RÅDET FÖR HÖGRE UTBILDNING
COUNCIL FOR THE RENEWAL OF HIGHER EDUCATION

Project number: 075/93
Name: Dr Bo Johansson
Institution: Chalmers University of Technology
Dept of Mathematics
S-412 96 Gothenburg
Tel: +46 (0)31 772 35 35
E-mail: bo@math.chalmers.se

Multimedia Project in Mathematics

Abstract

A common problem in all education is the learning strategy of the students and the level of their knowledge. Teachers in general have given this problem a lot of thought. For more than a decade now, mathematics has got a new tool with an incredible potential. This tool, the computer, is today an essential instrument in research, and will play a fundamental role in the undergraduate programme in the future.

Over the past years there has been a tremendous development of powerful software giving the possibilities to work numerically as well as symbolically in Mathematics. This kind of software could work as a supplementary tool to pen and paper. With such an instrument the students would e.g. be able to examine different models and analyse them.

In order to carry this through it is necessary to have a good understanding of this kind of software, which is not to be expected of the student. Furthermore, imagination and good insight in the subject of Mathematics are necessary in order to formulate interesting questions as well as relevant problems.
To attain this method of studying and working we are implementing a software package containing:

- the possibility of tackling the subject matter of a laboratory project in different ways, depending on questions, problems, facts, the student's own ideas, et cetera
- a general interface to applications such as Matlab, Maple, Mathematica and others
- an interface to the user (the student) with possibilities to use text, graphics in the form of both animations and visualizations, and in a later phase also sound, video, ...
- an interface to the teacher which makes the material easy to handle.

Such software will make it possible to create a laboratory environment reflecting the world of multimedia and steered from above as well as by one's own initiative. The interface to the user will not demand any knowledge of the underlying system and will be easy to handle also for the uninitiated.

Thus there would be several entry paths into the mathematical world with different tracks to choose among, all depending on interest and goal. During the journey there will be help available, references to read, hypotheses to formulate, problems to analyse and solve. All this will be done with the possibilities of simple as well as more advanced help. Help will be available on several levels.

The idea is to stimulate the user's curiosity in order to increase his/her mathematical understanding this multimedia mathematical world. The possibilities for the students to detect phenomena that they did not even think about previously will increase. Access to visualization will increase his stimulation and widen and enlarge understanding. This will strengthen the learning process in an efficient way with respect to understanding, time of learning and lasting knowledge. Ultimately, the work may culminate in natural generalizations with a wider perspective.

This project should enable the students to profit from the technological progress within the mathematical area during the latest years and increase the students' ability to learn efficiently and creatively. The pedagogical approach is to let the senses become more efficiently stimulated in the learning process and giving the support for picking up knowledge throughout. This way of learning appeals to one's own initiatives and independent seeking of knowledge.

Homepage: http://www.math.chalmers.se/~bo/

Projektredovisning*

Bo I Johansson

Maj 2000

Sammanfattning

Projektet *Matematik i en multimedial värld* startade hösten 1994 och avslutades, väsentligen, 1998. Projektet hade till syfte att på ett nytt sätt integrera datorn i matematikundervisningen, men också skapa möjligheter att interaktivt experimentera med matematik, inom givna ramar.

En grundidé har varit att skapa en nyfikenhet hos användaren, i första hand studenten, vilken naturligt kommer att ställa sig ett antal frågor som vederbörande inte kan avhålla sig från att analysera. Detta senare först och främst genom datorn, men i förlängningen också genom traditionellt matematiskt hantverk.

Bakgrund

Sedan flera år pågår en intensiv diskussion om ämnesinnehåll, inlärningsmetoder, datorberäkningar, etc inom matematikutbildningen vid Chalmers och Göteborgs universitet. Det pågår samtidigt ett stort förändringsarbete inom grundutbildningen där många är involverade och deltar med stor entusiasm. Lärarlag har initierats för att bättre kunna forma såväl enskilda kurser som utbildningsprogram.

Generellt kan sägas att vi bedriver kvalitativt experimenterande i form av nya arbetsmetoder inom utbildningen avseende inlärning och problemlösning. Problemformuleringarna är inte sällan numera av en mer djupgående och utvecklande karaktär. Det förekommer simuleringsproblem. Vi försöker även införa mer matematisk modellering, vilket på ett naturligt sätt kan överföra och utveckla den matematiska kunskapen till tillämpade ämnesområden.

Tillgången till datorkraft driver utvecklingen kring frågeställningar som räknefärdighet, examinationsformer, projektarbeten, simuleringar och modellering. Trots

* Grundutbildningsrådet har stött projektet

detta har introducerandet av datorn i undervisningen till dags dato inte uppenbart ökat den matematiska förståelsen eller den matematiska färdigheten hos studenterna. Ej heller har studentgenomströmningen ökat. Användningen av datorer har inte heller reducerat den traditionella examinationen i nämvärd utsträckning.

Däremot har användandet av matematiska datorprogram, framförallt *Matlab*, förbättrat möjligheten att överföra matematiken till tekniska och naturvetenskapliga ämnen genom utnyttjande av samma datorprogram. Den terminologi och beräkningstekniska överföring som på detta vis sker från matematikämnet till tillämpade ämnesområden har en stor potential, vilket kan användas för att fördjupa matematikkunskaperna och förstärka tillämpningarna. Detta måste utvecklas ytterligare.

Datorn har givit matematikämnet en experimentell vinkling som tidigare inte funnits. Detta är någonting som tillför matematiken ytterligare dimensioner och borde kunna användas för att stärka den matematiska förståelsen. Vid sidan av experimentverktyg och laborationsmiljö är datorn också ett mycket kraftfullt hjälpmedel vid såväl symboliska kalkyler som numeriska.

Projektidén

Det gäller att utnyttja de allt kraftfullare datorerna och, inte minst, de sedan länge använda och alltmer betydelsefulla matematiska programpaketen såsom Matlab, Mathematica och Matlab, för att nu nämna några av de vanligaste. Denna typ av program kan fungera som ett verktyg att användas vid sidan av den mer traditionella analysen av matematiska problem, som sällan hinns med i tillfredsställande omfattning. Med ett lämpligt matematikprogram skulle studenten exempelvis kunna hinna testa ett flertal olika matematiska modeller och analysera dessa.

För att kunna genomföra något sådant krävs dock en god insikt i hur denna typ av program fungerar, vilket inte studenten kan förväntas ha. Dessutom krävs fantasi och insiktsfullhet inom det matematiska ämnesområdet för att initialt kunna formulera intressanta frågor och relevanta problem.

Grundidén i detta projekt har varit att skapa förutsättningar för att komplettera de vanliga studierna med detta sätt att studera och arbeta. Vägen till realisering har baserats på att implementera ett programpaket som skall utgöra ett fundament för en laborativ utvecklingsmiljö. Detta program, kallat *Weblab*, har

- ett generellt gränssnitt mot applikationsprogram av typ Matlab, Maple, Mathematica mm
- ett gränssnitt mot användaren (studenten) som ges möjligheter att nyttja såväl text som grafik i form av både animeringar och visualiseringar för att bättre analysera problemställningarna

- ett gränssnitt mot läraren/utvecklaren som på detta sätt skall kunna redigera befintligt material eller utveckla nytt.

Programmet möjliggör skapandet av en laborativ miljö som utspelar sig i en multimedial värld där det kan förekomma både påtvingad styrning och krav på egna initiativ. Gränssnittet mot användaren måste inte nödvändigtvis förutsätta kunskaper om det underliggande matematikprogrammet utan kan göras lätthanterligt även för en oinitierad. Allt beroende hur man utvecklar eller redigerar laborationsmaterialet. Det kan utabetas på ett sätt som ger studenten möjlighet att arbeta sig igenom materialet på ett personligt vis, beroende av frågeställningar, problemuppgifter, faktamaterial, individens egna idéer, etc.

Således ges här en möjligheten att vandra in i en matematisk värld med flera olika stigar att välja mellan, alltefter intresse och målsättning. Under resans gång kan det finnas hjälp att tillgå, referenser att läsa, hypoteser att formulera, problem att analysera och lösa. Allt kan ske med möjligheter till såväl enkel som mer avancerad hjälp. Hjälp som kan ges i flera nivåer om så önskas.

Idén är att på detta sätt stimulera användaren att genom sin egen nyfikenhet öka den matematiska förståelsen allt eftersom denne arbetar vidare. Möjligheterna till att upptäcka fenomen som studenten inte tidigare reflekterat över ökar. Tillgång till visualisering ökar stimulansen, vidgar och befäster förståelsen. Användandet skall förstärka sinnesintrycken och därmed inlärningsprocessen på ett effektivt sätt m a p på förståelse, inlärningstid och bestående kunskap. Historiska beskrivningar och broar mot tillämpningar låter sig också enkelt åskådliggöras. Det hela kan sedan kulminera i naturliga generaliseringar med ett bredare perspektiv.

Systemkrav för datorstödd inlärning

För att analysen av ett problem skall vara tillfredsställande och genomarbetad krävs oftast en skriftig dokumentation. Detta låter sig göras genom att systemet automatisk skapar en dokumentation (i LaTeX) av arbetet/laborationen, vilken också kan/bör kompletteras med lämplig text. Den slutliga dokumentationen av arbetet kan sedan användas som del i examination, men kan även med fördel nyttjas i samband med en allmän presentation eller enbart som diskussionsmaterial.

En annan mycket relevant sak att observera är att programmet Weblab har enhetliga gränssnitt mot applikationsprogramvaror och användare respektive. Studiematerialet skall den enskilde läraren själv kunna utarbeta eller omarbeta allt efter eget önskemål. Detta kräver naturligtvis en viss arbetsinsats, men det skall vara förhållandevis enkelt även för en icke initierad lärare att modifiera, komplettera eller helt förändra kursinnehållet och därmed laborationsinnehållet. I förlängningen betyder

detta att innehållet i en laboration kan selekteras i det närmaste lika enkelt som föreläsaren idag redigerar sitt normala kursinnehåll.

Här skall också påpekas att programvaran *Weblab* inte nödvändigtvis kräver lärarhandledning utan även med fördel kan utnyttjas av den enskilde studenten i sin studielya eller kunna användas i samband med distansundervisning.

Projektet har utvecklas i befintlig miljö vid matematiska institutionen i Göteborg. Detta betyder att produkten kan nyttjas på arbetsstationer med Solaris som operativsystem. En ytterligare restriktion är att implementeringen enbart är gjord för Matlab, men att utan alltför stora ansträngningar kan detta kompletteras så att även andra programpaket kan anropas innefrån *Weblab*.

Realisering

Ovanstående projektidé har realiserats i en programvara, kallad *Weblab*, som är ett gränssnitt mot Matlab, men också mot webläsare. Weblab är uppbygd på så sätt att den utnyttjar en website varur dokumentationen kan nås. Det direkta gränssnittet gentemot användaren är en webläsare, ex.vis Netscape.

För att kunna använda Weblab krävs en för ändamålet konfigurerad website. Se dokumentationen *Weblab, installation och konfigurering*, [1]. Dessutom krävs att programmet *wlserver* har startats. Detta är ett program som hanterar kommunikationen mellan webläsaren och Matlab. Härefter kan användaren anropa den för ändamålet angivna webadressen, ex.vis http://fermat.math.chalmers.se:2223/, vilken är den webserver som materialet har utarbetats på.

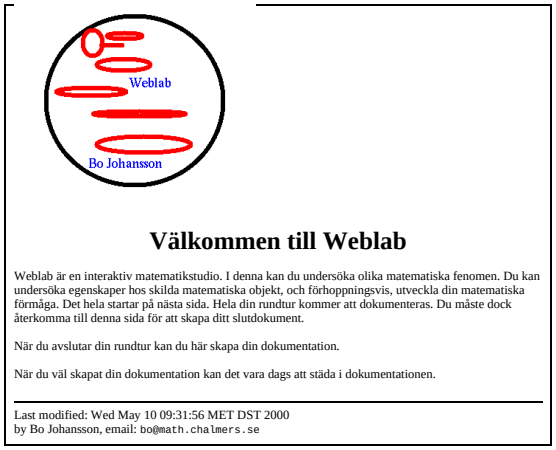
Vi skall nu närmast ge en beskrivning av de möjligheter som ges genom att utnyttja delar av det utarbetade materialet för att på detta sätt beskriva och illustrera de möjligheter som ges med Weblab.

När vi väl har nått Weblabs hemsida får vi då bilden i Figur 1 som enbart fyller funktionen att ge ett ansikte till programmet Weblab.

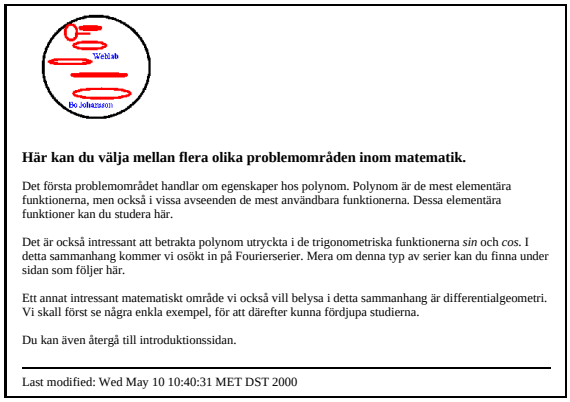
Den verkliga startsidan kommer först när vi har vandrat vidare till nästkommande sida, vilken finns illustrerad i Figur 2. Härifrån kan användaren sedan styra vidare till de matematiska områden/problemställningar denne vill/skall arbeta med.

En av flera möjligheter som ges här är att närmare studera polynom och deras egenskaper. Dessa funktioner är elementära och viktiga att fördjupa sig i. För den något matematiskt bevandrade läsaren är det välkänt hur fundamental familjen av polynomfunktioner är varför detta är ett såväl intressant som spännande problemområde. Dess initialsida är den som visas i Figur 3.

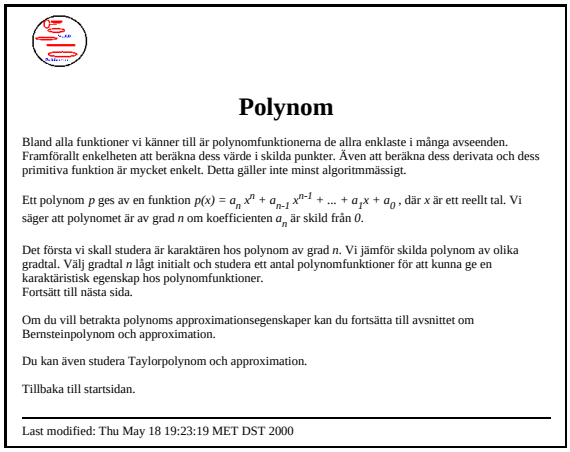
Här ges en kortfattad beskrivning av polynom. Användaren kan sedan vandra vi-



Figur 1: Introduktionssidan för Weblab



Figur 2: Startsidan för Weblab



Polynom

Bland alla funktioner vi känner till är polynomfunktionerna de allra enklaste i många avseenden. Framförallt enkelheten att beräkna dess värde i skilda punkter. Även att beräkna dess derivata och dess primitiva funktion är mycket enkelt. Detta gäller inte minst algoritmmässigt.

Ett polynom p ges av en funktion $p(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$, där x är ett reellt tal. Vi säger att polynomet är av grad n om koefficienten a_n är skild från 0.

Det första vi skall studera är karaktären hos polynom av grad n. Vi jämför skilda polynom av olika gradtal. Välj gradtal n lågt initialt och studera ett antal polynomfunktioner för att kunna ge en karaktäristisk egenskap hos polynomfunktioner.
Fortsätt till nästa sida.

Om du vill betrakta polynoms approximationsegenskaper kan du fortsätta till avsnittet om Bernsteinpolynom och approximation.

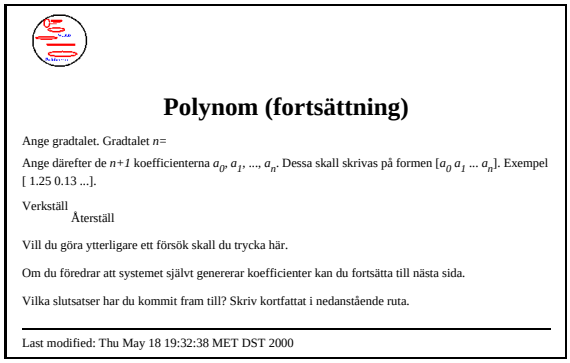
Du kan även studera Taylorpolynom och approximation.

Tillbaka till startsidan.

Last modified: Thu May 18 19:23:19 MET DST 2000

Figur 3: En sida om polynom

dare på flera olika sätt. Exempelvis är det möjligt att studera polynom och karaktäristiska egenskaper kopplade till dess gradtal. Det är också möjligt att studera polynomfunktionernas allmänt approximativa egenskaper till en given kontinuerlig funktion, men också mer speciella approximationer såsom Taylorpolynom och dess approximation av en funktion. Vi väljer här att betrakta karaktäristiska egenskaper hos polynom kopplade till dess gradtal och fram kommer då sidan i Figur 4.



Polynom (fortsättning)

Ange gradtalet. Gradtalet n=

Ange därefter de $n+1$ koefficienterna $a_0, a_1, ..., a_n$. Dessa skall skrivas på formen [a_0 a_1 ... a_n]. Exempel [1.25 0.13 ...].

Verkställ
Återställ

Vill du göra ytterligare ett försök skall du trycka här.

Om du föredrar att systemet självt genererar koefficienter kan du fortsätta till nästa sida.

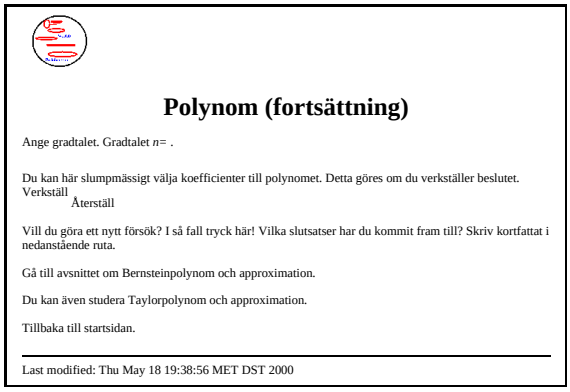
Vilka slutsatser har du kommit fram till? Skriv kortfattat i nedanstående ruta.

Last modified: Thu May 18 19:32:38 MET DST 2000

Figur 4: En sida om polynom och vissa egenskaper

Från denna sida kan användaren generera polynom av skilda gradtal och studera dess grafer. Därefter är det tänkt att användaren (kort) sammanfattar sina slutsatser.

Det är också möjligt att förenkla ovanstående studium genom att slumpmässigt generera polynomets koefficienter, men då krävs att användaren förflyttar sig till en annan angiven sida. Denna senare sida kan ses i Figur 5.



Polynom (fortsättning)

Ange gradtalet. Gradtalet *n*= .

Du kan här slumpmässigt välja koefficienter till polynomet. Detta göres om du verkställer beslutet.
Verkställ
Återställ

Vill du göra ett nytt försök? I så fall tryck här! Vilka slutsatser har du kommit fram till? Skriv kortfattat i nedanstående ruta.

Gå till avsnittet om Bernsteinpolynom och approximation.

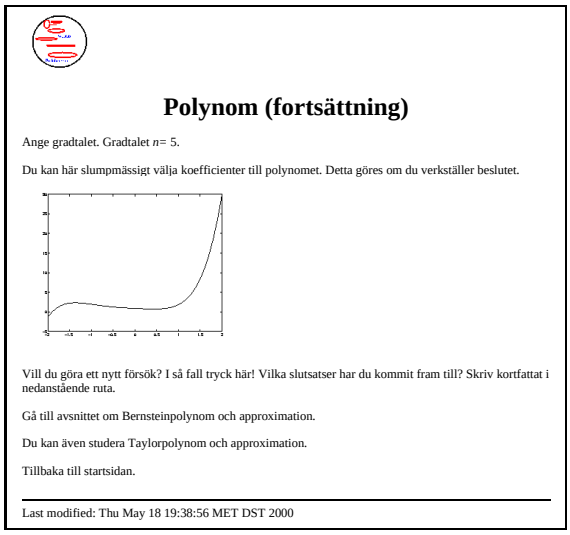
Du kan även studera Taylorpolynom och approximation.

Tillbaka till startsidan.

Last modified: Thu May 18 19:38:56 MET DST 2000

Figur 5: En sida liknande de föregående

Här behövs enbart välja ett gradtal, d v s ett positivt heltal, varefter ett polynom med det angivna gradtalet genereras. I detta fall har vi valt gradtalet till 5. Utfallet av detta kan vi se i Figur 6.



Polynom (fortsättning)

Ange gradtalet. Gradtalet *n*= 5.

Du kan här slumpmässigt välja koefficienter till polynomet. Detta göres om du verkställer beslutet.

Vill du göra ett nytt försök? I så fall tryck här! Vilka slutsatser har du kommit fram till? Skriv kortfattat i nedanstående ruta.

Gå till avsnittet om Bernsteinpolynom och approximation.

Du kan även studera Taylorpolynom och approximation.

Tillbaka till startsidan.

Last modified: Thu May 18 19:38:56 MET DST 2000

Figur 6: Ett resultat

Hela tiden är det dock tänkt att användaren skall kunna göra tvära kast i sitt vandrande i den matematiska djungeln. Det är alltså möjligt att göra långa vandringar eller korta promenader, att sträva framåt eller vandra tillbaka. För att slutligt dokumentera den matematiska vandringen måste användaren återgå till introduktionssidan varifrån det är möjligt att skapa ett dokument.

Vi har här exemplifierat för att visa de möjligheter som finns tillgängliga med detta program.

Erfarenhet

Materialet i detta projekt har använts till studenter för att testa ideérna och samtidigt få en tydlig återkoppling på dess värde. Resultaten har varit uppmuntrande och har visat att detta projekt definitivt är en föregångare som har visat goda möjligheter att fortsätta att exploatera detta ännu relativt obeträdda område.

Det kan också nämnas att detta material kan vidareutvecklas för att utnyttjas i samband med distansutbildning, men också i mindre ambitiösa projekt såsom interaktiva websidor riktade mot exempelvis gymnasister som på detta sätt skulle kunna stimuleras inom matematikområdet. Det är inte särskilt svårt att sätta upp en gymnasieportal där elever (och naturligtvis även andra intresserade) kan ges spännande och utmanande matematiska problem. Detta är något som skulle vara mycket spännande att genomföra. Diskussion om dess genomförande har redan förekommit.

Referenser

[1] Johansson, Bo I, *Weblab, installation och konfigurering*

Weblab
Installation och konfigurering

Bo I Johansson

Maj 2000

Introduktion

Detta programpaket, kallat *Weblab* har utvecklats för att erhålla ett webbaserat gränssnitt mot etablerade matematiska program såsom *Matlab*, *Mathematica*, etc. Enbart kommunikation gentemot Matlab är för närvarnade implementerat.

Ytterligare fokus har inriktats mot att detta programpaket, d v s Weblab, utöver gränssnittet, även skall inkludera en struktur som möjliggör ett sätt att skriva interaktiv dokumentation, vilken bl a kan användas för att skapa datorlaborationer inom matematik.

Detta dokument har till syfte att beskriva hur Weblab installeras, konfigureras och administreras samt ge en beskrivning av hur en webserver kopplas mot weblab. Dessutom kommer vi att beskriva hur man mste skriva sina dokument.

Allmänt

Vi startar med att beskriva dels hur man installerar programpaketet WebLab, dels hur man administrerar detsamma. Exempelvis såsom att lägga till och ta bort användare. Vi kommer också ingående beskriva hur man bygger de HTML-dokument som utgör det interaktiva inslaget i dokumentationen. För att kunna tillgodogöra sig informationen i den sistnämnda delen av detta dokument bör man ha goda kunskaper i sidbeskrivningsspråket HTML och framför allt hur man bygger formulär i HTML. (Detta kan annars relativt lätt inhämtas via lämplig webportal.)

Förutsättningar

Systemet har utvecklats under UNIX i SUN–miljö och har inte heller testats inom någon annan datormiljö. För att kunna installera och använda Weblab krävs att följande förutsättningar är uppfyllda.

- Kompilator: Suns C^{++}–kompilator, CC. Programmet är endast kompilerat och testat i denna miljö.
- Web-server med stöd för CGI, Apache rekommenderas, men är ingalunda nödvändig. Mjukvaran till Apache finns tillgänglig gratis via http://www.apache.org/.
- WebLab-servern måste vara upplagd på samma (fysiska) filsystem som användarna. Servern måste ha läs- och skrivrättigheter till biblioteket ~/.weblab, vilken skall finnas i användarens rotbibliotek, för alla användare som skall utnyttja WebLab.
- Matlab måste vara installerad så att WebLab-servern (wlserver) kan komma åt att exekvera programmet, d v s wlserver och Matlab måste vara installerade i samma filsystem och wlserver måste ha exekveringsrättighet för Matlab.
- Bildkonverteringsprogrammen *pstopnm* och *ppmtogif* måste vara installerat så att wlserver kan exekvera dem.

Installation

1. Skapa en specifik användare för att exekvera och administrera WebLab.

   En enkel lösning på problemet med att CGI-programmen måste ha läs- och skrivrättigheter i användarnas $HOME/.weblab/ är att skapa en användare för att administrera och exekvera WebLab. För att kunna ge denna användare rättigheter till övriga användares .weblab-bibliotek är det lämpligt att skapa en grupp inom vilken denna specifika användare får läs- och skrivrättigheter.

   Alltså

   - Skapa gruppen weblab.
   - Skapa användaren wl-adm. Denna användare skall vara *superuser* för gruppen weblab av användare.

2. Installera Apache eller den webserver du föredrar. Detta görs lämpligen i biblioteket /home/weblab/apache.

   - Kompilera och installera webservern på lämpligt ställe i filsystemet.
   - Sätt *documentroot* till /home/wl-adm/htdocs. (Här skall /home/wl-adm uppfattas som den absoluta sökvägen till wl-adm:s hemmabibliotek.)
   - Sätt *scriptalias* till /cgi-bin/ och mappa den till biblioteket /home/wl-adm/cgi-bin. Webservern måste konfigureras så att man kommer åt cgi-skriptbiblioteket genom URL:en http://localhost-/cgi-bin/ för att WebLab skall fungera.

3. Installera WebLab.

   För att programvaran Weblab skall bete sig på ett planerat sätt krävs att vissa variabler erhåller en korrekt definition. Dessa variabler är här nedan beskrivna och var eventuell förändring skall ske.

   - Editera filen `src/wlserver/wlconst.h` genom att i denna definiera de omgivningsvariabler som behövs.
     - `WWW_ROOT` skall vara *documentroot* för webservern, därför krävs en definition enligt ex.vis följande
       `#define WWW_ROOT` "/home/wl-adm/htdocs".
     - `IMG_DIR` definierar var wlserver kommer att lagra bilder från beräkningsapplikationerna temporärt. Detta kan ex.vis ske genom
       `#define IMG_DIR` "/home/wl-adm/imgtmp".
     - `PSTOPNM` anger sökvägen till bildkonverteringsprogrammet *pstopnm*, ex.vis
       `#define PSTOPNM` "/usr/pd/netpbm/bin/pstopnm".
     - `PPMTOGIF` anger sökvägen till bildkonverteringsprogrammet *ppmtogif*, ex.vis
       `#define PPMTOGIF` "/usr/pd/netpbm/bin/ppmtogif".
   - I biblioteket src/wlserver/ skall följande göras
     - Ändra nedanstående variabler i filen Makefile
       * `CC =` den C^{++}–kompilator du tänker använda, ex.vis
         `CC =` CC

        * `DESTDIR =` wl-adms binära hembibliotek, ex.vis `DESTDIR` = /home/wl-adm/bin.
      - Exekvera *make*.
  - I biblioteket src/wlclient/
    - ändra följande variabler i filen Makefile.
      * `CC =` den C^{++}–kompilator du tänker använda, ex.vis `CC =` CC
      * `DESTDIR` = wl-adms hembibliotek, ex.vis `DESTDIR` = /home/wl-adm/cgi-bin.
    - Exekvera *make wlclient*.
    - Exekvera *make wlcreatedoc*.
    - Exekvera *make wlremove*.

4. Web-servern skall startas av användaren wl-adm. Därefter exekverar denne programmet *wlserver*.

Weblabadministration

Weblabadministratören är den som även ansvarar för att nedanstående viktiga saker åtgärdas.

1. Lägg till användar-ID (se under *Åtkomstkontroll* nedan) som du bland annat behöver för att testa med när du bygger uppgiftsdokumenten.

2. Bygg web-sidorna med uppgifter mm (se *Grunddokument för beräkningsuppgifter* nedan). Lägg hela strukturen med så att web-servern kan komma åt dokumenten, alltså under ~wl-adm/htdocs.

3. Aktivera web-serverns autenticeringssystem så att man måste logga på för att kunna komma åt uppgiftssidorna (se *Skydd av dokument* nedan). Utan att användaren har loggat på fungerar inte cgi-programmen som hanterar uppgifter och resultatdokument i WebLab.

4. Lägg till användar-ID:n för studenterna.

Åtkomstkontroll

Åtkomstkontroll är en nödvändighet för att WebLab inte skall ge upphov till säkerhetsproblem. Programmet WebLab tillåter användaren att exekvera program på wl-adm:s system. Men också måste systemet ha kunskap om användarens ID för att kunna avgöra var de filer som genereras skall hamna.

De program som närmast beskrivs är gjorda för att administrera användare av WebLab under en webserver med mjukvaran Apache och garanteras inte fungera med någon annan webserver. Dessa program kräver också att omgivningsvariablen `APACHE_ROOT` är definierad till det bibliotek vari Apache är installerat d v s (med C-shell)
setenv `APACHE_ROOT` /home/weblab/apache.

Addera ny användare (adduser)

Exekvera programmet *adduser* med användarens kontonamn som argument, ex.vis
`% adduser lab-01`
Programmet adduser kommer nu att identifiera användar-ID:t lab-01" i systemets användardatabas och addera detta, inkluderat med användarens lösenord, till de båda filerna `$APACHE_ROOT/conf/userlist` och `$APACHE_ROOT/conf/htaccess`.

Addera flera användare (addusers)

Programmet *addusers* tar ett argument. Alla kontonamn som innehåller texten i argumentet kommer att adderas som användare i weblab. Ex
`% addusers lab-`
Addusers kommer nu att inkludera alla användare som har lab-" i kontonamnet i filerna `$APACHE_ROOT/conf/userlist` och `$APACHE_ROOT/conf/htaccess` med de lösenord som de har i systemet.

Ta bort en användare (removeuser)

Hj̈lpfunktionen *removeuser* skall ha ett användar-ID som argument och eliminerar efter exekvering detta namn ur Apaches användardatabas. Removeuser svarar med att på skärmen skriva ut det användar-ID som kommer att försvinna och ställer också frågan om du vill fortsätta. För att undvika denna fråga kan du ge flaggan y" som argument, ex.vis
`% removeuser lab-01 -y`
Detta leder till att användar-ID:t lab-01" tas bort ur filerna `$APACHE_ROOT/conf/userlist` och `$APACHE_ROOT/conf/htaccess` utan att du behöver bekräfta att du vill fortsätta.

Ta bort flera användare (removeusers)

Programmet *removeusers* tar som argument den del av kontonamnet som är gemensamt för alla användar-ID:n som skall tas bort. Flaggan y" gör att du slipper bekräfta varje användar-ID som skall elimineras. Ex.vis
`% removeusers lab- -y`
Detta kommando leder till att alla konton som har lab-" i namnet kommer att tas bort ur filerna `$APACHE_ROOT/conf/userlist` och `$APACHE_ROOT/-conf/htaccess` utan att du behöver bekräfta några borttagningar.

Uppdatering av användarnas lösenord (updatepasswords)

Exekvera programmet *updatepasswords*. Detta uppdaterar filen `$APACHE_-ROOT/conf/htaccess` med de användar-IDn som finns i `$APACHE_ROOT/conf-/userlist` genom att läsa in de aktuella systemlösenorden.

Lägga till åtkomstkontroll (enableverify)

Programmet *enableverify* behöver ett bibliotek som argument. Programmet kopierar detta bibliotek och alla dess underbibliotek s att endast de som lagts till i Apaches användardatabas enligt ovan kan komma åt dokumenten i dessa bibliotek via web-servern. Ex.vis
`% enableverify /home/weblab/htdocs/exercises`
Detta kommando skyddar alla dokument som ligger i angivet bibliotek och alla dess underbibliotek från att andra än de som har fått ett ID inlagt enligt ovan kan komma åt dokumenten via web-servern.

Grunddokument för beräkningsuppgifter

Detta avsnitt beskriver hur man skriver de HTML-dokument som behövs för att få WebLab att fungera. Rotdokument som startar wlclient med uppgiftsdokumentet som argument och uppgiftsdokument som definierar ett antal variabler som krävs dels för att kunna kontakta WebLabs serverdel och dels för att kunna beräkna resultatet av uppgiften.

Rotdokument

Rotdokumentet är den www–sida som fungerar som rot för ett antal övningsuppgifter. Uppgiftssidorna länkas från rotdokumentet som anrop till CGI-programmet wlclient. Adressen till ett uppgiftsdokument anges som sista

delen av URL:en till wlclient, som t ex kan se ut så här: http://127.1.1.1/cgi-bin/wlclient/testuppgift. Wlclient letar då efter filen testuppgift.html biblioteket som definierats i `WWW_ROOT` i filen wlconst.h **(skall ändras till omgivningsvariabel)**. Anledningen till att .html inte anges i sökvägen är att programmet skapar fler filer med namnet testuppgift men med andra suffix.

Uppgiftsdokument

Uppgiftsdokumenten är html-dokument, uppbyggda kring formulär, med några tillägg. Formuläret måste innehålla vissa fördefinierade variabler som bland annat beskriver uppgiften, hur den skall beräknas och vilken server som skall kontaktas. Resultaten av uppgifterna genereras i både LaTeX och HTML och eftersom dessa språk emellanåt skiljer sig kraftigt finns i WebLab möjlighet att skriva text som bara syns i LaTeX respektive HTML-dokument. Förutom de speciella kraven på formuläret och LaTeX/HTML-utökningen är designen av uppgiftsdokumenten fri.

Formuläret

Det viktigaste i uppgiftsdokumentet är ur WebLabs synvinkel formuläret. Detta innehåller dels HTML-kod som låter användaren mata in värden för att lösa uppgiften, till exempel funktioner eller variabler, samt data som styr beteendet hos WebLab. All data skickas till WebLab via ett antal variabler. För att inte förväxla dem med de variabler som ingår i den funktion som skall beräknas kallar vi dem för formulärvariabler". Vilka dessa är, vad de används till och hur de används beskrivs nedan.
Formuläret inleds med *HTML-märket* `FORM`, som skall anges så här:
`<FORM ACTION=/cgi-bin/wlclient METHOD=POST>`
och avslutas med
`</FORM>`.
De formulärvariabler som används för att kontrollera beteendet hos WebLab och inte har direkt med själva uppgiften att göra anges lämpligen som osynliga inmatningskontroller. Eftersom användaren/studenten inte skall kunna ändra dessa formulärvariabler och antagligen bara skulle bli konfunderad av dem är det lämpligt att dölja dem. HTML-koden för detta ser generellt ut så här:
`<INPUT TYPE="HIDDEN" NAME=FORMULÄRVARIABELNAMN" VALUE="VÄRDE">`

Formulärvariablerna och deras värden

Ingen av dessa formulärvariabler får vara av typen SUBMIT eller RESET. I övrigt kan de i princip vara av vilken typ som helst bara värdet blir definierat korrekt. Vilka olika typer av inmatningskontroller som finns och hur de används finns beskrivet på många ställen på Internet. Välj ditt favoritställe!

SERVER

Denna variabel måste anges. Formulärvariablen `SERVER` anger IP-adressen på den maskin som WebLabs serverprogram kör på. Den är nödvändig för att klientprogrammet (CGI-programmet) skall kunna hitta och upprätta kontakt med serverprogrammet.

`INPUT-parametrar:`

`TYPE:`	Sätts till `HIDDEN` om du inte vill ge användaren möjlighet att ändra server själv.
`NAME:`	`SERVER`
`VALUE:`	Adressen till servern.

Exempel:

`<INPUT TYPE="HIDDEN" NAME=SERVER" VALUE=`"void.math.chalmers.se"`>`

EXERCISE

Även denna variabel måste anges. Formulärvariablen `EXERCISE` anger filnamnet (utan suffix) på aktuell uppgift. Den behövs för att klientprogrammet skall kunna namnge och hantera de filer där resultaten lagras. Den måste alltså stämma exakt överens med uppgiftsdokumentets filnamn, vilket bland annat innebär att stora och små bokstäver måste matcha varandra.

`INPUT-parametrar:`

`TYPE:`	`HIDDEN` eftersom man inte bör exponera denna variabel för användaren i onödan.
`NAME:`	`EXCERCISE`
`VALUE:`	Filnamn på HTML-filen utan det avslutande .html eller annat.

Exempel:

Uppgiftsdokumentet heter TestUppgift.html. `EXERCISE` skall då vara satt till TestUppgift.

`<INPUT TYPE="HIDDEN" NAME="EXERCISE" VALUE=TestUppgift">`

PROGRAM

Ytterligare en variabel som måste anges. Formulärvariablen `PROGRAM` anger namnet på den applikation som serverprogrammet skall skicka beräkningen till. Namnet har egentligen inget med filnamn eller dylikt att göra utan är bara en sträng som serverprogrammet har associerat till en applikation. För tillfället stöds endast matlab". Stora eller små bokstäver spelar ingen roll.

`INPUT-parametrar:`

`TYPE:`	`HIDDEN` eftersom uppgiften sannolikt bara går att beräkna i ett program är det meningslöst att ge användaren möjlighet att ändra detta.
`NAME:`	`PROGRAM`
`VALUE:`	Namnet på det program som WebLab har associerat med den applikation som skall beräkna uppgiften. Detta är inte nödvändigvis namnet på applikationen eller den exekverbara filen.

Exempel:

```
<INPUT TYPE="HIDDEN" NAME=PROGRAM" VALUE=MATLAB">
```

VAR

Alla variabler som används i `COMMAND` (nedan) måste anges. Denna formulärvariabel är lite speciell eftersom den består av `VAR` sammanslaget med ett variabelnamn. Om vi exempelvis skall tilldela variabeln `X` ett värde blir namnet på formulärvariabeln "`VARX`". Om applikationen som skall beräkna `COMMAND` skiljer på stora och små bokstäver i variabelnamn, så måste variabelnamnet här och i `COMMAND` överensstämma med avseende på stora och små bokstäver. De regler som beräkningsapplikationen har för variabler gäller alltså även här. Matlab tillåter exempelvis att en variabel tilldelas ett uttryck som innehåller andra variabler. Matlab kräver att de variabler som används i uttrycket redan är definierade, vilket i WebLab återspeglas i att de måste definieras tidigare på web-sidan.

`INPUT-parametrar:`

`TYPE:`	`TEXT` är antagligen den vanligaste konstruktionen som tillåter användaren att fritt välja värde.
`NAME:`	`VAR`<variabelnamn>
`VALUE:`	Om du vill ge ett initialt värde i textinmatningsfältet kan du sätta det här.

Exempel:

```
<INPUT TYPE=TEXT" NAME="VARX">
```

(`VALUE` som vi specificerat tidigare är den text som visas i inmatningsfältet.

Om man vill ge användaren ett initialt värde kan man specificera `VALUE`, men det är inte nödvändigt.)

COMMAND

Formulärvariabel `COMMAND` måste alltid anges. Variabeln anger den funktion eller det uttryck som skall skickas till applikationen som specificeras av `PROGRAM` ovan. `COMMAND` behöver inte definieras med en <`INPUT TYPE="HIDDEN"`> eftersom man kanske vill låta användaren skriva in funktionen som lösning på uppgiften. Om användaren skall skriva in funktionen själv måste man göra klart för denne att

Uttrycket eller funktionen i `COMMAND` måste anges enligt syntaxen för den applikation som skall beräkna "`COMMAND`". De variabler som ingår i `COMMAND` måste anges såsom beskrivs i stycket VAR (ovan). De behöver inte definieras före `COMMAND` eftersom WebLabservern skickar alla variabler till applikationen innan den skickar `COMMAND`.

`INPUT-parametrar:`

`TYPE:`	`TEXT` om du vill låta användaren skriva in funktionen här. Om du vill bestämma funktionen som skall beräknas är det lämpligt att definiera den med `TYPE` satt till `HIDDEN` och beskriva funktionen i brödtexten på web-sidan i stället.
`NAME:`	`COMMAND`
`VALUE:`	Den funktion som skall beräknas.

Exempel:

Vi vill beräkna värdet av uttrycket X + Y där användaren får ange värdena på X och Y genom inmatningsfält (Hur detta kodas i HTML, se exemplet för VAR nedan**???**).

<`INPUT TYPE="HIDDEN" NAME="COMMAND" VALUE="X + Y"`>

RESTYPE

En formulärvariabel som måste anges. `RESTYPE` specificerar vilken typ av resultat som övningen ger. WebLab stöder beräkningar som resulterar i text eller bilder. `RESTYPE` kan ha värdena `TEXT`, `IMG` eller `BOTH`. `TEXT` anger att resultatet av beräkningen skall presenteras i textform. `IMG` innebär att resultatet är en bild, exempelvis en plot av en funktion, och att den skall visas på resultatsidan. `BOTH` innebär att WebLab skall visa både text och bild som beräkningen resulterar i.

`INPUT-parametrar:`

TYPE:	HIDDEN eftersom uppgiften sannolikt är gjord för en viss typ av resultat skall användaren inte ha möjlighet att ändra detta.
NAME:	RESTYPE
VALUE:	TEXT, IMG eller BOTH

Exempel:

```
<INPUT TYPE="HIDDEN" NAME=RESTYPE" VALUE=TEXT">
```

Detta ger en resultatsida med den text som beräkningen resulterade i.

IMGWIDTH

Denna formulärvariabel har endast betydelse om RESTYPE är IMG eller BOTH. Formulärvariablen IMGWIDTH anger bredden, i pixlar, på bilden som beräkningen resulterar i. Är IMGWIDTH inte specificerad får bilden en bredd av 400 pixlar.

INPUT-parametrar:

TYPE:	Det kan vara meningsfullt att låta användaren välja storlek på bilden. Sätt i så fall TYPE till TEXT eller kanske en OPTION–konstruktion. Sätt TYPE till HIDDEN om du själv vill kontrollera storleken på bilden och inte vill ge användaren den möjligheten.
NAME:	IMGWIDTH
VALUE:	Ett heltal som anger er önskad bredd (i pixlar) på resultatbilden.

Exempel:

```
<INPUT TYPE="HIDDEN" NAME="IMGWIDTH" VALUE="250">
```

EXPL

Formulärvariabeln EXPL behöver ej definieras. DEn används för att ge användaren en möjligthet att låta teckna ner en förklaring till uppgiften. Lämpligen en TEXTAREA, där användaren kan skriva sin motiveringar och förklaringar till uppgiften. Den text användaren skriver här kommer även med i det slutliga resultatdokumentet.

INPUT-parametrar:

TYPE:	Lämpligen TEXTAREA eftersom användaren kan tänkas vilja skriva en del.
NAME:	EXPL

Exempel:

```
<TEXTAREA name="EXPL"></TEXTAREA>
```

SUBMIT

En formulärvariabel som måste finnas för att användaren skall kunna starta beräkningen.

```
INPUT-parametrar:
 TYPE:       SUBMIT
 VALUE:      Submit
```

Exempel:

```
<INPUT TYPE=SUBMIT" VALUE=Submit">
```

RESET

Denna formulärvariabel behövs bara om du tycker att användaren skall kunna eliminera alla fält i formuläret med en knapptryckning.

```
INPUT-parametrar:
 TYPE:       RESET
 VALUE:      Reset
```

Exempel:

```
<INPUT TYPE=RESET" VALUE=Reset">
```