Advanced methods for ODEs

September 28, 2016

Outline

A) Runge-Kutta methods

Improved Euler method

Reminder: Euler method

$$x_{k+1} = x_k + hg(x_k)$$

We can build a "more accurate" method by taking the gradient elsewhere

$$\begin{aligned} \tilde{x}_k &= x_k + hg(x_k) \\ x_{k+1} &= x_k + hg(\tilde{x}_k). \end{aligned}$$

This is exactly the extra-gradient algorithm if we replace $g(x(t))$ by $-\nabla f(x(t))$:

$$\begin{aligned} \tilde{x}_k &= x_k - h\nabla f(x_k) \\ x_{k+1} &= x_k - h\nabla f(\tilde{x}_k). \end{aligned}$$

However, we can potentially improve the scheme if we give more flexibility:

$$\begin{aligned} \tilde{x}_k &= x_k - a_{21} h\nabla f(x_k) \\ x_{k+1} &= x_k - h\left(b_1 \nabla f(x_k) + b_2 \nabla f(\tilde{x}_k)\right). \end{aligned}$$

"Improved" extra-gradient method:

$$
\begin{aligned}
\tilde{x}_k &= x_k - a_{21} h \nabla f(x_k) \\
x_{k+1} &= x_k - h \left(b_1 \nabla f(x_k) + b_2 \nabla f(\tilde{x}_k) \right).
\end{aligned}
$$

We can rewrite the algorithm in another way

$$
\begin{aligned}
k_1 &= -\nabla f(x_k) \\
k_2 &= -\nabla (x_k + a_{21} h k_1) \\
x_{k+1} &= x_k + h(b_1 k_1 + b_2 k_2)
\end{aligned}
$$

Remark: There are **two** oracle calls!

Two-stages Runge-Kutta method (RK2)

$$\begin{aligned} k_1 &= g(x_k) \\ k_2 &= g(x_k + a_2 h k_1) \\ x_{k+1} &= x_k + h(b_1 k_1 + b_2 k_2) \end{aligned}$$

Unlike Euler's method, Runge-Kutta methods use derivatives in the interior of the interval.

Intuition:

- The idea is to *explore* the domain with intermediate points,
- The intermediate slopes are the stages k_i,
- The step is a linear combination (in fact, a weighted mean) of the k_i times h.

Recall: Implicit Euler method

$$x_{k+1} = x_k + hg(x_{k+1})$$

We can use the same trick as before:

$$\begin{aligned} k_1 &= g\big(x_k + h(a_{11}k_1 + a_{12}k_2)\big) \\ k_2 &= g\big(x_k + h(a_{21}k_1 + a_{22}k_2)\big) \\ x_{k+1} &= x_k + h(b_1k_1 + b_2k_2) \end{aligned}$$

This is the most general two-stages RK method.

Remark: We need to solve a nonlinear system of equation of **2*n* variables!**
$\rightarrow$ Not possible (in general) to solve the k_i independently.

General Runke-Kutta method with s stages (RKs)

$$k_i = g\left(x_k + h\sum_{j=1}^{s} a_{ij}k_j\right)$$

$$x_{k+1} = x_k + h\sum_{i=1}^{s} b_i k_i$$

If the method is explicit:

$$k_i = g\left(x_k + h\sum_{j=1}^{i-1} a_{ij}k_j\right)$$

$$x_{k+1} = x_k + h\sum_{i=1}^{s} b_i k_i$$

General RKs method:

$$k_i = g\left(x_k + h\sum_{j=1}^{s} a_{ij}k_j\right), \qquad x_{k+1} = x_k + h\sum_{i=1}^{s} b_i k_i$$

Butcher tableau (general representation of RKs method)

$$\begin{array}{|cccc}
a_{11} & a_{12} & \cdots & a_{1,s} \\
a_{21} & a_{22} & \cdots & a_{2,s} \\
\vdots & & \ddots & \\
a_{s1} & a_{s2} & \cdots & a_{s,s-1} \\
\hline
b_1 & b_2 & \cdots & b_s
\end{array}$$

Case for explicit methods:

$$\begin{array}{|ccccc}
0 & & & & \\
a_{21} & & & & \\
a_{31} & a_{32} & & & \\
\vdots & \vdots & \ddots & & \\
a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

Example: Butcher tableau for ODE45 (common function in Matlab for ODE)

$$
\begin{array}{l|ccccccc}
 & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & 0 \\
 & \frac{3}{40} & \frac{9}{40} & 0 & 0 & 0 & 0 & 0 \\
 & \frac{44}{45} & -\frac{56}{15} & \frac{32}{9} & 0 & 0 & 0 & 0 \\
 & \frac{19372}{6561} & -\frac{25360}{2187} & \frac{64448}{6561} & -\frac{212}{729} & 0 & 0 & 0 \\
 & \frac{9017}{3168} & -\frac{355}{33} & \frac{46732}{5247} & \frac{49}{176} & -\frac{5103}{18656} & 0 & 0 \\
 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & 0 \\
\hline
 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & 0 \\
 & \frac{71}{57600} & 0 & -\frac{71}{16695} & \frac{71}{1920} & -\frac{916280711922443}{18014398509481984} & \frac{22}{525} & -\frac{1}{40}
\end{array}
$$

The last line is the error estimator, used for adaptive stepsize.

Consistency and order

The coefficients a_{ij} and b_i are computed in function of the desired order.

(Recall) A method has order p if

$$T(h) = \frac{1}{h}\big(x_1 - x(t_1)\big) = O(h^p) \qquad \text{where } x_0 = x(t_0)$$

We thus need to compare the Taylor expansion of x_1 and $x(t_1)$.

Example: explicit RK2

Expansion of $x(t_1)$:

$$\begin{aligned}
x(t_1) &= x_0 + h\frac{d}{dt}x(t_0) + \frac{h^2}{2}\frac{d^2}{dt^2}x(t_0) + O(h^3) \\
&= x_0 + hg(x_0) + \frac{h^2}{2}\frac{d}{dt}g(x(t_0)) + O(h^3) \\
&= x_0 + hg(x_0) + \frac{h^2}{2}\frac{d}{dx}g(x(t_0))\frac{d}{dt}x(t_0) + O(h^3) \\
&= x_0 + hg(x_0) + \frac{h^2}{2}\frac{d}{dx}g(x_0)g(x_0) + O(h^3)
\end{aligned}$$

Expansion of x_1:

$$\begin{aligned}
k_1 &= g(x_0) \\
k_2 &= g(x_0 + ha_{21}k_1) = g(x_0 + ha_{21}g(x_0)) \\
x_1 &= x_0 + h(b_1k_1 + b_2k_2)
\end{aligned}$$

We only need to develop k_2:

$$k_2 = g(x_0) + ha_{21}g(x_0)\frac{d}{dx}g(x_0) + O(h^2)$$

Expansion of $x(t_1)$:

$$x(t_1) = x_0 + hg(x_0) + \frac{h^2}{2} g(x_0) \frac{d}{dx} g(x_0) + O(h^3)$$

Expansion of x_1:

$$\begin{aligned}
k_1 &= g(x_0) \\
k_2 &= g(x_0) + ha_{21} g(x_0) \frac{d}{dx} g(x_0) + O(h^2) \\
x_1 &= x_0 + h(b_1 k_1 + b_2 k_2) \\
&= x_0 + hb_1 g(x_0) + hb_2 g(x_0) + b_2 a_{21} h^2 g(x_0) \frac{d}{dx} g(x_0) + O(h^3)
\end{aligned}$$

The error becomes

$$x_1 - x(t_1) = (b_1 + b_2 - 1) hg(x_0) + \left(b_2 a_{21} - \frac{1}{2} \right) h^2 g(x_0) \frac{d}{dx} g(x_0) + O(h^3)$$

To have an order 2 RK2 method, we need

$$\begin{aligned}
b_1 + b_2 &= 1 \\
b_2 a_{21} &= \frac{1}{2}
\end{aligned}$$

Runge-Kutta methods: summary

- Unlike Euler method, RK methods use derivatives inside the interval $[t_k, t_{k+1}]$
- The stages k_i are the intermediates slopes in the interval
- The next iterate is a weighted mean of the k_i
- The coefficients are computed in function of the desired pupose (e.g. high order of accuracy, good error estimator, ...)

B) Linear multistep methods

Linear multistep method and Nesterov's method

Idea of linear multistep methods: combine linearly previous evaluations of x_k and $g(x_k)$. General formulation of explicit s-steps methods:

$$\begin{aligned} x_k &= \alpha_0 x_{k-s} + \alpha_1 x_{k-(s-1)} + \dots + \alpha_{s-1} x_{k-1} \\ &\quad + h\left(\gamma_0 g(x_{k-s}) + \gamma_1 g(x_{k-(s-1)}) + \dots + \gamma_{s-1} g(x_{k-1})\right) \end{aligned}$$

If the method is implicit,

$$\begin{aligned} x_k &= \alpha_{0s} x_{k-s} + \alpha_1 x_{k-(s-1)} + \dots + \alpha_{s-1} x_{k-1} \\ &\quad + h\left(\gamma_0 g(x_{k-s}) + \gamma_1 g(x_{k-(s-1)}) + \dots + \gamma_{s-1} g(x_{k-1}) + \gamma_s g(x_k)\right) \end{aligned}$$

Linear multistep methods are usually written into the following form:

$$\sum_{i=0}^{s} \alpha_{s-i} x_{k-i} = h \sum_{i=0}^{s} \gamma_{s-i} g(x_{k-i})$$

Convention: $\alpha_s = 1$. If the method is explicit, $\gamma_s = 0$.

Nesterov's method for L-smooth, μ-strongly convex functions:

$$y_{k+1} = x_k - \frac{1}{L} f'(x_k)$$
$$x_{k+1} = y_{k+1} + \beta(y_{k+1} - y_k)$$

where $\beta = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$. If we expand the recursion:

$$\begin{aligned} x_{k+1} &= (1+\beta)\left(x_k - \frac{1}{L} f'(x_k)\right) - \beta\left(x_{k-1} - \frac{1}{L} f'(x_{k-1})\right) \\ &= (1+\beta)x_k - \beta x_{k-1} - \frac{1}{L}\Big((1+\beta) f'(x_k) - \beta f'(x_{k-1})\Big) \end{aligned}$$

It is exactly a linear multistep method with parameters

$$\begin{array}{ccc} \alpha_2 = 1 & \alpha_1 = -(1+\beta) & \alpha_0 = \beta \\ h\gamma_2 = 0 & h\gamma_1 = \frac{1+\beta}{L} & h\gamma_0 = -\frac{\beta}{L} \end{array}$$

Zero-stability

Zero-stability is simply the stability of the method when $g(x) = 0$ (for any starting values x_i). In the case of linear multistep methods, zero-stability condition is

$$\forall x_0, ..., x_{s-1}, \qquad \text{if } x_k = -\sum_{i=0}^{s-1} \alpha_{s-i} x_{k-i} \text{ then } \lim_{k\to\infty} \|x_k\| = 0.$$

For zero-stable methods the errors made in initial conditions fade over time.

The explicit solution of

$$x_k = -\sum_{i=0}^{s-1} \alpha_{s-i} x_{k-i}$$

is linked to the roots r_i of the polynomial

$$\rho(z) = \sum_{i=0}^{s} \alpha_i z^i$$

Assume all roots r_i of $\rho(z)$ are distinct. For some constants m_i,

$$x_k = \sum_{i=1}^{s} m_i r_i^k$$

We have thus a simple condition for zero-stability.

Theorem

A linear multistep method is zero stable if and only if the roots r_i of $\rho(z)$ satisfy

$$\left| r_i \right| < 1$$

Consistency and order

Assume $t_i = ih$ and

$$x_0 = x(t_0),\ x_1 = x(t_1),\ ...,\ x_{s-1} = x(t_{s-1}).$$

Let x_s be computed using a linear multistep method. In that case, the truncation error is

$$T(h) = \frac{1}{h}\left(x(t_s) - x_s\right)$$

The coefficients α_i and γ_i are computed, for example, in function of the desired order of accuracy.

Example: Method of order one.

Expansion of $x(t_i)$ (Recall: constant stepsize implies $t_i = i \times h$):

$$
\begin{aligned}
x(t_i) = x(ih) &= x_0 + ih\frac{d}{dt}x(t_0) + O(h^2) \\
&= x_0 + ihg(x(t_0)) + O(h^2)
\end{aligned}
$$

Expansion of $g(x(t_i))$:

$$
g(x(t_i)) = g(x_0) + O(h)
$$

We can thus explicitly form x_s:

$$
\begin{aligned}
x_s &= -\sum_{i=0}^{s-1} \alpha_i x(t_i) + h\sum_{i=0}^{s} \gamma_i g(x(t_i)) \\
&= -\sum_{i=0}^{s-1} \alpha_i x_0 - \sum_{i=0}^{s-1} i\alpha_i h g(x_0) + h\sum_{i=0}^{s} \gamma_i g(x_0) \\
&= -\sum_{i=0}^{s-1} \alpha_i x_0 + h\sum_{i=0}^{s-1}(\gamma_i - i\alpha_i) g(x_0) + h\gamma_s g(x_0) + O(h^2)
\end{aligned}
$$

We can thus explicitly form x_s:

$$x_s = -\sum_{i=0}^{s-1} \alpha_i x_0 + h \sum_{i=0}^{s-1} (\gamma_i - i\alpha_i) g(x_0) + \gamma_s g(x_0) + O(h^2)$$

We can plug it in the expression of $T(h)$:

$$\begin{aligned}
T(h) &= \frac{1}{h}\Big(x_s - x(t_s)\Big) \\
&= \frac{1}{h}\Big(x_s - x_0 - shg(x_0)\Big) \\
&= \frac{1}{h}\Big(x_s - \alpha_s x_0 - sh\alpha_s g(x_0)\Big) \qquad \text{because } \alpha_s = 1 \\
&= \frac{1}{h}\Big(-\sum_{i=0}^{s} \alpha_i x_0 + h \sum_{i=0}^{s} (\gamma_i - i\alpha_i) g(x_0)\Big) + O(h)
\end{aligned}$$

Theorem

A linear multistep method

- *is consistent if* $\sum_{i=0}^{s} \alpha_i = 0 \Leftrightarrow \sum_{i=0}^{s-1} \alpha_i = -1$,
- *has order one if* $\sum_{i=0}^{s} \gamma_i = \sum_{i=0}^{s} i\alpha_i$,
- *has order* p *if for all* $q = 1, ..., p$, *we have* $\sum_{i=0}^{s} \frac{i^{q-1}}{(q-1)!}\gamma_i = \sum_{i=0}^{s} \frac{i^q}{q!}\alpha_i$.

Theorem

A linear multistep method

- *is consistent if* $\sum_{i=0}^{s-1} \alpha_i = -1$.

This condition is extremely important. Assume we already start at the solution of our problem, i.e.

$$x_0 = x^* = \arg\min_x f(x)$$

Assume $\sum_{i=0}^{s-1} \alpha_i \neq -1$. Then x_s will be

$$\begin{aligned} x_s &= -\sum_{i=0}^{s-1} \alpha_i x^* - h \sum_{i=0}^{s} \gamma_i \underbrace{\nabla f(x^*)}_{=0} \\ &= \left(-\sum_{i=0}^{s-1} \alpha_i\right) x^* \\ &\neq x^* \end{aligned}$$

So our method will diverge even if we start at the optimum. In fact, the condition $\sum_{i=0}^{s} \alpha_i = 0$ force the method to avoid multiplicative gain or damping.

Runge-Kutta versus Linear Multistep Methods

- The number of oracle calls for one step is higher for RK methods than linear multistep methods,
- Estimation of the error $T(h)$ for RK method is simpler than for linear multistep methods,
- RK methods are more flexible in the stepsize: we chan change h at each iteration, but this is not possible for multistep methods,
- In general, analysis of RK method is simpler,
- RK method are less sensitive to initial conditions,
- Implicit RKs method requires solving a system of $s \times n$ nonlinear equations, unlike linear multistep method (only a system of n nonlinear equations).

Runge-Kutta versus Linear Multistep Methods (part two)

Usually, RK methods

- are used for short interval, i.e. $[t_{\min}, t_{\max}]$ is "small",
- give accurate solution,
- are often explicit.

Linear multistep methods

- are used for large interval,
- can handle badly-conditioned problems
- are often implicit, but not always.

C) Absolute Stability

Region of absolute stability and A-Stability

(Recall) Region of absolute stability and A-stability

Definition

Assume x_k is generated by a specific integration method (with parameter h) on the test problem

$$\frac{d}{dt}x(t) = \lambda x(t).$$

Then, the region of absolute stability is the set of value $\lambda h \in \mathbb{Z}$ which makes

$$\lim_{k\to\infty} \|x_k\| \neq \infty.$$

Definition

A method is called A-stable when all values $\lambda h : \mathrm{R}(\lambda h) < 0$ is contained in the region of absolute stability.

We thus need to analyze Runge-Kutta method and linear multistep method when

$$g(x(t)) = \lambda x(t) \qquad \text{(Solution is } x(t) = x_0 e^{\lambda t}\text{)}.$$

Region of absolute stability for RK methods

Example: two-stages explicit RK method.

$$
\begin{aligned}
k_1 &= g(x_0) \\
&= \lambda x_0
\end{aligned}
$$

$$
\begin{aligned}
k_2 &= g(x_0 + a_{21} h k_1) \\
&= \lambda (x_0 + a_{21} \lambda h x_0) \\
&= x_0 \left(\lambda + a_{21} \lambda^2 h \right)
\end{aligned}
$$

$$
\begin{aligned}
x_1 &= x_0 + h b_1 k_1 + h b_2 k_2 \\
&= x_0 + b_1 \lambda h x_0 + x_0 \left(b_2 \lambda h + b_2 a_{21} \lambda^2 h^2 \right) \\
&= x_0 \underbrace{\left[1 + \lambda h \left(b_1 + b_2 \right) + \left(\lambda h \right)^2 \left(b_2 a_{21} \right) \right]}_{= p(\lambda h)}
\end{aligned}
$$

$$x_1 = x_0 \underbrace{\left[1 + \lambda h \left(b_1 + b_2\right) + (\lambda h)^2 \left(b_2 a_{21}\right)\right]}_{=p(\lambda h)}$$

We thus have

$$\|x_k\| \leq \|x_0\| \left|p(\lambda h)\right|^k$$

The region of absolute stability is thus the set

$$\left\{\lambda h : \left|p(\lambda h)\right| \leq 1\right\}$$

If we impose the method to have order 2, then $b_1 + b_2 = 1$ and $b_2 a_{21} = \frac{1}{2}$:

$$p(\lambda h) = 1 + \lambda h + \frac{1}{2}\lambda^2 h^2$$

We can extend this idea to explicit order s RK methods:

$$p_s(\lambda h) = 1 + \lambda h + \frac{1}{2}(\lambda h)^2 + ... + \frac{1}{s!}(\lambda h)^s$$

It is exactly the Taylor expansion of $e^{\lambda h}$ (exact solution of the ODE).

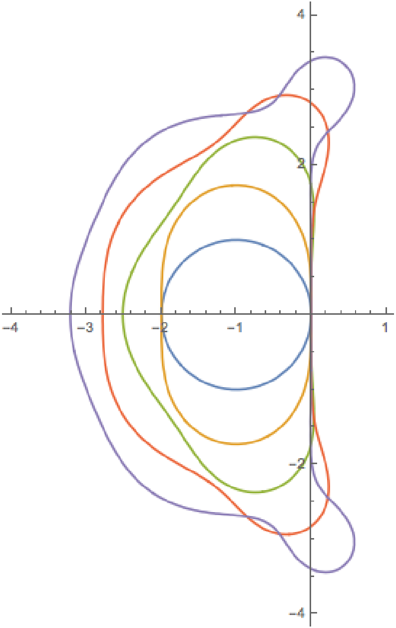

Figure: Region of absolute stability of some RK methods (order = 1, 2, . . . ,5)

Region of absolute stability for linear multistep methods

We have the recurrence

$$\begin{aligned}\sum_{i=0}^{s} \alpha_i x_{k-s+i} &= h \sum_{i=0}^{s} \gamma_i g(x_{k-s+i}) \\ &= h \sum_{i=0}^{s} \gamma_i \lambda x_{k-s+i}\end{aligned}$$

In other words, we have a homogenous difference equation

$$\sum_{i=0}^{s} (\alpha_i - \lambda h \gamma_i) x_{k-s+i} = 0$$

We have a homogenous difference equation

$$\sum_{i=0}^{s}(\alpha_i - \lambda h \gamma_i) x_{k-s+i} = 0$$

We have that x_k is linked to the roots r_i of the polynomial

$$q_{\lambda h}(z) = \sum_{i=0}^{s}(\alpha_i - \lambda h \gamma_i) z^i$$

Assume all roots are distinct, we have for some constant m_j

$$\|x_k\| \leq \|x_0\| \sum_{j=1}^{s} m_j |r_j(\lambda h)|^k$$

The region of absolute stability is thus defined by

$$\left\{\lambda h : \left|r_j(\lambda h)\right| \leq 1\right\}$$

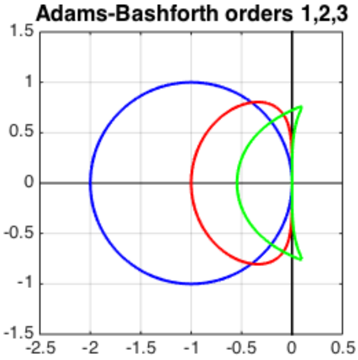

Figure: Region of absolute stability of some linear multistep methods (order = 1, 2, 3)