Runge–Kutta–Möbius methods

András Molnár

joint work with Imre Fekete and Gustaf Söderlind

July 2, 2019

Motivation

Goal

Construction of a new kind of Runge–Kutta method for the solution of the equation

$$\dot{x} = f(x).$$

Motivation

First requirement

- Assuming some conditions, the method should behave nicely regardless of the size of $L[hf]$.

Motivation

First requirement

- Assuming some conditions, the method should behave nicely regardless of the size of $L[hf]$.

$$L[f] = \sup_{u \neq v} \frac{\|f(u) - f(v)\|}{\|u - v\|}$$

Motivation

First requirement

- Assuming some conditions, the method should behave nicely regardless of the size of $L[hf]$.

$$L[f] = \sup_{u \neq v} \frac{\|f(u) - f(v)\|}{\|u - v\|}$$

- For example assuming A-stability is a good start.

Motivation

First requirement

- Assuming some conditions, the method should behave nicely regardless of the size of $L[hf]$.

$$L[f] = \sup_{u \neq v} \frac{\|f(u) - f(v)\|}{\|u - v\|}$$

- For example assuming A-stability is a good start.

A-stability, from another angle

Assuming $M[hA] \leq 0$, the norm of the numerical solution should be monotonically non-increasing in time.

Logarithmic Lipschitz-Constant

$$M[f] = \sup_{u \neq v} \frac{\mathrm{Re}\langle u - v, f(u) - f(v) \rangle}{\langle u - v, u - v \rangle}$$

Logarithmic Lipschitz-Constant

$$M[f] = \sup_{u \neq v} \frac{\mathrm{Re}\langle u - v, f(u) - f(v)\rangle}{\langle u - v, u - v\rangle}$$

The linear ($f = A$) case may be more familiar.

$$m[A]\langle u, u\rangle \leq \mathrm{Re}\langle u, Au\rangle \leq M[A]\langle u, u\rangle.$$

Logarithmic Lipschitz-Constant

$$M[f] = \sup_{u \neq v} \frac{\operatorname{Re}\langle u - v, f(u) - f(v) \rangle}{\langle u - v, u - v \rangle}$$

The linear ($f = A$) case may be more familiar.

$$m[A]\langle u, u \rangle \leq \operatorname{Re}\langle u, Au \rangle \leq M[A]\langle u, u \rangle.$$

If $M[hf] \leq 0$, and $\dot{u} = Au$, then

$$\frac{1}{2}\frac{d}{dt}\langle u, u \rangle = \frac{1}{2}\left(\langle u, \dot{u} \rangle + \langle \dot{u}, u \rangle\right) = \operatorname{Re}\langle u, Au \rangle \leq M[A]\langle u, u \rangle.$$

Motivation

First requirement

Assuming $M[hf] \leq 0$, the numerical method should behave nicely regardless of the size of $L[hf]$.

Motivation

First requirement

Assuming $M[hf] \leq 0$, the numerical method should behave nicely regardless of the size of $L[hf]$.

Second requirement

Let us use explicit Runge–Kutta methods

The "resolution" of the contradiction

Fundamental idea

Let's solve the equation

$$\dot{x} = g_h(x).$$

The "resolution" of the contradiction

Fundamental idea

Let's solve the equation

$$\dot{x} = g_h(x).$$

Where if $M[hf] \leq 0$, then

- if $h \to 0$, the solutions should tend to the solution of the original problem.

The "resolution" of the contradiction

Fundamental idea

Let's solve the equation

$$\dot{x} = g_h(x).$$

Where if $M[hf] \leq 0$, then

- if $h \to 0$, the solutions should tend to the solution of the original problem.
- $L[hg] \approx 1$ should hold independently of $L[hf]$.

The "resolution" of the contradiction

If $hf(x) = \lambda hx$, then

- $hg_h(x) = \mu(h) \cdot x$

The "resolution" of the contradiction

If $hf(x) = \lambda hx$, then

- $hg_h(x) = \mu(h) \cdot x$
- $M[hf] = \operatorname{Re} h\lambda$

The "resolution" of the contradiction

If $hf(x) = \lambda hx$, then

- $hg_h(x) = \mu(h) \cdot x$
- $M[hf] = \mathrm{Re}\, h\lambda$
- $L[hg] = |h\mu|$

The "resolution" of the contradiction

If $hf(x) = \lambda hx$, then

- $hg_h(x) = \mu(h) \cdot x$
- $M[hf] = \operatorname{Re} h\lambda \qquad \leq 0$
- $L[hg] = |h\mu|$

The "resolution" of the contradiction

If $hf(x) = \lambda hx$, then

- $hg_h(x) = \mu(h) \cdot x$
- $M[hf] = \mathrm{Re}\, h\lambda \qquad \leq 0$
- $L[hg] = |h\mu| \qquad \approx 1$

The “resolution” of the contradiction

If $hf(x) = \lambda hx$, then

- $hg_h(x) = \mu(h) \cdot x$
- $M[hf] = \operatorname{Re} h\lambda \qquad \leq 0$
- $L[hg] = |h\mu| \qquad \approx 1$

Möbius-transformations!

$$z \mapsto \frac{z}{1 - \gamma z}$$

The "resolution" of the contradiction

If $hf(x) = \lambda hx$, then

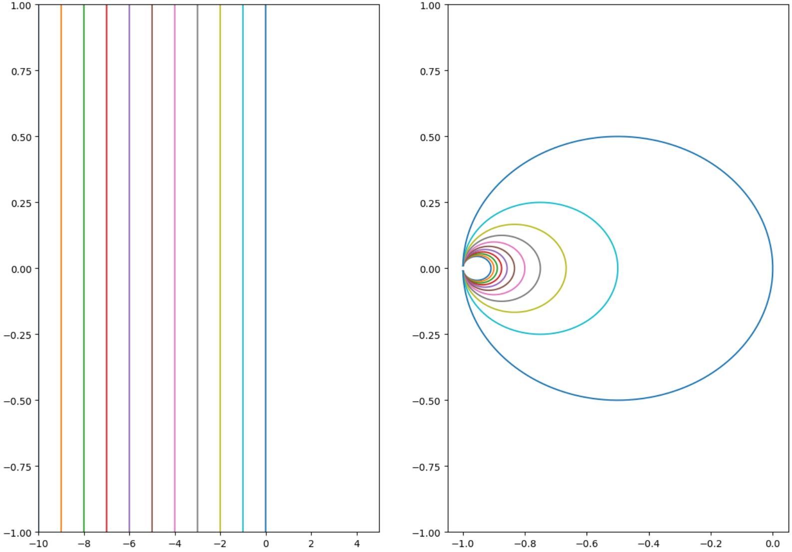
- $hg_h(x) = \mu(h) \cdot x$
- $M[hf] = \operatorname{Re} h\lambda \qquad \leq 0$
- $L[hg] = |h\mu| \qquad \approx 1$

Möbius-transformations!

$$z \mapsto \frac{z}{1 - \gamma z}$$

It follows

$$\operatorname{Re} z \leq 0 \quad \Rightarrow \quad \left| \frac{z}{1 - \gamma z} \right| \leq \frac{1}{\gamma}$$


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−10
−8
−6
−4
−2
0
2
4
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.0
−0.8
−0.6
−0.4
−0.2
0.0

The "resolution" of the contradiction

The transformed vector field hg_h

$$hf \quad \mapsto \quad hf \circ (I - \gamma hf)^{-1}$$

The "resolution" of the contradiction

The transformed vector field hg_h

$$hf \quad \mapsto \quad hf \circ (I - \gamma hf)^{-1}$$

It can be shown that

$$M[hf] \leq 0 \quad \Rightarrow \quad L[hf \circ (I - \gamma hf)^{-1}] \leq \frac{1}{\gamma}$$

On the implementation

Technologies used

- Python
- Octave
- PyTest

On the implementation

Technologies used

- Python (scipy, pyplot, ...)
- Octave
- PyTest

The finished solver includes

- General IRK, ERK, RK–Möbius solvers.
- Numerical order estimators.
- Iterative methods for the solution of the nonlinear system of equations.

ERK-SDIRK equivalency in general

$$\begin{array}{|cccc} \gamma & & & \\ a_{21} & \gamma & & \\ \vdots & & \ddots & \\ a_{s1} & a_{s2} & a_{s,s-1} & \gamma \\ \hline b_1 & b_2 & \dots & b_s \end{array} \quad \Leftrightarrow \quad \begin{array}{|cccc} 0 & & & \\ a_{21} & 0 & & \\ \vdots & & \ddots & \\ a_{s1} & a_{s2} & a_{s,s-1} & 0 \\ \hline b_1 & b_2 & \dots & b_s \end{array}$$

$$\dot{x} = f(x) \qquad \dot{x} = \left(f \circ (I - \gamma h f)^{-1}\right)(x)$$

The effect of the correspondence on the order of the methods

Question

Does the transformation keep the order of the ERK methods?

The effect of the correspondence on the order of the methods

Question

Does the transformation keep the order of the ERK methods?

Answer

No, methods of order at least two are reduced to first order.

The effect of the correspondence on the order of the methods

Question

Does the transformation keep the order of the ERK methods?

Answer

No, methods of order at least two are reduced to first order.

Loophole

Let's modify the quadrature vectors.

The effect of the correspondence on the order of the methods

Question

Does the transformation keep the order of the ERK methods?

Answer

No, methods of order at least two are reduced to first order.

Loophole

Let's modify the quadrature vectors. Necessary conditions yield a formula.

Notations

- The Möbius-correspondence image of an ERK method with Butcher-tableau $[A, d]$ shall be denoted $[A + \gamma I, b]$.

Notations

- The Möbius-correspondence image of an ERK method with Butcher-tableau $[A, d]$ shall be denoted $[A + \gamma I, b]$.

- Let us denote by $(s, p \to q)$ the set of s-stage Runge–Kutta pairs such that the first member of the pair is an explicit RK method of order p, and the other member is the corresponding SDIRK method of order q.

Second order is lost

The order condition for the explicit method with tableau $[A, d]$.

$$\sum_i d_i \sum_j a_{ij} = \frac{1}{2}$$

Second order is lost

The order condition for the explicit method with tableau $[A, d]$.

$$\sum_i d_i \sum_j a_{ij} = \frac{1}{2}$$

The order condition for Möbius-correspondent with tableau $[A + \gamma I, d]$.

$$\sum_i d_i \left(\sum_j a_{ij} + \gamma \right) = \sum_i d_i \left(\sum_j a_{ij} \right) + \gamma = \frac{1}{2}$$

The effect of the correspondence on the order of the methods

Question

Which are the method pairs for which the order is kept, perhaps even increased?

1-stage methods

$(1, 1 \to 1)$

The pairs are of the form (EE, EEM(γ)), where EE is the explicit Euler method, and the Butcher-tableau of the EEM(γ) method is $[\gamma, 1]$.

These pairs are A-stable if $\frac{1}{2} \leq \gamma$.

On the A-stability of SDIRK methods

The stability-function of an SDIRK $[A+\gamma I, d]$ method

$$R(z) = \frac{R_A(z)}{(z-\gamma)^s}$$

On the A-stability of SDIRK methods

The stability-function of an SDIRK $[A+\gamma I, d]$ method

$$R(z) = \frac{R_A(z)}{(z-\gamma)^s}$$

I-stability is enough

R is holomorphic on the LHP, thus by the maximum principle

$$|R|_{i\mathbb{R}}| \leq 1$$

is enough for A-stability.

1-stage methods

$(1, 1 \to 1)$

The pairs are of the form (EE, EEM(γ)), where EE is the explicit Euler method, and the Butcher-tableau of the EEM(γ) method is $[\gamma, 1]$.
These pairs are A-stable if $\frac{1}{2} \leq \gamma$.

$(1, 1 \to 2)$

There's only a single appropriate pair, the A-stable (EE, midpoint-method) pair.

2-stage methods

$(2, 2 \to 2)$

Second order can be kept. The tableau of the appropriate methods are of the form

$$
\begin{array}{c|cc}
 & & \\
 & a_{21} & \\
\hline
 & \cdot & d_2
\end{array}
\qquad\qquad
\begin{array}{c|cc}
 & \gamma & \\
 & \frac{1}{2d_2} & \gamma \\
\hline
 & \cdot & d_2(1-2\gamma).
\end{array}
$$

The pairs are A-stable for $\frac{1}{4} \le \gamma$.

$(2, 2 \to 3)$

There are two such pairs. The appropriate parameters are:

$$a_{21} = \mp\frac{1}{\sqrt{3}}, \quad d_2 = \mp\frac{\sqrt{3}}{2}, \quad \gamma = \frac{1}{2} \pm \frac{1}{2\sqrt{3}}.$$

3-stage methods

$(3, 3 \to 3)$

- Third order can be kept, i.e. $(3, 3 \to 3) \neq \emptyset$, but the formulae get more complicated.
- The pairs' A-stability holds for $\frac{1}{3} \leq \gamma \leq 1.06857902$.
- While the literature prefers the so called stiffly-accurate SDIRK methods, such pairs can not be constructed.

$(3, 3 \to 4)$

No third order ERK method can be transformed to a fourth order SDIRK in this way, so there are no such pairs.

On symbolical computations

Generating the order conditions

A program written in Haskell generates the Butcher-trees and the corresponding order conditions in a format that is digestible by the symbolical packages.

On symbolical computations

Generating the order conditions

A program written in Haskell generates the Butcher-trees and the corresponding order conditions in a format that is digestible by the symbolical packages.

Merging the order conditions

We then look for ERK-SDIRK paris satisfying the aforementioned order conditions using Mathematica and SymPy.

A 3-stage method family

An A-stable family with a simple tableau is the following.

$$
\begin{array}{c|ccc}
 & & & \\
 & \frac{2}{3} & & \\
 & -a & a & \\
\hline
 & \cdot & \frac{3}{4} & \frac{1}{4a}
\end{array}
\qquad
\begin{array}{c|ccc}
 & \frac{1}{3} & & \\
 & \frac{2}{3} & \frac{1}{3} & \\
 & -a & a & \frac{1}{3} \\
\hline
 & \cdot & \frac{1}{4} & -\frac{1}{12a}
\end{array}
$$

A two-component algorithm using the 3-stage method family

Fundamental idea

Let us estimate the (local) stiffness of the problem. If the problem is stiff, then let us step using the SDIRK method. Otherwise step using the ERK.

A two-component algorithm using the 3-stage method family

Fundamental idea

Let us estimate the (local) stiffness of the problem. If the problem is stiff, then let us step using the SDIRK method. Otherwise step using the ERK.

Embedded pairs

For the step-size selection we need local error estimates, thus embedded pairs:

$$
\begin{array}{c|ccc}
 & & & \\
 & \frac{2}{3} & & \\
 & -a & a & \\
\hline
d & \cdot & \frac{3}{4} & \frac{1}{4a} \\
\hline
\hat{d} & \cdot & \frac{3}{4} & \rho
\end{array}
\qquad\qquad
\begin{array}{c|ccc}
 & \frac{1}{3} & & \\
 & \frac{2}{3} & \frac{1}{3} & \\
 & -a & a & \frac{1}{3} \\
\hline
b & \cdot & \frac{1}{4} & -\frac{1}{12a} \\
\hline
\hat{b} & \cdot & \frac{1}{4} & \tau
\end{array}
$$

Notation

$$\rho_n = \frac{h_{n+1}}{h_n}, \qquad c_n = \left(\frac{\mathsf{TOL}}{r_n}\right)^{1/k}$$

A two-component algorithm using the 3-stage method family

Notation

$$\rho_n = \frac{h_{n+1}}{h_n}, \qquad c_n = \left(\frac{\mathsf{TOL}}{r_n}\right)^{1/k}$$

Controller for the ERK method: PI3333

$$\rho_n = c_n^{2/3} c_{n-1}^{-1/3}$$
$$h_{n+1} = \rho_n h_n$$

A two-component algorithm using the 3-stage method family

Notation

$$\rho_n = \frac{h_{n+1}}{h_n}, \qquad c_n = \left(\frac{\mathsf{TOL}}{r_n}\right)^{1/k}$$

Controller for the ERK method: PI3333

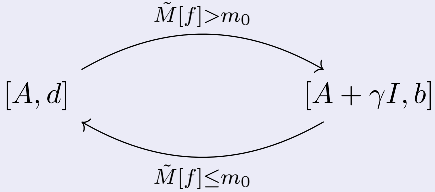
$$\rho_n = c_n^{2/3} c_{n-1}^{-1/3}$$
$$h_{n+1} = \rho_n h_n$$

Controller for the SDIRK method: H211b

$$\rho_n = c_n^{1/b} c_{n-1}^{1/b} \rho_{n-1}^{-1/b}$$
$$h_{n+1} = \rho_n h_n$$

Switching between the methods

The combined algorithm



Numerical experiments: An instructive example

Prothero-Robinson problem

$$\dot{x} = \lambda(x - g) + \dot{g}$$

Parametrization:

$$t \in (0, \pi), \qquad g(t) = \sin(t), \qquad \lambda = -500, \qquad x_0 = 1$$

Numerical experiments: An instructive example

Prothero-Robinson problem

$$\dot{x} = \lambda(x - g) + \dot{g}$$

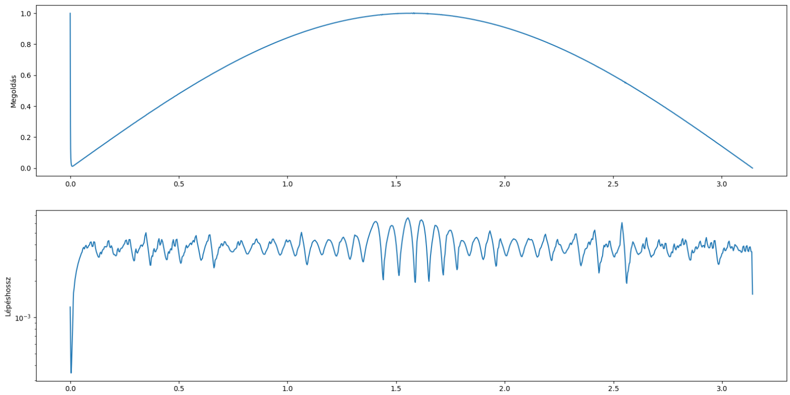
Parametrization:

$$t \in (0, \pi), \qquad g(t) = \sin(t), \qquad \lambda = -500, \qquad x_0 = 1$$

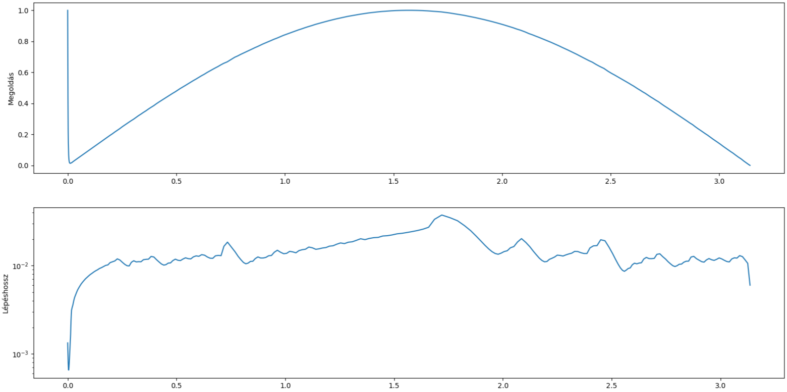
Parametrization of the methods: $a = 1$, $\rho = 1$

- Setting $\tau = 1$, the embedded pair is not A-stable.
- Letting $\tau = 0$, the embedded pair is A-stable.

Numerical experiments: non A-stable embedded pair



Numerical experiments: A-stable embedded pair



Conclusion

- Möbius-transformations shine a new light on the SDIRK methods.

Conclusion

- Möbius-transformations shine a new light on the SDIRK methods.
- At the same time we get a natural correspondence between ERK and SDIRK methods.

Conclusion

- Möbius-transformations shine a new light on the SDIRK methods.
- At the same time we get a natural correspondence between ERK and SDIRK methods.
- In general this correspondence trades the explicit methods' higher order for the implicit methods' better stability.

Conclusion

- Möbius-transformations shine a new light on the SDIRK methods.
- At the same time we get a natural correspondence between ERK and SDIRK methods.
- In general this correspondence trades the explicit methods' higher order for the implicit methods' better stability.
- However, for selected pairs of methods, the better stability is achievable without the loss of precision.

Conclusion

- Möbius-transformations shine a new light on the SDIRK methods.
- At the same time we get a natural correspondence between ERK and SDIRK methods.
- In general this correspondence trades the explicit methods' higher order for the implicit methods' better stability.
- However, for selected pairs of methods, the better stability is achievable without the loss of precision.
- Moreover, these pairs can be used to construct a working numerical method.

What's next?

More numerical tests

Rigorous testing on stiff problems, preferably on systems from real world applications.