

Symboliska beräkningar med MATLAB Analys

1 Inledning

Verktygslådan SYMBOLIC MATH TOOLBOX i MATLAB kan utföra symbolisk matematik. Nu skall vi se på ett antal exempel på symboliska beräkningar inom algebra och analys samt symbolisk lösning av t.ex. vissa differentialekvationer. Till våren, i kursen i linjär algebra, kommer vi se på symbolisk lösning av linjära ekvationssystem och några andra problemtyper inom linjär algebra.

Vi börjar med att göra en symbolisk formel för volymen av ett klot med radien r

```
>> syms r
>> volym=4/3*pi*r^3
volym =
(4*pi*r^3)/3
```

Med `syms` talar vi om vilka variabler som skall vara symboliska. Variabeln `volym` blir automatiskt symbolisk eftersom variabeln `r` var det.

Vi kan beräkna volymen för t.ex. $r = 3$ med `subs` enligt

```
>> v=subs(volym,r,3)
v =
36*pi
```

Symboliska variabeln `r` i uttrycket `volym` substitueras med värdet 3 och `v` räknas ut. Vill vi ha ett numeriskt värde får vi det med funktionen `double` enligt

```
>> vn=double(v)
vn =
113.0973
```

Senare skall vi se hur vi kan göra om ett symboliskt uttryck till en `function`, som ett alternativt sätt att göra samma beräkning.

2 Algebraiska uttryck och aritmetik

Vi kan faktorisera, utveckla och förenkla ett uttryck med `factor`, `expand` och `simplify` enligt

```
>> syms a b
>> factor(a^2+2*a*b+b^2)
ans =
[ a + b, a + b]
```

```

>> factor(sym('1273'))           % Primtalsfaktorisering
ans =
[ 19, 67]

>> expand((a+b)^4)
ans =
a^4 + 4*a^3*b + 6*a^2*b^2 + 4*a*b^3 + b^4

>> expand(sin(a+b))
ans =
cos(a)*sin(b) + cos(b)*sin(a)

>> simplify((a^2+b^2-2*a*b)/(a-b))
ans =
a - b

>> simplify(exp(a+b)/exp(a-b))
ans =
exp(2*b)

```

Vi kan beräkna summor som exempelvis $\sum_{i=1}^n i$ eller $\sum_{i=1}^n i^2$ med

```

>> syms i n
>> s=symsum(i,1,n)
s =
(n*(n + 1))/2

>> s=symsum(i^2,1,n)
s =
(n*(2*n + 1)*(n + 1))/6

```

Vi ser att

$$\sum_{i=1}^n i = \frac{n(n+1)}{2} \quad \text{och} \quad \sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$$

En lite mer komplicerad summa som $\sum_{i=1}^n r^{i-1}$ beräknar vi med

```

>> syms i n r
>> s=symsum(r^(i-1),i,1,n)
s =
piecewise([r == 1, n], [r ~= 1, (r^n - 1)/(r - 1)])

```

och med lite tolkning av utskriften ser vi att

$$\sum_{i=1}^n r^{i-1} = \begin{cases} \frac{r^n-1}{r-1}, & r \neq 1 \\ n, & r = 1 \end{cases}$$

Vi kan lösa polynomekvationer som exempelvis $x^4 - x^2 + 2x - 1 = 0$ med

```
>> syms x
>> s=solve(x^4-x^2+2*x-1==0)
s =
    - 5^(1/2)/2 - 1/2
    5^(1/2)/2 - 1/2
    1/2 - (3^(1/2)*1i)/2
    (3^(1/2)*1i)/2 + 1/2
```

dvs. vi har lösningarna (rötterna) $x_{1,2} = -\frac{1}{2} \pm \frac{\sqrt{5}}{2}$ och $x_{3,4} = \frac{1}{2} \pm \frac{\sqrt{3}i}{2}$.

Vi kan få MATLAB att skriva ut lite snyggare (mer lättöläst) med `pretty(s)`. Prova gärna!

Vi kan med exakt heltalsaritmetik beräkna t.ex. 2^{100} och $100!$ enligt

```
>> N=sym(2)^100
```

N = 1267650600228229401496703205376

```
>> double(N)           % Numerisk form
```

```
ans =  
1.2677e+30
```

```
>> prod(sym(1:100))    % sym('100!') går också bra
```

```
ans =  
933262154439441526816992388562667004907159682643816214685929638952175999932299\  
15608941463976156518286253697920827223758251185210916864000000000000000000000
```

```
>> double(ans)
```

```
ans =  
9.3326e+157
```

Vi beräknar π , $\sqrt{2}$ och e med t.ex. 100 decimaler med `vpa` (variable precision arithmetic) enligt

```
>> Pi=vpa(pi,100)
```

Pi = 3.1415926535897932384626433832795028841971693993751058209749445923078164062862\08998628034825342117068

```
>> c=vpa(sqrt(sym(2)),100)
```

C = 1.4142135623730950488016887242096980785696718753769480731766797379907324784621\07038850387534327641573

```
>> e=vpa(exp(sym(1)),100)
```

$$e = 2.7182818284590452353602874713526624977572470936999595749669676277240766303535\backslash$$

3 Numerisk funktion av symboliska formler

I exemplet från inledningen gjorde vi en symbolisk formel för volymen av ett klot med radien r enligt

```
>> syms r
>> volym=4/3*pi*r^3
volym =
(4*pi*r^3)/3
```

Vi beräknade volymen numeriskt för $r = 3$ med

```
>> v=double(subs(volym,r,3))
v =
113.0973
```

Alternativt kan vi göra en (numerisk) funktion av den symboliska formeln med `matlabFunction` enligt

```
>> volymfun=matlabFunction(volym) % volymfun=matlabFunction(volym,'vars',{'r'})
volymfun =
@(r)r.^3.*pi.*(4.0./3.0)
```

Nu kan vi beräkna volymen för t.ex. $r = 3$ som en helt vanlig funktionsberäkning

```
>> v=volymfun(3)
v =
113.0973
```

Konstruktionen med `matlabFunction` gör det möjligt att rita upp symboliska funktioner. För att rita en graf måste man ju ha konkreta siffervärden. Detta kommer vi använda senare i denna laboration.

4 Gränsvärden, derivator och integraler

Vi kan beräkna gränsvärden

$$\lim_{x \rightarrow 0} \frac{\sin(ax)}{x} \qquad \lim_{x \rightarrow \pi/2^-} \tan(x) \qquad \lim_{x \rightarrow 0} x \sin\left(\frac{1}{x}\right)$$

```
>> syms x a
>> limit(sin(a*x)/x,x,0)
ans =
a

>> limit(tan(x),x,pi/2,'left')
ans =
Inf

>> limit(x*sin(1/x),x,0)
ans =
0
```

Vi kan derivera en funktion. Låt oss som exempel ta

$$f(x) = \tan(1 + \cos(x^2))$$

Först talar vi om att vi vill ha en symbolisk variabel `x`, och sedan beskriver vi funktionen enligt

```
>> syms x
>> f=tan(1+cos(x^2));
```

För att derivera använder vi funktionen `diff` enligt

```
>> Df=diff(f,x)
Df =
-2*x*sin(x^2)*(tan(cos(x^2) + 1)^2 + 1)
```

Vi kan också beräkna högre ordningens derivator, t.ex. `diff(f,x,2)` beräknar andraderivatan.

Här passar vi på att nämna att MATLAB kan producera L^AT_EX-kod för en formel

```
>> kod=latex(Df);
```

Vi infogar denna kod i det L^AT_EX-dokumentet som genererat denna sida med följande resultat

$$f'(x) = -2x \sin(x^2) \left((\tan(\cos(x^2) + 1))^2 + 1 \right)$$

Vi kan beräkna primitiv funktion. Som exempel bestämmer vi en primitiv funktion, eller antiderivata, $F(x)$ till $f(x) = 15x^2 - 8x + 9$, dvs. $F(x) = \int f(x) dx$.

```
>> syms x
>> f=15*x^2-8*x+9;
>> F=int(f,x)
F =
x*(5*x^2 - 4*x + 9)
```

```
>> F=expand(F)
F =
5*x^3 - 4*x^2 + 9*x
```

Integrationskonstanten får vi hålla reda på själva. Vi deriverar för att kontrollera svaret

```
>> DF=diff(F,x)
DF =
15*x^2 - 8*x + 9
```

Avslutningvis beräknar vi den bestämda integralen $\int_0^1 x \sin(x) dx$ enligt

```
>> syms x
>> f=x*sin(x);
>> q=int(f,x,0,1)
q =
sin(1) - cos(1)
```

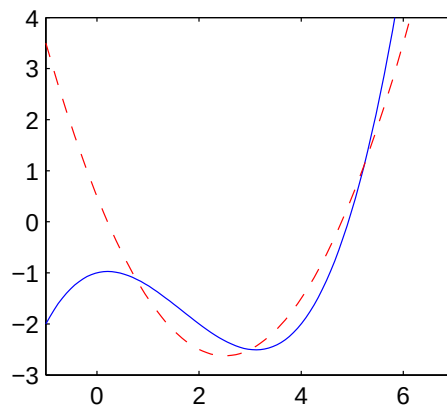
Kontrollera gärna svaret genom att räkna ut för hand med partialintegration.

Uppgift 1. Beräkna arean A av de områden som omsluts av graferna till funktionerna

$$f(x) = \frac{x^3}{8} - \frac{5x^2}{8} + \frac{x}{4} - 1 \quad g(x) = \frac{x^2}{2} - \frac{5x}{2} + \frac{1}{2}$$

Ledning: Definiera funktionerna, både symboliskt och numeriskt (avsnitt 3), och rita deras grafer med (man måste ha numeriska funktioner för att kunna rita grafer)

```
>> syms x
>> f=x^3/8-5*x^2/8+x/4-1;
>> g=x^2/2-5*x/2+1/2;
>> fn=matlabFunction(f);
>> gn=matlabFunction(g);
>> xn=linspace(-1,7);
>> plot(xn,fn(xn),xn,gn(xn),'r--')
>> axis equal, axis([-1 7 -3 4])
```



Bestäm sedan skärningspunkterna med `solve` (avsnitt 2) enligt

```
>> s=solve(f-g)
s =
      3
3 - 5^(1/2)
5^(1/2) + 3
```

för att sedan beräkna arean med `int` på lämpligt sätt. Integrationsgränserna plockar ni ur vektorn `s`, t.ex. `s(3)` ger skärningen längst till höger. (Svaret skall bli $A = \frac{25}{16}$.)

5 Differentialekvationer

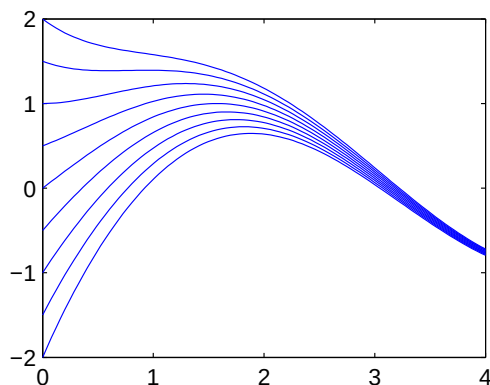
Vi beräknar lösningar till begynnelsevärdesproblemet

$$\begin{cases} u' = -u(t) + \sin(t) + \cos(t), & 0 \leq t \leq 4 \\ u(0) = c \end{cases}$$

```
>> u=dsolve('Du=-u+sin(t)+cos(t)', 'u(0)=c', 't')
u =
sin(t) + c*exp(-t)
```

Vi omvandlar det symboliska uttrycket till en numerisk funktion och ritar vi upp lösningarna för några olika värden på c enligt

```
>> ufun=matlabFunction(u,'vars',{ 't','c' })
ufun =
    @(t,c)sin(t)+c.*exp(-t)
>> T=linspace(0,4);
>> for c=-2:0.5:2
    plot(T,ufun(T,c)), hold on
end
```



Vi löser andra ordningens differentialekvation

$$u'' + 2u' + u = 2 \sin(t)$$

```
>> u=dsolve('D2u+2*Du+u=2*sin(t)','t')
u =
C1*exp(-t) - cos(t) + C2*t*exp(-t)
```

och kontrollerar att lösningen verkligen uppfyller ekvationen med

```
>> diff(u,2)+2*diff(u)+u
ans =
2*sin(t)
```

Uppgift 2. Rita graferna av lösningen till differentialekvationen $u' + \frac{u}{t+1} = (t-1) \sin(t)$ över intervallet $-10 \leq t \leq 10$, för några olika värden på konstanten i lösningen.

6 Taylorutveckling

Vi kan Taylorutveckla exempelvis $f(x) = \sqrt{x}$ runt $a = 4$, och ta med termer t.o.m. ordning $n = 3$, med funktionen `taylor` enligt

```
>> syms x a n
>> a=4; n=3;
>> t=taylor(sqrt(x),x,a,'Order',n+1)
t =
x/4 - (x - 4)^2/64 + (x - 4)^3/512 + 1
```

Lägg märke till att vi skall ge gradtalet plus 1 som parameter till `taylor`. Resttermen bestämmer vi med

```
>> syms xi
>> r=diff(sqrt(xi),n+1)/factorial(n+1)*(x-a)^(n+1)
r =
-(5*(x - 4)^4)/(128*xi^(7/2))
```

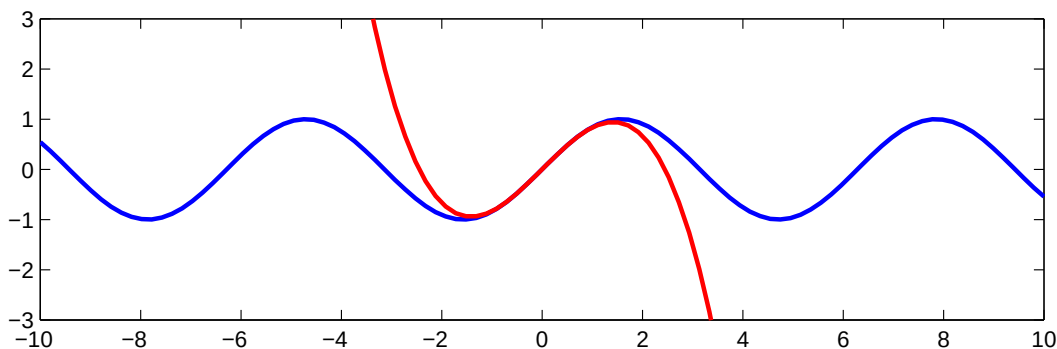
Vi får alltså $f(x) = 2 + \frac{x-4}{4} - \frac{(x-4)^2}{64} + \frac{(x-4)^3}{512} - \frac{5\xi^{-7/2}}{128}(x-4)^4$.

Vi kan Maclaurinutveckla $f(x) = \sin(x)$ med

```
>> syms x n
>> n=3;
>> f=sin(x);
>> t=taylor(f,'Order',n+1)
t =
- x^3/6 + x
```

och rita upp funktionen och Maclaurinpolynomet med

```
>> fn=matlabFunction(f);      % fn blir numerisk motsvarighet till f
>> tn=matlabFunction(t);      % tn blir numerisk motsvarighet till t
>> xn=linspace(-10,10);      % xn numeriska x-värden för uppritning
>> plot(xn,fn(xn),'b',xn,tn(xn),'r','linewidth',2)
>> axis equal, axis([xn(1) xn(end) -3 3])
```



Uppgift 3. Bestäm Maclaurinpolynomen till $f(x) = \sin(x)$ av successivt allt högre gradtal ($n = 1, 3, \dots, 21$ exempelvis). Rita successivt upp Maclaurinpolynomen i en figur där ni redan har ritat $f(x)$. Lägg en liten paus mellan varje uppritning.

1 Målsättning

Avsikten med denna laboration är att ge en introduktion till symboliska beräkningar med MATLAB. Vi ser hur vi kan använda exakt heltalsaritmetik (godtyckligt stora heltal) och variabel precision vid reella beräkningar. Vidare ser vi på beräkningar liknande de ni gjort för hand, ta gränsvärden, derivera och integrera. Avslutningsvis ser vi hur man kan lösa vissa differentialekvationer och hur man Taylorutvecklar.

2 Kommentarer och förklaringar

Inget kan ersätta handräkning, men ibland kan man vara hjälpt av att ha möjlighet att låta MATLAB eller något liknande programsystem utföra en del kalkyler.

Försöker vi lösa en polynomekvation av grad fem eller högre så ger `solve` inte någon formel. Lösningarna representeras istället av `root` och vi kan få numeriska värden om vi vill. T.ex. har vi

```
>> syms x
>> s=solve(x^5-x^3+4*x-1==0)
s =
root(z^5 - z^3 + 4*z - 1, z, 1)
root(z^5 - z^3 + 4*z - 1, z, 2)
root(z^5 - z^3 + 4*z - 1, z, 3)
root(z^5 - z^3 + 4*z - 1, z, 4)
root(z^5 - z^3 + 4*z - 1, z, 5)

>> double(s)
ans =
    0.2538 + 0.0000i
   -1.1743 + 0.8833i
   -1.1743 - 0.8833i
    1.0473 + 0.8531i
    1.0473 - 0.8531i
```

3 Lärandemål

Efter denna laboration skall du i MATLAB

- ha kännedom om möjlighet till symboliska formelhantering med `factor` och `expand`
- ha kännedom om möjlighet till exakt heltalsaritmetik och beräkningar i variabel precision med `vpa`
- kunna omvandla en symbolisk funktion till en numerisk med `matlabFunction`

- kunna beräkna derivator, primitiva funktioner och integraler med `diff` och `int`
- kunna beräkna symboliska lösningar till algebraiska ekvationer och differentialekvationer med `solve` respektive `dsolve`