# Exponential Integrators - History and Recent Developments

University of Düsseldorf

18-19 October, 2004, Germany.

Borislav V. Minchev

Borko.Minchev@ii.uib.no

http://www.ii.uib.no/∼borko

Joint work with Will Wright

University of Bergen, Norway

# Outline

- Introduction and Motivation
- Main classes of exponential integrators
  - Exponential linear multistep methods
  - Exponentila Runge–Kutta methods
  - Exponential general linear methods
- Exponential integrators for parabolic PDEs
- Exponential integrators and Schrödinger equation
- Exponential integrators for oscillatory problems
- Implementation issues
- Conclusions
- Open problems

# Introduction and Motivation

First exponential integrators were introduced as an alternative approach for solving *stiff problems*

- Certain'60 - multistep type
- Lawson'69 - multistage type

# Introduction and Motivation

First exponential integrators were introduced as an alternative approach for solving

*stiff problems*

- Certain'60 - multistep type
- Lawson'69 - multistage type

A new interest in exponential integrators for semilinear problems

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

where $u : \mathbb{R} \to \mathbb{R}^d, \ L \in \mathbb{R}^{d\times d}, N : \mathbb{R}^d \to \mathbb{R}^d$ and $d$ is a discretization parameter equal to the number of spatial grid points.

# Exponential multistep methods

- $IF$ = Integrating Factor methods (Lawson)

# Exponential multistep methods

- $IF$ = Integrating Factor methods (Lawson)
- $ETD$ = Exponential Time Differencing metods (Certain)

# Exponential multistep methods

- $IF$ = Integrating Factor methods (Lawson)

Recall

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

# Exponential multistep methods

- $IF$ = Integrating Factor methods (Lawson)

Recall

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

Solve exactly the linear part and then make a change of variables (also known as Lawson transformation)

$$v(t) = e^{-tL}u(t).$$

# Exponential multistep methods

- $IF$ = Integrating Factor methods (Lawson)

Recall

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

Solve exactly the linear part and then make a change of variables (also known as Lawson transformation)

$$v(t) = e^{-tL} u(t).$$

The initial value problem written in the new variable is then given by

$$v'(t) = e^{-tL} N(e^{tL} v(t), t) = \mathrm{g}(v,t), \qquad v(t_0) = v_0,$$

where $v_0 = e^{-t_0 L} u_0$.

# Integrating Factor

Consider the Jacobian of the transformed equation

$$\frac{\partial \mathrm{g}}{\partial v} = e^{-tL}\,\frac{\partial N}{\partial u}\,e^{tL},$$

Since $e^{-tL} = (e^{tL})^{-1}$, it follows that the eigenvalues of $\partial \mathrm{g}/\partial v$ are those of $\partial N/\partial u$.

# Integrating Factor

Consider the Jacobian of the transformed equation

$$\frac{\partial \mathrm{g}}{\partial v} = e^{-tL}\,\frac{\partial N}{\partial u}\,e^{tL},$$

Since $e^{-tL} = (e^{tL})^{-1}$, it follows that the eigenvalues of $\partial \mathrm{g}/\partial v$ are those of $\partial N/\partial u$.
The idea now is to apply any numerical method on the transformed equation and then to transform back the result into the original variable.

# Integrating Factor

Consider the Jacobian of the transformed equation

$$\frac{\partial \mathrm{g}}{\partial v} = e^{-tL}\,\frac{\partial N}{\partial u}\,e^{tL},$$

Since $e^{-tL} = (e^{tL})^{-1}$, it follows that the eigenvalues of $\partial \mathrm{g}/\partial v$ are those of $\partial N/\partial u$.
For example:
IF Euler method is

$$u_n = e^{hL}u_{n-1} + e^{hL}hN_{n-1},$$

where $h$ represents the stepsize of the method and $N_{n-1} = N(u_{n-1}, t_{n-1})$.

# Integrating Factor

Consider the Jacobian of the transformed equation

$$\frac{\partial \mathrm{g}}{\partial v} = e^{-tL}\,\frac{\partial N}{\partial u}\,e^{tL},$$

Since $e^{-tL} = (e^{tL})^{-1}$, it follows that the eigenvalues of $\partial \mathrm{g}/\partial v$ are those of $\partial N/\partial u$.
For example:
IF Euler method is

$$u_n = e^{hL}u_{n-1} + e^{hL}hN_{n-1},$$

where $h$ represents the stepsize of the method and $N_{n-1} = N(u_{n-1}, t_{n-1})$.
IF implicit Euler method is

$$u_n = e^{hL}u_{n-1} + e^{hL}hN_n.$$

# More IF multistep methods

Similarly $k$-step IF Adams methods are defined as

$$u_n = e^{hL} u_{n-1} + \sum_{i=0}^{k} \beta_i e^{ihL} h N_{n-i},$$

where $\beta_i$ are the coefficients of the Adams method and $N_{n-i} = N(u_{n-i}, t_{n-i})$ for $i = 0, 1, 2, \ldots, k$.

# More IF multistep methods

Similarly $k$-step IF Adams methods are defined as

$$u_n = e^{hL} u_{n-1} + \sum_{i=0}^{k} \beta_i e^{ihL} h N_{n-i},$$

where $\beta_i$ are the coefficients of the Adams method and $N_{n-i} = N(u_{n-i}, t_{n-i})$ for $i = 0, 1, 2, \ldots, k$.
IF BDF methods are defined as

$$u_n = \sum_{i=1}^{k} \alpha_i e^{ihL} u_{n-i} + \beta_0 h N_n,$$

where $\beta_0$ and $\alpha_i$ are the coefficients of the underlying BDF method.

# ETD multistep methods

Similar approach to the $IF$ methods, but we do not make a complete change of variables. Premultiplying the original problem (1) by the integrating factor $e^{-tL}$ we get

$$\begin{aligned} e^{-tL}u' &= e^{-tL}Lu + e^{-tL}N(u,t), \\ (e^{-tL}u)' &= e^{-tL}N(u,t). \end{aligned}$$

# ETD multistep methods

Similar approach to the $IF$ methods, but we do not make a complete change of variables. Premultiplying the original problem (1) by the integrating factor $e^{-tL}$ we get

$$\begin{aligned} e^{-tL}u' &= e^{-tL}Lu + e^{-tL}N(u,t), \\ (e^{-tL}u)' &= e^{-tL}N(u,t). \end{aligned}$$

Integrating the last equation between $t_{n-1}$ and $t_n = t_{n-1} + h$, we obtain

$$\text{(vcf)} \quad u(t_{n-1}+h) = e^{hL}u_{n-1} + \int_0^h e^{(h-\tau)L} N(u(t_{n-1}+\tau), t_{n-1}+\tau)\mathrm{d}\tau.$$

The approach now is to replace the nonlinear term in the variation of constants formulae by a Newton interpolation polynomial and then solve the resulting integral exactly.

# ETD multistep methods

Similar approach to the $IF$ methods, but we do not make a complete change of variables. Premultiplying the original problem (1) by the integrating factor $e^{-tL}$ we get

$$\begin{aligned} e^{-tL}u' &= e^{-tL}Lu + e^{-tL}N(u,t), \\ (e^{-tL}u)' &= e^{-tL}N(u,t). \end{aligned}$$

Integrating the last equation between $t_{n-1}$ and $t_n = t_{n-1} + h$, we obtain

$$\text{(vcf)} \quad u(t_{n-1}+h) = e^{hL}u_{n-1} + \int_0^h e^{(h-\tau)L} N(u(t_{n-1}+\tau), t_{n-1}+\tau) \mathrm{d}\tau.$$

The approach now is to replace the nonlinear term in the variation of constants formulae by a Newton interpolation polynomial and then solve the resulting integral exactly.

- When $N(u(t_{n-1}+\tau), t_{n-1}+\tau) \approx N_{n-1}$ we obtain the ETD Euler method

$$u_n = e^{hL}u_{n-1} + \phi^{[1]}(hL)hN_{n-1},$$

where $\phi^{[1]}(z)$ is

$$\phi^{[1]}(z) = \frac{e^z - 1}{z}.$$

# ETD multistep methods

Similar approach to the $IF$ methods, but we do not make a complete change of variables. Premultiplying the original problem (1) by the integrating factor $e^{-tL}$ we get

$$\begin{aligned} e^{-tL}u' &= e^{-tL}Lu + e^{-tL}N(u,t), \\ (e^{-tL}u)' &= e^{-tL}N(u,t). \end{aligned}$$

Integrating the last equation between $t_{n-1}$ and $t_n = t_{n-1} + h$, we obtain

$$\text{(vcf)} \quad u(t_{n-1}+h) = e^{hL}u_{n-1} + \int_0^h e^{(h-\tau)L} N(u(t_{n-1}+\tau), t_{n-1}+\tau)\mathrm{d}\tau.$$

The approach now is to replace the nonlinear term in the variation of constants formulae by a Newton interpolation polynomial and then solve the resulting integral exactly.

- When $N(u(t_{n-1}+\tau), t_{n-1}+\tau) \approx N_{n-1}$ we obtain the ETD Euler method

$$u_n = e^{hL}u_{n-1} + \phi^{[1]}(hL)hN_{n-1},$$

- In general, using higher order approximations to the nonlinear part $N$ we obtain

  ETD Adams–Bashforth (Nørsett'69,..., Cox–Matthews'02).

# An alternative approach

An alternative approach for deriving exponential Adams methods, both explicit and implicit, is to use the following result (Beylkin at al.'98)

**Lemma 1.** *The exact solution of the initial value problem*

$$u' = Lu + N(u,t), \qquad u(t_{n-1}) = u_{n-1},$$

*can be expressed in the form*

$$u(t_{n-1}+h) = e^{hL}u_{n-1} + \sum_{i=0}^{\infty} h^{i+1}\phi^{[i+1]}(hL)N_{n-1}^{(i)},$$

*where* $N_{n-1}^{(i)} = \frac{d^i}{dt^i}\Big|_{t=t_{n-1}} N(u(t),t)$, *and* $\phi^{[i]}(z)$ *are recursively defined as*

$$\phi^{[0]}(z) = e^z, \quad \phi^{[i+1]}(z) = \frac{\phi^{[i]}(z) - \frac{1}{i!}}{z}, \quad \text{for } i = 0,1,2,\ldots.$$

# ETD Adams methods

The exact solution

$$u(t_{n-1}+h) = e^{hL}u_{n-1} + \sum_{i=0}^{\infty} h^{i+1}\phi^{[i+1]}(hL)N_{n-1}^{(i)}.$$

Its numerical approximation

$$u_n = e^{hL}u_{n-1} + h\sum_{l=0}^{k}\beta_l N_{n-l},$$

where $\beta_0, \beta_1, \ldots, \beta_k$ are coefficients which have to be computed by expanding in Taylor series the nonlinear terms.

# ETD Adams methods

$$u_n = e^{hL} u_{n-1} + h \sum_{l=0}^{k} \beta_l N_{n-l},$$

| k | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\beta_4$ |
|---|---|---|---|---|
| 1 | $\phi^{[1]}$ | 0 | 0 | 0 |
| 2 | $\phi^{[1]} + \phi^{[2]}$ | $-\phi^{[2]}$ | 0 | 0 |
| 3 | $\phi^{[1]} + \frac{3}{2}\phi^{[2]} + \phi^{[3]}$ | $-2(\phi^{[2]} + \phi^{[3]})$ | $\frac{1}{2}\phi^{[2]} + \phi^{[3]}$ | 0 |
| 4 | $\phi^{[1]} + \frac{11}{6}\phi^{[2]} + 2\phi^{[3]} + \phi^{[4]}$ | $-3\phi^{[2]} - 5\phi^{[3]} - 3\phi^{[4]}$ | $\frac{3}{2}\phi^{[2]} + 4\phi^{[3]} + 3\phi^{[4]}$ | X |

where X $= -\frac{1}{3}\phi^{[2]} - \phi^{[3]} - \phi^{[4]}$

Coefficients of ETD Adams–Bashforth methods.

# ETD Adams methods

$$u_n = e^{hL} u_{n-1} + h \sum_{l=0}^{k} \beta_l N_{n-l},$$

| k | $\beta_0$ | $\beta_1$ | $\beta_2$ | $\beta_3$ |
|---|---|---|---|---|
| 0 | $\phi^{[1]}$ | 0 | 0 | 0 |
| 1 | $\phi^{[2]}$ | $\phi^{[1]} - \phi^{[2]}$ | 0 | 0 |
| 2 | $\frac{1}{2}\phi^{[2]} + \phi^{[3]}$ | $\phi^{[1]} - 2\phi^{[3]}$ | $-\frac{1}{2}\phi^{[2]} + \phi^{[3]}$ | 0 |
| 3 | $\frac{1}{3}\phi^{[2]} + \phi^{[3]} + \phi^{[4]}$ | $\phi^{[1]} + \frac{1}{2}\phi^{[2]} - 2\phi^{[3]} - 3\phi^{[4]}$ | $-\phi^{[2]} + \phi^{[3]} + 3\phi^{[4]}$ | X |

where X $= \frac{1}{6}\phi^{[2]} - \phi^{[4]}$

Coefficients of ETD Adams–Moulton methods.

# ETD Adams methods

$$u_n = e^{hL} u_{n-1} + h \sum_{l=0}^{k} \beta_l N_{n-l},$$

| k | $\beta_0$ | $\beta_1$ | $\beta_2$ | $\beta_3$ |
|---|---|---|---|---|
| 0 | $\phi^{[1]}$ | 0 | 0 | 0 |
| 1 | $\phi^{[2]}$ | $\phi^{[1]} - \phi^{[2]}$ | 0 | 0 |
| 2 | $\frac{1}{2}\phi^{[2]} + \phi^{[3]}$ | $\phi^{[1]} - 2\phi^{[3]}$ | $-\frac{1}{2}\phi^{[2]} + \phi^{[3]}$ | 0 |
| 3 | $\frac{1}{3}\phi^{[2]} + \phi^{[3]} + \phi^{[4]}$ | $\phi^{[1]} + \frac{1}{2}\phi^{[2]} - 2\phi^{[3]} - 3\phi^{[4]}$ | $-\phi^{[2]} + \phi^{[3]} + 3\phi^{[4]}$ | X |

where X $= \frac{1}{6}\phi^{[2]} - \phi^{[4]}$

Coefficients of ETD Adams–Moulton methods.

It is not possible to construct ETD BDF methods !

# Exponential Runge–Kuta methods

IF Runge–Kuta methods (Lawson'69)

For simplicity, we represent the initial value problem (1) in autonomous form

$$u' \; = \; Lu + N(u(t)), \quad u(t_0) = u_0.$$

Similarly to the the multistep case, the idea now is to apply an arbitrary $s$-stage Runge–Kutta method to the transformed equation

$$v'(t) = e^{-tL} N(e^{tL} v(t)) = \mathrm{g}(v), \qquad v(t_0) = v_0,$$

and then to transform back the result into the original variable. If $\mathcal{A} = (\alpha_{ij})$, $b = (\beta_i)$ and $c = (c_i)$ are the coefficients of the underlying multistage method then in terms of the original variable the computations performed are

# Exponential Runge–Kuta methods

IF Runge–Kuta methods (Lawson'69)

$$\begin{aligned}
U_1 &= u_0 \\
U_2 &= e^{c_2 hL}(u_0 + \alpha_{21} hN(U_1)) \\
U_3 &= e^{c_3 hL}(u_0 + \alpha_{31} hN(U_1) + \alpha_{32} h e^{-c_2 hL} N(U_2)) \\
U_4 &= e^{c_4 hL}(u_0 + \alpha_{41} hN(U_1) + \alpha_{42} h e^{-c_2 hL} N(U_2) \\
&\quad + \alpha_{43} h e^{-c_3 hL} N(U_3)) \\
u_1 &= e^{hL}(u_0 + \beta_1 hN(U_1) + \beta_2 h e^{-c_2 hL} N(U_2) \\
&\quad + \beta_3 h e^{-c_3 hL} N(U_3) + \beta_4 h e^{-c_4 hL} N(U_4))
\end{aligned}$$

# Exponential Runge–Kuta methods

IF Runge–Kuta methods (Lawson'69)

General form of an order 4 integrating factor method is

$$\left[\begin{array}{cccc|c} 0 & 0 & 0 & 0 & e^{c_1 hL} \\ \alpha_{21} e^{c_2 hL} & 0 & 0 & 0 & e^{c_2 hL} \\ \alpha_{31} e^{c_3 hL} & \alpha_{32} e^{(c_3 - c2) hL} & 0 & 0 & e^{c_3 hL} \\ \alpha_{41} e^{c_4 hL} & \alpha_{42} e^{(c_3 - c2) hL} & \alpha_{43} e^{(c_4 - c3) hL} & 0 & e^{c_4 hL} \\ \hline \beta_1 e^{hL} & \beta_2 e^{(1 - c_2) hL} & \beta_3 e^{(1 - c_3) hL} & \beta_4 e^{(1 - c_4) hL} & e^{hL} \end{array}\right]$$

# Exponential Runge–Kutta methods

IF Runge–Kuta methods (Lawson'69)

General form of an order 4 integrating factor method is

$$\left[\begin{array}{cccc|c} 0 & 0 & 0 & 0 & e^{c_1 hL} \\ \alpha_{21} e^{c_2 hL} & 0 & 0 & 0 & e^{c_2 hL} \\ \alpha_{31} e^{c_3 hL} & \alpha_{32} e^{(c_3 - c2) hL} & 0 & 0 & e^{c_3 hL} \\ \alpha_{41} e^{c_4 hL} & \alpha_{42} e^{(c_3 - c2) hL} & \alpha_{43} e^{(c_4 - c3) hL} & 0 & e^{c_4 hL} \\ \hline \beta_1 e^{hL} & \beta_2 e^{(1-c_2) hL} & \beta_3 e^{(1-c_3) hL} & \beta_4 e^{(1-c_4) hL} & e^{hL} \end{array}\right]$$

- Uniformly distributed $c$ vector provides cheapest methods.
- This structure requires only classical order conditions.
- IF RK methods perform poorly for stiff problems.

# History of ETD RK methods

ETD Runge–Kuta methods

- Friedli'78
  - Explicit ETD RK methods, nonstiff order 5 conditions
- Steihaug and Wolfbrand'79
  - $W$ methods, order conditions
- Hairer, Bader and Lubich'82
  - Linearly implicit methods
- Strehmel and Weiner'82
  - Adaptive RK methods, order theory, B-stability

# History of ETD RK methods

ETD Runge–Kuta methods

- Friedli'78

  - Explicit ETD RK methods, nonstiff order 5 conditions
- Steihaug and Wolfbrand'79

  - $W$ methods, order conditions
- Hairer, Bader and Lubich'82

  - Linearly implicit methods
- Strehmel and Weiner'82

  - Adaptive RK methods, order theory, B-stability

This methods can be seen as exponential integrators which use Padé approximations to the exponential and the related functions

# History of ETD RK methods

Pure exponential Runge–Kuta methods

- Hochbruck and Lubich'97
  - Rosenbrock-like exponential integrators, which use the first ETD function $\phi^{[1]}$
- Munthe-Kaas'99
  - Runge–Kutta Munthe-Kaas (RKMK) methods with affine action
- Celledoni Marthinsen and Owren'03
  - Commutator Free (CF) Lie group methods
- Krogstad'03
  - Generalized IF methods, connection with CF
- Hochbruck and Osterman'04
  - Exponential collocation methods, convergence analyze for parabolic PDEs

# General format of Exp. RK methods

Aims:

- Construct a class of exponential integrators which overcome the stiffness by using a set of precomputed functions $\phi^{[l]}$ along with evaluations of the nonlinear part of the differential equation.
- Include all exponential integrators of Runge–Kutta type in one framework.
- Derive the nonstiff order theory for this class of method

# The $\phi$ functions

- The IF $\phi$ functions are

$$\phi^{[i]}(c_j)(hL) = e^{(c_i - c_j)hL} \quad i = 1, 2, \ldots .$$

# The $\phi$ functions

- The IF $\phi$ functions are

$$\phi^{[i]}(c_j)(hL) = e^{(c_i - c_j)hL} \quad i = 1, 2, \ldots .$$

- The ETD $\phi$ functions are

$$\begin{aligned} \phi^{[0]}(c_j)(hL) &= e^{c_j hL}, \\ \phi^{[i]}(c_j)(hL) &= \frac{\phi^{[i-1]}(c_j)(hL) - \frac{1}{(i-1)!}}{c_j hL} \end{aligned}$$

# The $\phi$ functions

- The IF $\phi$ functions are

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

- The ETD $\phi$ functions are

$$\phi^{[i]}(c_j)(hL) = \frac{1}{i!}I + \frac{c_j}{(i+1)!}hL + \frac{c_j^2}{(i+2)!}(hL)^2 + \cdots \quad i = 1, 2, \ldots.$$

# The $\phi$ functions

In general, for $l \in \mathbb{N}$ and $\lambda \in \mathbb{R}$, the $\phi^{[l]}$ functions could be

$$\phi^{[l]}(\lambda)(hL) = \sum_{j \geq 0} \phi_j^{[l]}(\lambda)(hL)^j,$$

# The $\phi$ functions

In general, for $l \in \mathbb{N}$ and $\lambda \in \mathbb{R}$, the $\phi^{[l]}$ functions could be

$$\phi^{[l]}(\lambda)(hL) = \sum_{j \geq 0} \phi_j^{[l]}(\lambda)(hL)^j,$$

and must:

- Be computed exactly or to arbitrary high order cheaply
- Map the spectrum of $hL$ to a bounded region

Given the IF and ETD $\phi$ functions as basis elements then
- linear combinations
- products
- inverses
produce mehods.

# The $\phi$ functions

In general, for $l \in \mathbb{N}$ and $\lambda \in \mathbb{R}$, the $\phi^{[l]}$ functions could be

$$\phi^{[l]}(\lambda)(hL) = \sum_{j\geq 0} \phi_j^{[l]}(\lambda)(hL)^j,$$

and must:

- Be computed exactly or to arbitrary high order cheaply
- Map the spectrum of $hL$ to a bounded region

Given the IF and ETD $\phi$ functions as basis elements then
- linear combinations
- products
- inverses
produce mehods.
Other choices are also posible (approximations with trigonometric polynomials in vcf).
The exact structure of $\phi^{[l]}$, which leads to methods is still unclear!

# Formulation of the methods

The computations performed are

$$
\begin{aligned}
U_i &= \sum_{j=1}^{s} \sum_{l=1}^{m} \alpha_{ij}^{[l]} \phi^{[l]}(c_i)(hL) h N(U_j) + e^{c_i hL} u_{n-1}, \\
u_n &= \sum_{j=1}^{s} \sum_{l=1}^{m} \beta_{j}^{[l]} \phi^{[l]}(1)(hL) h N(U_j) + e^{hL} u_{n-1},
\end{aligned}
$$

where $m$ puts a limit on the number of $\phi^{[l]}$ functions which can be computed, $h$ represents the stepsize and $U_i$ denotes the internal stage approximation.

# Formulation of the methods

The computations performed are

$$
\begin{aligned}
U_i &= \sum_{j=1}^{s}\sum_{l=1}^{m} \alpha_{ij}^{[l]} \phi^{[l]}(c_i)(hL)hN(U_j) + e^{c_i hL} u_{n-1}, \\
u_n &= \sum_{j=1}^{s}\sum_{l=1}^{m} \beta_{j}^{[l]} \phi^{[l]}(1)(hL)hN(U_j) + e^{hL} u_{n-1},
\end{aligned}
$$

where $m$ puts a limit on the number of $\phi^{[l]}$ functions which can be computed, $h$ represents the stepsize and $U_i$ denotes the internal stage approximation.

Interpreted in a Runge–Kutta type tableau

$$
\begin{array}{c|c|c|c|c|c}
 & \phi^{[1]} & \phi^{[2]} & & \phi^{[m-1]} & \phi^{[m]} \\
c & \alpha^{[1]} & \alpha^{[2]} & \cdots & \alpha^{[m-1]} & \alpha^{[m]} \\
\hline
 & \beta^{[1]^T} & \beta^{[2]^T} & \cdots & \beta^{[m-1]^T} & \beta^{[m]^T}
\end{array}
$$

# Nonstiff order conditions

Use rooted trees and B-series.

Represent the elementary differentials using trees:

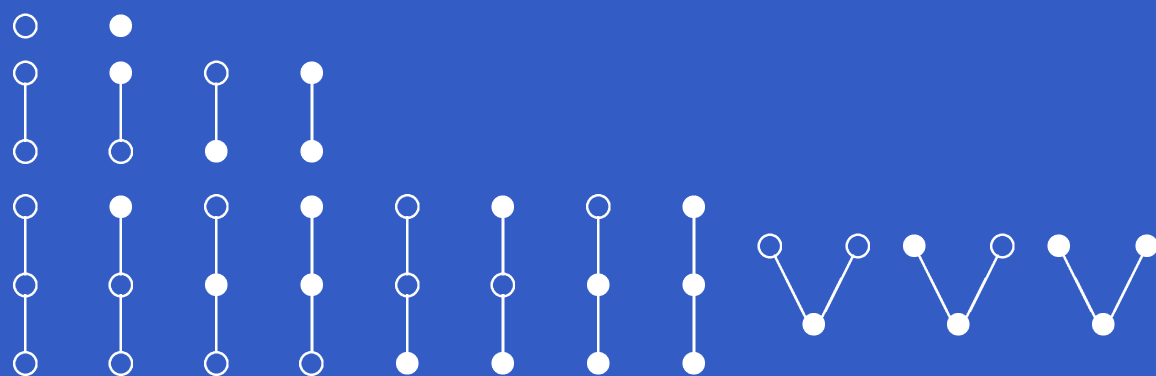
- Associate a closed node with $L$ and an open node with $N$
- 2T$^*$ - Bi-coloured rooted trees with one child closed nodes

# Nonstiff order conditions

Use rooted trees and B-series.
Represent the elementary differentials using trees:

- Associate a closed node with $L$ and an open node with $N$
- $2T^*$ - Bi-coloured rooted trees with one child closed nodes

Order 1:
Order 2:
Order 3:



# Nonstiff order conditions

Use rooted trees and B-series.
Represent the elementary differentials using trees:

- Associate a closed node with $L$ and an open node with $N$
- 2T$^*$ - Bi-coloured rooted trees with one child closed nodes

| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\theta_n$ | 2 | 4 | 11 | 34 | 117 | 421 | 1589 | 6162 | 24507 | 99268 |
| $\Theta = \sum_i^n \theta_n$ | 2 | 6 | 17 | 51 | 168 | 589 | 2178 | 8340 | 32847 | 132115 |

The number of rooted trees in $2$T$^*$ for all orders up to ten.

# Elementary differentials and B-series

The elementary differentials are recursively generated as

$$F(\tau)(u) = \begin{cases} LF(\tau_1)(u) & \text{if } \tau = [\circ; \tau_1] \\ N^{(\ell)}(u)(F(\tau_1)(u), \dots, F(\tau_\ell)(u)) & \text{if } \tau = [\bullet; \tau_1, \dots, \tau_\ell] \end{cases}$$

# Elementary differentials and B-series

The elementary differentials are recursively generated as

$$F(\tau)(u) = \begin{cases} LF(\tau_1)(u) & \text{if } \tau = [\circ; \tau_1] \\ N^{(\ell)}(u)(F(\tau_1)(u), \ldots, F(\tau_\ell)(u)) & \text{if } \tau = [\bullet; \tau_1, \ldots, \tau_\ell] \end{cases}$$

For an elementary weight function $a : 2\mathrm{T}^* \to \mathbb{R}$ the B-series is

$$B(a,u) = a(\emptyset)u + \sum_{\tau \in 2\mathrm{T}^*} h^{|\tau|} \frac{a(\tau)}{\sigma(\tau)} F(\tau)(u)$$

# Elementary differentials and B-series

The elementary differentials are recursively generated as

$$F(\tau)(u) = \begin{cases} LF(\tau_1)(u) & \text{if } \tau = [\circ; \tau_1] \\ N^{(\ell)}(u)(F(\tau_1)(u), \ldots, F(\tau_\ell)(u)) & \text{if } \tau = [\bullet; \tau_1, \ldots, \tau_\ell] \end{cases}$$

For an elementary weight function $a : 2\mathrm{T}^* \to \mathbb{R}$ the B-series is

$$B(a, u) = a(\emptyset)u + \sum_{\tau \in 2\mathrm{T}^*} h^{|\tau|} \frac{a(\tau)}{\sigma(\tau)} F(\tau)(u)$$

The elementary weight function for the exact solution is

$$a(\tau) = \frac{1}{\gamma(\tau)},$$

where $\gamma$ is the density of single coloured tree

# The lemmas

To obtain B-series expansions of the numerical solution we need three Lemmas.

# The lemmas

To obtain B-series expansions of the numerical solution we need three Lemmas.

**Lemma 2.** *Let* $a : 2\mathrm{T}^* \to \mathbb{R}$, *with* $a(\emptyset) = 1$, *then*

$$hN(B(a,u)) = B(a',u),$$

*where*

$$a'(\tau) = \begin{cases} 0 & \text{if } \tau = [\circ; \tau_1] \\ a(\tau_1)\dots a(\tau_\ell) & \text{if } \tau = [\bullet; \tau_1, \dots, \tau_\ell] \end{cases}$$

# The lemmas

To obtain B-series expansions of the numerical solution we need three Lemmas.

**Lemma 3.** *Let* $a : 2\mathrm{T}^* \to \mathbb{R}$*, with* $a(\emptyset) = 1$*, then*

$$hN(B(a,u)) = B(a',u),$$

*where*

$$a'(\tau) = \begin{cases} 0 & \textit{if } \tau = [\circ; \tau_1] \\ a(\tau_1)\dots a(\tau_\ell) & \textit{if } \tau = [\bullet; \tau_1, \dots, \tau_\ell] \end{cases}$$

**Lemma 4.** *Let* $a : 2\mathrm{T}^* \to \mathbb{R}$*, then*

$$(hL)^l B(a,u) = B(\mathcal{L}^l a, u),$$

*where*

$$(\mathcal{L}^l a)(\tau) = \begin{cases} (\mathcal{L}^{l-1}a)(\tau_1) & \textit{if } \tau = [\circ; \tau_1] \\ 0 & \textit{if } \tau = [\bullet; \tau_1, \dots, \tau_\ell] \end{cases}$$

# The lemmas

**Lemma 5.** *Let $\psi_x(z)$ be a power series*

$$\psi_x(z) = \sum_{l \geq 0} x^{[l]} z^l$$

*and let $a : 2\mathrm{T}^* \to \mathbb{R}$, then*

$$\psi_x(hL)B(a,u) = B(\psi_x(\mathcal{L})a,u),$$

*where the elementary weight function satisfies, $(\psi_x(\mathcal{L})a)(\emptyset) = x^{[0]}a(\emptyset)$, and*

$$(\psi_x(\mathcal{L})a)(\tau) = \sum_{l \geq 0} x^{[l]}(\mathcal{L}^l a)(\tau)$$

# General rule

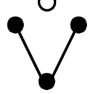
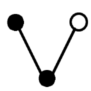
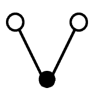
The rule for computing the elementary weight is:

- Attach ${b^{[i]}}^T$ to the root open (black) node
- Attach $A^{[i]}$ to all remaining nonterminal open (black) nodes
- Attach $A^{[i]}e$ to all terminal open (black) nodes
- Attach $C^{[i]}e$ to all terminal closed (white) nodes
- Attach $I$ to all remaining closed (white) nodes

where $i$ is the number of closed (white) nodes below the corresponding node and

$$A^{[i]} = \sum_{k=1}^{m} \phi_l^{[k]}(c)\alpha^{[k]}, \qquad {b^{[i]}}^T = \sum_{k=1}^{m} \phi_l^{[k]}(1){\beta^{[k]}}^T,$$

$$C^{[i]} = \frac{1}{(i+1)!}C^{i+1}.$$

| $\tau$ | $\gamma(\tau)$ | $F(\tau)(u)$ | $\alpha(\tau)$ |
|---|---|---|---|
|  | 1 | $N$ | ${b^{[0]}}^T e$ |
|  | 2 | $N'N$ | ${b^{[0]}}^T A^{[0]} e$ |
|  | 2 | $N'Lu$ | ${b^{[0]}}^T C^{[0]} e$ |
|  | 2 | $LN$ | ${b^{[1]}}^T e$ |
|  | 6 | $N'N'N$ | ${b^{[0]}}^T A^{[0]} A^{[0]} e$ |
|  | 6 | $N'N'Lu$ | ${b^{[0]}}^T A^{[0]} C^{[0]} e$ |
|  | 6 | $N'LN$ | ${b^{[0]}}^T A^{[1]} e$ |
|  | 6 | $LN'N$ | ${b^{[1]}}^T A^{[0]} e$ |
|  | 6 | $N'LLu$ | ${b^{[0]}}^T C_1 e$ |
|  | 6 | $LN'Lu$ | ${b^{[1]}}^T C^{[0]} e$ |
|  | 6 | $LLN$ | ${b^{[2]}}^T e$ |
|  | 3 | $N''(N, N)$ | ${b^{[0]}}^T (A^{[0]} e)(A^{[0]} e)$ |
|  | 3 | $N''(N, Lu)$ | ${b^{[0]}}^T (A^{[0]} e)(C^{[0]} e)$ |
|  | 3 | $N''(Lu, Lu)$ | ${b^{[0]}}^T (C^{[0]} e)(C^{[0]} e)$ |

Relations between elementary differentials and elementary weights

# Exp. RK methods for parabolic PDEs

- Stiff order theory for ETD RK methods for parabolic PDEs (Hochbruck–Ostermann'04)
  - Abstract ODEs on a Banach spaces
  - Sectorial operators
  - Locally Lipschitz continuous functions

# Exp. RK methods for parabolic PDEs

- Stiff order theory for ETD RK methods for parabolic PDEs (Hochbruck–Ostermann'04)
  - Abstract ODEs on a Banach spaces
  - Sectorial operators
  - Locally Lipschitz continuous functions

The error bounds depend form the space where the solution evolves!

It is not possible to construct stiff fourth order
explicit exponential Runge–Kutta method with only four stages

# Exp. RK methods for parabolic PDEs

- Stiff order theory for ETD RK methods for parabolic PDEs (Hochbruck–Ostermann'04)
  - Abstract ODEs on a Banach spaces
  - Sectorial operators
  - Locally Lipschitz continuous functions

- Implicit Exp RK methods of *collocation type*

The methods converge at least with their stage order. Higher and even fractional order of convergence is possible if additional temporal and spatial regularity are required Hochbruck–Ostermann'04.

# Exp. RK methods for parabolic PDEs

- Stiff order theory for ETD RK methods for parabolic PDEs (Hochbruck–Ostermann'04)
    - Abstract ODEs on a Banach spaces
    - Sectorial operators
    - Locally Lipschitz continuous functions

- Implicit Exp RK methods of *collocation type*

The methods converge at least with their stage order. Higher and even fractional order of convergence is possible if additional temporal and spatial regularity are required Hochbruck–Ostermann'04.

- No need of new order theory
- Higher stage order
- Solve the nonlinear equations by fixed-point iterations

# General linear methods

Consider $u' = f(u(t)), \quad u(t_0) = u_0, \quad f(u(t)) : \mathbb{R}^d \to \mathbb{R}^d$.
Assume that at the beginning of step number $n$, $r$ quantities

$$u_1^{[n-1]}, u_2^{[n-1]}, \ldots, u_r^{[n-1]},$$

are available from approximations computed in the previous steps. If

$$U_1, U_2, \ldots, U_s$$

are the internal stage approximations to the solution at points near the current time step, then then the quantities imported into and evaluated in step number $n$ are related by the equations

$$U_i = \sum_{j=1}^{s} a_{ij} h f(Uj) + \sum_{j=1}^{r} d_{ij} u_j^{[n-1]}, \quad i = 1, 2, \ldots s,$$

$$u_i^{[n]} = \sum_{j=1}^{s} b_{ij} h f(Uj) + \sum_{j=1}^{r} v_{ij} u_j^{[n-1]}, \quad i = 1, 2, \ldots r,$$

# Vector notations

$$U_i = \sum_{j=1}^{s} a_{ij} h f(Uj) + \sum_{j=1}^{r} d_{ij} u_j^{[n-1]}, \quad i = 1, 2, \ldots s,$$

$$u_i^{[n]} = \sum_{j=1}^{s} b_{ij} h f(Uj) + \sum_{j=1}^{r} v_{ij} u_j^{[n-1]}, \quad i = 1, 2, \ldots r,$$

# Vector notations

$$U_i = \sum_{j=1}^{s} a_{ij} h f(Uj) + \sum_{j=1}^{r} d_{ij} u_j^{[n-1]}, \quad i = 1, 2, \ldots s,$$

$$u_i^{[n]} = \sum_{j=1}^{s} b_{ij} h f(Uj) + \sum_{j=1}^{r} v_{ij} u_j^{[n-1]}, \quad i = 1, 2, \ldots r,$$

Introducing the vector notations

$$U = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_s \end{bmatrix}, \quad f(U) = \begin{bmatrix} f(U_1) \\ f(U_2) \\ \vdots \\ f(U_s) \end{bmatrix}, \quad u^{[n-1]} = \begin{bmatrix} u_1^{[n-1]} \\ u_2^{[n-1]} \\ \vdots \\ u_r^{[n-1]} \end{bmatrix}, \quad u^{[n]} = \begin{bmatrix} u_1^{[n]} \\ u_2^{[n]} \\ \vdots \\ u_r^{[n]} \end{bmatrix},$$

# Vector notations

$$U_i = \sum_{j=1}^{s} a_{ij} h f(Uj) + \sum_{j=1}^{r} d_{ij} u_j^{[n-1]}, \quad i = 1, 2, \ldots s,$$

$$u_i^{[n]} = \sum_{j=1}^{s} b_{ij} h f(Uj) + \sum_{j=1}^{r} v_{ij} u_j^{[n-1]}, \quad i = 1, 2, \ldots r,$$

allows us to rewrite the above method in the following more compact form

$$\left[\begin{array}{c} U \\ \hline u^{[n]} \end{array}\right] = \left[\begin{array}{c|c} A \otimes I_d & D \otimes I_d \\ \hline B \otimes I_d & V \otimes I_d \end{array}\right] \left[\begin{array}{c} hf(U) \\ \hline u^{[n-1]} \end{array}\right],$$

where $\otimes$ is the Kronecker product and $I_d$ is the $d \times d$ identity matrix.

# Examples of GLMs

Consider $k$-step linear multistep methods of Adams type

$$u_n = u_{n-1} + h \sum_{i=0}^{k} \beta_i f(u_{n-i}).$$

# Examples of GLMs

Consider $k$-step linear multistep methods of Adams type

$$u_n = u_{n-1} + h \sum_{i=0}^{k} \beta_i f(u_{n-i}).$$

In general linear form

$$\left[\begin{array}{c} U_1 \\ \hline u_n \\ hf(U_1) \\ hf(u_{n-1}) \\ \vdots \\ hf(u_{n-k-1}) \end{array}\right] = \left[\begin{array}{c|ccccc} \beta_0 & 1 & \beta_1 & \cdots & \beta_{k-1} & \beta_k \\ \hline \beta_0 & 1 & \beta_1 & \cdots & \beta_{k-1} & \beta_k \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{array}\right] \left[\begin{array}{c} hf(U_1) \\ \hline u_{n-1} \\ hf(u_{n-1}) \\ hf(u_{n-2}) \\ \vdots \\ hf(u_{n-k}) \end{array}\right].$$

# Examples of GLMs

The classical fourth order Runge–Kutta method

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array},$$

# Examples of GLMs

The classical fourth order Runge–Kutta method

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{array},
$$

can be written as

$$
\left[\begin{array}{c}
U_1 \\ U_2 \\ U_3 \\ U_4 \\ \hline u_n
\end{array}\right]
=
\left[\begin{array}{cccc|c}
0 & 0 & 0 & 0 & 1 \\
\frac{1}{2} & 0 & 0 & 0 & 1 \\
0 & \frac{1}{2} & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 \\
\hline
\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & 1
\end{array}\right]
\left[\begin{array}{c}
hf(U_1) \\ hf(U_2) \\ hf(U_3) \\ hf(U_4) \\ \hline u_{n-1}
\end{array}\right].
$$

# Examples of GLMs

It is not always appropriate to represent a Runge–Kutta method like a general liner method with $r = 1$. Example is the Lobatto IIIA method

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ \frac{1}{2} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}.$$

# Examples of GLMs

It is not always appropriate to represent a Runge–Kutta method like a general liner method with $r = 1$. Example is the Lobatto IIIA method

$$\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\
\frac{1}{2} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array}.$$

It has the following general linear form

$$\left[\begin{array}{c} U_1 \\ U_2 \\ \hline u_n \\ hf(U_2) \end{array}\right] = \left[\begin{array}{cc|cc} \frac{1}{3} & -\frac{1}{24} & 1 & \frac{5}{24} \\ \frac{2}{3} & \frac{1}{6} & 1 & \frac{1}{6} \\ \hline \frac{2}{3} & \frac{1}{6} & 1 & \frac{1}{6} \\ 0 & 1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} hf(U_1) \\ hf(U_2) \\ \hline u_{n-1} \\ hf(u_{n-1}) \end{array}\right].$$

# Practical GLMs

Introduce initial assumptions

- Stage order equal to the overall order of the method
- Quantities passed from step to step to be approximations of the Nordsieck vector
- Stability regions identical to those corresponding to RK methods (IRKS)

Wright'02
Butcher, Wright'03
Butcher, Zackiewicz'04

# Exp. general linear methods

Consider the following unified format of Exp. GLMs

$$U_i = \sum_{j=1}^{s}\sum_{l=1}^{m}\alpha_{ij}^{[l]}\,\phi^{[l]}(c_i)(hL)\,hN(U_j) + \sum_{j=1}^{r}\sum_{l=1}^{m}\delta_{ij}^{[l]}\,\phi^{[l]}(c_i)(hL)\,u_j^{[n-1]},$$

$$u_i^{[n]} = \sum_{j=1}^{s}\sum_{l=1}^{m}\beta_{ij}^{[l]}\,\phi^{[l]}(1)(hL)\,hN(U_j) + \sum_{j=1}^{r}\sum_{l=1}^{m}\nu_{ij}^{[l]}\,\phi^{[l]}(1)(hL)\,u_j^{[n-1]}.$$

# Exp. general linear methods

Consider the following unified format of Exp. GLMs

$$U_i = \sum_{j=1}^{s}\sum_{l=1}^{m} \alpha_{ij}^{[l]}\, \phi^{[l]}(c_i)(hL)\, hN(U_j) + \sum_{j=1}^{r}\sum_{l=1}^{m} \delta_{ij}^{[l]}\, \phi^{[l]}(c_i)(hL)\, u_j^{[n-1]},$$

$$u_i^{[n]} = \sum_{j=1}^{s}\sum_{l=1}^{m} \beta_{ij}^{[l]}\, \phi^{[l]}(1)(hL)\, hN(U_j) + \sum_{j=1}^{r}\sum_{l=1}^{m} \nu_{ij}^{[l]}\, \phi^{[l]}(1)(hL)\, u_j^{[n-1]}.$$

# Exp. general linear methods

Consider the following unified format of Exp. GLMs

$$U_i = \sum_{j=1}^{s} \sum_{l=1}^{m} \alpha_{ij}^{[l]} \, \phi^{[l]}(c_i)(hL) \, hN(U_j) + \sum_{j=1}^{r} \sum_{l=1}^{m} \delta_{ij}^{[l]} \, \phi^{[l]}(c_i)(hL) \, u_j^{[n-1]},$$

$$u_i^{[n]} = \sum_{j=1}^{s} \sum_{l=1}^{m} \beta_{ij}^{[l]} \, \phi^{[l]}(1)(hL) \, hN(U_j) + \sum_{j=1}^{r} \sum_{l=1}^{m} \nu_{ij}^{[l]} \, \phi^{[l]}(1)(hL) \, u_j^{[n-1]}.$$

Similarly, to the traditional GLMs, we can represent the method in the following matrix form

$$\left[\begin{array}{c} U \\ \hline u^{[n]} \end{array}\right] = \left[\begin{array}{c|c} A(\phi) & D(\phi) \\ \hline B(\phi) & V(\phi) \end{array}\right] \left[\begin{array}{c} hN(U) \\ \hline u^{[n-1]} \end{array}\right],$$

where each of the coefficient matrices $A(\phi), B(\phi), D(\phi), V(\phi)$ has entries which are linear combinations of the $\phi^{[l]}$ functions.

# Generalized IF methods

Seek for a solution of the form

$$u(t_n + t) = \phi_{t,\widehat{F}}(v(t)),$$

where $\phi_{t,\widehat{F}}$ is the flow of the differential equation

$$u' = \widehat{F}(u,t), \qquad u(0) = u_n.$$

# Generalized IF methods

Seek for a solution of the form

$$u(t_n + t) = \phi_{t,\widehat{F}}(v(t)),$$

where $\phi_{t,\widehat{F}}$ is the flow of the differential equation

$$u' = \widehat{F}(u,t), \qquad u(0) = u_n.$$

The vector field $\widehat{F}(u,t)$ must:

- Approximates the original vector field $f(u, t_n + t)$ around the point $u_n$.
- Have a flow $\phi_{t,\widehat{F}}$ which is easy to compute exactly.

# Generalized IF methods

Seek for a solution of the form

$$u(t_n + t) = \phi_{t,\widehat{F}}(v(t)),$$

where $\phi_{t,\widehat{F}}$ is the flow of the differential equation

$$u' = \widehat{F}(u,t), \qquad u(0) = u_n.$$

The vector field $\widehat{F}(u,t)$ must:

- Approximates the original vector field $f(u, t_n + t)$ around the point $u_n$.
- Have a flow $\phi_{t,\widehat{F}}$ which is easy to compute exactly.

The corresponding differential equation for the $v$ variable is

$$v'(t) = \left(\frac{\partial u}{\partial v}\right)^{-1} \left(f(u, t_n + t) - \widehat{F}(u,t), \right), \qquad v(0) = u_n.$$

Use numerical method on the transformed equation and then transform back.

# GIF for semilinear problems

For the semilinear problem (1), we can choice $\widehat{F}(u,t) = Lu + \mathsf{L}_k(N,t)$, where $\mathsf{L}_k(N,t)$ is the Lagrange interpolating polynomial of degree $k-1$ for the function $N(u(t_n+t),\, t_n+t)$, which passes through the $k$ points $N_n, N_{n-1}, \ldots, N_{n-k+1}$.

The transformed equation is

$$v'(t) = e^{-tL}\big(N(u(t_n+t),\, t_n+t) - \mathsf{L}_k(N,t)\big) \qquad v(0) = u_n.$$

# GIF for semilinear problems

For the semilinear problem (1), we can choice $\widehat{F}(u,t) = Lu + \mathsf{L}_k(N,t)$, where $\mathsf{L}_k(N,t)$ is the Lagrange interpolating polynomial of degree $k-1$ for the function $N(u(t_n+t),\, t_n+t)$, which passes through the $k$ points $N_n, N_{n-1}, \ldots, N_{n-k+1}$.

The transformed equation is

$$v'(t) = e^{-tL}\big(N(u(t_n+t),\, t_n+t) - \mathsf{L}_k(N,t)\big) \qquad v(0) = u_n.$$

- Applying a multistep method to the transformed equation leads to a class of methods which includes as a special cases all exponential multistep methods considered so far.
- Applying a multistage method to the transformed equation leads to a new class of methods known as GIF/RK methods (Krogstad'03).

# GIF/RK methods

$$\begin{aligned}
\mathsf{L}_0(N,t) &= 0 && - \quad \text{IF RK (Lawson)},\\
\mathsf{L}_1(N,t) &= N_n && - \quad \text{GIF1/RK (Krogstad)},\\
\mathsf{L}_2(N,t) &= N_n + t\left(\frac{N_n - N_{n-1}}{h}\right) && - \quad \text{GIF2/RK (Krogstad)},
\end{aligned}$$

# GIF/RK methods

$$
\begin{aligned}
\mathsf{L}_0(N,t) &= 0 && - && \text{IF RK (Lawson)},\\
\mathsf{L}_1(N,t) &= N_n && - && \text{GIF1/RK (Krogstad)},\\
\mathsf{L}_2(N,t) &= N_n + t\left(\frac{N_n - N_{n-1}}{h}\right) && - && \text{GIF2/RK (Krogstad)},
\end{aligned}
$$

$$
\left[\begin{array}{cccc|cc}
0 & 0 & 0 & 0 & I & 0\\
a_{21}(hL) & 0 & 0 & 0 & e^{c_2 hL} & d_{22}(hL)\\
a_{31}(hL) & \alpha_{32}e^{(c_3-c_2)hL} & 0 & 0 & e^{c_3 hL} & d_{32}(hL)\\
a_{41}(hL) & \alpha_{42}e^{(c_4-c_2)hL} & \alpha_{43}e^{(c_4-c_3)hL} & 0 & e^{c_4 hL} & d_{42}(hL)\\
\hline
b_1(hL) & \beta_2 e^{(1-c_2)hL} & \beta_3 e^{(1-c_3)hL} & \beta_4 e^{(1-c_4)hL} & e^{hL} & v_{12}(hL)\\
I & 0 & 0 & 0 & 0 & 0
\end{array}\right],
$$

Fourth order GIF2/RK method

# GIF/RK methods

$$\begin{aligned} \mathsf{L}_0(N,t) &= 0 &&- \quad \text{IF RK (Lawson)},\\ \mathsf{L}_1(N,t) &= N_n &&- \quad \text{GIF1/RK (Krogstad)},\\ \mathsf{L}_2(N,t) &= N_n + t\left(\frac{N_n - N_{n-1}}{h}\right) &&- \quad \text{GIF2/RK (Krogstad)}, \end{aligned}$$

where

$$a_{21}(hL) = c_2\phi^{[1]} + c_2^2\phi^{[2]},$$

$$a_{31}(hL) = c_3\phi^{[1]} + c_3^2\phi^{[2]} - \alpha_{32}\,(1+c_2)\,e^{(c_3-c_2)hL},$$

$$a_{41}(hL) = c_4\phi^{[1]} + c_4^2\phi^{[2]} - \alpha_{42}\,(1+c_2)\,e^{(c_4-c_2)hL} - \alpha_{43}\,(1+c_3)\,e^{(c_4-c_3)hL},$$

$$b_1(hL) = \phi^{[1]}+\phi^{[2]}-\beta_2(1+c_2)e^{(1-c_2)hL}-\beta_3(1+c_3)e^{(1-c_3)hL}-\beta_4(1+c_4)e^{(1-c_4)hL},$$

$$d_{22}(hL) = -c_2^2\phi^{[2]},$$

$$d_{32}(hL) = -c_3^2\phi^{[2]} + c_2\alpha_{32}e^{(c_3-c_2)hL},$$

$$d_{42}(hL) = -c_4^2\phi^{[2]} + c_2\alpha_{42}e^{(c_4-c_2)hL} + c_3\alpha_{43}e^{(c_4-c_3)hL},$$

$$v_{12}(hL) = -\phi^{[2]} + c_2\beta_2e^{(1-c_2)hL} + c_3\beta_3e^{(1-c_3)hL} + c_4\beta_4e^{(1-c_4)hL}.$$

# GIF/RK methods

Similarly, if $\widehat{F}(u,t) = Lu + \mathsf{T}_k(N,t)$, where

$$\mathsf{T}_k(N,t) = b + \sum_{\alpha=1}^{k} \left(c_\alpha \sin(\alpha t) + d_\alpha \cos(\alpha t)\right); \qquad k \in \mathbb{N},\ c_\alpha, d_\alpha \in,$$

it is possible to construct new GIF methods.

This approach leads to the followig $\phi^{[l]}$ funactions

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

$$\phi^{[\alpha]}_{\cos}(hL) = \frac{e^{hL}L - L\cos(\alpha h) + \alpha\sin(\alpha h)I}{\alpha^2 I + L^2}.$$

# Exp. Int. and Schrödinger equation

- Symmetric exponential integrators of firs, second and third order - convergence analysis (Hochbruck, Lubich'99).

- Higher order Exp. RK methods for the nonlinear Schrödinger equation are studied by Berland, Skaflestad, Owren'04.

IF RK methods seems to be the best for this problem.

# Exp. Int. for oscillatory problems

Consider the following oscillatory second-order differential equation

$$u'' = Lu + g(u), \qquad u(0) = u_0,\ y'(0) = y_0',$$

where $-L$ is positive semi-definite and has arbitrarily large norm.

# Exp. Int. for oscillatory problems

Consider the following oscillatory second-order differential equation

$$u'' = Lu + g(u), \qquad u(0) = u_0,\ y'(0) = y_0',$$

where $-L$ is positive semi-definite and has arbitrarily large norm.

Again base the methods on (vcf) (Hochbruck, Lubich'99)

$$u(t+\tau) = \cos(\tau\Omega)u(t) + \Omega^{-1}\sin(\tau\Omega)u'(t) + \int_0^\tau \Omega^{-1}\sin\left((\tau - s)\Omega\right) g(u(t+s))ds,$$

where $\Omega = \sqrt{-L}$.

# Exp. Int. for oscillatory problems

Consider the following oscillatory second-order differential equation

$$u'' = Lu + g(u), \qquad u(0) = u_0,\ y'(0) = y_0',$$

where $-L$ is positive semi-definite and has arbitrarily large norm.

Again base the methods on (vcf) (Hochbruck, Lubich'99)

$$u(t+\tau) = \cos(\tau\Omega)u(t) + \Omega^{-1}\sin(\tau\Omega)u'(t) + \int_0^\tau \Omega^{-1}\sin\left((\tau - s)\Omega\right) g(u(t+s))ds,$$

where $\Omega = \sqrt{-L}$.

Approximating $g(u(t))$ by a suitable constant $g_n$ we obtain the following numerical scheme

$$u_{n+1} - 2u_n + u_{n-1} = h^2\sigma(-h^2L)(Lu + g_n)$$
$$u_1 = \cos(h\Omega)u_0 + \Omega^{-1}\sin(h\Omega)u_0' + \tfrac{1}{2}h^2\sigma(-h^2L)g_0,$$

where

$$\sigma(z^2) = 2\frac{1-\cos z}{z^2}.$$

# Implementation issues

Consider the ETD $\phi^{[i]}$ functions

$$\phi^{[1]}(z) = \frac{e^z - 1}{z}, \qquad \phi^{[i+1]}(z) = \frac{\phi^{[i]}(z) - \frac{1}{i!}}{z}, \quad \text{for} \quad i = 2, 3, \ldots .$$

A straightforward implementation suffers from cancellation errors (Kassam and Trefethen).

# Implementation issues

Consider the ETD $\phi^{[i]}$ functions

$$\phi^{[1]}(z) = \frac{e^z - 1}{z}, \qquad \phi^{[i+1]}(z) = \frac{\phi^{[i]}(z) - \frac{1}{i!}}{z}, \quad \text{for} \quad i = 2, 3, \ldots.$$

A straightforward implementation suffers from cancellation errors (Kassam and Trefethen).

Numerical techniques

- Decomposition methods
- Cauchy integral approach
- Krylov subspace approximations

# Implementation issues

- At the heart of all decomposition methods is the similarity transformation

$$A = SBS^{-1},$$

where $A = \gamma h L$. Therefore

$$\phi^{[i]}(A) = S\phi^{[i]}(B)S^{-1}.$$

# Implementation issues

- At the heart of all decomposition methods is the similarity transformation

$$A = SBS^{-1},$$

where $A = \gamma h L$. Therefore

$$\phi^{[i]}(A) = S\phi^{[i]}(B)S^{-1}.$$

Two conflicting tasks:

- Make $B$ close to diagonal so that $\phi^{[i]}(B)$ is easy to compute.
- Make $S$ well conditioned so that errors are not magnified.

# Implementation issues

- At the heart of all decomposition methods is the similarity transformation

$$A = SBS^{-1},$$

where $A = \gamma h L$. Therefore

$$\phi^{[i]}(A) = S\phi^{[i]}(B)S^{-1}.$$

  - Based on the Cauchy integral formula

$$\phi^{[i]}(A) = \frac{1}{2\pi i}\int_{\Gamma_A} \phi^{[i]}(\lambda)(\lambda I - A)^{-1} d\lambda,$$

where $\Gamma_A$ is a contour in the complex plane that encloses the eigenvalue of $A$, and it is also well separated from $0$.

# Implementation issues

- At the heart of all decomposition methods is the similarity transformation

$$A = SBS^{-1},$$

where $A = \gamma h L$. Therefore

$$\phi^{[i]}(A) = S\phi^{[i]}(B)S^{-1}.$$

- Using the trapezoid rule, we obtain the following approximation

$$\phi^{[i]}(A) \approx \frac{1}{k}\sum_{j=1}^{k} \lambda_j \phi^{[i]}(\lambda_j)(\lambda_j I - A)^{-1},$$

where $k$ is the number of the equally spaced points $\lambda_j$ along the contour $\Gamma_A$.

# Krylov subspace approximations

Approximately project the action of $\phi^{[i]}(A)$ on a state vector $v \in \mathbb{C}^d$, to a small Krylov subspace

$$K_{\mathsf{m}} \equiv \mathrm{span}\{v, Av, \ldots, A^{\mathsf{m}-1}v\}.$$

Construct a orthogonal basis $V_{\mathsf{m}} = [v_1, v_2, \ldots, v_{\mathsf{m}}]$ of $K_{\mathsf{m}}$ (Arnoldi, Lanczos)
If $H_{\mathsf{m}}$ is the $\mathsf{m} \times \mathsf{m}$ upper Hessenberg matrix generated by the process then

$$V_{\mathsf{m}}^T A V_{\mathsf{m}} = H_{\mathsf{m}}.$$

Therefore, $H_{\mathsf{m}}$ is the orthogonal projection of $A$ to the subspace $K_{\mathsf{m}}$ and

$$\begin{aligned} \phi^{[i]}(A)v = V_{\mathsf{m}} V_{\mathsf{m}}^T \phi^{[i]}(A)v &= \beta V_{\mathsf{m}} V_{\mathsf{m}}^T \phi^{[i]}(A) V_{\mathsf{m}} e_1 \\ &\approx \beta V_{\mathsf{m}} \phi^{[i]}(H_{\mathsf{m}}) e_1, \end{aligned}$$

where $e_1$ is the first unit vector in $\mathbb{R}^{\mathsf{m}}$ and $\beta \equiv ||v||_2$.

# Krylov subspace approximations

- Superlinear convergence (Hochbruck, Lubich'98)
- Preconditioning the Lanczos process (Hochbruck,Van der Eshof'04)
- At every step we need to construct several Krylov bases
- Multiple Arnldi methods (Schmitt, Weiner'95)

# Conclusions

- The exponential integrators have a long history
- A general framework for analyzing the non-stiff order based on GLMs can help
- IF, ETD, GIF, CF, EC are special cases
- Exp. integrators with Krylov approximation techniques are very promising

# Open problems

- Need to understand the role of the $\phi^{[i]}$ functions.
- Effective algorithms for their computation.
- Are these methods competitive with variable stepsize.
- Extensive numerical experiments.
- Stability analysis - generalize the concept of IRKS
- Exponential integrators for oscillatory problems.

# References


- J. Certaine, *The solution of ordinary differential equations with large time constants,* Math. Meth. Dig. Comp., 129-132 (1960).
- 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).
- R. Strehmel, R. Weiner, *B-convergence results for linearly implicit one step methods,* BIT, 27, 264-282 (1987).
- 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).
- 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).

# References


- M. Hochbruck, Ch. Lubich, *On Krylov Subspace Approximations to the matrix exponential operator,* SIAM J. Numer.Anal., 34, 1911–1925 (1997).
- M. Hochbruck, Ch. Lubich, H. Selhofer, *Exponential integrators for large systems of differential equations,* SIAM J. Sci. Comput., 19, 1552–1574 (1998).
- H. Munthe-Kaas, High order Runge–Kutta methods on manifolds, Appl. Num. Math., 29, 115-127 (1999)
- 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. Marthinsen, 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`

# References


- M. Hochbruck, A. Osterman, Exponential Runge-Kutta methods for parabolic problems, available at: `http://techmath.uibk.ac.at/numbau/alex/publications.html`
- M. Hochbruck, A. Osterman, Explicit Exponential Runge-Kutta methods for semilinear parabolic problems, available at: `http://techmath.uibk.ac.at/numbau/alex/publications.html`
- M. Hochbruck, Ch. Lubich, Exponential integrators for quantum-classical molecular dynamics, *BIT* 39, 620-645 (1999)
- M. Hochbruck, Ch. Lubich, A Gautschi-type method for oscillatory second-order differential equations, *Numer. Math.* 83, 403-426 (1999)
- J. Van der Eshof, M. Hochbruck, Preconditioning Lanczos approximations to the matrix exponential, available at: `http://www.am.uni-duesseldorf.de/~marlis`
- B. Schmitt, R. Weiner, Matrix-free W-methods using a multiple Arnoldi iteration, *Appl. Num. Math.* 18, 307-320 (1995)