upplagd platta belastad med en jämnt fördelad last, måste dela in plattan i ett ganska stort antal element för att få bra resultat. Ur beräkningstidssynpunkt vore i sådana fall en lösning med hjälp av en utböjningsansats med några sinusuttryck (jämför avsnitt 6.3.7) klart överlägsen.

### 6.3.1 FEM för plattanlys

Då Finita Element metodiken kan sägas vara den mest generellt användbara metoden för plattanlys skall vi främst diskutera denna beräkningsmetod, trots att den, som påpektats ovan, inte är lika lämpad för detta som för skivanlys.

Att FE metodiken blir besvärlig har att göra med den differentialekvation och de energiuttryck som gäller för plattböjningsfallet. I stället för att töjningarna - som i skivfallet - fås som förstaderivator av förskjutningarna, fås de i plattfallet som andraderivator, en skillnad som leder till ökade krav på de basfunktioner som används vid formuleringen av elementens styvhetsegenskaper.

För plattfallet måste vi för att åstadkomma denna bättre representation normalt definiera flera nodparametrar än för skivfallet. För de vanligaste typerna av element innebär detta att man använder tre parametrar i varje nod. Förutom transversalutböjningen  $w$  brukar man också använda rotationerna  $\Theta_x$  och  $\Theta_y$ .

Ett mycket vanligt element är det fyrnodiga, rektangulära elementet i FIG 6.11.

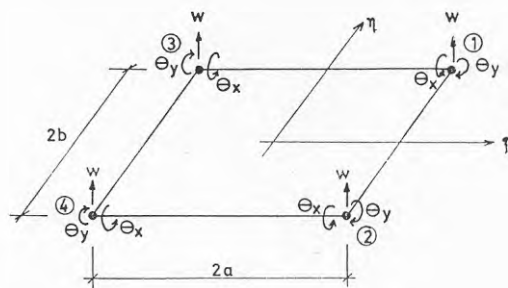


FIG 6.11 Fyrnodigt element för analys av plattor

För detta element kan man skriva en förskjutningsansats som är ett polynom i de lokala koordinaterna  $\xi$  och  $\eta$ . Förskjutningsuttrycket består i detta fall av 12 termer (eftersom man har 12 nodparametrar), vilket gör att man kan välja ett komplett tredjegradspolynom kompletterat med två termer av fjärde ordningen. Vi har då antagit att rotationsparametrarna  $\theta$  kan tolkas som derivator av utböjningen  $w$  i de båda axelriktningarna, vilket är det vanligaste antagandet.

Ur förskjutningsansatsen kan sedan krökningarna, och därur töjningarna, i elementet härledas. Det gäller ju nämligen att

$$\epsilon_x = -z \frac{d^2 w}{dx^2} = z k_x$$

$$\epsilon_y = -z \frac{d^2 w}{dy^2} = z k_y$$

$$\gamma_{xy} = -z 2 \frac{d^2 w}{dx dy} = z k_{xy}$$

där  $z$  är avståndet från medelytan, och  $k$  är krökningarna i de olika riktningarna.

Analogt med skivfallet, kan detta skrivas som en viss matris, innehållande vissa funktioner i  $\xi$  och  $\eta$ , multiplicerad med nodförskjutningsvektorn.

Krökningskomponenterna omvandlas till moment genom multiplikation med en matris innehållande böjstyheter, enligt

$$m_x = D_x k_x + D_1 k_y$$

$$m_y = D_1 k_x + D_y k_y$$

$$m_{xy} = D_{xy} k_{xy}$$

där  $m$  är moment per längdenhet.

I det isotropa fallet motsvaras dessa böjstyheter av

$$D_x = D_y = \frac{Eh^3}{12(1-\nu^2)}$$

$$D_1 = \nu D_x$$

$$D_{xy} = \frac{1-\nu}{2} D_x$$

där  $h$  är plattelementets totala tjocklek, samt  $E$  och  $\nu$  materialets elasticitetsmodul respektive Poisson-tal.

Ur kröknings- och momentuttrycken, kan ett uttryck för den inre, upplagrade energin i plattan härledas. På

analogt sätt som i avsnitt 6.2 sätts denna energi lika med ett uttryck för den yttre energin av pålagda laster, vilket resulterar i ett styvhets samband, som relaterar krafterna (och momenten) i nodfrihetsgraderna till förskjutningarna (och rotationerna).

Andra elementtyper kan också användas med samma principiella härledningar. Det är dock svårare att få fram bra plattelement, än att åstadkomma bra skivelement. Många försök har gjorts, och speciallitteraturen innehåller ett stort antal jämförelser av olika elementtyper beteenden för olika lastfall.

Ett begrepp som ofta används i samband med element för plattböjning är konvergenta och icke-konvergenta element. Bakgrunden till denna skillnad är teoretiska krav att, för att ett element skall kunna användas, den approximativa utböjningens lutning måste vara kontinuerlig i elementgränserna, det vill säga att inga "knyckar" i utböjningskurvan får förekomma. Sådana skulle ju, i princip, kräva ett oändligt stort moment i denna linje. Plattelement som uppfyller detta kontinuitetskrav kallas konvergenta (eng. conforming).

Det visar sig dock att icke-konvergenta element, som alltså inte uppfyller dessa teoretiska krav, i många sammanhang ger minst lika bra resultat som de konvergenta, trots att de alltså egentligen inte borde kunna användas. Till exempel är det element som visas i FIG 6.11 icke-konvergent.

### 6.3.2 Alternativa formuleringar

För att söka övervinna svårigheterna att formulera bra elementtyper för plattanlys har flera alternativa tillvägagångssätt provats. En vanlig utväg är att vända något på begreppen, och använda en kraftmetodbaserad härledning. I så fall kan man till exempel komma fram till så kallade hybridelement. I dessa ansätts inte enbart förskjutningar och rotationer som obekanta variabler, utan även vissa kraftbegrepp medtas. I FIG 6.12 visas ett rektangulärt hybridelement, som används i bland annat CHALMFEM programmet PREFEM. I detta används nodernas transversalutböjning tillsammans med momentresultanterna längs elementsidorna som obekanta parametrar.

För användaren av färdiga program leder inte användningen av hybridelementen till några särskilda svårigheter, då arbetet med programmen fortfarande blir det samma som med ett FEM program av vanlig typ.

För vissa fall kan dock dessa hybridelement visas ge goda resultat. Framförallt kan de beräknade momenten i plattelementen, på grund av elementhärledningen, antas bli bättre representerade, på bekostnad av noggrannheten i de beräknade utböjningarna.



Hybridelementen har dock nackdelen att vara svåra att koppla ihop med andra elementtyper.

Ett principiellt annat tillvägagångssätt kan fås genom en analogi till Timoshenko-teorin för balkelement, beskriven i avsnitt 6.1.3. Om man på samma sätt önskar ta hänsyn till tvärkrafterna i plattan kan man använda en platt-teori vanligen tillskriven Mindlin eller Reissner.

Liksom för balkfallet kommer då nodparametrarna  $\theta_x$  och  $\theta_y$  att motsvara lutningen av en yta, i det odeformerade läget vinkelrät mot plattans medelyta.

Plattelementets transversalutböjning kan, efter detta antagande beskrivas med hjälp av funktioner, som nu inte längre behöver ha kontinuerliga derivator, utan endast behöver ge kontinuetet i utböjningarna. En av de största fördelarna med detta är att man kan skapa isoparametriska plattelement, till exempel med form som i FIG 6.8. Förutom denna möjlighet skulle man alltid vara hänvisad till att använda triangulära plattelement, med deras förhållandevis låga noggrannhet, vid FE-modellering av oregelbundna plattor. (Vi bortser därmed från den teoretiska möjligheten att skapa så kallade sub-parametriska element).

Vissa problem uppstår dock vid användning av denna typ av plattelement. För det första representeras momentvariationen inom elementet med lägre noggrannhet än med motsvarande element baserat på vanlig platt-teori. Denna skillnad i representation kan också ge upphov till vissa skillnader i noggrannhet vid bestämning av en plattas nedböjning, se TABELL 6.3 som visar beräknad mittnedböjning för en kvadratisk platta utsatt för en koncentrerad punktlast i plattmitt, beräknad med hjälp av 4-nodselement av de två olika typerna.

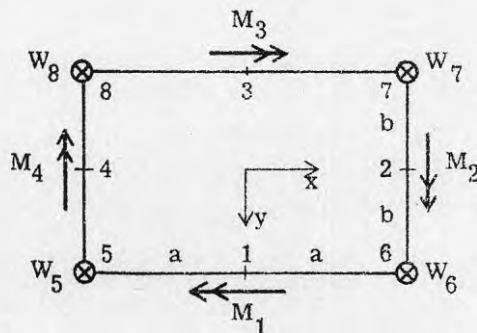


FIG 6.12 Hybridelement för plattanalis använt av programmet PREFEM

TABELL 6.3 Jämförelse av nedböjning för kvadratisk platta belastad med koncentrerad last i plattmitt. (Multiplikator: Exakt nedböjning enl teori med enbart böjning beaktad)

Element	Element per fjärdedel		
	2*2	4*4	8*8
Kirchhoff (4-nods)	1.0626	1.0196	1.0059
Mindlin (4-nods)	0.9928	0.9955	0.9986

En annan nackdel med Mindlin-elementen är att man kan få problem med så kallade lockingeffekter, vilka är en följd av att styvheten mot tvärkraftsdeformation blir för stor i plattan, med de gjorda antagandena. Om nämligen styvhetsmatrisen för plattelementet beräknas exakt kommer denna effekt att slå igenom, vilket får till följd att utböjningarna i plattan kommer att bli alltför små. I vissa fall kan felaktigheterna bli så stora att den beräknade nedböjningen med flera tiopotenser understiger den riktiga.

Receptet mot locking är att integrera plattans skjuvstyvhetsbidrag en aning inexakt. Man integrerar alltså numeriskt, och är tvungen att begränsa noggrannheten i denna integration genom att ta med ett alltför litet antal integrationspunkter, jämför avsnitt 5.7. Med detta knep, som alltså bryter mot den i kapitel 5 givna huvudprincipen att fler integrationspunkter ger bättre resultat, kan alltså svårigheterna med denna elementtyp övervinnas, och man kan få mycket goda resultat.

### 6.3.3 Utbredda laster

Liksom i fallet med balkelement är plattor ofta belastade med fördelade belastningar. För att ta rätt på hur dessa på bästa sätt skall representeras måste en analys liknande den i avsnitt 6.1.4 göras. Man kommer då att finna att för det 4-nodiga rektangulära elementet baserat på Kirchhoff-teorin en jämnt fördelad belastning med intensiteten  $q$  kommer att beskrivas enligt FIG 6.13, där alltså, liksom i balkfallet, de koncentrerade nodlasterna kompletteras med punktmoment i noderna.

Man inser dock ur denna figur att för en platta med jämnstora element, samtliga med samma lastintensitet, dessa koncentrerade moment i noder som ej ligger på plattans periferi kommer att ta ut varandra. De koncentrerade nodkrafterna behöver alltså i sådana fall

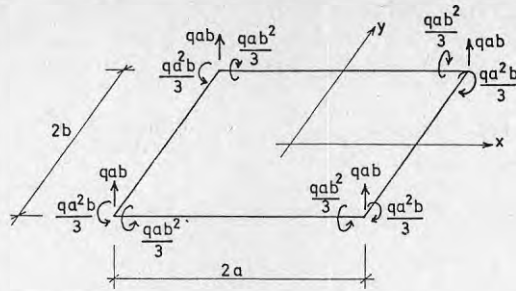


FIG 6.13 Representation av jämnt fördelad belastning på 4-nods plattelement (Kirchhoff-teori)

endast kompletteras med vissa nodmoment längs plattans periferi.

För plattor med någorlunda fin elementindelning kan också dessa nodmoment helt försummas, utan att resultaten nämnvärt kommer att försämrats.

För element baserade på Mindlin-teorin representeras en utbredd belastning alltid med hjälp av enbart koncentrerade nodkrafter, svarande mot en viss andel av den totala belastningen på elementet.

#### 6.3.4 Ortotropa och förstyvade plattor

Ovanstående resonemang har främst avsett plattor som är elastiska och isotropa, det vill säga har samma egenskaper i alla riktningar. Detta innebär att plattor av olika metaller lätt kan behandlas.

För byggnadskonstruktörens verksamhet är dock plattor av betong betydligt mera intressanta. Som vidare diskuteras i avsnitt 7.2.1 har betongen egenskaper som alltid ställer till problem vid analysen, varvid den mest problematiska är betongens oförmåga att bära dragspänningar av någon omfattning. Detta får till följd att plattor av betong borde analyseras under förutsättningar om att böjstyvheten varierar beroende av armeringsinnehåll, och kanske till och med av påkännings-tillståndet i olika delar av plattan.

Detta leder till att man har ett behov av att kunna analysera plattor med olika böjstyvhetssegenskaper i olika riktningar, och i olika punkter. Detta ställer inte heller till några större problem vid en FE analys. Den enda skillnaden som fås är att böjstyvhetserna  $D_x$ ,  $D_y$ ,  $D_1$  och  $D_{xy}$  inte kan bildas direkt ur tjocklek och materialparametrar, utan att armeringsinnehåll och/eller uppsprickning i olika riktningar måste beaktas.

Om analysen skall bli enkel förutsätts dock att man gör vissa antaganden om vilka delar av plattan som kommer att påverkas av dragspänningar, och tar hänsyn till detta vid bildandet av böjstyvheter, eftersom dessa strängt taget blir olika stora beroende på momentens riktning. En helt korrekt analys skulle egentligen - med antaganden om ospruckna förhållanden - göra en första analys och låta resultaten från denna styra antaganden om uppsprickning och armering samt de härav beroende böjstyvhetsförändringarna. Fortsatta iterationer skulle sedan ge ett resultat där uppsprickningen överensstämmer med påkänningstillståndet. Ett sådant förfarande blir dock komplicerat, och blir främst aktuellt för dimensioneringsprogram.

En annan sak som kan göra analys av plattor besvärlig är om plattan är försedd med olika förstävningar, exempelvis i form av ribbor på plattans undersida. Avsikten med dessa är ju att öka styvheten mot böjning, och denna effekt bör också medtas vid en analys av plattan. I samband med Finita Element beräkningar kan detta relativt enkelt behandlas genom att plattan ges olika böjstyvheter i x- och y-riktningarna.

En annan möjlighet finns också i många program, då man har element i form av balkar som kan utnyttjas för förstävningar. Dessa element, som även kan användas för beräkning av balkroster, har ett utseende och frihetsgrader enligt FIG 6.14. Man noterar att elementet inte överensstämmer med de vanliga balkelementen från avsnitt 6.1 utan istället för axialfrihetsgraden har en vridningsfrihetsgrad. För bildningen av denna typ av element är böjstyvhet och vridstyvhet nödvändiga parametrar. Det kan dock vara svårt att ge dessa rimliga värden, då det på grund av förstävningens ofta excentriska placering i förhållande till den samverkande plattan, är svårt att definiera de axlar som balken kan antas böjas och vridas omkring.

Vissa program tillåter dock att man definierar förstävningselement som ligger med en viss excentricitet i förhållande till plattans plan.

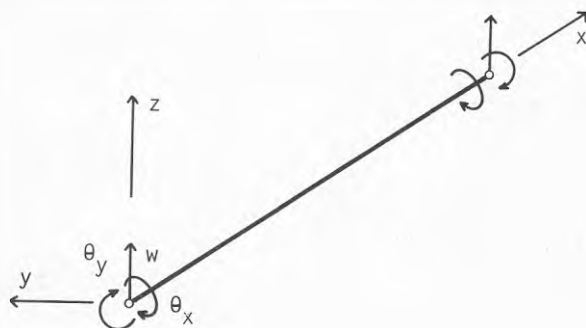


FIG 6.14 Balkelement för plattförstävningar

### 6.3.5 Fourieranalys av plattor

Vi skall även något diskutera alternativa beräkningsmetoder för plattor, vilka kan vara intressanta jämfört med FE beräkningar i vissa fall.

En möjlig metod för analys av regelbundna plattor är via ansatser av trigonometriska funktioner för utböjningen. Denna metodik kan endast tillämpas för plattor med regelbundet utseende, men är i gengäld mycket effektiv för sådana. Metoden har dock nackdelen att den är något svår att använda i generella program för plattanalys.

Bakgrunden till metoden är att man för utböjningen i plattans olika punkter ansätter en summa, med okända koefficienter, av trigonometriska funktioner. För en rektangulär, fritt upplagd platta kan man till exempel sätta

$$w(x,y) = \sum_{m,n} A_{mn} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right)$$

där  $a$  och  $b$  är plattans sidlängder och  $m$  och  $n$  får variera från 1 till någon lämplig gräns. Genom insättning av ovanstående uttryck i den differentialekvation som gäller för plattan, kan man då plattans belastning  $q(x,y)$  och dess böjstyvhetsvariation är kända, låta datorn lösa ut värden på koefficienterna  $A$ , vilket leder till en lösning som så bra som möjligt överensstämmer med den sanna.

Problemet med metoden är att finna en funktionsansats som uppfyller de randvillkor som gäller för plattan. Om bara det problemet löses är resten relativt enkel.

### 6.3.6 Brottlinjeteori för plattor

Vissa försök har gjorts att anpassa den vanligt använda brottlinjeteorin för datorberäkningar. Fördelen med denna metod är att den på ett någorlunda riktigt sätt tar hänsyn till betongplattans varierande böjstyvhetsgenskaper på grund av varierande armeringsinnehåll och dylikt. Några program för denna metodik finns också kommersiellt tillgängliga.

I FIG 6.15 visas en brottlinjeanalys av en kvadratisk, längs periferin fast inspänd platta, belastad med en centrisk punktkraft, hämtad ur Bäcklund, J: "Finite Element Analysis of Nonlinear Structures", CTH, Göteborg 1973.



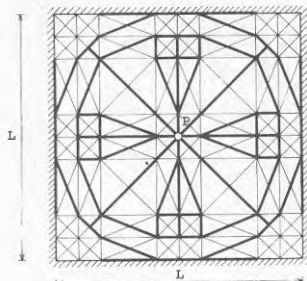


FIG 6.15 Brottlinjeanalys av platta (hämtad ur Bäcklund, 1973)

#### 6.4 Andra strukturer

I avsnitten 6.1 till 6.3 har de tre, för en byggnads-konstruktör, vanligaste strukturerna behandlats. Det finns dock ytterligare några strukturer som kan vara intressanta att analysera.

##### 6.4.1 Skalstrukturer

För beräkningar av olika typer av skalstrukturer finns ett stort antal olika Finita Element program utvecklade. De använda elementen består i dessa fall av en kombination av skiv- och plattelement som lagts samman på ett sätt som medför att mer eller mindre generella skalformer kan byggas upp.

Så redovisar exempelvis Bäcklund (refererad ovan) ett skalelement bestående av en kombination av ett triangulärt CST skivelement ( FIG 6.6 ) och ett triangulärt hybrid plattelement analogt med FIG 6.12 . Resultatet blir ett element som har som obekanta dels förskjutningarna av noderna i de tre axelriktningarna, dels momentresultanterna längs elementets sidor. Detta element kan användas för att bygga upp godtyckliga skalformer.

Ett rent förskjutningsmetodbaserat element för skalanalys redovisas i Eriksson, A: "The Finite Element Method for Sheet Metal Structures", BFR-rapport D31:1980. I denna beskrivs ett element som har sex frihetsgrader i var och en av sina fyra noder, nämligen tre förskjutningar i axelriktningarna och tre rotationer runt koordinataxlarna.

Över huvud taget finns ett mycket stort antal elementtyper utvecklade, avpassade för olika former av skal,

och de stora FEM programmen erbjuder goda möjligheter för skalanalys. Det är dock sämre med specialprogram för just denna strukturtyp.

Vid analys av skal av betong gäller samma synpunkter på materialbeteenden och dylikt som har angivits för plattor och skivstrukturer.

#### 6.4.2 Tre-dimensionella strukturer

De strukturtyper som hittills har behandlats har ur beräkningssynpunkt kunnat behandlas som varande två-dimensionella, även om fler av dem egentligen har en utsträckning i tre dimensioner. Problemen har dock på olika sätt, genom införande av diverse förenklingar, kunnat analyseras som plana.

Vissa andra typer av problem har dock en sådan utformning att de måste betraktas som tre-dimensionella. Detta gäller till exempel olika former av massiva eller solida kroppar, som i vissa tillämpningar har stor betydelse, även om byggnadskonstruktören inte så ofta kommer i kontakt med dem.

För dessa strukturer finns olika Finita Element utvecklade, vilka finns tillgängliga åtminstone i de stora, generella FEM programmen. Elementen bygger ofta på en generalisering av de plana rektangulära eller isoparametriska elementen, genom en utbyggnad i den tredje dimensionen. Så övergår till exempel det vanliga fyrenodiga rektangulära elementet i ett åtta-nodigt rätblockselement. De frihetsgrader som då används blir tre förskjutningar i koordinataxlarnas riktningar.

Möjligheterna att använda dessa element begränsas dock ofta av kostnaderna. Vid en modellering av en 3-dimensionell solidstruktur, kommer nämligen såväl antalet frihetsgrader som styvhetsmatrisens bandbredd att växa mycket snabbt, och beräkningarna blir mycket dyra. Det är alltså mycket angeläget att i dessa fall, om möjligt, bryta ned den komplexa 3-dimensionella verkligheten till en hanterbar, 2-dimensionell modell, jämför kapitlen 1 och 10.

En tredimensionell tillämpning som är intressant ur byggnadskonstruktörens synvinkel är beräkning av stabiliserande system i flervåningshus.

Dessa system består ju ofta av flera samverkande plana väggskivor, och dessutom trapphus, hiss-schakt och dylikt. För att återge dessas samverkan i en rymdstruktur kan inte väggskivorna analyseras en i taget, utan hela den tre-dimensionella strukturen måste tas med, för att till exempel den inbördes fördelningen av de verkande krafterna mellan de olika väggskivorna skall återges på ett riktigt sätt.

Det stabiliserande systemet påverkas dels av horisontella laster exempelvis i form av vindlaster, men det skall också styra de vertikala laster som påverkar de olika bjälklagen och som förs ned till marknivån via pendelpelarkedjor. På grund av samverkan mellan dessa belastningar kommer man att få inspänningsmoment i väggskivorna, moment som är beroende av de verkande vertikala lasternas storlek.

Vid en viss vertikal belastning kommer systemet att bli instabilt. Det är viktigt att kunna konstatera att denna last, med god marginal, överstiger de laster som den verkliga strukturen kan komma att belastas med.

Problemet att beräkna kraftfördelningen i ett på detta sätt stabiliserat hus kan behandlas på litet olika sätt. Några av dem presenteras i Peterson, H: "Analysis of Loadbearing Walls in Multistorey Buildings", CTH, Göteborg 1974.

En annan, förenklad metod för stabilitetsberäkningen, beskrivs i Lorentsen, M: "Stabilisering av byggnader", Inst för Brobyggnad, KTH, Stockholm 1974, med vidhängande övningskompendium.

Några färdiga program för denna typ av analys finns kommersiellt tillgängliga. Den sist refererade metoden kan också utföras med hjälp av något allmänt program för matrishantering, kompletterat med en del manuella beräkningar.

## 7 PROGRAM FÖR DIMENSIONERING

I detta kapitel skall vi behandla datorprogram för dimensionering och ge en sammanfattning av de normer som reglerar detta.

Någon allmänt vedertagen definition på dimensioneringsprogram finns inte så vitt vi vet. Man kan i dagligt tal mena program som ger utdata allt ifrån dimensionerande moment och snittkrafter till sådana som genererar "färdiga" arbetsritningar och mängdförteckningar. Var gränsen mellan analys- och dimensioneringsprogram går, kan vara svårt att fastställa. Därför har vi infört en egen definition som ligger till grund för vår indelning av programmen mellan analys och dimensionering. Den lyder som följer:

-dimensioneringsprogram är ett program, som med hänsyn till aktuella belastningar bestämmer strukturens utformning, så att givna normer och krav som ställs på strukturens funktion uppfylles.

Program som enbart ger dimensionerande moment och snittkrafter hänförs således inte till dimensioneringsprogram.

Man kan indela dimensioneringsprogrammen i två huvudtyper beroende på vilken sorts belastningar de kräver som indata. Den ena består av program som kan ta hand om yttre belastningar och räkna om dessa till snittpåkänningar. Program hörande till den andra klarar endast att ta emot snittbelastningar. I det förra fallet finns antingen en "inbyggd" analyserande programdel eller en möjlighet att överföra analysresultaten direkt (utan bearbetning) från ett analysprogram. De program som har en inbyggd analyserande del har tagit ett stort steg mot vad vi i kapitel 11 benämner "totalintegrerade system".

Om man tittar lite i någon av de programinventeringar som presenterades i kapitel 3, finner man att utbudet av program för dimensionering inte är särskilt stort i förhållande till det antal analysprogram som erbjuds. En av orsakerna kan vara att den forskning som bedrivits och som fortfarande bedrivs, i första hand inriktats på analys.

Man kan kanske finna en annan anledning i normsystemet som styr dimensioneringsarbetet. Alltefter som nya rön från forskningen för utvecklingen framåt, uppstår behov att revidera normerna. Eftersom dimensionering är normbunden innebär detta i sin tur att dimensioneringsprogrammen måste revideras parallellt med dessa. Naturligtvis kostar detta pengar och försämrar därmed programmets lönsamhet under deras "livslängd". Analysprogrammen drabbas inte lika mycket av detta eftersom de

inte är normberoende.

Ett annat problem som program för dimensionering brottas med är anpassningen till utländska normer. Det är nödvändigt för svenska företag att man kan använda sina befintliga dimensioneringsprogram även i arbetet med utlandprojekt. Därför är det en fördel om man lätt kan ändra i programmen så att de dimensionerar enligt de normer som beställaren föreskriver. Ett sätt att lösa detta problem ges i avsnitt 7.1.

Ytterligare en anledning till att det inte finns så många dimensioneringsprogram finner man i det faktum att det är svårt att göra dimensioneringsprogram. Svårigheterna finns på flera punkter.

Om man tittar på ett analysprogram finner man ofta i detta att man infört en rad förenklingar. En förenkling som görs är att man ersätter det verkliga tvärsnittet med en så kallad systemlinje i det verkliga tvärsnittets tyngdpunkt. För att ta hänsyn till konstruktionens verkliga utseende och verkningssätt ger man systemlinjen vissa egenskaper via parametrar som tröghetsmoment och elasticitetsmoduler samt att man i knutpunkterna antar att en viss grad av inspänning föreligger eller att anslutningen är friktionsfri.

Sådana förenklingar kan inte införas i ett dimensioneringsprogram, där arbetar man istället med "verkliga data" och därmed en mera komplex beräkningsmodell. Detta innebär vanligtvis inte några speciella svårigheter, för till exempel en balk med konstant tvärsnitt, men när man har besvärligare tvärsnittsutseenden uppstår problem. Hur skall man dra armeringen i en balk vid en plötslig förändring i tvärsnittet?

Den erfarna konstruktören använder sitt "sunda förnuft" och sina erfarenheter från tidigare liknande fall som han kommit i kontakt med. Han letar i minnet efter en liknande situation och kan utifrån sina tidigare erfarenheter dra slutsatser om hur det aktuella problemet kan lösas. Detta sätt att lösa problem har konstruktören skaffat sig genom lång yrkesutövning och består i en förmåga att tolka och överblicka en mängd "diffusa" data.

Att programmera in en sådan förmåga att lösa problem i en dator är mycket svårt, ofta omöjligt. Det blir programmerarens uppgift att finna en kompromiss som passar de fall som kan tänkas komma i fråga. Redan här stöter vi på ännu ett stort problem: Hur skall programmeraren veta vilka konstruktioner som programmet kommer att tillämpas på?

En annan svårighet ligger i att man under en dimensioneringsprocess gör en del ställningstaganden. Om en betongbalk inte klarar de belastningar den blir utsatt för väljer man att förändra någon parameter. Man kan kanske öka tvärsnittets höjd, öka mängden armering



eller välja en betong av högre kvalitet. Vilken man väljer att ändra på beror på konstruktörens egna överväganden. Dessa beslut måste också på något sätt klaras av i ett datorprogram. Hur dessa beslut skall styras är inte alls självklart, se avsnitt 11.2.4.

De svårigheter vi ovan pekat på har medfört att det finns ett relativt litet utbud av dimensioneringsprogram, men det har även inneburit att man i de program som finns ofta har begränsat problemen till relativt renodlade strukturer såsom balkar, pelare och plattor av speciell utformning.

Kapitlet behandlar fortsättningsvis krav som ställs på strukturer från olika håll och därefter lite speciella egenskaper för materialen betong, stål och trä som man måste beakta vid programmering. Att vi valt endast dessa tre beror bland annat på att utbudet för övriga material är ganska smalt och att det i vissa fall går att använda lätt modifierade program för att dimensionera vissa strukturer av andra material.

## 7.1 Krav på strukturer

Vi kommer att under respektive materialrubrik, i avsnitt 7.2, nämna lite om de normer som gäller för respektive material. I detta avsnitt skall vi försöka ge en överblick av normerna i stort och de förändringar som förväntas, samt hur detta kan påverka dimensionering med dator.

För närvarande har vi ett normpaket som består av den övergripande normen SBN 1980, kompletterad med ett antal speciella normer som gäller för betong, stål och aluminium.

Betongnormerna består av antingen,

-SBN 1980 tillsammans med B1-B11

eller

-SBN 1980 med avdelning 2A "Bärande konstruktioner" tillsammans med, Bestämmelser för Betong-Konstruktioner, BBK 79.

Dessa två normpaket får användas valfritt, men konsekvent, under en övergångsperiod. Konsekvent innebär att konstruktioner som samverkar i statistiskt hänseende, måste beräknas efter samma normer.

Normerna som reglerar dimensionering av stål är förutom SBN 1980 också StBK-N1 t o m N5 med StBK-K1 och StBK-K2.

Aluminiumkonstruktioner skall dimensioneras enligt SBN 1980, där kapitel 28 gäller speciellt för aluminium,

och StBK-N5 "Tunnplåtsnorm". Men dessutom gäller en del speciella föreskrifter,

- "Aluminiumkonstruktioner. Försöksnorm och kommentarer" (huvudnorm).
- "Aluminiumkonstruktioner. Stabilitetsproblem". (stabilitetsnorm)
- "Svetsade aluminiumkonstruktioner. Försöksnorm och kommentarer". (svetsnorm)

Det är denna flora av normer som man har för avsikt att ersätta med ett mera enhetligt normsystem, i vilket BBK-delen utgör ett steg. Inom en snar framtid kommer vi att få se en liknande norm gällande stålsidan, preliminärt benämnd BSK (Bestämmelser för Stål-Konstruktioner). Målet är en samlad norm för alla materialtillämpningar.

Tankegångarna bakom dessa nya normer är att man vill frångå de gamla normernas traditionella funktion som lärobok, handbok vid dimensionering och utförande och som regler för upphandling. Man vill ge konstruktören en frihet att själv komma med egna lösningar på aktuella problem. Dock gäller det för konstruktören att visa att den alternativa lösning som föreslås är bättre än den i normen angivna eller att den är tillräckligt bra med hänsyn till den aktuella situationen.

Motiven till att ändra normerna är flera. Några som kan nämnas är att,

- de äldre normerna är delvis föråldrade, rön från senare års forskning har införts i de nya normerna,
- det är önskvärt med enhetliga normer för olika typer av konstruktioner,
- man vill behandla dimensionering efter statistiska grunder, genom så kallade partialkoefficienter,
- man strävar mot en internationell samordning.

Givetvis finns fler skäl, men dessa har upplevts som de mest väsentliga.

Man har, med partialkoefficientmetoden, infört ett helt nytt sätt att behandla laster och materialparametrar. En annan stor förändring är att man övergår till att dimensionera i brottgränstillstånd med så kallade karakteristiska värden på laster och parametrar. Så kallade vanliga värden används därefter vid en funktionskontroll i bruksgränstillstånd.

Metoden är i stor utsträckning baserad på statistiska grunder. Detta kommer till uttryck i en mängd partialkoefficienter för olika lastfall och material. Koefficienterna finns föreskrivna i normen. Laster och mate-

rial behandlas därvid inte på samma sätt.

Vid beräkning av dimensionerande last utgår man från ett lastvärde, karakteristiskt eller vanligt beroende på aktuellt gränstillstånd och lastfall. Godtagna lastvärden för olika laster anges i normen. Det karakteristiska värdet för en variabel last, till exempel, antas till det värde som har sannolikheten 0.02 att överskridas en gång på ett år. Motsvarande vanliga värde bestäms med hänsyn till lastens relativa varaktighet under konstruktionens totala livslängd.

Aktuella laster på en konstruktion, som kan uppträda samtidigt, skall kombineras till lastkombinationer. Vid dessa olika kombinationer kan en och samma last erhålla olika partialkoefficienter beroende på lastfall och gränstillstånd. Lastkombinationerna består i brottgränstillstånd av,

- permanenta laster (multiplicerade med aktuell partialkoefficient),
- en (1) variabel last (multiplicerad med dennas aktuella partialkoefficient) och
- ett antal övriga variabla laster (multiplicerade med aktuell partialkoefficient).

För de två först nämnda lasterna skall karakteristiskt värde användas medan de senare skall uppgå till vanliga värden.

I bruksgränstillstånd har man endast permanenta och variabla laster med respektive vanliga värden, multiplicerade med partialkoefficienten lika med 1.0. I detta gränstillstånd utförs kontroll av besvärande deformationer, sprickor som kan ha betydelse för beständighet och utseende, etc.

Lastfallshanteringen enligt dessa principer kan i vissa fall ge upphov till ett mycket stort antal lastfall. Det kan även för en erfaren konstruktör vara svårt att genomskåda vilket av dessa som blir det dimensionerande, speciellt gäller detta om man utför beräkningarna enligt andra ordningens teori. En dator, som inte har konstruktörens förmåga att utesluta ofarliga lastkombinationer, måste beräkna konstruktionen för samtliga dessa fall, vilket kan medföra onödigt stora beräkningskostnader för ett i övrigt enkelt problem.

Materialparametrarna behandlas, som nämndes tidigare, på ett något annorlunda sätt. Dock har man, precis som i fallet laster, partialkoefficienter.

Koefficienternas värden styrs av hur pass noggrant man kan förutsäga en parameters värde. Till exempel är koefficienten för stål mindre än den för betong, vilket är naturligt eftersom betong har en större spridning av till exempel hållfasthetsvärden.

Den stora skillnaden ligger dock i att man tar hänsyn till olika säkerhetsklasser. Varje materialparameter divideras med den partialkoefficient som hör till den aktuella säkerhetsklassen. Dessa avspeglar hur pass allvarliga konsekvenserna antas bli vid skada på en konstruktion. Tre klasser förekommer, (med partialkoefficienterna 1.0, 1.1 respektive 1.2), där skadornas konsekvens betecknas mindre allvarlig, allvarlig respektive mycket allvarlig.

De nya dimensioneringsprinciperna medger ett högre materialutnyttjande än tidigare. Av denna anledning kan man förvänta sig att det kan bli aktuellt för beställaren att i högre grad än tidigare, specificera egna krav på, till exempel, tillåten sprickbildning och deformation i bruksgränstillståndet. Också dessa specificerade krav bör kunna infogas som indata till ett dimensioneringsprogram.

Denna genomgripande normändring har, ur datorberäkningssynpunkt, fått till följd att befintliga dimensioneringsprogram måste omarbetas. Förändringen gentemot det gamla normsystemet är så stor, att det i många fall bedömts vara lättare att göra helt nya program. Man har samtidigt fått möjlighet att rensa upp bland gamla program som med tiden, efter alla införda ändringar, kan ha blivit ganska snåriga och svåra att handskas med. Det kostar naturligtvis en hel del pengar och dessutom tar det tid att utveckla nya program.

Vi nämnde, som en av orsakerna till omarbetningen av normerna, att man eftersträvar en internationell samordning av normerna. Detta är naturligtvis värdefullt för de svenska företagen som i allt större utsträckning söker sig ut på den internationella marknaden. För att företagen skall kunna arbeta effektivt och konkurrenskraftigt krävs att man kan arbeta med sina befintliga datorprogram. Detta fordrar att man kan anpassa dem till de normer som gäller i beställarens land eller till en norm som godtages av denne.

Ett sätt att förbereda sina program för en sådan omställning är att modularisera dem. Detta innebär att programmet är uppbyggt av ett antal mer eller mindre fristående moduler, med en specifik uppgift. Till exempel generera indata, utföra diverse kontroller, behandla beställarens egna funktionskrav utöver normernas krav eller innehålla olika normsystem, se avsnitt 11.3.

På detta sätt kan man, med en välstrukturerad datalagring, uppnå även andra fördelar, till exempel:

- moduler kan lätt användas i flera program,
- lättare att hitta fel vid uttestning av program,
- mera överskådliga program, lättare för en utomstående att läsa och förstå programmets uppbyggnad och arbetsgång.



Genom ett antal företag har vi fått en del preliminära reaktioner på de nya normerna och anpassningen till datorberäkningar. Vi skall här sammanfatta i stora drag stämningarna vi mött.

-Möjligheten till högre utnyttjande av materialen inbjuder till "finräkning" vilket kan innebära att man sparar "mängder". Detta "finräknande" tar dock tid och vinsten kan ätas upp av ökad tidsåtgång. Med de sjunkande kostnader för datortid kommer detta att gynna utnyttjande av dator för dessa beräkningar.

-De flesta är överens om att de nya normerna är bättre anpassade till datorberäkningar än de gamla. Bland annat finns det numera oftare matematiska uttryck för angivna tabeller och kurvor. Dock anser man att det fortfarande återstår åtskilligt att göra på detta område.

Om man frågar efter något i normerna som man upplever som speciellt besvärligt får man ofta svaret,

-lastfallshantering. Från vissa håll framförs kritik mot lastfallshantering som sådan. Man anser att den saknar förankring i verkliga situationer. De flesta anser att den inbjuder till datorbehandling, på grund av den mångfald lastfall den ger upphov till. Man konstaterar dock samtidigt att även om man tar datorer till hjälp kan det uppstå problem.

Detta är några av de synpunkter vi mött bland företagen. Vi har inte möjlighet att här diskutera detta intressanta och för framtiden viktiga förhållande mellan normer och datorer, men det är ett synnerligen angeläget ämne för vidare forskning.

## 7.2 Materialanpassade program

Varje avsnitt behandlar de vanligaste förekommande strukturerna och lite om de speciella villkor som gäller för dessa. Därefter kommenteras det programutbud som finns och till sist lite om de normer som styr dimensioneringsförfarandet för aktuellt material.

I varje avsnitt ges ett par exempel på program som förekommer på marknaden. Det skall dock poängteras redan här, att valet av just dessa program inte beror på att de är bättre eller sämre än andra liknande program. Programmen är inte heller unika i sitt slag, det förekommer liknande program med motsvarande funktioner även hos andra företag än de som nämns i denna framställning. Urvalet gör heller inte anspråk på att vara representativt för programutbudet på marknaden i stort, utan skall således enbart ses som ett mer eller mindre slumpmässigt urval.



Program för "miniräknare" har vi medvetet hoppat över. Vi nöjer oss med att konstatera att sådana program finns, för till exempel tvärsnittsdimensionering och dimensionering av kontinuerliga balkar. Programmen är vanligtvis mycket hårt styrda vad avser antal laster, lasternas placering, möjlighet att definiera randvillkor och dylikt, på grund av räknarnas begränsade minnesutrymme.

### 7.2.1 Betong

Betong har en del speciella egenskaper som måste beaktas vid dimensionering av betongkonstruktioner. Inledningen av avsnittet skall beröra några av dessa och hur de påverkar dimensioneringsprogrammets utseende.

Betong är ett mycket formbart material. Man kan med formsättning ge den i princip godtycklig tvärsnittsform, innebärande att antalet parametrar som bestämmer tvärsnittets egenskaper blir stort. Ju fler parametrar man får att hålla reda på, desto svårare blir det att finna den, ur någon synpunkt, optimala utformningen av tvärsnittet. Detta gäller kanske speciellt för spännbetong.

Ytterligare en egenskap som trasslar till det är att betongen spricker vid dragpåkänningar. En strukturs verkningssätt kan komma att helt förändras beroende på hur uppsprickningen sker. Mera om detta skall vi diskutera sist i avsnittet, i samband med att vi behandlar de normer som reglerar dimensionering av betong.

Betong är ett inhomogent material bestående av ballast som med cementlim sammanfogats till att samverka med armeringsstålet. De olika materialen som ingår har mycket olika egenskaper när det gäller förmåga att ta tryck- och dragspänningar. Syftet med dimensioneringen är att kombinera dessa egenskaper så att de krav som ställs uppfylls med avseende på hållfasthet, styvhet, beständighet, med mera.

En annan variabel som måste beaktas är att betongens egenskaper är tidsberoende. Hållfasthetstillväxten, till exempel, är till en början mycket snabb, för att senare avta. Vid en viss bestämd tidpunkt antar man att den slutliga hållfastheten uppnåtts. Dock vet man att tillväxten fortsätter, om än mycket långsamt.

Ett annat tidsberoende fenomen, som uppstår vid långtidslast, är krypning. Denna innebär att det uppstår tidsberoende deformationer i betongen. Härvid kommer en omfördelning av moment och tvärkrafter att ske. Detta innebär, att de beräkningsförutsättningar man antog vid dimensioneringen av strukturen inte längre stämmer.

Eftersom krypningen uppträder under en lång tidsrymd, är det svårt att exakt förutse dess effekter på kon-

struktionens funktion. Även betongens krympning ger upphov till liknande problem. Speciellt vid beräkningar enligt icke-linjära teorier ställer dessa två egen-skaper till stora problem.

Betongens benägenhet att spricka vid dragpåkänningar kan man, helt eller delvis, undvika genom att använda spännarmering. Man undviker dock inte alla problem med detta, man får istället andra fenomen att beakta såsom den ovan nämnda krympningen i betongen samt dessutom relaxation i stålet, tvångsmoment på grund av excentrisk placering av armeringen, etc.

I princip kan man säga att man vid dimensionering av betongkonstruktioner har att kombinera betongkonglomerat och armeringsstål så att de krav som ställs i normerna uppfylls. Dessa krav kan uppdelas i,

1. krav på säkerhet mot brott, t ex
  - materialbrott (böj-, skjuv-, förankringsbrott, etc)
  - instabilitetsbrott
2. krav på funktion vid normal användning, t ex
  - begränsning av nedböjning
  - begränsning av sprickbredd
3. krav på beständighet, t ex
  - skydd mot armeringskorrosion
  - frostbeständighet

Men även från andra håll ställs det krav. Till exempel från beställaren som har sina speciella fordringar på den färdiga konstruktionens funktion. Se avsnitt 7.1.

Med betong kan man forma konstruktioner av vitt skilda slag. Från de enklaste balkarna till komplicerade flervåningsramar och skalkonstruktioner. Medan man relativt enkelt kan avgränsa en fritt upplagd balk från den övriga konstruktionen, definiera randvillkor och beräkna den för sig, är det betydligt svårare att göra motsvarande med t ex en flervåningsram med elastiska inspänningar.

Ovan har vi påpekat en del aspekter, där betongens beteende avviker från de ideala beräkningsteorier som man normalt använder. Dessa problem har återspeglat sig i det utbud av datorprogram som erbjuds på marknaden. De flesta beräknar relativt väldefinierade strukturer med relativt låsta utseenden.

Dimensioneringsprogram för renodlade betongtvärsnitt var den typ av program som man först började utveckla. Med dessa program gjorde man beräkningar för de intressanta snitten, ett i taget. De första av dessa kunde inte utföra någon analys av strukturerna. Användaren fick således skaffa fram de aktuella snittpåkänningarna på annat sätt, på det traditionella sättet (handberäkningar) eller kanske med ett datorprogram för analys. Därefter ges snittkrafterna som indata till dimen-

sioneringsprogrammet. Det skall dock poängteras att denna typ av program inte på något vis tillhör det förgångna och övergivits. Tvärtom, de kommer till stor användning ännu idag. Man utvecklar fortfarande program av denna typ med liknande grundstruktur.

En fördel med denna typ av program är att man under i princip hela dimensioneringsprocessen har kontroll över beräkningarna genom att man måste hämta indata från analysprogrammet och överföra dem till dimensioneringsprogrammet. Detta gör det lättare att upptäcka eventuella felaktigheter i beräkningarna. Dessutom är programstrukturen relativt enkel vilket vanligen betyder att riskerna för att fel i själva programmet skall förekomma är mindre än för ett mera avancerat program.

"SEKTION" är ett dimensionerande datorprogram som inte kan beräkna snittkrafter utifrån pålagda yttre laster. Däremot kan man utföra en viss analys av tvärsnitt, till exempel om man ger töjningsfördelningen över tvärsnittet erhålles motsvarande spänningsfördelning eller om man ger normalkraft och moment i x- respektive y-riktningen erhåller man resulterande spännings och töjningsfördelningar. Programmet kan förutom betong även dimensionera stål- och aluminiumtvärsnitt.

Programmet är utvecklat vid VBB men finns även tillgängligt via TeknikData AB.

Tillvägagångssättet vid körning är att man definierar en utbredd area med hjälp av ett godtyckligt antal hörnpunkter. Med denna utbredda area kan man definiera ett tvärsnitt som i princip kan vara hur oregelbundet som helst. I FIG 7.1 visas exempel på tvärsnitt som kan beräknas. Om man så önskar, kan man även definiera ett antal s k punktareor, som kan symbolisera armeringen i betong. Också antalet punktareor kan väljas godtyckligt. För varje definierad area definieras spännings-töjningsdiagram. Genom att definiera olika diagram för olika punktareor kan man simulera både slak och spänd armering i samma tvärsnitt. Vissa standard diagram kan anropas med ett antal enkla kommandon, exempelvis för betong i brottstadiet enligt BBK 79.

Några inbyggda begränsningar med avseende på antal definierade areor eller spännings-töjningsdiagram finns inte. Problemets storlek bestäms enbart av den tillgängliga maskinkapaciteten.

Resultaten erhålles på datablad där givna indata anges och därefter de beräknade storheterna såsom spänning, töjning eller punktareornas erforderliga storlek. Programmet arbetar i fria enheter vilket innebär att användaren själv bestämmer de sorter som skall användas vid beräkningen.

"SEKTION" är ett mycket generellt program. Det är möjligt att beräkna i stort sett godtyckliga tvärsnitt. Detta har både för- och nackdelar. Ju mera generellt

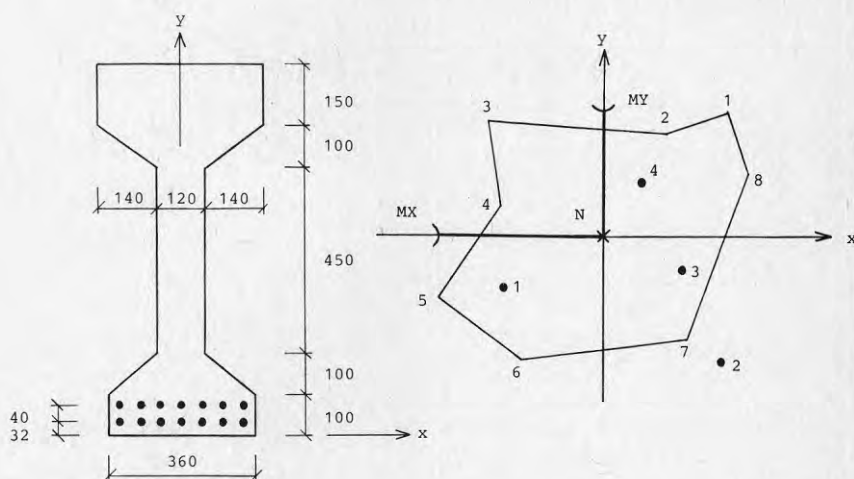


FIG 7.1 Tvärsnitt för beräkning med "SEKTION"

ett program blir, desto större är risken för att programmet används på ett sätt som programmeraren inte avsett. Därmed kan fel uppstå vid körning som inte uppmärksammas av programmets kontrollsystem.

Efter dessa "rena" dimensioneringsprogram, har man utvecklats allt fler kombinerade analys- och dimensioneringsprogram. Filosofin bakom är att man endast skall ge de på strukturen yttre verkande lasterna och uppgifter om de ingående materialens egenskaper och geometri, därefter skall programmet själv analysera och dimensionera konstruktionen. Två typer av sådana program förekommer.

Vid den ena typen av program kopplar man en dimensionerande programrutin direkt till ett analysprogram. Data från analysdelen överförs direkt via parametrar, enbart definierade i programmen, till den dimensionerande delen. Man kan då inte använda programdelarna var för sig.

I den andra typen, en enkel form av programmodulsystem, finns det möjlighet att välja ifall man vill ha en fullständig dimensionering eller om man enbart vill använda analysdelen av programmet. Detta är en fördel ifall man har en "färdig" konstruktion som man vill kontrollberäkna. Man spar på så sätt datortid genom att man slipper köra igenom dimensioneringsdelen av programmet.

Det är i allmänhet en stor fördel om programmen är mo-

duluppbyggda. Man gör på så sätt programmet mera flexibelt, dessutom kan man skaffa sig kontrollpunkter, där användaren kan kontrollera vissa mellanresultat. Modulariserade program och kontrollpunkter diskuteras mera ingående i kapitel 11.

"BETONGBALK" är ett program med en analyserande programfunktion som kan utnyttjas skilt från dimensionerings delen. Programmet har utvecklats och ägs av AB Jacobson&Widmark, J&W. Det ingår som en del av beräknings- och ritsystemet "BERIT". Programmet utför statisk och nedböjningsberäkning samt dimensionering för fritt upplagda och kontinuerliga strimlor och balkar.

Programmet är alltså ett analyserande program vilket gör det möjligt att som indata ge de yttre verkande lasterna, som kan vara relativt godtyckliga och utgöras, (enligt gamla normens benämningar), av både ständig och rörlig last. Genom att man på indatablanketterna ger en variabel DIM värdena 0, 1 eller 2 utför programmet enbart en statisk beräkning, en dimensionering för moment eller en dimensionering både med

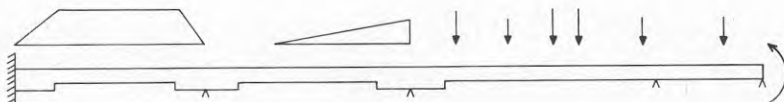


FIG 7.2 Datorprogrammet "BETONGBALK"

ARBETS-NR	KÖNTOR	SIGN	TEXT (MAX 56 TECKEN)

FALT	LFL	STÖDV	STÖDH	SPEC
ANTAL				0

L(1)	L(2)	L(3)	L(4)	L(5)	L(6)	L(7)	L(8)	L(9)
M	M	M	M	M	M	M	M	M

EI(1)	EI(2)	EI(3)	EI(4)	EI(5)	EI(6)	EI(7)	EI(8)	EI(9)
0	0	0	0	0	0	0	0	0

UTBREDDA LASTER VILANDE								
PU(1)	PU(2)	PU(3)	PU(4)	PU(5)	PU(6)	PU(7)	PU(8)	PU(9)
KN/M	KN/M	KN/M	KN/M	KN/M	KN/M	KN/M	KN/M	KN/M

UTBREDDA LASTER RÖRLIGA (OM LFL > 1)								
PU(1)	PU(2)	PU(3)	PU(4)	PU(5)	PU(6)	PU(7)	PU(8)	PU(9)
KN/M	KN/M	KN/M	KN/M	KN/M	KN/M	KN/M	KN/M	KN/M

FÖRKLARINGAR
*****
FALT: ANTA FALT, MAX 9 ST.
LFL : 1=ENDAST VILANDE LAST, ETT LASTFALL. 2=STÖDMÖMENT OCH TVÄRKRAFT BERÄKNAS MED FULL LAST. FALTMÖMENT BERÄKNAS MED ÖGYNNSAM PLACERING AV RÖRLIG LAST. 3=FALTMÖMENT, STÖDMÖMENT OCH TVÄRKRAFT BERÄKNAS MED ÖGYNNSAM PLACERING AV RÖRLIG LAST.

FIG 7.3 Indatablankett till programmet "BETONGBALK"



hänsyn till moment och tvärkraft. Denna facilitet gör programmet mera flexibelt och kan således användas både som analys- och dimensioneringsprogram.

Man har inte reviderat programmet så att det arbetar med de nya normerna utan man dimensionerar enligt B6 och B7 1968. Arbetet med uppdateringen pågår och beräknas vara klar inom en snar framtid.

Indata till programmet ges på indatablanketter, se FIG 7.3. Dessa beskrivs i en mycket utförlig programbeskrivning under punkten "Anvisningar för indata". Där framgår att indatablanketterna är fyra till antalet. I den enklaste formen anges endast spännvidder, balkstyheter och utbredda laster. Om man sedan vill, kan man ange så kallade specialuppgifter om voter, lastgrupper, speciallaster och dimensionering av betong.

I "Anvisningar för indata" går man igenom de variabler som används och förklarar dess innebörd. Programbeskrivningen består dessutom av en del som beskriver de beräkningsmetoder och förutsättningar som används i programmet. Detta är av stort värde för användaren som skall tolka de resultat som erhålles vid en körning.

På den sista indatablanketten "Dimensioneringsuppgifter" ges indata som gäller för de ingående materialen och de aktuella tvärsnitten som förekommer vid beräkningen. Diameter för böj- och skjuvarmering, tillåtna påkänningar för materialen och uppgifter om sprickvidd anges. Alternativt kan man låta programmet beräkna sprickvidden.

Förutom den sedvanliga utskriften av resultaten kan man även erhålla en grafisk resultatpresentation. Det är ritsystemet BERIT som ombesörjer detta. Också denna facilitet är värdefull vid en kontroll av en körning. Det är avsevärt lättare att kontrollera en "bild" istället för en mängd numeriska utskrifter.

De normer som styr dimensionering av betong har på senare tid omarbetats. Det gamla, utgående normpaketet, består av SBN 1980 tillsammans med B1-B11. Den nya betongnormen består förutom av BBK 79 och SBN 1980 också av ett komplement till den senare, avdelning 2A, "Bärande konstruktioner". Avdelningen innehåller,

- Allmänna bestämmelser för bärande konstruktioner
- Lastvärden
- Betongkonstruktioner
- Stålkonstruktioner

Detta har dock redan ingående behandlats i avsnitt 7.1.

Istället skall vi som sista punkt i detta avsnitt peka på ett fenomen som är specifikt för betong, nämligen uppsprickning. Denna regleras i de nya normerna, genom att man angivit tillåtna spricksäkerheter och sprickbredder för olika armeringsaggressiva miljöer, vid lång-

tidslast i farligaste lastställning i bruksgränstillstånd. För obetydligt armeringsaggressiv miljö har man inte angivit någon begränsning alls. Begränsningar under dessa förhållanden faller helt på användaren av konstruktionen att definiera.

Ett annat fenomen som har samma ursprung är nedsättning av styvheten på grund av uppsläende sprickor. De nya normerna tillåter ett högre utnyttjande av materialet, vilket i sin tur kommer att innebära att större risk för sprickbildning föreligger i bruksgränstillståndet. I normerna anger man uttryck för beräkning av styvheterna i osprucket respektive sprucket stadium. Däremot är det upp till den enskilde konstruktören att bedöma ifall uppsprickningen påverkar momentfördelningen i så stor utsträckning, att en ny beräkning av snittkrafterna är nödvändig. Man vet dock att även en begränsad sprickbildning i betongen ger upphov till en nedsättning av styvheten i konstruktionen och därmed en omfördelning av momenten.

I princip borde man, om man i en första analys av strukturen utfört beräkningarna enligt elasticitetsteori och antagit osprucket tillstånd, utföra en ny beräkning med antagande om viss uppsprickning. Om man skall vara riktigt noggrann, borde detta upprepas tills dess att man erhåller acceptabelt små avvikelser i fördelningen av momenten mellan två på varandra följande beräkningar. Se även avsnitt 6.1.5.

### 7.2.2 Stål

Stålet är, jämfört med betong, ett homogent material. Det tillverkas uteslutande under industrimässiga förhållanden i olika standardiserade hållfasthetsklasser vars egenskaper noggrant specificeras i normerna.

Av detta standardiserade stål svetsar eller valsar man profiler av olika typer. IPE, HEA, HSI är några av dessa. De flesta av dem lagertillverkas och finns att beställa ur tillverkarnas orderkataloger. Detta innebär att man i stor utsträckning är hänvisad till de sektionsutformningar som saluförs av tillverkarna. Dock förekommer det att man "måttbeställer", detta är vanligt när det handlar om extremt höga balkar t ex för balkar till vägbroar.

Den största fördelen med stål är dess relativt låga vikt i förhållande till dess bärförmåga. Denna egenskap utnyttjas i smäckra konstruktioner med en stor bärförmåga och för att nedbringa den bärande stommens egenvikt i extremt höga konstruktioner.

En av nackdelarna med smäckra konstruktioner är att det ofta inte längre är sträckgränsen som blir den dimensionerande storheten i beräkningarna. I stället framträder konstruktionens stabilitet som det viktigas-

te dimensioneringskriteriet. Detta innebär för det mesta inte enbart en större volym beräkningar utan dessutom, i de allra flesta fallen, mera komplicerade sådana. Anledningen till detta är naturligtvis att man inte längre kan försumma deformationernas inverkan på momentfördelningen i konstruktionen, utan man måste använda andra ordningens teori, jämför avsnitt 6.1.6.

En egenskap som är viktig att uppmärksamma är stålets känslighet för brandpåverkan. Som bekant förlorar det snabbt sin bärförmåga vid sådana temperaturer som är vanliga vid ordinära bränder. Särskilda åtgärder krävs för att begränsa dessa effekter. Utveckling av datorprogram som tar hänsyn till denna negativa inverkan pågår vid Lunds Tekniska Högskola, se Peterson, A: "Material Modelling of Structural Steel at High Temperature", LTH, Lund 1981.

Den vanligaste användningen av stål i byggnadskonstruktioner är att man med hjälp av balkar och pelare, tagna ur tillverkarnas tabeller, bygger upp en rymdstruktur. Men det förekommer naturligtvis även andra tillämpningar, till exempel olika former av skalkonstruktioner. En mycket vanlig sådan är stålskorstenar, ett exempel på ett datorprogram för dimensionering av just dessa skall ges senare i detta avsnitt. På senare tid har man även börjat använda tunnplåt allt mer i bärande funktioner. Ett rikligt sortiment av profilerad tunnplåt förekommer på marknaden.

Dimensionering av balk- och pelarstrukturer sker traditionellt "för hand", med användande av gängse statistiska beräkningsmetoder för att beräkna de dimensionerande momenten och krafterna. Därefter går man in i fabrikanternas balktabeller och väljer en balk som klarar de aktuella lasterna. Många av de idag förekommande programmen för ståldimensionering ersätter det sista steget i denna process, nämligen valet av profil och kontrollberäkning av att den klarar de aktuella lasterna och i övrigt passar in i konstruktionen, t ex med avseende på tillåten konstruktionshöjd. De förutsätter att användaren på annat sätt beräknat de dimensionerande lasterna, för hand eller med ett analyserande datorprogram.

Program som utför en fullständig dimensionering från analys till beräkning av de erforderliga dimensionerna är relativt ovanliga. De flesta programmen är av interaktiv natur, användaren får själv mata in dimensionerande snittkrafter och därefter låta programmet prova sig fram till en lämplig balk eller pelare.

Det finns på marknaden en del specialprogram, till exempel för dimensionering av svetsade ramhörn, men dessa uppfyller vanligen inte de villkor som anges i vår definition av dimensioneringsprogram. Utbudet av dimensioneringsprogram för stålkonstruktioner är, jämfört med motsvarande utbud för betong, relativt litet.

En anledning till detta kan vara, att det är förhållandevis mindre komplicerat att dimensionera ett ståltvärnsnitt än till exempel ett i betong. För stål finns dimensioneringstabeller där man kan gå in med sina aktuella snittkrafter och välja en balk eller pelare som klarar dessa. Därför är behovet inte lika stort av speciella dimensioneringsprogram. Men detta gäller inte vid beräkning av konstruktionens totala stabilitet, då blir beräkningarna betydligt mera komplicerade. Därvid kan man ha stor nytta av ett datorprogram för beräkning av konstruktionens totala stabilitet. Sådana program hör dock till kategorin analysprogram och behandlas i kapitel 6.

Vi presenterade ett program för dimensionering av betongtjärnsnitt, "SEKTION", i det föregående avsnittet. Detta program kan man, som vi nämnde redan då, använda för att dimensionera ståltjärnsnitt. Enda skillnaden blir att den så kallade utbredda arean representerar stål i stället för betong. Istället för en arbetskurva för betong ger man en som gäller för den stålsort man önskar använda i konstruktionen. I övrigt arbetar programmet på samma sätt som för dimensionering av betongtjärnsnitt.

"SKORSTEN" är ett datorprogram för dimensionering av skorstenar och liknande rotationssymmetriska stålkonstruktioner. Programmet är utvecklat av Stålbyggnadskontroll AB och finns allmänt tillgängligt för körning via terminal vid TeknikData AB.

Programmet dimensionerar med avseende på statisk håll-

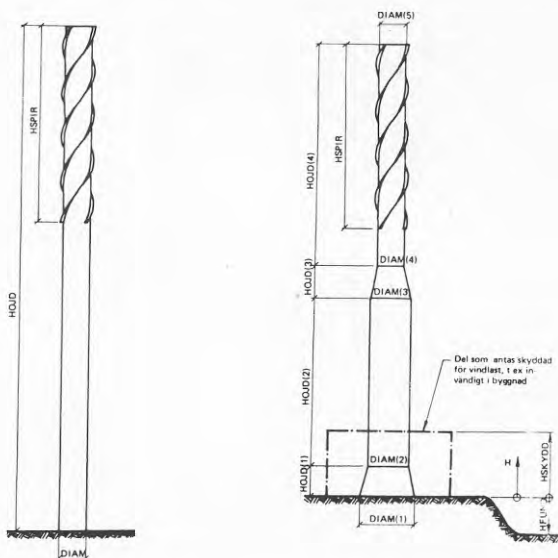


FIG 7.4 Datorprogrammet "SKORSTEN"



fasthet, utmattningshållfasthet och buckling av mantelplåt. Erforderliga dimensioner på manteltjocklekar beräknas. Även förankringen i fundamentet kan dimensioneras, dock inte själva fundamentet. Det finns också möjlighet att utnyttja programmet för kontroll av en given konstruktion.

Standardvärden för materialhållfastheter, terrängtyper, svetsklasser och tillåtna spänningar finns inlagda i programmet. Dessutom kan dimensionerande spänningskollektiv beräknas av programmet utgående från vindstatistik. Man behöver i princip bara ge höjden som indata för att programmet skall ge ett beräkningsresultat. Samtliga värden utom konstruktionens höjd är inlagda som standardvärden i programmet. Dessa värden kan dock lätt ändras med hjälp av indata.

Indata ges med s k NAMELIST-förfarande. Detta innebär att man namnger de variabler som skall ges värden som avviker från de inlagda standardvärdena. Efter variabelnamnet ger man det värde som önskas på aktuell variabel. På detta sätt bygger man upp en indatalista. Denna kan sparas mellan olika körningar och utnyttjas efter revidering vid nästa körning om man inte erhöll ett önskat resultat vid den första.

Som utdata från programmet erhåller man en fullständig utskrift av beräkningsresultaten, se FIG 10.11. Utskriften är disponerad så att man skall kunna läsa och granska beräkningen utan annat underlag.

Resultatutskriften redovisar bl a aktuella beräkningsförutsättningar, givna och/eller beräknade geometriska data för konstruktionen, beräkningsparametrar, egenfrekvenser, spänningar för statisk last, spänningsvidder för utmattningslast samt materialutnyttjande i dimensionerande punkter i grundmaterialet och svetsar för förekommande lastfall. Dessutom kan man om man så önskar få en stålspecifikation där stålvikter, dimensioner för mantelplåtar, plåtutbredning vid varierande manteldiametrar, materialkvaliteter, dimensioner och plåtutbredning för vindspiraler med mera, anges.

Vi nämnde tunnplåt som ett material som används allt flitigare i byggnadskonstruktioner. Detta ökade intresse för materialet har bland annat fått som följd att man utvecklat allt fler program för dimensionering av tunnplåtskonstruktioner. Som ett exempel på ett sådant program har vi valt ett som utvecklats av Tekn dr Arne Johnson Ingenjörbyrå AB. Datorsystemet heter "TRP200". Systemet är ett hjälpmedel vid projektering av PLANNJA AB's profil TRP200, dels vid offertskede dels vid slutlig projektering.

Beräkningarna kan utföras enligt två metoder, nämligen enligt metoden för tillåtna påkänningar och enligt partialkoefficientmetoden. Dessutom kan man anpassa beräkningarna till utländska marknader med avseende på dimensioneringsregler och tillåtna påkänningar. Hur den-



na anpassning kan göras beskrivs i användarinstruktionen. Bland annat använder man sig av olika variabelnamn och partialkoefficienter i olika länders normsystem.

TRP200 består av tre fristående datorprogram. Dels ett konverterande program som man använder för att skapa den nödvändiga indatafilen. Dels ett huvudprogram som utifrån indatafilens uppgifter specificerar de ingående komponenterna och utför alla erforderliga konstruktionsberäkningar. Programmet producerar även en indatafil till det tredje, avslutande plottprogrammet, som plottar montageplaner samt sektioner för montage av uppslagsplåtar.

I programmet finns en omfattande felkontroll inbyggd. Denna kontrollerar att man följt de förutsättningar som programmet arbetar efter och om man erhållit värden på den osäkra sidan eller på annat sätt fått ohållbara resultat. Körningen avbryts om maximalt antal fel överskrids eller om särskilt allvarliga fel begås. Den feldiagnos som programmet utfört under beräkningens gång lagras på utdatafilerna.

Huvudprogrammet utför först av allt en kontroll av indata, därefter skrivs de förutsättningar ut som gäller för aktuell beräkning. Nästa steg blir att utföra en skivverkansberäkning för de angivna skivorna och en utplacering av plåtarna. Efter detta utförs för varje definierad zon en dimensionering av TRP-plåten, eventuella håltagningar, uppslagsdimensionering, skarvning av plåtar och skarvning av kantbalk-huvudbalk, vindpelare. Sist utförs skarvning av kantbalkar, specificering av täckplåtar och avväxlingsbalkar.

Resultaten från körningen lagras på datafiler som efter behov kan skrivas ut i klartext. Om man vill ha en plottad bild av resultatet anropar man ritprogrammet som, med den i huvudprogrammet skapade indatafilen, ritar upp den beräknade konstruktionen.

Även normerna som styr dimensionering i stål kommer att revideras. De beräknas utkomma inom en snar framtid och går under benämningen BSK, Bestämmelser för Stål-Konstruktioner. Dessa kommer att genomsyras av samma grundläggande tankar som BBK 79. Till exempel kommer den att bygga på partialkoefficientmetoden och de lastbestämmelser som används tillsammans med BBK 79. Man kommer över huvud taget att eftersträva en så stor överensstämmelse normerna emellan som möjligt.

Men till dess de nya normerna är klara gäller fortfarande de gamla, nämligen SBN 1980 tillsammans med,

-StBK-N1 Stålbyggnadsnorm 70. Utgåva 2

-StBK-N2 Byggsvetsnorm

-StBK-N3 Skruvförbandsnorm 76

-StBK-N4 Rostskyddsnorm

-StBK-N5 Norm för tunnplåtskonstruktioner 79.

Dessutom förekommer det ett par speciella metoder som godtas under vissa förutsättningar.

-StBK-K1, "Gränslasthandbok", dimensionering av stålkonstruktioner med gränslastmetod.

-StBK-K2, "Knäckning, vippning, buckling", godtas under förutsättning av föreskriven tillverkningskontroll.

### 7.2.3 Trä

Program för dimensionering av träkonstruktioner har principiella likheter med sådana som dimensionerar konstruktioner av stål. Stålprogrammen väljer vanligen mellan olika standardprofiler (HEA, IPE, etc). På samma sätt använder träprogrammen bland sågade standarddimensioner av klassificerat konstruktionsvirke.

Trä är ett naturmaterial som formas av den miljö var i det växer. För att få ett grepp om dess egenskaper har man infört ett klassificeringssystem för konstruktionsvirke, T-klasser. Varje klass har definierats med avseende på vissa minimikrav vad gäller till exempel hållfasthet. Fyra klasser förekommer: T30, T24, T18 och Ö-virke. Indelning i dessa klasser sker efter visuell eller maskinell besiktning.

Det är detta virke som står till buds när man dimensionerar i trä. Som nämnts tidigare sågas virket i standarddimensioner. Att beställa specialsågat virke är sällan en lönsam affär.

Man kan urskilja två huvudtyper av träkonstruktioner som behandlas helt olika i produktionsprocessen. Dels har vi det vanliga sågade virket som skarvas och spikas ihop till strukturer sammansatta av stångelement, såsom regelstommar till småhus, ytter- och innerväggs-element i flervåningshus, takstolar, etc. Dels har vi limträ, bestående av flera (minst fyra) hoplimmade bräder (lameller) med fibrerna i strukturens längsriktning, som används i balkar, bågar, ramar.

De förstnämnda strukturerna byggs ofta upp direkt på byggarbetsplatsen av lösvirke. Dessa strukturers utformning bestäms ofta av andra villkor än statiska. Till exempel isoleringens tjocklek eller andra skäl som har med utförandet att göra. Ett undantag från detta utgör takstolarna. Dessa kan med fördel dimensioneras och tillverkas under industrimässiga förhållanden, för att därefter transporteras till byggarbetsplatsen. Det finns på marknaden ett antal datorprogram som dimensionerar sådana takstolar. Vi återkommer till ett

av dessa senare i detta avsnitt.

Limträkonstruktioner tillverkas uteslutande industriellt. Hela eller i delar, transporteras de till byggsplatsen för att där monteras in på sin avsedda plats. Limträ indelas på motsvarande sätt som lösvirke i konstruktionsklasser, L50, L40, L30 och L20.

Lamelluppbbyggnaden gör det möjligt att forma strukturen till önskat utseende. En ram till exempel, kan utformas på traditionellt vis med någon form av skarvning i hörnen (t ex fingerskarvning) men också kontinuerligt. Det finns således stora möjligheter att påverka det statiska verkningssättet.

I Sverige har vi endast ett fåtal tillverkare av limträ, en av dessa är Töreboda Limträ AB. De använder sig av dator för dimensionering av sina konstruktioner. Man har låtit utveckla program för dimensionering av fyra strukturtyper,

- balkar:       sadel-, jämnhög- och pulpetbalk
- bågar:       generell-, parabel-, cirkel-,  
              ellips- och sinusbåge
- ramar:       med krökt eller fingerskarvat ramhörn
- takstolar:   dragbandstakstol med dragband av stål.

Man har strävat efter att göra genereringen av indata så enkel som möjlig. Nyttig last, snö- och vindlast, egenvikter, c-avstånd, spännvidd och eventuellt taklutning är de indata typer som man arbetar med.

Om man kör ett balkprogram får man i första körningen, efter en grovdimensionering, en tabell över preliminära förslag till olika balkar som uppfyller dimensioneringsvillkoren. Ur denna tabell får sedan konstruktören själv välja ut den balk som, med hänsyn till den aktuella konstruktionen, uppfyller de statiska och funktionella kraven bäst. Därefter gör man en förnyad körning med indata som gäller för den valda balken och får på så sätt en slutlig dimensionering av denna.

Man har tyvärr inte någon möjlighet att erhålla grafisk utmatning från något av sina program. Detta upplever man som en brist och man har på senare tid försökt skaffa sig tillgång till sådana möjligheter. En anledning till att man inte redan har ritmöjligheter är svårigheter att rita de olika detaljlösningarna som man har. De kan utformas olika för liknande strukturer beroende på den aktuella byggnadens utseende.

Vi nämnde tidigare att det finns program på marknaden för dimensionering av takstolar uppbbyggda av lösvirke. Bjerking Ingenjörbyrå AB och Nordisk Kartro har gemensamt utvecklat ett dimensioneringssystem för sådana takstolar. Man har ett program för generella takstolar,

men också program för speciella takstolar såsom W-, WW-, WWW-takstolar, 1 ½-planstakstolar, ramverkstakstolar samt låga sadel- och pulpettakstolar.

Samtliga program är typgodkända och till vissa finns även av Planverket godkänd tillverkningskontroll. Detta innebär att själva datorberäkningen i princip inte skall behöva granskas, men väl de indata den bygger på. Takstolar beräknade med typgodkänt program skall märkas med tillverkare och typgodkännandenummer samt skall kunna identifieras genom objektets eller ritningens nummer. Om godkänd tillverkningskontroll är utförd behöver inte annan kontroll utföras på byggarbetsplatsen än identifikation och undersökning av eventuella transportskador. I avsnitt 10.4.1 behandlas förfarandet vid typgodkännade av datorprogram utförligare.

Användaren definierar på standardiserade indatablanketter den takstol som avses och de förutsättningar som gäller för dimensioneringen. (Takstolstyp, geometriska data, laster, virkeskvaliteter, maximi- respektive minimidimensioner, etc).

Som utdata erhålles dels ett tillverkningsunderlag, dels ett granskningsunderlag med tillhörande ritningar. Presentationen av resultaten är likartad för alla program.

Tillverkningsunderlaget består av kaplistor, spikplåtsspecifikationer, dimensioner, materialklasser, sågvinklar, etc. Detta underlag är till stor hjälp vid tillverkningen av takstolarna och sägs dessutom bidra till att minska spillet.

Först i utskriften av granskningsunderlaget erhålles samtliga av användaren givna indata samt de standardvärden som programmet valt. Det är dessa som ligger till grund för aktuell beräkning. Därefter följer ett par sidor med redovisning av virkesdimensioner, uppslagsreaktioner, maximal nedböjning, spikplåtar, skarvzoner och en fullständig redovisning av knutpunkternas nedböjningar.

Ritningarna består av översiktsritning, detaljritningar och kaplista, se FIG 7.5. Man kan antingen föreskriva en skala på översiktsritningen eller låta programmet välja den största möjliga, så att takstolen precis får plats. Det senare kan ibland bli förvirrande om skalan råkar bli "udda", till exempel 1:35. Detaljerna och kaplistan ritas i skala 1:20.

Programmen arbetar så att om de av användaren valda dimensionerna inte är tillräckliga, väljer det själv en grövre dimension och gör därefter (med nya förutsättningar) en ny beräkning med kontroll av dimensionerna. Om alla dimensioneringsvillkor enligt Svensk Byggnorm är uppfyllda avslutas beräkningen, annars upprepas förfarandet tills riktiga dimensioner genererats. Det finns möjlighet att låsa vissa dimensioner som av

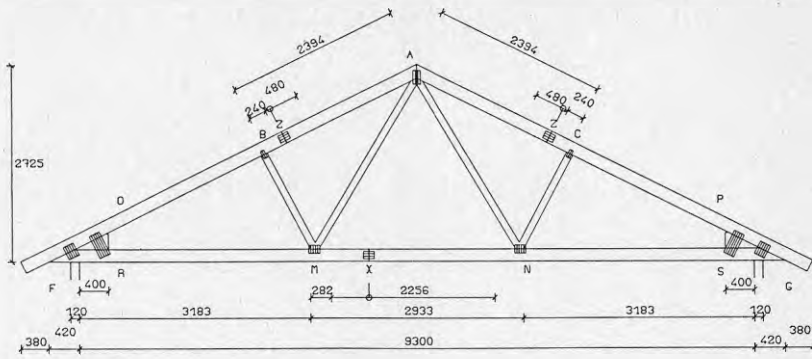
speciella skäl inte får ändras, programmet tar då hänsyn till detta vid beräkningarna.

Huvuddelen av programmen arbetar efter SBN 1980, kapitel 27 "Träkonstruktioner", och de handlingar som åberopas i typgodkännadebevisen för programmen, till exempel Statens Planverks Godkännanderegler nr 4 (1974), "Spikplåtsförband". För mera speciella beräkningar kan andra källor åberopas, till exempel Byggforskningsrapport R52:1973 för beräkning av kil- och diagonalkrafter.

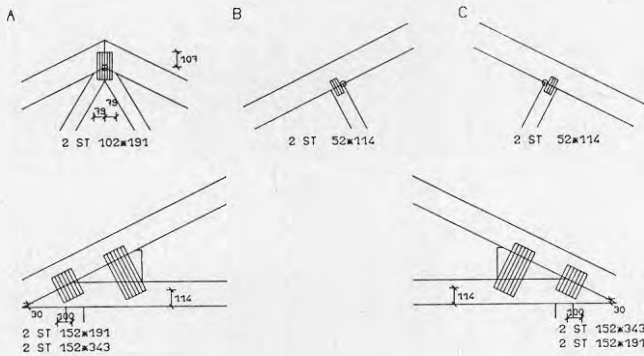
SBN 1980 är den norm som styr dimensionering av träkonstruktioner. Dessa behandlas speciellt i kapitel 27. Dessutom finns "Limträhandboken", utgiven av branschorganet Svenskt Limträ AB, vars kapitel 4 och 5 är godkända av Statens Planverk. I samband med beräkning av knäckning hänvisas även till StBK-N1 och handboken BYGG (1A och 3).

En av de stora stöttestenarna vid dimensionering av träkonstruktionerna är förbanden. Graden av samverkan kan ha stor betydelse för konstruktionens verknings sätt. Ta exemplet takstolar. Beroende på hur stor samverkan man antar i förbanden mellan de olika virkesdelarna kan man erhålla mycket olika resultat vad beträffar virkesdimensioner. Detta beror på att momentfördelningen ändras avsevärt med endast en obetydlig förskjutning i förbanden. Detta är ett förhållande som är mycket viktigt att man beaktar i sina beräkningar.

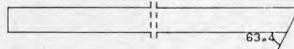




VIRKESDIMENSIONERING				SPIKPLÅTSTYP: HYDRO-NAIL E	
VIRKESDEL	BREDD	KVAL	UTN	KNÄCKN	AVST
	mm		z		
ÖVERRAM F-A-C	170	T20	80	A	2 ST 102*191
UNDERRAM F-C	170	T20	49	B	2 ST 52*114
DIAGONAL M-A	95	Ø	53	C	2 ST 52*114



ÖVERRAM F-A 47\*170 T20  
L<sub>TOT</sub> = 6093 mm



ÖVERRAM A-G LIKA DEL F-A

UNDERRAM F-A 47\*170 T20  
L<sub>TOT</sub> = 10140 mm



DIAGONAL M-A 47\*95 Ø  
L<sub>TOT</sub> = 2782 mm

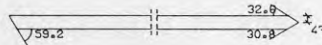


FIG 7.5 Utsnitt ur ritningar från takstolsprogram



## 8 TEXTBEHANDLING OCH DATABASER

I detta kapitel skall vi behandla två delområden som kan upplevas som en aning perifera för konstruktören i hans normala verksamhet, men som lika fullt kan bli en av de största datoranvändningarna inom byggnadsbranschen. Vi skall nämligen kortfattat diskutera textbehandling (eller ordbehandling) och databaser.

### 8.1 Textbehandling

Det sätt som konstruktören, i dagens läge, oftast kommer i kontakt med textbehandling är vid skapandet av program- och indatafiler, jämför kapitel 2. De "Editor"-program som används i dessa tillämpningar är ju nämligen en form av textbehandling, även om det inte är dessa man först tänker på.

Man skiljer ofta på tre olika former av datorbaserad textbehandling, nämligen tecken-, rad- och sidorienterad textbehandling. Dessa begrepp beskriver rätt väl komplexiteten hos de olika typerna, i det att de talar om hur stora "enheter" text som kan hanteras vid olika funktioner.

I små persondatorer har man oftast endast tillgång till teckenorienterade editor-program, exempelvis för uppbyggnad av en programfil. De funktioner man då kan utföra, förutom att från sin terminal skriva in text- eller sifferdata, är att ta bort, eller skjuta in tecken för att förändra textens innehåll.

I mera avancerade editorprogram för persondatorer, och i de flesta fleranvändarsystem med inriktning på beräkningar, har man också tillgång till ett radorienterat textbehandlingsprogram. Det som då tillkommer är framförallt möjligheter att byta ut vissa textsträngar i den skapade filen, att leta efter vissa teckenföljder, och att ta bort respektive lägga till hela rader. Man arbetar dock väsentligen på en rad i taget, och har ingen som helst koppling mellan slutet på en rad och början på nästa. Detta innebär att man knappast, åtminstone inte med några krav på effektivitet, kan arbeta med löpande text, eftersom varje tillägg eller borttagande av något ord kommer att kräva ett manuellt flyttande av ord mellan de efterföljande raderna.

För att kunna klara av denna typ av löpande texter behöver man tillgång till ett sidorienterat textbehandlingssystem. I detta finns den koppling mellan successiva rader som behövs för att man, på ett lätt sätt, skall kunna arbeta med och ändra i en text.

De sidor man talar om kan vara mer eller mindre diffusa begrepp. I många system svarar de mot den lagring

på skivminnet som datorn arbetar med, i andra fall åsyftas snarare den mängd text som på en gång ryms på datorns eller terminalens bildskärm.

Ett sidorienterat textbehandlingssystem kräver att man har tillgång till en bildskärmsterminal, och för det mesta också en relativt snabb kommunikation med datorn, eftersom bildskärmen mycket snabbt skall kunna visa upp utseendet av den nya text man skapat, efter varje ändring. Ett alternativ som ofta används i större textbehandlingssystem är "intelligenta" terminaler med en egen processor, och eget primärminne. I sådana system utförs det mesta textbehandlingsarbetet inom den egna terminalen, och man behöver kommunicera med huvuddatorn endast då större textavsnitt skall läsas över till skivminnet, eller skrivas ut.

När det gäller denna, mera avancerade form av textbehandling, kan den utföras på två olika sätt. Endera kan man ha, eller ha tillgång till, en speciell dator som är speciellt avsedd för textbehandling, och inte kan något annat, eller också kan man ha ett textbehandlingsprogram i den dator man normalt använder för sina beräkningar.

Dessa två alternativ behöver inte nödvändigtvis ge olika möjligheter, då vissa textbehandlingsprogram för generella datorer ger - åtminstone nästan - samma möjligheter för användaren att behandla texter på ett effektivt sätt. Skillnaderna får i stället snarast sägas ligga på ett ergonomiskt plan, då de specialiserade systemen ofta är utformade med tanke på att vara enkla, lättarbetade och säkra mot misstag. Så har man till exempel ofta specialtangenter för vissa funktioner som i den generella datorn sköts med hjälp av speciella koder, ofta genererade genom dubbeltryckta tangenter. Generellt krävs också en större allmän datorvana för att kunna utnyttja ett textbehandlingsprogram i en generellt användbar dator, i det att man måste veta en del om inloggning, operativsystem, filsystem och sådana saker.

De specialdedicerade textbehandlingsdatorerna ger ofta också vissa speciella möjligheter. Man kan till exempel ofta på ett ganska enkelt sätt lagra vissa standardformuleringar i ett bibliotek, och få dem fram-plockade och inlagda i en text genom att bara trycka ett par tangenter. Detta förfarande, som kanske främst är avsett för företag med en stor andel typ-brev och liknande, kan också antas ha ett visst värde för ett byggnadskonstruktionskontor, jämför också avsnittet om databaser och informationssökning. Av mera perifert intresse kan möjligheterna att disponera om texter vara, så att man till exempel omväxlande, på olika sidor, får ut texten i två kolumner och i en kolumn, eller andra layout-finesser.

För hantering av indata till program, och en del skrivning av korta texter i samband med byggnadskonstruk-

tionsarbetet ger dock ett editor-program i den vanliga beräkningsdatorn fullt tillräckliga möjligheter. Som ett exempel på ett sådant programs möjligheter kan nämnas att hela denna bok är framtagen med hjälp av ett sådant system. Det enda som inte lätt kunnat hanteras är de grekiska och matematiska tecken som finns i några av kapitlen, men detta är främst beroende på den använda skrivarens begränsade förmåga.

Ett intressant delområde när det gäller textbehandling, som på ett tydligt sätt påvisar svårigheterna att datorformulera problem som är lätta för den mänskliga hjärnan att hantera, är då det gäller avstavning av långa ord. Som framgår av våra dagstidningar, som helt och hållet framställs med hjälp av textbehandlingsprogram, är det mycket svårt att lära en dator att avstava på ett acceptabelt sätt - man behöver inte läsa så mycket i en dagstidning innan man hittar något exempel på en dum, om än inte direkt felaktig, avstavning. För att söka göra dessa avstavningar kan ett flertal olika strategier användas, av vilken dock ingen är 100-procentig. Det förefaller som om den bästa strategin, bortsett från tillämpningen på tidningar där ett sådant förfarande är tidsmässigt omöjligt, är att inte låta datorn avstava alls. I det program som denna bok framställts med används en så kallad interaktiv avstavning. Denna fungerar på så sätt att man, då ett ord inte får plats inom den textbredd man avsatt och det blir fullt att flytta hela ordet till nästa rad, får ett förslag till avstavning från datorn som man kan acceptera eller ändra, med hjälp av höger- och vänsterpilstangenterna på terminalen. Förbluffande ofta kommer datorn rätt genom att helt enkelt bryta ordet där högermarginalen skall vara.

## 8.2 Databaser

Vi skall i detta avsnitt något behandla begreppet databaser, ett av de många modeord som svävar över datoranvändningen, men som är svåra att ordentligt genomskåda. I kapitel 2 beskrevs begreppet "fil" som en samling information som lagras på ett eller annat sätt. En databas kan strängt taget definieras på samma sätt. Dock ställer man nog, för att databasbegreppet skall vara relevant, ganska stora krav på en långt driven strukturering av den lagrade informationen. Man skall nog dessutom kräva av databasen att en programmässig sökning av vissa nyckelbegrepp skall kunna ske.

För byggnadskonstruktören, och för byggbranschen, är det nog främst i tre sammanhang som databastekniken kan bli aktuell, nämligen i samband med informations-sökning, för lagring av beräkningsdata, och för projektdata om ett byggobjekt. Vi skall i de följande avsnitten behandla dessa tre typer var för sig.



CER BRASKAMINER		
ANTAL		
1	28	braskaminer, enkla braskaminer
2	30	braskaminer, varmluftskaminer
3	5	braskaminer, värmvattenkaminer

AB SVENSK BYGGTJÄNST BYGGVARUREGISTRERET 1983-01-17 SLUTSIDA SIDA

CER BRASKAMINER		
VÄROR		ORDN.NR 3
1	Energispisen/EB-öppen spis braskaminer, värmvattenkaminer Öppen spis, stål utg system f värmvatten ansl t panna o ackumulator Intermatic AB/EB-Pannan AB fd box 19	TILLV. 0340-30640 43020 VEDDIGE
2	Impuls 230 braskaminer, värmvattenkaminer Öppen spis av stålplåt med murad beklädnad avsedd för vattenburen värme bränselaren, Firma/Impuls Energi AB box 150	GENERALAG 0764-64610 18400 ÅKERBERGA
3	Jörgspisen 320 o 310 braskaminer, värmvattenkaminer Öppna spisar av stålplåt m värmekammare, typ 320 ansluts till radiatorer Elektro-Belinska AB Västmannagatan 47	GENERALAG 03-336460 11325 STOCKHOLM
4	Grants spisinsats braskaminer, värmvattenkaminer Öppen spisinsats för vattenburen värme, eldnig med ved,torv,flis el kol Finn-Takka AB Hantverkargatan 50	GENERALAG 08-530512 11231 STOCKHOLM

AB SVENSK BYGGTJÄNST BYGGVARUREGISTRERET 1983-01-17 FORTS. SIDA

CER BRASKAMINER		
H-NAMN		DATUM:8048
V-GRUPP	Grants spisinsats braskaminer, värmvattenkaminer	
VARUBESKRIVNING	Öppen spisinsats för vattenburen värme, eldnig med ved,torv,flis el kol LÅNG bxDxD:503x315x55/mm.bränsleförbrukning ca 1,5kg kol el motsvarande per timme för uppvärmning av vatten till 10-15 radiatorer samt tappvatten.	
GENERALAGENT	Finn-Takka AB Hantverkargatan 50	08-530512 11231 STOCKHOLM
TILLVERKARE	Grant Engineering Ltd Syngefield Birr Lo	IRI OFFFALY

SBK-UPPGIFTER DATABLAD

IG-NUMMER SBN UTGÅR

AB SVENSK BYGGTJÄNST BYGGVARUREGISTRERET 1983-01-17

FIG 8.1 Exempel på sökning i Byggvaruregistret

### 8.2.1 Informationssökning

Denna tillämpning av databastekniken är den som kommit längst i sin praktiska tillämpning. Tekniken ger en möjlighet att söka i register efter viss information, till exempel litteratur om ett visst ämne, information om olika byggnadsmaterial, eller -på sikt- sökande efter norm- och AMA-texter för en viss tillämpning. Vi skall exemplifiera med ett sökande efter information om byggnadsmaterial.

Sedan några år tillbaka erbjuder nämligen Svensk Byggtjänst användande av datorsökning i sitt byggvaruregister. Via en uppkopplad terminal kan man på detta sätt söka efter olika material med hjälp av olika sökbegrepp.

Förfarandet exemplifieras med hjälp av FIG 8.1, som visar utskrifterna vid en sökning. Den första delbilden visar resultatet då man bett datorn söka i databasen efter nyckelordet "braskaminer". Man får då först en uppställning av vissa undergrupper till detta nyckelord, och antalet lagrade fabrikat under var och en av de olika undergrupperna. Mer information kan fås fram genom specificerande av någon av undergrupperna - i den andra delfiguren syns början på en redovisning av innehållet i undergrupp 3 (som alltså totalt innehåller 5 produkter). En kort information om produkten fås tillsammans med tillverkarens namn och adress.

Ytterligare information om en speciell produkt kan sedan fås genom att specificera produktens nummer i listan. Denna information för en av produkterna framgår av den tredje delfiguren. Ur denna kan utläsas mera specialiserad information om produkten, samt uppgifter om till exempel ett eventuellt typgodkännande.

Det finns dessutom flera andra sätt att söka i registret. Om man till exempel känner till ett produktnamn, eller tillverkarens namn kan dessa användas som nycklar vid sökningen. Till exempel kan varunamnet "Grant" användas för sökning och datorn kommer då att plocka fram alla objekt i databasen som har produktnamnet Grant (vilket innebär att vi eventuellt kan få fram kökssnickrier, fogmassor och den visade braskaminen i samma lista).

Man kan även söka på olika typer av klassificeringskoder till exempel SFB eller BSAB-koderna.

För att ytterligare kunna förfina sökningen kan man ge dubbla sök-nycklar, till exempel som i FIG 8.1, men kompletterat med ett villkor att endast produkter med aktuellt typgodkännande skall listas.

Totalt innehåller registret i januari 1983 cirka 40000 produkter från cirka 10000 tillverkare.

På motsvarande sätt kan man även söka efter litteratur inom vissa ämnesområden. Vid olika bibliotek kan man via en terminal komma i kontakt med olika centraler för lagring av data om litteratur. Sökningen i dessa databaser går till på ungefär samma sätt som när det gäller byggnadsmaterial. Ofta har man dock möjligt att använda mera komplexa sökbegrepp. Till exempel kan man låta datorn, i databasen, söka efter litteratur som bland sina nyckelord har "Finita Element" och "Plattor" och "Betong" eller "Armering", en logisk konstruktion som kan ha sina sidor att formulera rätt.

En vidareutveckling av denna typ ligger nära till hands när det gäller normer och andra anvisningar. Man skulle ju på motsvarande sätt kunna lagra information om vad olika avsnitt i till exempel Svensk Byggnorm behandlar. På detta sätt skulle ju en konstruktör genom att ange nyckelorden "balkong" och "betong" kunna få fram alla relevanta normavsnitt och exempelvis AMA-stycken. En försöksverksamhet i denna riktning pågår också.

Man kan ju också, i förlängningen, se en koppling av denna sökningsteknik med ett textbehandlingssystem för framställning av vissa bygghandlingar o dyl.

Även om tekniken med datoriserad informationssökning har sina uppenbara fördelar, så har den också nackdelar. Vi kommer i kapitel 10 att utförligare diskutera riskerna att ett begränsat antal personer matar en dator, eller en databas, med sina personliga värderingar och erfarenheter. Det ställer ju till exempel i många fall stora krav på den som lagrar informationen att alla relevanta sökord kommer med, och att informationen hamnar under rätt rubrik.

Riskerna att den som skapar informationen och den som skall nyttja den använder olika beteckningar eller referenser är ju i vissa fall också stor, många produkter för byggbranschen kan ju till exempel ha flera olika beteckningar, tänk till exempel på masonit och träfiberskivor. Att för sådana fall få sökorden heltäckande kan ställa till problem.

### 8.2.2 Beräkningsdata

En annan intressant tillämpning av databastekniken, där viss verksamhet pågår är lagring av data för till exempel Finita Element beräkningar. I och med att programmen växer, och i samband med utveckling av pre- och postprocessorer för programmen, kommer ett behov av en systematisk lagring av data för beräkningarna att uppstå. Alldeles speciellt gäller detta kanske i samband med pre- och postprocessorer. Om bara olika Finita Element program kunde fås att lagra information om noder och element o dyl på ett gemensamt sätt, så skulle ju till exempel samma program för presentation av de be-

räknade data kunna användas. Samma sak gäller vid skapandet av beräkningsmodellerna.

Många pre-processorer för FE-beräkningar är också gjorda på detta sätt. Ta till exempel FEMGEN, utvecklat vid Lunds Tekniska Högskola, som är ett program med vars hjälp man interaktivt kan framställa FE modeller för olika typer av strukturer. Programmet arbetar internt mot en databas med ett standardiserat utseende. Informationen i denna databas kan sedan, med hjälp av översättningsprogram, anpassas till det exakta indatautseende som ett speciellt beräkningsprogram kan läsa, se principskissen i FIG 8.2 hämtad ur FEMGENS manual.

Ett intensivt arbete på detta område är också nödvändigt för en vidareutveckling av de nu tillgängliga beräkningsprogrammen. Som närmare kommer att diskuteras i kapitel 11 är det främsta önskemålet på beräkningsprogrammen en långt driven modularisering. En förutsättning för detta torde vara en väl utvecklad databashantering, så att de olika modulerna på ett standardiserat sätt kan utbyta data.

Att detta med databashantering var viktigt för framtiden var också en av de viktigaste slutsatserna av det inventeringsarbete som STU bedrev i mitten av 1970-talet, se avsnitt 3.3.4. Avsikten med projektet var egentligen att söka ena industrin och högskolorna i ett gemensamt samarbetsprojekt rörande FE-program. Då detta inte mötte något gehör från industrin, försökte man i stället enas om en gemensam datalagringsstruktur för att möjliggöra överföringar mellan olika program. Ett projekt har också pågått som fortsättning på det ursprungliga projektet, med uppgift att ta fram en databasstruktur lämpad för olika Finita Element prog-

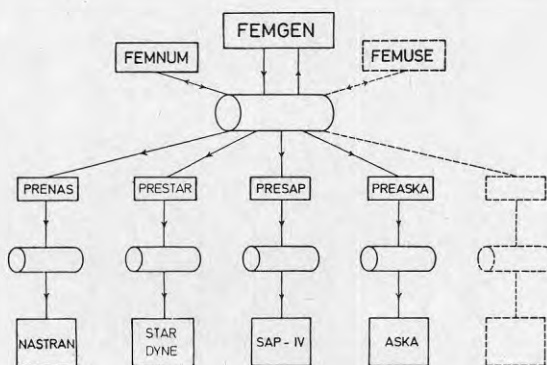


FIG 8.2 Principskiss av FEMGENS användning

ram. Projektet har dock tyvärr ännu inte fått någon större genomslagskraft bland de olika intressenterna.

### 8.3 Projektdatabaser

I kapitel 9 kommer användning av CAD (Computer Aided Design) att diskuteras. Med den definition av detta som vi vill använda kan CAD tolkas som ett interaktivt skapande av grunddata för exempelvis ett byggnadsprojekt. Den primära avsikten med dessa data är, åtminstone än så länge, att skapa underlag för automatisk ritningsframställning.

I samband med CAD är också databaser intressanta. Med tanke på de oerhörda mängder grafisk information som ett större byggnadsverk består av är en effektiv lagring av data absolut nödvändig. För detta används databastekniken. Den information som då behöver lagras är förutom vissa koordinater för olika byggnadselement även uppgifter om materialtyper och viss textinformation i form av littera-beteckningar o dyl.

För att CAD-tekniken skall kunna bli riktigt effektiv krävs det förmodligen att dessa databaser, som kräver många mantimmar att skapa kan utnyttjas för något mer än enbart som underlag för framställande av ritningar. Till exempel blir det nog nödvändigt att söka lösa problemen med överföring av geometridata och dylikt mellan de olika programsystem som används av olika företag.

Projektet pågår också för att söka lösa dessa problem med dataöverföring mellan olika CAD-program. Vid överföring mellan olika CAD-program är det dock inte tillräckligt att ha ett översättningsprogram som kan omforma data från ett program så att det passar till ett annat. Detta är beroende på de skillnader i logik som de olika programmen använder. När det ena programmet kanske lagrar sin information i form av data om ytor, vill ett annat ha data i form av linjer, en skillnad som kan vara nog så svår att övervinna.

Länkar mellan några olika system har dock etablerats. Dessa arbetar dock på en relativt låg nivå, så tillvida att endast färdiga ritningsdata kan överföras. Den bakomliggande logiken i uppbyggnaden av data kan dock inte flyttas. Med detta avses då exempelvis att information om exempelvis väggelement som skall hänga ihop ej kan flyttas.

Det finns även andra intressanta kopplingar för de data som CAD-systemen genererar. En vision är till exempel att överföra det av arkitekten önskade utseendet på byggnadsverket till data som kan användas av konstruktören för hans beräkningar. Även i denna riktning pågår ett stort arbete. I kapitlen 10 och 11 kommer vi att framföra vissa synpunkter på en sådan integrering,



och de risker denna utveckling kan medföra med tanke på de olika parternas möjligheter att kontrollera och påverka projekteringsprocessen.

Andra riktningar finns dock, i vilka man vill kunna överföra grunddata för ett projekt. Till exempel är ett uttagande av mängdberäkningar ur den av CAD-systemet skapade databasen en intressant möjlighet, som är under utarbetande. Flera av de nu befintliga CAD-systemen kan också göra mängdförteckningar, åtminstone för vissa, väldefinierade och kvantifierbara, delar av ett bygge, till exempel för snickeridetalljer, armering och dylikt.

Med utgångspunkt från detta ser man också en vidareutveckling av databaslagringen för ett byggnadsprojekt. Man kan till exempel ganska lätt komplettera den information som det första, kreativa steget skapar, med andra data som är intressanta för produktionsskedet, till exempel tidåtgång, erforderlig arbetsinsats och planerade kostnader. Med dessa som grund skulle tidplaneringen av ett helt projekt kunna automatiseras.

I princip skulle en enda databas kunna hålla all information om ett byggnadsverk, från de första skisserna, över projektering och produktion till underhåll, drift och till sist kanske en rivning, enligt FIG 8.3. I denna är det, på ett mycket förenklat sätt, tänkt att arkitekten framställer sina grunddata i form av byggnads-element, material och koordinater och lagrar dessa i databasen. Dessa grunddata kan sedan kompletteras, och eventuellt förändras av de övriga konsulterna, när des-

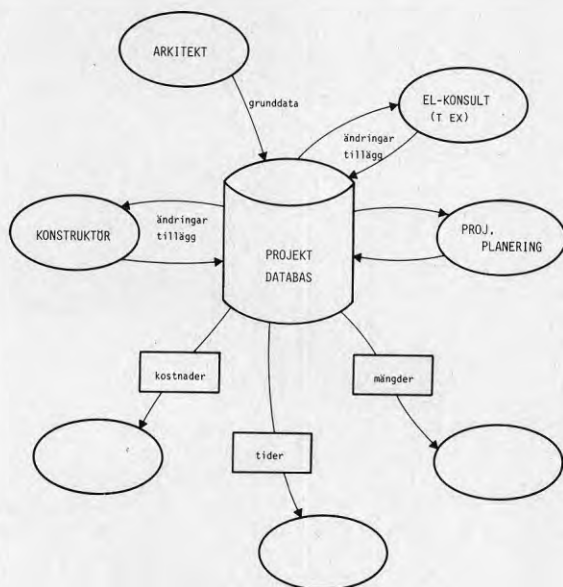


FIG 8.3 Vision om projektdatabas

sa lägger konstruktiva aspekter på byggnadsverket. Ur dessa kan sedan en produktionsplanering ta vid, som utifrån de givna grunddata kan räkna fram tidpunkter och kostnader, och komplettera databasen med dessa komponenter.

Fördelarna med en sådan långt driven strukturering av alla intressanta data för ett projekt, är att finna på kommunikationssidan. I stället för att de olika konsulterna skall hålla på att sända uppdaterade versioner av sina ritningar mellan sig, finns hela tiden en gemensam och aktuell samling data i databasen, vilket bör minska riskerna för att till exempel El-konsulten bygger sitt arbete på inaktuella grunder. Vilken som helst av de deltagande parterna kan ju när som helst ta ut en just nu gällande bunt ritningar.

Från dessa projektdatabaser kan ju också en firma, i efterhand, söka fram projekt som har utförts av en viss typ, för en viss kund eller på en viss plats, vilket kan vara värdefullt som referenser vid ett nytt projekt med liknande förutsättningar.

Steget torde också, från dessa projektdatabaser, vara kort till uppbyggnadet av typlösningssbanker där man, efter att en gång ha analyserat ett visst byggnadselement, kan lagra den slutliga lösningen för kommande behov.

Huruvida detta är önskvärt med tanke på det kreativa skapandet är dock diskutabelt. I detta måste fördelarna med successivt förfinade typlösningar vägas mot nackdelarna med en minskad variation i utföranden.

## 9 PROGRAM FÖR DESIGN OCH RITNING

### 9.1 Inledning och begreppsförklaringar

Det begrepp rörande datorer inom byggbranschen som för närvarande tilldrar sig det största intresset är otvivelaktigt CAD. Detta intresse har uppstått under 80-talet främst genom att ett antal företag har införskaffat generella system av utländskt ursprung. Viss egen utveckling har även skett hos vissa företag som därmed också försöker att rida på den uppkomna "CAD-vågen".

Det är dock de generella utländska systemen (främst engelska och amerikanska) som har skapat det stora intresset och nyfikenheten. Att intresset har blivit så stort är nog beroende av att det är så många "stora och tunga" konsultfirmor som har satsat mångmiljonbelopp på inköp av CAD-system samt i utbildning och i produktanpassning. Dessa satsningar har sedan utnyttjats som ett säljande argument.

Denna bok har huvudinriktningen att belysa datoriserade konstruktionsberäkningar. Idag är det en relativt liten koppling mellan detta och CAD vilket innebär att vi här endast ska ge en relativt ytlig genomgång av CAD. Dessutom är ingen av denna boks författare så kunnig om CAD som skulle krävas för att ge en fylligare genomgång. Vidare är CAD-tekniken inom byggbranschen så ny och utvecklas för närvarande så snabbt att det som gäller idag (1983) sannolikt är passé om några år. Detta innebär att den som önskar sig en djupdykning inom CAD-området hänvisas till speciallitteraturen, se kapitel 12.

Eftersom CAD är "morgondagens teknik" och är av stort PR-intresse är det också känsligt och prestigeladdat. Detta ska dock inte hindra oss att framföra en del problem och nackdelar med CAD, se avsnitt 9.4. Någon analys av olika fabrikat eller system kommer emellertid inte att utföras.

Innan detta behandlas skall först redogöras för ett antal begrepp och förkortningar som förekommer inom området.

Av själva huvudordet, CAD, finns det nästan lika många tolkningar som det finns personer som använder det. CAD står oftast för Computer Aided Design men används ibland även för Computer Aided Drafting. Eventuellt används även CADD, Computer Aided Design and Drafting. Detta skapar en viss förvirring som förstärks av att "design" tolkas olika på engelska och svenska.

Författarna av denna bok tolkar CAD som ett: "Interaktivt, grafiskt framställande av grunddata för ritningar och beräkningar".

På svenska används ibland istället benämningen datorstödd projektering, som ibland förkortas DSP.

Om än inte lika vanligt som CAD så är CAM, Computer Aided Manufacturing, också ett vanligt och vedertaget begrepp. På svenska benämns detta datorstödd tillverkning.

Det existerar ett stort antal ytterligare förkortningar som är mer eller mindre vanligt förekommande. Till exempel CAE, Computer Aided Engineering, och CAL, Computer Aided Learning.

Ofta skiljer man på "turnkey-system", d v s system där hårdvara och mjukvara är en enhet, och "öppna system", system där mjukvaran inte är kopplad till någon speciell datorutrustning.

I avsnitt 9.2 kommer en uppdelning mellan generella och speciella system att användas. Med generella system avses system som inte är specialskrivna för byggbranschen eller för någon del av byggbranschen. Dessutom kommer system som används enbart för ritningsframställning att skiljas från övriga system.

## 9.2 Existerande system

Om man i allmänhet talar om CAD, utan att specificera det närmare, är det många som tänker på elektronikindustrin och på kretskortskonstruktion. Det är ju inom detta område som CAD/CAM fått störst spridning.

Anledningar till detta är dels att elektronikindustrin tidigt "datoriserades" och dels att geometrin för kretskort är enkel, det rör sig huvudsakligen om två dimensioner. Den enkla geometrin medför att kraven på datorkapacitet är begränsade, vilket innebär att man kan använda förhållandevis små datorer.

Följaktligen finns det ett antal system på marknaden som är specialanpassade för elektronikindustrin. Dessa system är relativt billiga och kräver inte de miljoninvesteringar som de stora CAD-systemen kräver. Dessa speciella system är dock inte intressanta för byggare och kommer inte att omnämnas vidare.

Andra områden där CAD-tekniken tidigt utnyttjades är bilindustrin och rymd/flygindustrin. Inom dessa områden började utvecklingen redan på 1960-talet, för rymd/flygindustrin inom ramen för de stora amerikanska rymd- och försvarssatsningarna.

Många system som idag används inom andra områden, som till exempel inom den mekaniska industrin, är ursprungligen framtagna av rymd/flygindustrin. Ett exempel på detta är CADAM, som marknadsförs av IBM men som är ut-

vecklat av Lockheed. Eftersom systemen även används av byggnadsindustrin ska systemen för olika industrigrenar behandlas gemensamt i avsnitt 9.2.1 nedan.

Denna redovisning av de generella systemen, men även redovisningen av de speciella systemen och av ritsystemen i de följande avsnitten, är dock långt ifrån någon heltäckande genomgång. I stället har ett fåtal system utvalts mer eller mindre på måfå. Avsikten med detta är mer att visa ungefär vad man kan göra än att sams Mellan jämföra dem.

För att kunna visa systemens bygg-tillämpningar så är dock de flesta som utvalts i drift inom företag knutna till byggbranschen. Vilka system, och hur många, som har installerats inom byggbranschen genomgås emellertid i ett separat avsnitt, 9.3. Denna uppdelning har valts främst med avsikten att erhålla en så opartisk genomgång som möjligt.

### 9.2.1 Generella system

Med "generella system" avses stora system som är användbara inom flera olika industrigrenar. Systemen som är till salu ligger i prisklassen några miljoner kronor. Man behöver dock inte köpa ett system eftersom det finns system tillgängliga hos olika servicebyråer eller konsultföretag.

De flesta systemen är främst framtagna för andra industrigrenar än byggnadsindustrin, men det finns undantag till exempel GDS (General Drafting System). Många har dessutom genomgått en långtgående specialanpassning för byggnadsindustrin. Utan en sådan anpassning är det nog svårt att få ekonomi med systemen. Detta beror emellertid på vilken tillämpning man främst avser att bedriva med sitt CAD-system.

I detta avsnitt ska vi försöka avstå från värderingar men vi återkommer med en kort diskussion om detta i avsnitt 9.4.

Det finns åtskilliga generella CAD-system på den svenska marknaden. Uppskattningsvis rör det sig om ett tjugotal, till exempel APPLICON, CADAM, MEDUSA och GDS. Systemen kommer nästan uteslutande från USA eller England.

Ett öppet generellt system som utvecklats i England av Applied Graphics System är MEDUSA. Som ett exempel på ett amerikanskt turnkey-system kan nämnas INTERGRAPH. Dessa system som finns installerade i Sverige ska användas för att visa något om vad generella CAD-system kan användas till.

Det första man lägger märke till i ett generellt CAD-system är den grafiska presentationen. Det vill säga systemen kan generera såväl plana 2-dimensionella kon-



ventionella arbetsritningar som 3-dimensionella perspektivritningar, se FIG 9.1 som är ett exempel från MEDUSA. Perspektiven kan ses från en godtycklig punkt såväl utvändigt som inuti själva konstruktionen.

Dessutom kan figuren, ritningen redovisa endast en del av konstruktionen, till exempel det bärande systemet eller hela konstruktionen inklusive fasadbeklädnad. För att underlätta uppritandet finns dessutom olika standardföremål och standardbeteckningar inlagda i "bibliotek" för att enkelt kunna tas fram vid behov.

Med systemen kan man givetvis också betrakta valfria delar av konstruktionen i valfri skala. FIG 9.2 som visar en bild från INTERGRAPH-systemet ger en bild av detta. Systemet arbetar med två bildskärmar där man parallellt kan arbeta med olika delar av konstruktionen.

Ett generellt system måste emellertid vara mer än en "elektronisk ritare". Systemen ger därför inte enbart ritningar utan även vissa mängdförteckningar, till exempel armeringsförteckning eller antal stålbalkar av en viss dimension. Till mängderna kan eventuellt också kopplas kostnader, så att man samtidigt med förteckningarna kan få fram en kostnadskalkyl. Dessutom kan man snabbt och enkelt ändra någon parameter och få fram en jämförelsekostnad.

Det är kombinationer, till exempel ritningar-mängder, som kan innebära ekonomi med ett CAD-system, jämför diskussionen om projektdatabaser i avsnitt 8.3.

Förutom ritningar och mängdförteckningar är ordning och samordning centrala begrepp för CAD-system. Med ordning avses att alla ändringar som görs på en ritning kan utföras omedelbart och på ett enda ställe. På detta sätt kan man hela tiden arbeta med den senaste versionen av en ritning.

Med samordning avses ett samarbete mellan olika delar i konstruktionsprocessen men främst av allt avses en samordning mellan alla konsulter, A, K, V och E. En sådan samordning är mycket värdefull. Onödiga kollisioner kan därmed undvikas. Till exempel genom att man arbetar med samma underlag kan man se om ventilationskanalerna innebär att någon ändring av det bärande systemet måste göras. Om man då ändrar någon bärande del så går denna ändring även direkt till de övriga berörda, A och E.

Det ska dock framhållas att samordningen inte är så enkel som det kanske låter. Samordningsprincipen är dock onekligen en mycket attraktiv ide. Den har använts i såväl MEDUSA, INTERGRAPH som några andra CAD-system.

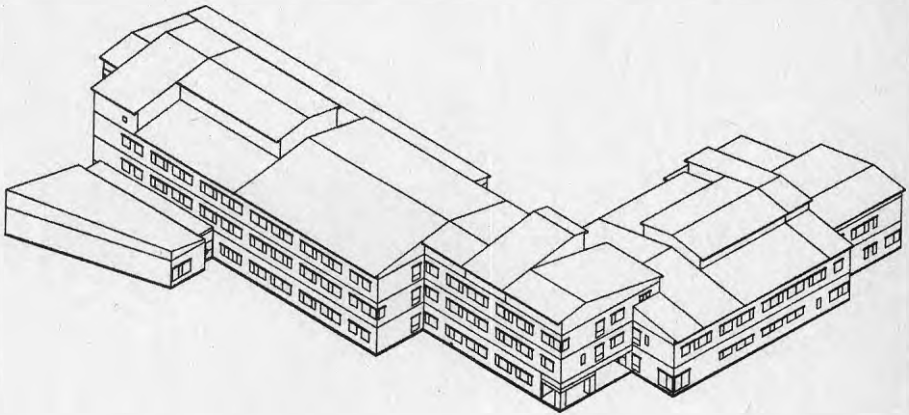
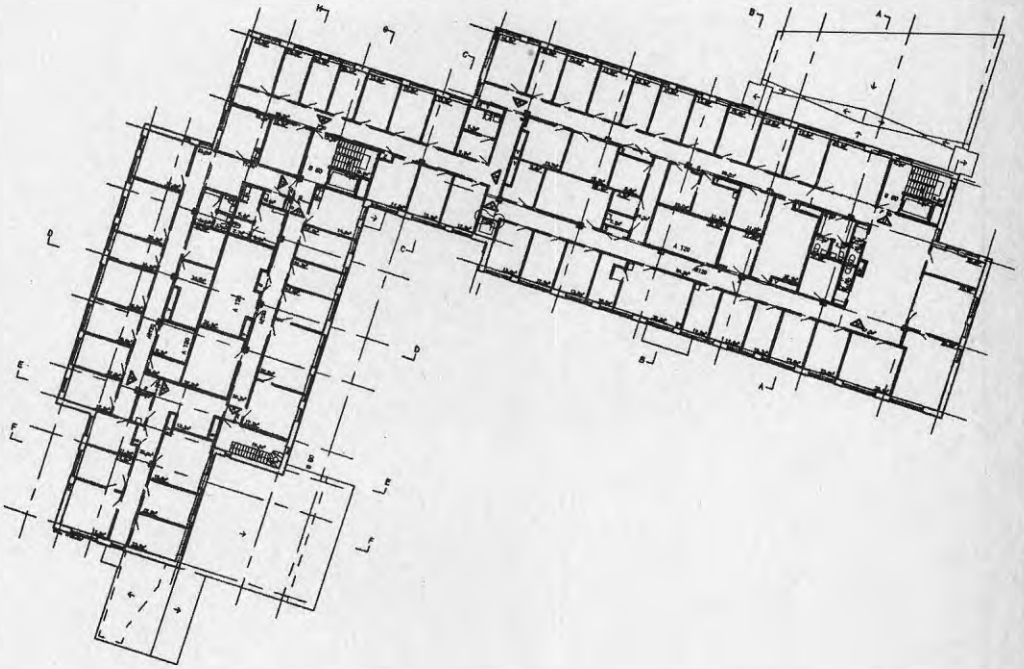


FIG 9.1 Ritningar utförda med hjälp av MEDUSA

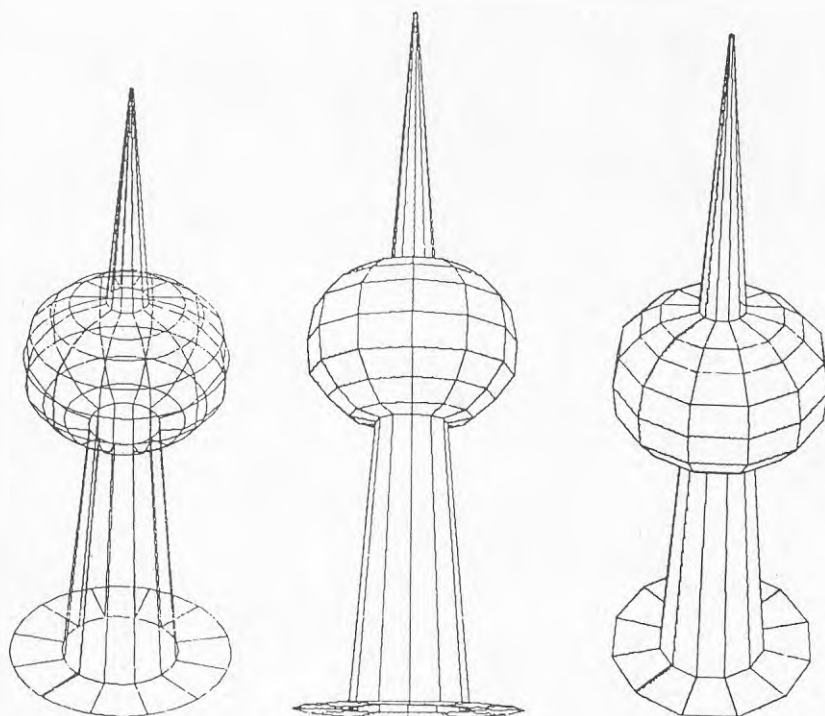


FIG 9.2 CAD-systemet INTERGRAPH

De grafiska möjligheterna i systemen skulle kunna användas för generering av geometri till konstruktionsberäkningarna. Utdata från beräkningarna kan sedan ge underlag för ritningar och mängdförteckningar. Kopplingen CAD-system och konstruktionsberäkningsprogram är dock inte så väl utvecklad idag.

Till ovanstående beskrivning av de generella CAD-systemen skall för fullständighetens skull även en del problem och nackdelar omnämnas. Som redan antytts så är ekonomin ett av problemen. Vi ska återkomma med en diskussion om ekonomin och andra problem, men även ytterligare några positiva aspekter i avsnitt 9.4.

### 9.2.2 Speciella system

De speciella systemen är en mera heterogen grupp än de generella. Det finns dock några punkter som särskiljer de speciella systemen från de generella, förutom det uppenbara förhållandet att de speciella systemen är anpassade för något avgränsat användningsområde. De speciella systemen är ofta utvecklade i Sverige eller anpassade till svenska förhållanden. Dessutom gäller för de flesta speciella system som är till salu att priset är av storleksordningen hundratusentals kronor. Detta skall jämföras med de generella systemens mångmiljonkostnader.

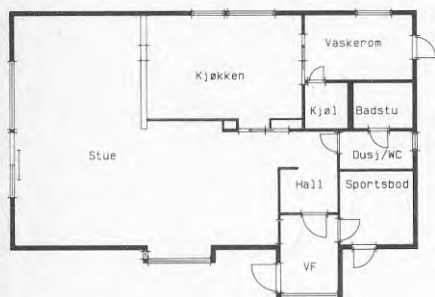
Man kan tycka att de generella systemen kan vara onödigt generella och dyrbara för en viss tillämpning. För en klart avgränsad inriktning kan kanske ett speciellt system klara allt vad ett generellt system klarar, eller åtminstone enklare, inom detta speciella område.

Utan att ge avkall på de stora CAD-systemens 3-dimensionella grafik, mängdförteckningar eller kostnadsberäkningar kan man skapa system till ett väsentligt lägre pris. Man måste dock ge avkall på generaliteten och kanske även på samordningen, jämför avsnitt 9.2.1.

Vi ska här ge exempel på två speciella system. Dessa är dock relativt åtskilda och ger därmed en bild av hur olika de speciella systemen kan vara.

HAMBO är ett system för projektering av småhus, utvecklat i Norge men finns tillgängligt även på den svenska marknaden. Systemet kan ge plan- såväl som perspektivritningar samt ge materialspecifikationer och beräkna kostnader.

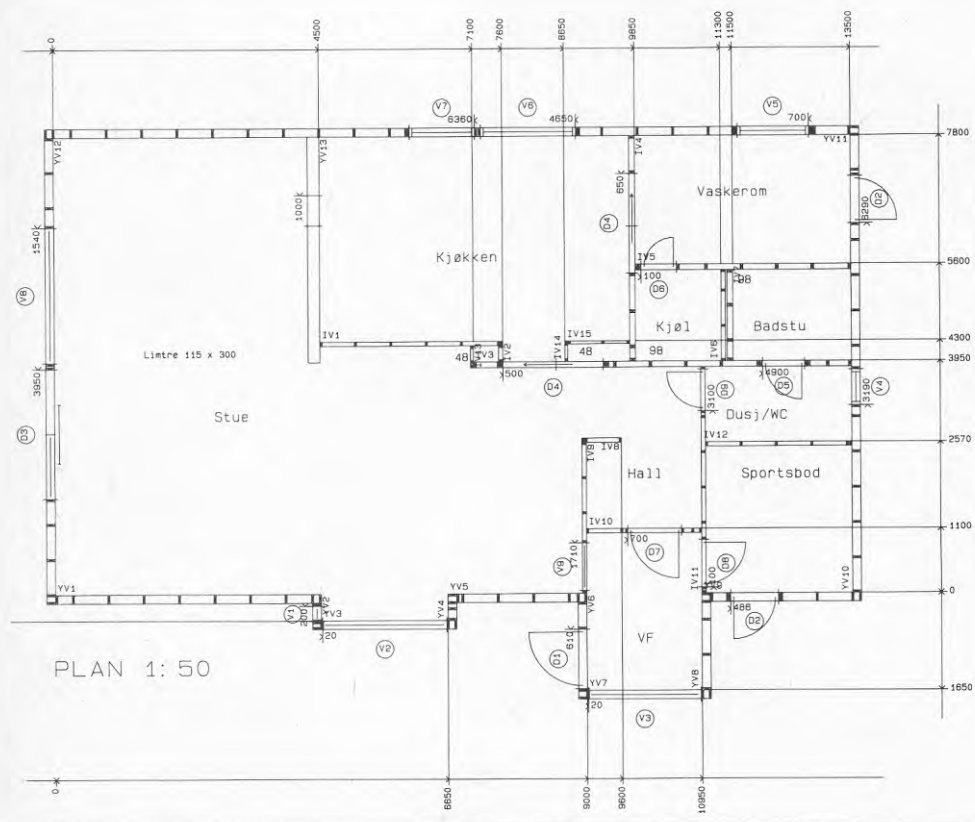
Ett exempel på specialanpassning i HAMBO är att ritsättet är beroende på val av skala. En plan i 1:100 ritas med ett förenklat ritsätt typ "arkitektplan". Planer i 1:50 eller större skala kedjemåttsätts och dörrar och fönster får littera, se FIG 9.3.



PLAN 1: 100

DØRER		
Nr	Stk	Beskrivelse
D1	1	Scadania YD, teak 636, 10 x 21 venstre
D2	2	Kvasnes Ytterdør, tett, 8 x 21 venstre
D3	1	Strømmen Heve/skyvedør 219 x 209, A, Venstre
D4	2	Karm 170 x 212, Blad Scadania Anna S01, furu
D5	1	Tyle badstuedør 71,5 x 185
D6	1	Kjøleromsdør 60 x 185, venstre
D7	1	Scad. Anna S01, furu, 9 x 21, H. m/sprosset glasskasett.
D8	1	Scadania Anna S01, furu, 8 x 21, høyre
D9	1	Scadania Anna S01, furu, m/ spalte 7 x 21, venstre

"48" angir lett bindingsverk, justert lekt 48 x 48 mm  
 "98" angir vegg, justert plank 48 x 98 mm  
 Innvendige vegger forøvrig: Justert lekt 36 x 68 mm



PLAN 1: 50

FIG 9.3 Ritninger från HAMBO



Det finns även specialsystem som inte är en "nedskalning och specialanpassning" av de stora CAD-systemen utan snarare kan ses som ett alternativ till dessa.

Ett system man inte kan undgå att nämna i detta sammanhang är BERIT. Systemet har en given plats i denna bok eftersom det kopplar konstruktionsberäkningar och ritningar. BERIT som började utvecklas i början av 70-talet är dessutom utvecklat av "byggare" (J&W) och är anpassat till de speciella förhållanden som råder inom byggnadsindustrin. Ett område där systemet har använts flitigt är betongkonstruktioner, se FIG 9.4 som visar en armeringsritning.

Systemet har däremot inte den generalitet och den grafiska kapacitet som de generella systemen har. Arbets sättet är dessutom mera av typen interaktiv "sifferinmatning", vilket ska jämföras med de generella systemens arbetssätt med grafiska terminaler, digitaliseringsbord och väl utvecklade menymöjligheter. Däremot är BERIT väl uttestat och har använts för projektering av åtskilliga byggnadskonstruktioner.

### 9.2.3 Ritsystem

Det finns system på marknaden som endast används för grafik och för ritningsframställning. De marknadsförs ofta som CAD-system, där D står för "drafting". Systemen är ofta väsentligt billigare än de generella systemen och kräver mindre datorkapacitet. De är i gengäld ofta begränsade till 2 dimensioner.

Inom byggnadsindustrin har man inte så besvärliga former som i, till exempel, den mekaniska industrin. För byggnadsritningar används ju huvudsakligen räta linjer och räta vinklar. Detta innebär att mera avancerade ritsystem nog har en större marknad inom andra industriegrenar än inom byggnadsindustrin.

För framställning av ritningar och figurer tillhandahåller många datorfabrikat grafiska paket. Dessa kan utgöra en bas för en egen programutveckling eller kan användas för att framställa diagram och figurer. För denna tillämpning finns det dessutom ett antal fristående system som inte är kopplade till någon speciell dator typ. Utan att lägga ner alltför lång tid kan man relativt enkelt framställa diagram som har en tilltalande layout. I FIG 9.5 visas en 3-dimensionell resultatredovisning skapat med systemet DISSPLA.

Slutligen skall vi även nämna något om de grafiska program som finns för att generera indata till beräkningsprogram, främst FE program. Dessa brukar benämnas preprocessorer och deras största användningsområde är att generera geometrier och Finita Elementnät. En del preprocessorer kan dock generera en helt komplett indatamängd för ett visst FE program.

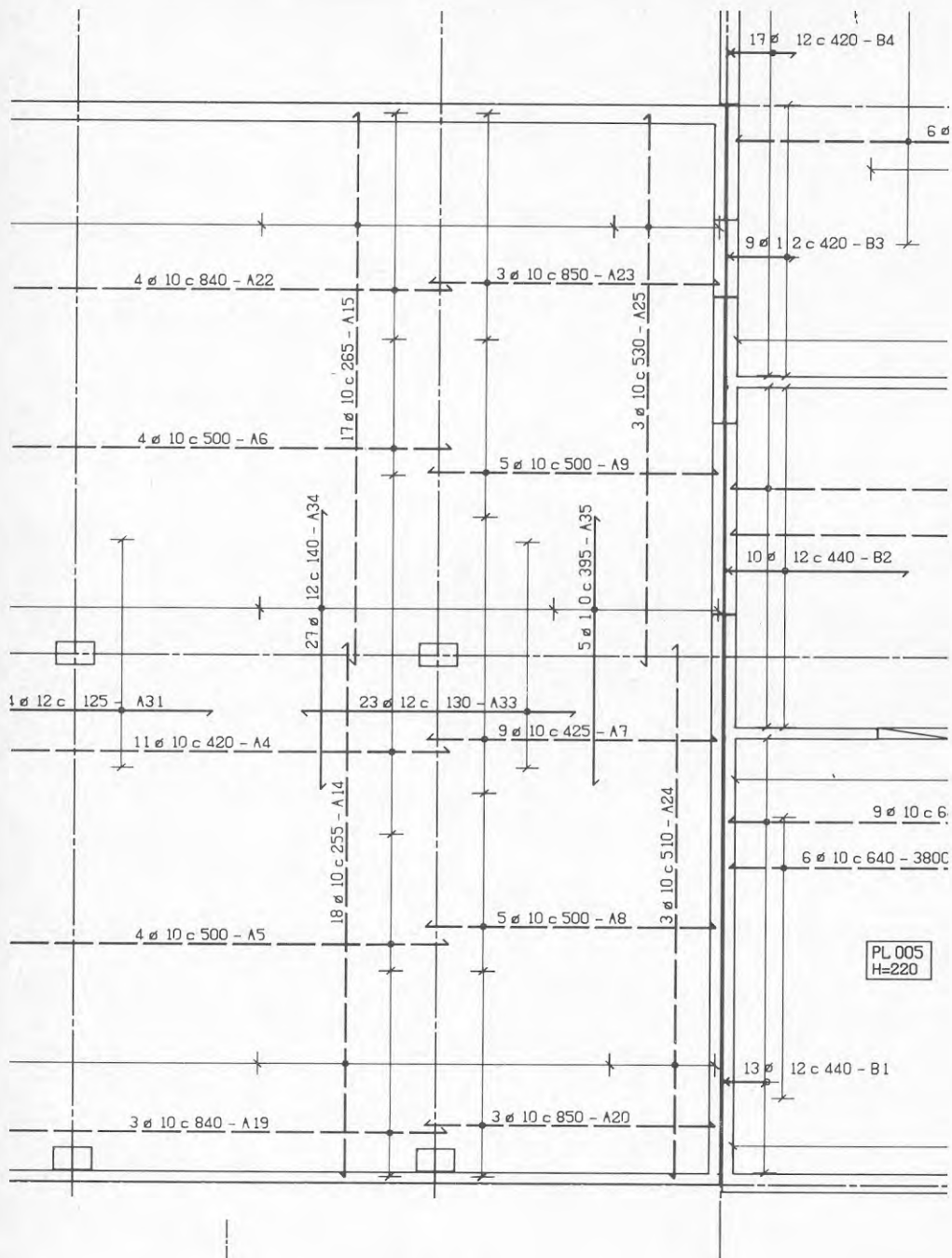


FIG 9.4 Armeringsritning från BERIT

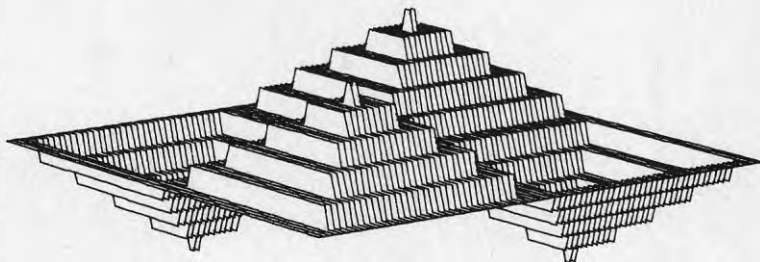


FIG 9.5 Utmatning från DISSPLA (hämtad ur "24 sidor om QZ")

Många program är interaktiva och utnyttjas bäst vid grafiska skärmar. Med preprocessorer kan man, på detta sätt, snabbare generera och kontrollera komplicerade geometrier. FIG 9.6 visar ett elementnät genererat med den i Sverige utvecklade preprocessorn FEMGEN, se avsnitt 12.8.

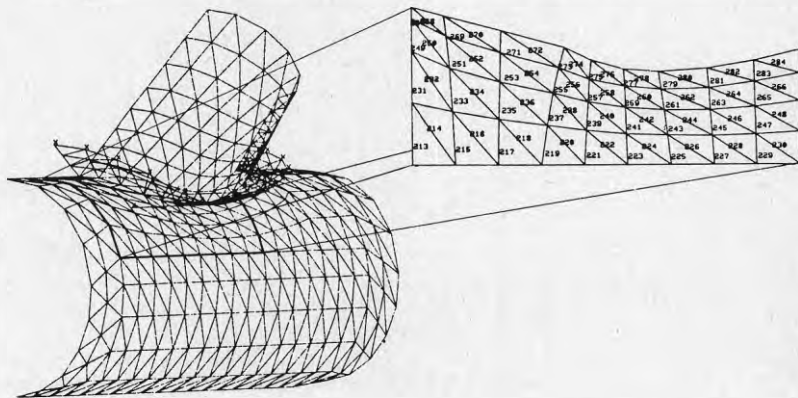


FIG 9.6 FE modell genererad av FEMGEN

För presentation av beräkningsresultat används så kallade postprocessorer. En del program kan användas både som pre- och postprocessor, till exempel GIFTS.

Pre- och postprocessorer är inte egentliga "ritsystem", men har omnämnts här eftersom de ofta är de enda system som används för den grafiska presentationen i anknytning till beräkningsprogram. I framtiden kan nog detta inkorporeras i de generella CAD-systemen. Till en viss del sker så redan idag.

### 9.3 Marknadsöversikt

I detta avsnitt skall ges en översikt av i vilken omfattning CAD-system finns installerade hos svenska företag med anknytning till byggbranschen. Översikten baseras på andra översikter i ett antal publikationer. Detta innebär dock att avsnittet främst täcker de generella systemen.

#### 9.3.1 Internationell utblick

Det kan vara lämpligt att först beröra situationen i en del andra länder innan vi ger en beskrivning av situationen i Sverige. Detta eftersom de i en del fall ligger före Sverige vad gäller utvecklingen av system. Sverige intar däremot en tätposition då man räknar det antal system som är i drift i förhållande till antal företag. Beskrivningen av situationen utomlands kommer dock att bli kortfattad och begränsad.

Det nämndes i avsnitt 9.2.1 att de generella systemen främst är av amerikanskt eller engelskt ursprung. Det är också dessa länder som är i förgrunden när det gäller CAD-användning inom byggbranschen.

Spridningen är dock begränsad även för dessa länder. Till exempel använder sig, av USAs cirka 100 000 företag med anknytning till byggnadsindustrin, endast något 100-tal av CAD. Trots att den procentuella andelen är så låg finns det företag som har hunnit mycket långt i utvecklingen. En del företag har flera olika CAD-system som används för skilda användningsområden.

Användningen av CAD-system i USA är ofta knuten till någon speciell nisch. Som ett exempel kan nämnas ett amerikanskt företag som använder CAD för sin projektering av skolor. Genom denna inriktning har de haft lättare att bygga upp standardbiblotek med typlösningar.

Intressant är att inget av de 10 största byggföretagen i Väst-Tyskland har något CAD-system i drift. Detta tekniskt högtstående land intar en klart avvaktande inställning till CAD (inom byggnadsindustrin) och menar att en CAD-arbetsplats idag ger dålig lönsamhet.

Angående lönsamheten skall vi återkomma till denna väsentliga aspekt på datoranvändning för CAD i avsnitt 9.4.2

I övriga västländer har man, precis som i Sverige, just påbörjat CAD-åldern. Om länder som är mer avlägsna kan det till exempel nämnas att det engelska GDS-systemet har installationer i Australien. I Japan går hårdvaruutvecklingen fram med stormsteg.

### 9.3.2 Svenska marknaden

I Sverige har, som nämdes i inledningen till kapitlet, ett flertal olika företag införskaffat CAD-system. Det är konsultföretagen som har satsat mest och som har kommit längst. De stora CAD-satsningarna är dock relativt unga och de inköpta generella systemen har oftast endast några få år på nacken.

Konsultföreningen har gjort en sammanställning av medlemsföretagens CAD-utveckling, se FIG 9.7. Som kan utläsas ur figuren är det främst de stora och välkända företagen som har CAD-system. Figuren visar även att det existerar ett alternativ till eget inköp, nämligen ett samarbete över företagsgränserna. Detta åskådliggörs av DAPAB, en formation av sex konsultföretag, som har skaffat sig ett gemensamt, generellt system.

Byggnadsentreprenörerna har ännu inte satsat lika hårt som konsultföretagen. De är dock klart medvetna om marknaden. Några företag har köpt egna system, medan andra hyr in sig hos andra företag. Detta gäller till exempel SIAB som har arbetsstationer kopplade till Arne Johnsons system, MEDUSA. På den kooperativa sidan har HSB länge använt sig av BERIT medan Riksbyggen nyligen har införskaffat GDS.

Inom trähusindustrin följer man också CAD-utvecklingen. Det finns installerade generella system och flera speciella system har utvecklats för detta område, jämför avsnitt 9.2.2.

Även den offentliga sidan försöker följa med. Som exempel på detta kan nämnas att flera kommuner, till exempel Mora och Göteborg, har CAD-system och att det vid Lunds Tekniska Högskola skall installeras ett generellt system under våren 83. Systemet skall användas för bland annat forskningsändamål.

Dessutom finns det naturligtvis CAD-system hos olika servicebyråer.



## Konsultbranschen och CAD-utvecklingen

Företag	Antal anställda	CAD-system/dator/kapacitet	Antal installerade arbetsstationer (okt. -82)
VBB	1300	Intergraph (USA) DEC PDP 11/70 primärminne 1 Mb yttre minne 2 x 300 Mb	5 st ytterligare 3 är beställda
<i>Utrustningen används främst (hittills) på integrerade samhällsprojekt utomlands. På VBBs utrustning har också Bergman&amp;Co utbildat personal och använt CAD-systemet i egna uppdrag. Även Theorells har utbildat personal. Den senast installerade utrustningen kommunicerar med datorn via telenätet. VBB planerar ytterligare en kraftfull utbyggnad bl a för Malmökontoret.</i>			
Tyréns FFNS (arkitekt) Samarbetar via NordCAD	325 250	GDS (England) Prime 550 1,5 Mb 380 Mb Prime 750 1 Mb 380 Mb	4
<i>En station är för produktion installerad hos vardera Tyréns och FFNS och två, delvis för utbildning och utveckling hos NordCAD. Tyréns har haft systemet i produktion i ett femtonårigt uppdrag, varav 3 å 4 tillsammans med FFNS.</i>			
Tekn. dr. Arne Johnson Ingenjörbyrå	120	Medusa (England) Prime 850 2 Mb 2 x 300 Mb	3
<i>Företaget är ett av de hårdast satsande. Företaget har utbildat personal från arkitektfirman Höjer-Ljungqvist och samarbetar med entreprenadföretagen SCG och SIAB vid utveckling av Medusa-system hos dessa.</i>			
J & W	950	Berit II DEC Vax 11/780 1 Mb 3 x 174 Mb	3
<i>J&amp;W var på 70-talet pionjär med "ritsystemet" Berit (I), som företaget tillsammans med HSB utvecklade från ett transkt system. Med detta som bas har J&amp;W utvecklat det interaktiva systemet Berit II.</i>			
DAPAB Elpgruppen H Hedlund & Co INPROJ KLT Konsult Rejlers Ing. Byrå Wahlinggruppen	1200	Medusa (England) Prime 750 2 Mb 380 Mb	6
<i>En strategisk och kraftfull satsning av sex el- och VVS-konsultföretag. Andra programutvecklare har bl a härigenom anledning att på allvar beakta konstruktörers och arkitekters samarbete med installationssidan. DAPAB svarar för ägarföretagens utbildning och programutveckling samt utför uppdrag åt ägarerna, men också åt konsultföretag utanför ägarretsen.</i>			
White Arkitekter	250	Rucaps (England) Prime 250/2 0,5 Mb 80 Mb	2
<i>Hittills enda göteborgsföretaget som satsat på CAD. Systemet har använts i 3-4 svenska uppdrag.</i>			
Viak	750	DIGIKART (egen utv.) HP 1000 0,5 Mb 85 Mb	5 varav 1 kopplad till stereoinstrument
<i>Viaks egen utveckling började för ca 5 år sedan. Idag har man i produktion ett interaktivt system för kartframställning och system för ytterligare två specialområden på gång.</i>			
Allmänna Ingenjörbyrån	500	Computer-vision (USA) CGP 200X 0,5 Mb 2 x 80 Mb	2
<i>AIB har en arbetsstation med lärskärm för elektronikutveckling (produktion). Vid den andra sker systemutveckling och utbildning, varvid idag prioriteras mekaniksektorn. Utveckling på mark- och konstruktionsområdet följer senare.</i>			
LEB VVS Teknik AB	110	Egen utveckling Prime 350 0,4 Mb 96 Mb	1
<i>Företaget har utvecklat ett eget ritssystem och överväger att koppla detta till likaledes egna beräkningssystem.</i>			

BOA

Konsulttidningen, nr 4, 1982

FIG 9.7 Konsultbranschen och CAD-utvecklingen (hämtad ur Konsulttidningen, nr 4, 1982)

#### 9.4 Användbarheten av CAD, en diskussion

I detta avsnitt ska kortfattat diskuteras vilka förändringar av konstruktionsprocessen som ett införande av CAD innebär. Två speciella frågor skall diskuteras lite mer ingående. Dessa är ritningsframställning samt ekonomi och lönsamhet.

Anledningen att ritningsframställning har valts är att debatten om CAD-system hitintills huvudsakligen varit inriktad på denna aspekt.

Som tidigare nämnts är det de generella systemen som har skapat det stora intresset för CAD. Det är också dessa system som innebär de största förändringarna av konstruktionsprocessen. Detta avsnitt behandlar därför främst de generella systemen och det förefaller naturligt att i samband med dessa diskutera ekonomi och lönsamhet. Detta eftersom CAD-systemen är relativt dyra och att de innebär stora investeringar.

Innan vi särskådar ritningsframställning samt ekonomi och lönsamhet, avsnitt 9.4.1 och 9.4.2, ska dock en del andra intressanta aspekter på CAD behandlas och diskuteras.

I fortsättningen kommer stundtals en del negativa aspekter på CAD att omnämnas. Därför kan det vara på sin plats att redan här fastslå att CAD-utvecklingen givetvis kommer att fortsätta, vilket också är önskvärt. Den kommer dessutom att spridas allt vidare. Vad vi vill peka på är att det idag, trots allt, finns flera frågetecken och tveksamheter. Genom att nämna dessa kan vi kanske verka som en motvikt till den stora PR-apparat som CAD-industrin har använt. Budskapet kan alltså tolkas som att vi ser CAD som något positivt, men inte som något undermedel.

Varför är då CAD lämpligt för byggbranschen? Ovan nämndes att systemen främst har betraktats som ritsystem. Om man jämför byggnadsindustrin med andra industrigrenar så framkommer det att man inom byggnadsindustrin lägger ned stor tid på just ritningsframställning. Att så är fallet är beroende av att enheterna (konstruktionerna) endast produceras i ett fåtal exemplar.

Inom byggnadsindustrin är ofta varje enhet unik, och kräver nya - och många - ritningar, medan till exempel den mekaniska industrin ofta massproducerar enheter, vilket då, relativt sett, kräver färre ritningar per producerad enhet. Om man med CAD kan effektivisera ritningsframställandet så innebär det med andra ord en större effektivitetsvinst för byggnadsindustrin än för den mekaniska industrin.

Att påstå att införandet av dagens CAD-system är kreativitetsbefrämjande är dock tveksamt. Givetvis upplevs systemen som lärorika och spännande när de introduceras. När arbetet blivit mer rutin kanske de upplevs

annorlunda. Att man med CAD skulle testa fler alternativ och prova sig fram framförs ofta som en fördel. Samma skäl framfördes för övrigt även vid introducerandet av datoriserade konstruktionsberäkningar. Det låter visserligen vackert men den bistra verkligheten är den att man varken har råd eller tid till detta. I varje fall inte i den industriella verksamheten.

I kapitel 1 berördes kortfattat vad en ökad datorisering innebär för besluts- och kontrollmöjligheterna. Det påpekades att en ökad datorisering oftast innebär att besluts- och kontrollpunkterna delvis hamnar inom själva datorsystemen. Detta är knappast önskvärt. Emellertid så är riskerna att "datorn" tar över ännu större för totalintegrerade system än de är för CAD-system. Många punkter är dock gemensamma. En utförligare diskussion om detta finns i avsnitt 11.2.4 och i viss mån i avsnitt 10.1.4.

I genomgången av de generella systemen, avsnitt 9.2.1, nämndes att kopplingen CAD-konstruktionsberäkningar idag är svag. Mycket talar dock för att analys- och dimensioneringsprogram kommer att vara en del av framtidens CAD-system, eller åtminstone vara starkt kopplade till CAD-systemen.

Att koppla ihop CAD, analysprogram och dimensioneringsprogram är dock inte så lätt som det kanske låter. För att få effektivitet och flexibilitet krävs en interaktiv samverkan mellan konstruktören och datorsystemet. Förenklat menas med detta att datorsystemet ger ett förslag, till exempel för armering av en platta, som sedan konstruktören antingen accepterar eller modifierar. Vid en modifikation krävs att konstruktionen analyseras om, efter de nya riktlinjerna. På detta sätt itererar man sig fram till en acceptabel lösning. Observera att det är konstruktören som styr processen och tar de väsentliga besluten.

Ett pågående forskningsprojekt i Lund behandlar denna koppling mellan CAD och datoriserade konstruktionsberäkningar. Projektet, "Datorstödd konstruktion", har för att konkretisera det hela valt att betrakta exempelvis analys av en betongplatta.

Forskningen rörande CAD behandlar dock inte endast kopplingen CAD-konstruktionsberäkningar. Vid högskolorna finns ett antal projekt som startat eller som planeras. Dessutom bedriver högskolorna i samarbete med industrin ett antal FoU-projekt. Intressant är att nämna en strävan till samordning mellan olika CAD-system och att ett tillvaratagande av de praktiska erfarenheter som nu finns av praktisk CAD-användning.

En dataöverföring mellan olika CAD-system kan dock ofta vara komplicerad att utföra. Detta beror på att systemen ofta lagrar grunddata på olika sätt i de olika databaserna, se avsnitt 8.3.

Även om forskningsinsatserna beträffande CAD inom byggnadsindustrin har ökat betydligt under de senaste åren finns det de som hävdar att detta inte räcker. De menar att om inte en rejäl FoU-insats snart kommer till stånd kommer vi att halka långt efter i utvecklingen.

#### 9.4.1 Ritningsframställning

Att överlåta ritningsframställningen på datorn har flera fördelar. Om man kunde bortse från den tid det tar att mata in grunddata, vore den onekligen en ofantligt snabb ritare, speciellt för komplicerade ritningar. Dessutom är den oöverträffad vad gäller att producera perspektivritningar. Tyvärr kan man inte bortse från den tid det tar att mata in grunddata, en tid som i många fall kan vara betydande, men sedan utförs ritningarna i det närmaste felfritt. Dessutom är det enkelt och går snabbt att ändra en befintlig ritning.

Vidare erhåller man en stor flexibilitet eftersom man kan välja skala godtyckligt, och ibland även detaljeringsgrad. Detta åstadkoms genom att man arbetar med flera olika nivåer där olika typer av information lagras i de olika nivåerna (väggar, stomkompletteringar, snickerier, El och VVS var för sig), och vid utritning de olika nivåerna kan kombineras godtyckligt. Detta innebär att vissa grunddata, till exempel koordinater för den bärande stommen som behövs för alla ritningar, snabbt och säkert kan tas fram, och alltid bli de samma.

Ur kontrollsynpunkt är därför en datorproducerad ritning att föredra framför en manuell ritad. Detta gäller även för måttsättningen av ritningen eller om ritningen kopplas ihop med en mängdförteckning. För armeringsritningar kan man till exempel, om man litar på programmet, vara säker på att om ett armeringsjärn finns med på ritningen så finns det även med på armeringsspecifikationen. Erfarenheten säger dock att man inte kan vara helt säker, även om det i detta avseende är en betydande skillnad mellan manuellt producerade handlingar och datorproducerade.

Det finns även en del nackdelar med datorproducerade ritningar. En av dessa är att dagens system ofta placerar text och linjer olämpligt. Även om detta inte sker på så många ställen så hittar man det oftast någonstans på en datorproducerad ritning. FIG 9.8 visar en del av en ritning producerad av ett vanligt generellt system. Trots att ritningen har visats upp i PR-syfte så finns det olämpligt placerade textsträngar. Det ska dock påpekas att man i många program i efterhand, manuellt, kan flytta dessa felplacerade textsträngar.



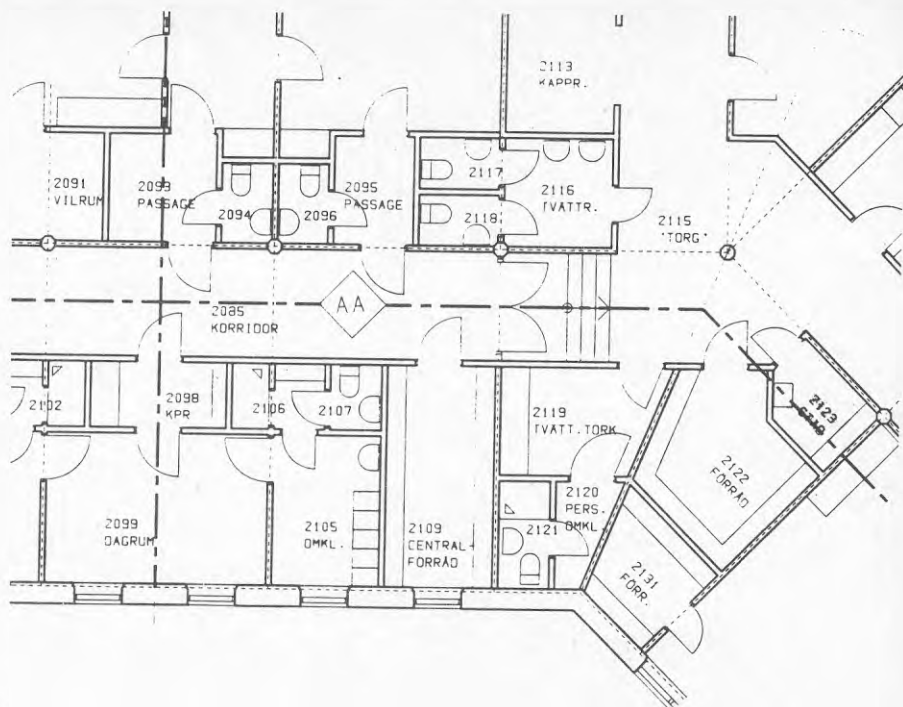


FIG 9.8 Exempel på olämplig textplacering i CAD-ritning

Vad som är lite svårare att lösa är hur man ska bete sig för att producera snygga "arkitekturritningar". Med detta avses snarare skisser än ritningar. En sådan skiss är ofta ritad på fri hand och är "full av liv". Att få datorn att producera "arkitekt-darr" är inte lätt. Det enda man kan åstadkomma med ett ritssystem är ett systematiskt darr, alternativt helt raka streck.

En arkitekt drar ofta vid hörn där olika linjer möts dessutom ut strecken någon millimeter extra. Man har försökt härma detta på datorproducerade ritningar. Dock blev då strecken exakt 0.5mm för långa, vilket inte gav något "liv" åt ritningen.

Vad det är som ger liv åt en ritning är svårt att entydigt definiera. Detsamma gäller en ritnings överskådlighet och betoning. En datorproducerad ritning lägger samma tonvikt på allt, om än med olika tjocka streck, medan en arkitekt eller en driven ritare på något odefinierbart sätt betonar det som ska betonas.

Detta är dock skönhetsaspekter som sannolikt inte kan konkurrera med lönsamhetsaspekter. Idag är det bara ett begränsat antal objekt som det lönar sig att datorrita. Denna andel lär dock växa fort i takt med billigare och effektivare CAD-system och den allt dyrare persontiden.



#### 9.4.2 Ekonomi och lönsamhet

Att ge ett entydigt svar på frågan om ett inköp av ett CAD-system varit lönsamt är inte lätt. Detsamma gäller för ett beslut om vad en CAD-station ska kosta per timme. Det finns åtskilliga mer eller mindre klara punkter som man måste ta i beaktande. Utan att ha några större krav på fullständighet skall här diskuteras en del aspekter beträffande CAD-kostnader.

Kostnaderna ses ofta i ett förenklat perspektiv. Man tar då hänsyn till inköpskostnader, kostnader för utbildning av personal samt till de insatser som lagts ner på egen utveckling och anpassning av systemet. Denna totala kostnad utgör sedan underlag för bestämning av en lämplig timkostnad. Om man räknar på detta sätt, vilket är väl snävt och inte tar hänsyn till positiva bieffekter, blir timkostnaden hög, ibland väldigt hög.

Att timkostnaden blir hög beror på att ett generellt system kan kosta några miljoner kronor i inköp. Till detta kommer sedan siffror av samma storleksordning för utveckling och utbildning.

Att så stora utvecklingsinsatser behövs är beroende på att man vill specialanpassa systemet efter det egna företagets behov och rutiner. Dessutom behövs oftast en anpassning av systemet för byggändamål. Som påpekades i avsnitt 9.2 så är systemen oftast framtagna för andra industrigrenar än för byggnadsindustrin.

Vad gäller utbildning talar man ofta om att det behövs några veckor för att få en CAD-operatör på ett generellt system att bli någolunda effektiv. Denna utbildningstid anges av försäljare av CAD. Sanningen är nog i stället den, att det snarare rör sig om månader än om veckor.

För att täcka de höga kostnader som är förknippade med ett CAD-system krävs naturligtvis en hög omsättning och lämpliga projekt. Det är detta som har medfört att det idag främst är de stora konsultfirmorna som har haft möjlighet att satsa på CAD.

Vad man även bör medta vid en diskussion om kostnader är eventuella bieffekter. Många av dessa effekter är positiva, men svåra att uppskatta i kronor och ören.

En sådan positiv bieffekt är PR. Ett företag som har skaffat sig ett CAD-system och som lägger ned stora utvecklingsinsatser får ett rykte om sig att "vara framtåt". Att företagen lägger stort värde på denna framtoning kan man se av de senaste årens marknadsföring av CAD. Ett gott rykte gör att företaget blir välkänt och lockar till sig kunder och kompetent personal.

Dessutom har det visat sig att CAD är ett viktigt hjälpmedel vid konkurrens på den internationella marknaden. En del länder och uppdragsgivare värderar

inte bara den totala projektkostnaden utan även i hög grad företagens kompetens och kunskap. Som exempel kan nämnas att konkurrensen på utlandsmarknaden uppges ha varit en starkt bidragande orsak till att VBB införskaffade ett CAD-system.

Bland positiva bieffekter kan också nämnas att en satsning på CAD idag innebär att steget blir mindre, och mer smärtfritt, när nästa generation av CAD-system dyker upp, en följd av att man bygger upp en allmän datorkompetens inom företaget, vilket får till följd att man successivt anpassar konstruktionsprocessen till en datorstödd verksamhet.

Vid ett eventuellt inköp av ett system tas dock en viss risk. Systemet kan visa sig vara svårt att anpassa till det egna företaget och dess verksamhet. Dessutom kan det visa sig att det system man har valt försvinner från den svenska marknaden. Detta innebär dålig assistans och att det kanske inte kommer några uppdateringar av systemet.

Det nämndes tidigare att ett 20-tal olika CAD-system finns tillgängliga på den svenska marknaden. Denna siffra är så hög att det är sannolikt att en del av systemen försvinner. Det är dock mångas förhoppning, inklusive författarnas, att inte alltför många system försvinner från marknaden. Utvecklingen främjas nog inte av att något system blir helt marknadsdominerande.

Vid en diskussion om kostnader kan det vara av intresse att jämföra situationen i Sverige med situationen i USA. Ingenjörslönerna är relativt sett högre i USA medan kostnaden för ett CAD-system, inklusive hårdvara, är lägre. Dessutom är det i USA vanligt att man bemannar en CAD-station i treskift. Detta är nog otänkbart i Sverige. Följaktligen gäller att om CAD inte är lönsamt i USA så är det än mindre lönsamt i Sverige.

Frågan är när ett företag ska hoppa på "CAD-tåget". Ska man vänta på effektivare och billigare CAD-system eller ska man utnyttja CAD på servicebasis eller rent av köpa ett system tillsammans med några konkurrenter? Svaret på den frågan överlåter vi åt företagsledarna ute i landet.

Ett alternativ kan vara inköp av ett billigare specialsystem. Som nämndes i avsnitt 9.2.2 är inköpskostnaderna för de "nedskalade" generella CAD-systemen av storleksordningen hundratusentals kronor. Till detta kommer det faktum att utbildningen på dessa system tar väsentligt kortare tid i anspråk. Avgörande är om man behöver de generella systemens flexibilitet och generalitet.

Givetvis är egen utveckling också ett alternativ. På detta sätt kan man få systemet specialskrivet för det egna företagets behov. Att skriva program är dock både dyrbart och tidsödande. Om man inte har en befintlig,

speciell kompetens, och viss befintlig programvara, är det nog mycket svårt att få ekonomiska förutsättningar för en sådan verksamhet.

Slutligen skall vi, om än försiktigt, våga oss på att sia något om framtidens system.

Vid utvecklingen av de nya systemen kommer man att försöka ta tillvara alla de synpunkter som har framkommit vid den praktiska användningen av CAD.

Nästa generations CAD-system kommer därför sannolikt att vara mer uppdelade i moduler, det vill säga bestå av mer eller mindre fristående delar som godtyckligt kan sammanfogas med varandra och med andra program.

Systemen kommer dessutom att kunna klara sig med mindre och billigare datorer. Detta eftersom hårdvaran ständigt utvecklas. CAD-systemen blir därför billigare, i relativa belopp mätt. Till detta bidrar även den hårdnande konkurrensen mellan de olika systemen.



## 10 RISKER MED DATORANVÄNDNING

I de tidigare kapitlen i denna bok har vi redogjort för de möjligheter som datorn ger oss att lösa komplicerade problem i byggnadskonstruktionsarbetet. Efter en sådan genomgång är det frestande lätt att tro att allting går att lösa med datorns hjälp. Och i och för sig är detta kanske riktigt, men datorn kan ingalunda göra detta själv - det krävs att de som programmerat och använder datorn är mycket kunniga, för att svåra problem skall kunna lösas, och dessutom att de två - då det vanligen inte är en och samma person - kan kommunicera på ett bra sätt.

Vi vill nu med de två kapitlen som återstår i boken också litet grann diskutera de problem och de risker som är förknippade med en snabbt ökande datoranvändning. Vi skall för detta dela problematiken i två delar. I detta kapitel skall framförallt diskuteras de risker som kan föreligga i en datoranvändning av nuvarande typ, medan kapitel 11 speciellt behandlar de möjligheter och risker som kan föreligga med en långt driven totalintegrering av konstruktionsberäkningarna.

Att vi ger dessa problem ett relativt stort utrymme skall naturligtvis inte tolkas så att vi är negativt inställda till datorn och dess möjligheter, vi vill bara försöka ge litet motvikt till den nyvunne entusiastens målsättning att "datorisera mera", och därmed stämma till en viss eftertanke.

### 10.1 Konstruktörens arbetssituation

Som en bakgrund till en diskussion om riskerna med en ohämmad datoranvändning skulle vi vilja ta tre ögonblicksbilder som visar konstruktörens arbetssituation "igår" (före datorutvecklingen), "idag" respektive "i morgon". Det skall dock genast påpekas att tidsperspektivet i denna framställning är mycket förenklat då utvecklingen har gått skiftande långt inom olika företag.

#### 10.1.1 Före datorisering

Konstruktörens arbetssituation i en icke datoriserad miljö kan beskrivas med FIG 10.1. I denna sitter konstruktören själv mitt i sitt arbete, omgiven av de informations-"depåer" han har behov av för verksamheten. Dessa utgörs främst av dels handböcker och normer, dels något räknehjälpmedel.

En hel del ytterligare information måste dock tillföras för att han skall klara sitt arbete med ett visst projekt. Denna karakteriseras av de övriga rutorna i



diagrammet, och innefattar mera subtila begrepp som "modeller", "erfarenheter" och "ideer". Även dessa spelar dock en stor roll vid utformningen av en konstruktion.

De olika informationsdelarna kommer att resultera i en lösning av det aktuella problemet. De tankesteg som lett fram till den färdiga lösningen finns redovisade på papper som en logisk följd av tankar och beräkningar. Att så är fallet är viktigt eftersom detta möjliggör en granskning av den gjorda analysen, såväl vad gäller de gjorda förutsättningarna som då det gäller själva beräkningarna.

Att granskning är möjlig är viktig ur flera synpunkter. Det är ju nämligen inte bara myndigheter, främst i form av byggnadsnämnden, som skall ha möjlighet att göra detta, utan det kan lika gärna vara konstruktören själv som i samband med någon ändring vill gå tillbaka till sina tidigare beräkningar. Andra kategorier som kan ha intresse av att kunna kontrollera en beräkningsgång är, till exempel, beställaren och en eventuell skadestredare.

Med denna skisserade arbetssituation sitter alltså konstruktören själv som "spindeln i nätet" och bearbetar all sin bakgrundsinformation till en färdig analys, där varje tankesteg är klart redovisat.

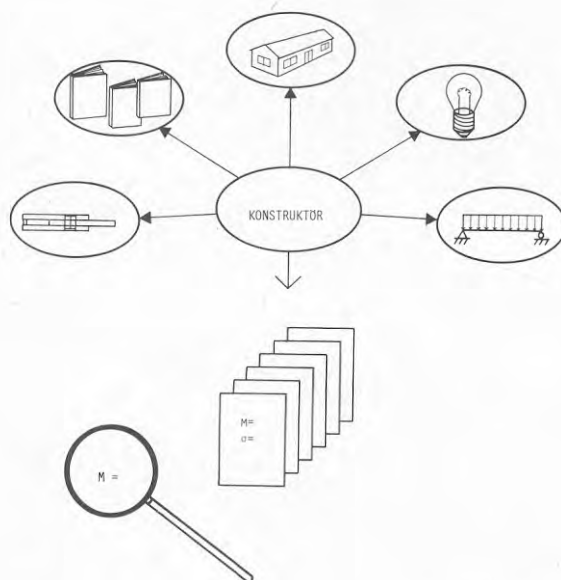


FIG 10.1 Konstruktörens arbetssituation före datorisering

### 10.1.2 Under datorisering

I och med utvecklingen på datorsidan förändrades arbetssituationen i viss mån. Datorn övertog till en början främst räknehjälpmedlets roll då program utvecklades för analys av relativt enkla delstrukturer, till exempel fackverk och kontinuerliga balkar. För dessa enkla beräkningar är steget från konstruktörens tänkta modell till programmets indata relativt kort genom att programmets möjligheter är begränsade, och att programmets arbetssätt främst är en formulering på datorspråk av kända beräkningsmetoder. I stället för att beräkningen skett helt och hållet på papper, utbyttes vissa mindre delar mot: formulering av indata, körning av program, och läsning av resultat. De resultat som producerades var dock fortfarande på låg nivå, till exempel stödmoment och maximala fältmoment för den kontinuerliga balken. Datorn har på detta stadium främst ersatt räknehjälpmedlet, men även, i viss mån, de tabellverk som tidigare använts.

Trots införandet av datorer på denna "primitiva" nivå är konstruktörens arbetssituation i stort sett oförändrad. Han har fortfarande hela beräkningsgången i sin hand och får lov att utnyttja all sin bakgrundsinformation - den största förändringen är att vissa enkla beräkningar går något fortare.

### 10.1.3 Efter datorisering

Strävanden finns dock att överflytta mer och mer till datorn. Som visats i kapitel 7 så utvecklas ju exempelvis dimensioneringsprogram som tar hänsyn till de gällande normerna för olika strukturer. Projekt pågår också för att lägga upp till exempel AMA-böcker och normer i form av databaser som är åtkomliga från olika konstruktionskontors terminaler.

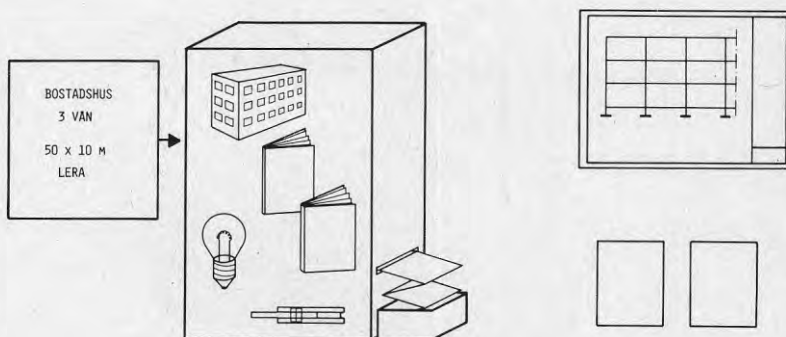


FIG 10.2 Visionen "Datorn klarar allt"

De strävanden som är på gång tycks, om de skrivs fram ett antal år, ha som mål att uppnå det tillstånd som visas i FIG 10.2, en vision där datorn klarar allt - hela analysgången från beslutet att man vill bygga ett hus till framställandet av färdiga arbetshandlingar. (Observera att vi inte ens i denna vilda vision tar med någon form av datorstödd produktion).

#### 10.1.4 Utvecklingslinjer

Datorn har kommit för att stanna på konstruktionskontoren, det är nog alla ense om. Men, vart går utvecklingen, sett mer i detalj. Vi tror att man för den närmaste tiden kan spåra två, eller eventuellt tre, olika utvecklingslinjer.

För det första, har man den allmänna spridningen. Även de små konstruktionskontoren, som ännu inte datoriserats i någon större omfattning kommer att dras med i suget, om inte annat genom att datorberäkningar kommer att kunna bli ett konkurrensmedel, både direkt genom möjligheterna att lösa många uppgifter snabbare och effektivare, men också indirekt på så sätt att man för många projekt kommer att kräva en datoranalys för att kunna godkänna en konstruktion. Tendenserna mot ett allt högre utnyttjande av materialens hållfasthet kommer också att tvinga fram alltmer förfinade analysmetoder, som inte kan realiseras med hjälp av enbart handräkning.

I denna spridning av datorresurserna finns flera möjliga vägar. För de större firmorna kommer säkert det ökade datorutnyttjandet att leda till etablerandet av speciella datoravdelningar med egna, stora datorer. För mindre företag blir kanske persondatorerna, med deras snabbt växande kapacitet, en annan möjlighet. Det förefaller dessutom möjligt att speciella data-servicebolag inriktade på byggbranschen, enligt modellen i avsnitt 3.1.2, kommer att dyka upp, även om några sådana inte direkt existerar idag.

Den andra utvecklingstendens man kan se är utvecklingen på den grafiska sidan, framförallt då när det gäller presentation av beräknade resultat i form av färdiga arbetsritningar. Strävanden finns också att koppla de program för design (med detta menar vi främst det direkta arkitektoniska skapandet) som redan finns till olika beräkningsprogram, med den slutliga visionen att sammankoppla stegen design-beräkning-redovisning till ett totalintegrerat system. Några synpunkter på denna utvecklingslinje kommer, som tidigare nämnts, att ges i kapitel 11.

En tredje linje för utveckling som borde kunna vara intressant, men där veterligt inte så mycket gjorts ännu är skapandet av databas-system för lagring av all information om ett byggprojekt. Detta skulle möjlig-

göra för alla som deltar i ett projekt att arbeta mot samma informationsmängd och alltid ha tillgång till den senaste uppdateringen av både egna och andras delprojekt. Möjligheterna för skapandet av sådana system behandlades i viss mån i kapitel 8 och skall inte närmare beröras här.

En alldeles säker tendens är att allt flera personer på konstruktionskontoren, och i andra delar av byggbranschen, kommer att påverkas av datorerna. Från att ha varit ett verktyg för några få personer inom några företag, kommer alla att få lov att använda olika datorer och program. Detta innebär i sin tur att ett stort antal personer utan någon utbildning på och vana vid datorer kommer att arbeta med datorkörningar. Att detta kan medföra vissa risker för fel är uppenbart. I fortsättningen av detta kapitel skall vi försöka diskutera dessa risker och vad som bör göras för att riskerna med datoranvändningen skall minimeras.

Låt oss för den skull litet grann återknyta till FIG 10.1 som visade en konstruktörs arbetssituation före en datorisering. Om vi antar att datoriseringen drivits så långt att datorn kan klara såväl analys- som dimensioneringsstegen, som beskrevs i kapitlen 6 och 7, kan konstruktörens arbetssituation kanske i stället beskrivas av FIG 10.3. I denna figur är utvecklingen förmodligen kraftigt överdriven, om man ser situationen inom den närmaste 10-årsperioden, men den skall efterlikna den situation som uppstår om utvecklingslinjerna hårdtras. Figuren kan därför ses som en överdrift av en situation som kan uppstå ganska snart, om strävandena att datorisera alltmer får genomslag.

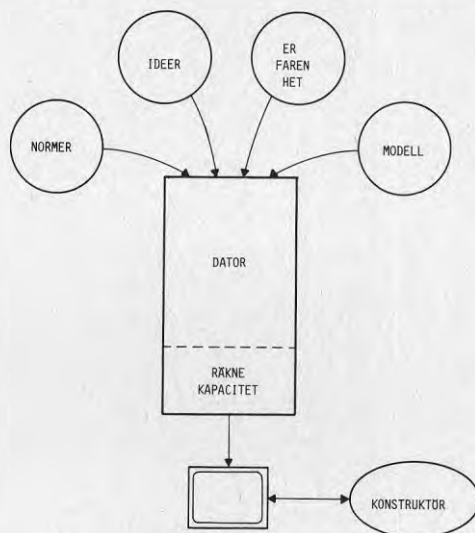


FIG 10.3 Datorn som informationsbärare

Man ser ur figuren hur datorn fått ta över som informationsbärare. Alla de informationsdelar som konstruktören behöver i sitt arbete har förts in i datorn, inte bara den räknekapacitet som utnyttjats under lång tid utan även normerna och föreskrifterna. Med hjälp av en avancerad databashantering kan man naturligtvis även lägga in erfarenheterna, det vill säga information om tidigare utförda projekt, ett strävande som med tiden resulterar i en bank av standardlösningar för olika byggnadsdelar.

Det som då är intressant att notera är att datorn naturligtvis inte kan allt detta av sig själv. Någon måste ju någon gång ha försett den med informationen, så den situation som visas ganska positiv i FIG 10.3 kan lätt förtydligas som i FIG 10.4, med en betydligt mindre attraktiv framtoning. Det är en enda person, eller kanske snarare ett lag, som matat datorn med sina ideer och sina erfarenheter. Att detta inte är direkt befrämjande för kreativiteten förefaller rätt klart. Detta accentueras naturligtvis än mer av den påvisade tendensen att allt fler personer kommer att använda datorn utan att själva behöva skriva sina program. Såväl genom uppväxandet av data-sevicebolag som genom spridandet av kassetband för persondatorer kommer ju antalet programanvändare snabbt att växa i förhållande till antalet programskapare, se FIG 10.5. De oundvikliga fel som alltid finns i alla program kan på detta sätt få en stor genomslagskraft.

## 10.2 Informationsflöden

I det förra avsnittet skisserades en tänkt arbetssituation för konstruktören i det skede då datorisering-

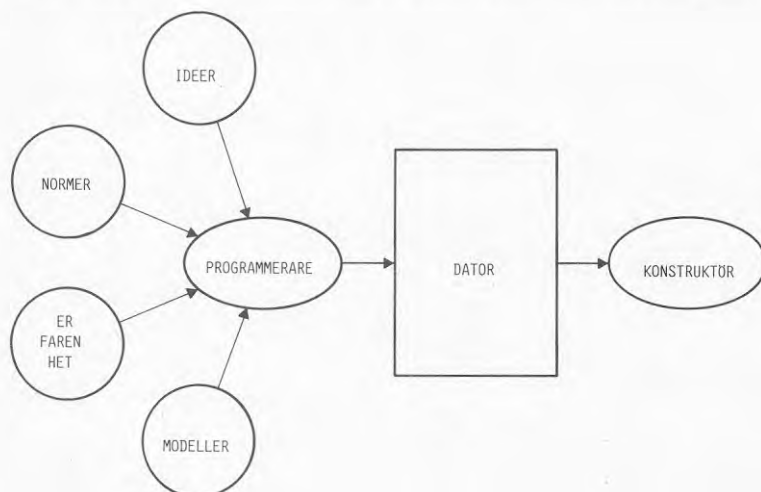


FIG 10.4 Variant av FIG 10.3



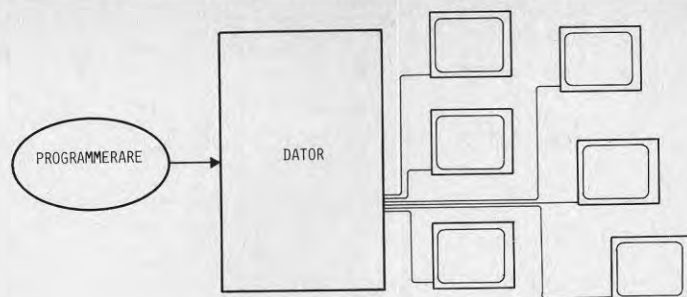


FIG 10.5 Koppling mellan användare och programmerare

en gått relativt långt. Vi noterade att det kan medföra vissa risker att konstruktionsarbetet blir alltför beroende av ett fåtal personers ideer (och deras förmåga att omforma dessa till korrekta program). I de följande avsnitten skall vi mer i detalj försöka kartlägga var risker för felaktigheter i samband med datoranvändning föreligger.

Som en bakgrund för detta skall vi dock först försöka klarlägga det informationssammanhang i vilket datorberäkningen ingår. Den situation vi då kommer att diskutera är en, där relativt avancerad datorbearbetning ingår i konstruktionsarbetet (även om diskussionen i och för sig är tillämpbar även för den mindre skalan).

I FIG 10.6 visas en schematisk bild över de informationsflöden som förekommer. Med datorn satt som medelpunkt kommer konstruktören att befinna sig på ena sidan av datorn och programmeraren på den andra. Den enda förbindelse som finns mellan dessa är dels den information som förmedlas direkt av datorn (dialog vid inmatning av data till ett program, felmeddelanden o s v), dels den programbeskrivning och den användarinstruktion som hör till programmet.

Med hjälp av den information som på detta sätt överförs från programmeraren skall användaren försöka skapa indata till det aktuella programmet. Dessa indata skall då återge den beräkningsmodell som konstruktören vill använda på ett sätt som programmeraren avsett, och med hänsyn till de av programmeraren antagna konventionerna. I många fall innebär detta att en komplex "tänkt" verklighet måste brytas ned och anpassas till de möjligheter som programmet medger.

Med hjälp av det program och de indata som datorn försetts med kommer den att göra en beräkning, som resulterar i vissa utdata. Dessa måste dock, åtminstone i viss utsträckning, bearbetas (tolkas) innan de kan sägas utgöra resultat från beräkningen, ett arbete som faller på konstruktören.

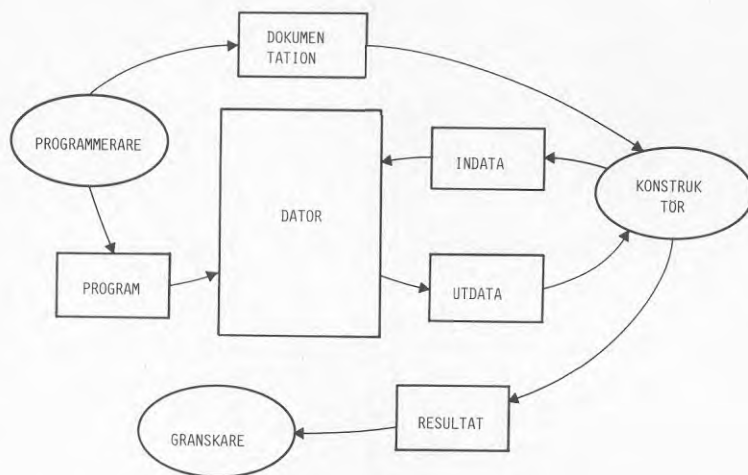


FIG 10.6 Informationsflöden vid datorbearbetning

Efter detta skall, förhoppningsvis, de framtagna resultaten överensstämma med de resultat som skulle ha åstadkommit med en handräkning, och den enda skillnaden mot det informationsflöde som visats i FIG 1.1, är att ett antal mellansteg i beräkningen saknas, fler ju mer komplicerat det använda programmet är.

Samtidigt som man då konstaterar att själva beräkningsgången har blivit svårare att följa i detalj, inser man dock också att, så länge programmet antas var riktigt, behovet att kontrollera varje steg i beräkningen minskat. Kontrollen borde i stället kunna fokuseras på de antaganden och modelleringar som gjorts.

Granskarens roll har därmed i viss mån förändrats. Med datorberäkningar införda i beräkningsgången kommer en del sifferdata som tidigare skrivits ner inte längre att finnas tillgängliga. Det blir i stället viktigt att kontrollera de gjorda antagandena, och att söka bedöma det använda programmets tillförlitlighet och användbarhet. I den visade figuren framgår ju också att granskaren kan ha svårt att få fram någon information från programmeraren, exempelvis om hur programmets resultat skall tolkas.

### 10.3 Felkällor

Med FIG 10.6 i minnet skall vi nu försöka hitta de möjligheter till fel som uppstår i datorberäkningen. För

att strukturera diskussionen något skall vi dela in de visade informationsflödena i ett antal delar, och behandla varje del för sig. Vi definierar då följande skeden i FIG 10.6 för den vidare diskussionen:

- 1) Programmeringen
- 2) Indatagivningen
- 3) Beräkningen
- 4) Resultattolkningen

### 10.3.1 I programmeringsskedet

Programmerarens uppgift är att översätta en beräkningsalgoritm till en följd kommandon som datorn skall följa. Till sitt förfogande har han för denna uppgift vissa bestämda formuleringar, och som visats i kapitel 2 vissa bestämda möjligheter att lagra data. Med hjälp av dessa skall han alltså åstadkomma en beräkningsmodell som på ett så generellt sätt som möjligt kan beskriva en struktur.

Programmeraren måste alltså välja, dels ett sätt att i ren sifferform beskriva en struktur, dels den beräkningsmetod han vill använda. Uppenbarligen sätter dessa val upp begränsningar för vad som går att åstadkomma med programmet.

När det gäller metoder så finns ett stort antal att välja på. Som diskuterades i avsnitt 6.1 kan ju nämligen en så enkel struktur som en kontinuerlig balk beräknas med hjälp av många olika metoder, till exempel

- Clapeyron's ekvation
- Primärmomentmetoden
- Cross' metod
- Kraftmetoden
- Förskjutningsmetoden
- Finita Element metoder (olika)
- Arbetsekvationer
- Gränslastmetoden
- Funktionsansatser

I och för sig är ju flera av dessa metoder egentligen sammanfallande, men skiljer sig i fråga om de vanligen använda beteckningarna, eller i beräkningsmetodik. Med hänsyn till detta kommer programformuleringar av metoderna eventuellt att skilja sig något. Vissa av metoderna avviker ju också helt från de övriga.

För att ytterligare komplicera det hela kan man, inom de olika metodernas ram, ibland formulera sina beräkningsförutsättningar på olika sätt. Med exempelvis en Finit Element formulering kan beräkningen av kontinuerliga balkar ske med ett stort antal olika antaganden, till exempel

- Enbart böjningsarbete
- Även tvärkraftsarbete
- Även normalkraftens arbete
- Andra ordningens teori (stabilitet)
- Ovanliga lastfördelningar
- Egenvikt av balken
- Voter
- Elastiskt underlag / elastisk inspänning
- Temperaturdifferenser eller andra tvångskrafter
- Uppsprickning av betong / plasticering

Uppenbarligen kommer de olika metoderna och de olika antagandena att ge olika svar på exempelvis nedböjningen av balken, eller momentfördelningen, utan att för den skull det ena eller det andra resultatet kan sägas vara mera rätt än de andra.

Att man får olika resultat är ju också riktigt och inget speciellt för datorberäkningen. Vad som dock är viktigt är att man får veta vilken metod som använts och vilka antaganden som gjorts. En konstruktör som räknar för hand väljer ju nämligen en beräkningsmetod som är lämplig för det aktuella fallet, och tar hänsyn till de relevanta effekterna. Vid skapandet av ett generellt program för beräkning av kontinuerliga balkar är detta "individuella" hänsynstagande betydligt svårare.

Förutom dessa val av beräkningsmodell måste programmeraren göra flera approximationer och förenklingar. Det kan till exempel röra antalet punkter som medtas vid en numerisk integration för bestämning av stödvinkeländringarna för en votförsedd balk.

När det gäller mera avancerade program, exempelvis för dimensionering måste också programmeraren ge datorn ett selektivt val av alternativ. Datorn måste till exempel få direktiv om vad den skall ändra i första hand om en provad betongbalk inte räcker till i fråga om bärförmåga eller styvhet - skall balkens höjd ändras, armeringshalten ökas eller betongkvaliteten höjas. Även om programmet är så sofistikerat att det kontrollerar alla dessa åtgärder och gör någon optimering mellan dem, måste någon ha försett den med vissa preferenser eller vissa optimeringskriterier, och i dessa kommer programmerarens egna erfarenheter att slå igenom.

De punkter som hittills räknats upp i detta avsnitt skall naturligtvis inte betraktas som fel. De val som görs vid programskrivningen är ju riktiga i sig själva. De kan dock vara potentiella felkällor om konstruktören använder programmet för beräkning av något som ligger långt från den tillämpning som programmeraren i första hand tänkt sig. Som exempel på detta kan nämnas antaganden om huruvida tvärkraftsdeformationer är värda att beaktas i en konsolbalk. Som en grundregel kan ju gälla att dessa är försumbara, men i ett speciellt fall med en kort, hög balk kan de bli intressanta. Det viktiga är då bara, som vidare diskuteras i avsnitt

10.4, att konstruktören kan få ett klart besked hur programmet gör. Detsamma gäller för de konventioner som programmet innehåller. Den förutsättning som programmet gör i fråga om positiva utböjningsriktningar och moment avviker ofta (speciellt i många Finita Element program) från dem som konstruktören är van vid.

Programmeraren kan också föra in olika former av begränsningar i sitt program. De har i många fall ett samband med den beräkningsmodell som valts men kan även vara av annan natur. Exempelvis kan ett program för armering av balkar ha en inbyggd förutsättning att endast ett lager armering kan användas, eller att tvärsnittet ej skall vara tryckarmerat.

Andra begränsningar har med lagringen av data om den beräknade strukturen att göra. Som ett exempel på detta kan nämnas att programmeraren förutsatt att mer än 20 fack aldrig kan bli aktuellt för en kontinuerlig balk. Om programmet sedan inte förses med tillräckliga spårar mot att fler fack definieras kan de beräknade resultaten bli grovt felaktiga.

Med detta är vi inne på direkta fel i programmen. Ett gammalt talesätt inom databranschen säger att det inte finns några felfria program, och detta är säkert sant för alla program av en viss storlek. Sedan är det ju en annan sak att felet kanske inte visar sig annat än för ytterst speciella kombinationer av indata. Men skäl för vaksamhet finns alltid. Det är inte alltid så, även om det är mycket vanligt, att de fel som programmet gör leder till katastrofalt felaktiga resultat.

### 10.3.2 I indataskedet

Det finns, tillspetsat uttryckt, ingenting som program- och datafolk är så dåliga på som att beskriva sina program. Åtminstone gäller detta den beskrivning av programmet som den potentielle användaren får i sin hand då han skall köra programmet. Som har diskuterats i avsnitt 3.3 så visade programinventeringarna som gjorts, att ganska avancerade program i vissa fall beskrivs med hjälp av så litet som en enda sida användarinstruktion.

Som diskuterats tidigare kan ett program ta hänsyn till ett visst antal effekter och modellera ett visst antal strukturer. De modeller som kan användas överensstämmer oftast endast på ett ungefär med den tänkta verklighet som konstruktören har att arbeta med. En anpassning måste alltså ske mellan den tänkta strukturen och en modell som kan formuleras i programmet. Den figur över arbetsgången som vi visade i början av kapitel 1 är därför i realiteten något mera komplicerad. Att modellera en viss struktur med hjälp av programmets möjligheter blir därför väsentligen ett iterativt förfarande där verkligheten skall anpassas till prog-



rammets möjligheter. Detta innebär att skedena "Modell - Indata" i FIG 1.1 snarare skall beskrivas med FIG 10.7.

Den som är mycket hemtam med ett program kan, med hjälp av en hel del "knep och knåp" till exempel genom att ge fiktiva materialparametrar o dyl, på detta sätt behandla strukturer som egentligen ligger långt utanför det tilltänkta tillämpningsområdet för programmet.

Detta leder naturligt till slutsatsen att beskrivningen av den beräkningsmodell som ett program använder är den allra viktigaste komponenten vid kontroll av resultatens giltighet. En hel del synpunkter på detta med programdokumentationer kommer att lämnas i avsnitt 10.4. I detta avsnitt skall vi nöja oss med att påpeka vissa vanliga brister som kan leda till missförstånd och felaktigheter.

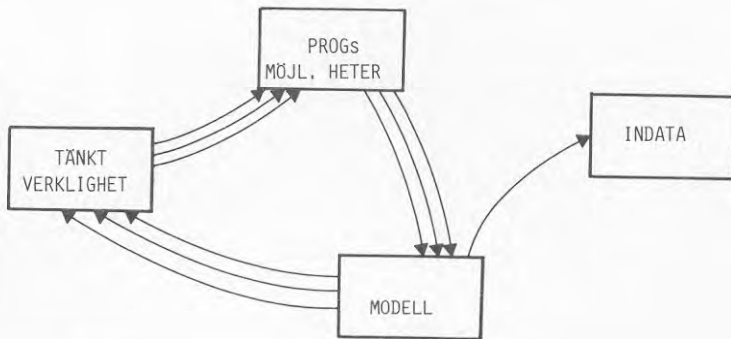


FIG 10.7 Förfining av delar av FIG 1.1

Den första upplysning man som eventuell användare behöver är information om programmets begränsningar. De begränsningar som då är aktuella är inte bara de fysiska begränsningarna som programmet (i den aktuella datorn) har, det vill säga hur många element, fack eller dylikt som datorn kan hantera. Viktigare är väl dock kanske ändå de begränsningar i beräkningsmodellens giltighet som programmet påtvingar användaren - kan elastiska inspänningar, uppsprickning av betongen och dylikt modelleras på ett tillfredsställande sätt. Denna information är viktig att få på ett tidigt stadium då det har betydelse för valet av program. När man fått fram upplysningar om detta kan man gå ner på den "lägre" nivå och kontrollera de fysiska begränsningarna, som dock oftast inte är så besvärande (utom vid användning av små persondatorer) att de sätter käppar i hjulet för en eventuell användning.

I samband med detta val av ett lämpligt program är det också viktigt att ta reda på hur programmet kan nyttjas. Från exempelvis de gjorda programinventeringarna får man, i bästa fall, upplysningar om ifall programmen säljs för körning på en egen dator ( i källkodsform), eller om man får köra programmet på programleverantörens dator. I så fall måste man också få reda på om programmet kan nås från den egna terminalen, eller om man måste förflytta sig till en datorcentral, för att där köra programmet till exempel med hjälp av kortbuntar med program och data.

När man hittat ett lämpligt program, och konstaterat att man kan komma åt att utnyttja det är nästa steg att få fram upplysningar om hur en viss beräkningsmodell skall representeras, det vill säga hur datorn kan fås att tolka användarens intentioner på ett riktigt sätt. För detta måste en användarinstruktion (oftast benämnd manual, från engelskan) utnyttjas. Som har visats i kapitel 3 är dessa av mycket varierande omfattning, från praktiskt taget ingenting till flera tusen sidor. Det senare gäller framförallt de stora, generella Finita Element programmen. Allmänt kan man nog, för övriga program, säga att dessa användarinstruktioner är i knappaste laget, i det att de främst beskriver något typfall.

För små program, med begränsade möjligheter är detta kanske tillfredsställande, då man relativt enkelt med ledning av ett exempel kan formulera data för det egna problemet. Tyvärr får man dock på detta sätt ingen uppfattning om de teorier som används i programmet (jämför de olika möjligheterna som visades i avsnitt 10.3.1), vilket kan leda till svårigheter då man vill försöka kontrollera de beräknade resultaten.

I och med att programmets möjligheter växer blir man dock alltmer beroende av en hel del bakgrundsmaterial, om teorier som används och representation av beräkningsmodellen. Som vi redan på ett tidigt stadium i denna bok konstaterade, se kapitel 6, bör man ju nämligen inte använda program för beräkningar, som man inte, åtminstone i princip, skulle kunna klara av att göra manuellt. Allt annat måste ju rimligtvis ha väldigt mycket av "hokus-pokus" över sig.

Man måste också med hjälp av beskrivningen få en del bakgrund och rekommendationer för de val man måste göra när man formulerar sina indata. Vi kan ta ett exempel från avsnitt 6.3, nämligen en tänkt Finit Element beräkning av en platta. Antag att det program vi vill använda frågar efter antalet Gauss-punkter vi skall ta med i den numeriska integrationen av styvhetsmatrisen. Med bakgrund av den generella sanning som diskuterades i kapitel 5 väljer man naturligtvis detta antal till ett högt tal (med den i och för sig hedervärda avsikten att ernå så goda resultat som möjligt). Det är då viktigt att användarinstruktionen för programmet talar om att man, om man utnyttjar ele-

ment som bygger på Mindlin's platt-teori inte får välja antalet alltför högt, med hänsyn till risken för så kallade "locking"-effekter.

Ett problem med många användarinstruktioner är att de är skrivna på ett svårförståeligt språk. Att man som dator-nyttjare måste vara beredd på att få över sig en mängd information på engelska tycks vara i det närmaste oundvikligt, det är oftast inte det värsta språkproblemet. Värre är ofta den orimliga inblandning av "data-ord" som utnyttjas i användarinstruktionerna. Ofta fyller dessa ingen vettig funktion för den vanliga användaren eftersom en uppgift om hur många element och noder ett program kan hantera säger betydligt mer än en primärminneskapacitet uttryckt i kilo-byte (se kapitel 2) .

Ovan har vi kritiserat programmets användarinstruktioner för att vara knapphändiga och ge en dålig bakgrund för användaren. Vi skall dock nu även kritisera de stora Finita Element programmets dokumentationer för att vara alltför omfattande. Det är dock inte omfattningen i sig själv utan snarare strukturen på informationen som ofta är olämplig. Man blandar nämligen rätt ofta information om den teoretiska bakgrunden med nyttiga råd och mallar för uppställning av indata på ett sätt som gör att den information man för tillfället söker har en tendens att drunkna i alla övriga upplysningar. Hellre än att offra en timme på att leta, hoppas man att programmet uppfattar en viss uppgift på samma sätt som man själv vill se den.

Bakgrunden till denna stora mängd information är naturligtvis dessa programs allmängiltighet. Med ett och samma program kan man beräkna statiska och dynamiska förlopp, värmeledning eller fukttransport, och dessutom ta hänsyn till ett mycket stort antal effekter (krypning, svällning, temperaturdifferenser och många andra). Alla dessa effekter måste man ta ställning till, vilket gör att en mängd styrinterinformation måste matas in om fenomen och teorier man kanske knappt hört talas om.

Låt oss för en kort stund betrakta det stora Finita Element programmet ANSYS, vilket är ett av de mest allmängiltiga programmen som finns på marknaden. Med det kan man lösa praktiskt taget alla problem som är möjliga att formulera i Finit Element form. En manual för detta program fyller två normala pärmar, och då har man ändå endast den del som kan betecknas som användarinstruktioner - den teoretiska bakgrunden finns i en annan skrift.

I FIG 10.8 visas en del av ett schema hur indata skall formuleras. Indata kan ges såväl kolumnbundet som i fritt format. Schemat, som i sig självt fyller två sidor, kompletteras med en kort förklaring för varje parameter (totalt 79 sidor), och en detaljerad indata-beskrivning om totalt 56 sidor. Till detta kommer de

ANSYS INPUT DATA SUMMARY

CARD	FIELD																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	Title - (20 A4 (Alpha-numeric) Fields) <sup>†</sup>																			
B	Name - (4 A4 Fields) <sup>†</sup> NONT    NSTP    A4 <sup>†</sup> A4 <sup>†</sup> --    IEQM    --    ICOR																			
C1	NCHK	KAN	KCE	KCDF	KTB	KNL	KRF	KBC	KGP	KSE	KED	KPRE	KPST	KAY1	KAY2	KAY3	KAY4	KAY5	KAY6	KAY7
	KAY8	KAY9	KY10	--	--	--	--	NOFS	--	--	KPRP	KNR	NOCL	KAPO	(These values are in Fields 21-34)					
C2	TREF	TUNF	ALPD	BETD	--	KRST	TOFS	--	KMOD	KORD	KGEO									
D	ITYP	JSTF	K1B	K1A	K1	K2B	K2A	K2	NJ	INPR										
DA	See Chapter 5 (Preprocessing)																			
D2	RC1	RC2	RC3	RC4	RC5	RC6	RC7	RC8												

FIELD COLUMNS    VARIABLE    MEANING

A CARD \*\*\* PROBLEM TITLE

1-80    IHEDD    TITLE FOR OUTPUT. IF COLUMNS 77-79 ARE LEFT BLANK AND A COMMA PUNCHED IN COLUMN 80, THE TITLE MAY BE CONTINUED ON THE FOLLOWING CARD. NO LIMIT.

SLASH (/) CARD DIRECTIVES FOR SPECIAL PROBLEM CONTROLS MAY ALTERNATELY BE INPUT AT THIS LEVEL. SEE SECTION 2.26.

B CARD \*\*\* NAME CARD (FREE-FORMAT NOT USED) ON THIS CARD SINCE NAME AND IACCNT FIELDS ALLOW COMMA AS A VALID ALPHA-NUMERIC CHARACTER)

1-16    NAME    (OPTIONAL) USFR IDENTIFICATION NAME

18    NONOTE    0 - PRINT NOTES (NEW FEATURES, MODIFICATIONS, ANNOUNCEMENTS, ETC.) AT END OF SOLUTION  
1 - SUPPRESS PRINTOUT OF NOTES (NOT RECOMMENDED)

19-24    NSTEPS    0 - LOAD STEPS (CARDS L-Q) END AT L CARD TERMINATOR  
N - TERMINATE LOAD STEPS AFTER N (OFTEN USED TO PARTIALLY READ PREVIOUSLY GENERATED LOAD STEP FILE)

25-32    IACCNT    (SYSTEM OPTION) ACCOUNT NUMBER. SEE APPENDIX S

37-42    IEQMAX    (OPTIONAL) MAXIMUM NUMBER OF EQUATIONS USED IN WAVE FRONT (TO CHECK FOR ADEQUATE CORE STORAGE)

73-80    ICORE    (SYSTEM OPTION) CORE SIZE PARAMETER. SEE APPENDIX S

Maximum Number of Equations Used (IEQMAX). This optional bit of information is intended to be a safeguard against a job terminating because of insufficient core storage during the wave front procedure. The model check problem (see Section 2.7), or any solution problem, prints the maximum equations used, the maximum equations allowed for the requested core storage (see this card, ICORE), and the core storage or data area required for this problem. The maximum equations used may then be input for IEQMAX of the solution problem to insure that the program will not start the solution unless adequate core space is available. The use of this optional check is recommended for large problems.

Core Size Request (ICORE). The amount of core storage is requested on a system control card but may also be required as input on this ANSYS data card. The data check problem (or any solution problem) prints the core storage or data area required for the solution of the problem. This required core storage or data area should be requested for the run. The ANSYS coordinator at your computer facility or Appendix S should be consulted to determine the form of input in this field, if any.

FIG 10.8 Indatamall för ANSYS



sidor som beskriver de olika indata som gäller för de cirka 60 olika elementtyper som kan användas.

Även om den information som finns tillgänglig måste sägas vara mycket väl strukturerad, inser man att det inte kan vara lätt att försöka utnyttja ett sådant program för vanliga beräkningar - många av de frågor man måste besvara är helt irrelevanta för det aktuella problemet.

Naturligtvis skall inte heller ett program av denna kaliber användas för små problem. Att vi över huvud taget diskuterar det i detta sammanhang, är för att ställa frågan om stora, allmängiltiga eller små, begränsade program på sin spets. Båda varianterna har sina fördelar och nackdelar. Med ett litet, speciellt program för en speciell problemtyp får man ett lätt-hanterligt och snabbt program, där de indata som skall ges blir minimala. Å andra sidan måste man då ha tillgång till flera olika program för alla de aktuella strukturerna, med allt vad detta innebär i fråga om upplärning på nya program och så vidare. Dessutom begränsas ju möjligheterna att behandla strukturer som avviker litet grann från det vanliga. Om man exempelvis har ett program för beräkning av rammar och ett för väggskevior, så har man ingen möjlighet att klara av ett fall där man vill att en ram och en väggskevior skall samverka. Precis detta kan också sägas vara den största nackdelen med CHALMFEM-programmen, ett av de mest spridda FEM-paketerna i landet. I detta paket finns ett stort antal program för beräkning av olika typer av strukturer, där de olika programmen överensstämmer relativt väl i fråga om indatautformning och dylikt.

Låt oss nu i stället betrakta själva indataformuleringen. Som diskuterades i kapitel 2 skedde all inmatning av indata tidigare med hjälp av hålkort, där de aktuella sifferdata stansades i vissa bestämda kolumner. Numera, när terminalerna nästan helt tagit över som indatamedium har denna situation förändrats. Visserligen körs, som tidigare nämnts, fortfarande många stora program med hjälp av indatafiler i kortbilsformat, men utvecklingen går mot ett alltmer interaktivt arbetssätt. I stället för de mer batch-inriktade metoderna har dialogkörningen av program utvecklats starkt.

Dialog innebär i detta fall att datorn ställer frågor till användaren via hans terminal. Fördelen med detta är att användaren har en bättre kontroll över skeendet, genom att han kan låta datorns reaktion på en inmatning styra fortsättningen av körningen - märker man att något blivit fel i indata kan man avbryta och börja om utan att hela beräkningsgången behöver köras genom med felaktiga indata.

Man inser också lätt nackdelarna med dialogarbetet. Om man tänker sig att köra exempelvis ANSYS på detta sätt kommer man alltså att få ett stort antal irrelevanta frågor att besvara tillsammans med de som är relevanta



för det aktuella problem man vill lösa. Och detta upprepas om man vill göra om körningen på grund av att något blev fel. Totalt blir nog riskerna för fel lika stora med detta arbetssätt som med kortbuntarna.

Ett annat problem är de frågor som ställs av datorn. För att de skall vara till någon riktig hjälp för den som är osäker på programmets funktion måste de vara ganska utförliga och tydliga. Detta innebär, för den som kan programmet väl och utan någon nämnvärd ledning kan formulera sina data, att en stor mängd text kommer att skrivas ut på terminalen, utan någon egentlig funktion. Detta problem med alltför omfattande, snabbt framblinkande texter och frågor på bildskärmen har också identifierats som en klar stressfaktor för den som sitter mycket vid terminaler. För att lösa detta har man försett en del program med en möjlighet att välja kortare eller längre ledtexter i dialogen. Inte heller detta är emellertid någon perfekt lösning. Det kan ju nämligen vara så att en användare mycket väl känner till vissa delar av ett program, men är mindre bekant med andra. I så fall har han ju olika behov av hjälp i dialogen i olika delar av en körning.

I programmet V-FEM som utvecklats vid Bro- och Stålbbyggnad, KTH har en annan lösning valts. Programmet är kommandostyrt och man kan alltså som användare själv (i viss mån) bestämma i vilken ordning olika indata skall ges. Efter att man givit exempelvis kommandot "NODER" innebärande att information om en modells noder skall ges, hamnar man på en undernivå, där resten av indatagivningen sker i dialog-form. För den ovane användaren kommer denna dialog att bestå av ett antal frågor, som skall besvaras med ett, eller eventuellt två, tal. Den rutinerade användaren kan då i stället utnyttja en rad text som ges före den första frågan, vilken i mycket förkortad form, visar i vilken ordning frågorna kommer, jämför FIG 10.9. Med ledning av denna rad kan alla de värden som skall matas in ges på en enda rad, och i och med att det sista önskade värdet matats in betraktas dialogen som avslutad.

Ett bra dialogsystem för datainmatning bör också vara försett med kontroller av indata. Denna kontroll bör omfatta såväl en formell kontroll (siffror när siffror förväntas, text när text önskas), som en logisk (rimliga värden på olika storheter). Av dessa kontroller är den första relativt lätt att utföra i programmet, medan den andra är betydligt svårare. Det är ju nämligen mycket svårt att definiera ett intervall för rimliga tjocklekar för en platta, speciellt med tanke på att de enheter som används oftast är valfria. Andra faktorer kan dock något lättare kontrolleras. Idealiskt sett bör programmet hitta alla inmatningsfel av denna typ, och ge användaren en chans att ge ett nytt värde. Till detta idealtillstånd är dock vägen lång.

```

UNDERKOMMANDO:                               LINJE
(((NR1,X,Y,ANTAL,DNR,DX,DY,FHG)))

(((NODNR FÖR FÖRSTA NODPUNKTEN PÅ LINJEN:      1
  KOORDINATER X OCH Y:                          0,0,0
  ANTAL NODPUNKTER PÅ LINJEN:                    5
  DIFFERANSEN MELLAN NODNR:                      1
  DIFFERANSEN MELLAN X OCH Y KOORD:              1,0
  ANTAL FG I NODEN:                               )))2

UNDERKOMMANDO:                               LINJE
(((NR1,X,Y,ANTAL,DNR,DX,DY,FHG)))

(((NODNR FÖR FÖRSTA NODPUNKTEN PÅ LINJEN:      1,0,0,5,1,1,0,2

```

FIG 10.9 Valfri dialogomfattning i V-FEM

### 10.3.3 I beräkningsskedet

De flesta fel som uppstår i beräkningsskedet måste egentligen hänföras till de tidigare skedena, då de är följder av den formulering av beräkningsgången som gjorts eller de indata som användaren givit. Direkta fel i datorn som påverkar resultaten är mycket sällsynta. Därmed inte sagt att datorer inte går sönder - det gör de nämligen, men oftast går de sönder på ett sådant sätt att inga resultat alls kommer ut.

En typ av fel kan dock urskiljas som datorn eventuellt kan lastas för, nämligen de felaktigheter i resultaten som beror på att datorn gör sina beräkningar med begränsad precision, det vill säga noggrannhetsfel. Som beskrivits i kapitel 2 används för representationen av ett flyttal i datorn normalt 32 eller 48 binära siffror, vilket medför att talet har en relativ noggrannhet av cirka  $10^{-6}$  resp  $10^{-8}$ . Detta kan förefalla tillräckligt bra men det kan leda till problem vid till exempel lösning av stora ekvationssystem. Visserligen har tidigare konstaterats att styvhetsmatrisen för en Finit Element modell är välkonditionerad ur lösningssynpunkt, men vissa förhållanden kan få den att ändå bli ganska elak ur noggrannhetssynpunkt. En vanlig orsak till detta är att en modellerad struktur innehåller såväl mycket styva som mycket veka element. Betrakta till exempel ramstrukturen i FIG 5.6. Ur de härledningarna som gjorts i kapitel 5 inses att komponent (5,5) i strukturstyvhetsmatrisen kommer att be-

stå av summan av dels bidraget  $\frac{EA}{L}$  från balkelement AB  
 dels bidraget  $12 \frac{EI}{(2L)^3}$  från elementet BC. Med  $L=5$  m  
 och HE120A balkar fås att diagonalelementet blir lika  
 med

$$K_{55} = 8.9208 + 0.0088$$

vilket innebär att den horisontella balkens styvhet nästan försvinner i jämförelse med den vertikala. Om motsvarande förekommer på flera ställen i styvhetsmatrisen kommer resultatet av beräkningen att bli dåligt på grund av att avrundningsfelen blir stora.

Motmedlet mot denna typ av fel i datorer med 32 bits representation av flyttal är att arbeta i så kallad dubbel precision. Om detta utnyttjas kommer flyttalen att bli bestämda med en noggrannhet av ca  $10^{-13}$ , vilket medför att avrundningsfelen blir betydligt ofarligare. Denna möjlighet finns dock normalt inte i persondatorer, utan där får man nöja sig med en precision av cirka  $10^{-6}$  utan att kunna göra något åt det.

För att söka utröna hur stor betydelse denna typ av fel haft för en lösning av ett Finit Element problem är det bästa sättet att kontrollera nodjämvikten. Med den typ av fel som ovan skisserades kommer nämligen denna jämvikt normalt inte att vara, ens tillnärmelsevis, uppfylld. Detta är en följd av att man, när jämvikten beräknas, återgår till elementnivå, och därmed inte får in den nämnda effekten att en hel del information om styvheten försvinner, då två olika stora tal adderas.

När det gäller de fel som kan uppstå i en beräkning finns en hel del skrivet i olika läroböcker i Numerisk Analys, och någon mer djupgående analys skall därför inte göras här.

#### 10.3.4 I resultatkedet

När datorn skriver ut utdata från en körning, har man bara begränsade möjligheter att kontrollera deras riktighet. Som diskuterats tidigare i detta kapitel, så kan ju nu fel från de tidigare stegen ha slagit igenom och, i större eller mindre omfattning, förstört resultatet.

Även om emellertid inga fel uppstått så här långt, så kan de uppstå då de av datorn givna utdata skall utnyttjas för att ta fram de resultat man är intresserad av. Risken är ju att exakt de resultat man behöver inte finns i de papper som datorn lämnat från sig. Ofta måste man exempelvis interpolera fram momentvärden i

intressanta punkter ur några värden som datorn givit. Detta gäller kanske speciellt för Finita Element program som ofta skriver ut spänningar, moment och dylikt bara för de punkter som använts för den numeriska integrationen. Dessa skall alltså extrapoleras för att den totala spänningsfördelningen skall fås. Detta är inte heller så lätt att göra om man inte vet exakt hur den sanna spänningsfördelningen approximeras av elementmodellen. Till exempel är spänningarna ofta diskontinuerliga över elementgränserna, och man har därför flera olika spänningsvärden i nodpunkterna. En stor portion vana att tolka de beräknade resultaten krävs därför ofta.

De utdata man skall tolka kan även vara besvärliga på andra sätt. Som programmerare har man normalt två sätt att skriva ut siffervärden i resultaten, endera med något som kan kallas en fix decimalpunktsinställning, där man alltid får samma antal decimaler i svaren, eller i så kallad tio-potensnotation ( F resp E-format för den som är hemma i FORTRAN-jargongen). Den trevligaste formen är decimalformen eftersom man på detta sätt lätt jämför värdena i till exempel en tabell över spänningar eller förskjutningar. I ett generellt program är dock denna form nästan omöjlig att använda, då man inte vet för vilka strukturer som programmet kommer att användas - lasterna för två olika strukturer kan ju exempelvis lätt skilja sig med en faktor på en miljon. Det tråkiga med detta är att man exempelvis kan få utskrifter av typen 0.000056, om man betraktar en mycket liten struktur, eller av typen \*\*\*\*\*.\*\*\*\*\*, vilket betyder att lasten är så stor att den inte kan skrivas ut inom det avsedda antalet sifferpositioner.

Motåtgärden mot detta är att använda tio-potensutskrifter. På detta sätt kommer alla laster eller förskjutningar att skrivas ut med ( i stort sett ) samma relativa precision, och man förlorar alltså ingen information. Nackdelen med detta utskriftsätt är att det ur exempelvis en spänningstabell är svårt att hitta de maximala spänningskomponenternas läge, jämför FIG 10.10.

Det finns emellertid metoder att lösa båda dessa problem. Till exempel CHALMFEM-programmen gör en automatisk skalning av de värden som skall matas ut, på så sätt att en jämn faktor av 1000 skalas bort. Avsikten är att den största lastkomponenten alltid skall skrivas ut som ett värde i storleksordningen 10000. Även detta kan emellertid leda till problem, om man beräknar flera lastfall på samma struktur och man får olika skalningar för de olika lastfallen, vilket gör resultaten svåra att jämföra.

En annan faktor som kan ställa till problem i tolkningen av utdata är den teckenregel som används i utmatningen. När det gäller exempelvis förskjutningsvärden så är ju x- och y-förskjutningarnas positiva riktningar väldefinierade med hjälp av det koordinatsystem som



1	3.7992	0.0354	0.030(	1.1850E-04	-2.2640E-05)
2	3.3492	-0.0549	-0.033(	1.0501E-04	-2.2649E-05)
3	3.8002	0.0268	-0.039(	1.1859E-04	-2.2915E-05)
4	3.3507	-0.0640	-0.043(	1.0511E-04	-2.2943E-05)
1	0.1567	-2.6406	-0.197(	2.1402E-05	-8.3497E-05)
2	-0.0823	-2.5966	-0.111(	1.3657E-05	-8.0630E-05)
3	0.9339	-0.7911	-0.534(	3.4129E-05	-3.0558E-05)
4	0.5806	-0.6329	-0.477(	2.2100E-05	-2.3406E-05)

FIG 10.10 Exempel på olika resultatpresentationer

ansatts. Värre är det emellertid för exempelvis en platta där rotationsvärden fås ut. Det är inte självklart om rotation-x (eller vilken beteckning man använder i programmet) skall förstås som en rotation kring x-axeln, enligt skruvregeln, eller lutningen av transversalutböjningen i x-riktningen. Eftersom dessutom utböjningen för en platta oftast anges som positiv uppåt, till exempel i en FE-analys kan begreppen bli ganska oklara om ingen tydlig teckenförklaring finns angiven i programmets dokumentation, och i resultatutskriften.

Det största problemet i resultatskedet, ur tillförlitlighetssynpunkt är dock de värden som redovisas, och deras samband med de värden som normen använder. Som exempel på detta kan nämnas de krav på nedböjningsbegränsning som ges i normen. Vilket datorberäknat värde skall då jämföras med kravet, om man exempelvis gjort en analys som omfattar användning av såväl första som andra ordningens teori, och skall hänsyn tas till uppsprickning av betongen och dylikt.

Över huvud taget kan, som tidigare diskuterats i kapitel 7, normerna anses vara ganska dåligt anpassade till datorberäknade resultat, vilket leder till att ett stort arbete måste läggas ned på att omforma de av datorn producerade utdata till resultat som kan jämföras med normvärden o dyl.

#### 10.4 Åtgärder mot felkällorna

I de tidigare delarna av detta kapitel har ett antal felkällor i samband med datorberäkningar i konstruktionsarbetet diskuterats. Naturligtvis kan diskussionen synas onödigt pessimistisk. Vi har dock sett det angeläget att försöka påvisa de risker som finns i sam-



band med datoranvändningen, för att kanske stämma de mest entusiastiska datoriseringsivrarna till en viss eftertanke. Avsikten med detta är givetvis inte att helt avråda från all datorisering. Budskapet skall i stället tolkas som en uppmaning till försiktighet vid skapandet och användandet av alltför komplicerade program.

Vi skall nu i detta avsnitt diskutera de åtgärder som kan vidtas för att minska riskerna, och även de försök i denna riktning som redan gjorts.

De krav och önskemål som kan ställas upp i detta sammanhang kan delas in i tre grupper, nämligen krav på programmen, på dokumentationerna och, sist men inte minst, på användarna.

#### 10.4.1 I programmeringsskedet

En stor arbetsinsats har gjorts av forskare, främst inom datalogi och numerisk analys, för att utreda de felkällor som finns i datorprogram. Denna analys har dock nästan helt inriktats på något som kan betecknas som beräkningsnoggrannhet. Tyvärr har dock analysen nästan helt inskränkt sig till betraktelser av avrundningsfelens inverkan vid lösning av linjära ekvations-system o dyl. Detta är en typ av fel som man dock, vid normalt förekommande praktiska tillämpningar, oftast kan ta ganska lätt på, möjligtvis med undantag av vid användning av Finita Element metoder i små datorer utan möjlighet till räkning i dubbel precision. I stället är andra typer av fel, som visats tidigare, av större betydelse.

För beräkningen av främst Finita Element modeller har Nils-Erik Wiberg i boken "Finita Elementmetoden, en datoranpassad beräkningsmetod för ingenjörproblem" nämnt tre olika typer av fel, nämligen Modelleringsfel, Diskretiseringsfel och Manipulationsfel.

Modelleringsfelen är fel som beror på att en komplex fysisk verklighet transformeras ned till enkla hanterbara strukturer, ofta i form av plana modeller. Dessutom innefattas i dessa fel som beror på att laster, randvillkor, geometri och materialegenskaper beskrivs på ett förenklat och osäkert sätt.

Diskretiseringsfelen omfattar bland annat välet av matematisk formulering, och de approximationer som görs vid beskrivningen av förskjutningarna i strukturen o dyl. Dessutom tas i detta sammanhang upp fel som beror av dålig, eller onoggrann, elementindelning, och representation av krökta linjer med raka.

Manipulationsfelen berör datorns arbete och omfattar noggrannhetsbrister i indata, förenklingar vid t ex numerisk integration och avrundningsfel i beräkningarna.

I dessa resonemang koncentreras hela feldiskussionen på beräkningsskedet, och i viss mån på indataformuleringen. Med detta antar man att alla inblandade parter alltid gör rätt. Ingenting nämns om de möjligheter till fel som finns till följd av felaktig eller olämplig programformulering, felaktig användning av programmen, eller felaktig tolkning av resultaten.

Detta förefaller dock vara litet väl optimistiskt i det skede av datorisering som pågår. Ett stort antal konstruktörer ställs för första gången inför möjligheten att låta datorn göra en betydande del av arbetet. Med den bakgrund som givits tidigare i detta kapitel blir slutsatsen att det finns många risker i datoranvändningen som beror på en dålig information mellan de olika parterna i FIG 10.6 - ett förhållande som gör programmen svåra att använda och resultaten svåra att tolka.

För att övervinna riskerna har man från flera håll föreslagit ett typgodkännandeförfarande för datorprogram, och Statens Planverk har sedan långt tillbaka ett förslag till riktlinjer för detta. Avsikten med dessa strävanden är naturligtvis lovvärd, men det kan ifrågasättas om ett sådant förfarande är lämpligt. Fle- ra skäl mot detta kan anföras. Bland annat torde det vara fullständigt omöjligt att garantera att ett visst program, åtminstone om det är av någon omfattning, alltid räknar rätt. Visserligen skulle ett typgodkännande av ett program förmodligen inte ta ifrån konstruktören ansvaret för den gjorda beräkningen, men nog måste det vara en uppenbar risk att en beräkning, försedd med beteckningen "Typgodkänd" inte kommer att ifrågasättas i någon nämnvärd omfattning.

Ett annat problem med detta förfarande är att programmet på något sätt måste bevaras i sin godkända form för "all framtid". Med kännedom om alla programmerares iver att "förbättra" sina program synes detta krav vara svårt såväl att uppfylla, som att kontrollera. I detta sammanhang måste även problemen lösas med hur skillnaderna i olika datorers arbetssätt skall överbryggas.

Intressant att studera, är också vad som krävs för att få ett program typgodkänt. I princip skall en användarinstruktion och en verifikation lämnas in för granskning, eventuellt kompletterad med en flödesplan. Verifikationen kan bestå av att ett antal fall kontrollberäknas, endera för hand eller med hjälp av något annat program. En annan möjlighet för verifikation kan vara provbelastningar eller modellförsök. Planverket kan även tänka sig att en listning av programmet lämnas in för granskning, vilket anses vara det mest önskvärda alternativet.

Naturligtvis kan en kontroll av ett program vara möjlig, om programmet är avsett för ett alldeles speciellt begränsat fall, exempelvis dimensionering av en

takstol med ett visst bestämt utseende, som kan specificeras med ett fåtal mått. I övrigt måste kontroll av ett programs riktighet betraktas som fullständigt omöjlig. Uppenbarligen kan man dock väldigt lätt hamna i gränsdragningsproblem mellan program som är så små att man har någon möjlighet att kontrollera dem och litet större program som man ej kan kolla.

En slutsats måste vara att typgodkännande av program ej bör begagnas. Ett undantag skulle då kunna vara för program som är kopplad till en speciell produkt, till exempel en standardiserad takstol, eller för beräkning av erforderlig tjocklek för ett plåttak, med dimensioner och snölast givna.

En mera framkomlig väg borde vara att införa någon typ av "Varudeklaration" för program, som beskriver ett programs funktion, de antaganden det bygger på och programmets begränsningar. Över huvud taget är det främst information som saknas mellan de olika parterna, vilket mera ingående kommer att behandlas i nästa avsnitt.

Vad gäller programmen som sådana, och då speciellt de större programmen behöver åtskilligt göras i fråga om kontroll av beräknade resultat. Detta gäller såväl för de indata som programmet matas med som de mellanresultat som kommer fram i beräkningen.

Man bör även sträva efter att göra indata och utdata så tydliga som möjligt, så att möjligheterna till fel-tolkningar minskas. Ett led i detta bör vara en satsning på dels dialog-körning av programmen, dels grafisk presentation av resultat.

En viss energi bör också offras på att välja vilken information som skall presenteras i de utdata som datorn levererar. Kraven på dessa är dock inbördes motstridiga, i det att dels en så noggrann beskrivning som möjligt önskas av den utförda beräkningen dels att onödiga, pappersfyllande resultat undvikas. Överhuvud taget behöver dessa utdata struktureras. En ide för detta har tillämpats för programmet SKORSTEN, tidigare refererat i avsnitt 7.2.2. Redovisningen av denna konstruktionsberäkning av stålskorstenar redigeras på samma sätt som en handberäkning, jämför utdraget ur resultatlistan i FIG 10.11.

Ett speciellt problem med resultatutskrifter bör också behandlas i samband med utarbetandet av rekommendationer för utdataredovisningar, nämligen identifikationen av en beräkning. Om man nämligen gör ett antal beräkningar av samma struktur, men med något olika förutsättningar, kan det efteråt bli svårt att skilja de olika listorna från varandra. Man kan alltså ganska lätt, av misstag förhoppningsvis, ta redovisningen av indata från en körning, och resultatet från en annan, och redovisa dem som en enhet. Lösningen på detta torde vara att datorn automatiskt genererar någon identi-

## 7 SKORSTENSFOT

## 7:1 GRUNDSKRUVAR

STRACKGRANS FÖR GRUNDSKRUVAR (MPA): FYSK = 300.  
 ANDEL AV YTTRE DRAGKRAFTSVARIATION  
 SOM ANTAS PÅVERKA SKRUVAR (PROC): PROCSK = 100.  
 FAKTOR FÖR TILLÄTEN SPANNINGSVIDD  
 ENLIGT STBK-N3 TABELL 3:431, NR 16: UTMFSK = 0.700  
 TILL DRAGSPÄNNING, VANL. LASTFALL (MPA): SVTILL = 200.0  
 TILL DRAGSPÄNNING, EXC LASTFALL (MPA): SETILL = 230.8  
 TILL DRAGSPÄNNINGSVIDD I SKRUV (MPA): SRTILL = 50.0

SKRUV- DIAM (M)	ANTAL GRUND- SKRUVAR	SKRUV- CIRKEL- DIAM (M)	DRAGSPÄNNING (MPA) VANL EXCEPT	DRAG- SPÄNNINGS- VIDD (MPA)	MAX UTNYTT JANDE	
0.0160	120	1.888	8.3	48.5	48.5	0.969
0.0200	76	1.908	8.3	48.6	48.6	0.970
0.0240	52	1.928	8.3	48.7	48.7	0.974
0.0300	32	1.958	8.3	49.1	49.1	0.981
0.0360	24	1.988	7.5	44.2	44.2	0.884
0.0420	16	2.018	8.1	47.7	47.7	0.952
0.0480	12	2.048	8.1	47.6	47.6	0.952

ANMÄRKNING: TABELLEN ANGER ERFORDERLIGT ANTAL GRUNDSKRUVAR  
 MED AVSEENDE PÅ SKRUVARNAS HALLFASTHET. ERFORDERLIGT ANTAL  
 MED HÄNSYN ÄVEN TILL MAXIMIAVSTÅND MELLAN GRUNDSKRUVAR  
 VID OLIKA TJOCKLEK HOS FOTFLANSAR ANGES NEDAN.

FIG 10.11 Utdrag ur resultat från programmet SKORSTEN

fikation för varje körning som redovisas på varje sida i resultatutskriften. Förslagsvis kan en maskingenere-rad tidsuppgift fungera som detta kännetecken.

#### 10.4.2 Vid programbeskrivandet

Som framgick av de tidigare avsnitten i detta kapitel, berodde de flesta felmöjligheterna vid datorberäkningar på en bristfällig kommunikation mellan de ingående parterna. En stor insats måste därför göras på detta område för att minska dessa risker.

Som visades i kapitel 3 av denna bok har ett visst intresse ägnats åt denna problematik av vissa forskare. Någon riktigt genomarbetad ide för hur man skall gå tillväga har dock inte presenterats. Det borde dock hos de granskande myndigheterna med Statens Planverk i spetsen finnas ett stort intresse för hur program och beräkningsmodeller skall beskrivas på ett sätt som underlättar granskning och minskar tillförlitlighetsriskerna. Som visats i rapporten R35:1982 så upplever ju framförallt byggnadsinspektörerna den nuvarande situationen som helt otillfredsställande.



Vi skall i detta avsnitt kortfattat redogöra för de krav, man som programanvändare och som granskare bör kunna ställa på tillgängliga program. Vad som för det första är alldeles klart är att dokumentationen av ett program måste utformas på olika sätt för de olika aktuella målgrupperna. Medan användaren till exempel behöver en klar information om hur han skall hantera programmet och dess indata, behöver granskaren egentligen bara uppgifter om de förutsättningar och begränsningar som programmet innehåller, den inmatade modellen och hur utdata från programmet skall tolkas. Den mellanliggande metodiken bör granskaren, om programmet bara är ordentligt testkört och verifierat, kunna acceptera utan granskning.

En första indelningsgrund för dokumentationen är därför den målgrupp till vilken informationen skall riktas. En andra är då de delar av beräkningsprocessen som beskrivs. I detta sammanhang är framförallt tre delar intressanta, nämligen

- Beräkningsmodellen
- Hanteringen av programmet
- Tolkningen av resultaten (utdata)

När det gäller beräkningsmodellen så är det ju, som tidigare nämnts, inte bara den rent geometriska beskrivningen av strukturen som är intressant, utan vi vill även i "modell"-begreppet innefatta sådant som representationen av materialbeteende, antaganden om statistiskt verknings sätt, införandet av randvillkor och möjligheterna att beskriva olika lastfall. Vid FEM-beräkningar tillkommer sedan förutom detta den använda elementtypens egenskaper och approximationer.

Hanteringsaspekten är en fråga om hur programmet kan utnyttjas, och hur indata skall ges. För programanvändaren är den senare delen starkt kopplad till de modelleringsmöjligheter som programmet innehåller.

Tolkningsanvisningar är nödvändiga för att användaren skall kunna ta ut de resultat han behöver ur de utdata som programmet levererar. De är dock kanske än viktigare för den som skall granska de resultat som producerats, eftersom denne ofta inte har samma vana vid ett visst program som den som gjort beräkningarna.

En fullständig dokumentation av ett program skulle alltså, idealiskt sett, bestå av ett antal byggbitar som kombineras ihop på olika sätt för de olika aktuella målgrupperna. Fyra frågor kan användas för att klargöra vad som bör ingå i var och en av de olika nivåerna, nämligen

- För vem?
- Hur mycket?
- Vad?
- Hur?



Man bör också ha klart för sig att de olika intressenterna har olika intressen vid olika tidpunkter. Konstruktören som skall lösa ett speciellt problem behöver för det första komma i kontakt med ett program som kan hantera den aktuella problemtypen. För att hitta detta behöver han då en programinventering över tillgängliga program, med en mycket kortfattad beskrivning av programmens möjligheter. Denna programinformation kan alltså omfatta, kanske endast ett tiotal rader text, vilket bör vara tillräckligt för att grovsortera fram ett antal tänkbara program. Bland dessa behöver han sedan kontrollera, något mera i detalj, vilka typer och storlekar av problem som kan hanteras - en beskrivning som kanske kräver ett par A4-sidor text.

Efter att ha valt ut ett program som skall användas behövs information om hur programmet och beräkningsmodellen skall hanteras. Denna information måste innehålla såväl en teoretisk bakgrund för den beräkningsmodell som används, dels en indatabeskrivning för programmet. Det allra bästa vore dock om dessa skildes åt, och redovisades i var sin skrift, för att inte onödigtvis dölja väsentlig information. I många fall är det ju, då programmet bygger på en välkänd beräkningsmetod, inte nödvändigt att studera de bakomliggande teorierna. I stället har man nytta av en mall för hur indata skall ges, för att översätta den tänkta verklighet man arbetar med till den beräkningsmodell som datorn kan tolka. För den som är van att använda det aktuella programmet kan även ett mycket kort sammandrag av indata, kanske skrivet på en enda sida, vara nyttig, åtminstone om programmet inte bygger på dialogstyrning.

I det sista skedet behöver konstruktören sedan viss information om hur resultaten skall tolkas, den förväntade noggrannheten i den gjorda lösningen och förklaringar till de eventuella diagnosmeddelanden som dykt upp under körningen, och deras eventuella inverkan på resultaten.

Vi har i detta avsnitt skisserat ett antal synpunkter på dokumentation av datorprogram och datorberäkningsmodeller. Texten har påvisat ett antal behov av information, och behov av att informationen struktureras på ett, för de olika intressenterna, riktigt sätt. Många av de skisserade målsättningarna kan synas överdrivna och i praktiken ouppnåeliga. I den gjorda beskrivningen finns dock en avsiktlig provokation gentemot dels programskaparna, men kanske framförallt mot de granskande myndigheterna. Det är ju nämligen, sedan länge, bekant bland dem som sysslar med beräkningsprogram att glappen i information kring programmen är stora. Vad som dock förmodligen krävs för att något skall hända är att endera Myndigheterna kommer med krav, eller att till exempel Byggstandardiseringen arbetar fram riktlinjer för hur dokumentation och redovisning av datorberäkningar skall utformas.

### 10.4.3 Vid programanvändandet

Vi har nu framfört ganska kraftiga krav på dem som skriver beräkningsprogram. För konsekvensens skull skall vi nu även diskutera användarens roll i den informationskedja som datorberäkningarna innebär, och de risker som kan föreligga vid användning av datorer för byggnadskonstruktionsarbetet.

När det gäller användaren som felkälla, finns nästan endast ett motmedel, nämligen utbildning. Undervisningen i datormetoder vid våra Tekniska Högskolor och våra gymnasier måste expandera i, minst, samma takt som datoriseringen av byggbranschen. I och med utbildningen kommer en sundare ( mera kritisk?? ) inställning till datorer och datorberäkningar att få genomslag.

Några råd i ett kortare perspektiv kan vi dock ge dem som är inblandade i datorberäkningar. Med ledning av de tidigare resonemangen är naturligtvis det första att iaktta stor försiktighet vid användande av obekanta datorer eller program, och noga kontrollera de resultat man får fram. Det är därför speciellt viktigt, i dessa fall, att göra omfattande kontroller av de beräknade resultaten, för att på detta sätt övertyga sig själv om att man förstår de anvisningar som manualer och bakgrundsmaterial innehåller.

Vi vill vidare påstå att en grundprincip vid all datorbearbetning måste vara att inte låta en dator utföra någon beräkning som man inte, åtminstone i princip och med gott om tid på sig, kan utföra själv. I annat fall kommer de beräknade resultaten att vara praktiskt taget omöjliga att bedöma riktigheten i, och det blir alltför lätt att utan vidare acceptera de resultat som datorn beräknat. Detta ger också upphov till det andra rådet, nämligen att använda en omvänd bevisföring för de beräknade resultaten. Antag därför alltid att de resultat som kommit ut är felaktiga, tills dess Du är helt och fullt övertygad om motsatsen.

För att detta skall vara möjligt räcker det normalt inte att kontrollera slutresultaten. Man måste, åtminstone då det gäller stora program, även kontrollera vissa väsentliga mellanresultat. Detta görs lämpligen med enkla överslagsberäkningar, där man förenklar en komplicerad struktur till något som man lätt kan analysera. Givetvis kommer inte den grova analysen att visa sanningen, men med en god vana bör man med relativt god tillförlitlighet bedöma om datorns resultat är rimliga. I utbildningssituationen är denna aspekt viktig då den innebär att man inte, bara för att datorerna tar över en allt större del av beräkningsarbetet, får minska på utlärningen av vanliga analysmetoder. Snarare blir förståelsen för en strukturs verkningssätt än mer väsentlig i denna arbetssituation.

Att, av princip, misstro alla resultat som datorn ger ifrån sig är en god tumregel. Det är annars alltför

lätt att lura sig av den höga noggrannhet som de många decimalerna i utmatningen ger sken av. En jämförande beräkning med någon förenklad metod kan i detta sammanhang vara en god ide.

Ur de angivna råden kan man kanske dra en gemensam röd tråd, nämligen att konstruktören skall hålla kontrollen över beräkningen själv. Det får inte bli så att ett visst datorprogramms möjligheter blir bestämmande för hur en konstruktion skall utformas. Datorn måste därför hela tiden ses som ett hjälpmedel som skall ge konstruktören underlag för beslut, inte som den som fattar besluten. Detta kommer emellertid att ytterligare diskuteras i kapitel 11, som behandlar utvecklingen att fler och fler beslut, kontroller och val förs över på datorn, i så kallade totalintegrerade system.

### 10.5 Målsättningar för fortsatt forskning

Med anledning av det som beskrivits i de tidigare avsnitten i detta kapitel kan ett par angelägna forskningsområden skisseras, som har det gemensamma målet att öka tillförlitligheten i datorberäkningar inom byggnadskonstruktionsverksamheten.

För det första bör en satsning göras på att ta fram rekommendationer för hur beräkningsmodeller och datorprogram skall beskrivas. Viktigt är, i detta sammanhang att rätt information, varken mer eller mindre, når de olika parterna i kedjan i FIG 10.6. Det är angeläget att såväl mängden som innehållet i dessa beskrivningar kan anpassas för de olika behov som finns, en diskussion som finns påbörjad i avsnitt 10.4.2, men som kräver betydligt mera arbete.

I samband med detta kan en diskussion tas upp om de gällande normernas anpassning till och lämplighet för datorberäkningar. Till mycket stor del kommer detta arbete att bestå i att finna en nomenklatur för hur olika antaganden och förutsättningar skall anges i redovisningen av gjorda beräkningar, för att jämförelsen med normvärden o dyl skall bli mera entydig.

Som ett andra, mycket väsentligt, forskningsområde måste utformningen av resultat från datorbearbetningar framhållas. Även för detta område borde krav och rekommendationer formuleras, som förhoppningsvis kan minska riskerna för missförstånd vid tolkning av datorberäknade resultat.



## 11 TOTAL-SYSTEM, DISKUSSION OM EN VISION

Detta kapitel skall ge en relativt utförlig genomgång av och en diskussion om vad vi kallar totalsystem, eller totalintegrerade system. Detta begrepp avser datorprogram som i en och samma exekvering, utan manUEllt ingripande, utför flera olika moment i konstruktionsarbetet. I sådana program ingår både analys och dimensionering. Vanligtvis ingår även någon form av bestämning eller optimering. Detta kan vara en beräkning av "optimal" sektionshöjd, val av armeringsmängd eller val av sektionformning. Ett totalsystem kan vara begränsat till ett speciellt material och en speciell sektionformning, till exempel I-balkar av stål, eller vara mera heltäckande och jämföra olika material såväl som olika utföranden.

Eftersom det har diskuterats en hel del om totalsystem, både i fackkretsar och mellan lekmän, så är det väsentligt att vi här ger genomgången av totalsystem ett relativt stort utrymme. Dessutom är kapitlet viktigt eftersom företagets val och inriktning, till exempel på något totalsystem, spelar en betydande roll för konstruktörernas framtida arbetssituation.

Visserligen har totalsystem redan tidigare behandlats, avsnitt 1.3 och kapitel 7, men dessa avsnitt var endast avsedda som introduktion respektive redovisning av befintliga system. Det här kapitlet skall försöka analysera och behandla ett antal olika aspekter såsom svårigheterna med att uppfylla alla de visioner som finns samt beröra den utformning och omfattning av totalsystem som kan anses lämplig.

Det bör, för att undvika missförstånd, påpekas att kapitlet har en subjektiv prägel och redovisar författarnas åsikter. Som kommer att framgå nedan så är vi ganska kritiska mot själva grundtanken på totalsystem. Därmed förkastar vi definitivt inte alla typer av system. Många av dagens system fyller ett reellt behov. Det gäller dock att inse faran och svårigheterna med alltför stora system.

Att vara kritisk mot datorer och datorsystem har blivit en allmänt utbredd företeelse. För att inte bara falla in i denna allmänna klagolåt skall i kapitlet även skisseras alternativa ideer till totalsystem, se avsnitt 11.3.

### 11.1 Visioner

Det finns många visioner om datorsystem. Föreställningar om vad dessa förväntas klara av, idag eller inom en snar framtid, är ofta högtflygande och orealistiska. För att belysa detta skall här återges, och



kommenteras, en del av de visioner som nämnts.

Låt oss börja med något som liknar "science fiction" för att sedan stegvis arbeta oss ner till de mera realistiska och genomförbara ideerna.

Det är allmänt accepterat att funktionskrav vore att föredra framför specificerade krav. Exempel på funktionskrav kan vara att kräva att en vägg skall ge ett tillräckligt skydd mot kyla. Ett specificerande krav är då att kräva att väggens k-värde skall vara högst  $0.25 \text{ W}/(\text{m}^2 \text{ C})$ . Svårigheten med funktionskrav är dock att identifiera och formulera dessa. Om man dock förutsätter att kraven kan formuleras så skulle de olika kraven på t ex ett byggnadsverk kunna matas in i en dator som sedan utgående ifrån dessa skulle bestämma objektets utformning.

Denna vision är vad vi närmast skulle kalla science fiction. Hur skulle något sådant kunna programmeras? Trots allt är det ju människor som måste skriva datorprogram för detta. Dessutom - hur skall funktionskrav kunna formaliseras i datortermer?

Nästa vision som vi skall beröra är en totaloptimering av något, till exempel en byggnad. Man specificerar vissa förutsättningar, som till exempel undergrundens kvalitet, antal våningar och yttermått. Detta matas in i datorn och ut kommer sedan specifikaationer och färdiga ritningar, se FIG 11.1

Indatamängden måste av nödvändighet bli mycket stor. Dessutom måste det nog till "superdatorer" för att kunna hantera program med en rimlig flexibilitet. Programmen måste ju jämföra "allt" och kontrollera "allt" och därmed bli näst intill oändligt stora.

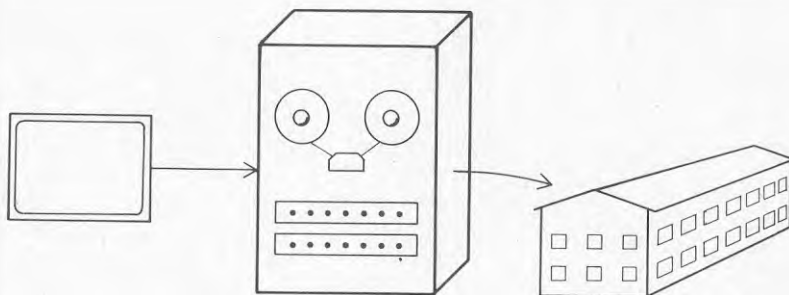


FIG 11.1 Totaloptimering av byggnad inom vissa ramar

Att denna vision över huvud taget berörs är framkallat av att den inte alls är ovanlig bland lekmän, även om den oftast inte är alltför genomtänkt. För att hålla diskussionen och tankarna på rätt plan måste sådana visioner om datorer motverkas. Speciellt som de ibland används för att motivera tankar om att flera yrkeskategorier skulle bli arbetslösa och därmed används för "skrämselpropaganda".

Förutom att den är en fantasivision, som för tillfället är praktisk ogenomförbar, så tillkommer det faktum att de flesta besluts- och kontrollpunkterna skulle finnas i programvaran. Att kunna infoga all den mångfald av aspekter rörande stora och viktiga beslut, som till exempel ingår i ett val av material för den bärande stommen, i ett datorprogram är omöjligt. Många aspekter som inverkar är dessutom svåra att kvantifiera, till exempel personalens vana att handskas med olika material, och krav på flexibilitet i planlösningen.

Att de ovan beskrivna visionerna är orealistiska borde nu klart framgå.

De visioner som i fortsättningen skall behandlas är dock möjliga att praktiskt förverkliga.

Även om man inte kan "totaloptimera" en hel byggnad så kan man skapa programvara för en deloptimering, till exempel av primärt bärande system. Antalet alternativ kan vara begränsat, till exempel bestående av endast stålpelare och stålbalkar. Vad systemet skulle åstadkomma är hållfasthetsberäkningar, ekonomiska bedömningar och att framställa arbetsritningar.

Om man ytterligare trappar ned ambitionsnivån hamnar man på system som endast behandlar ett material och en given utformning. Ett exempel på detta kan vara trätakstolar till bostadshus. Systemet kontrollerar hållfastheten och framställer planritningar och detalj-skisser samt mängdspecifikationer.

De fyra ovan skisserade visionerna, som givetvis saknar specifika gränser, beskriver i all sin enkelhet de olika tankar som förespråkarna för totalsystem har för framtiden. I de följande avsnitten skall det diskuteras vad en satsning på sådana system innefattar och för med sig.

## 11.2 Diskussion om totalsystem

### 11.2.1 Behov och önskemål

De olika datorvisioner, som beskrivits ovan, har inte uppkommit av en slump. Tvärtom så är dessa framkallade som en följd av de områden som man tror kommer att datoriseras, och hur denna datorisering kommer att genomföras. Science fiction visionerna lämnar vi åt

sitt öde och inriktar oss istället på de mera realistiska visionerna.

Konstruktionsberäkningar är ett område som tidigt utnyttjade datorbaserade hjälpmedel. Först gällde det endast för rena matematiska beräkningar, till exempel matrisberäkningar, sedan i allt vidare bemärkelse, till exempel FE-beräkningar.

Som diskuterats i kapitel 6 och 7 är det mesta av den befintliga programvaran antingen analysprogram eller dimensioneringsprogram. Det finns endast några enstaka program för både analys och dimensionering, jämför avsnitt 11.2.2. Dessa program är dessutom oftast av speciell karaktär.

Det verkar logiskt att utvidgningsprocessen fortsätter. Nästa steg av datoriseringen av konstruktionsarbetet skulle därmed innefatta hela konstruktionsberäkningen, d v s en koppling av beräkningar för analys och beräkningar för dimensionering.

En rimlig fråga som en konstruktör kan ställa sig är varför man måste utföra ett manuellt arbete med att överföra utdata från ett analysprogram till indata i ett dimensioneringsprogram. Detta kan tyckas än mer förvånande då utdata från analysprogrammet vanligen inte alls är anpassat till den indatiform som dimensioneringsprogrammet önskar.

Som exempel på detta kan nämnas att utdata från ett datorprogram för analys av plattor vanligtvis är böjande och vridande moment,  $m_{xx}$  respektive  $m_{xy}$ . För dimensionering av betongplattor önskar man dimensionerande moment enligt,  $m = m_{xx} + k * m_{xy}$ , se BBK79 eller B7.

Att manuellt behöva göra denna beräkning kan uppfattas som onödigt. Speciellt som beaktandet av det vridande momentet endast behövs för en elasticitetsteoretisk beräkning och inte för de traditionellt vanligare metoderna för beräkning av betongplattor, som till exempel brottlinjeteori.

Ännu mer förvånande, och besvärande, är det om utdata från datorprogrammet inte ger moment och tvärkrafter utan istället ger böjspänningar och skjuvspänningar. Detta eftersom spänningarna då är beräknade enligt förutsättningen om homogent tvärsnitt, och därmed ogiltiga för betong. Denna utdataform är inte ovanlig från FE-program, då ju dessa oftast inte är skrivna av byggare. För att erhålla dimensionerande moment och tvärkraft måste man manuellt räkna fram dessa från spänningarna och tvärsnittsdata.

Det finns alltså ett stort behov av en närmare koppling mellan analysprogram och dimensioneringsprogram. Ur detta uppkommer då naturligt en vision om totalsystem. I och med kopplingen kan man ha ett totalsystem

för hela beräkningsdelen av konstruktionsprocessen.

Ytterligare utvidgningar innefattar delar som ligger närmast före respektive närmast efter själva konstruktionsberäkningen. Nästa system innefattar sålunda även val av konstruktionssystem och/eller ritnings- och specifikationsframställning, jämför visionen om del-optimering i avsnitt 11.1.

Det finns andra önskemål och krav när det gäller datorer i konstruktionsarbetet. Dessa har dock ingen direkt koppling till de tidigare skisserade visionerna.

Områden som allmänt anses lämpliga för datorisering är arbeten som är av rutin- eller standardkaraktär.

Arbete som kan hänföras till rutinarbete är, för att återigen ta betongplattor som exempel, beräkningar för armeringsspecifikationer. Armeringsmängderna skall beräknas och korrektioner för bockning skall utföras. Detta innebär, eftersom det är lätt att missa något, att arbetet säkrare och snabbare utförs av datorn. För att få effektivitet och ekonomi krävs dock att datorn även utför armeringsritningarna.

Den moderna byggnadstekniken använder sig ofta av standardiserade prefabricerade element. Vid en dimensionering av prefabelementen har man ofta använt sig av tabeller eller diagram som tillhandahållits av leverantörerna.

För att erhålla en större flexibilitet kan dimensioneringen istället ske med ett datorprogram. Den datamängd som kan finnas i ett datorprogram överstiger ju vida vad som är möjligt med överskådliga tabellverk. Till exempel kan flexibiliteten för en kombination av olika lastfall och belastningar ökas om man utnyttjar ett datorprogram vid dimensioneringen. Sådana program kan ju dessutom utgöra underlag för en framtagning av dimensioneringstabeller.

Vad man däremot inte vill datorisera är kontrollen över konstruktionsarbetet.

Vad som skall göras, och när, är beslut som bör ligga på den ansvarige konstruktören. Dessutom vill denne avgöra, eller åtminstone ha kontroll över, de flesta beslut som tas.

Om programflexibiliteten är för liten upplevs lätt användandet som kreativitetshämmande. Ett önskemål i samband med detta är att det bör vara användaren som styr programmet och inte att användarens val och beslut styrs av programmet. Man vill ju inte vara låst av ett begränsat antal standardlösningar.

Att inte minska kontrollmöjligheterna när man datoriserar konstruktionsarbetet är ett givet krav både från kontrollerande myndighet som från den interna gransk-



ningen. Mellanresultat och jämviktskontroller bör redovisas av datorprogrammen. Till exempel bör resultaten från analysdelen redovisas liksom resultaten från dimensioneringsdelen.

I många sammanhang, speciellt för konstruktionsfirmor med stor andel utlandsprojekt, är det önskvärt att dimensioneringsdelen av ett datorprogram inte är alltför starkt kopplad till något speciellt normsystem. Normkrav bör därför fungera som undersystem till huvudsystemet. Allt för att underlätta användandet av olika normer men även för att underlätta programändringar då normerna revideras.

### 11.2.2 Existerande system

Vi ska i detta avsnitt redogöra för kopplingen mellan dagens existerande totalsystem och de visioner, behov och önskemål som redovisades i de föregående avsnitten. Vad som främst skiljer detta avsnitt från kapitel 7 är att här skall totalsystemen ses från ett vidare perspektiv medan kapitel 7 mer faktamässigt redovisade totalsystem som är i drift.

Trots att vi inte har utfört någon heltäckande inventering av olika datorprogram så framgick det ändå klart, se kapitel 7, att det endast finns ett begränsat antal totalsystem på marknaden. Det som är karakteristiskt med dessa system är att de endast behandlar ett material och att de bara är avsedda för en speciell konstruktionstyp.

Ett exempel på detta är dimensionering av trätakstolar, där det dessutom finns flera olika programsystem att välja emellan. Det finns flera olika skäl att just beräkning av trätakstolar har datoriserats. Av dessa skall vi här nämna tre tunga faktorer.

För det första så har det länge funnits datorprogram för analysdelen i form av olika ramverksprogram. Trätakstolar är vanligtvis flerfaldigt statistiskt obestämda och därmed ganska komplicerade att handberäkna.

Konstruktionsformen, till exempel svensk takstol, är till stor del standardiserad. Dessutom är belastningarna av standardtyp, till exempel som utvändigt last i form av snölast och vindlast. Till stor del "standard" är därmed den andra faktorn.

Den tredje faktorn är ekonomiskt betingad. För att få ekonomi i beräkningarna måste resultaten av analysdelen anpassas och kopplas ihop med dimensioneringsdelen. Annars blir det billigare att utnyttja dimensioneringstabeller.

Att det endast är för vissa speciella fall som det finns totalsystem är inget unikt för Sverige. FIG 11.2



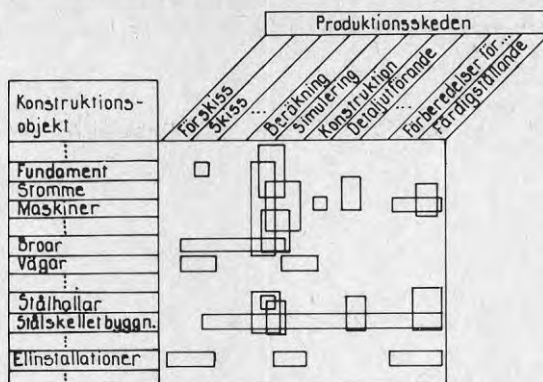


FIG 11.2 Befintlig programvara i Tyskland, efter Lang & Lendorff i Bergamo-rapporten, se avsnitt 12.3 (fig hämtad ur R35:1982)

redovisar befintlig mjukvara i Tyskland. Som kan utläsas i figuren så finns totalsystem, horisontella utsträckta markeringar, för till exempel stålskelettbyggnader. Dessutom visas, som framgått av tidigare kapitel av denna bok, att beräkningsdelen är den del där det finns störst antal datorprogram.

Man ser i FIG 11.2 att för broar, som ju är mer beräkningsmässigt standardiserade än bostadshus, det finns program som täcker in flera delar av dimensioneringsprocessen.

Ett annat exempel där det finns programsystem är offshorekonstruktioner. Till exempel så har Lloyds Register of Shipping i England flera beräkningssystem för olika typer av offshorekonstruktioner. Dessa system täcker grafisk geometri-inmatning, global och lokal hållfasthetsanalys samt kontroll av uppfyllandet av normkrav. Systemen har utvecklats genom en utvidgning av beräkningsdelen, jämför avsnitt 11.2.1. Att man till så stor del använder datorhjälpmedel inom offshore är beroende på att hållfastheten är väsentligt mera avgörande för projektets ekonomi än vad som är fallet för till exempel bostadshus.

Även om det finns ytterligare områden än de ovan nämnda där det finns totalsystem, så existerar det idag inte några system som uppfyller visionen "deloptimering". Vad som tycks vara svårast att klara av är själva optimeringsdelarna. Detta är dock inte så förvånande då dessa delar ofta är svårkvantifierbara. Givetvis så finns det heller inte heller några system som uppfyller de högre visionerna.

### 11.2.3 Svårigheter med totalsystem

Att det inte finns några totalsystem som uppfyller de högre visionerna är delvis beroende på att sådana system har haft en tendens att aldrig bli klara. Det finns många storslagna projekt som skrinlagts. För att citera en etablerad engelsk konsultfirma: "Totalsystemet (som aldrig blev helt klart) har vi kastat i papperskorgen"

Varför är det så? Vilka är svårigheterna med totalsystem? Dessa frågor skall vi försöka belysa i detta avsnitt.

En av de främsta anledningarna till att totalsystem sällan blir klara är att programmen ofta blir för stora och svårbemästrade. Detta innebär att de är svåra att få att fungera och att eventuella programmeringsfel blir svåra att lokalisera.

För att programmen ska bli överskådliga krävs att man redan på ett tidigt stadium har klart för sig vad som ska medtas i programmet och hur beräkningsgången ska struktureras. Detta kräver att en mycket noggrann och genomarbetad projektplan ställs upp innan själva programmeringsarbetet startar.

Ofta har det emellertid varit så att arbetet baserats på ett befintligt program, vanligtvis ett analysprogram. Detta program har sedan med tiden successivt utökats i flera omgångar. Eftersom basprogrammet, analysprogrammet, har skapats utan hänsyn tagen till andra programdelar är det stor risk att detta gör programmeringen för de övriga programdelarna något konstlad och svåröverskådlig. Att ändra och strukturera upp basprogrammet bedöms ofta som onödigt eller för dyrbart.

Att det är svårt och dyrt att ändra befintliga program beror på att gårdagens programmeringsteknik var mindre utvecklad än dagens teknik. Programmerarna var oftast självlärda, jämför avsnitt 1.1.4 om god och dålig programmering. Dessutom är det så att programmering inte är någon exakt vetenskap utan snarare en konststart. Inget problem har en entydig "rätt programmering" utan denna kan utföras på en mångfald olika sätt. Detta innebär att varje programmerare har sin egen "stil", vilket i sin tur innebär att det kan vara relativt svårt att läsa andras program.

Det har inte varit ovanligt att programmerare blivit outhärliga för ett företag eftersom de har varit de enda som utan anseende ansträngning kunnat "läsa" programmet. Förhoppningsvis är problemen mindre idag, eftersom den tydligare och mer strukturerade programmeringen blivit allt mer spridd. Fortfarande är det dock ett stort avbräck om en nyckelperson byter projekt eller företag.

Risken är rätt stor för detta vad gäller totalsystem eftersom projekten pågår under flera år med flera programmerare inblandade. Dessutom tillkommer det faktum att datorbranschen har en relativt stor personalomsättning.

Detta ställer inte bara till problem vid programframställande utan i ännu högre grad vid programunderhåll. Varje program måste uppdateras och rättas för att inte bli omodernt. Detta kan gälla till exempel nya normkrav eller någon ändring som bör göras för att tillfredsställa önskemål från en användare. Dessutom är det närmast omöjligt att erhålla ett felfritt program. Med fel avses då fel i vid bemärkelse. Exempel på detta är att man i en manual ej talat om att programmet förutsätter att  $g=9.81$ , vilket omöjliggör användandet av alternativa enheter.

Att utföra ändringar i ett stort programsystem kan vara mycket komplicerat, speciellt för ostrukturerade program. Var ändringen skall ske och hur detta påverkar övriga programdelar tar ofta betydligt längre tid att utröna än vad man vid första anblicken tror. En ändring kan dessutom lätt medföra att något annat, och oönskat, också ändras.

Om totalsystemen skall uppfylla ett behov måste de bli stora eftersom användbara program måste vara flexibla. Om programmen inte är flexibla, eller om de "dödar" användarens kreativitet, så kommer programmen att användas enbart i nödfall. Att skapa ett totalsystem som upplevs både som flexibelt och kreativitetsbefrämjande ställer stora krav på programmerarna. Dessa måste delvis tänka som en konstruktör. Dessutom måste de vara programmeringsexperter. Om sådana personer finns så är de nog värda sin vikt i guld.

Ytterligare argument varför totalsystem har en svårighet att bli klara är att söka inom optimeringsdelen. Optimering är en vetenskap som varit "inne" men som allt fler har övergivit. Idag finns det få praktiska exempel på tillfällen där optimering används framgångsrikt. Detta beror på att en optimering med flera variabler är svår att matematiskt utföra men det beror även på att det är mycket svårt att matematiskt formulera optimeringsmål och krav. Till exempel är det nog omöjligt att ge sådan information att programmet kan välja mellan en stål- och en betongbalk. Hur skall man kunna ta med kostnader för brandkrav, återvinning, rivning med mera.

Vi nämnde tidigare att totalsystem kräver superdatorer. Även om detta i många fall är en överdrift så kräver de stora programmen stora datorer. Detta är dock svår-förenligt med den ökande persondatoranvändningen. Dessa datorer återfinns hos de små och medelstora företagen, som dessutom varken har råd att utveckla eller köpa ett totalsystem. Problem som detta hamnar inom diskussionen om centraliserad, decentraliserad eller

distribuerad datorkraft.

Det ökande intresset för persondatorer innebär inte direkt någon "svårighet med totalsystem". Indirekt uppstår dock svårigheter då utvecklingen av totalsystemen, som tidigare nämnts, kräver att de skickligaste programmerarna står till förfogande. Dessa lär nog i stället syssla med den intressantare persondatorutvecklingen. Att utveckla totalsystem, som ju kräver stora datorer, kan komma att upplevas som omodernt och "ute".

Vi har ovan diskuterat och omnämnt ett flertal nackdelar och svårigheter med totalsystem. Vad är då alternativet? Det måste ju finnas ett alternativ ty datoriseringen lär knappast avstanna. Det alternativ vi önskar, och tror på, skall skisseras i avsnitt 11.3 men först skall vi något behandla besluts- och kontrollmöjligheter.

#### 11.2.4 Besluts- och kontrollmöjligheter

Besluts- och kontrollmöjligheter är mycket viktiga vad gäller datorprogram. Det är av denna anledning som de har fått ett eget avsnitt, även om de egentligen likaväl kunde ingå i avsnittet om svårigheter med totalsystem.

I avsnitt 11.2.1 berördes en del krav som man borde ställa på ett datorprogram i fråga om kontrollmöjligheter. Det rekommenderades att resultaten från analysdelen skulle redovisas. Till exempel bör i en takstolsdimensionering moment och tvärkraft i olika snitt vara åtkomliga. Med åtkomliga menas att de lätt kan tas fram då ett eventuellt behov föreligger. Om de skrivs ut för varje lastfall vid varje beräkning uppstår emellertid en konflikt mellan resultatöverskådlighet och kontrollmöjlighet. Detta eftersom man vill ha en begränsad utdatamängd, det vill säga tunna och överskådliga data-listor. Dessa problem diskuterades utförligare i kapitel 10 men nämns även här eftersom de är speciellt aktuella för totalsystem, som ju innehåller flera olika beståndsdelar.

Det bör alltså finnas en möjlighet att ta ut nästan vilket delresultat som helst i vilken punkt som helst. Detta innebär dock att om detta ska vara möjligt så måste detta kunna specificeras i indata till programmet. Samtidigt vill man att indata skall vara enkla att ge. Att för stora system erbjuda en flexibel men begränsad indatamängd är inte helt enkelt. Som jämförelse kan tas stora FEM-program, till exempel ANSYS som omnämndes i avsnitt 10.3.2, för dynamisk eller icke-linjär analys. De program som existerar på marknaden idag uppfyller knappast ovan nämnda önskemål. Att märka är då att de trots allt bara innehåller analysdelen, om än i en väldigt omfattande form.



Som ett annat exempel på problem som kan uppstå i ett totalsystem betraktar vi en tänkt optimeringsdel. Antag att programmet skall välja lämpligaste balkhöjd och armeringsmängd för en betongbalk. Olika krav och kostnader har specificerats som programmet skall uppfylla. Detta innebär att programmet måste iterera sig fram till en optimal lösning. Att utföra en optimal iterering är trots datorns snabbhet att räkna en oerhört svår och omfattande uppgift.

Eftersom krav sällan är helt fixerade så är det en rimlig önskan att användaren skall kunna se vilka alternativ som provats och deras resultat. Om alla dessa skrivs ut så kommer åtskilliga, i användarens tycke, onödiga alternativ att presenteras och utdatamängden att bli omfattande. Om detta dessutom skall skrivas ut för varje konstruktionselement i systemet, förutom alla de övriga resultaten, är en överskådlig resultatframställning nästan omöjlig att erhålla.

Istället väljer då sannolikt användaren att inte skriva ut delresultaten. Resultaten är då mycket svårkontrollerade och användaren - och kontrollanten - är hänvisade till att blint tro på de slutresultat som datorprogrammet kommit fram till.

Eftersom dagens datorprogram inte är felfria, eller saknar bugs som det heter på datorspråk, så lär knappast morgondagens stora totalsystem bli det heller. Vad gäller åtgärder mot felkällor så har detta diskuterats tidigare, avsnitt 10.4. I kapitel 10 påpekades också att resultaten måste kontrolleras, hur nu detta ska gå till om delresultaten inte blivit redovisade?

Vad som ovan diskuterats gällande kontroll gäller i stort också för beslut. Det är ju så att den som är ansvarig för projekteringen även vill besluta om vad som ska göras eller vilket alternativ som ska väljas. Även om denne inte direkt tar alla beslut själv, så bör ändå möjligheter till detta finnas samt en möjlighet till en övergripande kontroll över hela beslutsprocessen.

Av samma skäl som för kontrollen, det vill säga av flexibilitets- och utrymmesskäl, så är det svårt att skapa totalsystem som överlåter besluten på användaren.

Beslut grundar sig vanligtvis på såväl kvantifierbara som icke kvantifierbara faktorer. Dessutom är beslut oftast subjektiva. Detta har inneburit att det har varit svårt att datorisera beslutspunkterna i en projektering. Om detta ändå görs blir det lätt så att tonvikten läggs på de kvantifierbara faktorerna, till exempel kostnad per  $m^3$  betong, medan de icke kvantifierbara faktorerna, till exempel värdet av våningshöjden 2.70m istället för 2.80m, försummas. Detta medför att besluten anses som för "teoretiska" och utan praktiskt värde.



Det har tidigare påtalats att det är konstruktören som är ansvarig för att resultatet är riktigt. Att då som konstruktör använda ett program där det är svårt att kontrollera resultatet, och där olika beslut tas som inte går att påverka, är därmed knappast rekommendabelt. Det är då man kan börja tala om att "datorerna har tagit över herraväldet" (fast det ju egentligen är programmerarna).

### 11.3 Alternativa ideer

De ovan påtalade svårigheterna och nackdelarna med totalsystem bedömer vi så allvarliga att alternativa ideer bör undersökas.

Först skall emellertid göras klart att det finns en gräns för totalsystemens omfattning då systemen blir för stora och svårbemästrade. Dagens system har i de flesta fallen inte nått upp till denna gräns, men så har de också bara uppfyllt de lägsta visionerna.

Vi bedömer att gränsen går någonstans mellan system för en viss konstruktionstyp och de mer generella systemen. Uttryckt i visionerna enligt avsnitt 11.1 går gränsen mellan de två "lägsta" visionerna. Som exempel kan nämnas att program för trätakstolar, stålskorstenar och liknande kan anses acceptabla. Man bör dock inte gå mycket längre än vad dessa har gjort.

Eftersom vi inte kan, eller vill, stoppa datoriseringen måste alternativ sökas som till en stor del utnyttjar datorerna samtidigt som nackdelarna med totalsystemen undviks. Tre av de väsentligaste nackdelarna, som identifierades i avsnitt 11.2, var att programmen blir för stora samt att de är svåra att förena med besluts- och kontrollönskemål.

En lösning som uppfyller detta är då man använder ett antal olika fristående delprogram, moduler, som var och en utför någon specifik del av en konstruktionsberäkning. Detta ger begränsade programstorlekar samtidigt som alla beslut kan tas utanför datorn, det vill säga i gränssnittet mellan utdata från en modul och indata till en annan.

Storleken på modulerna, som givetvis tillåts variera, bör anpassas så att programmen blir överskådliga samtidigt som informationsmängden inte blir för stor. Om modulerna väljs för små blir informationsflödet alltför komplicerat och beräkningarna oekonomiska.

Med ett modulsystem uppnås en stor flexibilitet. En del beräkningar kan utföras för hand medan andra kan datorberäknas, allt efter tycke och smak. Detta innebär att konstruktören själv väljer och bestämmer beräkningsgången och vilka moduler som ska användas. Dessutom kan beräkningarna anpassas och korrigeras under beräkningens gång, om det är önskvärt, beroende på utdata från använda moduler.

Ett modulsystem är också fördelaktigt ur kontrollsynpunkt. Delresultaten är ju lätt åtkomliga och inga väsentliga steg överhoppas.

Modulerna, som kan användas för vitt skilda beräkningar, ger också en flexibilitet i programmeringen. Detta gör att dubbelarbete, där samma sak programmeras i olika program, kan undvikas. Dessutom är det enklare att rätta, korrigera eller utöka en modul än vad som är fallet för stora programsystem. Om till exempel någon norm ändras så behöver endast ingrepp göras i en modul.

Det finns många likheter mellan modulsystem och programsystem där man systematiskt använder i storlek begränsade underrutiner (subrutiner). I båda fallen har man begränsade, väl specificerade och utbytbara beståndsdelar. Främsta skillnaden är dock att ett modulsystem tillåter att alla besluts- och kontrollpunkter ligger utanför datorn.

Man kan också jämföra ett modulsystem med existerande subrutinpaket för matematiska beräkningar, till exempel NAG eller IMSL. Det finns i dessa subrutinpaket rutiner för till exempel matrisinvertering eller för lösning av någon differentialekvation. Systemen innebär att programmerarna kan utnyttjas mer effektivt och på de områden de är bäst. Emellertid bör nog modulerna utföra mer än vad en enstaka matematisk subrutin gör. Annars blir modulerna för små och "hopplockandet" till en beräkning blir för komplicerat för "vanliga" konstruktörer.

Hittills har endast fördelarna med ett modulsystem nämnts, men givetvis finns det även nackdelar och frågetecken.

Samtidigt som modulerna ger en stor flexibilitet, och därmed kan upplevas som kreativitetsfrämjande, så kan paradoxalt nog systemet innebära en stelhet och vara innovationshämmande. Om det blir för strikt standard på ett rekommenderat modulutseende kan så bli fallet. En bättre modul, som dock inte uppfyller standarden, kan bli svåränvändbar bara därför att den inte passar in i systemet.

I och med detta kommer vi in på frågan om vem som ska bestämma hur modulerna ska vara beskaffade. Det finns säkert åtskilliga olika "skolor" som kommer att ha bestämda åsikter, vilket kan ge en prestigekamp istället för samarbete. Man behöver dock inte driva standardiseringen för långt utan kan tillåta att modulerna ser litet olika ut. Huvudsaken är att indata, utdata och beräkningsbeskrivningarna för modulerna blir genomarbetade, väldefinierade och lättolkade.

Det finns dock risker för feltolkning av resultaten från en modul i och med att det är flera olika programmerare inblandade. Om modulerna dessutom blir för små

blir in- och utdatamängden växande och risken för feltolkning ökar. Detta eftersom det då finns fler tillfällen för feltolkningar men även beroende på att man inte orkar, eller har tid, att sätta sig in i hur varje modul fungerar.

Ett STU-finansierat projekt rörande databaser för FE-program, se avsnitt 8.2.2, använde samma grundtankar som ovan föreslagna modulsystem. Det vill säga väl strukturerade grundenheter som fritt kan användas och sammankopplas. Projektet har dock inte fått någon fortsättning eftersom det inte fick någon externfinansiering. Att de inte fick detta kan kanske tolkas som att man ute i industrin inte anser ideerna intressanta. Eventuellt beror det också delvis på att olika företag inte så gärna samarbetar. De vill ofta välja sin egen utveckling där dessutom programvaran kan användas i konkurrenssyfte.

Ett problem med modulsystem är att man redan har kommit långt i datoriseringen av konstruktionsberäkningar. Att ändra riktning kan därmed vara svårt och tidsödande. Hur man ska kunna använda redan existerande program eller programdelar måste utrönas.

Att skapa ett väl uppbyggt modulbiblotek tar lång tid och är knappast realiserbart inom de närmaste åren. Det viktiga är dock att man försöker bryta utvecklingen mot stora totalsystem och istället satsar på fristående och mer flexibla enheter.

## 12 LITTERATUR FÖR VIDARE STUDIER

Den främsta avsikten med denna rapport har varit att ge vissa grundläggande kunskaper inom databehandlingsområdet för byggnadskonstruktörer. Inom den begränsade ram som en bok av denna typ sätter kan naturligtvis ingen fullständig lärobok skapas, utan många delar av området har täckts dåligt eller inte alls. Som en avslutning tänker vi därför ge en litteraturlista för fortsatta studier. För att ge läsaren en möjlighet att själv välja lämplig, och lagom långtgående litteratur skall vi dessutom försöka ge en kortfattad beskrivning av de böcker vi tar upp, och inte bara författare och namn.

För att få en viss struktur på referenserna är de upptagna i en ordning som överensstämmer med bokens kapiteldisposition, med ett avsnitt i detta kapitel, för varje kapitel i boken.

### 12.1 Projekteringsprocessen

En relativt god genomgång av, framförallt, arkitektens roll i den datorbaserade projekteringen ges i "Datorstödd projektering" av Keijer, Strand och Wåhlström, KTH, Projekteringsmetodik, 1981. I denna ges också ett stort antal referenser.

### 12.2 Hård- och mjukvara

En hel mängd litteratur existerar om datorns funktion, såväl ur hård- som mjukvarusynvinkel. En beskrivning av mikrodatorns ABC80 elektronik finns i "Mikrodatorns ABC" av Markesjö, Esselte Studium, 1978. Denna bok ger en lättfattlig beskrivning av datorns funktion, som också gäller för andra datortyper.

I de olika programmeringsspråken finns ett stort antal kursböcker, till exempel en hel serie utgiven av Studentlitteratur.

För det vanligaste programmeringsspråket, FORTRAN, finns i denna serie "Programmering i FORTRAN 77" av Ekman & Eriksson, Studentlitteratur, 1981.

För BASIC, det vanligaste språket för mikrodatorer, gäller tyvärr en förbistring, såtillvida att varje dator har sin egen variant av språket. Därför är det svårt att rekommendera någon speciell bok, utan användaren hänvisas till de manualer som gäller för den egna datorn.

### 12.3 Programsammanställningar

I kapitel 3 refereras ett stort antal olika inventeringar av datorprogram för olika tillämpningar, både nyare och äldre.

Några allmänna synpunkter på dokumentation och beskrivning av program ges i "Bergamo-rapporten", det vill säga rapporten från IABSE Colloquium, Zurich, 1978. Bergamo.

Rekommendationer och hjälp inför investeringar i datorkraft ges i "Inför datorköpet" av Boden, Ingenjörsförbundet, 1981.

### 12.4 Statiska beräkningsmetoder

Grunderna för datorberäkningsmetoder ges i flera olika kompendier från de Tekniska Högskolorna, till exempel från kurserna vid avdelningen för "Byggnadsmekanik" vid Lunds Tekniska Högskola, LTH.

Vid institutionen för Brobyggnad, KTH, utges bland annat ett kompendium i "Matris- och Datormetodik" av Eriksson & Jahlenius, vilket bildat grund till kapitel 4 och i viss mån 5 av denna bok.

Motsvarande avsnitt behandlas också, som ett första steg mot stabilitetsberäkningar för flervåningshus, i kompendiet "Stabilisering av byggnader" av Lorentsen.

### 12.5 Finita Element Metoder

Litteraturen om Finita Elementmetoder för olika tillämpningar är oräknelig. Först under senare år har dock några lämpliga grundläggande böcker kommit. Även här finns till exempel en del material från högskolorna, till exempel "Finita Element Metoden från grunden" av Eriksson, KTH Stålbyggnad, 1981.

På svenska finns även "Finita Elementmetoden, en datoranpassad beräkningsmetod för ingenjörproblem", redigerad av Wiberg, LiberLäromedel Lund, 1975.

Utmärkta böcker om metodens möjligheter har skrivits av Hinton & Owen, till exempel "An Introduction to Finite Element Computations", Pineridge Press, 1979, och - mera programmeringsinriktad - "Finite Element Programming", Academic Press, 1977.



En mycket bra introduktionsbok är också "A Practical Introduction to Finite Element Analysis" av Cheung & Yeo, Pitman, 1979.

Ingen sammanställning av FE litteratur är fullständig utan nämnande av "The Finite Element Method" av Zienkiewicz, McGraw Hill, 1977. Denna bok innehåller en stor samling tillämpningar av FE metodiken, men kan kanske inte direkt sägas vara en lärobok.

När det gäller de mer teoretiska aspekterna på FEM är "An Analysis of the Finite Element Method" av Strang & Fix, New Jersey, 1973 en utmärkt om än något svårläst bok.

Något mera praktiskt inriktad är "Numerical Methods in Finite Element Analysis" av Bathe & Wilson, New Jersey, 1976.

När det gäller datorprogram för enkel FE analys rekommenderas en studie av manualen till programmet "CALFEM", LTH Byggnadsmekanik, 1981.

## 12.6 Analysprogram

Det mest spridda programpaketet för analys är CHALMFEM-programmen, utvecklade vid Chalmers Tekniska Högskola, CTH. Programmen finns tillgängliga, bland annat via Göteborgs Datacentral. Bland programmen kan nämnas "SERFEM" för skivor, "PREFEM" för plattor, "PRAM" för plana ramar, "PFVIBAT" för ramdynamik och "GENFEM" som är ett generellt FE program.

Programmet "ANSYS" beskrivs i en manual, utgiven av Swanson Analysis Systems, Inc, och finns tillgängligt via Control Data AB.

Utvecklingsprogrammet "V-FEM" är beskrivet i en manual av Eriksson, KTH 1981, och finns tillgängligt vid sektion V:s datorcentral, KTH.

Ett antal program för icke-linjär statisk och transient analys finns beskrivna i "Finite Elements in Plasticity - Theory and Applications" av Owen & Hinton, Swansea, 1980.

## 12.7 Dimensioneringsprogram

Programmet "SEKTION" är utvecklat av Olle Ersvik, och finns beskrivet i en manual utgiven av VBB, 1980.

Programmet "BETONGBALK" ingår i BERIT-systemet och finns beskrivet i en manual från AB Jacobson & Widmark.

Programmet "SKORSTEN" är utvecklat av Göran Alpsten, och finns beskrivet i en användarmanual utgiven av Stålbyggnadskontroll AB, 1981 (provutgåva).

Programmet "TRP200" är utvecklat av Tekn dr Arne Johnson Ingenjörbyrå ab för PLANNJA AB, 1982.

Information om datorberäkning av limträkonstruktioner har erhållits från Töreboda Limträ AB.

Programmen "BITAK" och "BITRAN", med varianter, för takstolsberäkningar finns beskrivna i manualer utgivna av Bjerking Ingenjörbyrå AB och Nordisk Kartro AB.

## 12.8 Text och databaser

Det dataregister över byggnadsmaterial som Svensk Byggtjänst har lagt upp överensstämmer i princip med "Byggvarukatalogen" som utges med jämna mellanrum.

Programmet "FEMGEN" finns beskrivet i en "User Manual" utgiven av datacentralen vid Lunds Universitet.

## 12.9 Ritningsframställning, grafik

En grafisk, interaktiv metodik för framställning av indata till FE program diskuteras i "Contributions to Interactive Computer Aided Design in Structural Engineering" av Keijer, KTH Brobyggnad, 1980.

En utvärdering av CAD i byggbranschen ges i "Datorstödd projektering. CAD i tillämpning" av Christiansson m fl, Byggeforskningsrådet, 1983.

Ett antal kursdagar om CAD ordnades under 1982, bland annat "Var står CAD idag? Ett seminarium om CAD i byggprojekteringen - utblick, dagsläge och tendenser" anordnat av tidskriften Byggmästaren.

En intresseförening för grafisk databehandling, SIGRAD, finns i Sverige, som förmedlar information inom detta område.

### 12.10 Risker med datoranvändning

Som nämndes redan i förordet har det projekt som redovisas i denna bok, och andra projekt vid LTH, initierats med utgångspunkt från BFR-rapporten "Tillförlitlighet och kommunikation i en datoriserad byggbransch" ( R35:1982 ), av Christiansson m fl.

### 12.11 Totalsystem

Optimeringsproblematiken och dess matematiska formulering behandlas till exempel i boken "Optimum Structural Design. Theory and Applications", redigerad av Gallagher & Zienkiewicz, John Wiley, 1977.

### 12.12 Matristeori

En utmärkt bok om arbete med matriser, ur en mera matematisk synvinkel är "Linear Algebra and its Applications" av Strang, Academic Press, 1976.

En bok på svenska om numeriska metoder som också kan rekommenderas är "Numeriska Metoder" av Björck & Dahlquist, CWK Gleerup, 1969.



## APPENDIX - Matriser och ekvationssystem

I främst kapitlen 4 och 5 behandlades statistiska beräkningsmetoder. Såväl kraft- som förskjutningsmetodiken visades - i det generella fallet - resultera i linjära ekvationssystem för lösning av krafterna eller förskjutningarna i strukturen. Dessa ekvationssystem är exempel på användning av matriser och vektorer. För att öka förståelsen för de uppställda sambanden skall detta appendix kortfattat redogöra för en del teoretiska och praktiska aspekter på denna form av linjär algebra.

### A.1 Matriser

Låt oss först, för att skapa en grund för den vidare diskussionen definiera en matris.

En korrekt definition av en matris diskuteras i läroböcker i Linjär Algebra. I detta sammanhang kan vi dock nöja oss med att konstatera att en matris är en samling tal ordnade i en rektangel med  $m$  stycken rader och  $n$  stycken kolumner. För att talsamlingen skall vara en matris skall vi dock kräva ytterligare två saker. Dels att de olika talens plats i matrisen skall ha någon betydelse - det vill säga att talen inte bara är utplacerade i en slumpmässig rektangelform. Dessutom skall vi kräva att vissa räkneregler, exempelvis addition och multiplikation går att utföra, och att dessa operationer har någon mening.

Man talar om matrisens dimension i samband med talen  $m$  och  $n$ , det vill säga matrisens storlek. En vanlig beteckning för exempelvis dimensionen  $m=4$ ,  $n=2$  är  $(4/2)$ , och ett exempel på en sådan matris med (vilket är ett specialfall) endast heltal är

$$(A) = \begin{bmatrix} 1 & 2 \\ 4 & 5 \\ 9 & 8 \\ 13 & -4 \end{bmatrix}$$

I de fall då ett speciellt tal i matrisen behöver omnämnas brukar det betecknas med  $a_{ij}$  vilket då avser talet i rad  $i$  i kolumn  $j$  i matrisen  $(A)$ .

Ett viktigt specialfall av matris har en kolumn -  $(m/1)$  - eller en rad -  $(1/n)$ . Man talar om kolumn- resp radvektorer. Den enda variant vi har någon nytta av är kolumnvektorn, och vi inför beteckningen  $\bar{x}$  för vektorn  $x$ .



Ett annat användbart specialfall av matris är den kvadratiske för vilken antalet rader är lika med antalet kolumner. Om det dessutom för alla värden på  $i$  och  $j$  gäller att talet i matrisens  $i$ :te rad,  $j$ :te kolumn är lika med talet i  $j$ :te raden,  $i$ :te kolumnen, det vill säga att  $a_{ij} = a_{ji}$ , sägs matrisen vara symmetrisk.

## A.2 Räkeregler för matriser

Med matriser kan man göra nästan allting som man kan göra med vanliga, skalära tal - dock inte på samma sätt och endast under vissa villkor.

Två matriser kan adderas om de är av samma dimension. Varje tal i summamatrisen är lika med summan av talen på motsvarande plats i de två adderade matriserna.

Motsvarande gäller för subtraktion.

Att multiplicera två matriser är avgjort mera komplicerat än att addera eller subtrahera dem. Villkoret för att det skall gå att multiplicera matriserna (A) och (B) är att (A) har lika många kolumner som (B) har rader. Exempelvis kan multiplikationen  $(C) = (A)(B)$  utföras om (A) har dimensionen  $(3/2)$  och (B) dimensionen  $(2/4)$ . Den resulterande matrisen (C) ärver antalet rader från (A) och antalet kolumner från (B), d v s i exemplet blir (C) av dimensionen  $(3/4)$ . Observera att med de givna matriserna matrismultiplikationen (B)(A) ej går att utföra.

För multiplikationen ovan ges en formel för komponenterna i resultatmatrisen (C) som

$$c_{ij} = \sum_{k=1}^n (a_{ik} b_{kj})$$

vilket innebär att talen i  $i$ :te raden av (A) multipliceras med talen i  $j$ :te kolumnen av (B). Summationsgränsen  $n$  är lika med antalet kolumner i (A) vilket alltså måste vara lika med antalet rader i (B).

## A.3 Linjära ekvationssystem

Ovanstående villkor för matrismultiplikation utnyttjas vid formulering i matrisform av ett linjärt ekvationssystem.

Betrakta till exempel ekvationssystemet i avsnitt 4.6.1. I bakgrunden för förskjutningsmetodiken sades att målsättningen med analysen var att formulera ett samband av typen

$$(K) \bar{r} = \bar{R}$$

där (K) är en styvhetsmatris av dimensionen (n/n),  $\bar{r}$  en vektor innehållande förskjutningar i de n stycken frihetsgraderna och  $\bar{R}$  en vektor med korresponderande lastvärden.

Med hänsyn till att de ingående matriserna kan expanderas som

$$(K) = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \quad \bar{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \quad \bar{R} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}$$

kan matrisprodukten i vänsterledet lätt utvärderas som de respektive raderna i (K) multiplicerade med den första (och enda) kolumnen i  $\bar{r}$ . Resultatet blir en (3/1) matris med innehållet

$$(K) \bar{r} = \begin{bmatrix} k_{11}r_1 + k_{12}r_2 + k_{13}r_3 \\ k_{21}r_1 + k_{22}r_2 + k_{23}r_3 \\ k_{31}r_1 + k_{32}r_2 + k_{33}r_3 \end{bmatrix}$$

Eftersom denna vektor skall vara lika med kraftvektorn i varje komponent, inses att den kompakta matris-vektor ekvationen motsvarar ett ekvationssystem

$$\begin{aligned} k_{11}r_1 + k_{12}r_2 + k_{13}r_3 &= R_1 \\ k_{21}r_1 + k_{22}r_2 + k_{23}r_3 &= R_2 \\ k_{31}r_1 + k_{32}r_2 + k_{33}r_3 &= R_3 \end{aligned}$$

#### A.4 Lösning av ekvationssystem

Ur kapitel 4 och 5 framgick hur ekvationssystem uppstod vid statistiska beräkningar. I avsnittet A.3 ovan har visats ett kompakt sätt att beskriva dessa system. Det är emellertid även nödvändigt att kunna lösa de system som uppstår. För detta måste en dator utnyttjas om de betraktade ekvationssystemen är av någon storlek. Hjälpprogram för detta finns i de flesta datorer om man själv skriver ett program. Det är emellertid ändå viktigt att känna till en del om den bakomliggande teorin för att på ett bra sätt kunna utnyttja de tillgängliga rutinerna.

Antag att ett ekvationssystem på formen

$$(K) \bar{r} = \bar{R}$$

betraktas, och antag att systemet verkligen är lösbart (vilket det måste vara om förskjutningsmetoden använts och riktiga randvillkor införts). Systemet kan formellt lösas som

$$\bar{R} = (K)^{-1} \bar{r}$$

där  $(K)^{-1}$  är inversen till matrisen  $(K)$ . Att göra på så sätt är dock i de flesta fall inte lämpligt, då det kräver ett väldigt stort räknearbete för datorn.

Det bästa sättet att lösa systemet är i stället med hjälp av så kallad Gauss-elimination. I de fall man skall behandla olika lastfall på en viss struktur (det vill säga olika högerledsvektorer) kan det vara fördelaktigt att i stället göra en så kallad LR-faktorisering av styvhetsmatrisen, vilket kraftigt minskar räknearbetet från och med det andra fallet. LR-faktorisering kan göras med flera olika metoder, till exempel Gauss, Choleski, Crout eller Doolittle.

Vissa iterativa lösningsmetoder finns också tillgängliga för lösning av ekvationssystem. De är dock oftast inte speciellt lämpliga för de fall som blir aktuella i statisk analys. Om man har ont om primärminneskapacitet i datorn kan de dock möjligen vara attraktiva.

Om man skall utnyttja de rutiner som finns tillgängliga i datorn är det viktigt att känna till de olika egenskaper som ekvationssystemet har. Som har konstaterats tidigare så är ju till exempel såväl flexibilitets- som styvhetsmatriser symmetriska, vilket kan visas vara en följd av Maxwell's sats. Denna egenskap kan utnyttjas vid lösning av ekvationssystemen för att spara såväl minnesutrymme som beräkningstid.

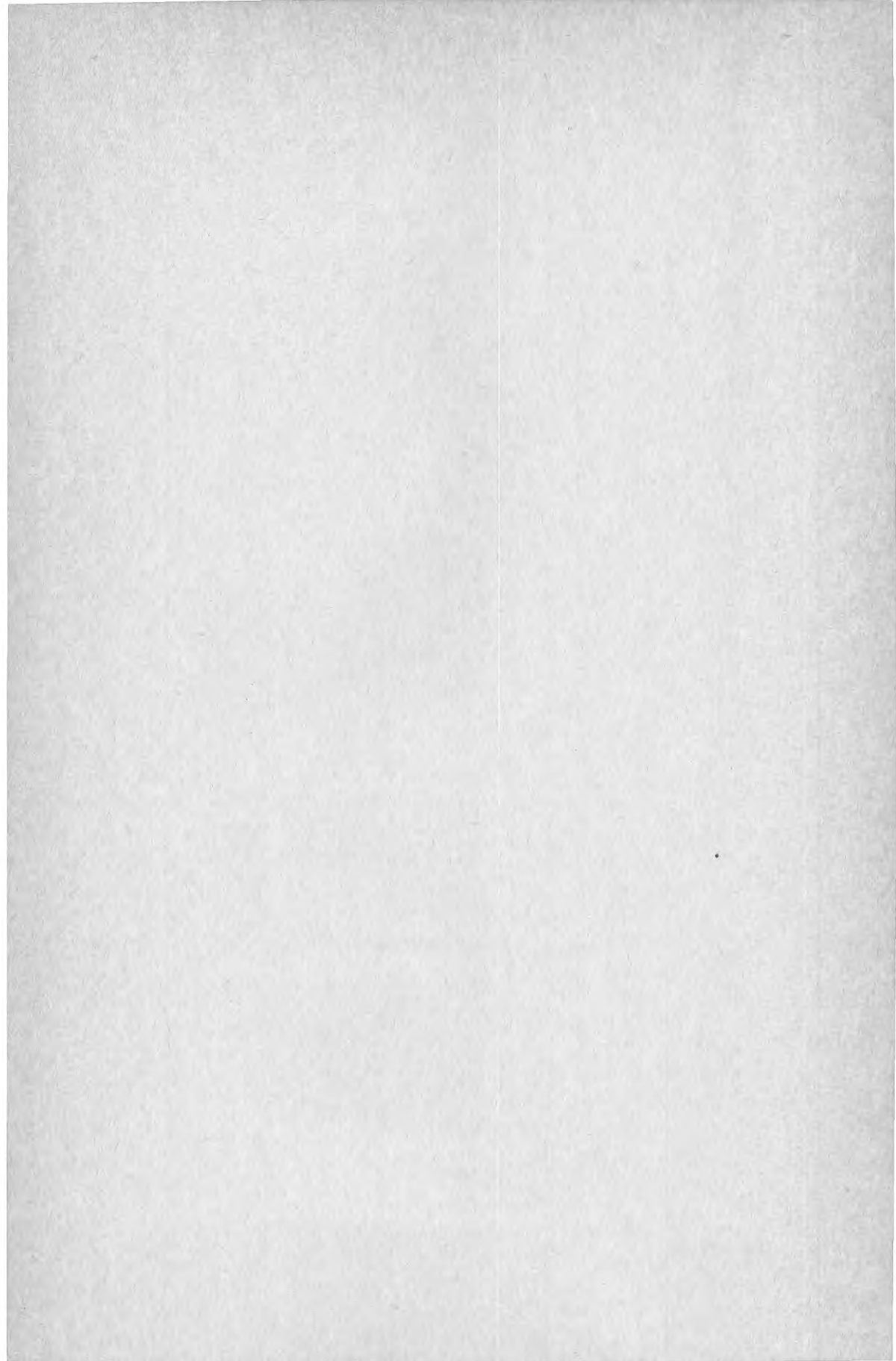
Ytterligare besparingar kan göras då man arbetar med förskjutningsmetoder, och i vissa fall med kraftmetoden. Den matris som formuleras kan nämligen fås bandad, vilket innebär att matrisen innehåller ett stort antal nollor i de övre högra och nedre vänstra hörnen. Exempelvis kan man genom olika val av nodnumrering i Finita Element beräkningar få det band längs matrisens diagonal som innehåller tal skilda från noll, och som alltså måste behandlas, bli av större eller mindre bredd. Mycket stora besparingar i beräkningstid och lagringsutrymme kan fås om detta utnyttjas på rätt sätt.

En viktig beräkningsmässig skillnad mellan kraftmetoden och förskjutningsmetoden finns också, som påverkar valet av strategi för lösning av ekvationssystemet. För en linjär beräkning med förskjutningsmetodik fås

nämligen en positivt definit styvhetsmatris. Denna egenskap innebär teoretiskt att matrisen endast har positiva egenvärden, och praktiskt att lösningen av ekvationssystemet är ett "väl-konditionerat", det vill säga snällt, problem. Denna egenskap kan inte påräknas vid användning av kraftmetoden. I själva verket kan flexibilitetsmatrisen även för, som man tycker, enkla problem bli så "illa-konditionerad" att den lösning som fås är mycket dålig. Detta gäller dock knappast för beräkning med Clapeyron's ekvation, se avsnitt 4.5.1.









**Denna rapport hänför sig till forskningsanslag  
820325-4 från Statens råd för byggnadsforskning  
till Institutionen för brobyggnad, Tekniska  
högskolan, Stockholm.**

**Art.nr: 6700837**

**Abonnemangsgrupp:  
Z. Konstruktioner och material**

**Distribution:  
Svensk Byggtjänst, Box 7853  
103 99 Stockholm**

**Cirkapris: 50 kr exkl moms**

**R137: 1983**

**ISBN 91-540-4025-6**

**Statens råd för byggnadsforskning, Stockholm**