Linearly Implicit Peer Methods for the Compressible Euler Equations

Stefan Jebens

16. Südostdeutsches Kolloquium

30.04.2010
1 Peer methods
   - Motivation
   - Formulation of the methods

2 Linear stability theory
   - Linearization of Euler equations
   - Amplitude and phase properties

3 Numerical tests
   - The 2D compressible Euler equations
   - Rising bubble
   - Flow over mountain
   - Zeppelin test

4 Conclusions
1. Peer methods
   - Motivation
   - Formulation of the methods

2. Linear stability theory
   - Linearization of Euler equations
   - Amplitude and phase properties

3. Numerical tests
   - The 2D compressible Euler equations
   - Rising bubble
   - Flow over mountain
   - Zeppelin test

4. Conclusions
<table>
<thead>
<tr>
<th>Peer methods</th>
<th>Linear stability theory</th>
<th>Numerical tests</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Peer methods
- Linear stability theory
- Numerical tests
- Conclusions
In compressible models occur:

- Energetically relevant slow waves (e.g. advection, Rossby waves)
- Energetically irrelevant fast waves (e.g. sound waves)

In explicit models the fast waves restrict the maximal time step size

One ansatz to overcome this is operator splitting

- Advantages: Every step is cheap, easy to implement, parallelization
- Disadvantages: Still explicit (i.e. only small time steps allowed especially when used together with cut-cells), complicated derivation of order conditions and stability results

Another ansatz is the use of implicit methods

- Advantages: Allows very big time steps, order conditions and stability issues are easier
- Disadvantages: Requires solution of huge (non-)linear systems of equations, needs efficient (parallel) preconditioners
In compressible models occur:

- Energetically relevant slow waves (e.g. advection, Rossby waves)
- Energetically irrelevant fast waves (e.g. sound waves)

In explicit models the fast waves restrict the maximal time step size

One ansatz to overcome this is operator splitting

- Advantages: Every step is cheap, easy to implement, parallelization
- Disadvantages: Still explicit (i.e. only small time steps allowed especially when used together with cut-cells), complicated derivation of order conditions and stability results

Another ansatz is the use of implicit methods

- Advantages: Allows very big time steps, order conditions and stability issues are easier
- Disadvantages: Requires solution of huge (non-)linear systems of equations, needs efficient (parallel) preconditioners
In compressible models occur:
- Energetically relevant slow waves (e.g. advection, Rossby waves)
- Energetically irrelevant fast waves (e.g. sound waves)

In explicit models the fast waves restrict the maximal time step size.

One ansatz to overcome this is operator splitting
- Advantages: Every step is cheap, easy to implement, parallelization
- Disadvantages: Still explicit (i.e. only small time steps allowed especially when used together with cut-cells), complicated derivation of order conditions and stability results

Another ansatz is the use of implicit methods
- Advantages: Allows very big time steps, order conditions and stability issues are easier
- Disadvantages: Requires solution of huge (non-)linear systems of equations, needs efficient (parallel) preconditioners
• In compressible models occur:
  • Energetically relevant slow waves (e.g. advection, Rossby waves)
  • Energetically irrelevant fast waves (e.g. sound waves)

• In explicit models the fast waves restrict the maximal time step size

• One ansatz to overcome this is operator splitting
  • Advantages: Every step is cheap, easy to implement, parallelization
  • Disadvantages: Still explicit (i.e. only small time steps allowed especially when used together with cut-cells), complicated derivation of order conditions and stability results

• Another ansatz is the use of implicit methods
  • Advantages: Allows very big time steps, order conditions and stability issues are easier
  • Disadvantages: Requires solution of huge (non-)linear systems of equations, needs efficient (parallel) preconditioners
In compressible models occur:

- Energetically relevant slow waves (e.g. advection, Rossby waves)
- Energetically irrelevant fast waves (e.g. sound waves)

In explicit models the fast waves restrict the maximal time step size

One ansatz to overcome this is operator splitting

- Advantages: Every step is cheap, easy to implement, parallelization
- Disadvantages: Still explicit (i.e. only small time steps allowed especially when used together with cut-cells), complicated derivation of order conditions and stability results

Another ansatz is the use of implicit methods

- Advantages: Allows very big time steps, order conditions and stability issues are easier
- Disadvantages: Requires solution of huge (non-)linear systems of equations, needs efficient (parallel) preconditioners
In compressible models occur:

- Energetically relevant slow waves (e.g. advection, Rossby waves)
- Energetically irrelevant fast waves (e.g. sound waves)

In explicit models the fast waves restrict the maximal time step size

One ansatz to overcome this is operator splitting

- Advantages: Every step is cheap, easy to implement, parallelization
- Disadvantages: Still explicit (i.e. only small time steps allowed especially when used together with cut-cells), complicated derivation of order conditions and stability results

Another ansatz is the use of implicit methods

- Advantages: Allows very big time steps, order conditions and stability issues are easier
- Disadvantages: Requires solution of huge (non-)linear systems of equations, needs efficient (parallel) preconditioners
In compressible models occur:
- Energetically relevant slow waves (e.g. advection, Rossby waves)
- Energetically irrelevant fast waves (e.g. sound waves)

In explicit models the fast waves restrict the maximal time step size

One ansatz to overcome this is operator splitting
- Advantages: Every step is cheap, easy to implement, parallelization
- Disadvantages: Still explicit (i.e. only small time steps allowed especially when used together with cut-cells), complicated derivation of order conditions and stability results

Another ansatz is the use of implicit methods
- Advantages: Allows very big time steps, order conditions and stability issues are easier
- Disadvantages: Requires solution of huge (non-)linear systems of equations, needs efficient (parallel) preconditioners
In compressible models occur:
- Energetically relevant slow waves (e.g. advection, Rossby waves)
- Energetically irrelevant fast waves (e.g. sound waves)

In explicit models the fast waves restrict the maximal time step size.

One ansatz to overcome this is operator splitting
- Advantages: Every step is cheap, easy to implement, parallelization
- Disadvantages: Still explicit (i.e. only small time steps allowed especially when used together with cut-cells), complicated derivation of order conditions and stability results

Another ansatz is the use of implicit methods
- Advantages: Allows very big time steps, order conditions and stability issues are easier
- Disadvantages: Requires solution of huge (non-)linear systems of equations, needs efficient (parallel) preconditioners
In compressible models occur:
- Energetically relevant slow waves (e.g. advection, Rossby waves)
- Energetically irrelevant fast waves (e.g. sound waves)

In explicit models the fast waves restrict the maximal time step size

One ansatz to overcome this is operator splitting
- Advantages: Every step is cheap, easy to implement, parallelization
- Disadvantages: Still explicit (i.e. only small time steps allowed especially when used together with cut-cells), complicated derivation of order conditions and stability results

Another ansatz is the use of implicit methods
- Advantages: Allows very big time steps, order conditions and stability issues are easier
- Disadvantages: Requires solution of huge (non-)linear systems of equations, needs efficient (parallel) preconditioners
- In compressible models occur:
  - Energetically relevant slow waves (e.g. advection, Rossby waves)
  - Energetically irrelevant fast waves (e.g. sound waves)
- In explicit models the fast waves restrict the maximal time step size
- One ansatz to overcome this is operator splitting
  - Advantages: Every step is cheap, easy to implement, parallelization
  - Disadvantages: Still explicit (i.e. only small time steps allowed especially when used together with cut-cells), complicated derivation of order conditions and stability results
- Another ansatz is the use of implicit methods
  - Advantages: Allows very big time steps, order conditions and stability issues are easier
  - Disadvantages: Requires solution of huge (non-)linear systems of equations, needs efficient (parallel) preconditioners
Write numerical solutions as:

\[
Y_m := \begin{pmatrix} Y_{m1} \\ \vdots \\ Y_{ms} \end{pmatrix} \approx \begin{pmatrix} y(t_m + c_1 h) \\ \vdots \\ y(t_m + c_s h) \end{pmatrix} \in \mathbb{R}^{s \times n}, \quad F_m := f(Y_m) \in \mathbb{R}^{s \times n}
\]

Runge-Kutta methods (for autonomous systems) read:

\[
Y_m = Y_{m-1,s} + \Delta t A F_m
\]

Explicit peer methods are defined by:

\[
Y_{mi} = B_i Y_{m-1} + \Delta t A_i F_{m-1} + \Delta t R_i F_m
\]

Performing one Newton step results in the considered class of linearly implicit peer methods:

\[
Y_m (I - h \gamma J)^T = B Y_{m-1} + \Delta t A F_{m-1} + \Delta t R F_m + \Delta t G Y_{m-1} J^T + \Delta t H Y_m J^T
\]
Write numerical solutions as:

\[
Y_m := \begin{pmatrix} Y_{m1} \\ \vdots \\ Y_{ms} \end{pmatrix} \approx \begin{pmatrix} y(t_m + c_1 h) \\ \vdots \\ y(t_m + c_s h) \end{pmatrix} \in \mathbb{R}^{s \times n}, \quad F_m := f(Y_m) \in \mathbb{R}^{s \times n}
\]

Runge-Kutta methods (for autonomous systems) read:

\[
Y_m = Y_{m-1,s} + \Delta t A F_m
\]

Explicit peer methods are defined by:

\[
Y_{mi} = B_i Y_{m-1} + \Delta t A_i F_{m-1} + \Delta t R_i F_m
\]

Performing one Newton step results in the considered class of linearly implicit peer methods:

\[
Y_m(I - h \gamma J)^T = B Y_{m-1} + \Delta t A F_{m-1} + \Delta t R F_m + \Delta t G Y_{m-1} J^T + \Delta t H Y_m J^T
\]
Write numerical solutions as:

\[
Y_m := \begin{pmatrix} Y_{m1} \\ \vdots \\ Y_{ms} \end{pmatrix} \approx \begin{pmatrix} y(t_m + c_1 h) \\ \vdots \\ y(t_m + c_s h) \end{pmatrix} \in \mathbb{R}^{s \times n}, \quad F_m := f(Y_m) \in \mathbb{R}^{s \times n}
\]

Runge-Kutta methods (for autonomous systems) read:

\[
Y_m = Y_{m-1,s} + \Delta t A F_m
\]

Explicit peer methods are defined by:

\[
Y_{mi} = B_i Y_{m-1} + \Delta t A_i F_{m-1} + \Delta t R_i F_m
\]

Performing one Newton step results in the considered class of linearly implicit peer methods:

\[
Y_m(I - h_\gamma J)^T = B Y_{m-1} + \Delta t A F_{m-1} + \Delta t R F_m + \Delta t G Y_{m-1} J^T + \Delta t H Y_m J^T
\]
Write numerical solutions as:

\[ Y_m := \begin{pmatrix} Y_{m1} \\ \vdots \\ Y_{ms} \end{pmatrix} \approx \begin{pmatrix} y(t_m + c_1 h) \\ \vdots \\ y(t_m + c_s h) \end{pmatrix} \in \mathbb{R}^{s \times n}, \quad F_m := f(Y_m) \in \mathbb{R}^{s \times n} \]

Runge-Kutta methods (for autonomous systems) read:

\[ Y_m = Y_{m-1,s} + \Delta t A F_m \]

Implicit peer methods are defined by:

\[ Y_{mi} = B_i Y_{m-1} + \Delta t A_i F_{m-1} + \Delta t R_i F_m + \Delta t \gamma f(Y_{mi}) \]

Performing one Newton step results in the considered class of linearly implicit peer methods:

\[ Y_m (I - h \gamma J)^T = B Y_{m-1} + \Delta t A F_{m-1} + \Delta t R F_m + \Delta t G Y_{m-1} J^T + \Delta t H Y_m J^T \]
Write numerical solutions as:

\[
Y_m := \begin{pmatrix} Y_{m1} \\ \vdots \\ Y_{ms} \end{pmatrix} \approx \begin{pmatrix} y(t_m + c_1 h) \\ \vdots \\ y(t_m + c_s h) \end{pmatrix} \in \mathbb{R}^{s \times n}, \quad F_m := f(Y_m) \in \mathbb{R}^{s \times n}
\]

Runge-Kutta methods (for autonomous systems) read:

\[
Y_m = Y_{m-1,s} + \Delta t A F_m
\]

Implicit peer methods are defined by:

\[
Y_{mi} = B_i Y_{m-1} + \Delta t A_i F_{m-1} + \Delta t R_i F_m + \Delta t \gamma f(Y_{mi})
\]

Performing one Newton step results in the considered class of linearly implicit peer methods:

\[
Y_m(I - h \gamma J)^T = B Y_{m-1} + \Delta t A F_{m-1} + \Delta t R F_m + \Delta t G Y_{m-1} J^T + \Delta t H Y_m J^T
\]
Order conditions for \( p = s \) can be written in compact matrix form

\[
B \mathbf{1} = \mathbf{1},
\]
\[
A = CV_0D^{-1}V_1^{-1} - B(C - I)V_1D^{-1}V_1^{-1} - RV_0V_1^{-1},
\]
\[
G = -\Gamma V_0V_1^{-1} - HV_0V_1^{-1}
\]

with \( \mathbf{1} = (1, \ldots, 1)^T \), \( C = \text{diag}(c_1, \ldots, c_s) \), \( \Gamma = \gamma I \),
\( D = \text{diag}(1, 2, \ldots, s) \),
\[
V_0 = \begin{pmatrix}
1 & c_1 & \cdots & c_1^{s-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & c_s & \cdots & c_s^{s-1}
\end{pmatrix}
\]
and
\[
V_1 = \begin{pmatrix}
1 & c_1 - 1 & \cdots & (c_1 - 1)^{s-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & c_s - 1 & \cdots & (c_s - 1)^{s-1}
\end{pmatrix}.
\]

In the remainder we will concentrate on second-order methods with \( s = 2 \) stages. Furthermore we choose \( c_s = 1 \) so that \( Y_{ms} \approx y(t_{m+1}) \).

Remaining parameters are \( c_1, \gamma, b_{11}, b_{21}, r_{21} \) and \( h_{21} \). These will be optimized with respect to good stability properties.
Order conditions for \( p = s \) can be written in compact matrix form

\[
B \mathbb{1} = \mathbb{1},
A = CV_0D^{-1}V_1^{-1} - B(C - I)V_1D^{-1}V_1^{-1} - RV_0V_1^{-1},
G = -\Gamma V_0V_1^{-1} - HV_0V_1^{-1}
\]

with \( \mathbb{1} = (1, \ldots, 1)^T \), \( C = \text{diag}(c_1, \ldots, c_s) \), \( \Gamma = \gamma I \),
\( D = \text{diag}(1, 2, \ldots, s) \),
\[
V_0 = \begin{pmatrix} 1 & c_1 & \cdots & c_{s-1}^s \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_s & \cdots & c_{s-1}^s \end{pmatrix} \quad \text{and} \quad V_1 = \begin{pmatrix} 1 & c_1 - 1 & \cdots & (c_1 - 1)^{s-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_s - 1 & \cdots & (c_s - 1)^{s-1} \end{pmatrix}.
\]

In the remainder we will concentrate on second-order methods with \( s = 2 \) stages. Furthermore we choose \( c_s = 1 \) so that \( Y_{ms} \approx y(t_{m+1}) \).

Remaining parameters are \( c_1, \gamma, b_{11}, b_{21}, r_{21} \) and \( h_{21} \). These will be optimized with respect to good stability properties.
Order conditions for $p = s$ can be written in compact matrix form

$$B \mathbb{1} = \mathbb{1},$$

$$A = CV_0 D^{-1} V_1^{-1} - B(C - I)V_1 D^{-1} V_1^{-1} - RV_0 V_1^{-1},$$

$$G = -\Gamma V_0 V_1^{-1} - HV_0 V_1^{-1}$$

with $\mathbb{1} = (1, \ldots, 1)^T$, $C = \text{diag}(c_1, \ldots, c_s)$, $\Gamma = \gamma I$, $D = \text{diag}(1, 2, \ldots, s)$,

$$V_0 = \begin{pmatrix} 1 & c_1 & \cdots & c_1^{s-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_s & \cdots & c_s^{s-1} \end{pmatrix} \quad \text{and} \quad V_1 = \begin{pmatrix} 1 & c_1 - 1 & \cdots & (c_1 - 1)^{s-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_s - 1 & \cdots & (c_s - 1)^{s-1} \end{pmatrix}.$$ 

In the remainder we will concentrate on second-order methods with $s = 2$ stages. Furthermore we choose $c_s = 1$ so that $Y_{ms} \approx y(t_{m+1})$.

Remaining parameters are $c_1$, $\gamma$, $b_{11}$, $b_{21}$, $r_{21}$ and $h_{21}$. These will be optimized with respect to good stability properties.
1. Peer methods
   - Motivation
   - Formulation of the methods

2. Linear stability theory
   - Linearization of Euler equations
   - Amplitude and phase properties

3. Numerical tests
   - The 2D compressible Euler equations
   - Rising bubble
   - Flow over mountain
   - Zeppelin test

4. Conclusions
One-dimensional compressible Euler equations in conservative form:

\[
\begin{align*}
\dot{\rho} &= -\frac{\partial \rho u}{\partial x}, \\
\dot{\rho}u &= -\frac{\partial \rho u u}{\partial x} - \frac{\partial p}{\partial x}, \\
\dot{\rho}\theta &= -\frac{\partial \rho u \theta}{\partial x}, \\
p &= \left(\frac{R \rho \theta}{p_0^\kappa}\right)^{\frac{1}{1-\kappa}}
\end{align*}
\]

After elimination of pressure and linearization by considering the disturbed quantities (e.g. \(\rho' := \rho - \bar{\rho}\)) and dropping all nonlinear terms the linearized Euler equations read in compact matrix form:

\[
\begin{pmatrix}
\dot{\rho}' \\
\dot{(\rho u)'} \\
\dot{\frac{1}{\theta} (\rho \theta)'}
\end{pmatrix} =
- \begin{pmatrix}
0 & 1 & 0 \\
-\bar{u}^2 & 2\bar{u} & c_s^2 \\
-\bar{u} & 1 & \bar{u}
\end{pmatrix}
\begin{pmatrix}
\rho_x' \\
(\rho u)_x' \\
\frac{1}{\theta} (\rho \theta)_x'
\end{pmatrix}
\]
• One-dimensional compressible Euler equations in conservative form:

\[
\begin{align*}
\dot{\rho} &= -\frac{\partial \rho u}{\partial x} \\
\dot{\rho}u &= -\frac{\partial \rho uu}{\partial x} - \frac{\partial p}{\partial x} \\
\dot{\rho}\theta &= -\frac{\partial \rho u\theta}{\partial x} \\
p &= \left(\frac{R \rho\theta}{p_0^\kappa}\right)^{\frac{1}{1-\kappa}}
\end{align*}
\]

• After elimination of pressure and linearization by considering the disturbed quantities (e.g. \(\rho' := \rho - \bar{\rho}\)) and dropping all nonlinear terms the linearized Euler equations read in compact matrix form:

\[
\begin{pmatrix}
\dot{\rho}' \\
(\rho u)' \\
\frac{1}{\theta} (\rho \theta)'
\end{pmatrix} = - \begin{pmatrix}
0 & 1 & 0 \\
-\overline{u}^2 & 2\overline{u} & c_s^2 \\
-\overline{u} & 1 & \overline{u}
\end{pmatrix} \begin{pmatrix}
\rho'_x \\
(\rho u)'_x \\
\frac{1}{\theta} (\rho \theta)'_x
\end{pmatrix}
\]
Variables are defined on a staggered grid

For investigation of spatial discretizations perform von Neumann stability analysis, e.g. it holds:

\[ \rho u(t, x_{j+1/2}) = \rho u(t) e^{ikx_{j+1/2}} \]
\[ \Rightarrow \frac{\partial \rho u}{\partial x} \bigg|_{(t,x_j)} = \rho u(t) \frac{e^{ikx_j}}{\Delta x} \left( e^{\frac{ik\Delta x}{2}} - e^{-\frac{ik\Delta x}{2}} \right) \]

Three spatial discretizations appear:

\[ D_1 = \frac{1}{\Delta x} \left( 1 - e^{-ik\Delta x} \right) \]
\[ D_2 = \frac{1}{\Delta x} \left( e^{\frac{ik\Delta x}{2}} - e^{-\frac{ik\Delta x}{2}} \right) \]
\[ D_3 = \frac{1}{6\Delta x} \left( 2e^{ik\Delta x} + 3 - 6e^{-ik\Delta x} + e^{-2ik\Delta x} \right) \]
• Variables are defined on a staggered grid

\[ \begin{array}{cccccccc}
\cdots & j-1 & j-1/2 & j & j+1/2 & j+1 & \cdots \\
\rho u & \rho \theta & \rho u & \rho \theta & \rho u & \rho \theta & \rho u
\end{array} \]

• For investigation of spatial discretizations perform von Neumann stability analysis, e.g. it holds:

\[ \rho u(t, x_{j+1/2}) = \rho u(t)e^{ikx_{j+1/2}} \]

\[ \Rightarrow \quad \left. \frac{\partial \rho u}{\partial x} \right|_{(t, x_j)} = \rho u(t) \frac{e^{ikx_j}}{\Delta x} \left( e^{\frac{ik\Delta x}{2}} - e^{-\frac{ik\Delta x}{2}} \right) \]

• Three spatial discretizations appear:

\[ D_1 = \frac{1}{\Delta x} (1 - e^{-ik\Delta x}) \]

\[ D_2 = \frac{1}{\Delta x} \left( e^{\frac{ik\Delta x}{2}} - e^{-\frac{ik\Delta x}{2}} \right) \]

\[ D_3 = \frac{1}{6\Delta x} \left( 2e^{ik\Delta x} + 3 - 6e^{-ik\Delta x} + e^{-2ik\Delta x} \right) \]
• Variables are defined on a staggered grid

\[ \begin{array}{ccccccc}
  j-1 & j-1/2 & j & j+1/2 & j+1 \\
  \rho u & \rho \theta & \rho u & \rho \theta & \rho u & \rho \theta & \rho u \\
\end{array} \]

• For investigation of spatial discretizations perform von Neumann stability analysis, e.g. it holds:

\[ \rho u(t, x_{j+1/2}) = \rho u(t) e^{i k x_{j+1/2}} \]

\[ \Rightarrow \frac{\partial \rho u}{\partial x} \bigg|_{(t,x_j)} = \rho u(t) \frac{e^{i k x_j}}{\Delta x} \left( e^{\frac{i k \Delta x}{2}} - e^{-\frac{i k \Delta x}{2}} \right) \]

• Three spatial discretizations appear:

\[ D_1 = \frac{1}{\Delta x} \left( 1 - e^{-i k \Delta x} \right) \]

\[ D_2 = \frac{1}{\Delta x} \left( e^{\frac{i k \Delta x}{2}} - e^{-\frac{i k \Delta x}{2}} \right) \]

\[ D_3 = \frac{1}{6 \Delta x} \left( 2 e^{i k \Delta x} + 3 - 6 e^{-i k \Delta x} + e^{-2 i k \Delta x} \right) \]
Using these operators results in the ODE:

\[
\begin{pmatrix}
\dot{\rho}' \\
(\rho u)' \\
\frac{1}{\theta} (\rho \dot{\theta})'
\end{pmatrix} = - \begin{pmatrix}
0 & D_2 & 0 \\
-u^2 D_3 & 2u D_3 & c_s^2 D_2 \\
-u D_3 & D_2 & u D_3
\end{pmatrix} \begin{pmatrix}
\rho' \\
(\rho u)' \\
\frac{1}{\theta} (\rho \dot{\theta})'
\end{pmatrix}
\]

To save storage and gain computational efficiency we make two simplifications for the Jacobian \( J \):

- Use Jacobian of the advection form of the Euler equations
- Use first-order upwind scheme for spatial discretization

So for the Jacobian we use the matrix which belongs to:

\[
\begin{pmatrix}
\dot{\rho}' \\
\frac{\partial \rho'}{\partial \rho u} \\
\frac{1}{\theta} (\rho \dot{\theta})'
\end{pmatrix} = - \begin{pmatrix}
u D_1 & D_2 & 0 \\
0 & u D_1 & c_s^2 D_2 \\
0 & D_2 & u D_1
\end{pmatrix} \begin{pmatrix}
\rho' \\
\frac{\partial \rho'}{\partial \rho u} \\
\frac{1}{\theta} (\rho \dot{\theta})'
\end{pmatrix}
\]
Using these operators results in the ODE:

$$
\begin{pmatrix}
\dot{\rho}' \\
(\rho u)' \\
\frac{1}{\theta}(\rho \theta)'
\end{pmatrix} = -
\begin{pmatrix}
0 & \mathcal{D}_2 & 0 \\
-\bar{u}^2 \mathcal{D}_3 & 2\bar{u} \mathcal{D}_3 & c_s^2 \mathcal{D}_2 \\
-\bar{u} \mathcal{D}_3 & \mathcal{D}_2 & \bar{u} \mathcal{D}_3
\end{pmatrix}
\begin{pmatrix}
\rho' \\
(\rho u)' \\
\frac{1}{\theta}(\rho \theta)'
\end{pmatrix}
$$

To save storage and gain computational efficiency we make two simplifications for the Jacobian $J$:

- Use Jacobian of the advection form of the Euler equations
- Use first-order upwind scheme for spatial discretization

So for the Jacobian we use the matrix which belongs to:

$$
\begin{pmatrix}
\dot{\rho}' \\
\dot{\rho} u' \\
\frac{1}{\theta}(\rho \theta)'
\end{pmatrix} = -
\begin{pmatrix}
\bar{u} \mathcal{D}_1 & \mathcal{D}_2 & 0 \\
0 & \bar{u} \mathcal{D}_1 & c_s^2 \mathcal{D}_2 \\
0 & \mathcal{D}_2 & \bar{u} \mathcal{D}_1
\end{pmatrix}
\begin{pmatrix}
\rho' \\
\rho u' \\
\frac{1}{\theta}(\rho \theta)'
\end{pmatrix}
$$
Using these operators results in the ODE:

\[
\begin{pmatrix}
\rho' \\
\rho u' \\
\frac{1}{\theta} (\rho \theta)'
\end{pmatrix} = -\begin{pmatrix}
0 & \mathcal{D}_2 & 0 \\
-u^2 \mathcal{D}_3 & 2u \mathcal{D}_3 & c_s^2 \mathcal{D}_2 \\
-u \mathcal{D}_3 & \mathcal{D}_2 & \frac{1}{u} \mathcal{D}_3
\end{pmatrix}
\begin{pmatrix}
\rho' \\
\rho u' \\
\frac{1}{\theta} (\rho \theta)'
\end{pmatrix}
\]

To save storage and gain computational efficiency we make two simplifications for the Jacobian \( J \):

- Use Jacobian of the advection form of the Euler equations
- Use first-order upwind scheme for spatial discretization

So for the Jacobian we use the matrix which belongs to:

\[
\begin{pmatrix}
\rho' \\
\rho u' \\
\frac{1}{\theta} (\rho \theta)'
\end{pmatrix} = -\begin{pmatrix}
\bar{u} \mathcal{D}_1 & \mathcal{D}_2 & 0 \\
0 & \bar{u} \mathcal{D}_1 & c_s^2 \mathcal{D}_2 \\
0 & \mathcal{D}_2 & \frac{1}{\bar{u}} \mathcal{D}_1
\end{pmatrix}
\begin{pmatrix}
\rho' \\
\rho u' \\
\frac{1}{\theta} (\rho \theta)'
\end{pmatrix}
\]
Using these operators results in the ODE:

\[
\begin{pmatrix}
\frac{\dot{\rho}}{\rho} \\
\frac{\dot{\rho}u}{(\rho u)'} \\
\frac{1}{\theta}(\rho \dot{\theta})'
\end{pmatrix}
= -
\begin{pmatrix}
0 & D_2 & 0 \\
-u^2 D_3 & 2 u D_3 & c_s^2 D_2 \\
-u D_3 & D_2 & u D_3
\end{pmatrix}
\begin{pmatrix}
\frac{\rho}{(\rho u)'} \\
\frac{1}{\theta}(\rho \dot{\theta})'
\end{pmatrix}
\]

To save storage and gain computational efficiency we make two simplifications for the Jacobian \( J \):

- Use Jacobian of the advection form of the Euler equations
- Use first-order upwind scheme for spatial discretization

So for the Jacobian we use the matrix which belongs to:

\[
\begin{pmatrix}
\frac{\dot{\rho}}{\rho} \\
\frac{\dot{\rho}u}{\rho u'} \\
\frac{1}{\theta}(\rho \dot{\theta})'
\end{pmatrix}
= -
\begin{pmatrix}
\bar{u} D_1 & D_2 & 0 \\
0 & \bar{u} D_1 & c_s^2 D_2 \\
0 & D_2 & \bar{u} D_1
\end{pmatrix}
\begin{pmatrix}
\frac{\rho}{\rho u'} \\
\frac{1}{\theta}(\rho \dot{\theta})'
\end{pmatrix}
\]
Using these operators results in the ODE:

\[
\begin{pmatrix}
\dot{\rho}' \\
\dot{(\rho u)'} \\
\frac{1}{\theta} (\rho \theta)'
\end{pmatrix}
= -
\begin{pmatrix}
0 & \mathcal{D}_2 & 0 \\
-\bar{u}^2 \mathcal{D}_3 & 2\bar{u}\mathcal{D}_3 & c_s^2 \mathcal{D}_2 \\
-\bar{u} \mathcal{D}_3 & \mathcal{D}_2 & \bar{u} \mathcal{D}_3
\end{pmatrix}
\begin{pmatrix}
\rho' \\
(\rho u)' \\
\frac{1}{\theta} (\rho \theta)'
\end{pmatrix}
\]

To save storage and gain computational efficiency we make two simplifications for the Jacobian \(J\):
- Use Jacobian of the advection form of the Euler equations
- Use first-order upwind scheme for spatial discretization

So for the Jacobian we use the matrix which belongs to:

\[
\begin{pmatrix}
\dot{\rho}' \\
\dot{\bar{\rho} u}' \\
\frac{1}{\theta} (\rho \theta)'
\end{pmatrix}
= -
\begin{pmatrix}
\bar{u} \mathcal{D}_1 & \mathcal{D}_2 & 0 \\
0 & \bar{u} \mathcal{D}_1 & c_s^2 \mathcal{D}_2 \\
0 & \mathcal{D}_2 & \bar{u} \mathcal{D}_1
\end{pmatrix}
\begin{pmatrix}
\rho' \\
\bar{\rho} u' \\
\frac{1}{\theta} (\rho \theta)'
\end{pmatrix}
\]
Eigenvalues of correct and simplified Jacobian

![Eigenvalues of correct Jacobian](image1)

![Eigenvalues of simplified Jacobian](image2)
Stability regions for exact and simplified Jacobian
Amplitude and phase

Amplitude

Relative Phase

- Implicit Peer
- Rosenbrock
- Explicit Peer
1 Peer methods
   ● Motivation
   ● Formulation of the methods

2 Linear stability theory
   ● Linearization of Euler equations
   ● Amplitude and phase properties

3 Numerical tests
   ● The 2D compressible Euler equations
   ● Rising bubble
   ● Flow over mountain
   ● Zeppelin test

4 Conclusions
\[
\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u}{\partial x} - \frac{\partial \rho w}{\partial z}
\]
\[
\frac{\partial \rho u}{\partial t} = -\frac{\partial \rho uu}{\partial x} - \frac{\partial \rho uw}{\partial z} - \frac{R}{1 - \kappa} \pi \frac{\partial \rho \theta}{\partial x}
\]
\[
\frac{\partial \rho w}{\partial t} = -\frac{\partial \rho uw}{\partial x} - \frac{\partial \rho ww}{\partial z} - \frac{R}{1 - \kappa} \pi \frac{\partial \rho \theta}{\partial z} - \rho g
\]
\[
\frac{\partial \rho \theta}{\partial t} = -\frac{\partial \rho u \theta}{\partial x} - \frac{\partial \rho w \theta}{\partial z}
\]
\[
\pi = \left( \frac{R \rho \theta}{p_0} \right)^{\frac{\kappa}{1 - \kappa}}
\]
\[
\frac{\partial \rho}{\partial t} = - \frac{\partial \rho u}{\partial x} - \frac{\partial \rho w}{\partial z}
\]
\[
\frac{\partial \rho u}{\partial t} = - \frac{\partial \rho uu}{\partial x} - \frac{\partial \rho uw}{\partial z} - \frac{R}{1 - \kappa} \pi \frac{\partial \rho \theta}{\partial x}
\]
\[
\frac{\partial \rho w}{\partial t} = - \frac{\partial \rho uw}{\partial x} - \frac{\partial \rho ww}{\partial z} - \frac{R}{1 - \kappa} \pi \frac{\partial \rho \theta}{\partial z} - \rho g
\]
\[
\frac{\partial \rho \theta}{\partial t} = - \frac{\partial \rho u \theta}{\partial x} - \frac{\partial \rho w \theta}{\partial z}
\]
\[
\pi = \left( \frac{R \rho \theta}{p_0} \right)^{\frac{1}{1 - \kappa}}
\]

<table>
<thead>
<tr>
<th></th>
<th>correct Jacobian</th>
<th>simplified Jacobian</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D</td>
<td>$3D_2 + 4D_3 = 22$</td>
<td>$3D_2 + 3D_1 = 12$</td>
<td>55%</td>
</tr>
<tr>
<td>2D</td>
<td>$6D_2 + 14D_3 = 68$</td>
<td>$6D_2 + 8D_1 = 28$</td>
<td>41%</td>
</tr>
<tr>
<td>3D</td>
<td>$9D_2 + 30D_3 = 138$</td>
<td>$9D_2 + 15D_1 = 48$</td>
<td>35%</td>
</tr>
</tbody>
</table>
Rising bubble
<table>
<thead>
<tr>
<th>Peer methods</th>
<th>Linear stability theory</th>
<th>Numerical tests</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Flow over mountain

Witch of Agnesi mountain

Vertical Velocity

Vertical Velocity
<table>
<thead>
<tr>
<th>Peer methods</th>
<th>Linear stability theory</th>
<th>Numerical tests</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Zeppelin test
1 Peer methods
   • Motivation
   • Formulation of the methods

2 Linear stability theory
   • Linearization of Euler equations
   • Amplitude and phase properties

3 Numerical tests
   • The 2D compressible Euler equations
   • Rising bubble
   • Flow over mountain
   • Zeppelin test

4 Conclusions
• Development of a linearly implicit two-stage peer method which
  • is second-order independently of the Jacobian
  • is A-stable in the common sense and for the simplified Jacobian
  • has acceptable amplitude and phase errors

• Despite of the large CFL numbers the solutions of the linearly implicit peer method are as good as the solutions computed with the explicit method with tiny time steps

• Only exception is the transported rising bubble where the impact of damping and phase errors is visible, but
  • the explicit method is a three-stage method, there is no explicit two-stage method which is stable with the time steps used in the first test
  • the implicit peer method might not be the best one, perhaps there are better optimization criteria
Development of a linearly implicit two-stage peer method which
- is second-order independently of the Jacobian
- is A-stable in the common sense and for the simplified Jacobian
- has acceptable amplitude and phase errors

Despite of the large CFL numbers the solutions of the linearly implicit peer method are as good as the solutions computed with the explicit method with tiny time steps

Only exception is the transported rising bubble where the impact of damping and phase errors is visible, but
- the explicit method is a three-stage method, there is no explicit two-stage method which is stable with the time steps used in the first test
- the implicit peer method might not be the best one, perhaps there are better optimization criteria
Development of a linearly implicit two-stage peer method which
  - is second-order independently of the Jacobian
  - is A-stable in the common sense and for the simplified Jacobian
    - has acceptable amplitude and phase errors

Despite of the large CFL numbers the solutions of the linearly implicit peer method are as good as the solutions computed with the explicit method with tiny time steps

Only exception is the transported rising bubble where the impact of damping and phase errors is visible, but
  - the explicit method is a three-stage method, there is no explicit two-stage method which is stable with the time steps used in the first test
  - the implicit peer method might not be the best one, perhaps there are better optimization criteria
● Development of a linearly implicit two-stage peer method which
  ● is second-order independently of the Jacobian
  ● is A-stable in the common sense and for the simplified Jacobian
  ● has acceptable amplitude and phase errors

● Despite of the large CFL numbers the solutions of the linearly implicit peer method are as good as the solutions computed with the explicit method with tiny time steps

● Only exception is the transported rising bubble where the impact of damping and phase errors is visible, but
  ● the explicit method is a three-stage method, there is no explicit two-stage method which is stable with the time steps used in the first test
  ● the implicit peer method might not be the best one, perhaps there are better optimization criteria
Development of a linearly implicit two-stage peer method which
   - is second-order independently of the Jacobian
   - is A-stable in the common sense and for the simplified Jacobian
   - has acceptable amplitude and phase errors

Despite of the large CFL numbers the solutions of the linearly implicit peer method are as good as the solutions computed with the explicit method with tiny time steps

Only exception is the transported rising bubble where the impact of damping and phase errors is visible, but
   - the explicit method is a three-stage method, there is no explicit two-stage method which is stable with the time steps used in the first test
   - the implicit peer method might not be the best one, perhaps there are better optimization criteria
• Development of a linearly implicit two-stage peer method which
  • is second-order independently of the Jacobian
  • is A-stable in the common sense and for the simplified Jacobian
  • has acceptable amplitude and phase errors

• Despite of the large CFL numbers the solutions of the linearly implicit peer method are as good as the solutions computed with the explicit method with tiny time steps

• Only exception is the transported rising bubble where the impact of damping and phase errors is visible, but
  • the explicit method is a three-stage method, there is no explicit two-stage method which is stable with the time steps used in the first test
  • the implicit peer method might not be the best one, perhaps there are better optimization criteria
Development of a linearly implicit two-stage peer method which
  - is second-order independently of the Jacobian
  - is A-stable in the common sense and for the simplified Jacobian
  - has acceptable amplitude and phase errors

Despite of the large CFL numbers the solutions of the linearly implicit
peer method are as good as the solutions computed with the explicit
method with tiny time steps

Only exception is the transported rising bubble where the impact of
damping and phase errors is visible, but
  - the explicit method is a three-stage method, there is no explicit
two-stage method which is stable with the time steps used in the first
test
  - the implicit peer method might not be the best one, perhaps there are
    better optimization criteria
Development of a linearly implicit two-stage peer method which
- is second-order independently of the Jacobian
- is A-stable in the common sense and for the simplified Jacobian
- has acceptable amplitude and phase errors

Despite of the large CFL numbers the solutions of the linearly implicit peer method are as good as the solutions computed with the explicit method with tiny time steps

Only exception is the transported rising bubble where the impact of damping and phase errors is visible, but
- the explicit method is a three-stage method, there is no explicit two-stage method which is stable with the time steps used in the first test
- the implicit peer method might not be the best one, perhaps there are better optimization criteria
Danke für Ihre Aufmerksamkeit!
Danke für Ihre Aufmerksamkeit!

Stefan Jebens, Oswald Knoth, Rüdiger Weiner:
*Linearly Implicit Peer Methods for the Compressible Euler Equations*,
to appear in *Applied Numerical Mathematics*