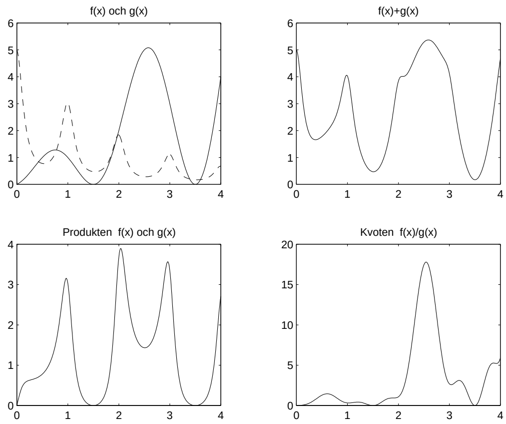
Matlab-övningar, MÖ

1. **Kurvritning:** Rita kurvor för funktionerna

$$f(x) = x(1 + \sin \pi x),$$

$$g(x) = \frac{5e^{-x/2}}{3 - 2\cos 2\pi x}$$

i intervallet $0 \leq x \leq 4$. Pröva dig fram till lämpligt steg.
Rita i en ny bildruta (med `subplot`) kurvan $f(x)+g(x)$. Fortsätt med produkten $f(x)\cdot g(x)$ i bildruta 3 och kvoten $f(x)/g(x)$ i ruta 4.

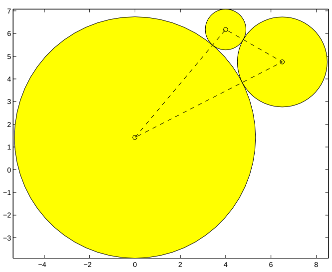


2. **Ytterligare en funktionskurva:** $y = \frac{10}{\sqrt{1+x^2}} + \frac{e^{x/2}}{\sqrt{2+\sin \pi x}} + \frac{4}{x-5}$
Rita kurvan i intervallet $-2 \leq x \leq 4$ med ett lagom litet steg så att kurvan inte ser kantig ut. Kolla riktigheten med kurvan i facit!

3. **Triangel:** Hörnen i en triangel har x-värdena 0, 4, 6.5 och y-värdena 1.42, 6.18, 4.75. Markera hörnen och rita upp triangeln. Beräkna sidorna, omkretsen och arean. Skriv koden MATLAB-mässigt utan for-slingor.

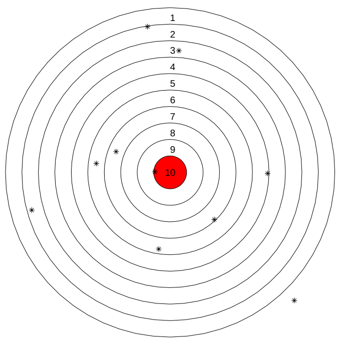
4. **Cirkel:** Cirklar ritas bäst på parameterform, $x = x_c + R\cos\varphi$, $y = y_c + R\sin\varphi$, där vinkeln φ går från 0 till 2π. Rita en cirkel med radien 3 och centrum i (0, 1.42). Använd steget $2\pi/60$. Markera mittpunkten. Glöm inte `axis equal` efter plot-satsen, annars blir cirkeln en oval.

5. **Tre tangerande cirklar:** Låt cirklarna ha centrum i triangelhörnen i uppgift 3. Bestäm radierna så att cirklarna tangerar varandra. Rita triangeln samt de tre cirklarna.
Sambanden $r_1 + r_2 = s_1$, $r_2 + r_3 = s_2$, $r_3 + r_1 = s_3$ mellan radier och triangelsidor skrivs i MATLAB enklast på matris-vektor-form (bara ettor och nollor i matrisen `A`).
Lös med `r=A\s`.



6. **Pilkastning:** Rita tio cirklar med radierna 1, 2, ..., 10. Fyll den innersta med rött. Lägg in poängmarkering på piltavlan med `text`-funktionen. Slumpa 10 pilkast normalfördelat (till exempel med standardavvikelsen 5 i x-led och 4 i y-led) enligt:

```
for pil=1:10, plot(5*randn, 4*randn,'*'), pause(0.7), end
```



Pause-satsen ger 0.7 sekunders fördröjning mellan pilkasten. Blinkningen som syns vid varje pilkast beror på att gammal plot-bild raderas och ersätts av uppdaterad bild. Man kan slippa flimret genom att före for-slingan skriva

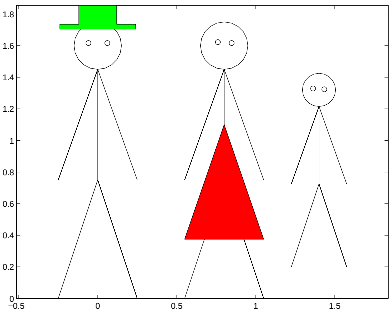
```
set(gcf,'DoubleBuffer','on')
```

Egenskapen `DoubleBuffer` styr hur grafiken ska uppdateras hos grafikobjektet figure i MATLAB.

Beräkna poängen för varje kast, summera och skriv ut totalpoängen. Exekvera flera gånger. Simulera sedan en sämre pilkastare genom att öka standardavvikelsen (gör `help randn` för mer info).

7. **Stjärnhimmel:** Femuddsstjärnor kan man ju rita utan att lyfta pennan. Rita en sådan först för hand och sedan i MATLAB. Låt en udd ligga i origo och resten på lämpliga koordinatvärden (i facit för femudden används `x=[0 5 0 3.5 3 0]`, `y=[0 2.8 3.3 0 5 0]`).
Rita också en stjärna som är förskjuten fyra enheter i både x-led och y-led. Använd `fill` istället för `plot` på den. Gör sedan en stjärnhimmel genom att slumpa ut femton ofyllda och femton fyllda stjärnor likformigt fördelat (med funktionen `rand`) i området $0 \le x \le 60$ och $0 \le y \le 45$.

8. **Streckgubbefamilj:** Ria en streckgubbe med runt huvud på ett rutat papper och fundera ut hur gubben med hjälp av vektorer ska hamna i grafikfönstret i MATLAB. Pröva!
Placera en likadan gubbe bredvid och en tredje fast i mindre skala vid sidan om. I facit finns den här pappan med hatt, mamman med kjol och ett litet barn. Ögonen klickas in. Studera och exekvera `gubbar.m` som finns i kurskatalogen.



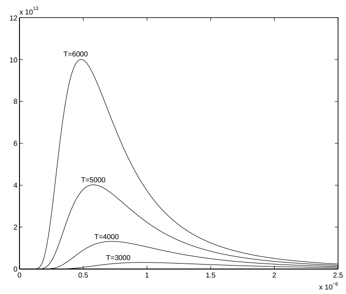
9. **Triangel genom klickning:** Modifiera uppgifterna 3 och 5 genom inklickning av tre önskade hörnpunkter till en triangel. Börja med ett `axis`-kommando med lämpliga värden för x- och y-axlarna, till exempel: `clear, clf, axis([0 10 0 8]), hold on`
Klickning av tre punkter kan göras med `[x,y]=ginput(3); plot(x,y,'o')`
Nackdelen är att punkterna inte markeras förrän alla tre matats in. Pröva!
Det är bättre att klicka in en punkt i taget, markera den och uppdatera x-vektorn och y-vektorn:

```
x=[]; y=[];
for nr=1:3, [xp,yp]=ginput(1); plot(xp,yp,'*'), x=[x; xp]; y=[y; yp]; end
```

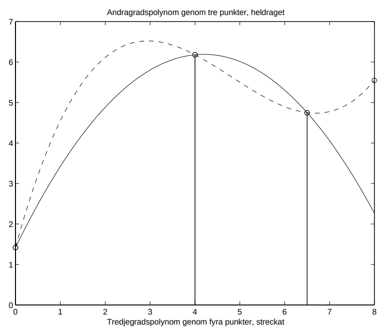
10. **Svartkroppsstrålningskurvor:** Rita en skara av intensitetskurvor för den våglängds- och temperaturberoende funktionen $w(\lambda, T)$ som gäller svartkroppsstrålning enligt Planck:

$$w(\lambda, T) = \frac{\alpha}{\lambda^5 \left(e^{\beta/(\lambda T)} - 1\right)}$$

där $\alpha = 3.7415 \cdot 10^{-16}$ och $\beta = 0.014388$. Kurvor önskas för temperaturvärdena $T = 3000, 4000, 5000, 6000$ K i det intressanta våglängdsområdet $0 < \lambda \le 250 \cdot 10^{-8}$.
Använd `gtext` för att skriva '`T=3000`' vid första kurvan och motsvarande vid de övriga kurvorna.



11. **Polynom genom givna punkter:** Genom fyra givna punkter går ett entydigt bestämt tredjegradspolynom som kan skrivas $P(x) = c_1 + c_2 x + c_3 x^2 + c_4 x^3$. Det leder till ett ekvationssystem $\mathbf{Ac} = \mathbf{y}$ för bestämning av c_1, c_2, c_3, c_4. Matrisen kan skrivas `A=[ones(size(x)) x x.^2 x.^3]` där `x` är en kolumnvektor med x-värdena.
a) Till att börja med har vi bara tillgång till tre punkter, nämligen triangelhörnen i uppgift 3. Beräkna och rita den entydigt bestämda parabel (andragradskurva) som går genom punkterna.

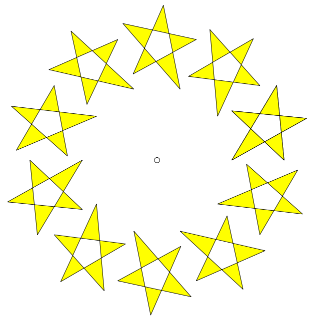


b) Lägg till punkten (8, 5.55) och beräkna och rita tredjegradspolynomet som passerar genom de fyra punkterna. Punkterna ska naturligtvis också markeras i figuren, pröva gärna `stem(x,y)` som ritar en stolpe från x-axeln upp till varje punkt som markeras med ring.

12. **Cirkel genom tre punkter:** Formeln $c_1 + c_2x + c_3y = x^2 + y^2$ är ett bekvämt sätt att ange cirkelns ekvation då man vill hitta den entydigt bestämda cirkel som passerar genom tre givna punkter, låt oss ta triangelhörnen i uppgift 3. Skriv systemet i matrisform och lös med MATLAB. Cirkelns mittpunkt är $(c_2/2, c_3/2)$. Radien kan uttryckas i c_1, c_2 och c_3; visa det och beräkna radien. Rita upp punkterna och cirkeln.

13. **Vridna figurer:** Vridningsmatrisen `S=[cos(v) -sin(v); sin(v) cos(v)]` är användbar. Här startar vi med femudden i uppgift 7, flyttar den fem enheter åt höger och roterar allt kring origo. Så skapas stjärnbilden här (`femuddsnurr.m`).

```
x=5+[0 5 0 3.5 3 0]; y=[0 2.8 3.3 0 5 0];
plot(0,0,'o', x,y), axis equal, hold on
n=10; v=2*pi/n;
S=[cos(v) -sin(v); sin(v) cos(v)];
for k=1:n
  P=S*[x;y]; x=P(1,:); y=P(2,:); fill(x,y,'y')
end, axis off
```



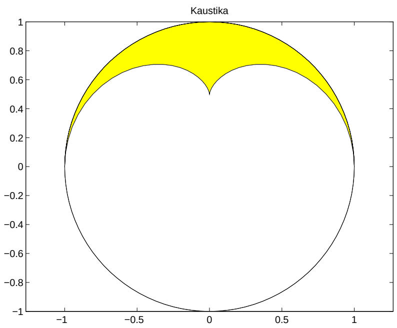
Satsen `y=P(2,:)` betyder att vektorn `y` blir lika med rad 2 av matrisen `P`.
Modifiera koden så att en fylld stor bokstav (gärna också en fylld tvåa) vrids i n steg. I facit visas detta för D2 (i `d2.m`).

14. **Nya vektorer ur gamla:** Givet är kolumnvektorn **y** med komponenterna $y_1, y_2, \ldots, y_{10}$, till exempel siffrorna i ditt personnummer. Bilda i MATLAB en ny vektor **dy** som innehåller differenserna $y_{j+1} - y_j$. Bilda utan for-slinga en vektor med nio komponenter som är medelvärdena $(y_j + y_{j+1})/2$.
Bilda också – utan att använda for-slinga – en vektor **b** med tio komponenter enligt följande:
$$b_1 = dy_1,\;\; b_j = dy_{j-1} + dy_j,\; j = 2, 3, \ldots, 9,\;\; b_{10} = dy_9.$$
Vektorn **b** med dessa egenskaper är viktig bland annat vid interpolation med splines.

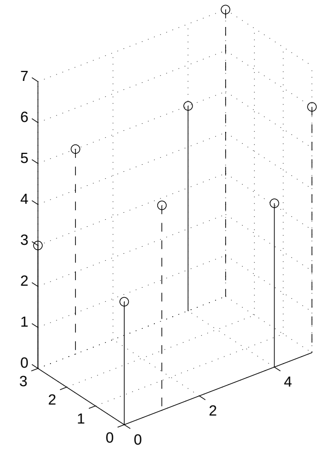
15. **Tridiagonala ekvationssystem med `tridia`:** Kurskatalogfilen `tridia.m` (*kopiera den!*) innehåller en funktion för effektiv lagring och effektiv lösning av tridiagonala system.
Betrakta en tridiagonal matris med alla diagonalelement lika med fyra och med ettor i super- och subdiagonalen. Låt högerledet bestå av vektorn `b=1:n`.
Lös detta ekvationssystem med `tridia` (utan att skriva ut resultatet) då antalet obekanta, n, är stort: $n = 200$, 400, 800 och 1600. Beräkna och skriv ut tidsåtgången (med hjälp av `tic` och `toc`) och notera hur den växer med n.
Som jämförelse ska du lösa samma ekvationssystem med den för bandmatriser oekonomiska metoden `x=A\b` och studera tidsåtgången. Matrisen `A` kan skapas i en sats:
`A=4*eye(n)+diag(sup,1)+diag(sub,-1)`
där `sup` och `sub` är vektorerna för super- och subdiagonalerna.

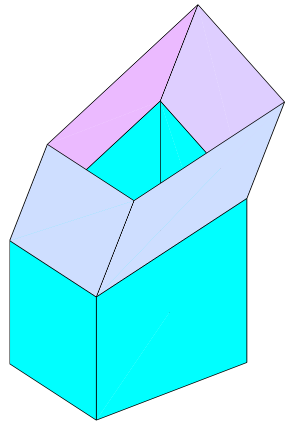
16. **Kaustika inuti en cirkel:** Om parallellt ljus faller in mot en sfärisk spegel bildar de reflekterande strålarna en yta, kaustika, vars skärning med ett plan ger en karakteristisk kurva. Den framträder tydligt om man snedbelyser en ring som ligger på ett plant underlag (*kaustika* i Nationalencyklopedin). Låna en förlovningsring om du inte har en egen och betrakta kaustikan. Man kan också se den i ett halvfyllt mjölkglas.

Kaustikan inuti enhetscirkeln då ljuset faller in parallellt med y-axeln (ljuskälla vid $y = -\infty$) bestäms av det trigonometriska uttrycket
$$x = \cos^3\varphi,\;\; y = \frac{3}{2}\sin\varphi - \sin^3\varphi,\;\; 0 \le \varphi \le \frac{\pi}{2},$$
vilket ger halva kurvan. Den andra kurvdelen fås genom spegling i y-axeln. Rita upp enhetscirkeln och kaustikan. Fyll den gärna med gult.

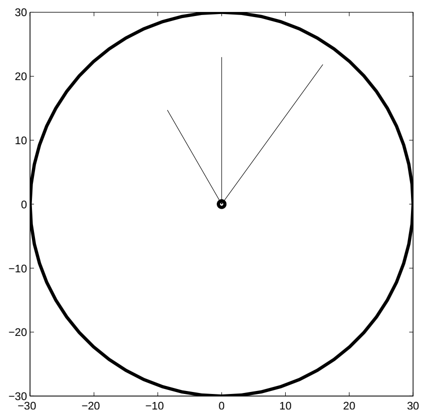


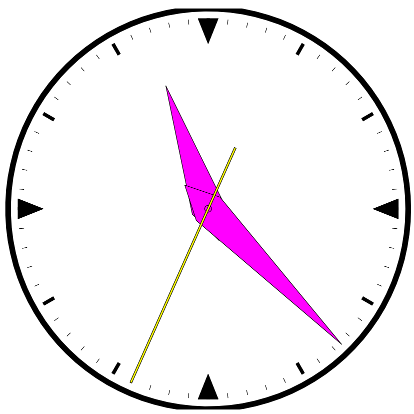
17. **Sned låda, 3D-ritning:** Den sneda lådan i högra bilden är konstruerad av stolparna som visas i vänstra figuren och plottas med `stem3(x,y,z)`. Rektangeln i botten är fyra enheter i x-led och tre i y-led. De heldragna stolphöjderna är `[3 5 5 3]`. De streckade stolparna har förflyttats en enhet i x-led och höjderna har ökats med två enheter. Rita figuren med `surf`-kommandot; se facit.





18. **Klocka som visar rätt tid:** Konstruera en enkel klocka med urtavlan i form av en cirkel och försedd med färggranna sekund-, minut- och timvisare. Det behöver inte finnas siffror på urtavlan. Rita om visarna vid varje hel sekund. utnyttja `drawnow` efter plot-satsen. Använd MATLAB-funktionen `clock` som returnerar en radvektor med sex element: `[year month day hour minute seconds]`, där de fem första är heltal och det sista, `seconds`, har några få decimaler. Det avhjälps med `fix(clock)`.
Exempel: `fix(clock)` returnerar `[2008 8 31 11 0 6]` om klockan råkar vara 0 minuter och 6 sekunder över 11 (vänstra figuren) den 31 augusti 2008.
Gör som i uppgift 6 för att undvika flimmer då bilden uppdateras.
Det är extra bra om timvisaren även är beroende av antalet gångna minuter, så att den inte står på nio då klockan är en sekund i tio, och då plötsligt hoppar till tio.





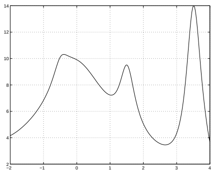
Det går förstås att lägga ner mer arbete på uppgiften och åstadkomma klockan i högra figuren. Den finns inte i facit!

FACIT till MÖ-uppgifterna

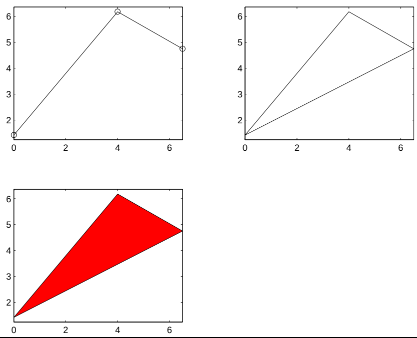
Obs! Lösningar som inleds med ett programnamn finns i kurskatalogen.

```
% MÖ 1.
  x=0:0.01:4;
  f=x.*(1+sin(pi*x));
  g=5*exp(-x/2)./(3-2*cos(2*pi*x));
  subplot(2,2,1), plot(x,f, x,g,'--'), title('f(x) och g(x)')
  subplot(2,2,2), plot(x,f+g), title('f(x)+g(x)')
  subplot(2,2,3), plot(x,f.*g), title('Produkten f(x)g(x)')
  subplot(2,2,4), plot(x,f./g), title('Kvoten f(x)/g(x)')
```

```
% MÖ 2.
  x=-2:0.02:4;      % eller med linspace (gör help linspace)
  x=linspace(-2,4,300);
  y=10./sqrt(1+x.^2)+exp(x/2)./(sqrt(2)+sin(pi*x))+4./(x-5);
  plot(x,y), grid                % grid gör rutnät
```



```
% MÖ 3.
  x= [0    4   6.5]'                      % kolumnvektor med x-koordinater
  y=[1.42 6.18 4.75]'                     % kolumnvektor med y-koordinater
  subplot(2,2,1), plot(x,y,'o', x,y), axis equal
  xx=[x; x(1)]; yy=[y; y(1)];             % lägg till första punkten sist
  subplot(2,2,2), plot(xx,yy),   axis equal
  subplot(2,2,3), fill(x,y,'r'), axis equal
% Beräkna sidorna, omkretsen och arean
  dx=diff(xx),  dy=diff(yy)               % differenser i x-led och y-led
  s=sqrt(dx.^2+dy.^2)                     % Pythagoras sats
  omkrets=sum(s)
  p=omkrets/2; area=sqrt(p*prod(p-s))     % arean med Herons formel (se Ex.saml ex 8.4)
```



```
% MÖ 4.
  n=60;  dfi=2*pi/n;
  fi=0:dfi:2*pi;
  xc=0; yc=1.42; R=3;
  plot(xc+R*cos(fi),yc+R*sin(fi), xc,yc,'o'), axis equal
```

```
% MÖ 5.
% trecirk, cirklar som tangerar varandra
  format compact              % Ger kompakt radutskrift (inga tomma rader)
  x=[0 4 6.5]';  y=[1.42 6.18 4.75]';
  plot(x,y,'o'), axis equal, hold on
  xx=[x; x(1)]; yy=[y; y(1)];
  dx=diff(xx);  dy=diff(yy);  s=sqrt(dx.^2+dy.^2);
  A=[1 1 0; 0 1 1; 1 0 1];  r=A\s
  fi=0:2*pi/60:2*pi;
  for j=1:3, fill(x(j)+r(j)*cos(fi) ,y(j)+r(j)*sin(fi),'y'),  end
  plot(x,y,'o', xx,yy,'--')
```

```
% MÖ 6.
% piltavla, koncentriska cirklar
  clear, clf
  fi=0:2*pi/60:2*pi; cc=cos(fi); ss=sin(fi);
  fill(cc, ss,'r'), hold on, axis equal, axis off
  text(-0.3, 0, '10')
  for R=2:10, plot(R*cc, R*ss), text(0,R-0.6,int2str(11-R)), end
  totpoeng=0;
  for pil=1:10
    x=5*randn; y=4*randn;
    s=sqrt(x^2+y^2);
    if s>10, poeng=0, else poeng=10-fix(s), end
    totpoeng=totpoeng+poeng;
    plot(x,y,'*'), pause(0.7)
  end, totpoeng
```

```
% MÖ 7.
% femudd, femuddiga stjärnor
  clear, clf
  x=[0 5 0 3.5 3 0]; y=[0 2.8 3.3 0 5 0]; plot(x,y), hold on
  fill(4+x,4+y,'r'), axis equal, pause(1)
  for i=1:15, plot(55*rand+x, 40*rand+y), fill(55*rand+x, 40*rand+y,'r'), end
```

```
% MÖ 8.
% gubbar, familjebild
  clear, clf
  a=0.25;  b=0.75;  h=1.45;
% Ritas: höger arm, vänster arm, kroppen, höger ben, vänster ben
  xg=[a 0 -a 0 0 a 0 -a];
  zg=[b h  b h b 0 b  0];
  r=0.15; v=0:2*pi/20:2*pi;
  xhuv=r*cos(v); zhuv=h+r+r*sin(v);
  plot(xg,zg, xhuv,zhuv), hold on
  Lm=0.8; plot(Lm+xg,zg, Lm+xhuv,zhuv)                % mamma på avståndet Lm
  xkjol=[0 a -a 0]; zkjol=[(b+h)/2 b/2 b/2 (b+h)/2];  % mammas kjol
  fill(Lm+xkjol,zkjol,'r')
  s=0.7; Lx=1.4; Lz=0.2;                    % barnets förminskning och placering
  plot(Lx+s*xg,Lz+s*zg, Lx+s*xhuv,Lz+s*zhuv)
  xhatt=0.8*r*[2 2 1 1 -1 -1 -2 -2 2];                % pappas hatt
  q=0.2*r; zhatt=h+1.7*r+[0 q q r r q q 0 0];
  fill(xhatt,zhatt,'g'), axis equal
  disp('Klicka in ögon på alla tre!')
  for nr=1:6, [x,y]=ginput(1); plot(x,y,'o'), end
```

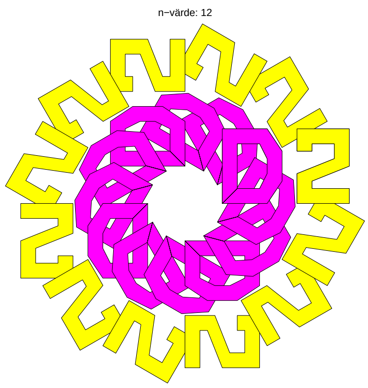
```
% MÖ 9.
% triangklick, triangelns hörn med klickning
  clear, clf, axis([0 10 0 8]), hold on
  x=[]; y=[];  disp('Klicka tre punkter')
  for i=1:3
   [xp,yp]=ginput(1); plot(xp,yp,'*'),
   x=[x; xp]; y=[y; yp];
  end
  xt=[x; x(1)]; yt=[y; y(1)];  plot(xt,yt,'m'), axis equal
  dx=diff(xt);  dy=diff(yt);  s=sqrt(dx.^2+dy.^2);
% De tre tangerande cirklarna ska beräknas och ritas
  A=[1 1 0; 0 1 1; 1 0 1];  r=A\s
  fi=0:2*pi/60:2*pi;
  for j=1:3, plot(x(j)+r(j)*cos(fi), y(j)+r(j)*sin(fi)), end
```

```
% MÖ 10.
% planck, svartkroppsstrålning enligt Planck
  clear, clf
  alfa=3.7415e-16; beta=0.014388;  ds=1e-8; s=ds*(1:250)';
  for T=3000:1000:6000
    p=beta./(s*T); w=alfa./(s.^5.*(exp(p)-1)); plot(s,w), hold on
  end
  disp('Klicka in textplaceringarna!')
  for i=3:6, gtext(['T=' int2str(i*1000)]), end
```

```
% MÖ 11.
% polynom,  polynom genom givna punkter
  clear, clf
  x=[0  4  6.5]';  y=[1.42  6.18  4.75]';
  A=[ones(size(x)) x x.^2], c=A\y
  X=0:0.1:8;  P2=c(1)+c(2)*X+c(3)*X.^2;
  stem(x,y), hold on, plot(X,P2,'g')
  title('Andragradspolynom genom tre punkter, heldraget'), pause(1)
% Lägg till en punkt:
  x=[x; 8];  y=[y; 5.55]; stem(x,y)
  A=[ones(size(x)) x x.^2 x.^3], cc=A\y
  P3=cc(1)+cc(2)*X+cc(3)*X.^2+cc(4)*X.^3;
  plot(X,P3,'r--')
  xlabel('Tredjegradspolynom genom fyra punkter, streckat')
```

```
% MÖ 12.
% Cirkel genom tre punkter
  clear, clf
  x=[0  4  6.5]';  y=[1.42  6.18  4.75]';
  A=[ones(3,1) x y],  b=x.^2+y.^2,  c=A\b
  xc=c(2)/2,  yc=c(3)/2,  R=sqrt(c(1)+xc^2+yc^2)
  fi=0:2*pi/60:2*pi;  xx=xc+R*cos(fi);  yy=yc+R*sin(fi);
  plot(x,y,'o', xc,yc,'x', xx,yy), axis equal
```

```
% MÖ 13.
% d2, snurra figuren D2
  clear, clf
% Koordinaterna för bokstaven D och siffran 2
  xD=[0  0  6 8 8 6 0 2 5 6 6 5 2 2]+5;
  yD=[0 10 10 7 3 0 0 2 2 4 6 8 8 2];
  x2=[0  0  7 7 2 2 7 7 0 0 5 5 2 2 0]+15;
  y2=[7 10 10 5 3 2 2 0 0 4 6 8 8 7 7];
  plot(xD,yD, x2,y2), axis equal, hold on
  n=input('n-värde: ');  v=2*pi/n;
  A=[cos(v) -sin(v); sin(v) cos(v)];
  for k=1:n
    D=A*[xD; yD];    xD=D(1,:);    yD=D(2,:);
    two=A*[x2; y2];  x2=two(1,:);  y2=two(2,:);
    fill(xD,yD,'m'), fill(x2,y2,'y')
  end
  title(['n-värde: ', int2str(n)]), axis off
```



```
% MÖ 14.
% vecnygam,  nya vektorer ur gamla
  y=[7 2 0 6 2 3 0 0 6 7];
  dy=diff(y)                           % differensvektorn
  z=(y(1:9)+y(2:10))/2                 % medelvärdesvektorn
  b=[0 dy(1:9)]+[dy(1:9) 0]            % b-vektorn
  balt=[dy(1) dy(1:8)+dy(2:9) dy(9)]   % b-vektorn alternativ
```

```
% MÖ 15.
% trisys, Hur effektiv är tridia?  (Kopiera först tridia.m)
  nvec=[200 400 800 1600]; tritid=[]; gautid=[];
  for sys=1:4
    n=nvec(sys), dia=4*ones(n,1); sup=ones(n-1,1); sub=sup;
    b=(1:n)';
    tic, x=tridia(dia,sup,sub,b); tid=toc, tritid=[tritid tid];
    A=4*eye(n)+diag(sup,1)+diag(sub,-1);
    tic, x=A\b; gtid=toc, gautid=[gautid gtid];
  end
  nvec, tritid, gautid
```

```
% MÖ 16.
% kaustika, optiskt fenomen i en ring
  dfi=2*pi/100;  fi=(0:dfi:2*pi)';
  plot(cos(fi),sin(fi)), hold on
  v=(0:dfi:pi/2)';
  xk=cos(v).^3;
  yk=3/2*sin(v)-sin(v).^3;
  plot(xk,yk, -xk,yk), axis equal
% Fyll hela kaustikan med gult, lagra koordinater runt om!
  u=(0:dfi:pi)';
  X=[cos(u); -xk; flipud(xk)];      % flipud vänder på vektorn
  Y=[sin(u);  yk; flipud(yk)];
  fill(X,Y,'y')
```

```
% MÖ 17.
% snedlada,  låda ritad med surf
  clear, clf
  x=[0 4 4 0]'; y=[0 0 3 3]'; z=[3 4 5 3]';
  subplot(1,2,1),  stem3(x,y,z), hold on
  stem3(x+1,y,z+2,'--'), axis equal
  x=[x; x(1)]; y=[y; y(1)]; z=[z; z(1)];
% Bilda matriser X, Y och Z:
  X=[x x x+1]; Y=[y y y]; Z=[0*z z z+2];
  subplot(1,2,2), surf(X,Y,Z), axis equal
  colormap cool, axis off
```

```
% MÖ 18.
% urtavla, väggur med sekund-, minut- och timvisare
  clear, clf
  R=30;
  LS=27; LM=23; LT=17;                                 % visarnas längder
  fi=0:2*pi/60:2*pi;
  plot(0,0,'o', R*cos(fi), R*sin(fi), 'Linewidth',3)
  axis square, hold on, drawnow                        % rita genast
  sx=0; sy=0; mx=0; my=0; tx=0; ty=0;
  dv=2*pi/60; olds=100;
  set(gcf,'DoubleBuffer','on')                         % undvik blinkningar
  while 1==1                                           % bryt med control-C
    tid=fix(clock);  t=tid(4); m=tid(5); s=tid(6);
    if s~=olds
      plot([0 sx],[0 sy],'w', [0 mx],[0 my],'w', [0 tx],[0 ty],'w')
      sx=LS*sin(dv*s); sy=LS*cos(dv*s);
      mx=LM*sin(dv*m); my=LM*cos(dv*m);
      v=2*pi*t/12+dv*m/12;
      tx=LT*sin(v); ty=LT*cos(v);
      plot(0,0,'o'),
      plot([0 sx],[0 sy],'g', [0 mx],[0 my],'r', [0 tx],[0 ty],'b')
      drawnow, olds=s;
    end
  end
```
