Using Projective Integration for Accelerated Computation of Stiff ODEs

Julian Koellermeier

University of Science and Technology Beijing

December 11th, 2018
1 Introduction
- Stiff problems
- Runge-Kutta Schemes

2 Projective Integration for Moment Equations
- Definition
- Stability
- Telescopic PI

3 Summary
Consider the following ODE

\[
\begin{align*}
\frac{\partial}{\partial t} y_1 &= -80.6y_1 + 119.4y_2 \\
\frac{\partial}{\partial t} y_2 &= 79.6y_1 - 120.4y_2
\end{align*}
\]

or

\[
\frac{\partial}{\partial t} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} -80.6 & 119.4 \\ 79.6 & -120.4 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}
\]

with exact solution

\[
y(t) = c_1 \begin{pmatrix} 3 \\ 2 \end{pmatrix} e^{-t} + c_2 \begin{pmatrix} -1 \\ 1 \end{pmatrix} e^{-200t}.
\]
\begin{align*}
  y(t) &= c_1 \begin{pmatrix} 3 \\ 2 \end{pmatrix} e^{-t} + c_2 \begin{pmatrix} -1 \\ 1 \end{pmatrix} e^{-200t}.
\end{align*}

Two time scales. Fast relaxation of $y_1, y_2$ in initial phase.
Explicit Euler method for stiff ODEs

model ODE

$$\partial_t y = f(y) = -\lambda y.$$  

explicit Euler scheme

$$y^{n+1} = y^{n} + \Delta t \cdot f(y)$$

$$= y^{n} - \Delta t \cdot \lambda y^{n}$$

$$= (1 - \Delta t \cdot \lambda)y^{n}$$
Explicit Euler method for stiff ODEs

\[ \partial_t y = f(y) = -\lambda y. \]

explicit Euler scheme

\[
\begin{align*}
    y^{n+1} &= y^n + \Delta t \cdot f(y) \\
    &= y^n - \Delta t \cdot \lambda y^n \\
    &= (1 - \Delta t \cdot \lambda)y^n \\
\Rightarrow \quad y^{n+1} &= (1 - \Delta t \cdot \lambda)^{n+1}y^0
\end{align*}
\]
Explicit Euler method for stiff ODEs

model ODE

\[ \partial_t y = f(y) = -\lambda y. \]

explicit Euler scheme

\[
\begin{align*}
y^{n+1} &= y^n + \Delta t \cdot f(y) \\
&= y^n - \Delta t \cdot \lambda y^n \\
&= (1 - \Delta t \cdot \lambda)y^n
\end{align*}
\]

\[ \Rightarrow y^{n+1} = (1 - \Delta t \cdot \lambda)^{n+1} y^0 \]

Explicit Euler method is only stable for \( \Delta t \lambda < 2. \)
Consider the following hyperbolic relaxation PDE for $0 < \epsilon \ll 1$

\[
\partial_t u + \partial_x F(u) = -\frac{1}{\epsilon} S(u), \quad x \in \Omega, \ t \in \mathbb{R}^+
\]

\[
\Rightarrow \quad \partial_t u = -\frac{1}{\epsilon} S(u) - \partial_x F(u)
\]

Spatial discretization $x_1, x_2, \ldots, x_N$, for $N \in \mathbb{N}$ using $u = (u_1, u_2, \ldots, u_N)$

\[
\partial_t u = D_t (u)
\]
Consider the following hyperbolic relaxation PDE for $0 < \epsilon \ll 1$

$$\partial_t u + \partial_x F(u) = -\frac{1}{\epsilon} S(u), \quad x \in \Omega, \ t \in \mathbb{R}^+$$

$$\Rightarrow \quad \partial_t u = -\frac{1}{\epsilon} S(u) - \partial_x F(u)$$

Spatial discretization $x_1, x_2, \ldots, x_N$, for $N \in \mathbb{N}$ using $u = (u_1, u_2, \ldots, u_N)$

$$\partial_t u = D_t (u)$$

Topic of this talk: time integration method for stiff right-hand side
Definition of stiff problem

Define the stiffness ratio using eigenvalues of linearized problem

\[
\frac{\max |Re(\lambda_i)|}{\min |Re(\lambda_i)|}
\]

Large stiffness ratio can lead to stability problems.

Stiffness property

- large stiffness ratio
- some components of the solution decay much faster than others
- time step size \(\Delta t\) limited by stability instead of accuracy
Examples for stiff ODEs

- Chemical reactions with different reaction speeds
  
  \[ A + X \xrightarrow{k_1} 2X \]
  \[ X + Y \xrightarrow{k_2} 2Y \]
  \[ Y \xrightarrow{k_3} B \]

- Combustion problems
- Fluid dynamics including sonic waves versus bulk movement of flow
- Kinetic equation
  
  \[ \partial_t u + A_M \partial_x u = -\frac{1}{\epsilon} S(u) \]
Explicit vs Implicit schemes

\[ u^{n+1} = u^n + \Delta t \cdot D_t(u^n) \]

Explicit schemes, e.g. Forward Euler (FE)

+ explicit update formula
+ straightforward implementation
- restrictive time step constraint (no A-stability)

\[ u^{n+1} = u^n + \Delta t \cdot D_t(u^{n+1}) \]

Implicit schemes, e.g. Implicit Euler (IE)

- require solution of (non-)linear system \( \Rightarrow \) slow
- more difficult to implement
+ no time step constraint (A-stability)
Example: Runge-Kutta Scheme

\[ \partial_t u = D_t (u) \]

Compute solution using

\[ u^{n+1} = u^n + \Delta t \sum_{j=1}^{s} b_j k_j \]

Using right-hand side evaluations

\[ k_j = D_t \left( u^n + \Delta t \sum_{l=1}^{s} a_{jl} k_l \right), \quad j = 1, \ldots, s. \]

with \( s \) the number of steps, \( \Delta t \) the time step size, and \( a_{jl}, b_j \) coefficients for evaluations/update at times \( c_j \).
Butcher tableau

\[ u^{n+1} = u^n + \Delta t \sum_{j=1}^{s} b_j k_j, \quad k_j = D_t \left( u^n + \Delta t \sum_{l=1}^{s} a_{jl} k_l \right), \quad j = 1, \ldots, s \]

\[
\begin{array}{c|cccc}
   \mathbf{c} & a_{11} & a_{12} & \ldots & a_{1s} \\
   c_2 & a_{21} & a_{22} & \ldots & a_{2s} \\
   \vdots & \vdots & \vdots & \ddots & \vdots \\
   c_s & a_{s1} & a_{s2} & \ldots & a_{ss} \\
   \mathbf{b}^T & b_1 & b_2 & \ldots & b_s \\
\end{array}
\]

- \( c_i \) evaluating times
- \( a_{jl} \) coefficients for computation of evaluations \( k_j \)
- \( b_j \) coefficients for computation of next time step \( u^{n+1} \)

If \( A \) is lower diagonal matrix \( \Rightarrow \) explicit scheme
Diagonally Implicit Runge Kutta (DIRK) method

\[
\begin{array}{c|cccccc}
  c_1 & a_{11} & 0 & \ldots & \ldots & 0 \\
  c_2 & a_{21} & a_{22} & 0 & \ldots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\
  \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
  c_s & a_{s1} & a_{s2} & \ldots & \ldots & a_{ss} \\
  b_1 & b_2 & \ldots & \ldots & b_s \\
\end{array}
\]

Upper diagonal matrix is zero.
Only diagonal entries are implicit.
Implicit-explicit RK methods, [Pareschi, Russo, 2005]

Idea: Split ODE into stiff and non-stiff parts

\[ \partial_t u = -\partial_x F(u) - \frac{1}{\epsilon} S(u) \]
\[ \partial_t u = D_{t}^{EX}(u) + D_{t}^{IM}(u) \]

\[ u^{n+1} = u^n + \Delta t \sum_{j=1}^{s} \tilde{b}_j D_{t}^{EX}(u_j) + \Delta t \sum_{j=1}^{s} b_j D_{t}^{IM}(u_j) \]

Using intermediate values

\[ u_l = u^n + \Delta t \sum_{j=1}^{l-1} \tilde{a}_{lj} D_{t}^{EX}(u_j) + \Delta t \sum_{j=1}^{s} a_{lj} D_{t}^{IM}(u_j) \]
Implicit-explicit (IMEX) methods

\[
\begin{align*}
    u^{n+1} &= u^n + \Delta t \sum_{j=1}^{s} \tilde{b}_j D_{t}^{EX} (u_j) + \Delta t \sum_{j=1}^{s} b_j D_{t}^{IM} (u_j) \\
    u_l &= u^n + \Delta t \sum_{j=1}^{l-1} \tilde{a}_{lj} D_{t}^{EX} (u_j) + \Delta t \sum_{j=1}^{s} a_{lj} D_{t}^{IM} (u_j)
\end{align*}
\]

Use two Butcher tableaus

\[
\begin{array}{c|c}
    \tilde{c} & \tilde{A} \\
    \tilde{b}^T
\end{array}
\quad \text{and} \quad
\begin{array}{c|c}
    c & A \\
    b^T
\end{array}
\]

Implicit scheme is usually a DIRK scheme.
Asymptotic Preserving (AP) property [JIN, 1999]

\[ \partial_t u + A_M \partial_x u = -\frac{1}{\epsilon} S(u) \]  

(1)

System depends on parameter \( \epsilon \), but for \( \epsilon \to 0 \) the system converges to a limit equation (e.g. Euler equation).

**Definition: Asymptotic-Preserving (AP)**

A scheme for problem (1) with discretization parameter \( \Delta t \) is called **Asymptotic Preserving** (AP) if its stability requirement on \( \Delta t \) is independent of \( \epsilon \) and its limit \( \epsilon \to 0 \) is consistent with the limit solution.

Usually, small \( \epsilon \) makes equation stiffer  
\[ \Rightarrow \text{smaller } \Delta t = \Delta t(\epsilon) \text{ is needed for explicit schemes } \Rightarrow \text{ not AP.} \]
Projective Integration for Moment Equations

joint work with
Giovanni Samaey, KU Leuven
Motivation for Projective Integration

Problem:
1. System contains fast modes that we need to solve for
2. We are usually interested in the slow modes, i.e. long-term behavior

Approach:
1. Take care of the fast modes using some simple method
2. Consider slow modes after fast modes are damped
**Projective Integration (PI), [Kevrekidis, 2003]**

**Idea:**
1. Perform $K + 1$ small time steps $\delta t$ first to damp fast modes.
2. Then extrapolate derivative with one large step $\Delta t$.

0. initialize: $u^{n,0} = u^n$

1. for $k = 0, 1, \ldots, K$ : $u^{n,k+1} = u^{n,k} + \delta t \, D_t (u^{n,k})$

2. extrapolate: $u^{n+1} = u^{n,K+1} + (\Delta t - (K + 1)\delta t) \cdot \frac{u^{n,K+1} - u^{n,K}}{\delta t}$
Projective Integration as Runge-Kutta method

Standard definition:

\[
\begin{align*}
    u^{n,1} &= u^{n,0} + \delta t D_t (u^{n,0}) \\
    u^{n,2} &= u^{n,1} + \delta t D_t (u^{n,1}) \\
    &\vdots \\
    u^{n,K+1} &= u^{n,K} + \delta t D_t (u^{n,K}) \\
    u^{n+1} &= u^{n,K+1} + (\Delta t - (K + 1)\delta t) \cdot \frac{u^{n,K+1} - u^{n,K}}{\delta t}
\end{align*}
\]

Note

\[
\frac{u^{n,j+1} - u^{n,j}}{\delta t} = D_t (u^{n,j}) = k_j
\]

Write this as Runge-Kutta scheme with evaluations \( k_j \)
Projective Integration as Runge-Kutta method

\[ k_0 = D_t (u^{n,0}) \]
\[ k_1 = D_t (u^{n,1}) = D_t (u^{n,0} + \delta t k_0) \]
\[ k_2 = D_t (u^{n,2}) = D_t (u^{n,0} + \delta t k_0 + \delta t k_1) \]
\[ \vdots \]
\[ k_K = D_t (u^{n,K}) = D_t \left( u^{n,0} + \delta t \sum_{j=0}^{K-1} k_j \right) \]
\[ u^{n+1} = u^{n,0} + \Delta t \left( \sum_{j=0}^{K-1} \frac{\delta t k_j}{\Delta t} + \frac{\Delta t - K \delta t}{\Delta t} k_K \right) \]
Projective Integration as Runge-Kutta method

\[ k_l = D_t (u^{n,l}) = D_t \left( u^{n,0} + \Delta t \sum_{j=0}^{l-1} \frac{\delta t}{\Delta t} k_j \right) \]

\[ u^{n+1} = u^{n,0} + \Delta t \left( \sum_{j=0}^{K-1} \frac{\delta t}{\Delta t} k_j + \frac{\Delta t - K \delta t}{\Delta t} k_K \right) \]

Write Projective Integration in Butcher tableau

\[
\begin{bmatrix}
0 \\
1 \cdot \frac{\delta t}{\Delta t} \\
\vdots \\
K \cdot \frac{\delta t}{\Delta t}
\end{bmatrix} 
\begin{bmatrix}
0 \\
\frac{\delta t}{\Delta t} \\
\vdots \\
\frac{\delta t}{\Delta t}
\end{bmatrix} 
= 
\begin{bmatrix}
0 \\
\frac{\delta t}{\Delta t} \\
\vdots \\
\frac{\delta t}{\Delta t}
\end{bmatrix} 
\begin{bmatrix}
\frac{\delta t}{\Delta t} \\
\vdots \\
\frac{\delta t}{\Delta t} \\
1 - K \frac{\delta t}{\Delta t}
\end{bmatrix}
\]

Projective integration does \( K + 1 \) small FE steps and one extrapolation
Projective RK scheme (PRK) [LAFITTE et al., 2017]

Use standard RK scheme

\[
\begin{align*}
    \mathbf{c} & \left| A \right| \mathbf{b}^T \\
    \mathbf{u}^{n+1} & = \mathbf{u}^n + \Delta t \sum_{j=1}^{s} b_j k_j
\end{align*}
\]

\[
    k_j = D_t \left( \mathbf{u}^n + \Delta t \sum_{l=1}^{s} a_{jl} k_l \right), \quad j = 1, \ldots, s
\]

Replace each time derivative \( k_l \) by an inner integrator and a time derivative estimate

\[
    \mathbf{u}_{l}^{n,k+1} = \mathbf{u}_{l}^{n,K} + \delta t D_t \left( \mathbf{u}_{l}^{n,K} \right), \quad 0 \leq k \leq K
\]

\[
    k_l = \frac{\mathbf{u}_{l}^{n,K+1} - \mathbf{u}_{l}^{n,K}}{\delta t} = D_t \left( \mathbf{u}_{l}^{n,K} \right)
\]
Dahlquist test equation

\[ \partial_t u = \lambda u, \quad \lambda < 0 \]

\[ u^{n+1} = \tau (\lambda \delta t) u^n \]

Amplification factor \( \tau (\lambda \delta t) \).

Stability requires

\[ |\tau (\lambda \delta t)| \leq 1 \]

Forward Euler: \[ u^{n+1} = u^n + \Delta t \lambda u^n = (1 + \Delta t \lambda) u^n \]

\[ \Rightarrow \tau^{FE} = (1 + \Delta t \lambda) \]
Stability of Projective Integration

Stability region (PFE): \( \mathcal{D}^{PFE} = D \left( 1 - \frac{\delta t}{\Delta t}, \frac{\delta t}{\Delta t} \right) \cup D \left( 0, \left( \frac{\delta t}{\Delta t} \right)^{1/K} \right) \)

- First part corresponds to quickly damped modes
- Second part corresponds to slowly decaying modes

Choosing \( \delta t \) properly leads to accurate solution of slow modes while maintaining stability of fast modes.
If inner integrator is stable PRK parameters only need to fulfill standard RK stability conditions.

\[
\begin{pmatrix}
c \\
A \\
b^T
\end{pmatrix}
\]

Stability regions of lower-order methods are contained within those of higher-order methods.

Parameters for PFE will be valid for PRK.
Stability of Projective Integration, [Kevrekidis, 2014]

Projective Integration does not have unlimited stability w.r.t. λ.

For larger λ, more inner time steps K are necessary

\[ u^{n,k+1} = u^{n,k} + \delta t D(u^{n,k}) \]

for \( k = 0, 1, \ldots, K \):

Equivalently, smaller extrapolation steps \( \Delta t \) could be chosen

\[ u^{n+1} = u^{n,K+1} + (\Delta t - (K + 1)\delta t) \cdot \frac{u^{n,K+1} - u^{n,K}}{\delta t} \]

Still speedup w.r.t. standard Forward Euler method.
Projective Integration does not have unlimited stability w.r.t. $\lambda$.

For larger $\lambda$, more inner time steps $K$ are necessary

$$u^{n,k+1} = u^{n,k} + \delta t D\left(u^{n,k}\right)$$

for $k = 0, 1, \ldots, K$.

Equivalently, smaller extrapolation steps $\Delta t$ could be chosen

$$u^{n+1} = u^{n,K+1} + (\Delta t - (K + 1)\delta t) \cdot \frac{u^{n,K+1} - u^{n,K}}{\delta t}$$

Still speedup w.r.t. standard Forward Euler method.

For kinetic equations, we can still get unlimited stability.
Parameter choice, [Lafitte et al., 2017]

\[ \partial_t f^\epsilon + \frac{v}{\epsilon^\gamma} \partial_x f^\epsilon = -\frac{Q(f^\epsilon)}{\epsilon^{\gamma+1}} \]

\( \gamma = 0 \) hydrodynamic scaling, \\
\( \gamma = 1 \) diffusive scaling

**Parameter choice**

- \( \Delta t = O(\Delta x) \) for hydrodynamic limit
- \( \delta t = O(\epsilon) \) for hydrodynamic limit
- \( \Delta t = O(\Delta x^2) \) for diffusive limit
- \( \delta t = O(\epsilon^2) \) for diffusive limit
- \( K \) is small number, typically \( K \leq 3 \)
Parameter Example, [Lafitte et al., 2017]

Discrete Velocity Method, linearized BGK, third-order upwind discretization

$$\delta t = \epsilon,$$

$$K \geq 2,$$

$$\Delta t \leq \min \left( \frac{3\Delta x}{4c_0}, \frac{3\Delta x}{8} \right)$$

where $c_0$ is related to the DVM discretization.

$\Delta t$ might be larger, this is only an estimate.
Properties of PI

- Small number of inner steps sufficient, $K$ independent of $\epsilon$
- Inner and outer integrators can be RK schemes $\Rightarrow$ high-order
- Works best for spectral gap
Spectral gap in shock tube

Moment models are ideally suited for Projective Integration
No classical spectral gap

Discrete values for $\epsilon$

Continuous $\epsilon \sim \rho(t, x)$

- adjust number of inner time steps: $K = O(\log(1/\epsilon))$
- prevent stability region split $\Rightarrow [0,1]$-stable method $\Rightarrow \Delta t$ limited by $\epsilon$
- Telescopic Projective Integration (TPI)
**Idea:**

Nested Projective Integration
Integrators at different levels

**Innermost integrator**
- needs to capture fastest components and damp them
- higher-order methods result in stability restriction for other levels
- simply use Forward Euler (FE)

**Projective integrators**
- outermost dominates accuracy of the scheme
- simplest version Forward Euler (TPFE)
- higher-order (outermost) Runge Kutta (TPRK)
Parameter selection [Melis, Samaey, 2018]

- number of steps at level $l$: $K_l$
- time step size at level $l$: $\delta t_l$
- extrapolation size at level $l$: $M_l$

Choice based on spectrum of the respective method
Parameter selection

\[ \partial_t u + A_M \partial_x u = -\frac{\omega(x)}{\epsilon} Qu \]

discrete \( \omega \) levels

- Capture each eigenvalue cluster with one level
- Merge adjacent clusters to one
- Possibility to construct a connected stability region, depends on \( \epsilon \)
Parameter selection

\[ \partial_t u + A_M \partial_x u = -\frac{\omega(x)}{\epsilon} Qu \]

**continuous \( \omega \)**

- Need to construct a connected stability region, depends on \( \epsilon \)
- [0, 1]-stable method
- Number of levels depends on \( \log(1/\epsilon) \)
Summary of Projective Integration

\[ \text{PI} = \text{few inner steps} + \text{extrapolation} \]

- PRK, TPI extensions
- AP for standard kinetic equations
- TPI speedup w.r.t. FE

Next steps
- use PI for moment models
Summary of Projective Integration

**PI** = few inner steps + extrapolation

- PRK, TPI extensions
- AP for standard kinetic equations
- TPI speedup w.r.t. FE

Next steps

- use PI for moment models

Thank you for your attention!
References

Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations,

Projective methods for stiff differential equations: problems with gaps in their
eigenvalue spectrum,

Implicit-Explicit Runge-Kutta schemes and applications to hyperbolic systems with
relaxations

A high-order relaxation method with projective integration for solving nonlinear
systems of hyperbolic conservation laws,

Telescopic Projective Integration for Linear Kinetic Equations with Multiple
Relaxation Times,