Solving ODE's with the theta or the trapezoidal rule

The theta method uses a parameter 0 < θ < 1 for solve ODE's. Special values of θ are:

θ = 0: Backward Euler
θ = 1/2: Trapezoid method
θ = 1: Forward Euler

DSolRes

θ rule + Newton Raphson ODE Solver

```
DSolRes(Ds, toe, xi, θ, N) :=
m := length(xi)  r := [1..m]  c := r  hId := eval(h·identity(m))
X := xi_r  T := eval([min(toe)])  dt := eval((max(toe) − T_1)/N)  J := 0
for k ∈ [1..N]
  to := T_k  xo := col(X, k)
  t := to + dt  x := xo + dt·D(to, xo)
  res(x) := x − xo − (t − to)·(θ·feval(Ds, to, xo) + (1 − θ)·feval(Ds, t, x))
  R := res(x)
  for iter ∈ [1..MaxI]
    if normi(R) ≤ εy
      break
    else
      J_{r c} := eval((1/h)·(res(x + col(hId, c))_r − R_r))
      Δx := eval(−J^{−1}·R)
      if normi(Δx) ≤ εx
        break
      else
        x := eval(x + Δx)  R := eval(res(x))
  T_{k+1} := t  X_{c k+1} := x_c
augment(T, X^T)
```

The same, but using alg lib solver: faster.

θ rule + Newton Raphson ODE Solver

```
DSolRes(Ds, toe, xi, θ, N) :=
m := length(xi)  r := [1..m]  c := r  hId := eval(h·identity(m))
X := xi_r  T := eval([min(toe)])  dt := eval((max(toe) − T_1)/N)
for k ∈ [1..N]
  to := T_k  xo := col(X, k)
  t := to + dt  x := xo + dt·D(to, xo)
  res(x) := x − xo − (t − to)·(θ·feval(Ds, to, xo) + (1 − θ)·feval(Ds, t, x))
  T_{k+1} := t  X_{c k+1} := al_nleqsolve(x, res)_c
augment(T, X^T)
```

Utils

$$feval(f\#, x\#) := \text{str2num}(\text{concat}(\text{num2str}(f\#), "(", \text{num2str}(x\#), ")"))$$

$$feval(f\#, x\#, y\#) := \text{str2num}(\text{concat}(\text{num2str}(f\#), "(", \text{num2str}(x\#), ",", \text{num2str}(y\#), ")"))$$

$$MCols(M) := \begin{vmatrix} C := 0 \\ \text{for } k \in [1..\text{cols}(M)] \\ \quad C_{1\,k} := \text{col}(M, k) \\ C \end{vmatrix}$$

Defaults

$h := 10^{-12}$ $\quad \varepsilon y := 10^{-14}$ $\quad \varepsilon x := 10^{-14}$

$0 := 10^{-7}$ $\quad MaxI := 100$

DSolRes Example

Example: Exact linear ODE $\quad \boxed{x'' = A' \cdot x + A \cdot x' + B'}$

Given $\quad to := 0 \quad te := 3 \quad xo := [0\ 1] \quad N := 15$

$A(t) := 0.1 \cdot t \quad B(t) := e^{-0.2 \cdot t} \cdot t$

let

$$A'(t) := \frac{d}{d\,t} A(t) = \frac{1}{10} \qquad IA(t) := \int_{to}^{t} A(\tau 2)\, d\,\tau 2$$

$$B'(t) := \frac{d}{d\,t} B(t) = \frac{5 - t}{5 \cdot \sqrt[5]{e}^{\,t}}$$

Convert the ODE into a system

$$D(t, x) := \begin{bmatrix} x_2 \\ A'(t) \cdot x_1 + A(t) \cdot x_2 + B'(t) \end{bmatrix}$$

Exact solution, numerically evaluated

$$x_s := \int_{to}^{t} (1 - B(0) + B(\tau)) \cdot e^{-IA(\tau)}\, d\,\tau \cdot e^{IA(t)} \qquad x_s(t) := x_s$$

The numerical evaluation for the integrals could be very slow. Enable this for speed up the calculations, only if Maple can find a close solution and SMath recognize all functions.

$$x_s(t) := feval("maple", x_s) = \sqrt{5} \cdot \left(2 \cdot \left(\sqrt{5} \cdot \left(1 - \exp\left(-\frac{t \cdot (4 + t)}{20}\right)\right) + \sqrt{\pi} \cdot \left(-\text{erf}\left(\frac{\sqrt{5} \cdot (2 + t)}{10}\right) + \text{erf}\left(\frac{1}{\sqrt{5}}\right)\right) \cdot \exp\left(\frac{1}{5}\right)\right) + \sqrt{\pi} \cdot$$

Numerical solutions $\quad xoT := xo^{T}$

Method a: Trapezoidal rule $\quad [T\ Xa\ Xa'] := MCols(DSolRes("D", [to\ te], xo, 0.5, N))$

Method b: Backward Euler $\quad [Tb\ Xb\ Xb'] := MCols(DSolRes("D", [to\ te], xo, 1 - 0, N))$

Method c: Forward Euler $\quad [Tc\ Xc\ Xc'] := MCols(DSolRes("D", [to\ te], xo, 0, N))$

Method d: rkfixed from uni $\quad [Td\ Xd\ Xd'] := MCols(\text{rkfixed}(xoT, to, te, N, D))$

Method e: Rkadapt from uni $\quad [Te\ Xe\ Xe'] := MCols(\text{Rkadapt}(xoT, to, te, N, D))$

(T's are the same) $\text{normi}\left(\left[\text{normi}\left(T - Td\right)\ \text{normi}\left(T - Te\right)\right]\right) = 0$

Compare with the symbolic solution $k := [1..(N+1)]$ As exècted, Rkadapt it's better.

$Xs_k := x_s\left(T_k\right)$

$Xs'_k := \dfrac{x_s\left(T_k + h\right) - Xs_k}{h}$

$\text{normi}\left(Xa - Xs\right) = 1.42 \cdot 10^{-2}$

$\text{normi}\left(Xb - Xs\right) = 5.33 \cdot 10^{-1}$

$\text{normi}\left(Xc - Xs\right) = 6.4 \cdot 10^{-1}$

$\text{normi}\left(Xd - Xs\right) = 6.19 \cdot 10^{-2}$

$\text{normi}\left(Xe - Xs\right) = 1.67 \cdot 10^{-10}$

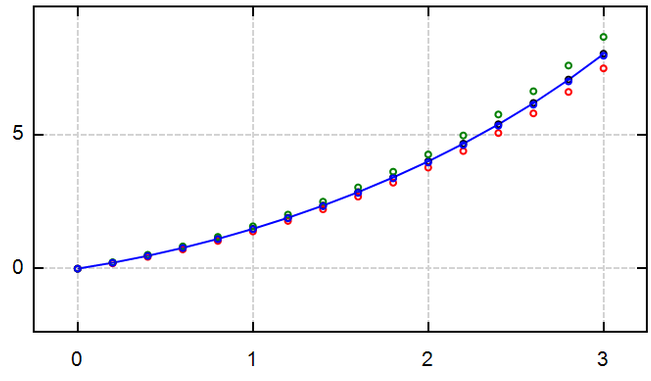
$\text{normi}\left(Xa' - Xs'\right) = 3.38 \cdot 10^{-2}$

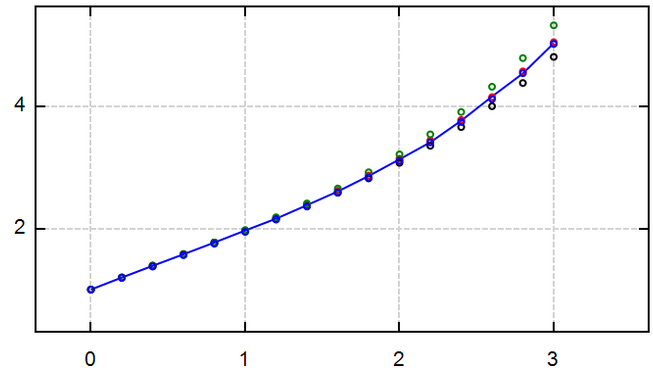
$\text{normi}\left(Xb' - Xs'\right) = 2.22 \cdot 10^{-1}$

$\text{normi}\left(Xc' - Xs'\right) = 2.96 \cdot 10^{-1}$

$\text{normi}\left(Xd' - Xs'\right) = 4.04 \cdot 10^{-2}$

$\text{normi}\left(Xe' - Xs'\right) = 2.66 \cdot 10^{-2}$






```
augment(T, Xs)
augment(T, Xa, "o", 3, "black")
augment(T, Xb, "o", 3, "red")
augment(T, Xc, "o", 3, "green")
augment(T, Xd, "o", 3, "blue")
```

```
augment(T, Xs')
augment(T, Xa', "o", 3, "red")
augment(T, Xb', "o", 3, "black")
augment(T, Xc', "o", 3, "green")
augment(T, Xd', "o", 3, "blue")
```

DSolRes Example

Example: How to build an example

Obviously, start from the solution: Thus $n := [1..4]$

$$S(t) := \begin{bmatrix} t \cdot \sin\left(t^2\right) + 2 \\ \ln(1+t) - \ln(4) \\ e^{-t} \\ \cos\left(t^2\right) \end{bmatrix}$$

$$D(t, x) := \left(dx_n := \frac{d}{dt} S(t)_n\right) = \begin{bmatrix} 2 \cdot t^2 \cdot \cos\left(t^2\right) + \sin\left(t^2\right) \\ \frac{1}{1+t} \\ -\frac{1}{e^t} \\ -2 \cdot t \cdot \sin\left(t^2\right) \end{bmatrix}$$

You can use this D for, but just complicate the things

$$D(t, x) := \begin{bmatrix} 2 \cdot t^2 \cdot x_4 - \dfrac{x_1 - 2}{\ln\left(x_3\right)} \\ \dfrac{1}{4 \cdot e^{x_2}} \\ -x_3 \\ -2 \cdot \left(x_1 - 2\right) \end{bmatrix}$$

Try to disable this for check that both D's are the same.

Given $to := 0$ $te := 2$ $N := 25$ Avoid t = 0

Now initials conditios are just $xo := S(to)^{T} = [\,2 \;\; -1.3863 \;\; 1 \;\; 1\,]$ $xoT := xo^{T}$

Numerical solutions

Method a: Trapezoidal rule $[\,T \;\; Xa \;\; Ya \;\; Za \;\; Ua\,] := MCols(DSolRes("D", [\,to \;\; te\,], xo, 0.5, N))$

Method b: Backward Euler $[\,Tb \;\; Xb \;\; Yb \;\; Zb \;\; Ub\,] := MCols(DSolRes("D", [\,to \;\; te\,], xo, 1 - 0, N))$

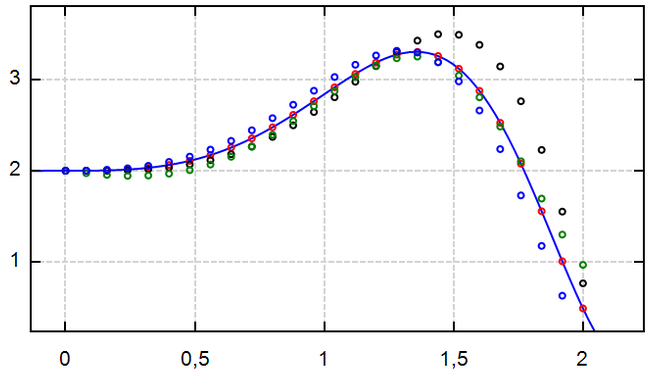
Method c: Forward Euler $[\,Tc \;\; Xc \;\; Yc \;\; Zc \;\; Uc\,] := MCols(DSolRes("D", [\,to \;\; te\,], xo, 0, N))$

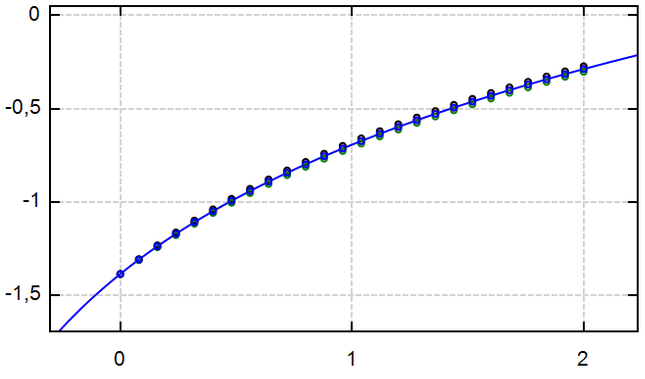
Method d: rkfixed from uni $[\,Td \;\; Xd \;\; Yd \;\; Zd \;\; Ud\,] := MCols(\mathrm{rkfixed}(xoT, to, te, N, D))$

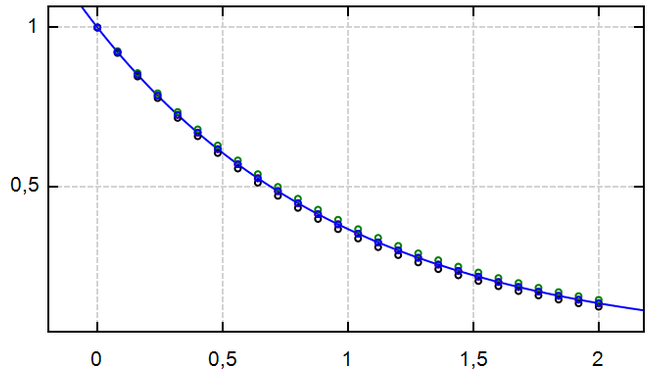
Method e: Rkadapt from uni $[\,Te \;\; Xa \;\; Ya \;\; Za \;\; Ua\,] := MCols(\mathrm{Rkadapt}(xoT, to, te, N, D))$

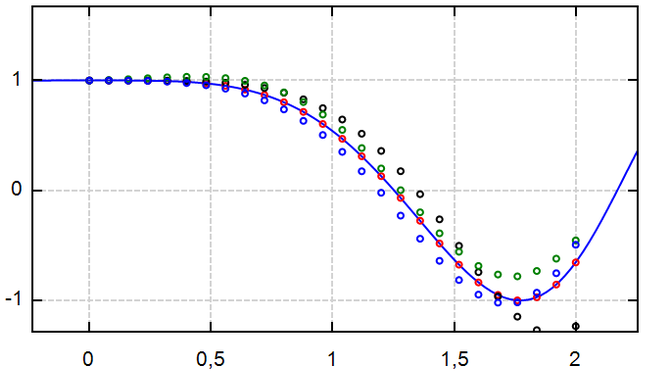
Compare with the (given) symbolic solution S(t)

Hard to decide wich's better.









Alvaro