Introduction to Simulation - Lecture 13

Convergence of Multistep Methods

Jacob White

Thanks to Deepak Ramaswamy, Michal Rewienski, and Karen Veroy
Outline

Small Timestep issues for Multistep Methods
- Local truncation error
- Selecting coefficients.
- Nonconverging methods.
- Stability + Consistency implies convergence

Next Time Investigate Large Timestep Issues
- Absolute Stability for two time-scale examples.
- Oscillators.
Multistep Methods

Nonlinear Differential Equation:
\[
\frac{d}{dt} x(t) = f(x(t), u(t))
\]

k-Step Multistep Approach:
\[
\sum_{j=0}^{k} \alpha_j \hat{x}^{l-j} = \Delta t \sum_{j=0}^{k} \beta_j f\left(\hat{x}^{l-j}, u\left(t_{l-j}\right)\right)
\]

Solution at discrete points

Multistep coefficients

Time discretization

General Notation
Multistep Methods

Basic Equations

Common Algorithms

Multistep Equation: \[ \sum_{j=0}^{k} \alpha_j \hat{x}^{l-j} = \Delta t \sum_{j=0}^{k} \beta_j f \left( \hat{x}^{l-j}, u \left( t_{l-j} \right) \right) \]

Forward-Euler Approximation: \[ x \left( t_l \right) \approx x \left( t_{l-1} \right) + \Delta t f \left( x \left( t_{l-1} \right), u \left( t_{l-1} \right) \right) \]

FE Discrete Equation: \[ \hat{x}^l - \hat{x}^{l-1} = \Delta t f \left( \hat{x}^{l-1}, u \left( t_{l-1} \right) \right) \]

Multistep Coefficients: \[ k = 1, \alpha_0 = 1, \alpha_1 = -1, \beta_0 = 0, \beta_1 = 1 \]

BE Discrete Equation: \[ \hat{x}^l - \hat{x}^{l-1} = \Delta t f \left( \hat{x}^l, u \left( t_l \right) \right) \]

Multistep Coefficients: \[ k = 1, \alpha_0 = 1, \alpha_1 = -1, \beta_0 = 1, \beta_1 = 0 \]

Trap Discrete Equation: \[ \hat{x}^l - \hat{x}^{l-1} = \frac{\Delta t}{2} \left( f \left( \hat{x}^l, u \left( t_l \right) \right) + f \left( \hat{x}^{l-1}, u \left( t_{l-1} \right) \right) \right) \]

Multistep Coefficients: \[ k = 1, \alpha_0 = 1, \alpha_1 = -1, \beta_0 = \frac{1}{2}, \beta_1 = \frac{1}{2} \]
Multistep Methods

Basic Equations

Definitions and Observations

Multistep Equation: \[ \sum_{j=0}^{k} \alpha_j \hat{x}^{l-j} = \Delta t \sum_{j=0}^{k} \beta_j f \left( \hat{x}^{l-j}, u(t_{l-j}) \right) \]

1) If \( \beta_0 \neq 0 \) the multistep method is implicit
2) A \( k \) – step multistep method uses \( k \) previous \( x \)'s and \( f \)'s
3) A normalization is needed, \( \alpha_0 = 1 \) is common
4) A \( k \)-step method has \( 2k + 1 \) free coefficients

How does one pick good coefficients?

Want the highest accuracy
**Multistep Methods**

**Simplified Problem for Analysis**

Scalar ODE: \( \frac{d}{dt} v(t) = \lambda v(t), \ v(0) = v_0 \quad \lambda \in \mathbb{C} \)

Why such a simple Test Problem?

- Nonlinear Analysis has many \textit{unrevealing} subtleties
- Scalar is equivalent to vector for multistep methods.

\[ \frac{d}{dt} x(t) = Ax(t) \text{ multistep discretization} \]

Let \( Ey(t) = x(t) \)

\[ \sum_{j=0}^{k} \alpha_j \hat{x}^{l-j} = \Delta t \sum_{j=0}^{k} \beta_j A \hat{x}^{l-j} \]

Decoupled Equations

\[ \sum_{j=0}^{k} \alpha_j \hat{y}^{l-j} = \Delta t \sum_{j=0}^{k} \beta_j E^{-1} AE \hat{y}^{l-j} \]
**Multistep Methods**

**Simplified Problem for Analysis**

Scalar ODE:

\[
\frac{d}{dt}v(t) = \lambda v(t), \quad v(0) = v_0 \quad \lambda \in \mathbb{C}
\]

Scalar Multistep formula:

\[
\sum_{j=0}^{k} \alpha_j \hat{v}^{l-j} = \Delta t \sum_{j=0}^{k} \beta_j \lambda \hat{v}^{l-j}
\]

**Must Consider ALL \( \lambda \in \mathbb{C} \)**

- **Decaying Solutions**
- **Growing Solutions**
- **Oscillations**

**Re(\( \lambda \))**

**Im(\( \lambda \))**
Definition: A multistep method for solving initial value problems on $[0,T]$ is said to be convergent if given any initial condition

$$\max_{l \in \left[0, \frac{T}{\Delta t}\right]} \left\| \hat{v}^l - v(l\Delta t) \right\| \to 0 \text{ as } \Delta t \to 0$$

$\hat{v}^l$ computed with $\Delta t$

$\hat{v}^l$ computed with $\frac{\Delta t}{2}$

$v_{exact}$
Definition: A multi-step method for solving initial value problems on \([0,T]\) is said to be order p convergent if given any \(\lambda\) and any initial condition

\[
\max_{l \in \left[0, \frac{T}{\Delta t}\right]} \| \hat{v}^l - v(l\Delta t) \| \leq C (\Delta t)^p
\]

for all \(\Delta t\) less than a given \(\Delta t_0\)

Forward- and Backward-Euler are order 1 convergent
Trapezoidal Rule is order 2 convergent
Multi-step Methods

Convergence Analysis

Reaction Equation Example

For FE and BE, \( \text{Error} \propto \Delta t \)

For Trap, \( \text{Error} \propto (\Delta t)^2 \)
Multistep Methods

Convergence Analysis

Two Conditions for Convergence

1) Local Condition: “One step” errors are small (consistency)
   Typically verified using Taylor Series

2) Global Condition: The single step errors do not grow too quickly (stability)
   All one-step (k=1) methods are stable in this sense.
   Multi-step (k > 1) methods require careful analysis.
Multistep Methods

Multistep formula:

\[ \sum_{j=0}^{k} \alpha_j \hat{v}^{l-j} - \Delta t \sum_{j=0}^{k} \beta_j \hat{\lambda} \hat{v}^{l-j} = 0 \]

Exact solution Almost satisfies Multistep Formula:

\[ \sum_{j=0}^{k} \alpha_j v(t_{l-j}) - \Delta t \sum_{j=0}^{k} \beta_j \frac{d}{dt} v(t_{l-j}) = e^l \]

Global Error: \( E^l \equiv v(t_l) - \hat{v}^l \)

Difference equation relates LTE to Global error

\[ \left( \alpha_0 - \lambda \Delta t \beta_0 \right) E^l + \left( \alpha_1 - \lambda \Delta t \beta_1 \right) E^{l-1} + \cdots + \left( \alpha_k - \lambda \Delta t \beta_k \right) E^{l-k} = e^l \]
Forward-Euler definition

\[ \hat{v}^{l+1} - \hat{v}^l - \Delta t \lambda \hat{v}^l = 0 \]

Substituting the exact \( v(t) \) and expanding

\[
v((l+1)\Delta t) - v(l\Delta t) - \Delta t \frac{dv(l\Delta t)}{dt} = \frac{(\Delta t)^2}{2} \frac{d^2 v(\tau)}{dt^2} e^l \]

where \( e^l \) is the LTE and is bounded by

\[ |e^l| \leq C(\Delta t)^2, \text{ where } C = 0.5 \max_{\tau \in [0,T]} \left| \frac{d^2 v(\tau)}{dt^2} \right| \]
Forward-Euler definition
\[ \hat{v}^{l+1} = \hat{v}^l + \Delta t \lambda \hat{v}^l \]

Using the LTE definition
\[ v((l+1)\Delta t) = v(l\Delta t) + \Delta t \lambda v(l\Delta t) + e^l \]

Subtracting yields global error equation
\[ E^{l+1} = (I + \Delta t \lambda) E^l + e^l \]

Using magnitudes and the bound on \( e^l \)
\[ |E^{l+1}| \leq |I + \Delta t \lambda| |E^l| + |e^l| \leq (1 + \Delta t |\lambda|) |E^l| + C (\Delta t)^2 \]
Forward-Euler

Convergence Analysis

A helpful bound on difference equations

A lemma bounding difference equation solutions

If \[ |u^{l+1}| \leq (1 + \varepsilon)|u^l| + b, \quad u^0 = 0, \quad \varepsilon > 0 \]

Then \[ |u^l| \leq \frac{e^{\varepsilon l}}{\varepsilon} |b| \]

To prove, first write \( u^l \) as a power series and sum

\[ |u^l| \leq \sum_{j=0}^{l-1} (1 + \varepsilon)^j |b| = \frac{1 - (1 + \varepsilon)^l}{1 - (1 + \varepsilon)} |b| \]
To finish, note \((1 + \varepsilon) \leq e^\varepsilon \Rightarrow (1 + \varepsilon)^l \leq e^{\varepsilon l}\)

\[
\left| u^l \right| \leq \frac{1 - (1 + \varepsilon)^l}{1 - (1 + \varepsilon)} \left| b \right| = \frac{(1 + \varepsilon)^l - 1}{\varepsilon} \left| b \right| \leq \frac{e^{\varepsilon l}}{\varepsilon} \left| b \right|
\]
One-step Methods

Convergence Analysis

Back to Forward Euler
Convergence analysis.

Applying the lemma and cancelling terms

\[ |E^{l+1}| \leq \left( 1 + \frac{\Delta t |\lambda|}{\varepsilon} \right) |E^l| + C (\Delta t)^2 \leq \frac{e^{l\Delta t |\lambda|}}{\Delta t |\lambda|} C (\Delta t)^2 \]

Finally noting that \( l\Delta t \leq T \),

\[ \max_{l \in [0,L]} |E^l| \leq e^{\frac{|\lambda| T}{|\lambda|}} \frac{C}{\Delta t} \]
Forward-Euler

Convergence Analysis

Observations about the forward-Euler analysis.

\[
\max_{t \in [0, L]} |E^l| \leq e^{\lambda T} \frac{C}{|\lambda|} \Delta t
\]

- forward-Euler is order 1 convergent
- Bound grows exponentially with time interval.
- \( C \) related to exact solution’s second derivative.
- The bound grows exponentially with time.
Convergence Analysis

Exact and forward-Euler (FE)
Plots for Unstable Reaction.

Forward-Euler Errors appear to grow with time
Note error grows exponentially with time, as bound predicts.
Forward-Euler

Convergence Analysis

Exact and forward-Euler (FE) Plots for Circuit.

Forward-Euler Errors don’t always grow with time
Error does not always grow exponentially with time!

**Bound is conservative**
Multistep Methods

Making LTE Small

Exactness Constraints

Local Truncation Error: 
\[\sum_{j=0}^{k} \alpha_j v(t_{k-j}) - \Delta t \sum_{j=0}^{k} \beta_j \frac{d}{dt} v(t_{k-j}) = e^k\]

Can't be from 
\[\frac{d}{dt} v(t) = \lambda v(t)\]

If \[v(t) = t^p \Rightarrow \frac{d}{dt} v(t) = pt^{p-1}\]

\[\sum_{j=0}^{k} \alpha_j \left((k-j)\Delta t\right)^p - \Delta t \sum_{j=0}^{k} \beta_j p\left((k-j)\Delta t\right)^{p-1} = e^k\]
Multistep Methods

Exactness Constraints Cont.

\[
\sum_{j=0}^{k} \alpha_j ((k - j) \Delta t)^p - \Delta t \sum_{j=0}^{k} \beta_j p ((k - j) \Delta t)^{p-1} = \\
(\Delta t)^p \left( \sum_{j=0}^{k} \alpha_j (l - j)^p - \sum_{j=0}^{k} \beta_j p (l - j)^{p-1} \right) = e^k
\]

If \( \left( \sum_{j=0}^{k} \alpha_j ((k - j))^p - \sum_{j=0}^{k} \beta_j p (k - j)^{p-1} \right) = 0 \) then \( e^k = 0 \) for \( v(t) = t^p \)

As any smooth \( v(t) \) has a locally accurate Taylor series in \( t \):

if \( \left( \sum_{j=0}^{k} \alpha_j (k - j)^p - \sum_{j=0}^{k} \beta_j p (k - j)^{p-1} \right) = 0 \) for all \( p \leq p_0 \)

Then \( \left( \sum_{j=0}^{k} \alpha_j v(t_{l-j}) - \sum_{j=0}^{k} \beta_j \frac{d}{dt} v(t_{l-j}) \right) = e^l = C(\Delta t)^{p_0+1} \)
Exactness Constraints: \[
\sum_{j=0}^{k} \alpha_j (k - j)^p - \sum_{j=0}^{k} \beta_j p (k - j)^{p-1} = 0
\]

For \(k=2\), yields a 5x6 system of equations for Coefficients

<table>
<thead>
<tr>
<th>(p)</th>
<th>(1)</th>
<th>(1)</th>
<th>(1)</th>
<th>(0)</th>
<th>(0)</th>
<th>(0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p=0)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
</tr>
<tr>
<td>(p=1)</td>
<td>(2)</td>
<td>(1)</td>
<td>(0)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
</tr>
<tr>
<td>(p=2)</td>
<td>(4)</td>
<td>(1)</td>
<td>(0)</td>
<td>(-4)</td>
<td>(-2)</td>
<td>(0)</td>
</tr>
<tr>
<td>(p=3)</td>
<td>(8)</td>
<td>(1)</td>
<td>(0)</td>
<td>(-12)</td>
<td>(-3)</td>
<td>(0)</td>
</tr>
<tr>
<td>(p=4)</td>
<td>(16)</td>
<td>(1)</td>
<td>(0)</td>
<td>(-32)</td>
<td>(-4)</td>
<td>(0)</td>
</tr>
</tbody>
</table>

\[\begin{bmatrix}
\alpha_0 \\
\alpha_1 \\
\alpha_2 \\
\beta_0 \\
\beta_1 \\
\beta_2 \\
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]

Note \(\sum \alpha_i = 0\)

Always
**Multistep Methods**

### Making LTE Small

#### Example Continued

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>-4</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>-12</td>
<td>-3</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>0</td>
<td>-32</td>
<td>-4</td>
<td>0</td>
</tr>
</tbody>
</table>

$$
\begin{bmatrix}
\alpha_0 \\
\alpha_1 \\
\alpha_2 \\
\beta_0 \\
\beta_1 \\
\beta_2 \\
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
$$

**Exactness Constraints for k=2**

- **Forward-Euler** $\alpha_0 = 1, \ \alpha_1 = -1, \ \alpha_2 = 0, \ \beta_0 = 0, \ \beta_1 = 1, \ \beta_2 = 0$,
  - FE satisfies $p = 0$ and $p = 1$ but not $p = 2 \Rightarrow LTE = C(\Delta t)^2$

- **Backward-Euler** $\alpha_0 = 1, \ \alpha_1 = -1, \ \alpha_2 = 0, \ \beta_0 = 1, \ \beta_1 = 0, \ \beta_2 = 0$,
  - BE satisfies $p = 0$ and $p = 1$ but not $p = 2 \Rightarrow LTE = C(\Delta t)^2$

- **Trap Rule** $\alpha_0 = 1, \ \alpha_1 = -1, \ \alpha_2 = 0, \ \beta_0 = 0.5, \ \beta_1 = 0.5, \ \beta_2 = 0$,
  - Trap satisfies $p = 0, 1, or 2$ but not $p = 3 \Rightarrow LTE = C(\Delta t)^3$

\[ p = 0 \Rightarrow LTE = \Delta t \]
First introduce a normalization, for example $\alpha_0 = 1$

\[
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
1 & 0 & -1 & -1 & -1 \\
1 & 0 & -4 & -2 & 0 \\
1 & 0 & -12 & -3 & 0 \\
1 & 0 & -32 & -4 & 0
\end{bmatrix} \begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\beta_0 \\
\beta_1 \\
\beta_2
\end{bmatrix} = \begin{bmatrix}
-1 \\
-2 \\
-4 \\
-8 \\
-16
\end{bmatrix}
\]

Solve for the 2-step method with lowest LTE

$\alpha_0 = 1, \alpha_1 = 0, \alpha_2 = -1, \beta_0 = 1/3, \beta_1 = 4/3, \beta_2 = 1/3$

Satisfies all five exactness constraints $LTE = C(\Delta t)^5$

Solve for the 2-step explicit method with lowest LTE

$\alpha_0 = 1, \alpha_1 = 4, \alpha_2 = -5, \beta_0 = 0, \beta_1 = 4, \beta_2 = 2$

Can only satisfy four exactness constraints $LTE = C(\Delta t)^4$
Multistep Methods

Making LTE Small

LTE Plots for the FE, Trap, and “Best” Explicit (BESTE).

\[ \frac{d}{dt} v(t) = v(t) \]

Best Explicit Method has highest one-step accurate
Multistep Methods

Making LTE Small

Global Error for the FE, Trap, and “Best” Explicit (BESTE).

\[ \frac{d}{dt} v(t) = v(t) \quad t \in [0,1] \]

Where’s BESTE?

Max Error vs. Timestep
Multistep Methods

Making LTE Small

Global Error for the FE, Trap, and “Best” Explicit (BESTE).

Best Explicit Method has lowest one-step error but global error increases as timestep decreases.
Why did the “best” 2-step explicit method fail to Converge?

\[
\nu(l\Delta t) - \hat{v}^l = (\alpha_0 - \lambda \Delta t \beta_0) E^l + (\alpha_1 - \lambda \Delta t \beta_1) E^{l-1} + \cdots + (\alpha_k - \lambda \Delta t \beta_k) E^{l-k} = e^l
\]

Global Error

We made the LTE so small, how come the Global error is so large?
An Aside on Solving Difference Equations

Consider a general kth order difference equation

\[ a_0 x^l + a_1 x^{l-1} + \cdots + a_k x^{l-k} = u^l \]

Which must have k initial conditions

\[ x^0 = x_0, \ x^{-1} = x_1, \ \cdots, \ x^{-k} = x_k \]

As is clear when the equation is in update form

\[ x^1 = -\frac{1}{a_0} \left( a_1 x^0 + \cdots + a_k x^{-k+1} - u^1 \right) \]

Most important difference equation result

\[ x \text{ can be related to } u \text{ by } x^l = \sum_{j=0}^{l} h^{l-j} u^j \]
An Aside on Difference Equations Cont.

If \( a_0 z^k + a_1 z^{k-1} + \cdots + a_k = 0 \) has distinct roots \( \varsigma_1, \varsigma_2, \cdots, \varsigma_k \)

Then \( x^l = \sum_{j=0}^{l} h^{l-j} u^j \) where \( h^l = \sum_{j=1}^{k} \gamma_j (\varsigma_j)^l \)

To understand how \( h \) is derived, first a simple case

Suppose \( x^l = \varsigma x^{l-1} + u^l \) and \( x^0 = 0 \)

\( x^1 = \varsigma x^0 + u^1 = u^1 \),  \( x^2 = \varsigma x^1 + u^2 = \varsigma u^1 + u^2 \)

\( x^l = \sum_{j=0}^{l} \varsigma^{l-j} u^j \)
Three important observations

If $|\zeta_i| < 1$ for all $i$, then $|x^l| \leq C \max_j |u^j|$ where $C$ does not depend on $l$

If $|\zeta_i| > 1$ for any $i$, then there exists a bounded $u^j$ such that $|x^l| \to \infty$

If $|\zeta_i| \leq 1$ for all $i$, and if $|\zeta_i| = 1$, $\zeta_i$ is distinct then $|x^l| \leq Cl \max_j |u^j|$
Multistep Method Difference Equation

\[(\alpha_0 - \lambda \Delta t \beta_0) E^l + (\alpha_1 - \lambda \Delta t \beta_1) E^{l-1} + \cdots + (\alpha_k - \lambda \Delta t \beta_k) E^{l-k} = e^l\]

**Definition:** A multistep method is stable if and only if

\[\max_{l \in [0, T/\Delta t]} \|E^l\| \leq C \frac{T}{\Delta t} \max_{l \in [0, T/\Delta t]} \|e^l\| \quad \text{for any } e^l\]

**Theorem:** A multistep method is stable if and only if

The roots of \(\alpha_0 z^k + \alpha_1 z^{k-1} + \cdots + \alpha_k = 0\) are either

Less than one in magnitude or equal to one and distinct
Given the Multistep Method Difference Equation

\[
(\alpha_0 - \lambda \Delta t \beta_0) E^l + (\alpha_1 - \lambda \Delta t \beta_1) E^{l-1} + \cdots + (\alpha_k - \lambda \Delta t \beta_k) E^{l-k} = e^l
\]

If the roots of \( \sum_{j=0}^{k} \alpha_j z^{k-j} = 0 \) are either

- less than one in magnitude
- equal to one in magnitude but distinct

Then from the aside on difference equations

\[
\|E^l\| \leq C l \max_{l} \|e^l\|
\]

From which stability easily follows.
roots of $\sum_{j=0}^{k} \alpha_j z^{k-j} = 0$

As $\Delta t \to 0$, roots move inward to match $\alpha$ polynomial.

roots of $\sum_{j=0}^{k} \left(\alpha_j - \lambda \Delta t \beta_j \right) z^{k-j} = 0$ for a nonzero $\Delta t$
Best explicit 2-step method
\[ \alpha_0 = 1, \quad \alpha_1 = 4, \quad \alpha_2 = -5, \quad \beta_0 = 0, \quad \beta_1 = 4, \quad \beta_2 = 2 \]

roots of \( z^2 + 4z - 5 = 0 \)

Method is Wildly unstable!
For a stable, explicit $k$-step multistep method, the maximum number of exactness constraints that can be satisfied is less than or equal to $k$ (note there are $2k$ coefficients). For implicit methods, the number of constraints that can be satisfied is either $k+2$ if $k$ is even or $k+1$ if $k$ is odd.
1) Local Condition: One step errors are small (consistency)

Exactness Constraints up to $p_0$ ($p_0$ must be $> 0$)

$$\Rightarrow \max_{l \in \left[0, \frac{T}{\Delta t}\right]} \left\| e^l \right\| \leq C_1 \left(\Delta t\right)^{p_0+1} \text{ for } \Delta t < \Delta t_0$$

2) Global Condition: One step errors grow slowly (stability)

roots of $\sum_{j=0}^{k} \alpha_j z^{k-j} = 0$ inside or simple on unit circle

$$\Rightarrow \max_{l \in \left[0, \frac{T}{\Delta t}\right]} \left\| E^l \right\| \leq C_2 \frac{T}{\Delta t} \max_{l \in \left[0, \frac{T}{\Delta t}\right]} \left\| e^l \right\|$$

Convergence Result: $$\max_{l \in \left[0, \frac{T}{\Delta t}\right]} \left\| E^l \right\| \leq C T \left(\Delta t\right)^{p_0}$$
Summary

Small Timestep issues for Multistep Methods
  Local truncation error and Exactness.
  Difference equation stability.
  Stability + Consistency implies convergence.

Next time
  Absolute Stability for two time-scale examples.
  Oscillators.
  Maybe Runge-Kutta schemes