2006–05–08

System av ODE $y' = A\,y,\ y(t_0) = y_0,\ A \in \mathbb{R}^{n\times n}$.

Ett problem är **styvt** om engenvärdena till A har olika storleksordning.

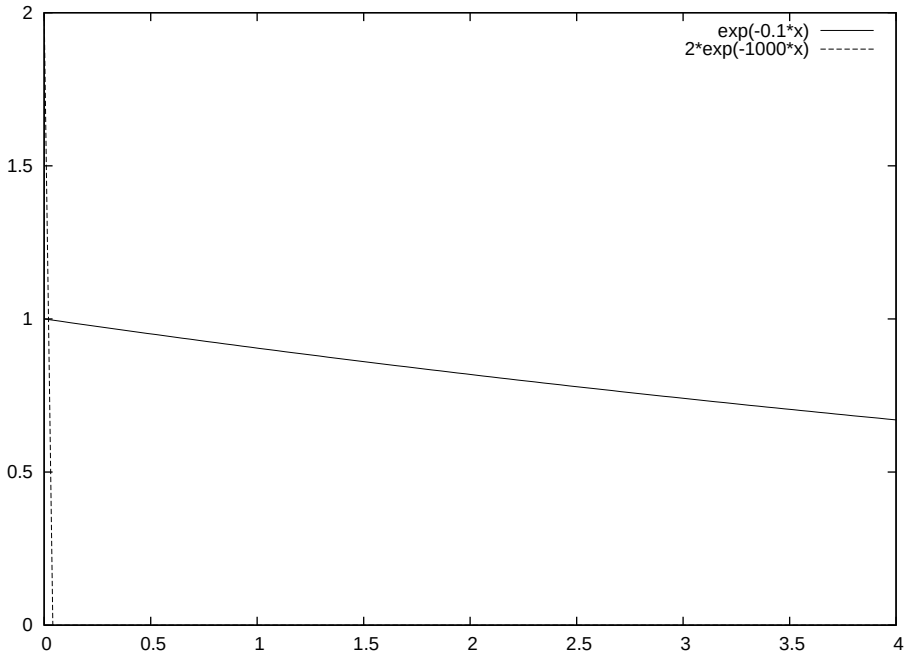
EXEMPEL:

$$A = \begin{pmatrix} -0.1 & 0 \\ 0 & -1000 \end{pmatrix}, \quad y_0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Lösning:

$$\begin{cases} y_1 = \mathrm{e}^{-0.1t} = \mathrm{e}^{\lambda_1 t} \\ y_2 = 2\,\mathrm{e}^{-1000t} = 2\,\mathrm{e}^{\lambda_2 t} \end{cases}$$

```
  This is a TeXmacs interface for GNUplot.
GNUplot] plot [ 0 to 4 ] exp(-0.1*x),2*exp(-1000*x)
```




```
GNUplot]
```

y_2 avklinggar snabbt, $y_2(0.03) \approx 10^{-13}$, dvs för $t \geqslant 0.03$ är vi inte intresserade av y_2, utan bara av y_1. Vi kan då välja en steglängd avpassad för bara y_1, dvs stor steglängd dunger. **Men:** För stabilitet hos en metod krävs att $h\,\lambda_j \in S$, där S är metodens stabilitetsområde.

$$S_{\text{Euler framåt}}\colon |1 + h\,\lambda| \leqslant 1$$

$\lambda_2 = -1000$ kräver då $h \leqslant 2 \cdot 10^{-3}$. Orimligt med så litet steg hela tiden. Efter $t = 0.03$ vill vi välja mycket längre steg.

För Euler bakåt eller trapetsmetoden har vi inga problem med detta eftersom de är A-stabila. Med dessa metoder väljer vi steglängd *endast* vilken noggrannhet vi vill ha.

MATLAB: `ode15s, ode23s`.

Interpolation Heath, kapitel 7

Problem: Givet n punkter i planet $(t_i, y_i), i = 1, ..., n$, med $t_1 < t_2 < \cdots < t_n$. Bestäm en funktion f så att $f(t_i) = y_i$. f kallas interpolant eller interpolerande funktion.

ANVÄNDNING: mjuka kurvor mellan punkter, approximera värden mellan punkter, approximera besvärliga funktioner med enklare. Grund för approximation av integraler. Polynomapproximation vid optimering.

Interpolanten f väljs som en enkel funktion, vanligen polynom eller styckvisa polynom (*splines*). f byggs upp med basfunktioner $\{\varphi_j\}_{j=1}^n$. Vi skriver alltså

$$f(t) = \sum_{j=1}^{n} x_j \varphi_j(t)$$

EXEMPEL: $\varphi_1 = 1$, $\varphi_2 = t$, $\varphi_3 = t^2$, ... så har vi en bas för polynom.

Interpolationskravet (att stämma överens i givna punkter) blir

$$f(t_i) = \sum_{j=1}^{n} x_j \varphi_j(t_i) = y_i, \quad i = 1, ..., n$$

Detta är ett linjärt ekvationssystem: $A\boldsymbol{x} = \boldsymbol{y}$ där $a_{i,j}$.

Polynominterpolation, `polyfit` i MATLAB.

Det finns ett entydigt polynom av grad $\leqslant n-1$ som går genom n givna punkter. Som bas kan vi ta $\varphi_j(t) = t^{j-1}$, $j = 1, ..., n$, men det finns andra lämpligare val.

ANMÄRKNING: Från tidigare övning i linjär algebra vet vi att denna bas tenderar att bli linjärt beroende då $n \to \infty$ med avseende på standardskalärprodukten i $\mathcal{C}^0[0, 1]$.

Med basen φ ovan får vi matrisen:

$$A = \begin{pmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & t_n^2 & \cdots & t_n^{n-1} \end{pmatrix}$$

A är illa konditionerad, $n = 10$ ger $\kappa = 10^8$.

Bättre bas praktiskt: Newtons bas:

$$\varphi_j(t) = \prod_{k=1}^{j-1} (t - t_k)$$

$$\varphi_1 = 1$$

$$\varphi_2 = (t - t_1)$$

$$\varphi_3 = (t - t_1)(t - t_2)$$

EXEMPEL. Approximera $y(2)$ genom kvadratisk interpolation då y är given genom tabellen:

t	1	3	4
y	3	7	11

Newtons form:

$$p_2(t) = x_1 + x_2 (t - 1) + x_3 (t - 1) (t - 3)$$

Interpolationsvillkoren $p_2(t_i) = y_i$ ger ekvationssystem:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 3 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 7 \\ 11 \end{pmatrix}$$

Framåtsubstitution:

$$\boldsymbol{x} = \begin{pmatrix} 3 \\ 2 \\ \frac{2}{3} \end{pmatrix}$$

dvs $p_2(t) = 3 + 2\,(t-1) + \frac{2}{3}\,(t-1)\,(t-3)$, med $y(2) \approx p_2(2) = 4\frac{1}{3}$. Om en punkt tillkommer är det enkelt att komplettera Newtons form. Ny punkt i exemplet:

t	1	2	3	4
y	3	5	7	11

Vi bygger ut $p_2(t)$ till ett polynom av grad 3:

$$p_3(t) = p_2(t) + x_4\,(t-1)\,(t-3)\,(t-4)$$

med $p_3(2) = p_2(2) + x_4\,(1)\,(-1)\,(-2) = 4\frac{1}{3} + 2\,x_4$. Villkor $p_3(2) = 5$ ger $x_4 = \frac{1}{3}$, och

$$p_3(t) = 3 + 2\,(t-1) + \frac{2}{3}\,(t-1)\,(t-3) + \frac{1}{3}\,(t-1)\,(t-3)\,(t-4)$$

Splines

DEFINITION: En **spline** är ett styckvis polynom av grad k med derivator av ordning $\leqslant k-1$ kontinuerliga i knutpunkterna (noderna).

|||.

Med $s_1, ..., s_5 \in \mathbb{P}_k$, i noden mellan s_2 och s_3 gäller $s_2^{(j)} = s_3^{(j)}$ för $j = 0, ..., k-1$, etc.

Vanliga splines: $k = 1$: linjär spline $s(t)$. (sick-sack mellan punkterna). Vi kan skriva

$$s(t) = \sum_{i=1}^{n} y_i\,\mathcal{B}_{i-1}(t)$$

där basfunktionerna är hattfunktioner: ⟨fig18⟩. Lokalt stöd.

$$\mathcal{B}_{i-1}(t_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

$k = 3$: kubiska spline: ⟨fig19⟩. Antal parametrar som bestämmer $s(t)$. Varje s_j har fyra parametrar. Antal $s_j = n-1$. Totalt antal parametrar $4(n-1)$. Antal villkor: 3 per inre nod, antal inre noder är $n-2$. Vi har alltså $n+2$ fria villkor kvar. Om vi använder splinen till interpolation, dvs skas överensstämma med givna värden i alla n punkterna. Vi har två fria villkor kvar. Man brukar ta ändpunktsvillkor:

- naturliga: $s''(t_1) = s''(t_n) = 0$.
- periodiska: $s'(t_1) = s'(t_n), s''(t_1) = s''(t_n)$.
- rätta: $s'(t_1) = y'(t_1), s'(t_n) = y'(t_n)$.