Exp^3 Integrators for semilinear problems

Séminaire d'analyse numérique

23 March. 2004, Geneve.

Borislav V. Minchev

Borko.Minchev@ii.uib.no

http://www.ii.uib.no/∼borko

Joint work with Will Wright

Department of computer science

University of Bergen, Norway

Exp^3 = Explaining Explicit Exponential Integrators

Outline

- Motivation and History
- Formulation of the method
- Connections with Lie group methods
- Numerical schemes
- Numerical experiments
- Generalized IF methods
- Conclusions
- Future work

Motivation and History

A new interest in exponential integrators for semilinear problems

$$u_t \ = \ \mathcal{L}u + \mathcal{N}(u,t), \ u(t_0) = u_0$$

Motivation and History

A new interest in exponential integrators for semilinear problems

$$u_t = \mathcal{L}u + \mathcal{N}(u,t), \; u(t_0) = u_0$$

Brief History

- Lawson, 1967
  - IF methods, A-stability
- Friedli, 1978
  - Exp. Exp. RK Methods, nonstiff order
- Strehmel and Weiner, 1987
  - Adaptive RK Methods, Order theory, B-stability
- Hochbruck and Lubich, 1997
  - Exp. Integrators (EXP4) with inexact Jacobian

Motivation and History

A new interest in exponential integrators for semilinear problems

$$u_t = \mathcal{L}u + \mathcal{N}(u,t), \ u(t_0) = u_0$$

- Cox and Matthews, 2002
  - ETDRK Methods of 3-th and 4-th order
- Kasam and Trefethen, 2002
  - Extensive numerical experiments, overcome the numerical instability
- Celledoni, Marthinsen, Owren, 2003
  - Commutator Free Lie group methods
- Krogstad, 2003
  - Generalized IF Methods, connection with CF
- Hochbruck and Osterman, 2003
  - Exp. Collocation Methods, convergence analyze

Pure Exponential Integrators

- IF = Integrating Factor

Pure Exponential Integrators

- IF = Integrating Factor
- ETD = Exponential Time Differencing
  - ECM = Exponential Collocation Methods
  - $ETDRK$ = Exponential time Differencing Runge-Kutta Methods

Pure Exponential Integrators

- IF = Integrating Factor
- ETD = Exponential Time Differencing
  - ECM = Exponential Collocation Methods
  - $ETDRK$ = Exponential time Differencing Runge-Kutta Methods
- CF = Commutator Free Lie-group methods

Pure Exponential Integrators

- IF = Integrating Factor
- ETD = Exponential Time Differencing
  - ECM = Exponential Collocation Methods
  - $ETDRK$ = Exponential time Differencing Runge-Kutta Methods
- CF = Commutator Free Lie-group methods
- GIF = Generalized Integrating Factor

Integrating Factor

After space discretisation we obtain a systems of ODEs

$$(1) \qquad u' = \mathbf{L}u + \mathbf{N}(u(t)), \; u(t_0) = u_0$$

Integrating Factor

After space discretisation we obtain a systems of ODEs

$$u' = \mathbf{L}u + \mathbf{N}(u(t)), \; u(t_0) = u_0 \tag{1}$$

Solve exactly the linear part then make a change of variables

$$\begin{aligned} v(t) &= \exp(-\mathbf{L}t)u(t) \\ \underbrace{\exp(-\mathbf{L}t)(u_t - \mathbf{L}u)}_{v_t} &= \exp(-\mathbf{L}t)\mathbf{N}(u) \\ v_t &= \exp(-\mathbf{L}t)\mathbf{N}(\exp(\mathbf{L}t)v) \end{aligned}$$

Integrating Factor

After space discretisation we obtain a systems of ODEs

$$u' \;=\; \mathbf{L}u + \mathbf{N}(u(t)),\;\; u(t_0) = u_0 \tag{1}$$

Solve exactly the linear part then make a change of variables

$$\begin{aligned} v(t) &= \exp(-\mathbf{L}t)u(t) \\ \underbrace{\exp(-\mathbf{L}t)(u_t - \mathbf{L}u)}_{v_t} &= \exp(-\mathbf{L}t)\mathbf{N}(u) \\ v_t &= \exp(-\mathbf{L}t)\mathbf{N}(\exp(\mathbf{L}t)v) \end{aligned}$$

Apply a numerical method to the transformed equation

Integrating Factor

After space discretisation we obtain a systems of ODEs

$$u' = \mathbf{L}u + \mathbf{N}(u(t)), \; u(t_0) = u_0 \tag{1}$$

Solve exactly the linear part then make a change of variables

$$\begin{aligned} v(t) &= \exp(-\mathbf{L}t)u(t) \\ \underbrace{\exp(-\mathbf{L}t)(u_t - \mathbf{L}u)}_{v_t} &= \exp(-\mathbf{L}t)\mathbf{N}(u) \\ v_t &= \exp(-\mathbf{L}t)\mathbf{N}(\exp(\mathbf{L}t)v) \end{aligned}$$

Apply a numerical method to the transformed equation

Transform back the approximate solution to the original variable

Example of IF

In terms of the original variable the computations performed are

$$
\begin{aligned}
U_1 &= u_0 \\
U_2 &= e^{c_2 hL}(u_0 + a_{21} hN(U_1)) \\
U_3 &= e^{c_3 hL}(u_0 + a_{31} hN(U_1) + a_{32} he^{-c_2 hL} N(U_2)) \\
U_4 &= e^{c_4 hL}(u_0 + a_{41} hN(U_1) + a_{42} he^{-c_2 hL} N(U_2) \\
&\quad + a_{43} he^{-c_3 hL} N(U_3)) \\
u_1 &= e^{hL}(u_0 + b_1 hN(U_1) + b_2 he^{-c_2 hL} N(U_2) \\
&\quad + b_3 he^{-c_3 hL} N(U_3) + b_4 he^{-c_4 hL} N(U_4))
\end{aligned}
$$

Example of IF

General form of the order 4 integrating factor method is

$$\left[\begin{array}{cccc|c} 0 & 0 & 0 & 0 & 1 \\ a_{21}e^{c_2 hL} & 0 & 0 & 0 & e^{c_2 hL} \\ a_{31}e^{c_3 hL} & a_{32}e^{(c_3-c2)hL} & 0 & 0 & e^{c_3 hL} \\ a_{41}e^{c_4 hL} & a_{42}e^{(c_3-c2)hL} & a_{43}e^{(c_4-c3)hL} & 0 & e^{c_4 hL} \\ \hline b_1 e^{hL} & b_2 e^{(1-c_2)hL} & b_3 e^{(1-c_3)hL} & b_4 e^{(1-c_4)hL} & e^{hL} \end{array}\right]$$

- Uniformly distributed c vector provides cheapest methods.
- This structure requires only classical order conditions.

Exponential Time Differencing

Similar approach to IF but we do not make a complete change of variables

$$\frac{d}{dt}(\exp(-\mathbf{L}t)u) \;=\; \exp(-\mathbf{L}t)\mathbf{N}(u(t))$$

Exponential Time Differencing

Similar approach to IF but we do not make a complete change of variables

$$\frac{d}{dt}(\exp(-\mathbf{L}t)u) \;=\; \exp(-\mathbf{L}t)\mathbf{N}(u(t))$$

Integrate over a single time step of length $c_i h$

$$u(t_n + c_i h) \;=\; \exp(c_i h\mathbf{L})u_n + \exp(c_i h\mathbf{L}) \int_0^{c_i h} \exp(-\mathbf{L}\tau)\mathbf{N}(u(t_n+\tau))d\tau$$

Exponential Time Differencing

Similar approach to IF but we do not make a complete change of variables

$$\frac{d}{dt}(\exp(-\mathbf{L}t)u) \;=\; \exp(-\mathbf{L}t)\mathbf{N}(u(t))$$

Integrate over a single time step of length $c_i h$

$$u(t_n + c_i h) \;=\; \exp(c_i h\mathbf{L})u_n + \exp(c_i h\mathbf{L}) \int_0^{c_i h} \exp(-\mathbf{L}\tau)\mathbf{N}(u(t_n + \tau))d\tau$$

ETD methods of multistep type use polynomial approximation to the function $\mathbf{N}(u(t_n+\tau))$

(Nørsett'69, Cox and Matthews'02)

Approximations to $N(u(t))$

- If $N(u(t)) \approx \delta_i e^{Lt}$, where δ_i is a constant such that the approximation matches $N(u(t))$ for $t = c_i h$, then the coefficients of the method will be linear combinations of

$$\phi^{[i]}(\lambda)(hL) = e^{(\lambda - c_i)hL} \quad i = 1, 2, 3, \dots$$

Approximations to $N(u(t))$

- If $N(u(t)) \approx P_{s-1}(t)$, where $P_{s-1}(t)$ is a Lagrange interpolation polynomial of degree $s-1$ that matches $N(u(t)$ at the points $t = c_1 h, c_2 h, \ldots, c_s h$ then

Approximations to $N(u(t))$

- If $N(u(t)) \approx P_{s-1}(t)$, where $P_{s-1}(t)$ is a Lagrange interpolation polynomial of degree $s-1$ that matches $N(u(t)$ at the points $t = c_1 h, c_2 h, \ldots, c_s h$ then

$$\phi^{[1]}(\lambda)(hL) = \frac{e^{\lambda hL} - I}{\lambda hL}, \quad \phi^{[2]}(\lambda)(hL) = \frac{e^{\lambda hL} - \lambda hL - I}{(\lambda hL)^2},$$

$$\phi^{[i+1]}(\lambda)(hL) = \frac{\phi^{[i]}(\lambda)(hL) - \phi^{[i]}(0)(hL)}{\lambda hL} \quad i = 1, 2, 3, \ldots$$

Approximations to $N(u(t))$

- If $N(u(t)) \approx P_{s-1}(t)$, where $P_{s-1}(t)$ is a Lagrange interpolation polynomial of degree $s-1$ that matches $N(u(t)$ at the points $t = c_1 h, c_2 h, \ldots, c_s h$ then

$$\begin{aligned}
\phi^{[1]}(\lambda)(hL) &= 1I_m + \frac{\lambda}{2!}hL + \frac{\lambda^2}{3!}(hL)^2 + \frac{\lambda^3}{4!}(hL)^3 + \frac{\lambda^4}{5!}(hL)^4 + \cdots, \\
\phi^{[2]}(\lambda)(hL) &= \frac{1}{2!}I_m + \frac{\lambda}{3!}(hL) + \frac{\lambda^2}{4!}(hL)^2 + \frac{\lambda^3}{5!}(hL)^3 + \frac{\lambda^4}{6!}(hL)^4 + \cdots, \\
\phi^{[3]}(\lambda)(hL) &= \frac{1}{3!}I_m + \frac{\lambda}{4!}(hL) + \frac{\lambda^2}{5!}(hL)^2 + \frac{\lambda^3}{6!}(hL)^3 + \frac{\lambda^4}{7!}(hL)^4 + \cdots.
\end{aligned}$$

Approximations to $N(u(t))$

- If $N(u(t)) \approx P_{s-1}(t)$, where $P_{s-1}(t)$ is a Lagrange interpolation polynomial of degree $s-1$ that matches $N(u(t)$ at the points $t = c_1 h, c_2 h, \ldots, c_s h$ then

$$\begin{aligned}
\phi^{[1]}(\lambda)(hL) &= 1I_m + \frac{\lambda}{2!}hL + \frac{\lambda^2}{3!}(hL)^2 + \frac{\lambda^3}{4!}(hL)^3 + \frac{\lambda^4}{5!}(hL)^4 + \cdots, \\
\phi^{[2]}(\lambda)(hL) &= \frac{1}{2!}I_m + \frac{\lambda}{3!}(hL) + \frac{\lambda^2}{4!}(hL)^2 + \frac{\lambda^3}{5!}(hL)^3 + \frac{\lambda^4}{6!}(hL)^4 + \cdots, \\
\phi^{[3]}(\lambda)(hL) &= \frac{1}{3!}I_m + \frac{\lambda}{4!}(hL) + \frac{\lambda^2}{5!}(hL)^2 + \frac{\lambda^3}{6!}(hL)^3 + \frac{\lambda^4}{7!}(hL)^4 + \cdots.
\end{aligned}$$

- If $N(u(t)) \approx T_{s-1}(t)$, where $T_{s-1}(t)$ is a trigonometrical polynomial of $\sin(\alpha t)$, then

$$\phi^{[1]}(\lambda)(hL) = \frac{e^{\lambda hL} - I}{\lambda hL}, \quad \phi^{[2]}(\lambda)(hL) = \frac{e^{\lambda hL} - L\sin(\lambda h) - \cos(\lambda h)}{\lambda h(L+I)^2},$$

$$\phi^{[\alpha+1]}(\lambda)(hL) = \frac{\alpha e^{\lambda hL} - L\sin(\alpha\lambda h) - \alpha\cos(\alpha\lambda h)}{\lambda h(L^2 + \alpha^2 I)} \quad \alpha = 1, 2, 3, \ldots$$

Exponential Collocation Methods (1/2)

Recall vcf

$$u_{n+1} \;=\; exp(\mathbf{L}h)u_n + exp(\mathbf{L}h)\int_0^h exp(-\mathbf{L}\tau)\mathbf{N}(u(t_n+\tau))d\tau$$

Exponential Collocation Methods (1/2)

Recall vcf

$$u_{n+1} \;=\; exp(\mathbf{L}h)u_n + exp(\mathbf{L}h)\int_0^h exp(-\mathbf{L}\tau)\mathbf{N}(u(t_n+\tau))d\tau$$

- choose collocation nodes $c_1, \ldots, c_s$
- let $u_n \approx u(t_n)$, $U_{n,i} \approx u(t_n + c_i h)$
- p_n collocational polynomial of degree $s-1$ $\;p_n(c_i h) = \mathbf{N}(U_{n,i})$

Exponential Collocation Methods (1/2)

Recall vcf

$$u_{n+1} = exp(\mathbf{L}h)u_n + exp(\mathbf{L}h)\int_0^h exp(-\mathbf{L}\tau)\mathbf{N}(u(t_n+\tau))d\tau$$

- choose collocation nodes $c_1, \ldots, c_s$
- let $u_n \approx u(t_n)$, $U_{n,i} \approx u(t_n + c_i h)$
- p_n collocational polynomial of degree $s-1$ $\;p_n(c_i h) = \mathbf{N}(U_{n,i})$
- explicit methods: $p_{n,i}$ polynomial of degree $i-1$ using $\;\mathbf{N}(U_{n,j}),\quad j \leq i-1$

$$U_{n,i} = e^{\mathbf{L}h}u_n + \int_0^{c_i h} e^{(c_i h-\tau)\mathbf{L}} p_{n,i}(\tau)d\tau$$

$$u_{n,i} = e^{\mathbf{L}h}u_n + \int_0^{h} e^{(h-\tau)\mathbf{L}} p_{n}(\tau)d\tau$$

Exponential Collocation Methods (2/2)

key point: integrals can be calculated exactly

$$\int_0^{c_i h} e^{(c_i h - \tau)\mathbf{L}} p_{n,i}(\tau) d\tau \;=\; h \sum_{j=1}^{s} a_{ij}(c_i h \mathbf{L}) \mathbf{N}(U_{n,j})$$

$$\int_0^{h} e^{(h - \tau)\mathbf{L}} p_{n}(\tau) d\tau \;=\; h \sum_{i=1}^{s} b_{i}(h \mathbf{L}) \mathbf{N}(U_{n,i})$$

Exponential Collocation Methods (2/2)

key point: integrals can be calculated exactly

$$\int_0^{c_i h} e^{(c_i h-\tau)\mathbf{L}} p_{n,i}(\tau) d\tau \;=\; h \sum_{j=1}^{s} a_{ij}(c_i h\mathbf{L}) \mathbf{N}(U_{n,j})$$

$$\int_0^{h} e^{(h-\tau)\mathbf{L}} p_{n}(\tau) d\tau \;=\; h \sum_{i=1}^{s} b_{i}(h\mathbf{L}) \mathbf{N}(U_{n,i})$$

coefficients b_i, a_{ij} are linear combination of $\phi^{[1]}, \ldots, \phi^{[s]}$ (Beylkin, Keiser, Vozovi'98) defined as

$$\phi^{[i]}(-t\mathbf{L}) \;=\; \frac{1}{(i-1)!t^j} \int_0^t (t-\tau)^{i-1} e^{-\tau\mathbf{L}} d\tau, \quad i \geq 1$$

Exponential Collocation Methods (2/2)

key point: integrals can be calculated exactly

$$\int_0^{c_i h} e^{(c_i h-\tau)\mathbf{L}} p_{n,i}(\tau)d\tau \;=\; h\sum_{j=1}^{s} a_{ij}(c_i h\mathbf{L})\mathbf{N}(U_{n,j})$$

$$\int_0^{h} e^{(h-\tau)\mathbf{L}} p_{n}(\tau)d\tau \;=\; h\sum_{i=1}^{s} b_{i}(h\mathbf{L})\mathbf{N}(U_{n,i})$$

coefficients $b_i, \quad a_{ij}$ are linear combination of $\phi^{[1]}, \ldots, \phi^{[s]}$ (Beylkin, Keiser, Vozovi'98) defined as

$$\phi^{[i]}(-t\mathbf{L}) \;=\; \frac{1}{(i-1)!t^j}\int_0^t (t-\tau)^{i-1} e^{-\tau\mathbf{L}} d\tau, \quad i \geq 1$$

- No need of new order theory
- Easy to solve all order conditions

Exponential Collocation Methods (2/2)

key point: integrals can be calculated exactly

$$\int_0^{c_i h} e^{(c_i h-\tau)\mathbf{L}} p_{n,i}(\tau)d\tau \;=\; h\sum_{j=1}^{s} a_{ij}(c_i h\mathbf{L})\mathbf{N}(U_{n,j})$$

$$\int_0^{h} e^{(h-\tau)\mathbf{L}} p_{n}(\tau)d\tau \;=\; h\sum_{i=1}^{s} b_{i}(h\mathbf{L})\mathbf{N}(U_{n,i})$$

coefficients b_i, a_{ij} are linear combination of $\phi^{[1]}, \ldots, \phi^{[s]}$ (Beylkin, Keiser, Vozovi'98) defined as

$$\phi^{[i]}(-t\mathbf{L}) \;=\; \frac{1}{(i-1)!t^j}\int_0^t (t-\tau)^{i-1}e^{-\tau\mathbf{L}}d\tau, \quad i \geq 1$$

- Are they too restrictive in the choice of the quadrature formula?
- What if $c_i = c_j$?
- How to analyze the methods of Cox and Matthews and Krogstad?

The ϕ function

The generality of the ϕ functions is not known. For $\lambda \in \mathbb{R}$ and $hL \in \mathbb{R}^{m \times m}$, let

$$\phi^{[i]}(\lambda)(hL) = \sum_{j=0}^{\infty} \phi_j^{[i]}(\lambda)(hL)^j.$$

The ϕ function

The generality of the ϕ functions is not known. For $\lambda \in \mathbb{R}$ and $hL \in \mathbb{R}^{m\times m}$, let

$$\phi^{[i]}(\lambda)(hL) = \sum_{j=0}^{\infty} \phi_j^{[i]}(\lambda)(hL)^j.$$

The ϕ function must

- Be computed exactly or to arbitrary high order cheaply.
- Other requirement?

The ϕ function

The generality of the ϕ functions is not known. For $\lambda \in \mathbb{R}$ and $hL \in \mathbb{R}^{m\times m}$, let

$$\phi^{[i]}(\lambda)(hL) = \sum_{j=0}^{\infty} \phi_j^{[i]}(\lambda)(hL)^j.$$

The ϕ function must

- Be computed exactly or to arbitrary high order cheaply.
- Other requirement?

The ϕ functions of the IF and ETD work. Certain their linear combinations also work. Are there others?

The method coefficients

The matrix A is defined as

$$A = \sum_{i=1}^{d} \Phi^{[i]}(c)(hL)(\alpha^{(i)} \otimes I_m),$$

where $\alpha^{(i)}$ are $s \times s$ strictly lower triangular matrices with the first i rows equal to zero and m is the dimensionality of the problem.

The method coefficients

The matrix A is defined as

$$A = \sum_{i=1}^{d} \Phi^{[i]}(c)(hL)(\alpha^{(i)} \otimes I_m),$$

where $\alpha^{(i)}$ are $s \times s$ strictly lower triangular matrices with the first i rows equal to zero and m is the dimensionality of the problem.
The matrix b^T is defined as

$$b^T = \sum_{i=1}^{s} \phi^{[i]}(1)(hL)({\beta^{(i)}}^T \otimes I_m).$$

where $\beta^{(i)}$ are arbitrary s vectors.

The method coefficients

The matrix A is defined as

$$A = \sum_{i=1}^{d} \Phi^{[i]}(c)(hL)(\alpha^{(i)} \otimes I_m),$$

where $\alpha^{(i)}$ are $s \times s$ strictly lower triangular matrices with the first i rows equal to zero and m is the dimensionality of the problem.
The matrix b^T is defined as

$$b^T = \sum_{i=1}^{s} \phi^{[i]}(1)(hL)({\beta^{(i)}}^T \otimes I_m).$$

where $\beta^{(i)}$ are arbitrary s vectors.
The matrix $\Phi^{[i]}$ is

$$\Phi^{[i]}(c)(hL) = \mathrm{diag}(\phi^{[i]}(c_1)(hL), \ldots, \phi^{[i]}(c_s)(hL))$$

Formulation of the method

$$
\begin{aligned}
U^{[n]} &= AhN(U^{[n]}) + e^{chL}u_{n-1}, \\
u_n &= b^T hN(U^{[n]}) + e^{hL}u_{n-1}
\end{aligned}
$$

Formulation of the method

$$\begin{aligned} U^{[n]} &= AhN(U^{[n]}) + e^{chL}u_{n-1}, \\ u_n &= b^T hN(U^{[n]}) + e^{hL}u_{n-1} \end{aligned}$$

The computations performed in step number n, are

$$\left[\begin{array}{c} U^{[n]} \\ \hline u_n \end{array}\right] = \left[\begin{array}{c|c} A & e^{chL} \\ \hline b^T & e^{hL} \end{array}\right] \left[\begin{array}{c} hN(U^{[n]}) \\ \hline u_{n-1} \end{array}\right]$$

Formulation of the method

$$
\begin{aligned}
U^{[n]} &= AhN(U^{[n]}) + e^{chL}u_{n-1}, \\
u_n &= b^T hN(U^{[n]}) + e^{hL}u_{n-1}
\end{aligned}
$$

The computations performed in step number n, are

$$
\left[\begin{array}{c} U^{[n]} \\ \hline u_n \end{array}\right] = \left[\begin{array}{c|c} A & e^{chL} \\ \hline b^T & e^{hL} \end{array}\right] \left[\begin{array}{c} hN(U^{[n]}) \\ \hline u_{n-1} \end{array}\right]
$$

where the corresponding vectors are

$$
U^{[n]} = \begin{bmatrix} U_1^{[n]} \\ U_2^{[n]} \\ \vdots \\ U_s^{[n]} \end{bmatrix} \quad N(U^{[n]}) = \begin{bmatrix} N(U_1^{[n]}) \\ N(U_2^{[n]}) \\ \vdots \\ N(U_s^{[n]}) \end{bmatrix} \quad e^{chL} = \begin{bmatrix} e^{c_1 hL} \\ e^{c_2 hL} \\ \vdots \\ e^{c_s hL} \end{bmatrix}
$$

Formulation of the method

$$\begin{aligned} U^{[n]} &= AhN(U^{[n]}) + e^{chL}u_{n-1}, \\ u_n &= b^T hN(U^{[n]}) + e^{hL}u_{n-1} \end{aligned}$$

The computations performed in step number n, are

$$\left[\begin{array}{c} U^{[n]} \\ \hline u_n \end{array}\right] = \left[\begin{array}{c|c} A & e^{chL} \\ \hline b^T & e^{hL} \end{array}\right] \left[\begin{array}{c} hN(U^{[n]}) \\ \hline u_{n-1} \end{array}\right]$$

alternative representation of the method in a more Runge-Kutta tabelau formulation is

$$\begin{array}{c|c|c|c|c|c} c & \alpha^{(1)} & \alpha^{(2)} & \dots & \alpha^{(s-1)} & \\ \hline & {\beta^{(1)}}^T & {\beta^{(2)}}^T & \dots & {\beta^{(s-1)}}^T & {\beta^{(s)}}^T \end{array}$$

General formulation

A 3 stage example of the general formulation is

$$
\left[\begin{array}{ccc|c}
0 & 0 & 0 & e^{c_1 hL} \\
a_{21}^1 \phi^1 & 0 & 0 & e^{c_2 hL} \\
a_{31}^1 \phi^1 + a_{31}^2 \phi^2 & a_{32}^1 \phi^1 + a_{32}^2 \phi^2 & 0 & e^{c_3 hL} \\
\hline
b_1^1 \phi^1 + b_1^2 \phi^2 + b_1^3 \phi^3 & b_2^1 \phi^1 + b_2^2 \phi^2 + b_3^3 \phi^3 & b_3^1 \phi^1 + b_3^1 \phi^2 + b_3^3 \phi^3 & e^{hL}
\end{array}\right]
$$

General formulation

ETD4-Kr of Krogstad

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2}\phi^{[1]} & & & \\
\frac{1}{2} & \frac{1}{2}\phi^{[1]} - \phi^{[2]} & \phi^{[2]} & & \\
1 & \phi^{[1]} - 2\phi^{[2]} & 0 & 2\phi^{[2]} & \\
\hline
 & 4\phi^{[3]} - 3\phi^{[2]} + \phi^{[1]} & -4\phi^{[3]} + 2\phi^{[2]} & -4\phi^{[3]} + 2\phi^{[2]} & 4\phi^{[3]} - \phi^{[2]}
\end{array}
$$

General formulation

ETDRK4SW of Strehmel and Weiner

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2}\phi^{[1]} & & & \\
\frac{1}{2} & \frac{1}{2}\phi^{[1]} - \frac{1}{2}\phi^{[2]} & \frac{1}{2}\phi^{[2]} & & \\
1 & \phi^{[1]} - 2\phi^{[2]} & -2\phi^{[2]} & 4\phi^{[2]} & \\
\hline
 & 4\phi^{[3]} - 3\phi^{[2]} + \phi^{[1]} & 0 & -8\phi^{[3]} + 4\phi^{[2]} & 4\phi^{[3]} - \phi^{[2]}
\end{array}
$$

Order theory

- Associate a white node with L and black node with N
- Elementary differentials can be represented by bi-coloured rooted trees where any white node has only one child.

Order theory

- Associate a white node with L and black node with N
- Elementary differentials can be represented by bi-coloured rooted trees where any white node has only one child.

The number of rooted trees

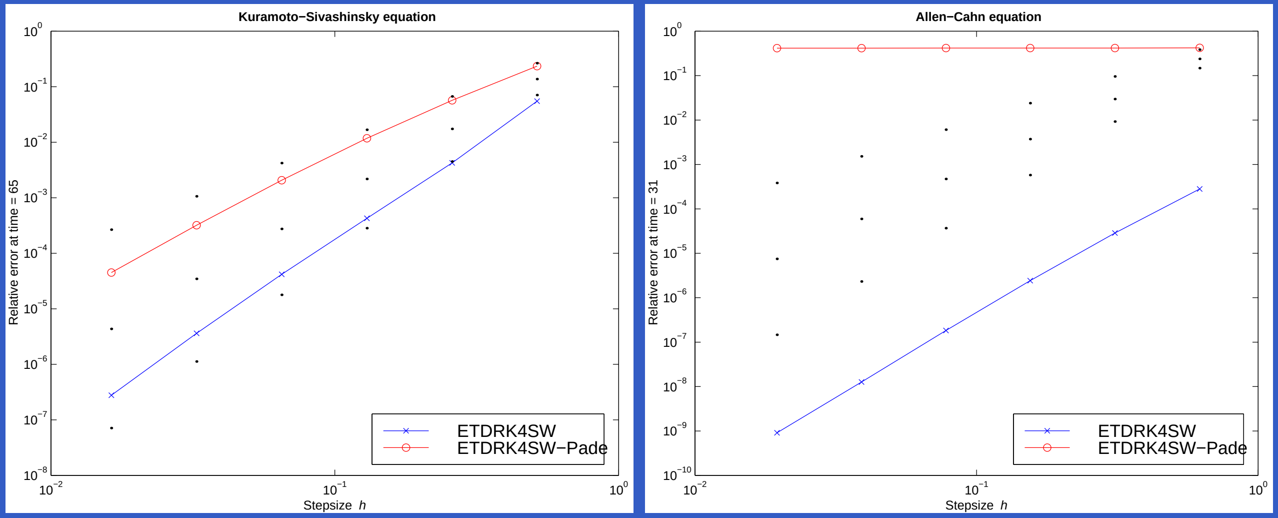
n	1	2	3	4	5	6	7	8
θ_n	2	4	11	34	117	421	1589	6162
$\Theta = \sum_i^n \theta_n$	2	6	17	51	168	589	2178	8340

Comparison with ARK Methods

- The order theory is the same

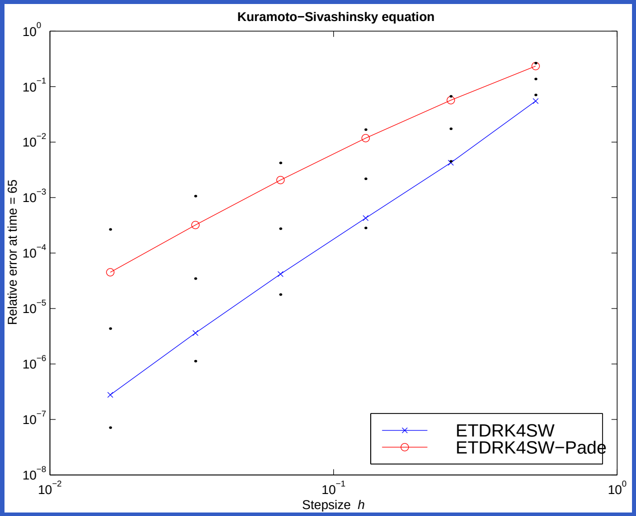
Comparison with ARK Methods

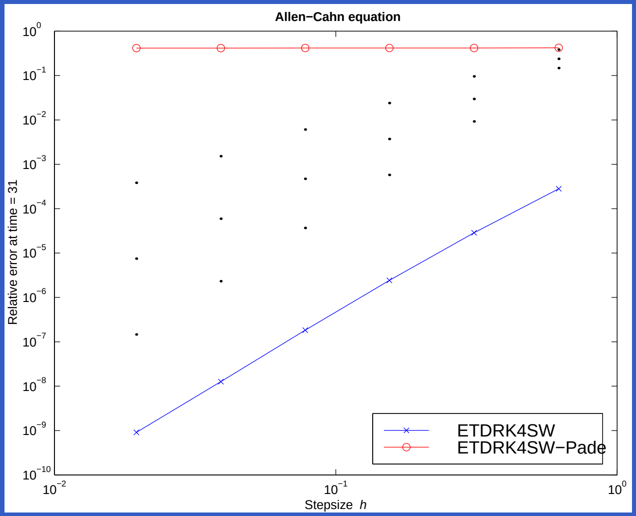
- The order theory is the same
- ARK use rational approximations to the $\exp$ and $\phi^{[i]}$



Comparison with ARK Methods

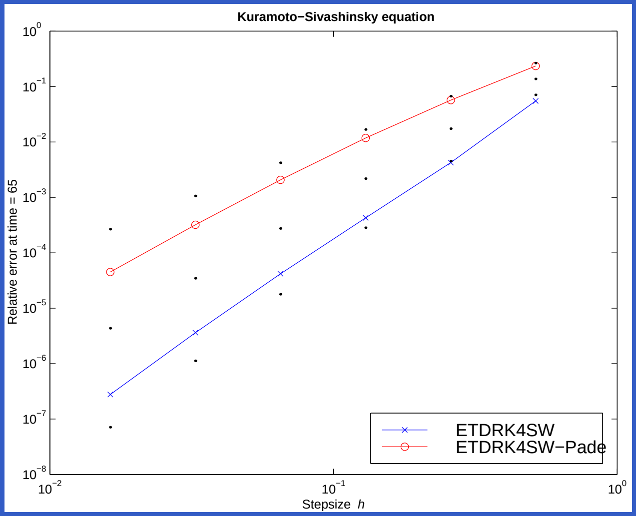
- The order theory is the same
- ARK use rational approximations to the $\exp$ and $\phi^{[i]}$
- The choice of the $\phi^{[i]}$ functions in ARK is restricted to the ETD set.

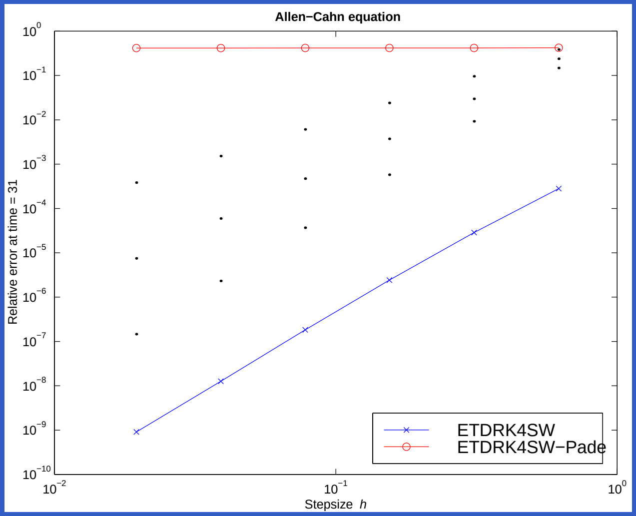




Comparison with ARK Methods

- The order theory is the same
- ARK use rational approximations to the $\exp$ and $\phi^{[i]}$
- The choice of the $\phi^{[i]}$ functions in ARK is restricted to the ETD set.
- Exp. RK methods include IF, ECM and ETDRK methods of Krogstad and Cox and Matthews as special cases.





Connections with CF

We can rewrite the equation (1) in the form

$$u' \; = \; (\mathbf{L}, \mathbf{N}(u,t)).u \; = \; F_{u,t}(u), \;\; u(0) = u_0$$

where . represents the *Lie algebra action*

$$(\mathbf{A}, a).u \; = \; \mathbf{A}u + a$$

Connections with CF

We can rewrite the equation (1) in the form

$$u' \;=\; (\mathbf{L}, \mathbf{N}(u,t)).u \;=\; F_{u,t}(u), \;\; u(0) = u_0$$

where . represents the *Lie algebra action*

$$(\mathbf{A}, a).u \;=\; \mathbf{A}u + a$$

Let $\hat{F}_{\hat{u},\hat{t}}(u)$ be the *Frozen Vector Field* at the point $(\hat{u}, \hat{t})$

$$\hat{F}_{\hat{u},\hat{t}}(u) \;=\; (\mathbf{L}, \mathbf{N}(\hat{u},\hat{t})).u \;=\; \mathbf{L}u + \mathbf{N}(\hat{u},\hat{t})$$

Connections with CF

We can rewrite the equation (1) in the form

$$u' \;=\; (\mathbf{L},\mathbf{N}(u,t)).u \;=\; F_{u,t}(u), \;\; u(0)=u_0$$

where . represents the *Lie algebra action*

$$(\mathbf{A},a).u \;=\; \mathbf{A}u + a$$

Let $\hat{F}_{\hat{u},\hat{t}}(u)$ be the *Frozen Vector Field* at the point $(\hat{u},\hat{t})$

$$\hat{F}_{\hat{u},\hat{t}}(u) \;=\; (\mathbf{L},\mathbf{N}(\hat{u},\hat{t})).u \;=\; \mathbf{L}u + \mathbf{N}(\hat{u},\hat{t})$$

The flow of such vector field is the solution of $u' \;=\; \hat{F}_{\hat{u},\hat{t}}(u), \quad u(0)=u_0$

$$\phi_{t,\hat{F}}(u_0) \;=\; \text{Exp}(t\hat{F}_{\hat{u},\hat{t}}).u_0 \quad = \quad \exp(t\mathbf{L})u_0 + t\phi^{[1]}(t\mathbf{L})\mathbf{N}(\hat{u},\hat{t})$$

CF Method

Algorithm (CF)

for $r = 1 : s$ do

$$Y_r = \text{Exp}(\textstyle\sum_k \alpha^k_{r,J} F_k) \cdots \text{Exp}(\textstyle\sum_k \alpha^k_{r,1} F_k)(p)$$

$$F_r = hF_{Y_r} = h \textstyle\sum_i f_i(Y_r) E_i$$

end

$$y_1 = \text{Exp}(\textstyle\sum_k \beta^k_J F_k) \cdots \text{Exp}(\textstyle\sum_k \beta^k_1 F_k) p$$

Numerical schemes

ETDCF4 of Celledoni, Marthinse and Owren

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2}\phi^{[1]} & & & \\
\frac{1}{2} & 0 & \frac{1}{2}\phi^{[1]} & & \\
\frac{1}{2} & \frac{1}{2}\phi^{[1]} & 0 & 0 & \\
\frac{1}{2} & -\frac{1}{2}\phi^{[1]} & 0 & \phi^{[1]} & \\
\hline
 & \frac{1}{4}\phi^{[1]} & \frac{1}{6}\phi^{[1]} & \frac{1}{6}\phi^{[1]} & -\frac{1}{12}\phi^{[1]} \\
 & -\frac{1}{12}\phi^{[1]} & \frac{1}{6}\phi^{[1]} & \frac{1}{6}\phi^{[1]} & \frac{1}{4}\phi^{[1]}
\end{array}
$$

Numerical schemes

ETDRK4 of Cox and Matthews

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2}\phi^{[1]} & & & \\
\frac{1}{2} & 0 & \frac{1}{2}\phi^{[1]} & & \\
\frac{1}{2} & \frac{1}{2}\phi^{[1]} & 0 & 0 & \\
\frac{1}{2} & -\frac{1}{2}\phi^{[1]} & 0 & \phi^{[1]} & \\
\hline
 & 4\phi^{[3]} - 3\phi^{[2]} + \phi^{[1]} & -4\phi^{[3]} + 2\phi^{[2]} & -4\phi^{[3]} + 2\phi^{[2]} & 4\phi^{[3]} - \phi^{[2]}
\end{array}
$$

Numerical experiments

Example 1: Kuramoto-Sivashinsky equation

$$u_t \;=\; -\,uu_x - u_{xx} - u_{xxxx}\,, \qquad x \in [0, 32\pi]$$

with periodic boundary conditions and with the initial condition

$$u(x,0) = \cos(\frac{x}{16})(1 + \sin(\frac{x}{16})).$$

Numerical experiments

Example 1: Kuramoto-Sivashinsky equation

$$u_t = -uu_x - u_{xx} - u_{xxxx}, \quad x \in [0, 32\pi]$$

with periodic boundary conditions and with the initial condition

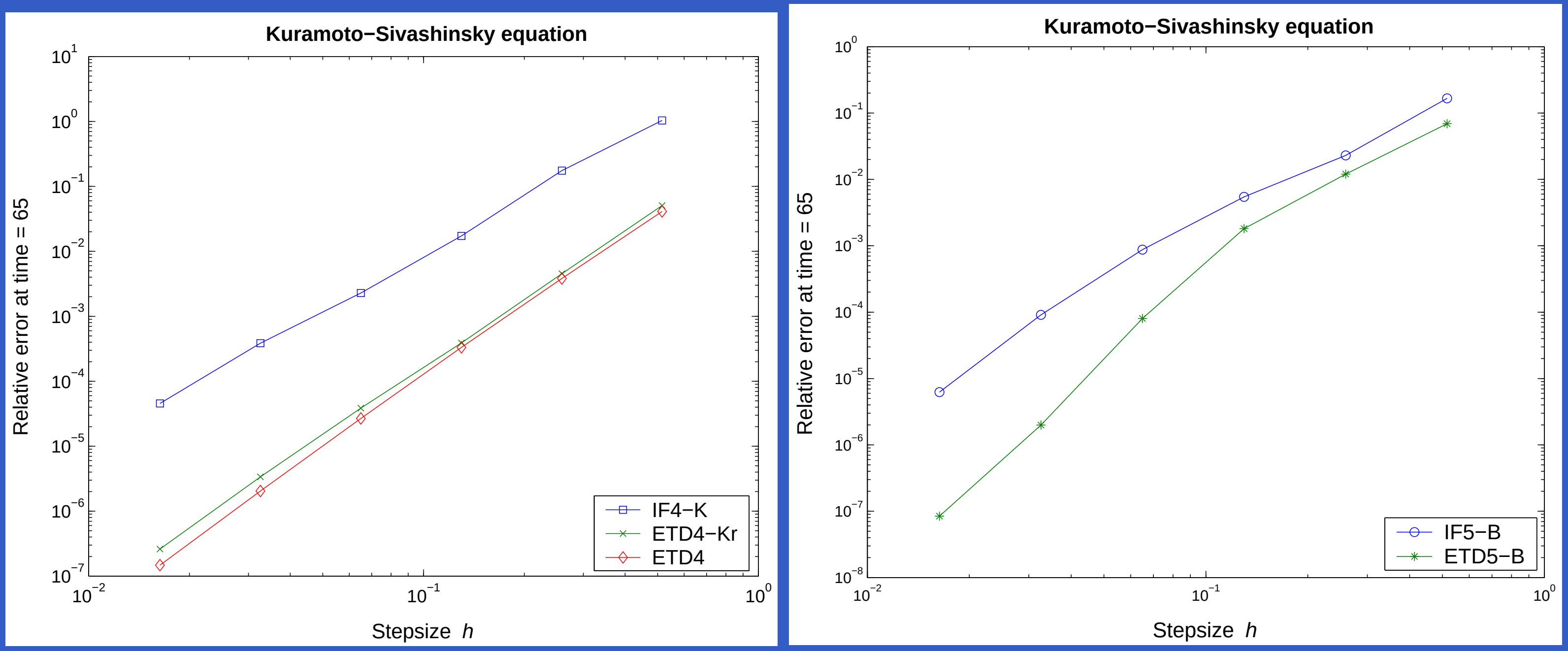
$$u(x,0) = \cos(\frac{x}{16})(1 + \sin(\frac{x}{16})).$$

We discretise the spatial part using Fourier spectral method. The transformed equation in the Fourier space is

$$\hat{u}_t = -\frac{ik}{2}\hat{u^2} + (k^2 - k^4)\hat{u},$$

$$(\mathbf{L}\hat{u})(k) = (k^2 - k^4)\hat{u}(k) \text{ and } \mathbf{N}(\hat{u}, t) = -\tfrac{ik}{2}(F((F^{-1}(\hat{u}))^2))$$

Kuramoto-Sivashinsky equation



Numerical experiments

Example 2: Allen-Cahn equation

$$u_t = \varepsilon u_{xx} + u - u^3, \quad x \in [-1, 1]$$

with $\varepsilon = 0.01$ and with boundary and initial conditions

$$u(-1,t) = -1, \;\; u(1,t) = 1, \quad u(x,0) = .53x + .47\sin(-1.5\pi x)$$

Numerical experiments

Example 2: Allen-Cahn equation

$$u_t = \varepsilon u_{xx} + u - u^3, \quad x \in [-1, 1]$$

with $\varepsilon = 0.01$ and with boundary and initial conditions

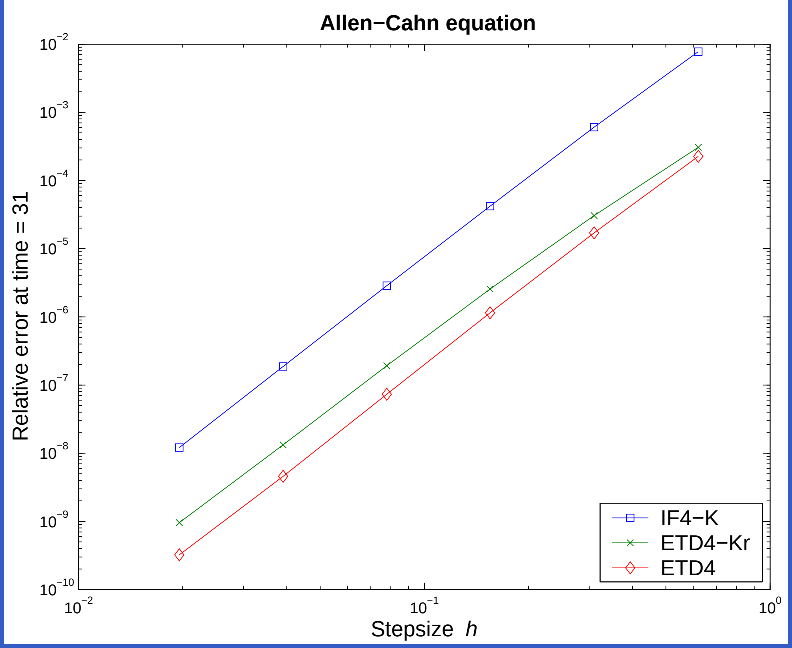
$$u(-1,t) = -1, \;\; u(1,t) = 1, \quad u(x,0) = .53x + .47\sin(-1.5\pi x)$$

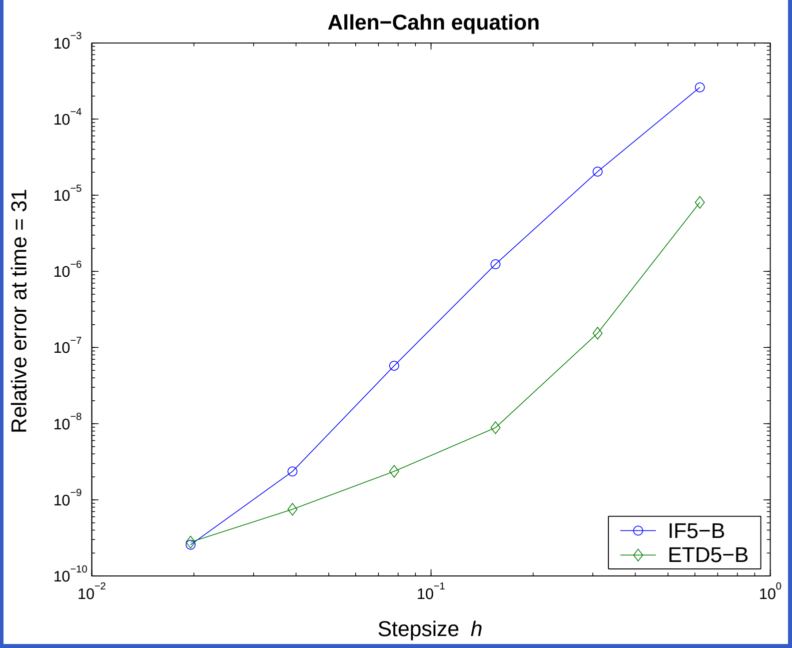
After discretisation in space.

$$u_t = \mathbf{L}u + \mathbf{N}(u(t))$$

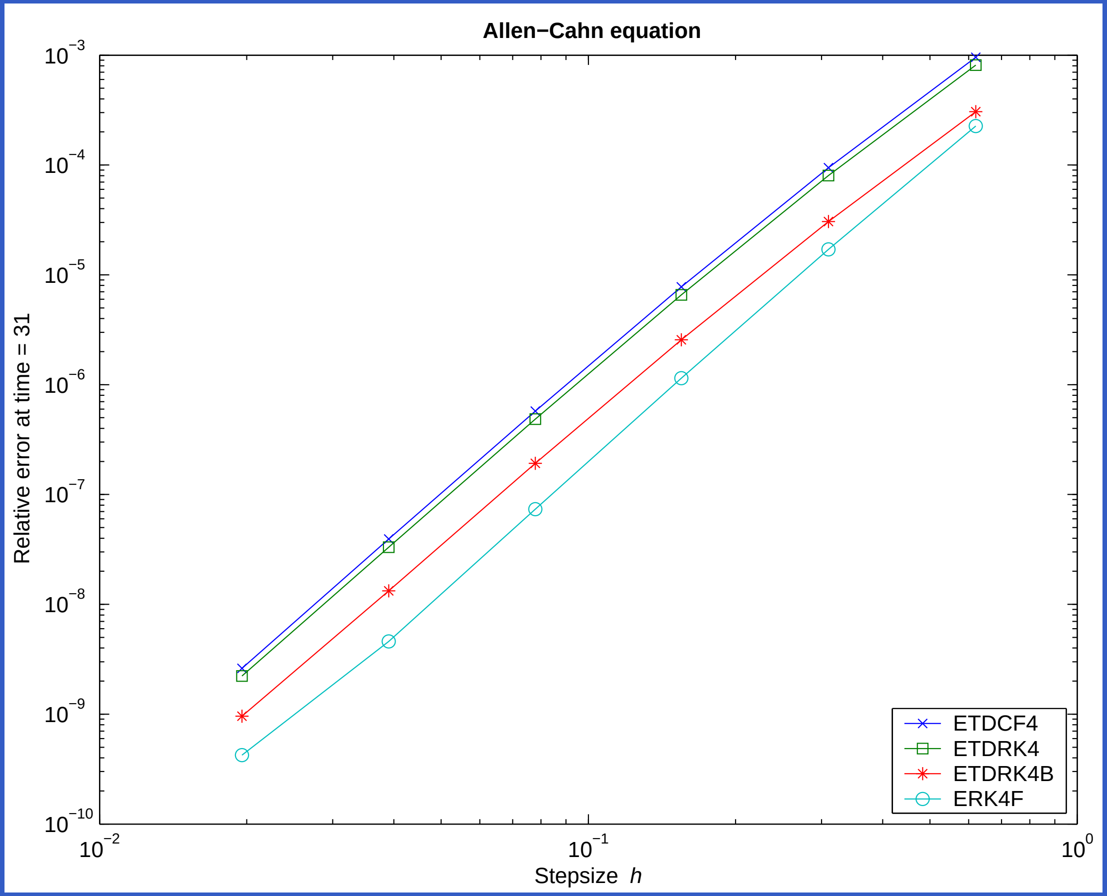
where $\mathbf{L} = \varepsilon D^2$, $\mathbf{N}(u(t)) = u - u^3$ and D is the Chebyshev differentiation matrix

Allen-Cahn equation





Allen-Cahn equation



Numerical experiments

Example 3: Korteweg de Vries equation

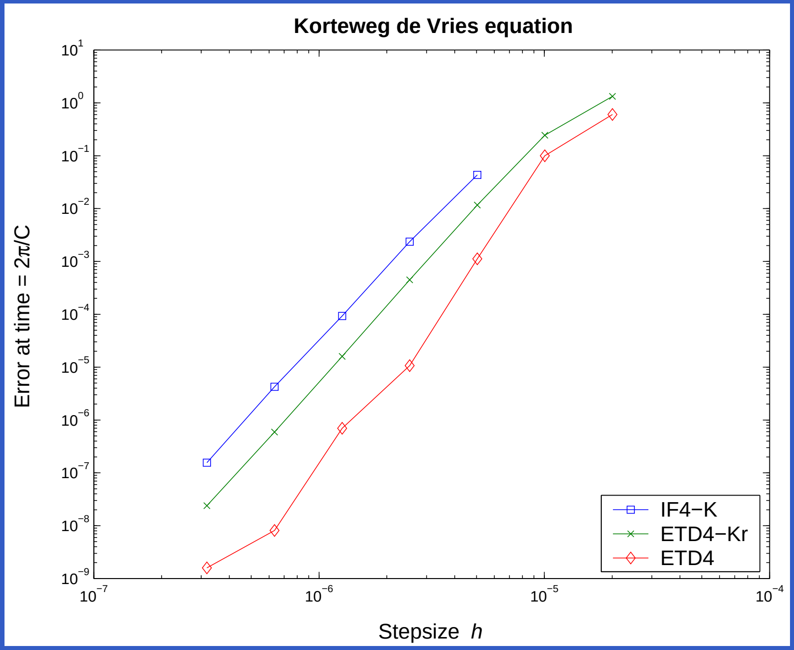
$$u_t = -u_{xxx} - uu_x, \qquad x \in [-\pi, \pi],$$

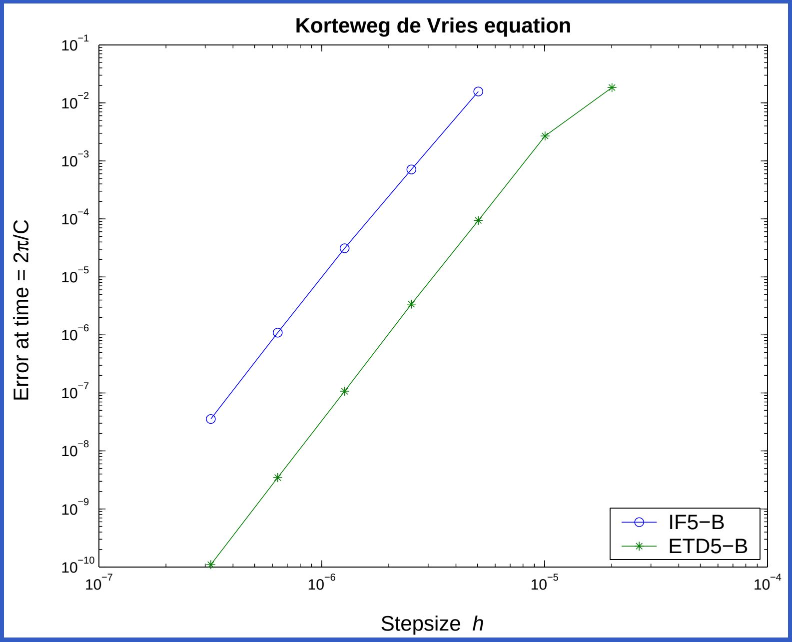
with periodic boundary conditions and with initial condition

$$u(x,0) = 3C/\cosh^2(\sqrt{C}x/2),$$

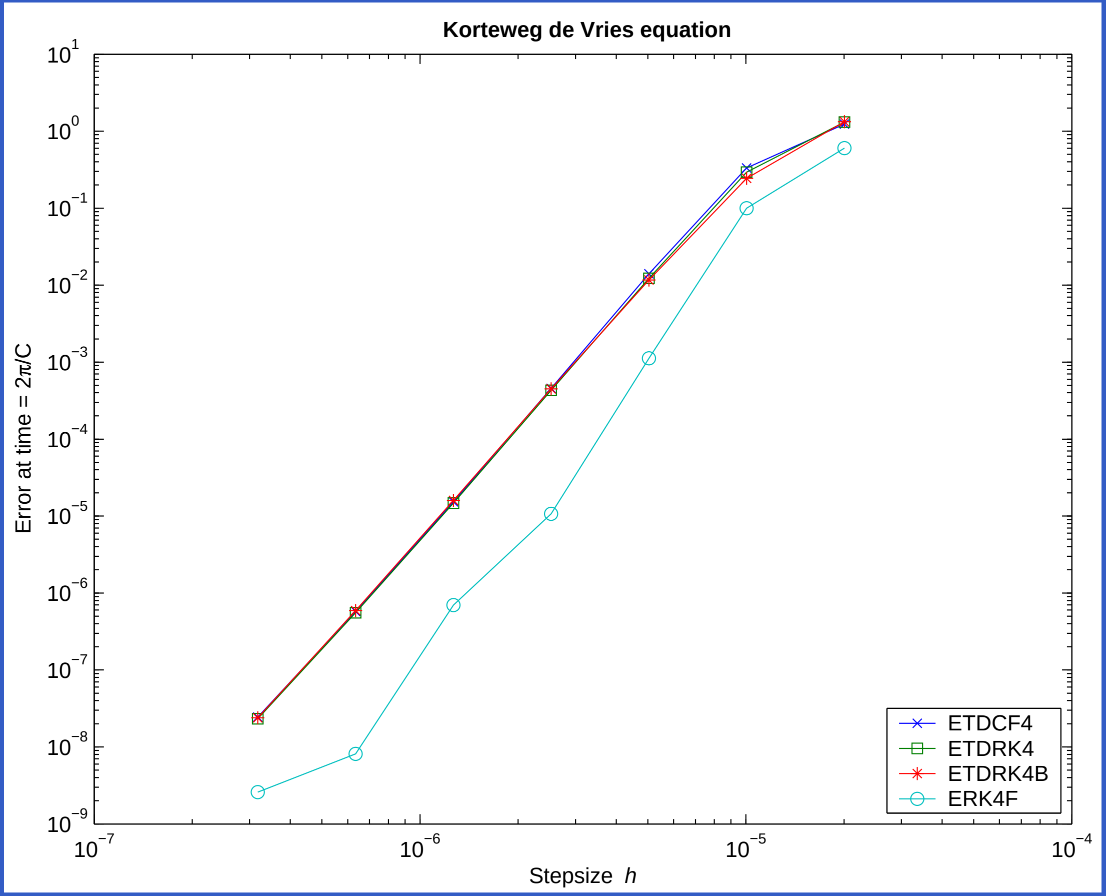
where $C = 625$. The exact solution is $2\pi/C$ periodic and is given by $u(x,t) = u(x - Ct, 0)$. We use a 256-point Fourier spectral discretization in space. In this case the matrix L is again diagonal. The integration in time is done for one period.

KdV equation





KdV equation



Generalized IF Methods

Consider the semi discretised problem (1)

$$u' \ = \ \mathbf{L}u + \mathbf{N}(u(t)), \ u(t_0) = u_0$$

Generalized IF Methods

Consider the semi discretised problem (1)

$$u' \;=\; \mathbf{L}u + \mathbf{N}(u(t)),\; u(t_0) = u_0$$

Change of variables

$$u(t) = \exp(tL)v(t) = \phi_{t,\hat{F}}(v(t)),$$

where $\hat{F}(\hat{u}, t) = L\hat{u}$ approximates F around u_0.
The transformed equation is

$$v'(t) = \exp(-tL)N(\exp(tL)v(t))$$

Generalized IF Methods

Consider the semi discretised problem (1)

$$u' \ = \ \mathbf{L}u + \mathbf{N}(u(t)), \ u(t_0) = u_0$$

In general (Krogstad; Mayday, Patera and Rønquist) substitute

$$u(t) = \phi_{t,\hat{F}}(v(t))$$

where $\hat{F}(\hat{u}, t) = L\hat{u} + \mathsf{N}(\mathsf{t})$ approximates F around u_0.
The transformed equation is

$$v'(t) = \exp(-tL)\big[N(\exp(tL)v(t)) - \mathsf{N}(\mathsf{t})\big]$$

Now apply a numerical method to the transformed equation.

GIF like GLMs

- Information from past was used to capture key features of F.

GIF like GLMs

- Information from past was used to capture key features of F.
- These methods reduced to a methods with maximum stage order when $L = 0$.

GIF like GLMs

- Information from past was used to capture key features of F.
- These methods reduced to a methods with maximum stage order when $L = 0$.
- This is at the cost of smaller stability regions.

GIF like GLMs

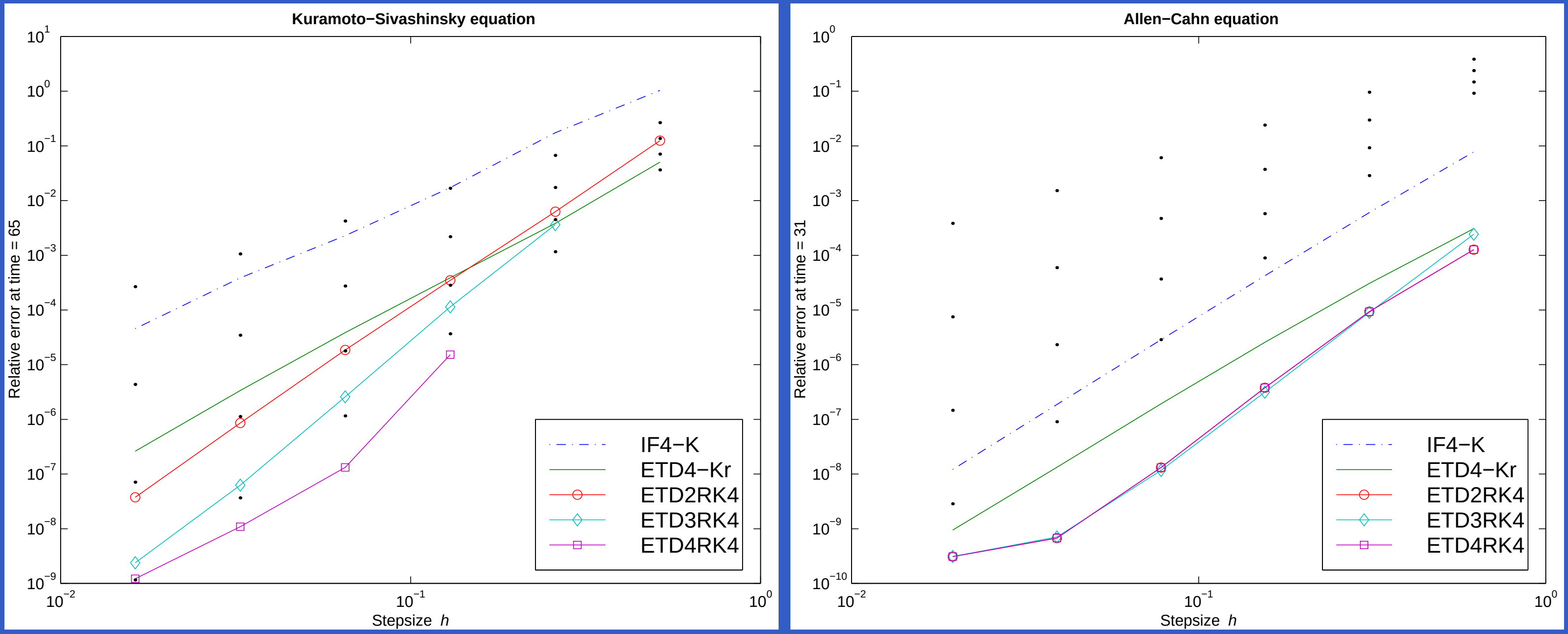
- Information from past was used to capture key features of F.
- These methods reduced to a methods with maximum stage order when $L = 0$.
- This is at the cost of smaller stability regions.
- When $L = 0$ the GIF methods are GLMs.

GIF like GLMs

$$
\left[\begin{array}{c} U_1 \\ U_2 \\ U_3 \\ U_4 \\ \hline u_{n+1} \\ hN_n \\ hN_{n-1} \end{array}\right] = \left[\begin{array}{cccc|ccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{17}{24} & 0 & 0 & 0 & 1 & -\frac{7}{24} & \frac{1}{12} \\ -\frac{11}{48} & \frac{1}{2} & 0 & 0 & 1 & \frac{1}{3} & -\frac{5}{48} \\ \frac{1}{24} & 0 & 1 & 0 & 1 & -\frac{1}{12} & \frac{1}{24} \\ \hline \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \left[\begin{array}{c} hN(U_1) \\ hN(U_2) \\ hN(U_3) \\ hN(U_4) \\ \hline u_n \\ hN_{n-1} \\ hN_{n-2} \end{array}\right]
$$

The GIF showed significant improvements over all other methods.

More Experiments



GIF like Lie-group methods

We can rewrite the equation $u' = \mathbf{L}u + \mathbf{N}(u(t), t),\ u(t_0) = u_0$ in the form

$$\left| \begin{array}{l} \dot{y} = (\mathbf{A}_{u,t}, a).y = \tilde{F}_{u,t}(y) \\ y(0) = y_0 \end{array} \right.$$

where

$$\mathbf{A}_{u,t} = \begin{pmatrix} \mathbf{L} & \frac{\mathbf{N}(u,t)-\mathbf{N}_0}{t} \\ & \\ 0 & 0 \end{pmatrix},\ a = \begin{pmatrix} \mathbf{N}_0 \\ \\ 1 \end{pmatrix},$$

$$y = \begin{pmatrix} u \\ \\ v \end{pmatrix},\ y_0 = \begin{pmatrix} u_0 \\ \\ 0 \end{pmatrix},\ \mathbf{N}_0 = \mathbf{N}(u_0, 0)$$

GIF like Lie-group methods

Let $\tilde{F}_{\hat{u},\hat{t}}(y)$ be the *Frozen Vector Field* at the point $(\hat{u},\hat{t})$

$$\tilde{F}_{\hat{u},\hat{t}}(u) \;=\; (\mathbf{A}_{\hat{u},\hat{t}}, a).y \;=\; \mathbf{A}_{\hat{u},\hat{t}}\, y + a,$$

where

$$\mathbf{A}_{\hat{u},\hat{t}} = \begin{pmatrix} \mathbf{L} & \frac{\hat{\mathbf{N}}-\mathbf{N}_0}{\hat{t}} \\ & \\ 0 & 0 \end{pmatrix}, \quad \hat{\mathbf{N}} = \mathbf{N}(\hat{u},\hat{t})$$

GIF like Lie-group methods

Let $\tilde{F}_{\hat{u},\hat{t}}(y)$ be the *Frozen Vector Field* at the point $(\hat{u},\hat{t})$

$$\tilde{F}_{\hat{u},\hat{t}}(u) \;=\; (\mathbf{A}_{\hat{u},\hat{t}}, a).y \;=\; \mathbf{A}_{\hat{u},\hat{t}}\, y + a,$$

where

$$\mathbf{A}_{\hat{u},\hat{t}} \;=\; \begin{pmatrix} \mathbf{L} & \frac{\hat{\mathbf{N}}-\mathbf{N}_0}{\hat{t}} \\ & \\ 0 & 0 \end{pmatrix}, \quad \hat{\mathbf{N}} \;=\; \mathbf{N}(\hat{u},\hat{t})$$

The flow of such vector field is the solution of $\dot{y} \;=\; \tilde{F}_{\hat{u},\hat{t}}(y), \quad y(0) = y_0$

$$\phi_{t,\tilde{F}}(y_0) = \text{Exp}(t\tilde{F}_{\hat{u},\hat{t}}).y_0 \quad = \quad \begin{pmatrix} \exp(t\mathbf{L})u_0 + t\phi^{[1]}(t\mathbf{L})\mathbf{N}_0 + t^2\phi^{[2]}(t\mathbf{L})\frac{\hat{\mathbf{N}}-\mathbf{N}_0}{\hat{t}} \\ t \end{pmatrix}$$

GIF like Lie-group methods

Let $\tilde{F}_{\hat{u},\hat{t}}(y)$ be the *Frozen Vector Field* at the point $(\hat{u},\hat{t})$

$$\tilde{F}_{\hat{u},\hat{t}}(u) \;=\; (\mathbf{A}_{\hat{u},\hat{t}}, a).y \;=\; \mathbf{A}_{\hat{u},\hat{t}}\, y + a,$$

where

$$\mathbf{A}_{\hat{u},\hat{t}} \;=\; \begin{pmatrix} \mathbf{L} & \frac{\hat{\mathbf{N}}-\mathbf{N}_0}{\hat{t}} \\ & \\ 0 & 0 \end{pmatrix}, \quad \hat{\mathbf{N}} \;=\; \mathbf{N}(\hat{u},\hat{t})$$

The flow of such vector field is the solution of $\dot{y} \;=\; \tilde{F}_{\hat{u},\hat{t}}(y), \quad y(0) = y_0$

$$\phi_{t,\tilde{F}}(y_0) = \text{Exp}(t\tilde{F}_{\hat{u},\hat{t}}).y_0 \quad = \quad \begin{pmatrix} \exp(t\mathbf{L})u_0 + t\phi^{[1]}(t\mathbf{L})\mathbf{N}_0 + t^2\phi^{[2]}(t\mathbf{L})\frac{\hat{\mathbf{N}}-\mathbf{N}_0}{\hat{t}} \\ t \end{pmatrix}$$

and satisfies

$$\phi_{\alpha t,\tilde{F}} \;=\; \phi_{t,\alpha\tilde{F}} \qquad \phi_{\alpha t,\tilde{F}} \circ \phi_{\beta t,\tilde{F}} \;=\; \phi_{(\alpha+\beta)t,\tilde{F}}$$

Conclusion

- The exponential Runge-Kutta methods designed to solve stiff semi discretised PDEs were introduced.

Conclusion

- The exponential Runge-Kutta methods designed to solve stiff semi discretised PDEs were introduced.
- The order theory were rederived in a more general settings.

Conclusion

- The exponential Runge-Kutta methods designed to solve stiff semi discretised PDEs were introduced.
- The order theory were rederived in a more general settings.
- The IF, ECM (ETDRK) methods are special cases of these methods.

Work in progress

- Other $\phi^{[i]}$ functions

Work in progress

- Other $\phi^{[i]}$ functions
- Effective ways of their computation

Work in progress

- Other $\phi^{[i]}$ functions
- Effective ways of their computation
- Construct alternative exponential general linear methods

Work in progress

- Other $\phi^{[i]}$ functions
- Effective ways of their computation
- Construct alternative exponential general linear methods
- B-convergence and stability results

Work in progress

- Other $\phi^{[i]}$ functions
- Effective ways of their computation
- Construct alternative exponential general linear methods
- B-convergence and stability results

References

- J. Certaine, *The solution of ordinary differential equations with large time constants,* Math. Meth. Dig. Comp., 129-132 (1960).
- Y. Maday, A. T. Patera and E. M. Rønquist, *An operator-integration-factor splitting method for time dependant problems: Application to incompressible fluid flow,* J. Sci. Comput., 263-292 (1990).
- J. D. Lawson, *Generalized Runge-Kutta processes for satble systems with large Lipschitz constants,* SIAM J. Num. Anal., 4, 372-380 (1967).
- S.P. Nørsett, *An A-Stable modification of the Adams-Bashforth methods,* Lecture Note in Mathematics, 109, 214-219 (1969).
- A. Friedli, *Verallgemeinerte Runge-Kutta Verfahren zur Lösung steifer Differential gleichundssysteme,* Lecture Note in Mathematics, 631, (1978).
- T. Steihaug A. Wolfbrandt, *An Attempt to Avoid Exact Jacobian and Nonlinear Equations in the Numerical Solution of Stiff Differential Equations,* Mathematics of computation, 33, 521-534 (1979).
- R. Strehmel, R. Weiner, *B-convergence results for linearly implicit one step methods,* BIT, 27, 264-282 (1987).

References

- M. Hochbruck, Ch. Lubich, H. Selhofer, *Exponential integrators for large systems of differential equations,* SIAM J. Sci. Comput., 19, 1552–1574 (1998).
- G. Beylkin, J. M. Keiser, and L. Vozovoi, *A new class of time discretization schemes for the solution of nonlinear PDEs,* J. Comput. Phys. 147, 362–387 (1998).
- S.M. Cox, P.C. Matthews, Exponential time differencing for stiff systems, *J. Comp. Phys.* 176, 430-455 (2002)
- A.K. Kassam, L.N. Trefethen, Fourth-order time stepping for stiff PDEs, Submitted to SIAM J. Sci. Comput. (2002)
- E. Celledoni, A. Martinsen, B. Owren, Commutator-free lie group methods, *FGCS* 19(3), 341-352 (2003)
- S. Krogstad, Generalized integrating factor methods for stiff PDEs, available at: `http://www.ii.uib.no/~stein`
- M. Hochbruck, A. Osterman, Exponential Runge-Kutta methods for parabolic problems, available at: `http://techmath.uibk.ac.at/numbau/alex/publications.html`