Numerisk Analys, MMG410. Lecture 16.

Kvadratur (Singulariteter)

Om någon av f:s lägre derivator har en singularitet i $[a, b]$ kan dock metoderna konvergera avsevärt långsammare.

Example

Trapetsmetoden på $f(x) = x^p, \ 0 < p < 1, \ [a, b] = [0, 1]$.
Vi kan ej använda feluppskattningen på hela intervallet eftersom f' och f'' har en singularitet i nollan. Vi kan dock räkna ut skillnaden mellan integral och approximation för $x \in [0, h]$:

$$\int_0^h x^p dx - \frac{h[0^p + h^p]}{2} = \frac{(1-p)}{2(1+p)} h^{1+p}$$

Man skulle kunna använda feluppskattningen på $[h, 1]$ för att visa konvergens (felet går mot noll när $h \to 0$), men det blir ett väldigt svagt resultat.
Använder man uppskattningen på $[h, 2h]$, $[2h, 3h]$ etc. får man ett bra resultat som visar att felet uppför sig som h^{1+p}. Det förväntar man sig även för de övriga metoderna.

Kvadratur (Singulariteter)

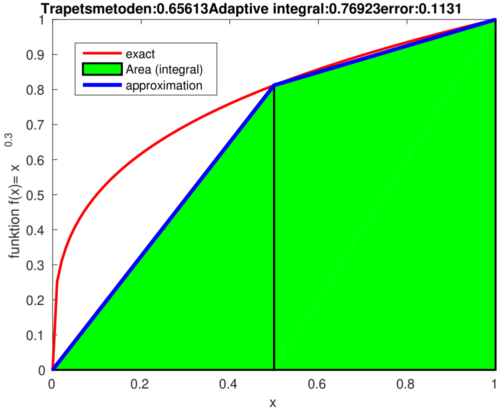
Tar vi $p = 0.3$ med samma tolerans som i föregående exempel, så kräver Simpson inte 52 funktionsberäkningar utan 1 697 157. Problemet är väsentligen av samma slag som när vi interpolerade $\sqrt{t}$ kring $t \geq 0$. Vad kan man göra? Man kan byta till en bättre metod, `integral` t.ex. som kräver 150 funktionsberäkningar. Kanske kan man byta parametrisering av f och betrakta x som funktion av y (givetvis förutsatt att f^{-1} existerar lokalt) och sedan integrera i y-led (lite mer fixande krävs för att få rätt integral).

I exemplet: vi gör variabelbyte $y = x^{0.3}$ ger $x = y^{1/0.3}$ som har en singularitet först i fjärdederivatan: $dx = 1/0.3y^{1/0.3-1}dy$

$$\int_0^1 x^{0.3} dx = \int_0^1 \underbrace{y}_{x^{0.3}} \cdot \underbrace{\frac{1}{0.3} \frac{y^{1/0.3}}{y} \, dy}_{dx} \approx 0.7692$$

Simpson tar 83 funktionsberäkningar för y-integralen, `quadl` tar 48, `integral` tar 150, med väsentligen nollfel.

Numerisk integration av $\int_0^1 x^{0.3} dx$



Matlab's program Integrsingular_ex3.m som beräknar $\int_0^1 x^{0.3} dx$ (finns på kursens hemsida): Integral beräknad med adaptivt metod (Matlabs funktion **integral**: 0.7692, integral beräknad med Simpson's formel i 3 punkter: 0.7082, integral beräknad med trapetsmetoden i 3 punkter: 0.6561.

Kvadratur (Adaptivitet)

Normalt vill vi inte ha ekvidistanta punkter, utan vi vill att metoden automatiskt ska anpassa sig efter funktionens utseende och använda tätare med punkter där det behövs. Vi behöver då en uppskattning av felet.
Att direkt uppskatta feltermen gör man normalt inte. En vanlig metod är att räkna ut resultatet med två metoder (en med mindre fel) och jämföra resultaten. Kostnaden bör vara som för en metoden. Man kan också använda samma metoden men med olika antal punkter.
I boken används den senare varianten med trapetsmetoden (Simpson, eller bättre, är vanligare). Här följer en genomgång.

Vi börjar med intervallet $[a, b]$ räknar ut trapetsapproximationen med två punkter. Vi lägger sedan till mittpunkter, $m = (a + b)/2$ och räknar ut en ny approximation, nu med tre punkter. Observera att detta kräver ett nytt funktionsvärde, $f(m)$.

Kvadratur (Adaptivitet)

Vi fortsätter nu så rekursivt på intervallen $[a, m]$ och $[m, b]$. När felet över ett intervall är tillräckligt litet halverar vi inte detta intervall vidare. Antag att vi har kommit ner till ett delintervall av längd h. Approximationerna kan skrivas (I är det exakta värdet av integralen över detta delintervall).

$$I = T_h - h^3 f''(\xi)/12 \text{ resp. } I = T_{h/2} - h(h/2)^2 f''(\theta)/12$$

Antag att $c = -f''(\xi) \approx -f''(\theta)$ (behöver inte vara sant). Då gäller:

$$\begin{aligned}
0 &\approx T_h + h^3 c/12 - (T_{h/2} + h(h/2)^2 c/12) && (1)\\
&= T_h - T_{h/2} + ch^3(1 - 1/4)/12 = T_h - T_{h/2} + 3ch^3/(4 \cdot 12) && (2)\\
&= T_h - T_{h/2} + 3(I - T_{h/2}). && (3)
\end{aligned}$$

Men felet $I - T_{h/2}$ är ju $ch^3/(4 \cdot 12) = ch(h/2)^2/12$. Alltså

$$I \approx T_{h/2} + \frac{T_{h/2} - T_h}{3} = (4T_{h/2} - T_h)/3.$$

Kvadratur (Adaptivitet)

Man kan notera att formeln ovan även ger upphov till en ny metod. Om vi lägger till feluppskattningen får vi

$$I \approx (4T_{h/2} - T_h)/3$$

och bakom denna formel döljer sig Simpsons formel:

$$I = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right].$$

Richardsonextrapolation

Ovanstående är ett specialfall av Richardsonextrapolation. Man kan visa att det existerar en serieutveckling av felet

$$\left(\int_a^b f(x)dx = \right) = I = T_h + a_1 h^2 + a_2 h^4 + a_3 h^6 + \ldots \tag{4}$$

Vi halverar nu h och får

$$I = T_{h/2} + a_1 h^2/4 + a_2 h^4/16 + a_3 h^6/64 + \ldots$$

Vi ser att

$$4I = 4T_{h/2} + a_1 h^2 + a_2 h^4/4 + a_3 h^6/16 + \ldots \tag{5}$$

Vi beräknar nu differens (5) - (4) för att bli av med h^2-termen:

$$3I = 4I - I = 4T_{h/2} - T_h + \underbrace{(a_1 h^2 - a_1 h^2)}_{=0} + \underbrace{(a_2 h^4/4 - a_2 h^4)}_{\frac{-3a_2 h^4}{4}} + \ldots$$

så att

$$I = \frac{4T_{h/2} - T_h}{3} - \frac{a_2 h^4}{4} + \ldots$$

Detta kan man upprepa (med $T_{h/4}$) för att bli av med h^4-termen. Denna process (upprepad Richardsonextrapolation) kallas Rombergkvadratur. Rombergs metoden kan definieras induktivt:

$$R(0,0) = h_1(f(a) + f(b)), \tag{6}$$

$$R(n,0) = \tfrac{1}{2}R(n-1,0) + h_n \sum_{k=1}^{2^{n-1}} f(a + (2k-1)h_n), \tag{7}$$

$$R(n,m) = R(n,m-1) + \tfrac{1}{4^m-1}(R(n,m-1) - R(n-1,m-1)), \tag{8}$$

eller

$$R(n,m) = \tfrac{1}{4^m-1}(4^m R(n,m-1) - R(n-1,m-1))$$

var $n \geq m$ och $m \geq 1$.

Felet för $R(n,m)$ är (Mysovskikh 2002):

$$O\left(h_n^{2m+2}\right).$$

Noll extrapoleringen, $R(n,0)$, motsvarar trapezmetoden med $2^n + 1$ punkter. Den första extrapoleringen, $R(n,1)$, motsvarar Simpsons formel med $2^n + 1$ punkter.

Gausskvadratur

Antag att vi vill beräkna $\int_a^b f(x)dx$ och tillåts göra tre funktionsberäkningar, $f(x_1), f(x_2)$ samt $f(x_3)$. Om vi väljer $x_1 = a$, $x_2 = (a+b)/2$ och $x_3 = b$ så kommer Simpsons formel att vara optimal när det gäller polynomiellt gradtal. Dvs. om vi vill att metoden ska vara exakt för polynom av grad $0, 1, ..., m$ för så stort m som möjligt så är Simpsons metod det bästa valet $(m = 3)$. Det visar sig dock att vi kan få större m genom att välja andra x_k-värden. Detta är kärnan i Gausskvadratur, att välja både x_k-värden och vikter för att maximera m.

Gausskvadratur

Gausskvadratur är konstruerad för att ge ett exakt resultat för polynomier av grad $2n - 1$ eller mindre med ett lämpligt val av punkterna x_i och vikterna $w_i, \; i = 1, ..., n$:

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} w_i f(x_i),$$

Gausskvadratur ger bra resultat om funktionen $f(x)$ är väl approximerad av en polynomfunktion inom intervallet $[-1, 1]$. Metoden är exempelvis inte lämplig för funktioner med singulariteter.

Kvadratur (Gausskvadratur)

Tag intervallet $[-1, 1]$. Ska välja x_1, x_2, x_3 och vikter w_1, w_2, w_3 s.a.

$$\int_{-1}^{1} x^k dx = w_1 x_1^k + w_2 x_2^k + w_3 x_3^k, \;\; k = 0, 1, ..., m$$

för maximalt m. Integralens värde blir 0 om k är uddda och $2/(k+1)$ annars. Vi få lösa det ickelinjära ekvationssystemet:

$$\begin{aligned}
2 &= w_1 + w_2 + w_3 && k = 0 \\
0 &= w_1 x_1 + w_2 x_2 + w_3 x_3 && k = 1 \\
2/3 &= w_1 x_1^2 + w_2 x_2^2 + w_3 x_3^2 && k = 2 \\
0 &= w_1 x_1^3 + w_2 x_2^3 + w_3 x_3^3 && k = 3 \\
2/5 &= w_1 x_1^4 + w_2 x_2^4 + w_3 x_3^4 && k = 4 \\
0 &= w_1 x_1^5 + w_2 x_2^5 + w_3 x_3^5 && k = 5
\end{aligned}$$

Det verkar inte rimligt att ta med en ekvation till. Vi har ju $3 + 3 = 6$ obekanta och sex ekvationer. För att lösa systemet kan man använda "brute force", men vi utnyttjar symmetri och antar att $x_1 < x_2 < x_3$ med $x_2 = 0$ och $x_1 = -x_3$.

Detta leder ($k = 1$) till att $w_1 = w_3$ och satisfiering av fallen $k = 3, 5$. Kvarstår då ekvationerna $2 = 2w_1 + w_2$, $2/3 = 2w_1 x_1^2$ samt $2/5 = 2w_1 x_1^4$. Vi får $x_1 = -\sqrt{3/5}$ och $w_1 = 5/9$. Metoden blir alltså:

$$\int_{-1}^{1} f(x)dx \approx \frac{5}{9} f\left(-\sqrt{3/5}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\sqrt{3/5}\right)$$

Man ser att metoden inte är exakt för $m = 6$ så det polynomiella gradtalet är 5 (det var 3 för Simpsons metod).
Eftersom integration är en linjär operation så är metoden exakt för alla polynom av grad högst 5.
För en Gausskvadraturformel har vi gradtalet $2n - 1$ med n punkter. Vi har dock offrat i enkelhet. Härledningen kan dock förenklas (man använder teorin för ortogonala polynom och kan blanda in egenvärdesproblem för tridiagonala matriser). En annan nackdel är att värdena måste skrivas in i ett program (stora tabeller).

Gausskvadratur:

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} w_i f(x_i),$$

för Gausspunkter x_i och Gaussvikter ω_i, $i = 1, ..., n$.

n	Gausspunkter, x_i	Vikter, ω_i	Gausskvadratur	pol. gradtal $2n-1$
1	0	2	$w_1 f(x_1)$	$f(x) \approx p(x) = x^1 =$
2	$\pm\sqrt{1/3}$	1	$\sum_{i=1}^{2} w_i f(x_i)$	$f(x) \approx p(x) = x^3$
3	0	$8/9$	$\sum_{i=1}^{3} w_i f(x_i)$	5, x^k, $k = 5$
	$\pm\sqrt{3/5}$	$5/9$		$f(x) \approx p(x) = x^5$
4	$\pm\sqrt{3/7 - 2/7\sqrt{6/5}}$	$\frac{18+\sqrt{30}}{36}$	$\sum_{i=1}^{4} w_i f(x_i)$	$f(x) \approx p(x) = x^7$
	$\pm\sqrt{3/7 + 2/7\sqrt{6/5}}$	$\frac{18-\sqrt{30}}{36}$		
5	0	$128/225$	$\sum_{i=1}^{5} w_i f(x_i)$	9
	$\pm\frac{1}{3}\sqrt{5 - 2\sqrt{\frac{10}{7}}}$	$\frac{322+13\sqrt{70}}{900}$		$f(x) \approx p(x) = x^9$
	$\pm\frac{1}{3}\sqrt{5 + 2\sqrt{\frac{10}{7}}}$	$\frac{322-13\sqrt{70}}{900}$		

Övning

Vi har följande kvadraturformel:

$$\int_0^1 f(x)dx = \sum_{i=1}^{n} w_i f(x_i),$$

där vi vet att vi approximerar $f(x)$ med polynom $p(x)$ som har polynomiella gradtalet minst ett. Visa att $\sum_{i=1}^{n} w_i = 1$.

Övning

Svar:
Vi har följande kvadraturformel:

$$\int_0^1 f(x)dx = \sum_{i=1}^{n} w_i f(x_i),$$

där vi vet att vi approximerar $f(x)$ med polynom $p(x)$ som har polynomiella gradtalet minst ett. Visa att $\sum_{i=1}^{n} w_i = 1$.
Svar:

$$1 = \int_0^1 1dx = \sum_{i=1}^{n} w_i p(x_i) = \sum_{i=1}^{n} w_i,$$

Övning

Välj w_1, w_2, x_1, x_2, i kvadraturformeln nedan, så att den får så högt polynomiellt gradtal m som möjligt. Vad blir detta gradtal ?

$$\int_0^1 x^k dx = w_1 x_1^k + w_2 x_2^k, \quad k = 0, 1, ..., m.$$

Övning

Välj w_1, w_2, x_1, x_2, i kvadraturformeln nedan, så att den får så högt polynomiellt gradtal m som möjligt. Vad blir detta gradtal ?

$$\int_0^1 x^k dx = w_1 x_1^k + w_2 x_2^k, \quad k = 0, 1, ..., m.$$

Svar: Formeln skall vara exakt för polynom $x^k, k = 0, 1, ..., m$ för maximalt m. Vi beräknar först

$$\int_0^1 x^k dx = \frac{x^{k+1}}{k+1}\Big|_0^1 = 1/(k+1).$$

Ekvationerna blir:

$$\begin{aligned} 1 &= w_1 + w_2, k = 0, \\ 1/2 &= w_1 x_1 + w_2 x_2, k = 1, \\ 1/3 &= w_1 x_1^2 + w_2 x_2^2, k = 2, \\ 1/4 &= w_1 x_1^3 + w_2 x_2^3, k = 3, \\ 1/5 &= w_1 x_1^4 + w_2 x_2^4, k = 4. \end{aligned}$$

Övning

$$
\begin{aligned}
1 &= w_1 + w_2, k = 0,\\
1/2 &= w_1x_1 + w_2x_2, k = 1,\\
1/3 &= w_1x_1^2 + w_2x_2^2, k = 2,\\
1/4 &= w_1x_1^3 + w_2x_2^3, k = 3,\\
1/5 &= w_1x_1^4 + w_2x_2^4, k = 4.
\end{aligned}
$$

Första ekvationen ger $w_{1,2} = 1/2$. Lös ut för $k = 1, 2$ ekvationen $2x_2^2 - 2x_2 + \frac{1}{3} = 0$ (vi noterar, att $x_1 < x_2$) för att få $x_1 = \frac{1-1/\sqrt{3}}{2}, x_2 = \frac{1+1/\sqrt{3}}{2}$. Vi kollar nu fall $k = 3$. Utnyttjar vi binomialsatsen ser vi att $(1 + c)^3 + (1 - c)^3 = 2(1 + 3c^2)$ så att $w_1x_1^3 + w_2x_2^3 = (1/2^4) \cdot 2(1 + 3/3) = 1/4$, vilket är lika med det exakta värdet. Stämmer det för k = 4? Inte. Så, det polynomiella gradtalet är 3.

Kvadratur (Gausskvadratur)

Det allvarligaste problemet är dock att man inte kan återanvända funktionsvärdena när man gör adaptiva metoder. Det finns dock varianter, Gauss-Kronrodkvadratur, där man har en kompromiss mellan optimaliteten i Gausskvadratur och kraver på återanvändning av funktionsvärden, se boken. Finns `quadgk` i Matlab.

Hur ser vår metod ut på intervallet $[a, b]$, $\int_a^b f(z)dz$?
Sätt $z = \alpha x + \beta$ där $\alpha = (b-a)/2$ och $\beta = (a+b)/2$.
$z \in [a, b] \to x \in [-1, 1]$. $dz = \alpha\, dx$. Alltså:

$$\int_a^b f(z)dz = \int_{-1}^1 f(\alpha x + \beta)\alpha\, dx \approx \sum_{k=1}^{3} (\alpha w_k) f(\alpha x_k + \beta)$$

Kvadratur (Gausskvadratur)

Om vi ska approximera integral

$$I(g) = \int_a^b g(t)dt,$$

och t ligger i ett annat intervall, $[a, b]$, och x ligger på $[-1, 1]$, får vi göra en linjär avbilding till detta intervall:

$$\frac{b-a}{2}[-1, 1] + \frac{a+b}{2} = [a, b],$$

eller

$$t = \frac{b-a}{2}x + \frac{a+b}{2},$$

och integral $I(g) = \int_a^b g(t)dt$ kan beräknas som

$$I(g) = \int_a^b g(t)dt = \frac{b-a}{2}\int_{-1}^{1} g\left(\frac{b-a}{2}x + \frac{a+b}{2}\right)\, dx \tag{9}$$

$$\approx \frac{b-a}{2}\sum_{i=1}^{n} \omega_i g\left(\frac{b-a}{2}x_i + \frac{a+b}{2}\right). \tag{10}$$

Kvadratur (Gausskvadratur)

Example

Vi vill beräkna

$$\int_0^3 f(x)dx = \int_0^3 e^{-x^2} dx$$

med hjälp av Gausskvadratur med 3 vikter.
Metoden (Gausskvadratur med 3 vikter) är:

$$\int_{-1}^{1} f(x)dx \approx \frac{5}{9} f\left(-\sqrt{3/5}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\sqrt{3/5}\right)$$

Vi transformerar interval $[0, 3]$ för x, till $[-1, 1]$ för t, med hjälp av formula:

$$x = \frac{b-a}{2} t + \frac{a+b}{2} = \frac{3-0}{2} t + \frac{3+0}{2}$$

Example

och integral $\int_0^3 e^{-x^2}dx$ för $f(x) = e^{-x^2}$ kan beräknas som

$$
\begin{aligned}
\int_0^3 e^{-x^2}dx &= \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}t + \frac{a+b}{2}\right)\, dt \\
&\approx \frac{b-a}{2}\sum_{i=1}^{3}\omega_i f\left(\frac{b-a}{2}t_i + \frac{a+b}{2}\right) \\
&= \frac{3-0}{2}\cdot\left(\frac{5}{9}\cdot f\left(\frac{3-0}{2}t_1 + \frac{3+0}{2}\right)\right. \\
&+ \frac{8}{9}\cdot f\left(\frac{3-0}{2}t_2 + \frac{3+0}{2}\right) + \frac{5}{9}\cdot f\left(\frac{3-0}{2}t_3 + \frac{3+0}{2}\right))
\end{aligned}
$$

i Gausspunkter

$$t_1 = -\sqrt{3/5}; t_2 = 0; t_3 = \sqrt{3/5};$$

med vikter

$$\omega_1 = 5/9; \omega_2 = 8/9; \omega_3 = 5/9.$$

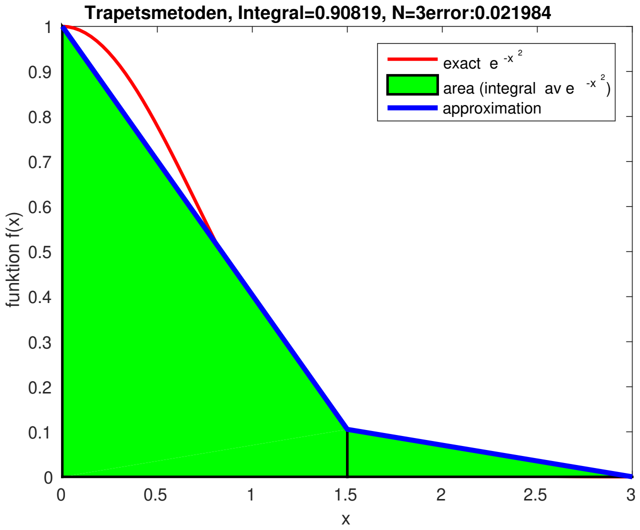
Gausskvadratur i Matlab

```
fun = @(x) exp(-x.^2)
% definition av integration intervalet [n,p]
n= 0.0; = 3.0
% adaptivt integral
Q =  integral(fun,n,p)
%  Gauss punkter
x(1) = -sqrt(3/5); x(2) = 0; x(3) = sqrt(3/5);
for i=1:3
t(i) = ((p-n)/2.0)*x(i) + (p+n)/2.0;
end
%vikter
omega(1)= 5/9;
omega(2)= 8/9;
omega(3) = 5/9;
Int = ((p-n)/2.0)*(omega(1)*fun(t(1))+ omega(2)*fun(t(2)) + ...
omega(3)*fun(t(3)))
```

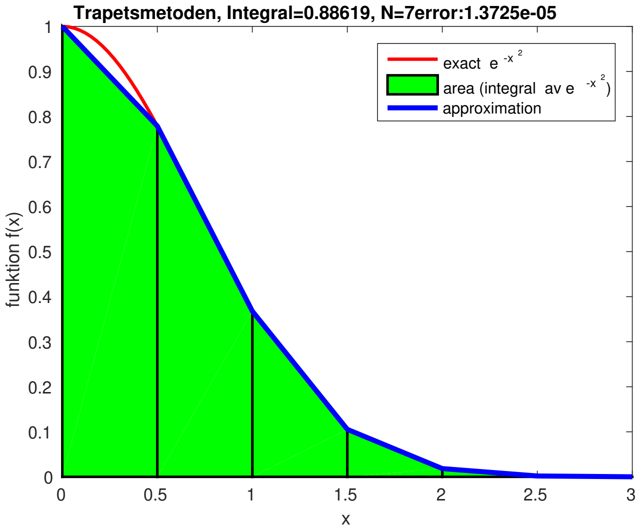
Gausskvadratur i Matlab

Vi får följande svar:
Adaptivt metod i Matlab (integral): Q= 0.8862
Gausskvadratur med 3 vikter: Int = 0.8845
Trapetsmetoden med 21 punkt: 0.8862
Trapetsmetoden med 3 punkter: 0.9082
Trapetsmetoden med 7 punkter: 0.8862

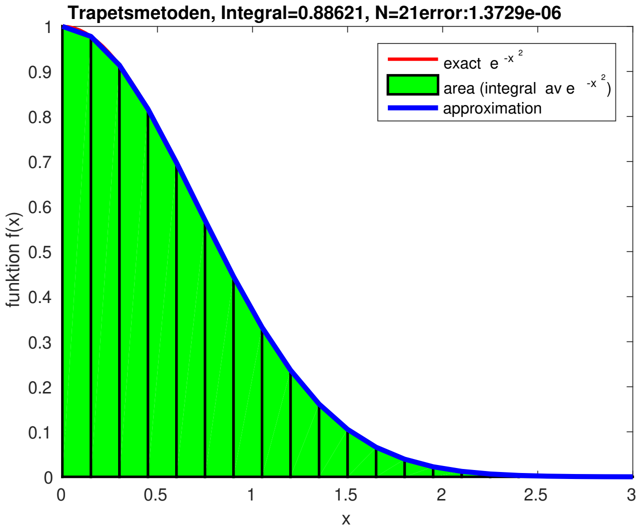
Trapetsmetoden för $\int_0^3 f(x) = exp(-x^2)$



Trapetsmetoden för $\int_0^3 f(x) = exp(-x^2)$



Trapetsmetoden för $\int_0^3 f(x) = exp(-x^2)$



Ordinära differentialekvationer

Vi kommer enbart att studera begynnelsevärdesproblem, t.ex.

$$y'(t) = t^2 + \sin y(t),\ 3 < t \leq 10,\ y(3) = 4$$

t är tiden och $3 < t \leq 10$ anger det intervall där vi vill approximera lösningen. $y(3) = 4$ är ett begynnelsevillkor som anger y:s begynnelsevärde, 4, vid tiden $t = 3$. Normalt skriver man aldrig ut t i $y(t)$. Vi struntar även i intervallet (tiden i begynnelsevillkoret är vänster ändpunkt, och man får anta något lämpligt slutvärde). Problemet kan då formuleras

$$y' = \underbrace{t^2 + \sin y(t)}_{f(t,y)},\ y(3) = 4$$

Normalt vill vi studera ett generellt problem, vi skriver:

$$y' = f(t, y),\ y(t_0) = y_0$$

Så, i exemplet ovan är $f(t, y) = t^2 + \sin y$. Begynnelsetiden är t_0 (3 i exemplet) och y vid detta värde är y_0 (4 i exemplet). Både t_0 och y_0 måste vara kända.

```
y0 = 4;             % begynnelsevarden
t0 = 3;            % begynnelsetid
ts = 10;             % slut-tid

h= 0.1;             %steglangd h
N = (ts - t0)/h    %antal punkter

[t, y] = ode45(@func_example1, linspace(t0, ts, N), y0);

figure
plot(t, y, 'b -o', 'LineWidth',2)

  xlabel('t');
ylabel('y(t)')
legend('t_0= 3, y(3)= 4')
title('ode45  f\"or dy/dt = t^2 + sin(y)')
```

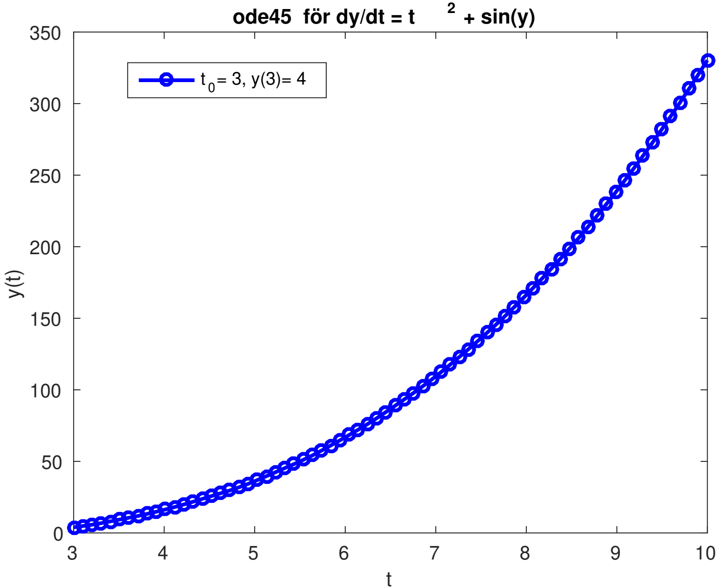
ODE (Matlabs ode45)

Vi definerar separat matlabs-file func_example1.m med funktion

```
function [dy] = func_example1(t, y)

dy = 0;
dy =  t^2 + sin(y);
```

Example: begynnelsevärdesproblem för $y' = t^2 + \sin y(t)$



Lösningsmetoderna genererar approximationer till lösningen för en uppsättning tidpunkter: $(t_0, y_0), (t_1, y_1), \ldots, (t_n, y_n)$, där t_n är slut-tiden och $y_k \approx y(t_k)$.

y_k är en <u>approximation</u> av lösningen vid tiden $t = t_k$.
Det <u>exakta</u> värde är $y(t_k)$.

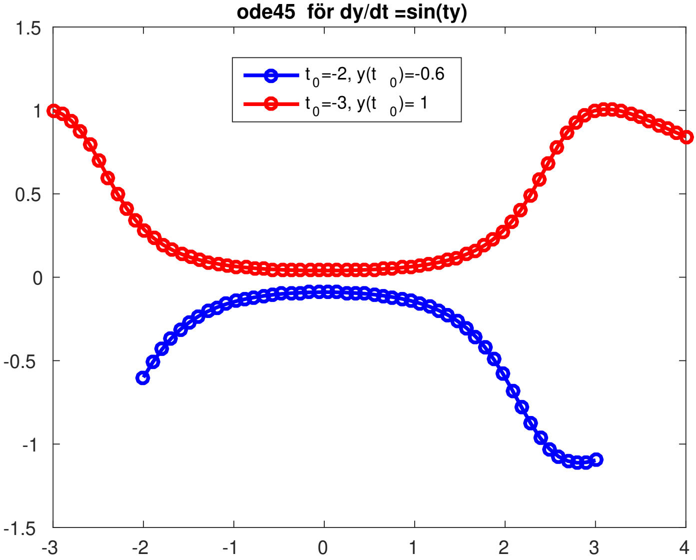
Senare kommer system av ekvationer. Sådana behövs för att vi skall kunna lösa problem som innehåller högre derivator., t.ex.

$$y''' = t + 2y'' + (y')^2 + \sin y, \ y(0) = 2, \ y'(0) = -3, \ y''(0) = -4$$

Låt oss studera problemet $y' = 1$. Detta är inget begynnelsevärdesproblem (eftersom vi saknar $y(t_0) = y_0$). Ett problem av detta slag har oändligt många lösningar, i detta fall $y(t) = t + c$, där c är ett godtyckligt reellt tal. När vi ger ett begynnelsevillkor väljer vi ut en av alla dessa oändligt många lösningar. $y(3) = 4$. ger oss lösningen $y(t) = t + 1$. Med grafiska verktyg kan vi skaffa oss en bild av lösningsmängden även för problemet $y' = f(t, y)$. Detta kan göras genom att i ett lämpligt antal punkter i (t, y)-planet rita en vektor som svarar mot den derivata som lösningen måste ha enligt ekvationen $y' = f(t, y)$.

Det finns begynnelsevärdesproblem som saknar, eller har flera lösningar. Det kan också vara så att $y(t)$ inte existerar för alla $t > t_0$ (om $y(t) \to \infty$ t.ex.).

Example: begynnelsevärdesproblem för $y' = sin(ty)$



Ordinära differentialekvationer (Eulers metod)

En enkel lösningsmetod: Vi startar i (t_0, y_0) (som är känd) och tar sedan ett litet steg utmed tangenten till lösningen (som kan beräknas med hjälp av $f(t, y)$.). Antag att vi stegar med fix steglängd, h, i t så att: $t_1 = t_0 + h,\ t_2 = t_1 + h,\ t_3 = t_2 + h, \dots$. Allmänt $t_k = t_0 + kh$. Vi får Eulers metod:

$$y_{k+1} = y_k + hf(t_k, y_k),\ k = 0, 1, 2, \dots$$

eller utskrivet
$y_1 = y_0 + hf(t_0, y_0),\ y_2 = y_1 + hf(t_1, y_1),\ y_3 = y_2 + hf(t_2, y_2), \dots$

Example

$y' = \sin(ty),\ y(-1) = 1$. Så $t_0 = -1$ och $y_0 = 1$ och $f(t, y) = \sin(ty)$.
Om $h = 0.1$ får vi approximationerna

$$y_1 = y_0 + hf(t_0, y_0) = 1 + 0.1\sin(-1 \cdot 1) \approx 0.9159$$
$$y_2 = y_1 + hf(t_1, y_1) \approx 0.9159 + 0.1\sin(-0.9 \cdot 0.9159) \approx 0.8425$$
$$y_3 = y_2 + hf(t_2, y_2) \approx 0.8425 + 0.1\sin(-0.8 \cdot 0.8425) \approx 0.7801 \text{ osv.}$$

```
t0 = -1;           % begynnelsetid
ts = 5;            % slut-tid
h= 0.5;            %steglangd h
N = (ts - t0)/h   % antal punkter

  t = linspace(t0,ts,N);
  y = linspace(t0,ts,N);
  y(1) = 1;          % begynnelsevarden

for k = 1:N
    y(k+1) = y(k) + h*func_example3(t(k),y(k));
    t(k+1) = t(k) + h;

 end
 figure
plot(t, y, 'g -o ', 'LineWidth',2)
```

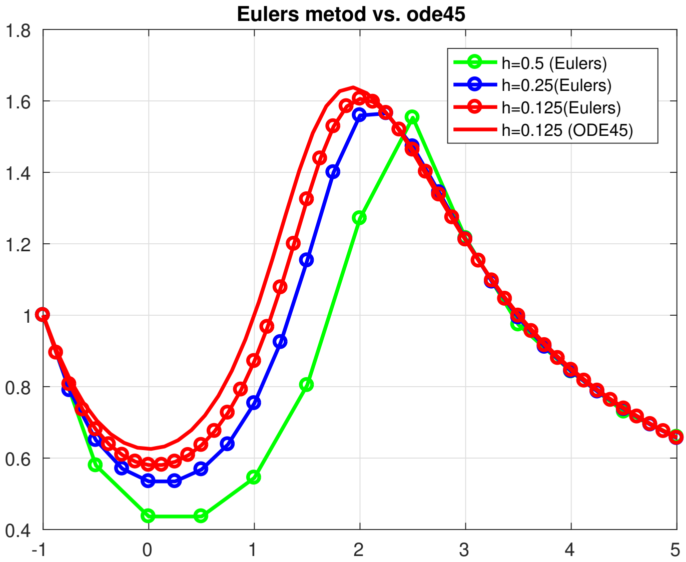
Framåt Eulers metod i Matlab för $y' = \sin(ty)$, $y(-1) = 1$

Vi definerar separat matlabs-file func_example3.m med funktion

```
function dy = func_example3(t, y)

dy = 0;
dy =  sin(t*y);
```

ODE: Eulers metod vs ode45 för $y' = \sin(ty),\ y(-1) = 1$



ode45 i Matlab för $y' = \sin(ty), \; y(-1) = 1$

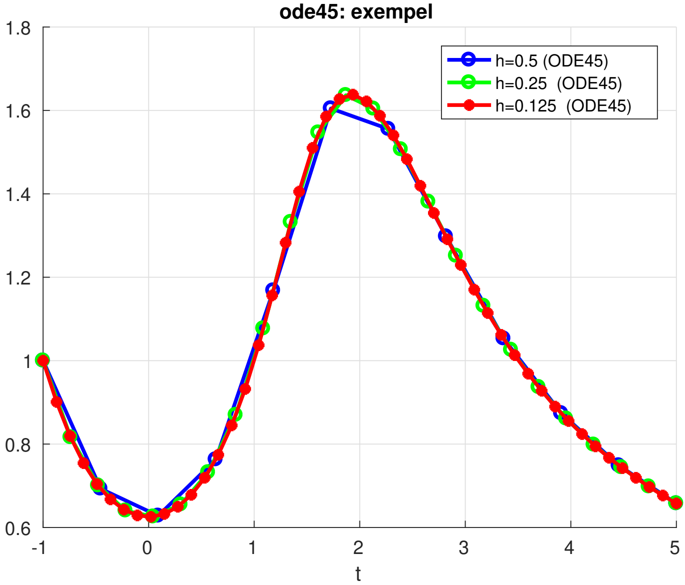
```
y0 = 1;             % begynnelsevarden
t0 = -1;            % begynnelsetid
ts = 5;             % slut-tid

h= 0.5;             %steglangd h
N = (ts - t0)/h     %antal punkter

[t, y] = ode45(@func_example3, linspace(t0, ts, N), y0);

figure
hold on
plot(t, y(:, 1), 'b -o', 'LineWidth',2)
```

ODE: ode45 i Matlab



Ordinära differentialekvationer (Eulers metod)

Alternativa härledningar av Eulers metod: Först Taylorutveckling:

$$y(t_k + h) = y(t_k) + hy'(t_k) + \frac{h^2}{2}y'(t_k) + \ldots$$

Nu är $y'(t_k) = f(t_k, y(t_k))$ och $t_{k+1} = t_k + h$ så att:

$$y(t_{k+1}) \approx y(t_k) + hf(t_k, y(t_k))$$

Vi approximerar nu $y_k \approx y(t_k)$, $y_{k+1} \approx y(t_{k+1})$ och får:

$$y_{k+1} = y_k + hf(t_k, y_k)$$

Härledning med hjälp av integration:

$$y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} y'(t)\,dt = \int_{t_k}^{t_{k+1}} f(t, y(t))\,dt$$

Vi approximerar nu integralen med arean av en rektangel;

$$y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} f(t, y(t))\,dt \approx \underbrace{(t_{k+1} - t_k)}_{h} f(t_k, y(t_k))$$

Så $y_{k+1} = y_k + hf(t_k, y_k)$.

ODE (system av ekvationer)

$$u''' = u'' - 2tu' + u^2 - t + 1 \quad \begin{cases} u(3) & = 2 \\ u'(3) & = -1 \\ u''(3) & = 0 \end{cases}$$

Inför nya funktioner

$$\begin{aligned} y_1 &= u \\ y_2 = u' &\Rightarrow y_2 = y_1' \\ y_3 = u'' &\Rightarrow y_3 = y_2' \end{aligned}$$

Vi får

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ y_3' = y_3 - 2ty_2 + y_1^2 - t + 1 \end{cases}, \quad \begin{cases} y_1(3) = 2 \\ y_2(3) = -1 \\ y_3(3) = 0 \end{cases}$$

Detta problem kan fortfarande skrivas $y' = f(t, y)$ om vi inför vektorerna y och f, dvs.

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix}$$

$$f(t, y) = \begin{bmatrix} y_2 \\ y_3 \\ y_3 - 2ty_2 + y_1^2 - t + 1 \end{bmatrix}, \quad y^{(0)} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}$$

Alla metoder vi har sett kan enkelt generaliseras till systemfallet. Skalära y_k byts ut mot vektorn $y^{(k)}$. $f(t_k, y_k)$ går över i $f(t_k, y^{(k)})$. Tiden t_k och steglängden h är fortfarande skalärer. Eulers metod för exemplet ovan blir, med $t_0 = 3$ och $h = 0.1$:

$$y^{(0)} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}, \quad y^{(1)} = y^{(0)} + hf(t_0, y^{(0)})$$

Dvs.

$$\begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_3^{(1)} \end{bmatrix} = \begin{bmatrix} y_1^{(0)} \\ y_2^{(0)} \\ y_3^{(0)} \end{bmatrix} + h \begin{bmatrix} y_2^{(0)} \\ y_3^{(0)} \\ y_3^{(0)} - 2t_0 y_2^{(0)} + \left(y_1^{(0)}\right)^2 - t_0 + 1 \end{bmatrix}$$

$$\begin{bmatrix} 1.9 \\ -1 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} + 0.1 \begin{bmatrix} -1 \\ 0 \\ 0 - 2 \cdot 3(-1) + 2^2 - 3 + 1 \end{bmatrix}$$

och så vidare.

Lösning av system av ekvationer med Matlabs ode45

För att lösa system av ekvationer

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ y_3' = y_3 - 2ty_2 + y_1^2 - t + 1 \end{cases}, \quad \begin{cases} y_1(3) = 2 \\ y_2(3) = -1 \\ y_3(3) = 0 \end{cases}$$

i tiden $t = [3, 10]$ vi kan också använda Matlabs ode45.

ODE (Matlabs ode45)

```
y0 = [2 -1 0]';   % begynnelsevarden
t0 = 3;           % begynnelsetid
ts = 10;          % slut-tid

[t, y] = ode45(@f, linspace(t0, ts, 100), y0);

figure(1); hold off
plot(t, y(:, 1), 'k-', t, y(:, 2), 'r-', ...
     t, y(:,3), 'b-')
legend({'y', 'y''', 'y'''''}, ...
       'Location', 'NorthWest','LineWidth',2)
xlabel('t')
grid on
title(' System av ODE: exempel')
```

ODE (Matlabs ode45)

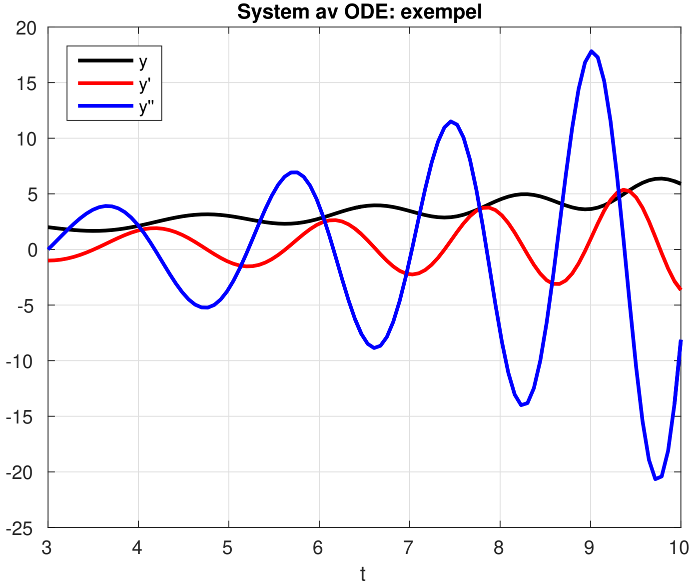
Vi måste definera separat matlabs-file f.m med funktion

```
function dy = f(t, y)

dy = zeros(3,1);

dy(1) = y(2);
dy(2) = y(3);
dy(3) = y(3)-2*t*y(2)+y(1)^2-t+1;
```

ODE (Matlabs ode45)



Övning

Sätt upp Eulers metod för problemet

$$y' = t + 2y, \;\; y(0) = 1$$

och beräkna $y_k, k = 0, 1, 2, 3$ med $h = 0.1$.
Eulers metod:

$$y_{k+1} = y_k + hf(t_k, y_k), y_0 = y(t_0).$$

Svar:
Eulers metod:

$$y_{k+1} = y_k + hf(t_k, y_k), y_0 = y(t_0).$$

I detta fall är

$$f(t, y) = t + 2y, f(t_k, y_k) = t_k + 2y_k, t_0 = 0, y(0) = 1, h = 0.1. \tag{11}$$
$$y_{k+1} = y_k + h(t_k + 2y_k), y_0 = 1. \tag{12}$$

så vi får följande approximationer:

$$y_0 = 1, \tag{13}$$
$$y_1 = y_0 + 0.1(t_0 + 2 \cdot y_0) = 1 + 0.1(0 + 2 \cdot 1) = 1.2, \tag{14}$$
$$y_2 = y_1 + 0.1(t_1 + 2 \cdot y_1) = 1.2 + 0.1(0.1 + 2 \cdot 1.2) = 1.45, \tag{15}$$
$$y_3 = y_2 + 0.1(t_2 + 2 \cdot y_2) = 1.45 + 0.1(0.2 + 2 \cdot 1.45) = 1.76. \tag{16}$$

Framåt Eulers metod i Matlab för $y' = t + 2y, \; y(0) = 1$

```
t0 = 0;          % begynnelsetid
ts = 2;           % slut-tid
h= 0.1;           %steglangd h
N = (ts - t0)/h  % antal punkter

  t = linspace(t0,ts,N);
  y = linspace(t0,ts,N);
  y(1) = 1;         % begynnelsevarden

for k = 1:N
    y(k+1) = y(k) + h*func_example4(t(k),y(k));
    t(k+1) = t(k) + h;

 end
 figure
plot(t, y, 'g -o ', 'LineWidth',2)
```

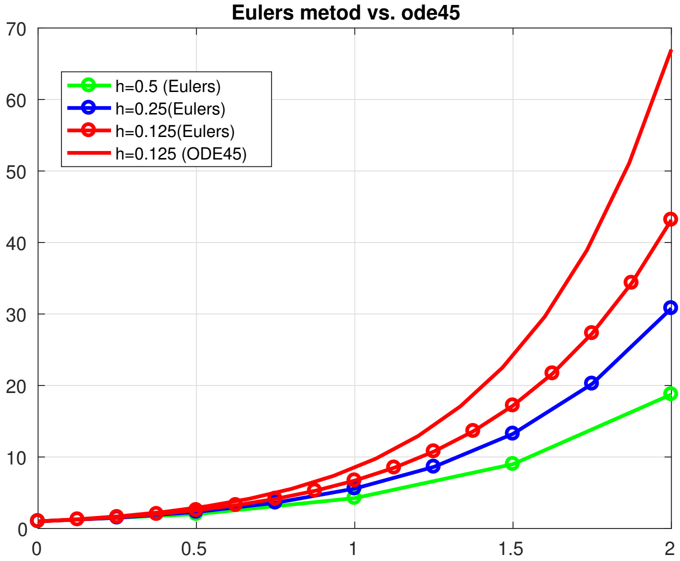
Framåt Eulers metod i Matlab för $y' = t + 2y, \; y(0) = 1$

Vi definerar separat matlabs-file func_example4.m med funktion

```
function dy = func_example4(t, y)

dy = 0;
dy = t + 2*y;
```

ODE: Eulers metod vs ode45 för $y' = t + 2y, \; y(0) = 1$



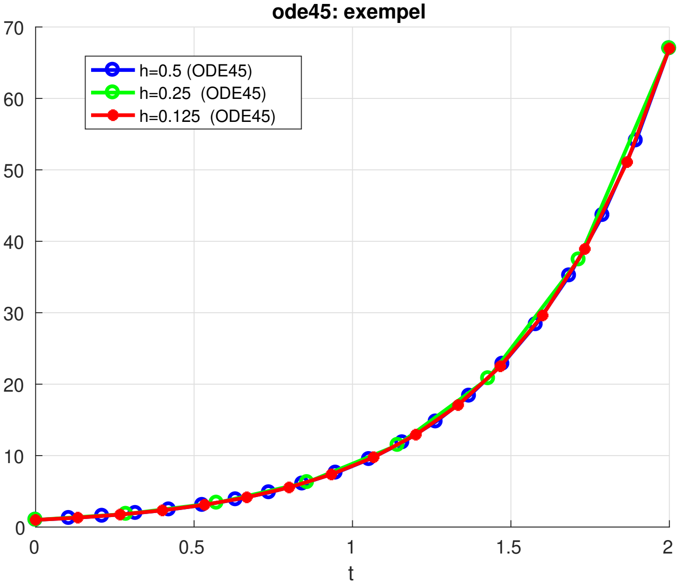
ode45 i Matlab för $y' = t + 2y, \; y(0) = 1$

```
y0 = 1;                % begynnelsevarden
t0 = 0;               % begynnelsetid
ts = 2;                % slut-tid

h= 0.1;                %steglangd h
N = (ts - t0)/h    %antal punkter

[t, y] = ode45(@func_example4, linspace(t0, ts, N), y0);

figure
hold on
plot(t, y(:, 1), 'b -o', 'LineWidth',2)
```


ode45: exempel
h=0.5 (ODE45)
h=0.25 (ODE45)
h=0.125 (ODE45)
70
60
50
40
30
20
10
0
0
0.5
1
1.5
2
t