```
PROGRAM TIL INVERSION AF LAPLACE TRANSFORM 120365 edda sveinsdottir
dog først
Procedurer til styring af CALCOMP-plotter.
begin
integer xxx,yyy,l111,pl111,plabc; real deltax,deltay;

procedure plotgraph(x0,x1,x,y,dx);
value x0,x1; real x0,x1,x,y,dx;
begin integer xx,yy,tr; boolean hop; boolean array B[1:l111];
tr:=drumplace; drumplace:=pl111;
from drum(B); drumplace:=tr;
x:=x0; xx:=x0/deltax-xxx; yy:=y/deltay-yyy;
gierproc(B[2],hop,xx,yy);
xxx:=xxx+xx; yyy:=yyy+yy;
for x:=x0 step dx until x1 do
begin
     xx:=x/deltax-xxx; yy:=y/deltay-yyy;
     gier(hop);
     xxx:=xxx+xx; yyy:=yyy+yy
end
end plotgraph;

procedure plotline(x0,y0,x1,y1);
value x0,y0,x1,y1; real x0,y0,x1,y1;
begin integer xx,yy,tr; boolean hop; boolean array B[1:l111];
tr:=drumplace; drumplace:=pl111;
from drum(B); drumplace:=tr;
xx:=x0/deltax-xxx; yy:=y0/deltay-yyy;
gierproc(B[2],hop,xx,yy);
xxx:=xxx+xx; yyy:=yyy+yy;
xx:=x1/deltax-xxx; yy:=y1/deltay-yyy;
gier(hop);
xxx:=xxx+xx; yyy:=yyy+yy
end plotline;


real procedure plotchar(t,x,y,h);
value t,x,y,h; integer t; real x,y,h;
begin integer xx,yy,n1,n2,hh,i,x1,y1; boolean hop,AA,flyt;
boolean array A[0:1],B[1:l111];
if t≥58 then goto exit; hh:=h×20;
xx:=drumplace; drumplace:=pl111;
from drum(B); drumplace:=plabc-(57-t)×2;
from drum(A); drumplace:=xx;
AA:=A[0];
x1:=split(AA,0,4,n1,5,9,y1):6;
y1:=(y1-x1×6)×hh; x1:=x1×hh;
xx:=x1+x/deltax-xxx; yy:=y1+y/deltay-yyy;
xxx:=x/deltax; yyy:=y/deltay;
if kbon then write(⦃-nddd⦄,writecr,xx,yy);
gierproc(B[2],hop,xx,yy);
n2:=(n1-8)×5; n1:=if n1≤7 then n1×5 else 35;
flyt:=false;
for i:=10 step 5 until n1,0 step 5 until n2 do
begin
     if i=0 then AA:=A[1];
     xx:=split(AA,i,i+4,yy):6;
     if yy=31 then
     begin
        flyt:=true;
     end
     else
     begin
     yy:=(yy-xx×6)×hh-y1;
```

```
     xx:=xx×hh-x1;
if kbon then write(⊰-nddd⊱,writecr,xx,yy);
if flyt then
begin
   flyt:=false;
gierproc(B[2],hop,xx,yy);
end else
     gier(hop);
     x1:=x1+xx; y1:=y1+yy
     end
     end i;
xxx:=xxx+x1; yyy:=yyy+y1;
exit:plotchar:=if t<58 then x+100×h×deltax else x end plotchar;
xxx:=yyy:=0; deltax:=deltay:=0.01;
pl111:=drumplace; gierdrum(111,l111);
plabc:=drumplace; gierdrum(113,0);

comment program af 120365 til inversion af Laplace transform = f(t)
arbejder med 3 slags data:
      1)  syntetiske data af form f(t)=Sum[N[i]×exp(-lambda[i]×t)]
som det selv genererer ved hjælp af real procedure f(x). I dette
tilfælde skrives o på skrivemaskinen inden kompileringen, for at
inkludere procedurekrop svarende til denne type data. pause-
signalet ignoreres.
      2)  syntetiske data, som ikke er en sum af eksponential
funktioner, men hvor f(t)=f(exp(x)) kan udtrykkes på explicit form.
I dette tilfælde skrives der ikke o på skrivemaskinen, men når
pause-signalet kommer, tastes procedurekroppen til real procedure
f(x) ind fra skrivemaskinen. Eks:
          f:= exp(-0.001×exp(x))/(exp(x)+0.009)↑2.
      3)  experimentelle data i form af et array af funktionsværdier
aflæst for ækvidistante værdier af lnt=x. I dette tilfælde tastes o
før compileringen. Ved at gøre n negativ (se nedenfor) vil maskinen
være parat til at indlæse en strimmel indeholdende data-array F[n,-n].
pause-signalet ignoreres.
DATASTRIMMEL I: maskinprocedure plot 111, 112, 113.
DATASTRIMMEL II:bestaar af tabel over den komplekse funktion,
Gamma(1+iu) hvor (u=0(0.1)10).
INPUT FRA SKRIVEMASKINE:
dx     skridtlængde ved x-integration  (f.eks. 0.25)
du     skridtlængde ved u-integration, skal være mult af 0.1
xmax   øvre grænse ved x-integrationen, tastes evt først ind , efter
       at plotteren har tegnet grafen af integranden ved x-integration
       skal være lige mult af dx
umax   øvre grænse ved u-integration, skal være lige mult af du
lamax  hhv øvre og nedre grænse for det interval for lamda, hvori
lamin  man ønsker at undersøge g(lamda)/lamda. lamax og lamin skal
       hele mult af 10-i
noofy  antal plottepunkter i det angivne lamda-interval
n, n>0 antal exponentialled i den syntetiske kurve f(t) af typen 1)
       med parametrene
N[i], lambda[i]   (i=1,...,n)
n, n<0 i dette tilfælde indlæses strimmel med rigtige data
F[n,-n] som er den experimentelle kurve aflæst med ækvidistancen dx
       i x-intervallet [n×dx,-n×dx];

begin integer n,m,noofy,Kplads;  boolean nykurve;
real dx,xmax,du,umax,lamax,lamin;  array Kc,Ks[0:100];

procedure plotcond(x,y,t,e1,e2,b1,b2);
real x,y,t,e1,e2; boolean b1,b2;
begin integer tr,xx,yy; boolean hop,s;
boolean array B[1:l111];
tr:= drumplace; drumplace:= pl111;
```

```
from drum(B); drumplace:= tr; s:= true;
for t:= e1,e2 while b1 do
if -, b2 then s:= true else begin
xx:= x/deltax - xxx; yy:= y/deltay - yyy;
if s then begin gierproc(B[2],hop,xx,yy);
xxx:= xxx+xx; yyy:= yyy+yy;
s:= false end else begin gier(hop);
xxx:= xxx+xx; yyy:= yyy+yy end
end
end plotcond;

for n:=0 step 1 until 100 do input(Kc[n], Ks[n]);  n:= 0;
DATA:
writecr;  writetext(⊰<dx,xmax,du,umax,lamax,lamin,noofy:
⊱);  outcr;  outtext(⊰<program 120365 - inversion af Laplace transform⊱);
dx:= typein;  xmax:= typein;  du:= typein;  umax:= typein;
lamax:= typein;  lamin:= typein;  noofy:= typein; m:= 1;
writecr;  writetext(⊰<m:⊱); m:= typein;
if m=0 then goto NÆSTYDERSTE BLOK;
if m<0 then begin n:= -m;  nykurve:= true; goto NÆSTYDERSTE BLOK end;
n:= m;  writecr;  writetext(⊰<N(i), lambda(i): ⊱);  nykurve:= true;

NÆSTYDERSTE BLOK:
begin integer i,j,k,uo,us,K,S,grænse;
real x,u,y,ymax,ymin,dy,gy,pi,Fcu,Fsu,Kcu,Ksu,nævner,lamda,lstep,gmax;
array N,lambda[1:n],g[0:noofy],F[-n:n];
real procedure f(x);  value x;  real x;
begin real fx;  fx:= 0;
for i:=1 step 1 until n do fx:= fx + N[i]×exp(-lambda[i]×exp(x));
f:= fx;
end f;*;
if m<0 then begin input(F);  outcr;  outtext(⊰<real data⊱);  goto XINT end;
if m>0 then for i:=1 step 1 until n do
  begin N[i]:= typein;  lambda[i]:= typein end;
outcr;  outcr;  outtext(⊰<N(i), lambda(i): ⊱);
for i:=1 step 1 until n do output(⊰-ndddd.ddd⊱,outcr,N[i],lambda[i]);
if m=0 then goto XINT;

begin real DX;
comment i denne blok bliver log y, hvor y=exp(x)×f(x), plottet mod x
(x=-10,-9,...,10) i skridt af o.25. x-enheden=0.25 tomme. y-dekaden
=1.5 tomme. Der er 5 dekader;
deltax:= 0.04;  deltay:= 0.02/3×2.302585;
deltax:= 0.02;  deltay:= 0.01/3×2.302585;
comment faktoren 2.302585 i deltay er lig ln 10. Dvs kan ved kald af
plotprocedurer benytte ln i stedet for log10;  writetext(⊰<
GØR CALCOMP KLAR t×f(t) plottes mod ln(t) - hvis dette uønsket tast 0: ⊱);
j:= typein;  if j=0 then goto XINT;
x:= -10;  DX:= 0.4; xxx:= yyy:= 0;
YAXE:
for j:=1,10,100,1000,10000 do
  for k:=2 step 1 until 10 do
  begin plotline(x,ln((k-1)×j),x,ln(k×j));
    plotline(x,ln(k×j),x+DX,ln(k×j));
  end k;  if x=10 then goto XAXE;
for k:=6 step -1 until 1 do
begin x:= -10-k×0.7;  j:= 1;
  for j:=j while x<-10.6 do
  begin x:= plotchar(j,x,ln(10↑(k-1)),0.175);  j:= 16 end;
end k;
x:= 10; DX:= -0.4;  goto YAXE;
XAXE:
deltay:= deltay/2.302585×1.5;  comment y-enhed = 1 tomme;
for j:=8 step -2 until -10 do
```

```
begin plotline(j+2,0,j,0);  plotline(j,0,j,-0.1);
  if j=0 then
  begin plotchar(16,-0.24,-0.4,0.15);  x:= plotchar(23,-1.5,-0.9,0.15);
    plotline(x+0.16,-0.8,x+0.44,-0.8);  plotline(x+0.16,-0.85,x+0.44,-0.85);
    x:= x+0.6;  for i:=35,37,19 do x:= plotchar(i,x,-0.9,0.15);
  end else
    if j>0 then plotchar(j,j-0.24,-0.4,0.15) else
      if j≠-10 then begin plotline(j-0.8,-0.325,j-0.55,-0.325);
        plotchar(-j,j-0.24,-0.4,0.15) end;
end;
  x:= -8;  x:= plotchar(53,x,7,0.15);  x:= plotchar(23,x,7.1,0.1);
  for j:=54,53 do x:= plotchar(j,x,7,0.15);  plotchar(23,x,7.1,0.1);
deltay:= deltay×2.302585/1.5; comment y-dekade = 1.5 tomme;
plotcond(x,x+ln(f(x)),x,-10,x+0.15,x≤10,x+ln(f(x))>ln(0.5));
writecr;  writetext(⊂<xmax (lige mult af dx): ⊃);  xmax:= typein;
end paa blok med plotning af exp(x)×f(x);

XINT:  grænse:= entier(xmax/dx);  j:= entier(umax×10);
begin real k1,k2,k3,k4,k5;  array Fc,Fs[0:j];  integer u1,du1,umax1;
begin real plus,minus,Fcos,Fsin;  array fc,fs[0:grænse];
comment i denne blok foretages x-integrationen ved hjælp af
Simpson. Dvs Fc(u) og Fs(u), hhv den reelle og imaginære del
af F(u) beregnes;
for k:=0 step 1 until grænse do
begin x:= k×dx;
   write(⊂nddd⊃,writecr,writetext(⊂<k,grænse I: ⊃),k,grænse);
  if m<0
    then begin plus:= exp(x)×F[k];  minus:= exp(-x)×F[-k] end
    else begin plus:= exp(x)×f(k);  minus:= exp(-x)×f(-k) end;
  fc[k]:= plus+minus;   fs[k]:= plus-minus;
end k;
du1:= du×10;  umax1:= umax×10;
   write(⊂nddd.dd⊃,writecr,writetext(⊂<du,du1: ⊃),du,du1);
for u1:=0 step du1 until umax1 do
begin u:= u1/10;
  writecr; writetext(⊂<kilroy 1⊃);
  Fcos:= fc[0]+4×fc[1]×cos(u×dx)+fc[grænse]×cos(u×xmax);
  Fsin:= 4×fs[1]×sin(u×dx)+fs[grænse]×sin(u×xmax);
  for k:=2 step  2 until grænse-2 do
  begin
   write(⊂nddd⊃,writecr,writetext(⊂<k,grænse II: ⊃),k,grænse);
    Fcos:= Fcos + 2×fc[k]×cos(u×k×dx) + 4×fc[k+1]×cos(u×(k+1)×dx);
    Fsin:= Fsin + 2×fs[k]×sin(u×k×dx) + 4×fs[k+1]×sin(u×(k+1)×dx);
  end k;
   write(⊂nddd⊃,writecr,writetext(⊂<u1,du1,umax1: ⊃),u1,du1,umax1);
  Fc[u1] := Fcos×dx/3;
  writecr; writetext(⊂<kilroy 2⊃);
  Fs[u1] := Fsin×dx/3;
  writecr; writetext(⊂<kilroy 3⊃);
end u;
end x-integration;

comment nu kendes Fc,Fs,Kc,Ks og u-integrationen kan
foretages (Simpson);
uo:= entier(umax/0.1);  us:= du/0.1;
write(⊂ndddd⊃,writecr,writetext(⊂<uo,us: ⊃),uo,us);
ymax:= -ln(lamin);  ymin:= -ln(lamax);
dy:= (ymax-ymin)/noofy;  i:= 0;
writecr; writetext(⊂<k1:⊃);
k1:= (Fc[0]×Kc[0]+Fs[0]×Ks[0])/(Kc[0]↑2+Ks[0]↑2);
writecr; writetext(⊂<k2:⊃);
k2:= 4×(Fc[us]×Kc[us]+Fs[us]×Ks[us])/(Kc[us]↑2+Ks[us]↑2);
writecr; writetext(⊂<k3:⊃);
k3:= 4×(Fs[us]×Kc[us]-Fc[us]×Ks[us])/(Kc[us]↑2+Ks[us]↑2);
```

```
writecr; writetext(⦃<k4:⦄);
k4:= (Fc[uo]×Kc[uo]+Fs[uo]×Ks[uo])/(Kc[uo]↑2+Ks[uo]↑2);
writecr; writetext(⦃<k5:⦄);
k5:= (Fs[uo]×Kc[uo]-Fc[uo]×Ks[uo])/(Kc[uo]↑2+Ks[uo]↑2);
for y:=ymin step dy until ymax do
begin
  gy:= k1+k2×cos(y×du)+k3×sin(y×du)+k4×cos(y×umax)+k5×sin(y×umax);
  for k:=2×us step 2×us until uo-2 do
  begin K:= k;   S:= 2;
OM: Fcu:= Fc[K];  Fsu:= Fs[K];  Kcu:= Kc[K];  Ksu:= Ks[K];
    nævner:= Kcu↑2 + Ksu↑2;  u:= K×0.1;
    gy:= gy+S×(Fcu×Kcu+Fsu×Ksu)/nævner×cos(y×u)
           +S×(Fsu×Kcu-Fcu×Ksu)/nævner×sin(y×u);
    K:=K + us;  S:= 4;  if K= k+us then goto OM;
  end k;
  g[i]:= gy;  i:= i+1;
end y og u-integration;
end integrationer;

comment nu udskrives og plottes resultaterne;
outcr;  outcr;  outtext(⦃<dx,xmax,du,umax,lamax,lamin,noofy: ⦄);
outcr;  output(⦃-nddd.dddd⦄, dx,xmax,du,umax,lamax,lamin,noofy);
outcr;  outcr;  outtext(⦃<y,lambda,g(exp(-y)): ⦄);
for i:=0 step 1 until noofy do
begin y:= ymin + i×dy;  lamda:= exp(-y);  outcr;
  output(⦃-ndd.dd⦄, y);  output(⦃-nddd.dddd⦄, lamda);
  output(⦃-ndd.d10+d⦄, outsp(4), g[i]);
end i;
gmax:= g[0];
for i:= 1 step 1 until noofy do if gmax<g[i] then gmax:= g[i];
outcr;  outcr;  outtext(⦃<100 = ⦄);  output(⦃-ndd.d10+d⦄, gmax);
for j:=0 step 1 until noofy do g[j]:= g[j]/gmax×100;
writecr;  writetext(⦃<gør Calcomp klar til ny plotning⦄);  j:= typein;
deltax:= 0.01×(ymax-ymin)/5;
deltay:= 0.01×100/5; comment y-akses længde = 9 tommer;
xxx:= ymin;  yyy:= 0;
for i:= 10 step 10 until 140 do
begin plotline(ymin,i-10,ymin,i);  plotline(ymin,i,ymin+10×deltax,i);
  if i=50 then begin x:= ymin-36×deltax;
  x:= plotchar(5,x,50,0.175);  plotchar(16,x,50,0.175) end;
  if i=100 then
  begin x:= ymin-54×deltax;  x:= plotchar(1,x,100,0.175);
  x:= plotchar(16,x,100,0.175);  plotchar(16,x,100,0.175);
  end i=100;
end i;
for i:= -10 step -10 until -40 do begin
  plotline(ymin,i+10,ymin,i);  plotline(ymin,i,ymin+10×deltax,i) end;
plotline(ymin,0,entier(ymin+1),0);
plotline(entier(ymin+1),0,entier(ymin+1),-2);  i:= 1;
for i:=i+1 while entier(ymin+i)≤ymax do
begin y:= entier(ymin+i);  plotline(y-1,0,y,0);  plotline(y,0,y,-2) end;
plotline(entier(ymin+i-1),0,ymax,0);  plotline(ymax,-40,ymax-10×deltax,-40);
for i:=-30 step 10 until 140 do
begin plotline(ymax,i-10,ymax,i);  plotline(ymax,i,ymax-10×deltax,i) end;
for i:=1 step 1 until noofy do plotline(ymin+(i-1)×dy,g[i-1],ymin+i×dy,g[i]);
i:=1;  lstep:= 0.1;
for i:=i×10 while entier(i×lamax)=0 do lstep:= 1/i/10;
i:= 0;  y:= ymin;  lamda:= lamax;
for i:= i+1 while y<ymax do
begin for k:= 0 step 1 until 4 do
  if abs(lamda-10↑(-k))<10-7 then begin
    if k=0 then plotchar(1,ymin-8×deltax,-49,0.15)
    else begin x:= y-k×8×deltax;  x:= plotchar(16,x,-49,0.03);
    for i:=2 step 1 until k do x:= plotchar(16,x,-49,0.15);
```

```
    plotchar(1,x,-49,0.15) end
  end;
  lamda:= lamda - lstep;
  if lamda<10-7 then begin lamda:= lamda+lstep-lstep/10;  lstep:= lstep/10 end;
  y:= -ln(lamda);
  if y>ymax then begin plotline(-ln(lamda+lstep),-40,ymax,-40);  goto l1 end;
  plotline(-ln(lamda+lstep),-40,y,-40);  plotline(y,-40,y,-42);
l1:  end lamdaakse 1;
i:= 1;  lstep:= 0.1;  for i:=i×10 while entier(i×lamax)=0 do lstep:=1/i/10;
i:= 0;  y:= ymin;  lamda:= lamax;
for i:=i+1 while y<ymax do
begin lamda:= lamda - lstep;  if lamda<10-7 then
  begin lamda:= lamda+lstep-lstep/10;  lstep:= lstep/10 end;  y:= -ln(lamda);
  if y>ymax then begin plotline(-ln(lamda+lstep),140,ymax,140);  goto l2 end;
  plotline(-ln(lamda+lstep),140,y,140);  plotline(y,140,y,138);
l2:  end lamdaakse 2;
end næstyderste blok;
nykurve:= false;  goto DATA;
end yderste blok
end som skal svare til plotprograms begin;
```