Octave Conference 2017

March 21, 2017

GSoC 2016: Exponential Integrators

Chiara Segala

Mentor: **Prof. Marco Caliari**

Outline

- Exponential Integrators
- Matrix functions
- Implementation of solvers
- Numerical experiments

Exponential Integrators

Parabolic problem

$$u'(t) + Au(t) = f(t), \quad u(t_0) = u_0$$

Exponential Integrators

Parabolic problem

$$u'(t) + Au(t) = f(t), \quad u(t_0) = u_0$$

- Solution (variation of constants formula)

$$u(t_{n+1}) = e^{-h_n A} u(t_n) + \int_0^{h_n} e^{-(h_n - \tau)A} f(t_n + \tau) d\tau$$

Exponential Integrators

Parabolic problem

$$u'(t) + Au(t) = f(t), \quad u(t_0) = u_0$$

- Solution (variation of constants formula)

$$u(t_{n+1}) = e^{-h_n A} u(t_n) + \int_0^{h_n} e^{-(h_n - \tau)A} f(t_n + \tau) d\tau$$

- First numerical scheme (exponential quadrature rule)

$$u_{n+1} = e^{-h_n A} u_n + h_n \sum_{i=1}^{s} b_i(-h_n A) f(t_n + c_i h_n)$$

weights $b_i(z)$ can be expressed as linear combinations of the functions

$$\varphi_k(z) = \int_0^1 e^{(1-\theta)z} \frac{\theta^{k-1}}{(k-1)!} d\theta, \quad k \geq 1.$$

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}, \quad \varphi_k(0) = \frac{1}{k!}, \quad \varphi_0(z) = e^z$$

Example

$s = 1$

$$
\begin{aligned}
u_{n+1} &= e^{-h_n A} u_n + h_n \varphi_1(-h_n A) f(t_n + c_1 h_n) \\
&= u_n + h_n \varphi_1(-h_n A)(f(t_n + c_1 h_n) - A u_n)
\end{aligned}
$$

Example

$s = 1$

$$u_{n+1} = e^{-h_n A} u_n + h_n \varphi_1(-h_n A) f(t_n + c_1 h_n)$$
$$= u_n + h_n \varphi_1(-h_n A)(f(t_n + c_1 h_n) - A u_n)$$

- Exponential Euler method for $c_1 = 0$
  Butcher tableau

$$\begin{array}{c|c} 0 & \\ \hline & \varphi_1 \end{array}$$

- Exponential midpoint rule for $c_1 = \frac{1}{2}$

Exponential Runge-Kutta methods

Semilinear problem

$$u'(t) = F(t,u) = Au(t) + g(t,u(t)), \quad u(t_0) = u_0$$

Exponential Runge-Kutta methods

Semilinear problem

$$u'(t) = F(t,u) = Au(t) + g(t,u(t)), \quad u(t_0) = u_0$$

Numerical Exponential Runge-Kutta scheme

$$u_{n+1} = u_n + h_n \sum_{i=1}^{s} b_i(h_n A)(G_{ni} + Au_n)$$

$$U_{ni} = u_n + h_n \sum_{j=1}^{s} a_{ij}(h_n A)(G_{nj} + Au_n)$$

$$G_{nj} = g(t_n + c_j h_n, U_{nj})$$

Exponential Runge-Kutta methods

Semilinear problem

$$u'(t) = F(t,u) = Au(t) + g(t,u(t)), \quad u(t_0) = u_0$$

Numerical Exponential Runge-Kutta scheme

$$u_{n+1} = u_n + h_n \sum_{i=1}^{s} b_i(h_n A)(G_{ni} + Au_n)$$

$$U_{ni} = u_n + h_n \sum_{j=1}^{s} a_{ij}(h_n A)(G_{nj} + Au_n)$$

$$G_{nj} = g(t_n + c_j h_n, U_{nj})$$

Reformulation

$$u_{n+1} = u_n + h_n \varphi_1(h_n A) F(t_n, u_n) + h_n \sum_{i=2}^{s} b_i(h_n A) D_{ni}$$

$$U_{ni} = u_n + h_n c_i \varphi_1(c_i h_n A) F(t_n, u_n) + h_n \sum_{j=2}^{i-1} a_{ij}(h_n A) D_{nj}$$

$$D_{nj} = g(t_n + c_j h_n, U_{nj}) - g(t_n, u_n)$$

Exponential Rosenbrock methods

$$u'(t) = F(t, u), \quad u(t_0) = u_0$$

Exponential Rosenbrock methods

$$u'(t) = F(t, u), \quad u(t_0) = u_0$$

Numerical Exponential Rosenbrock scheme

$$u_{n+1} = u_n + h_n \varphi_1(h_n J_n) F(t_n, u_n) + h_n^2 \varphi_2(h_n J_n) v_n + h_n \sum_{i=2}^{s} b_i(h_n J_n) D_{ni}$$

$$U_{ni} = u_n + h_n c_i \varphi_1(c_i h_n J_n) F(t_n, u_n) + h_n^2 c_i^2 \varphi_2(c_i h_n J_n) v_n + h_n \sum_{j=2}^{i-1} a_{ij}(h_n J_n) D_{nj}$$

Exponential Rosenbrock methods

$$u'(t) = F(t, u), \quad u(t_0) = u_0$$

Numerical Exponential Rosenbrock scheme

$$u_{n+1} = u_n + h_n \varphi_1(h_n J_n) F(t_n, u_n) + h_n^2 \varphi_2(h_n J_n) v_n + h_n \sum_{i=2}^{s} b_i(h_n J_n) D_{ni}$$

$$U_{ni} = u_n + h_n c_i \varphi_1(c_i h_n J_n) F(t_n, u_n) + h_n^2 c_i^2 \varphi_2(c_i h_n J_n) v_n + h_n \sum_{j=2}^{i-1} a_{ij}(h_n J_n) D_{nj}$$

where

$$J_n = \frac{\partial F}{\partial u}(t_n, u_n)$$

$$v_n = \frac{\partial F}{\partial t}(t_n, u_n)$$

$$g_n(t, u) = F(t, u) - J_n u - v_n t$$

$$D_{nj} = g_n(t_n + c_j h_n, U_{nj}) - g_n(t_n, u_n)$$

φ-functions

$$\varphi_k(z) = \int_0^1 e^{(1-\theta)z} \frac{\theta^{k-1}}{(k-1)!} d\theta, \quad k \geq 1.$$

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}, \quad \varphi_k(0) = \frac{1}{k!}, \quad \varphi_0(z) = e^z$$

φ-functions

$$\varphi_k(z) = \int_0^1 e^{(1-\theta)z} \frac{\theta^{k-1}}{(k-1)!} d\theta, \quad k \geq 1.$$

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}, \quad \varphi_k(0) = \frac{1}{k!}, \quad \varphi_0(z) = e^z$$

φ-functions code: scaling and squaring + Padé approximation

Accuracy

$\varphi_k(\pm 10)$	`phikm(±10)`	`phipade(±10,k)`
$\varphi_1(10)$	2202.54657948068	2202.54657962449
$\varphi_2(10)$	220.154657948068	220.154657962442
$\varphi_3(10)$	21.9654657948068	21.9654657961828
$\varphi_4(10)$	2.17987991281401	2.17987991294347
$\varphi_1(-10)$	0.0999954600070237	0.0999954600070233
$\varphi_2(-10)$	0.0900004539992977	0.0900004539992175
$\varphi_3(-10)$	0.0409999546000702	0.0409999546000398
$\varphi_4(-10)$	0.0125666712066596	0.0125666712066530

Augmented matrix

Theorem *(see Al-Mohy and Higham, 2011, Th. 2.1)*

Let $A \in \mathbb{C}^{n\times n}$, $W = [w_1, w_2, \ldots, w_p] \in \mathbb{C}^{n\times p}$, $\tau \in \mathbb{C}$, and

$$\widetilde{A} = \begin{bmatrix} A & W \\ 0 & J \end{bmatrix} \in \mathbb{C}^{(n+p)\times(n+p)}, \quad J = \begin{bmatrix} 0 & I_{p-1} \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{p\times p},$$

Then for $X = \varphi_l(\tau\widetilde{A})$ with $l \geq 0$ we have

$$X(1:n, n+j) = \sum_{k=1}^{j} \tau^k \varphi_{l+k}(\tau A) w_{j-k+1}, \quad j = 1:p.$$

Augmented matrix for exponential integrators

- $W(:, p-k+1) = u_k, \quad k = 1:p, \quad l = 0, \quad \tau = t - t_0$

$$X = \varphi_0((t-t_0)\widetilde{A}) = e^{(t-t_0)\widetilde{A}} = \begin{bmatrix} e^{(t-t_0)A} & X_{12} \\ 0 & e^{(t-t_0)J} \end{bmatrix}$$

$$X(1:n, n+p) = \sum_{k=1}^{p} \varphi_k((t-t_0)A)(t-t_0)^k u_k$$

Augmented matrix for exponential integrators

- $W(:, p-k+1) = u_k, \quad k = 1:p, \quad l = 0, \quad \tau = t - t_0$

$$X = \varphi_0((t-t_0)\widetilde{A}) = e^{(t-t_0)\widetilde{A}} = \begin{bmatrix} e^{(t-t_0)A} & X_{12} \\ 0 & e^{(t-t_0)J} \end{bmatrix}$$

$$X(1:n, n+p) = \sum_{k=1}^{p} \varphi_k((t-t_0)A)(t-t_0)^k u_k$$

-

$$\begin{aligned} \hat{u}(t) &= e^{(t-t_0)A} u_0 + \sum_{k=1}^{p} \varphi_k((t-t_0)A)(t-t_0)^k u_k \\ &= e^{(t-t_0)A} u_0 + X(1:n, n+p) \\ &= \begin{bmatrix} I_n & 0 \end{bmatrix} e^{(t-t_0)\widetilde{A}} \begin{bmatrix} u_0 \\ e_p \end{bmatrix} \end{aligned}$$

Augmented matrix for exponential integrators

- $W(:, p-k+1) = u_k, \quad k = 1:p, \quad l = 0, \quad \tau = t - t_0$

$$X = \varphi_0((t-t_0)\widetilde{A}) = e^{(t-t_0)\widetilde{A}} = \begin{bmatrix} e^{(t-t_0)A} & X_{12} \\ 0 & e^{(t-t_0)J} \end{bmatrix}$$

$$X(1:n, n+p) = \sum_{k=1}^{p} \varphi_k((t-t_0)A)(t-t_0)^k u_k$$

-

$$\begin{aligned} \hat{u}(t) &= e^{(t-t_0)A} u_0 + \sum_{k=1}^{p} \varphi_k((t-t_0)A)(t-t_0)^k u_k \\ &= e^{(t-t_0)A} u_0 + X(1:n, n+p) \\ &= [I_n \quad 0]\, e^{(t-t_0)\widetilde{A}} \begin{bmatrix} u_0 \\ e_p \end{bmatrix} \end{aligned}$$

-

$$\hat{u}(t) = [I_n \quad 0]\, exp\left((t-t_0)\begin{bmatrix} A & \eta W \\ 0 & J \end{bmatrix}\right)\begin{bmatrix} u_0 \\ \eta^{-1} e_p \end{bmatrix}, \quad \eta = 2^{-\lceil log_2(\|W\|_1) \rceil}$$

Double augmented matrix

Theorem (Double augmented matrix)

Let $A \in \mathbb{C}^{n\times n}$, $W = [w_1, w_2, \ldots, w_p] \in \mathbb{C}^{n\times p}$,
$V = [v_1, v_2, \ldots, v_q] \in \mathbb{C}^{n\times q}$, $\tau \in \mathbb{C}$, and

$$\tilde{\tilde{A}} = \begin{bmatrix} A & W & V \\ 0 & J & 0 \\ 0 & 0 & K \end{bmatrix} \in \mathbb{C}^{(n+p+q)\times(n+p+q)}, \quad J \in \mathbb{C}^{p\times p}, \quad K \in \mathbb{C}^{q\times q}.$$

Then for $X = \varphi_l(\tau \tilde{\tilde{A}})$ with $l \geq 0$ we have

$$X(1:n, n+j) = \sum_{k=1}^{j} \tau^k \varphi_{l+k}(\tau A) w_{j-k+1}, \quad j = 1:p.$$

$$X(1:n, n+i) = \sum_{k=1}^{i-p} \tau^k \varphi_{l+k}(\tau A) v_{i-k+1-p}, \quad i = (p+1):(p+q).$$

expmv

$$e^A B \approx (T_m(s^{-1}A))^s B$$

$m \implies$ `select_taylor_degree`

expmv

$$e^A B \approx (T_m(s^{-1}A))^s B$$

$m \Longrightarrow$ `select_taylor_degree`

- `function [f] = expmv (A, b)`

```
...
M = select_taylor_degree (A, b);
...
```

expmv

$$e^A B \approx (T_m(s^{-1}A))^s B$$

$m \implies$ `select_taylor_degree`

- ```
  function [f] = expmv (A, b)
                  ...
                  M = select_taylor_degree (A, b);
                  ...
  ```
- ```
  M = select_taylor_degree (A, b);
  f = expmv (A, b, M);
  ```

expmv

$$e^{A}B \approx (T_m(s^{-1}A))^{s}B$$

$m \Longrightarrow$ `select_taylor_degree`

- ```
  function [f] = expmv (A, b)
                ...
                M = select_taylor_degree (A, b);
                ...
  ```
- ```
  M = select_taylor_degree (A, b);
  f = expmv (A, b, M);
  ```

`select_taylor_degree` $\Longrightarrow \|A^{k}\|_1$

Estimate

Theorem (Estimate)

Let $\widetilde{A} = \begin{bmatrix} A & W \\ 0 & J \end{bmatrix} \in \mathbb{C}^{(n+p)\times(n+p)}$ be the matrix of Higham Theorem. Suppose that $\|W\|_1 = \gamma \leq 1$. Then we have the following estimate

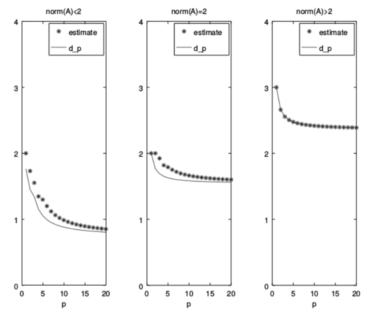
$$\|\widetilde{A}^k\|_1 \leq \max\Big\{\|A^k\|_1, \gamma \cdot \textstyle\sum_{j=k-\min\{k,p\}}^{k-1} \|A^j\|_1 + \max\{\min\{p-k,1\},0\}\Big\}.$$

Estimate

Theorem (Estimate)

Let $\widetilde{A} = \begin{bmatrix} A & W \\ 0 & J \end{bmatrix} \in \mathbb{C}^{(n+p)\times(n+p)}$ be the matrix of Higham Theorem. Suppose that $\|W\|_1 = \gamma \leq 1$. Then we have the following estimate

$$\|\widetilde{A}^k\|_1 \leq \max\Big\{\|A^k\|_1, \gamma \cdot \textstyle\sum_{j=k-\min\{k,p\}}^{k-1} \|A^j\|_1 + \max\{\min\{p-k,1\},0\}\Big\}.$$



odexprk23

$$
\begin{array}{c|ccc}
0 & & & \\
c_2 & c_2\varphi_{1,2} & & \\
\frac{2}{3} & \frac{2}{3}\varphi_{1,3} - \frac{8}{9}\varphi_{2,3} & \frac{8}{9}\varphi_{2,3} & \\
\hline
 & (1 - \frac{1}{2c_2})\varphi_1 & \frac{1}{2c_2}\varphi_1 & \\
 & \varphi_1 - \frac{3}{2}\varphi_2 & 0 & \frac{3}{2}\varphi_2
\end{array}
$$

$\varphi_{j,k} = \varphi_j(c_k hA)$ and $\varphi_j = \varphi_j(hA)$

odexprk23

$$\begin{array}{c|ccc}
0 & & & \\
c_2 & c_2\varphi_{1,2} & & \\
\frac{2}{3} & \frac{2}{3}\varphi_{1,3} - \frac{8}{9}\varphi_{2,3} & \frac{8}{9}\varphi_{2,3} & \\
\hline
 & (1 - \frac{1}{2c_2})\varphi_1 & \frac{1}{2c_2}\varphi_1 & \\
 & \varphi_1 - \frac{3}{2}\varphi_2 & 0 & \frac{3}{2}\varphi_2
\end{array}$$

$\varphi_{j,k} = \varphi_j(c_k hA)$ and $\varphi_j = \varphi_j(hA)$

$$U_{n1} = u_n$$

$$U_{n2} = u_n + \frac{1}{2}h_n\varphi_1(\frac{1}{2}h_nA)F(t_n, u_n)$$

$$U_{n3} = u_n + \frac{2}{3}h_n\varphi_1(\frac{2}{3}h_nA)F(t_n, u_n) + \frac{2}{3}h_n\varphi_2(\frac{2}{3}h_nA)\frac{2}{3}(2)D_{n2}$$

$$u_{n+1} = u_n + h_n\varphi_1(h_nA)F(t_n, u_n) + h_n\varphi_2(h_nA)\frac{3}{2}D_{n3}$$

$$\hat{u}_{n+1} = u_n + h_n\varphi_1(h_nA)[F(t_n, u_n) + \frac{1}{2}(2)D_{n2}]$$

exp_runge_kutta_23

```
function [t_next, x_next, x_est, k] = exp_runge_kutta_23 (f, t, x, dt,
                                                          options,
                                                          k_vals = [],
                                                          t_next = t + dt)
x = x.';
M = options.GetInfo;
myexpmv = @(t,A,v) expmv (t, A, v, M); % myexpmv = @(t,A,v) expmv (t, A, v); myexpmv = @(t,A,v) expm (t*A)*v;
A = options.LinOp;
g = options.GFun;
d = size (A, 1);
c2 = 1/2;
k = zeros (length (x), 3);
Fn = feval (f, t, x(:));
gn = feval (g, t, x(:));
[Atilde, eta] = augmatrix ( A, Fn);
X = feval(myexpmv, c2*dt, Atilde, [zeros(size (Atilde)-1, 1); 1/eta])(1:d, :);
U2 = x(:) + X;
g2 = feval (g, t + c2*dt, U2);
D2 = g2 - gn;
[Atilde, eta] = augmatrix ( A, [Fn, (2/(3*c2)*D2)./(2/3*dt)]);
X = feval(myexpmv, 2/3*dt, Atilde, [zeros(size (Atilde)-1, 1); 1/eta])(1:d, :);
U3 = x(:) + X;
g3 = feval (g, t + 2/3*dt, U3);
D3 = g3 - gn;
[Atilde, eta] = augmatrix ( A, [Fn, (3/2*D3)./dt], [Fn + 1/(2*c2)*D2]);
ep = [[zeros(1, 1); 1/eta; 0], [zeros(2, 1); 1/eta]];
X = feval(myexpmv, dt, Atilde, [zeros(d, 1), zeros(d, 1); ep])(1:d, :);
x_next = x(:) + X(:,1); x_est = x(:) + X(:,2);
k(:, 1) = Fn;   k(:, 2) = A*U2 + g2;   k(:, 3) = A*U3 + g3;
```

odexprb34

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2}\varphi_1(\frac{1}{2}\cdot) & & \\
1 & 0 & \varphi_1 & \\
\hline
 & \varphi_1 - 14\varphi_3 + 36\varphi_4 & 16\varphi_3 - 48\varphi_4 & -2\varphi_3 + 12\varphi_4 \\
 & \varphi_1 - 14\varphi_3 & 16\varphi_3 & -2\varphi_3
\end{array}
$$

$$\varphi_j = \varphi_j(hJ_n)$$

odexprb34

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2}\varphi_1(\frac{1}{2}\cdot) & & \\
1 & 0 & \varphi_1 & \\
\hline
 & \varphi_1 - 14\varphi_3 + 36\varphi_4 & 16\varphi_3 - 48\varphi_4 & -2\varphi_3 + 12\varphi_4 \\
 & \varphi_1 - 14\varphi_3 & 16\varphi_3 & -2\varphi_3
\end{array}
$$

$$\varphi_j = \varphi_j(hJ_n)$$

$$
\begin{aligned}
U_{n1} &= u_n \\
U_{n2} &= u_n + \frac{1}{2}h_n\varphi_1(\frac{1}{2}h_nJ_n)F(t_n, u_n) + \frac{1}{2}h_n\varphi_2(\frac{1}{2}h_nJ_n)\frac{1}{2}h_nv_n \\
U_{n3} &= u_n + h_n\varphi_1(h_nJ_n)[F(t_n, u_n) + D_{n2}] + h_n\varphi_2(h_nJ_n)h_nv_n \\
u_{n+1} &= u_n + h_n\varphi_1(h_nJ_n)F(t_n, u_n) + h_n\varphi_2(h_nJ_n)h_nv_n \\
&\quad + h_n\varphi_3(h_nJ_n)[16D_{n2} - 2D_{n3}] + h_n\varphi_4(h_nJ_n)[-48D_{n2} + 12D_{n3}] \\
\hat{u}_{n+1} &= u_n + h_n\varphi_1(h_nJ_n)F(t_n, u_n) + h_n\varphi_2(h_nJ_n)h_nv_n + h_n\varphi_3(h_nJ_n)[16D_{n2} - 2D_{n3}]
\end{aligned}
$$

exp_rosenbrock_34

```
function [t_next, x_next, x_est, k] = exp_rosenbrock_34 (f, t, x, dt,
                                                          options,
                                                          k_vals = [],
                                                          t_next = t + dt)
x = x.';
k = zeros (length (x), 4);
if (isempty (options.Jacobian))
  fu = @(xx) f(t,xx);
  Jn = numjac ( x(:), fu);
else
  J = options.Jacobian;
  Jn = feval (J, t, x(:));
end
M = select_taylor_degree_aug (Jn, [], 2, [], [], [], false, false, true);
myexpmv = @(t,A,v) expmv (t, A, v, M);
d = size (Jn, 1);
Fn = feval (f, t, x(:));
if (isempty (options.DFdt))
  ft = @(tt) f(tt, x(:));
  vn = numjac ( t, ft);
else
  V = options.DFdt;
  vn = feval (V, t, x(:));
end
g = @(t,x) f(t,x) - Jn*x - vn*t;
gn = feval (g, t, x(:));
[Atilde, eta] = augmatrix ( Jn, [Fn, 1/2*vn]);
X = feval(myexpmv, 1/2*dt, Atilde, [zeros(size (Atilde)-1, 1); 1/eta])(1:d, :);
U2 = x(:) + X;   g2 = feval (g, t + 1/2*dt, U2);   D2 = g2 - gn;
[Atilde, eta] = augmatrix ( Jn, [Fn+D2, vn]);
X = feval(myexpmv, dt, Atilde, [zeros(size (Atilde)-1, 1); 1/eta])(1:d, :);
U3 = x(:) + X;   g3 = feval (g, t + dt, U3);   D3 = g3 - gn;
M = select_taylor_degree_aug (Jn, [], 4, [], [], [], false, false, true);
myexpmv = @(t,A,v) expmv (t, A, v, M);
[Atilde, eta] = augmatrix ( Jn, [Fn, vn, (16*D2-2*D3)./(dt^2), (-48*D2+12*D3)./(dt^3)], [Fn, vn, (16*D2-2*D3)./(dt^2)]);
ep = [[zeros(3, 1); 1/eta; zeros(3, 1)], [zeros(6, 1); 1/eta]];
X = feval(myexpmv, dt, Atilde, [zeros(d, 1), zeros(d, 1); ep])(1:d, :);
x_next = x(:) + X(:, 1);
x_est = x(:) + X(:, 2);
k(:, 1) = Fn;   k(:, 2) = feval (f, t + 1/2*dt, U2);   k(:, 3) = feval (f, t + dt, U3);
```

Derivative approximation

$u' = F(u)$
$u(t_0) = u_0$
$J = \frac{\partial F}{\partial u}(u_0)$
$\delta = \sqrt{eps}$

- $\implies J(:,j) \approx \frac{F(u_0 + \delta e_j) - F(u_0)}{\delta}$
- $\implies J(:,j) \approx Im(F(u_0 + i\delta e_j)/\delta)$

Semilinear parabolic problem

Semilinear parabolic problem

$$\frac{\partial U}{\partial t}(x,t) - \frac{\partial^2 U}{\partial x^2}(x,t) = \frac{1}{1 + U(x,t)^2} + \Phi(x,t) \tag{1}$$

Semilinear parabolic problem

Semilinear parabolic problem

$$\frac{\partial U}{\partial t}(x,t) - \frac{\partial^2 U}{\partial x^2}(x,t) = \frac{1}{1+U(x,t)^2} + \Phi(x,t) \tag{1}$$

- $x \in [0,1]$
- $t \in [0,1]$
- homogeneous Dirichlet boundary conditions
- Φ chosen such that the exact solution is $U(x,t) = x(1-x)e^t$
- space discretization: standard finite differences with 200 grid points
- we expect to see order three for *odexprk*23 and order four for *odeprb*34

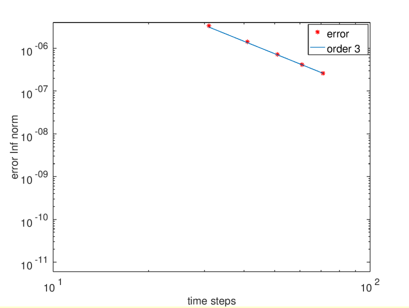
Example

```
% RUNGE KUTTA

a = 0;
b = 1;
m = 200;
x = linspace (a, b, m+2)';
x = x(2:m+1);
h = (b - a)/(m + 1);
A = toeplitz (sparse([1, 1], [1, 2], [-2, 1]/h^2, 1, m));
g = @(t,U) [ 1./(1 + U.^2) + exp(t)*(2 + x - x.^2) - 1./(1 + x.^2.*(1 - x).^2 * exp(2*t))];
F = @(t,U) A*U + g(t,U);
U0 = (x.*(1 - x));
trange = [0; 1];
opt = odeset ('AbsTol',  10^-2, 'GFun', g, 'LinOp', A, 'RelTol', 10^-2);
[t,U] = odexprk23 (F, trange, U0, opt);



% ROSENBROCK

a = 0;
b = 1;
m = 200;
x = linspace (a, b, m+2)';
x = x(2:m+1);
h = (b - a)/(m + 1);
A = toeplitz (sparse([1, 1], [1, 2], [-2, 1]/h^2, 1, m));
g = @(t,U) [ 1./(1 + U.^2) + exp(t)*(2 + x - x.^2) - 1./(1 + x.^2.*(1 - x).^2 * exp(2*t))];
F = @(t,U) A*U + g(t,U);
J = @(t,U) A - diag ([ 2*U./(1 + U.^2).^2 ]);
V = @(t,U) [exp(t)*(2 + x - x.^2) + ( 2*x.^2.*(1 - x).^2 * exp(2*t) )./( (1 + x.^2.*(1 - x).^2 * exp(2*t)).^2 )];
U0 = (x.*(1 - x));
trange = [0; 1];
opt = odeset ('AbsTol',  10^-2, 'DFdt', V, 'Jacobian', J, 'RelTol', 10^-2);
[t,U] = odexprb34 (F, trange, U0, opt);
```

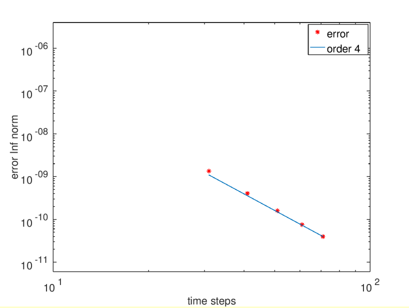
Convergence

Test with semilinear parabolic problem (1)

Runge-Kutta method



Rosenbrock method

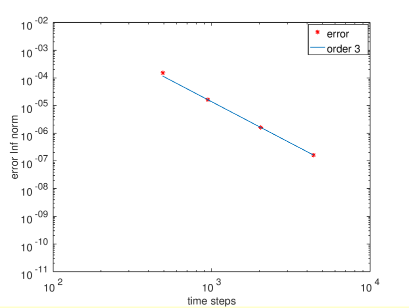


Schrödinger equation

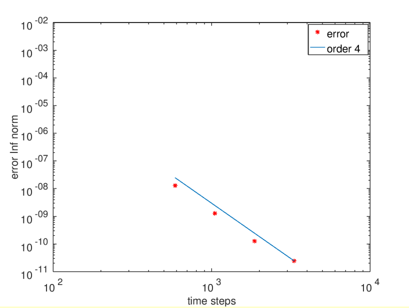
$$i\frac{\partial \psi}{\partial t} = H(x,t)\psi$$

$$H(x,t) = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + k\frac{x^2}{2} + \mu(\sin t)^2 x,$$

Runge-Kutta method



Rosenbrock method

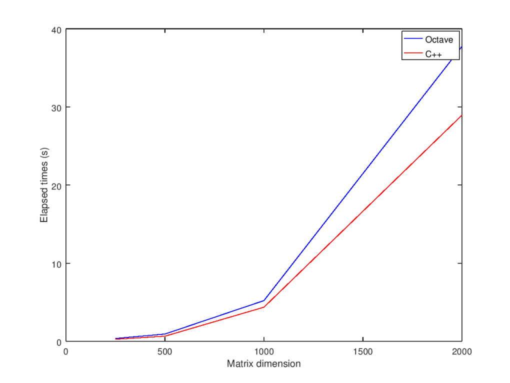


expmv.m and Oct-Files

```
%.cc file
DEFUN_DLD (taylor, args, , "Computing the Action of the Matrix Exponential")

%.m file
f = taylor (s, t, A, b, ...);
```

expmv.m and Oct-Files

```
%.cc file
DEFUN_DLD (taylor, args, , "Computing the Action of the Matrix Exponential")

%.m file
f = taylor (s, t, A, b, ...);
```



Thank you for your attention