Numerical Solutions to Differential Equations
Lecture Notes #16 — Hybrid Methods

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Spring 2015
Outline

1. Hybrid Methods
   - Introduction
   - Byrne-and-Lambert’s Pseudo Runge-Kutta Methods
   - Generalized Linear Multistep Methods

2. General Linear Methods
   - First Pass: GLM-lite
   - GLM-lite: Old Methods, New Notation...
   - New Methods, New Notation...
So far we have looked at three strategies for improving on Euler’s method

1. Taylor Series Methods
   - Best used when the Taylor expansion of \( f(t, y(t)) \) is available and cheap/easy to compute.
   - **Stiffness:** Small stability region. Step-size \( h \) very restrictive.
Runge-Kutta Methods

- When the Taylor expansion of \( f(t, y(t)) \) is not easily computable (or prohibitively expensive), but multiple evaluation of \( f(t, y(t)) \) incur a reasonable amount of work, then RK-methods are a good choice.

- **Stiffness:** When the problem is stiff, we have to use fully implicit RK-methods. We have seen that there are A-stable \( s \)-stage \( 2s \)-order methods (Gauss-Legendre) for arbitrary \( s \), as well as L-stable \( s \)-stage \((2s-1)\)-order methods (Radau I-A, and II-A).
In the Rear-view Mirror III

3 Linear Multistep Methods

- Explicit LMMs only require one new function evaluation per step, making them very competitive (fast and cheap). Used in the predictor-corrector context P(EC)$^\mu$, only $(1+\mu)$ evaluations per step are required.

- The main **drawback** is that LMMs are not self-starting, so we need an RK- or Taylor-series method (possibly with Richardson Extrapolation) to generate enough accurate starting information.

- **Stiffness:** If/when we can live with an A($\alpha$)-stable method, implementing efficient LMM-based stiff solvers is quite straightforward (at least up to order 6...)

Peter Blomgren, ⟨blomgren.peter@gmail.com⟩
We have 4 fundamental strategies on hand

1. Use more derivatives of $y(t)$, *(Taylor series methods)*
2. Use more past values, *(Linear Multistep Methods)*
3. Use more calculations per step, *(Runge Kutta)*
4. Use derivatives of $f(t, y(t))$, *(not used so far*)
Strategies

We have 4 fundamental strategies on hand

1. Use more derivatives of $y(t)$, *Taylor series methods*
2. Use more past values, *Linear Multistep Methods*
3. Use more calculations per step, *Runge Kutta*
4. Use derivatives of $f(t, y(t))$, *not used so far* *

Combinations in the literature:

- **Obreshkov**
  - more (past values + derivatives of $y(t)$)
  - $\sim$ LMM + Taylor

- **Rosenbrock**
  - more (derivatives of $y(t)$, and $f(t, y(t))$ + calculations per step)
  - $\sim$ RK + Taylor + $f$-derivatives

- **General Linear**
  - more (past values + calculations per step)
  - $\sim$ LMM + RK
1960–1970 Combining Runge-Kutta and Linear Multistep Method ideas; use of stage-derivatives (the RK-$k_i$s) in previous steps in the formation of the final step. (Byrne and Lambert, 1966)

1964–1965 Hybrid Methods (Gragg and Stetter, 1964; Gear, 1965; Butcher, 1965); now “modified multistep methods.” (Butcher)

* The class of multivalue^{LMM}-multistage^{RK} methods are referred to as General Linear Methods.
Byrne-and-Lambert’s RK+LMM idea boils down to $s$ standard RK-stages (1-2)

\[ Y_i = y_{n-1} + h \sum_{j=1}^{s} a_{ij} k_j^{[n]} \]  

(1)

\[ k_i^{[n]} = f(t_{n-1} + hc_i, Y_i) \]  

(2)

\[ y_n = y_{n-1} + h \left( \sum_{i=1}^{s} b_{i,0} k_i^{[n]} + \sum_{i=1}^{s} b_{i,1} k_i^{[n-1]} \right) \]  

(3)

followed by a modified step-assembly (3) using not only “current,” but also past $k_i$-values.

The associated Butcher array:

\[
\begin{array}{c|c}
\vec{c} & \begin{bmatrix} A \end{bmatrix} \\
\hline
\begin{bmatrix} b_0^T \\ b_1^T \end{bmatrix} & \end{array}
\]
or, in general

\[
Y_i = y_{n-1} + h \sum_{j=1}^{s} a_{ij} k_j^{[n]}
\]

\[
k_i^{[n]} = f(t_{n-1} + hc_i, Y_i)
\]

\[
y_n = y_{n-1} + h \left( \sum_{i=1}^{s} b_{i,0} k_i^{[n]} + \sum_{p=1}^{P} \left( \sum_{i=1}^{s} \bar{b}_{i,p} k_i^{[n-p]} \right) \right)
\]
The following \((s = 3)\)-stage Pseudo-RK method is order \((p = 4)\):

\[
\begin{array}{c|ccc}
0 & 1 & 2 & 1 \\
1 & \frac{1}{3} & 2 & 1 \\
2 & \frac{1}{12} & \frac{1}{3} & \frac{1}{4} \\
\hline
1 & \frac{11}{12} & \frac{1}{3} & \frac{1}{4} \\
\hline
& \frac{1}{12} & -\frac{1}{3} & -\frac{1}{4}
\end{array}
\]

Recall (Lecture #5)

**Theorem (Butcher, 2008: p.187)**

*If an explicit \(s\)-stage Runge-Kutta method has order \(p\), then \(s \geq p.\)*
Note that Pseudo-RK methods “inherit” the non-self-starting, and difficult-to-change-step-size properties from the LMM framework.

Starting and step-size changes can be handled with “classical” RK-methods, whose order of course must match the Pseudo-RK method in use.
Generalized Linear Multistep Methods, I  

- Generalizes LMM Predictor-Corrector pairs, by inserting additional *predictors*
- Additional predictors, usually, at off-step points

**Example (Off-step predictor at $\frac{8}{15} h$ — part 1)**

1. Predict the value at $t = t_{n-1} + \frac{8}{15} h = t_n - \frac{7}{17} h$  —  $y_{n-\frac{7}{15}}^{[p1]}$
2. Predict the value at $t = t_n$  —  $y_{n}^{[p2]}$
3. Correct the value at $t = t_n$  —  $y_{n}^{[c]}$
Example (Off-step predictor at $\frac{8}{15} h$ — part 2)

\[
y_{n-\frac{7}{15}}^{[p1]} = -\frac{529}{3375} y_{n-1} + \frac{3904}{3375} y_{n-2} + h \left( \frac{4232}{3375} f_{n-1} + \frac{1472}{3375} f_{n-2} \right)
\]
\[
f_{n-\frac{7}{15}}^{[p1]} = f \left( t_n - \frac{7}{15} h, y_{n-\frac{7}{15}}^{[p1]} \right)
\]
\[
y_n^{[p2]} = \frac{152}{25} y_{n-1} - \frac{127}{25} y_{n-2} + h \left( \frac{189}{92} f_{n-\frac{7}{15}}^{[p1]} - \frac{419}{100} f_{n-1} - \frac{1118}{575} f_{n-2} \right)
\]
\[
f_n^{[p2]} = f \left( t_n, y_n^{[p2]} \right)
\]
\[
y_n^{[c]} = y_{n-1} + h \left( \frac{25}{168} f_n^{[p2]} + \frac{3375}{5152} f_{n-\frac{7}{15}}^{[p1]} + \frac{19}{96} f_{n-1} - \frac{1}{552} f_{n-2} \right)
\]
Given that \( r \) (\( d \)-dimensional) quantities are passed from step-to-step, one full step is completed once given the values \( \vec{y}^{[n-1]} \) we have computed \( \vec{y}^{[n]} \):

\[
\vec{y}^{[n-1]} = \begin{bmatrix}
y_1^{[n-1]} \\
y_2^{[n-1]} \\
\vdots \\
y_r^{[n-1]}
\end{bmatrix} \quad \Rightarrow \quad \vec{y}^{[n]} = \begin{bmatrix}
y_1^{[n]} \\
y_2^{[n]} \\
\vdots \\
y_r^{[n]}
\end{bmatrix}
\]

using an \( s \)-stage method, during the step we compute \( s \) stage-values (\( \vec{Y}_i \)), and \( s \) associated stage derivatives (\( \vec{F}_i \))... We let, \( \vec{Y} \) and \( \vec{F} \) be the “supervectors” that contain the respective \( Y_i \) and \( F_i \) sub-vectors.
As for RK-methods, stages consist of linear combinations of stage-derivatives.

Additional linear combinations are needed to express the dependence on the *input* information.

... and the *output* quantities depend linearly on both the stage derivatives, and the input quantities.

All-in-all, we need 4 matrices to capture the computations of one stage:

\[
A = [a_{ij}]_{s,s}, \quad U = [u_{ij}]_{s,r}, \quad B = [b_{ij}]_{r,s}, \quad V = [v_{ij}]_{r,r}.
\]
The stage-computations are given by

\[ Y_i = h \sum_{j=1}^{s} a_{ij} F_j + \sum_{j=1}^{r} u_{ij} y[n-1]_j, \quad i = 1, 2, \ldots, s \]

\[ y_i^{[n]} = h \sum_{j=1}^{s} b_{ij} F_j + \sum_{j=1}^{r} v_{ij} y[n-1]_j, \quad i = 1, 2, \ldots, r \]

or, in more compact notation

\[ Y = h(A \otimes I_d)F + (U \otimes I_d)y^{[n-1]} \]

\[ y^{[n]} = h(B \otimes I_d)F + (V \otimes I_d)y^{[n-1]} \]
In all cases, we can express the GLM using an \((s + r) \times (s + r)\) matrix:

\[
\begin{bmatrix}
A_{s,s} & U_{s,r} \\
B_{r,s} & V_{r,r}
\end{bmatrix}
\]

it turns out we can cast quite a few of our well-known schemes in this notation...

Euler’s Method

\[
\begin{bmatrix}
0 & 1 \\
1 & 1
\end{bmatrix}
\]

Implicit Euler

\[
\begin{bmatrix}
1 & 1 \\
1 & 1
\end{bmatrix}
\]
The 2nd order, 2-stage Runge-Kutta, with Butcher array

\[
\begin{array}{c|cc}
0 & 0 & 1 \\
1 & 1 & \\
\hline
\frac{1}{2} & \frac{1}{2} & \\
\end{array}
\]

...can be expressed as a GLM:

\[
\begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 1 \\
\frac{1}{2} & \frac{1}{2} & 1
\end{bmatrix}
\]
The 3rd order, 3-stage Runge-Kutta, with Butcher array

\[
\begin{array}{c|ccc}
0 & 0 & 0 & 1 \\
\frac{1}{2} & 0 & 0 & 1 \\
1 & -1 & 2 & 1 \\
\hline
\frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 1 \\
\end{array}
\]

can be expressed as a GLM:

\[
\begin{bmatrix}
0 & 0 & 0 & 1 \\
\frac{1}{2} & 0 & 0 & 1 \\
-1 & 2 & 0 & 1 \\
\hline
\frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 1 \\
\end{bmatrix}
\]
The 4th order, 4-stage Runge-Kutta, with Butcher array

\[
\begin{bmatrix}
0 & 1 \\
\frac{1}{2} & 0 \frac{1}{2} \\
\frac{1}{2} & 0 \frac{1}{2} \\
1 & 0 0 1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \\
\end{bmatrix}
\]

can be expressed as a GLM:

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 \\
\frac{1}{2} & 0 & 0 & 0 & 1 \\
0 & \frac{1}{2} & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 \\
\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & 1 \\
\end{bmatrix}
\]
2nd order Adams-Bashforth, and Adams-Moulton methods:

\[ y_{n+1}^{AB} = y_n + \frac{h}{2} [3f_n - f_{n-1}] \]
\[ y_{n+1}^{AM} = y_n + \frac{h}{2} [f_{n+1} + f_n] \]

in GLM-notation:

\[
\begin{bmatrix}
0 & 1 & \frac{3}{2} & -\frac{1}{2} \\
0 & 1 & \frac{3}{2} & -\frac{1}{2} \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
\frac{1}{2} \\
\frac{1}{2} \\
1
\end{bmatrix}
\]
Uh?!?

\[
\begin{bmatrix}
0 & 1 & \frac{3}{2} & -\frac{1}{2} \\
0 & 1 & \frac{3}{2} & -\frac{1}{2} \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

This means,

\[
Y_1 = y_1^{[n-1]} + \frac{3}{2} y_2^{[n-1]} - \frac{1}{2} y_3^{[n-1]}
\]

solution

\[
y_1^{[n]} = y_1^{[n-1]} + \frac{3}{2} y_2^{[n-1]} - \frac{1}{2} y_3^{[n-1]}
\]

step-derivative

\[
y_2^{[n]} = h F_1 \equiv f(Y_1)
\]

step-derivative

\[
y_3^{[n]} = y_2^{[n-1]}
\]
2nd order Adams-Bashforth, and Adams-Moulton methods:

\[
\begin{bmatrix}
0 & 1 & \frac{3}{2} & -\frac{1}{2} \\
0 & 1 & \frac{3}{2} & -\frac{1}{2} \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

Operating in P(EC)E mode:

\[
\begin{bmatrix}
\frac{1}{2} & 0 & 1 & \frac{3}{2} & -\frac{1}{2} \\
\frac{1}{2} & 0 & 1 & \frac{1}{2} & 0 \\
\frac{1}{2} & 0 & 1 & \frac{1}{2} & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

Peter Blomgren, ⟨blomgren.peter@gmail.com⟩
Byrne-Lambert’s 4th order 3-stage Pseudo-RK:

\[
\begin{array}{ccc}
0 & 1 & 2 \\
\frac{1}{2} & \frac{1}{3} & \frac{4}{3} \\
1 & \frac{1}{3} & \frac{1}{4} \\
\frac{1}{12} & -\frac{1}{3} & -\frac{1}{4}
\end{array}
\]

GLM:

\[
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 \\
-\frac{1}{3} & \frac{4}{3} & 0 & 1 & 0 & 0 & 0 \\
\frac{11}{12} & \frac{1}{3} & \frac{1}{4} & 1 & \frac{1}{12} & -\frac{1}{3} & -\frac{1}{4} \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\]
Off-step predictor at $\frac{8}{15} h$ —

\[
\begin{bmatrix}
0 & 0 & 0 & \frac{-529}{3375} & \frac{3904}{3375} & \frac{4232}{3375} & \frac{1472}{3375} \\
\frac{189}{92} & 0 & 0 & \frac{152}{25} & \frac{-127}{25} & \frac{-419}{100} & \frac{-1118}{575} \\
\frac{3375}{5152} & \frac{25}{168} & 0 & 1 & 0 & \frac{19}{96} & \frac{-1}{552} \\
\frac{3375}{5152} & \frac{25}{168} & 0 & 1 & 0 & \frac{19}{96} & \frac{-1}{552} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\]

The output quantities are:

\[
\begin{align*}
y_1^{[n]} & \approx y(t_n), \\
y_2^{[n]} & \approx y(t_{n-1}), \\
y_3^{[n]} & \approx h y'(t_n), \\
y_4^{[n]} & \approx h y'(t_{n-1}).
\end{align*}
\]