

# Construction and Performance of Exponential Integrators


Mayya Tokman *(joint work with John Loffeld)*

*University of California, Merced*


2 hrs

1.5 hrs

## Outline:

- *Motivation*
- *Why Exponential Integrators?*
- *Building Blocks for Constructing an Exponential Integrator for General Nonlinear Problems*
  - *How are the methods constructed*
  - *Order conditions*
  - *Performance and adaptivity*
- *Conclusions and future work*

## General Nonlinear Initial Value Problem for System of ODEs:

$$\frac{dy}{dt} = f(y) \qquad y \in \mathbb{R}^N$$

$$y(t_0) = y_0 \qquad N >> 1$$

Difficult to solve numerically when it is stiff!

# What is Stiffness? And Where Do We Encounter It?

- There are many definitions of stiffness, e.g.

  *Lambert'91: A problem is stiff on a particular interval if a numerical method is forced to use an excessively small step size in relation to the smoothness of the solution.*

- Stiff problems are encountered in many fields, e.g.

  Plasma physics

  Combustion

  Fluid Mechanics

# General Nonlinear Problem:

$$\frac{dy}{dt} = f(y) \qquad y \in \mathbb{R}^N$$
$$y(t_0) = y_0 \qquad N >> 1$$

- *stiff, where stiffness can come from either linear OR nonlinear part of*

$$f(y) = Ly + N(y)$$

- $\Delta t_{Stability} << \Delta t_{Accuracy}$

- *no efficient preconditioner available for implicit methods.*

# Elementary Example:

1D Heat Equation $$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2}$$

Discretized in space

$$\frac{dU}{dt} = AU$$

$$A = \frac{\nu}{\Delta x^2}\begin{bmatrix} -2 & 1 & \dots & 0 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \dots & 1 & -2 \end{bmatrix}$$

Explicit scheme

$$\boxed{\begin{aligned} U^{n+1} &= U^n + A\Delta t U^n \\ U^{n+1} &= (I + A\Delta t)U^n \end{aligned}}$$

Implicit scheme

$$\boxed{\begin{aligned} U^{n+1} &= U^n + A\Delta t U^{n+1} \\ U^{n+1} &= (I - A\Delta t)^{-1}U^n \end{aligned}}$$

*But there is also exact solution!*

$$U^{n+1} = e^{A\Delta t}U^n$$

# Numerical Difficulties with Stiffness:

Eigenvalues of A place stability restriction on the time step for the explicit scheme

$$\mu_k = -\frac{4\nu}{(\Delta x)^2}\left(\sin\left(\frac{\pi k}{2(N+1)}\right)\right)^2$$

Explicit scheme

$$U^{n+1} = (I + A\Delta t)U^n$$

Stability requirement

$$|1 + \mu_k \Delta t| < 1$$

Implicit scheme VS. Exponential integrator

$$U^{n+1} - (I - A\Delta t)^{-1} U^n$$

$$U^{n+1} = e^{A\Delta t} U^n$$

*Both A-stable but…*

# Implicit vs. Exponential schemes:

Since A is large we need to approximate

$$(I - A\Delta t)^{-1}b = (I + (A\Delta t) + (A\Delta t)^2 + (A\Delta t)^3 + \ldots)b$$

or

$$e^{A\Delta t}b = \left(I + \frac{(A\Delta t)}{1!} + \frac{(A\Delta t)^2}{2!} + \frac{(A\Delta t)^3}{3!} + \ldots\right)b$$

Approximation method of choice for large nonsymmetric matrices is Krylov subspace projection

$$S_k = span\{b, Ab, \ldots, A^{k-1}b\}$$

(e.g. to invert the matrix we can use GMRES or FOM)

Convergence of Krylov iteration to estimate f(A)b depends on ||b||, eigenvalues of A *and function f(x)!*

# Test problem – 1D Brusselator:

$$\begin{aligned}\frac{du_i}{dt} &= 1 + u_i^2 v_i - 4u_i + \alpha \frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2} \\ \frac{dv_i}{dt} &= 3u_i - u_i^2 v_i + \alpha \frac{v_{i-1} - 2v_i + v_{i+1}}{(\Delta x)^2}, \quad i = 1, ..., N\end{aligned}$$

Jacobian matrix:

$$J = \begin{bmatrix} diag(2u_i v_i - 4) & diag(u_i^2) \\ diag(3 - u_i v_i) & diag(-u_i^2) \end{bmatrix} + \frac{\alpha}{(\Delta x)^2} \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix}$$

$$L = \begin{bmatrix} -2 & 1 & \dots & 0 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \dots & 1 & -2 \end{bmatrix}$$

# Krylov iteration convergence comparison:

Tolerance = 1E-5

| Problem size, 2N | GMRES $(I - A\Delta t)^{-1}b$ | FOM $(I - A\Delta t)^{-1}b$ | $e^{A\Delta t}b$ | $\psi_{21}(A\Delta t)b$ | $\psi_{22}(A\Delta t)b$ |
|---|---|---|---|---|---|
| 200 | 92 | 85 | 35 | 27 | 19 |
| 400 | 187 | 174 | 70 | 55 | 38 |
| 800 | 382 | 359 | 140 | 112 | 78 |

Similar result also holds for Jacobian calculated at different times and for other examples.

## Integrators for Nonlinear Problems:

- *Building an integrator while minimizing the number of Krylov projections*

- *Order conditions derivation - Butcher's work*

- *Performance and adaptivity*

## **Integrators for Nonlinear Problems:**

- *Building an integrator while minimizing the number of Krylov projections*

- *Order conditions derivation - Butcher's work*

- *Performance and adaptivity*

# Building Exponential Integrators for Nonlinear Problems:

$$y' = f(y) = f(y_0) + \frac{Df}{Dy}(y_0)(y - y_0) + R(y)$$

$$R(y) = f(y) - f(y_0) - \frac{Df}{Dy}(y_0)(y - y_0)$$

*Denote the Jacobian matrix* $A_0 = \frac{Df}{Dy}(y_0)$

*Use the integrating factor* $e^{hA_0}$ *to compute the integral form of the solution:*

$$\boxed{y(t_0 + h) = y_0 + \frac{e^{hA_0} - I}{hA_0} h f(y_0) + \int_{t_0}^{t_0+h} e^{(t_0+h-s)A_0} R(y(s))\,ds}$$

# Building Exponential Integrators:

$$y(t_0+h) = y_0 + \frac{e^{hA_0} - I}{hA_0} h f(y_0) + \int_{t_0}^{t_0+h} e^{(t_0+h-s)A_0} R(y(s))ds$$

To construct an exponential method approximate $R(y(s))$ with a polynomial.

This results in a expression for an approximate solution which involves functions

$$\phi_k(z) = \int_0^1 e^{(1-s)z} \frac{s^{k-1}}{(k-1)!} ds$$

or their linear combinations, e.g.

$$\psi_{\gamma k}(z) = \int_0^1 e^{(1-s)z} \binom{\gamma s}{k} ds$$

# Examples of EIs – EPIRK3 (*Tokman’06*):

$$k_1 - y_0 + \psi_1\left(\frac{h}{2}f'\right)hf(y_0)$$

$$y_1 = y_0 + \psi_1(hf')hf(y_0) + \frac{1}{3}\psi_2(hf')R(k_1)$$

$$\psi_1(z) = \frac{e^z - 1}{z}$$

$$\psi_2(z) = 2\frac{e^z - (1+z)}{z^2}$$

*Third-order scheme EPIRK3 – 2 Krylov projections*

## Computational Complexity of EI:

*NOTE: Krylov projections account for the majority of computational time!*

*Therefore to construct an efficient EI*

- *minimize number of Krylov projections needed each time step*
- *minimize the number of iterations needed for each Krylov projection*

## **Computational Complexity of EI:**

*NOTE: Krylov projections account for the majority of computational time!*

*Therefore to construct an efficient EI*

- *minimize number of Krylov projections needed each time step*

- *minimize the number of iterations needed for each Krylov projection*

# Examples of EIs – Lawson scheme:

$$
\begin{aligned}
k_1 &= f_n - A_n y_n \\
k_i &= f(p_i) - A_n p_i \\
p_i &= e^{c_i h A_n} y_n + h \sum_{j=1}^{i-1} e^{(c_i - c_j) h A_n} k_j \\
y_{n+1} &= e^{h A_n} y_n + h \sum_{i=1}^{m} e^{(1 - c_i) h A_n} k_i
\end{aligned}
$$

*Fourth-order scheme LWS4 – 4 Krylov projections*

# Examples of EIs – Hochbruck-Osterman:

$$
\begin{aligned}
g(y) &= f(y) - A_n y \\
k_1 &= e^{hA_n} y_n + \frac{1}{2}\phi_1(hA_n/2)hg(y_n) \\
k_2 &= e^{hA_n} y_n + \phi_1(hA_n)hg(k_1) \\
y_{n+1} &= e^{hA_n} y_n + \phi_1(hA_n)hg(y_n) \\
&+ \phi_3(hA_n)h(g(k_2) - 14g(y_n) + g(k_1))
\end{aligned}
$$

*Fourth-order scheme EROW4 – 4 Krylov projections*

# Examples of EIs – Hochbruck-Lubich:

$$
\begin{aligned}
k_1 &= \phi_1(hA_n/3)f(y_0) \\
k_2 &= \phi_1(2hA_n/3)f(y_0) \\
k_3 &= \phi_1(hA_n)f(y_0) \\
w_4 &= -\frac{7}{300}k_1 + \frac{97}{150}k_2 - \frac{37}{300}k_3 \\
u_4 &= y_0 + hw_4 \\
d_4 &= f(u_4) - f(y_0) - hA_n w_4 \\
k_4 &= \phi_1(hA_n/3)d_4 \\
k_5 &= \phi_1(2hA_n/3)d_4 \\
k_6 &= \phi_1(hA)d_4 \\
w_7 &= \frac{59}{300}k_1 - \frac{7}{75}k_2 + \frac{269}{300}k_3 + \frac{2}{3}(k_4 + k_5 + k_6)
\end{aligned}
$$

$$
\begin{aligned}
u_7 &= y_0 + hw_7 \\
d_7 &= f(u_7) - f(y_0) - hAw_7 \\
k_7 &= \phi_1(hA/3)d_7 \\
y_1 &= y_0 + h(k_3 + k_4 - \frac{4}{3}k_5 \\
&+ k_6 + \frac{1}{6}k_7)
\end{aligned}
$$

*Fourth-order scheme EXP4 – 3 Krylov projections*

# Examples of EIs – Tokman:

$$
\begin{aligned}
R(y) &= f(y) - f(y_n) - A_n(y - y_n) \\
k_1 &= y_n + a_{11}\psi_1(hA_n/3)hf(y_n) \\
k_2 &= y_n + a_{21}\psi_1(2hA_n/3)hf(y_n) + a_{22}\psi_2(2hA_n/3)hR(k_1) \\
y_{n+1} &= y_n + b_1\psi_1(hA_n)hf(y_n) + b_2\psi_2(hA_n)hR(k_1) \\
&+ b_3\psi_3(hA_n)h(-2R(k_1) + R(k_2))
\end{aligned}
$$

$$
\begin{aligned}
\psi_1(z) &= \phi_1(z) \\
\psi_2(z) &= 3\phi_2(z) \\
\psi_3(z) &= 3/2(-\phi_3(z) + 6\phi_2(z))
\end{aligned}
$$

*Fourth-order scheme EPIRK4 – 3 Krylov projections*

## Computational Complexity of EI:

*NOTE: Krylov projections account for the majority of computational time!*

*Therefore to construct an efficient EI*

- *minimize number of Krylov projections needed each time step*
- *minimize the number of iterations needed for each Krylov projection (work very much in progress…)*

# What functions should we project?

$$\phi_k(z) = \frac{1}{k!} + \frac{z}{(k+1)!} + \dots + \frac{z^n}{(k+n)!} + \dots$$

*Numerical experiments show that Arnoldi iteration convergence is correlated with Taylor series convergence.*

*Consider a linear combination of function*

$$\begin{aligned} k &= 3 \\ \chi_3(z) &= \phi_3(z) + a_2\phi_2(z) + a_1\phi_1(z) \end{aligned}$$

*with Taylor coefficients*

$$p_2(n) = \frac{1}{(n+3)!}\{1 - a_2(n+3) - a_1(n+3)(n+2)\}$$

# What functions should we project?

| Iteration # | Residual of $\phi_4(z)$ | Residual of $\chi_4(z)$ |
|---|---|---|
| 1 | 1.2812e-01 | 2.2956e-01 |
| 2 | 5.8218e-03 | 3.9212e-03 |
| 3 | 2.2079e-04 | 3.3878e-05 |
| 4 | 7.0092e-06 | 2.0108e-08 |
| 5 | 2.0214e-07 | 3.0619e-09 |
| 6 | 5.1313e-09 | 4.9313e-10 |
| 7 | 1.2053e-10 | 5.4611e-11 |
| 8 | 2.5530e-12 | 3.2960e-12 |
| 9 | 4.9000e-14 | 1.4000e-13 |
| 10 | 1.0000e-15 | 2.0000e-14 |

# Ideas for New Integrators:

$$Y_1 = y_0 + a_{11}\chi_1(c_{11}hf'(y_0))hf(y_0)$$

$$Y_2 = y_0 + a_{21}\chi_1(c_{21}hf'(y_0))hf(y_0) + a_{22}\chi_2(c_{22}hf'(Y_1))R(Y_1)$$

$$\begin{aligned} y_1 = y_0 \quad &+ \quad \chi_1(hf'(y_0))hf(y_0) + b_1\chi_2(hf'(Y_1))R(Y_1) \\ &+ \quad b_2\chi_3(hf'(Y_2))(d_2R(Y_2) - d_1R(Y_1)) \end{aligned}$$

$\chi_k(z), R(Y_k)$ are chosen to optimize Krylov projections

## **Integrators for Nonlinear Problems:**

- *Building an integrator while minimizing the number of Krylov projections*

- *Order conditions derivation - Butcher's work*

- *Performance and adaptivity*

# Performance:

2D Brusselator

$$\frac{\partial u}{\partial t} = 1 + uv^2 - 4u + \alpha \nabla^2 u$$
$$\frac{\partial v}{\partial t} = 3u - u^2 v + \alpha \nabla^2 v$$

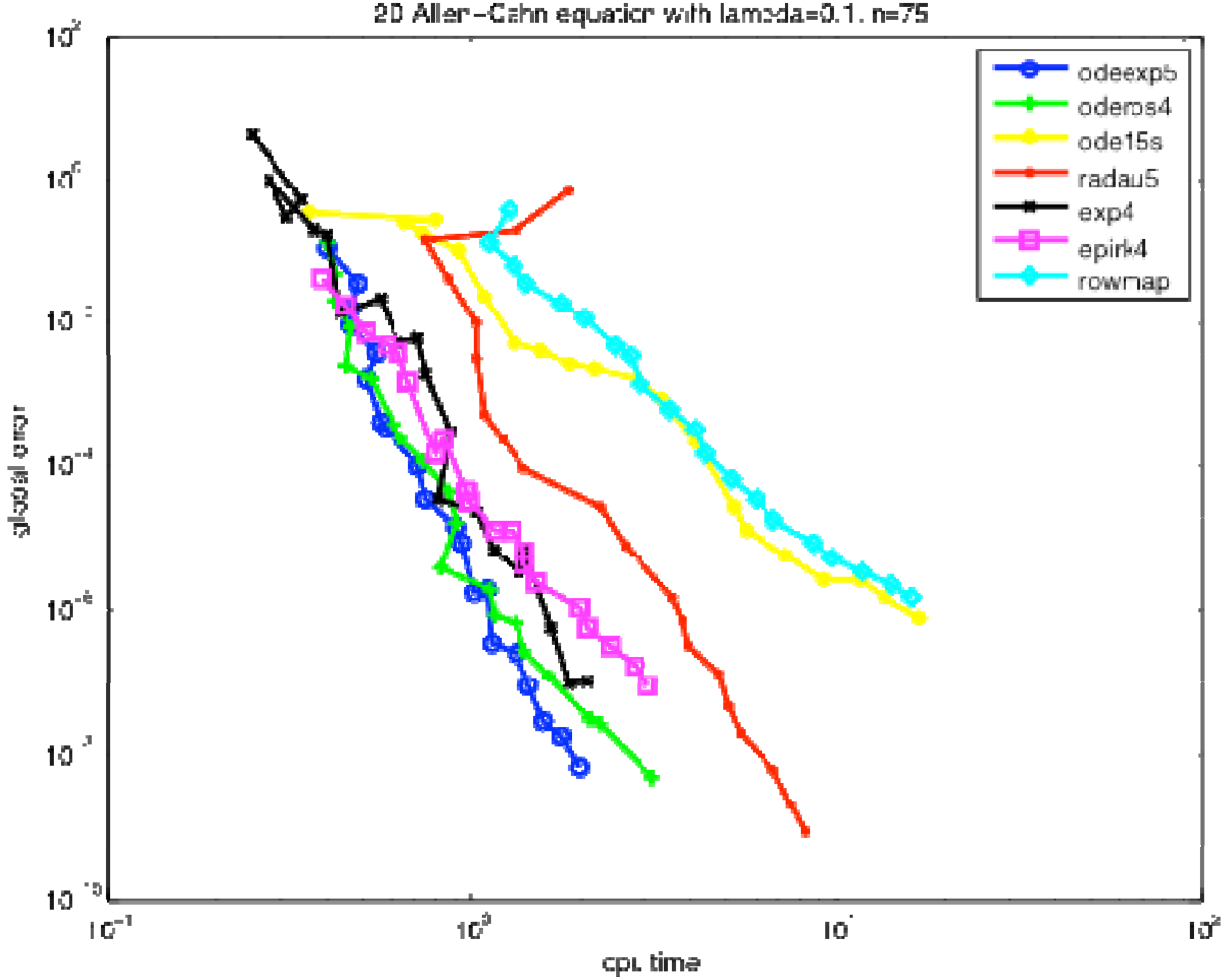
2D Gray-Scott Equation

$$\frac{\partial u}{\partial t} = -uv^2 + 0.04(1-u) + 0.2\nabla^2 u$$
$$\frac{\partial v}{\partial t} = uv^2 + 0.1v + 0.1\nabla^2 v$$

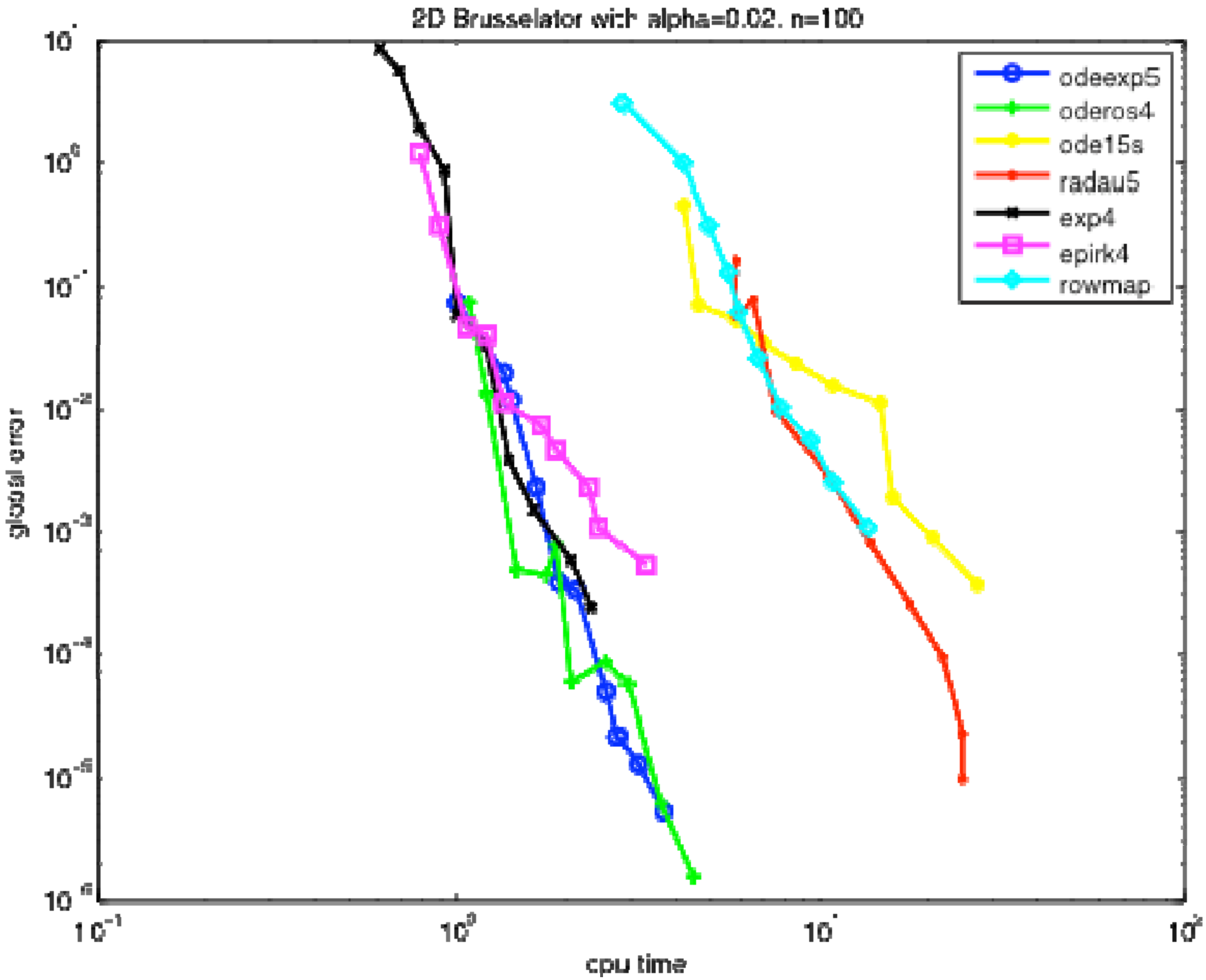
2D Allen-Cahn Equation

$$\frac{\partial u}{\partial t} = u - u^3 + 0.1\nabla^2 u$$

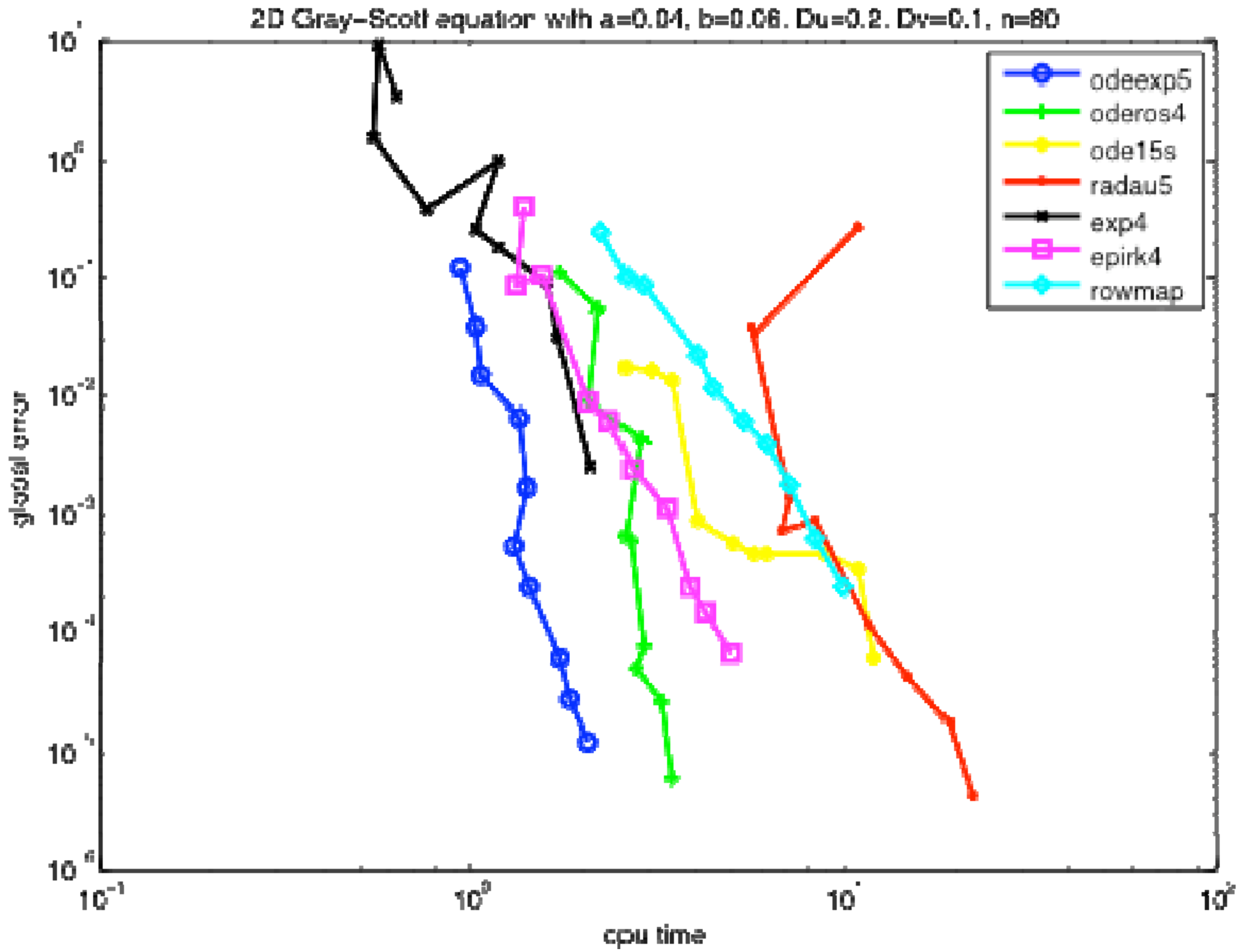
# 2D Allen-Cahn Equation:



# 2D Brusselator System:



# 2D Gray-Scott System:



# Adaptivity:

*Two level time stepping adaptivity*

- *Error estimator using embedded methods*
- *Optimization of the Krylov subspace size*

*Efficiency is ensured by effectively coupling the two stages of adaptivity.*

# Current and Future Work:

- *Thorough comparisons of EIs and standard integrators performance on both sample problems and real applications*

- *Butcher's results will be used to develop new integrators and automate derivation of order conditions*

- *Publicly available serial and parallel implementations of EIs is under development*

- *Applications: nonlinear optics, plasma physics, biomodeling, etc.*