NEW NUMERICAL INTEGRATORS BASED ON SOLVABILITY AND SPLITTING

Fernando Casas

Universitat Jaume I, Castellón, Spain

Fernando.Casas@uji.es

(on sabbatical leave at DAMTP, University of Cambridge)

Edinburgh, June 2004

...approach suggested by

Arieh Iserles

Outline of the talk

1. Some (well known) Lie group methods for linear problems (Fer and Magnus expansions).

2. Schemes based on triangular matrices (splitting + solvability).

3. Some methods and practical issues in their construction

1 Lie group methods (linear problems)

Let us consider a linear matrix ODE evolving in a Lie group $\mathcal{G}$

$$Y' = A(t)Y, \qquad Y(t_0) = Y_0 \in \mathcal{G} \tag{0}$$

with $A : [t_0, \infty[\times \mathcal{G} \longrightarrow \mathfrak{g}$ smooth enough.

$\mathfrak{g}$: Lie algebra associated with $\mathcal{G}$

Examples of $\mathcal{G}$: $\mathrm{SL}(n)$, $\mathrm{O}(n)$, $\mathrm{SU}(n)$, $\mathrm{Sp}(n)$, $\mathrm{SO}(n)$, ...

$Y(t) \in$ Lie group $\mathcal{G}$ if $A(t) \in$ Lie algebra $\mathfrak{g}$

$*$ There are several schemes preserving this feature (Magnus, Fer, Cayley,...)

1.1 Magnus expansion

For the equation

$$Y' = A(t)Y, \qquad Y(t_0) = I,$$

$*$ **Magnus** (1954) proposed

$$Y(t) = e^{\Omega(t)}, \qquad \Omega(t) = \sum_{k=1}^{\infty} \Omega_k(t) \tag{1}$$

1.1 Magnus expansion

For the equation

$$Y' = A(t)Y, \qquad Y(t_0) = I,$$

∗ **Magnus** (1954) proposed

$$Y(t) = e^{\Omega(t)}, \qquad \Omega(t) = \sum_{k=1}^{\infty} \Omega_k(t) \tag{1}$$

with $\log(Y(t))$ satisfying

$$\Omega' = d\exp_{\Omega}^{-1} A(t) = \sum_{k=0}^{\infty} \frac{B_k}{k!} \mathrm{ad}_{\Omega}^{k} A(t), \qquad \Omega(t_0) = 0, \tag{2}$$

1.1 Magnus expansion (II)

Here

$$\begin{aligned} \mathrm{ad}_{\Omega}^{0} A &= A \\ \mathrm{ad}_{\Omega}^{k} A &= [\Omega, \mathrm{ad}_{\Omega}^{k-1} A] \end{aligned}$$

$$[\Omega, A] \equiv \Omega A - A\Omega$$

and B_k are Bernoulli numbers.

1.1 Magnus expansion (III)

First terms in the expansion ($A_i \equiv A(t_i)$):

$$\begin{aligned}
\Omega_1(t) &= \int_{t_0}^{t} A(t_1)dt_1 \\
\Omega_2(t) &= \frac{1}{2}\int_{t_0}^{t} dt_1 \int_{t_0}^{t_1} dt_2 [A_1, A_2] \\
\Omega_3(t) &= \frac{1}{6}\int_{t_0}^{t} dt_1 \int_{t_0}^{t_1} dt_2 \int_{t_0}^{t_2} dt_3 ([A_1,[A_2,A_3]] + [A_3,[A_2,A_1]])
\end{aligned}$$

$\mathrm{e}^{\Omega(t)} \in \mathcal{G}$ even if the series Ω is truncated

1.1 Magnus expansion (III)

First terms in the expansion ($A_i \equiv A(t_i)$):

$$\Omega_1(t) = \int_{t_0}^{t} A(t_1) dt_1$$

$$\Omega_2(t) = \frac{1}{2} \int_{t_0}^{t} dt_1 \int_{t_0}^{t_1} dt_2 [A_1, A_2]$$

$$\Omega_3(t) = \frac{1}{6} \int_{t_0}^{t} dt_1 \int_{t_0}^{t_1} dt_2 \int_{t_0}^{t_2} dt_3 ([A_1, [A_2, A_3]] + [A_3, [A_2, A_1]])$$

$e^{\Omega(t)} \in \mathcal{G}$ even if the series Ω is truncated

* Expansion widely used in Quantum Mechanics, NMR spectroscopy, infrared divergences in QED, control theory,...

1.1 Magnus expansion (IV)

Magnus as a numerical integration method (Iserles & Nørsett, 1997)

1.1 Magnus expansion (IV)

Magnus as a numerical integration method (Iserles & Nørsett, 1997)

Two critical factors in the computational cost of the resulting algorithms:

1.1 Magnus expansion (IV)

Magnus as a numerical integration method (Iserles & Nørsett, 1997)

Two critical factors in the computational cost of the resulting algorithms:

(1) Evaluation of $\exp(\Omega)$

(Moler & Van Loan, Celledoni & Iserles,...)

1.1 Magnus expansion (IV)

Magnus as a numerical integration method (Iserles & Nørsett, 1997)

Two critical factors in the computational cost of the resulting algorithms:

(1) Evaluation of $\exp(\Omega)$

(Moler & Van Loan, Celledoni & Iserles,...)

(2) Number of commutators involved in the expansion

1.1 Magnus expansion (IV)

Magnus as a numerical integration method (Iserles & Nørsett, 1997)

Two critical factors in the computational cost of the resulting algorithms:

(1) Evaluation of $\exp(\Omega)$

(Moler & Van Loan, Celledoni & Iserles,...)

(2) Number of commutators involved in the expansion

To reduce this number is particularly useful the concept of **graded free Lie algebra** (Munthe-Kaas, Owren 1999)

1.1 Magnus expansion (V)

As a result,

1.1 Magnus expansion (V)

As a result,

* Numerical schemes based on Magnus up to order 8 have been constructed involving the minimum number of commutators in terms of quadratures and/or univariate integrals.

1.1 Magnus expansion (V)

As a result,

* Numerical schemes based on Magnus up to order 8 have been constructed involving the minimum number of commutators in terms of quadratures and/or univariate integrals.

* Efficient in applications

1.2 Other schemes

Fer expansion. Built by Francis Fer (1958).

1.2 Other schemes

Fer expansion. Built by Francis Fer (1958).

∗ Referred erroneously in the (mathematical physics) literature (e.g., Wilcox 1967), but ...

1.2 Other schemes

Fer expansion. Built by Francis Fer (1958).

$\ast$ Referred erroneously in the (mathematical physics) literature (e.g., Wilcox 1967), but ...

$\ast$ proposed (as an exercise!) by R. Bellman, ‘Introduction to Matrix Analysis’, 1960, page 204:

1.2 Other schemes

Fer expansion. Built by Francis Fer (1958).

$\ast$ Referred erroneously in the (mathematical physics) literature (e.g., Wilcox 1967), but ...

$\ast$ proposed (as an exercise!) by R. Bellman, ‘Introduction to Matrix Analysis’, 1960, page 204:

“The solution of $dX/dt = Q(t)X$, $X(0) = I$, can be put in the form $\mathrm{e}^{P}\mathrm{e}^{P_1}\cdots\mathrm{e}^{P_n}\cdots$, where $P = \int_0^t Q(s)ds$, and $P_n = \int_0^t Q_n ds$, with

$$Q_n = \mathrm{e}^{-P_{n-1}}Q_{n-1}\mathrm{e}^{P_{n-1}} + \int_0^{-1} \mathrm{e}^{sP_{n-1}}Q_{n-1}\mathrm{e}^{-sP_{n-1}}ds$$

The infinite product converges it t is sufficiently small.”

(See also Mathematical Reviews 21 2771, review done by R. Bellman)

1.2 Other schemes (II)

$*$ used as an (analytic) procedure in perturbation theory in Quantum Mechanics by Klarsfeld & Oteo (1989), but...

1.2 Other schemes (II)

$*$ used as an (analytic) procedure in perturbation theory in Quantum Mechanics by Klarsfeld & Oteo (1989), but...

$*$ numerical integration method built by Iserles (1984).

1.2 Other schemes (II)

∗ used as an (analytic) procedure in perturbation theory in Quantum Mechanics by Klarsfeld & Oteo (1989), but...

∗ numerical integration method built by Iserles (1984).

∗ This class of methods can actually be built from Magnus.

1.2 Other schemes (II)

$\ast$ used as an (analytic) procedure in perturbation theory in Quantum Mechanics by Klarsfeld & Oteo (1989), but...

$\ast$ numerical integration method built by Iserles (1984).

$\ast$ This class of methods can actually be built from Magnus.

$\ast$ They require the computation of several matrix exponentials.

1.3 Methods based on the Cayley transform

Let us suppose that $Y' = A(t)Y$ is defined in a J-orthogonal Lie group,

$$\mathrm{O}_J(n) = \{A \in \mathrm{GL}_n(\mathbb{R}) \, : \, A^T JA = J\},$$

J: constant matrix

1.3 Methods based on the Cayley transform

Let us suppose that $Y' = A(t)Y$ is defined in a J-orthogonal Lie group,

$$\mathrm{O}_J(n) = \{A \in \mathrm{GL}_n(\mathbb{R}) \, : \, A^T JA = J\},$$

J: constant matrix

Examples: orthogonal group ($J = I$), symplectic group, Lorentz group ($J = \mathrm{diag}(1,-1,-1,-1)$).

Solution:

$$Y(t) = \left(I - \frac{1}{2}C(t)\right)^{-1} \left(I + \frac{1}{2}C(t)\right)$$

1.3 Methods based on the Cayley transform (II)

with $C(t) \in \mathfrak{o}_J(n)$ satisfying (Iserles 2001)

$$\frac{dC}{dt} = A - \frac{1}{2}[C,A] - \frac{1}{4}CAC, \qquad t \geq t_0, \qquad C(t_0) = 0.$$

$\Rightarrow$ efficient methods without matrix exponentials!

1.3 Methods based on the Cayley transform (II)

with $C(t) \in \mathfrak{o}_J(n)$ satisfying (Iserles 2001)

$$\frac{dC}{dt} = A - \frac{1}{2}[C,A] - \frac{1}{4}CAC, \qquad t \geq t_0, \qquad C(t_0) = 0.$$

$\Rightarrow$ efficient methods without matrix exponentials!

In fact, one can also combine Magnus with Padé to avoid the use of matrix exponentials in J-orthogonal groups!

1.3 Methods based on the Cayley transform (II)

with $C(t) \in \mathfrak{o}_J(n)$ satisfying (Iserles 2001)

$$\frac{dC}{dt} = A - \frac{1}{2}[C,A] - \frac{1}{4}CAC, \qquad t \geq t_0, \qquad C(t_0) = 0.$$

$\Rightarrow$ efficient methods without matrix exponentials!

In fact, one can also combine Magnus with Padé to avoid the use of matrix exponentials in J-orthogonal groups!

* It is possible to construct methods which are more efficient than those based on the Cayley transform (Blanes, C., Ros 2002).

1.4 Summary

∗ These methods require the evaluation of one or several matrix exponentials

1.4 Summary

$\ast$ These methods require the evaluation of one or several matrix exponentials

$\Rightarrow$ They are expensive when n is very large

1.4 Summary

$*$ These methods require the evaluation of one or several matrix exponentials

$\Rightarrow$ They are expensive when n is very large

$*$ In some cases, if the exponential is approximated by rational functions the method does not preserve the Lie group structure,

in particular, when $\mathcal{G} = \mathrm{SL}(n)$

1.4 Summary

$*$ These methods require the evaluation of one or several matrix exponentials

$\Rightarrow$ They are expensive when n is very large

$*$ In some cases, if the exponential is approximated by rational functions the method does not preserve the Lie group structure,

in particular, when $\mathcal{G} = \mathrm{SL}(n)$

$\Longrightarrow$ Another class of methods is required.

2 Solvability + splitting

The procedure

For the linear system

$$Y' = A(t)Y, \qquad Y(0) = I,$$

we denote $Y_0 \equiv Y$, $A_0 \equiv A$ and suppose that

$$A_0(t) = A_{0_+}(t) + A_{0_-}(t),$$

where

$A_{0_+} \in \nabla_n$ is strictly upper-triangular

$A_{0_-} \in \tilde{\Delta}_n$ is weakly lower-triangular.

2 Solvability + splitting (II)

The idea is to write the solution as a product of upper and lower triangular matrices.

2 Solvability + splitting (II)

The idea is to write the solution as a product of upper and lower triangular matrices.

More specifically, we propose the following factorization:

$$Y_0(t) = L_0(t)U_0(t)Y_1(t)$$

such that

$$L_0' = A_{0\_}(t)\,L_0, \qquad L_0(0) = I$$

2 Solvability + splitting (II)

The idea is to write the solution as a product of upper and lower triangular matrices.

More specifically, we propose the following factorization:

$$Y_0(t) = L_0(t)U_0(t)Y_1(t)$$

such that

$$L_0' = A_{0_-}(t)\,L_0, \qquad L_0(0) = I$$

Observe then that $L_0(t)$ can be obtained by quadratures and $L_0(t) \in \tilde{\triangle}_n$.

2 Solvability + splitting (III)

Now we form the matrix

$$C_0 = L_0^{-1} A_{0_+} L_0$$

2 Solvability + splitting (III)

Now we form the matrix

$$C_0 = L_0^{-1} A_{0_+} L_0$$

which can also be split as

$$C_0(t) = C_{0_+}(t) + C_{0_-}(t),$$

where

$C_{0_+} \in \tilde{\bigtriangledown}_n$ is weakly upper-triangular

$C_{0_-} \in \triangle_n$ is strictly lower-triangular.

2 Solvability + splitting (IV)

Next we choose U_0 as the solution of

$$U_0' = C_{0_+}(t)U_0, \qquad U_0(0) = I$$

so that $U_0(t)$ can also be obtained by quadratures.

2 Solvability + splitting (IV)

Next we choose U_0 as the solution of

$$U_0' = C_{0_+}(t)U_0, \qquad U_0(0) = I$$

so that $U_0(t)$ can also be obtained by quadratures.

It is easy to show that Y_1 satisfies

$$Y_1' = A_1(t)Y_1, \qquad Y_1(0) = I,$$

with

$$A_1 = U_0^{-1}C_{0_-}U_0.$$

2 Solvability + splitting (V)

This gives a single step of the *solvable cycle*, which we repeat with A_1.

$$A_1 = A_{1_+} + A_{1_-}, \qquad A_{1_+} \in \nabla_n, \qquad A_{1_-} \in \tilde{\Delta}_n$$

$$Y_1 = L_1 U_1 Y_2$$

$$L_1' = A_{1_-} L_1, \qquad L_1(0) = I$$

etc.

2 Solvability + splitting (VI)

In this way one has the following algorithm:

$$Y \equiv Y_0 = L_0 U_0 L_1 U_1 \cdots L_k U_k Y_{k+1}$$

with $(k = 0, 1, 2, \ldots)$

$$A_k = A_{k_+} + A_{k_-}, \qquad A_{k_+} \in \nabla_n, \qquad A_{k_-} \in \tilde{\Delta}_n$$

$$L_k' = A_{k_-} L_k, \qquad L_k(0) = I$$

$$C_k \equiv L_k^{-1} A_{k_+} L_k = C_{k_+} + C_{k_-}$$

$$C_{k_+} \in \tilde{\nabla}_n, \qquad C_{k_-} \in \Delta_n$$

$$U_k' = C_{k_+} U_k, \qquad U_k(0) = I$$

2 Solvability + splitting (VII)

and finally

$$A_{k+1} \equiv U_k^{-1} C_{k\_} U_k, \qquad Y'_{k+1} = A_{k+1} Y_{k+1}$$

2 Solvability + splitting (VII)

and finally

$$A_{k+1} \equiv U_k^{-1} C_{k\_} U_k, \qquad Y'_{k+1} = A_{k+1} Y_{k+1}$$

Usually the factorization is truncated by taking $Y_{k+1} = I$.

2 Solvability + splitting (VII)

and finally

$$A_{k+1} \equiv U_k^{-1} C_{k\_} U_k, \qquad Y'_{k+1} = A_{k+1} Y_{k+1}$$

Usually the factorization is truncated by taking $Y_{k+1} = I$.

In what follows we will analyse the main features of this procedure as a *numerical integrator*.

2.1 Order of the method

Suppose that $A(t) = \varepsilon(a_0 + a_1 t + a_2 t^2 + \cdots)$ for some parameter $\varepsilon > 0$.

2.1 Order of the method

Suppose that $A(t) = \varepsilon(a_0 + a_1 t + a_2 t^2 + \cdots)$ for some parameter $\varepsilon > 0$.

Then

$$\begin{aligned} A_{j_-} &= t^{n_j}\varepsilon^{n_j}\left(\varepsilon\alpha_1 + t(\varepsilon\alpha_2 + \varepsilon^2\alpha_3) + O(t^2)\right) \\ A_{j_+} &= t^{m_j}\varepsilon^{m_j}\left(\varepsilon\beta_1 + t(\varepsilon\beta_2 + \varepsilon^2\beta_3) + O(t^2)\right) \end{aligned}$$

for $j = 1, 2, \ldots$, so that

$$\begin{aligned} L_j(t) &= I + \frac{1}{n_j+1}(t\varepsilon)^{n_j+1}\alpha_1 + \frac{1}{n_j+2}t^{n_j+2}\varepsilon^{n_j}(\varepsilon\alpha_2 + \varepsilon^2\alpha_3) + \cdots \\ U_j(t) &= I + \frac{1}{m_j+1}(t\varepsilon)^{m_j+1}\beta_1 + \frac{1}{m_j+2}t^{m_j+2}\varepsilon^{m_j}(\varepsilon\beta_2 + \varepsilon^2\beta_3) + \cdots \end{aligned}$$

2.1 Order of the method (II)

Furthermore,

$$
\begin{aligned}
n_{j+1} &= n_j + m_j + 1 \\
m_{j+1} &= n_j + 2m_j + 2 \qquad\qquad j = 1, 2, \ldots
\end{aligned}
$$

2.1 Order of the method (II)

Furthermore,

$$\begin{aligned} n_{j+1} &= n_j + m_j + 1 \\ m_{j+1} &= n_j + 2m_j + 2 \qquad\qquad j = 1, 2, \ldots \end{aligned}$$

j	n_j	m_j
1	1	2
2	4	7
3	12	20
4	33	54
5	88	143

2.1 Order of the method (III)

In consequence,

2.1 Order of the method (III)

In consequence,

(1) This algorithm could be useful for problems of the form

$$Y' = (B_0 + \varepsilon B_1)Y$$

if the system $Y' = B_0 Y$ can be solved exactly.

2.1 Order of the method (III)

In consequence,

(1) This algorithm could be useful for problems of the form

$$Y' = (B_0 + \varepsilon B_1)Y$$

if the system $Y' = B_0 Y$ can be solved exactly.

(2) The order of approximation is...

2.1 Order of the method (IV)

Y_0	$\approx$	L_0U_0	is order	1
Y_0	$\approx$	$L_0U_0L_1$		2
Y_0	$\approx$	$L_0U_0L_1U_1$		4
Y_0	$\approx$	$L_0U_0L_1U_1L_2$		7
Y_0	$\approx$	$L_0U_0L_1U_1L_2U_2$		12
Y_0	$\approx$	$L_0U_0L_1U_1L_2U_2L_3$		20
Y_0	$\approx$	$L_0U_0L_1U_1L_2U_2L_3U_3$		33

2.1 Order of the method (IV)

$Y_0 \approx L_0U_0$	is order	1
$Y_0 \approx L_0U_0L_1$		2
$Y_0 \approx L_0U_0L_1U_1$		4
$Y_0 \approx L_0U_0L_1U_1L_2$		7
$Y_0 \approx L_0U_0L_1U_1L_2U_2$		12
$Y_0 \approx L_0U_0L_1U_1L_2U_2L_3$		20
$Y_0 \approx L_0U_0L_1U_1L_2U_2L_3U_3$		33

With only 4 solvable cycles we get order 33!

2.1 Order of the method (IV)

$Y_0 \approx L_0U_0$	is order	1
$Y_0 \approx L_0U_0L_1$		2
$Y_0 \approx L_0U_0L_1U_1$		4
$Y_0 \approx L_0U_0L_1U_1L_2$		7
$Y_0 \approx L_0U_0L_1U_1L_2U_2$		12
$Y_0 \approx L_0U_0L_1U_1L_2U_2L_3$		20
$Y_0 \approx L_0U_0L_1U_1L_2U_2L_3U_3$		33

With only 4 solvable cycles we get order 33!

...if we can compute L_k and U_k up to this order...

2.2 Questions

Several problems involved

2.2 Questions

Several problems involved

(1) Does the approximate solution evolve in the Lie group if A is in the Lie algebra, i.e., is it a Lie group method?

2.2 Questions

Several problems involved

(1) Does the approximate solution evolve in the Lie group if A is in the Lie algebra, i.e., is it a Lie group method?

(2) Solve explicitly the systems $L_k' = A_{k_-} L_k$ and $U_k' = C_{k_+} U_k$

2.2 Questions

Several problems involved

(1) Does the approximate solution evolve in the Lie group if A is in the Lie algebra, i.e., is it a Lie group method?

(2) Solve explicitly the systems $L_k' = A_{k_-} L_k$ and $U_k' = C_{k_+} U_k$

(3) Approximate efficiently the (multiple) integrals involved.

3 Practical issues

(1) Preservation of the Lie-group structure

If $A(t) \in \mathfrak{sl}(n)$, the algorithm provides by construction approximations to $Y(t)$ in $\mathrm{SL}(n)$.

3 Practical issues

(1) Preservation of the Lie-group structure

If $A(t) \in \mathfrak{sl}(n)$, the algorithm provides by construction approximations to $Y(t)$ in $\mathrm{SL}(n)$.

Proof. $A_k = A_{k_+} + A_{k_-}$, with $A_{k_+} \in \nabla_n$, $A_{k_-} \in \tilde{\Delta}_n$. In fact A_{k_-} belongs to a solvable subalgebra of $\mathfrak{sl}(n)$. Therefore the solution of

$$L_k' = A_{k_-} L_k, \qquad L_k(0) = I$$

$L_k(t) \in \mathrm{SL}(n)$ (in fact, a solvable subgroup of).

3 Practical issues

(1) Preservation of the Lie-group structure

If $A(t) \in \mathfrak{sl}(n)$, the algorithm provides by construction approximations to $Y(t)$ in $\mathrm{SL}(n)$.

Proof. $A_k = A_{k_+} + A_{k_-}$, with $A_{k_+} \in \nabla_n$, $A_{k_-} \in \tilde{\Delta}_n$. In fact A_{k_-} belongs to a solvable subalgebra of $\mathfrak{sl}(n)$. Therefore the solution of

$$L_k' = A_{k_-} L_k, \qquad L_k(0) = I$$

$L_k(t) \in \mathrm{SL}(n)$ (in fact, a solvable subgroup of).

$\mathrm{Tr}(A_{k_+}) = 0$, and the trace is invariant under similarity, so that

$$\mathrm{Tr}(C_k) = \mathrm{Tr}(L_k^{-1} A_{k_+} L_k) = \mathrm{Tr}(A_{k_+}) = 0 \;\Rightarrow\; C_k \in \mathfrak{sl}(n)$$

3 Practical issues (II)

Next, $C_k = C_{k_+} + C_{k_-}$, with $C_{k_+} \in \tilde{\nabla}_n$, $C_{k_-} \in \triangle_n$ and U_k, solution of

$$U_k' = C_{k_+} U_k, \qquad U_k(0) = I$$

belongs to $\mathrm{SL}(n)$. Finally

$$A_{k+1} \equiv U_k^{-1} C_{k_-} U_k \ \in \mathfrak{sl}(n)$$

and the process is repeated. $\square$

3 Practical issues (II)

Next, $C_k = C_{k_+} + C_{k_-}$, with $C_{k_+} \in \tilde{\nabla}_n$, $C_{k_-} \in \triangle_n$ and U_k, solution of

$$U_k' = C_{k_+} U_k, \qquad U_k(0) = I$$

belongs to $\mathrm{SL}(n)$. Finally

$$A_{k+1} \equiv U_k^{-1} C_{k_-} U_k \;\; \in \mathfrak{sl}(n)$$

and the process is repeated. □

Other properties (i.e., orthogonality) are preserved only up to the order of the method.

3 Practical issues (III)

(2a) Explicit solution of $L_k' = A_{k\_} L_k$

3 Practical issues (III)

(2a) Explicit solution of $L_k' = A_{k\_} L_k$

Consider $k = 0$ and denote $A_0(t) = (a_{ij})$, $i, j = 1, \ldots, n$, $L_0(t) = (L_{ij})$, $j \le i$

$$A_{ii}(t) \equiv \int_0^t a_{ii}(t_1) dt_1.$$

Then the solution of $L_0' = A_{0\_}(t) L_0$, $L_0(0) = I$ is

$$\begin{aligned} L_{ii}(t) &= e^{A_{ii}(t)}, \qquad i = 1, \ldots, n \\ L_{ij}(t) &= e^{A_{ii}(t)} \int_0^t e^{-A_{ii}(t_1)} \left(\sum_{k=j}^{i-1} a_{ik}(t_1) L_{kj}(t_1) \right) dt_1 \end{aligned} \tag{3}$$

$i = 2, \ldots, n$, $j = 1, \ldots, i-1$.

3 Practical issues (IV)

(2b) Explicit solution of $U_k' = C_{k_+} U_k$

3 Practical issues (IV)

(2b) Explicit solution of $U_k' = C_{k_+} U_k$

Consider $k = 0$ and denote $C_0(t) = (c_{ij})$, $i, j = 1, \ldots, n$, $U_0(t) = (U_{ij})$, $j \geq i$

$$C_{ii}(t) \equiv \int_0^t c_{ii}(t_1) dt_1.$$

Then the solution of $U_0' = C_{0_+}(t) U_0$, $U_0(0) = I$ is

$$\begin{aligned} U_{ii}(t) &= \mathrm{e}^{C_{ii}(t)}, \qquad i = 1, \ldots, n \\ U_{ij}(t) &= \mathrm{e}^{C_{ii}(t)} \int_0^t \mathrm{e}^{-C_{ii}(t_1)} \left(\sum_{k=i+1}^{j} c_{ik}(t_1) U_{kj}(t_1) \right) dt_1 \end{aligned} \qquad (4)$$

$i = 1, \ldots, n-1$, $j = i+1, \ldots, n$.

3 Practical issues (V)

⇒ Explicit expressions for the elements of L_k and U_k in terms of multivariate integrals.

3 Practical issues (V)

$\Rightarrow$ Explicit expressions for the elements of L_k and U_k in terms of multivariate integrals.

They can be evaluated in sequence as follows:

$$
\begin{array}{ll|ll}
L_{ii} & i = 1, \ldots, n & U_{ii} & i = 1, \ldots, n \\
L_{i,i-1} & i = 2, \ldots, n & U_{i,i+1} & i = 1, \ldots, n-1 \\
L_{i,i-2} & i = 3, \ldots, n & U_{i,i+2} & i = 1, \ldots, n-2 \\
\vdots & & \vdots & \\
L_{n1} & & U_{1n} &
\end{array}
$$

3 Practical issues (VI)

In principle, the integrals appearing in L_k and U_k can be approximated by quadrature rules.

3 Practical issues (VI)

In principle, the integrals appearing in L_k and U_k can be approximated by quadrature rules.

To minimise the computational cost this has to be done by using the *minimum* number of A evaluations in each integration step.

3 Practical issues (VI)

In principle, the integrals appearing in L_k and U_k can be approximated by quadrature rules.

To minimise the computational cost this has to be done by using the *minimum* number of A evaluations in each integration step.

Question: Is it possible to approximate *all* the nested integrals with the evaluations required to compute

$$A_{ii} = \int_0^t a_{ii}(t_1)dt_1,$$

i.e., *à la* Magnus?

3 Practical issues (VI)

In principle, the integrals appearing in L_k and U_k can be approximated by quadrature rules.

To minimise the computational cost this has to be done by using the *minimum* number of A evaluations in each integration step.

Question: Is it possible to approximate *all* the nested integrals with the evaluations required to compute

$$A_{ii} = \int_0^t a_{ii}(t_1)dt_1,$$

i.e., *à la* Magnus?

YES!

3.1 Example

Illustration: method of order 4 with 2 A evaluations

Step $t = 0 \longmapsto t = h$.

3.1 Example

Illustration: method of order 4 with 2 A evaluations

Step $t=0 \longmapsto t=h$.

1- Approximate $A_{ii}(h)$, $i=1,\ldots,n$ up to order 4

$$\begin{aligned} A_{ii}(h)=\int_0^h a_{ii}(t)dt &= \frac{h}{3}\big(a_{ii}(0)+4a_{ii}(h/2)+a_{ii}(h)\big)+O(h^5) \\ &\equiv \tilde{A}_{ii}(h)+O(h^5) \end{aligned}$$

and $A_{ii}(h/2)$, $i=1,\ldots,n-1$, up to order 3 (necessary to approximate L_{ij}):

$$A_{ii}(h/2)=\frac{h}{24}(5a_{ii}(0)+8a_{ii}(h/2)-a_{ii}(h))+O(h^4)$$

3.1 Example (II)

2- $L_{ii}(h) = \exp(\tilde{A}_{ii}(h)) + O(h^5)$ ($i = 1, \ldots, n$) and $L_{ii}(h/2) = \exp(\tilde{A}_{ii}(h/2)) + O(h^4)$ ($i = 1, \ldots, n-1$).

3- Obtain an approximation to $L_{ij}(h)$, $j < i$, of order 4 and $L_{ij}(h/2)$ of order 3

$$L_{ij}(h) = \mathrm{e}^{A_{ii}(h)} \int_0^h F_{ij}(t) dt$$

with

$$F_{ij}(t) \equiv \mathrm{e}^{-A_{ii}(t)} \sum_{k=j}^{i-1} a_{ik}(t) L_{kj}(t)$$

3.1 Example (III)

Then

$$L_{ij}(h) = \mathrm{e}^{\tilde{A}_{ii}(h)} \frac{h}{3} \left(F_{ij}(0) + 4F_{ij}(h/2) + F_{ij}(h)\right) + O(h^5)$$

where $F_{ij}(0) = a_{ij}(0)$ and $F_{ij}(h/2)$ and $F_{ij}(h)$ have to be approximated up to order h^3.

The sequence of computation is ($i = 2, \ldots, n$):

(a) $F_{i,i-1}(h/2) = \mathrm{e}^{-\tilde{A}_{ii}(h/2)} a_{i,i-1}(h/2) L_{i-1,i-1}(h/2) + O(h^4)$

(b) $F_{i,i-1}(h) = \mathrm{e}^{-\tilde{A}_{ii}(h/2)} a_{i,i-1}(h) L_{i-1,i-1}(h) + O(h^5)$

(c) $L_{i,i-1}(h)$, $i = 2, \ldots, n$ up to order 4

3.1 Example (IV)

(d)

$$L_{i,i-1}(h/2) = e^{\tilde{A}_{ii}(h/2)} \frac{h}{24} (5a_{i,i-1}(0) + 8F_{i,i-1}(h/2) - F_{i,i-1}(h)) + O(h^4)$$

(e) $L_{i,i-2}(h)$, $i = 3, \ldots, n$, up to order 4 and $L_{i,i-2}(h/2)$ up to order 3

...and so on.

3.1 Example (IV)

(d)

$$L_{i,i-1}(h/2) = \mathrm{e}^{\tilde{A}_{ii}(h/2)}\frac{h}{24}\left(5a_{i,i-1}(0) + 8F_{i,i-1}(h/2) - F_{i,i-1}(h)\right) + O(h^4)$$

(e) $L_{i,i-2}(h)$, $i = 3, \ldots, n$, up to order 4 and $L_{i,i-2}(h/2)$ up to order 3

...and so on.

In this way we have $L_0(h)$ computed up to order $O(h^5)$ and also $L_0(h/2)$ up to order $O(h^4)$ with 2 evaluations of $A(t)$.

3.1 Example (V)

3- Next we compute C_0:

$$
\begin{aligned}
C_0(0) &= A_{0_+}(0) && \text{error } O(h^5) \\
C_0(h/2) &= L_0^{-1}(h/2)A_{0_+}(h/2)L_0(h/2) && \text{error } O(h^4) \\
C_0(h) &= L_0^{-1}(h)A_{0_+}(h)L_0(h) && \text{error } O(h^5)
\end{aligned}
$$

4- $C_{ii}(h) = \frac{h}{3}\big(c_{ii}(0)+4c_{ii}(h/2)+c_{ii}(h)\big)+O(h^5)$

$$C_{ii}(h/2) = \frac{h}{24}(5c_{ii}(0)+8c_{ii}(h/2)-c_{ii}(h))+O(h^4)$$

3.1 Example (VI)

5- $U_{i,i+1}(h)$, $i = 1, \ldots, n-1$, up to order $O(h^5)$;

$U_{i,i+1}(h/2)$, $i = 1, \ldots, n-1$, up to order $O(h^4)$;

$U_{i,i+2}(h)$, $i = 1, \ldots, n-2$, up to order $O(h^5)$;

$U_{i,i+2}(h)$, $i = 1, \ldots, n-2$, up to order $O(h^4)$;

... and so on.

Thus we compute $U_0(h)$ with error $O(h^5)$ and also $U_0(h/2)$ with error $O(h^4)$.

3.1 Example (VII)

6- A_1:

$$
\begin{aligned}
A_1(0) &= C_{0_-}(0) \qquad \text{error } O(h^5) \\
A_1(h/2) &= U_0^{-1}(h/2)C_{0_-}(h/2)U_0(h/2) \qquad \text{error } O(h^4) \\
A_1(h) &= U_0^{-1}(h)C_{0_-}(h)U_0(h) \qquad \text{error } O(h^5)
\end{aligned}
$$

... and the process is repeated again for the second cycle

3.1 Example (VII)

6- A_1:

$$
\begin{aligned}
A_1(0) &= C_{0\_}(0) \qquad \text{error } O(h^5) \\
A_1(h/2) &= U_0^{-1}(h/2)C_{0\_}(h/2)U_0(h/2) \qquad \text{error } O(h^4) \\
A_1(h) &= U_0^{-1}(h)C_{0\_}(h)U_0(h) \qquad \text{error } O(h^5)
\end{aligned}
$$

... and the process is repeated again for the second cycle

$\Rightarrow$ it is possible to construct a method of order 4 with only 2 $A(t)$ evaluations (3 for the first step).

3.2 Other possibilities

One could use other quadrature rules instead, for instance Gauss–Legendre, but...

3.2 Other possibilities

One could use other quadrature rules instead, for instance Gauss–Legendre, but...

in that case there are not enough nodes to approximate all the multivariate integrals up to the required order.

3.2 Other possibilities

One could use other quadrature rules instead, for instance Gauss–Legendre, but...

in that case there are not enough nodes to approximate all the multivariate integrals up to the required order.

Remark. This is not the case with Newton–Cotes rules, although the error introduced may be important for higher order (negative coefficients)

3.2 Other possibilities

One could use other quadrature rules instead, for instance Gauss–Legendre, but...

in that case there are not enough nodes to approximate all the multivariate integrals up to the required order.

Remark. This is not the case with Newton–Cotes rules, although the error introduced may be important for higher order (negative coefficients)

Solution: use G–L with matrix evaluations in the previous/next step.

3.2 Other possibilities

One could use other quadrature rules instead, for instance Gauss–Legendre, but...

in that case there are not enough nodes to approximate all the multivariate integrals up to the required order.

Remark. This is not the case with Newton–Cotes rules, although the error introduced may be important for higher order (negative coefficients)

Solution: use G–L with matrix evaluations in the previous/next step.

$\Rightarrow$ method of order 4 with 2 evaluations (and 1 in the next step)

3.3 Some methods

Order 4

$$Y \approx L_0 U_0 L_1 U_1$$

$*$ Quadratures NC / GL, 2 matrix evaluations per step

Order 6

$$Y \approx L_0 U_0 L_1 U_1 L_2$$

$*$ order 6 with a 5 points NC quadrature (4 evaluations per step)

$*$ order 7 with a 7 points NC (6 evaluations)

Order 12

$$Y \approx L_0 U_0 L_1 U_1 L_2 U_2$$

$*$ with a 11 points NC (or GL involving several steps).

3.4 Variable step size

Local extrapolation technique is trivial to implement in this setting.

3.4 Variable step size

Local extrapolation technique is trivial to implement in this setting.

For instance,

$$\begin{aligned} Y_1 &\equiv L_0 U_0 L_1 \\ \hat{Y}_1 &\equiv L_0 U_0 L_1 U_1 = Y_1 U_1 \end{aligned}$$

Then

$$\hat{Y}_1 - Y_1 = Y_1 U_1 - Y_1 = Y_1 (U_1 - I)$$

and $\|\hat{Y}_1 - Y_1\|$ can be used as a measure of the error

3.5 Future work

∗ Analyse the convergence of the procedure

3.5 Future work

- ∗ Analyse the convergence of the procedure
- ∗ Consider numerical examples in $\mathrm{SL}(n)$ with (very) large n

3.5 Future work

- ∗ Analyse the convergence of the procedure
- ∗ Consider numerical examples in $\mathrm{SL}(n)$ with (very) large n
- ∗ Highly oscillatory problems (with special quadratures)

3.5 Future work

- Analyse the convergence of the procedure
- Consider numerical examples in $\mathrm{SL}(n)$ with (very) large n
- Highly oscillatory problems (with special quadratures)
- Analyse in practice the preservation of other structures (Blanes & Moan)

3.5 Future work

- $*$ Analyse the convergence of the procedure
- $*$ Consider numerical examples in $\mathrm{SL}(n)$ with (very) large n
- $*$ Highly oscillatory problems (with special quadratures)
- $*$ Analyse in practice the preservation of other structures (Blanes & Moan)
- $*$ Try to generalize to nonlinear problems

The End

Fernando Casas

Universitat Jaume I, Castellón, Spain

Fernando.Casas@uji.es

(on sabbatical leave at DAMTP, University of Cambridge)