CFD SPRING 2013

A. DONEV

TEMPORAL INTEGRATORS

Temporal integrators can be clasified along several lines:

- one-step (multistage)
- multi-step single stage
- multi-step multi stage

- explicit
- implicit
- mixed implicit-explicit

RUNGE-KUTTA SCHEMES

$$w'(t) = F(t, w(t)), \quad t > 0$$

One-step RK with s stages

$$\left\{ \begin{aligned} & w^{n+1} = w^n + \tau \sum_{i=1}^{s} b_i \, F(t_n + c_i \tau, w^{n,i}) \\ & w^{n,i}, \quad i = 1 \dots s \leftarrow \text{intermediate stage value} \\ & w^{n,i} = w^n + \tau \sum_{j=1}^{s} \alpha_{ij} \, F(t_n + c_j \tau, w^{n,j}) \end{aligned} \right.$$

The coefficients $\alpha_{ij} = a_{ij}$, b_i, and

$C_i = \sum_{j=1}^{s} \alpha_{ij}$ determine the method

These coefficients are usually collected (3)
in a Butcher-array (tableau)

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \rightarrow \begin{cases} \text{Explicit if } A \text{ is} \\ \text{lower triangular} \\ \alpha_{ij} = 0 \text{ for } j \geq i \end{cases}$$

In CFD, due to high cost of linear solvers, usually A is at most <u>diagonally-implicit</u> (so each stage requires a single linear solve independently of the other stages)

Order conditions

$$\begin{cases} p=1: & b^T e = 1 & \text{also} \\ p=2: & b^T c = 1/2 & \text{also} \\ p=3: & b^T c^2 = 1/3 & b^T A c = 1/6 \end{cases}$$

The method also has a *stage order* q

$$q = \min_i \{q_i\}, \quad p \geq q$$

$$\boxed{w(t_n + c_i \tau) - w_*^{n,i} = O(\tau^{q+1})}$$

If $w(t_n) = w_*$

Explicit trapezoidal rule

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1/2 & 1/2 \end{array}$$

(we already studied it)

Explicit midpoint rule:

$$\begin{array}{c|cc} 0 & & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$$

$$\begin{cases} w^{n+1} = w^n + \tau F^{n+1/2} \\ F^{n+1/2} = F\left(t_n + \frac{\tau}{2}, w^{n+1/2}\right) \\ w^{n+1/2} = w^n + \frac{\tau}{2} F^n \end{cases}$$

Note that linearly these two are equivalent.

One can also write implicit versions

Rosenbrock methods

Purely implicit methods require *nonlinear* solves. An alternative is to limit to *linear* systems only by linearizing F around (t^n, w^n) (or some related point). This gives *linearly implicit* or *Rosenbrock* schemes.

$$w'(t) = F(w(t)) \quad \text{(autonomous)}$$

$$\begin{cases} w^{n+1} = w^n + \sum_{i=1}^{s} b_i k_i \\ k_i = \tau F\Big(w^n + \sum_{j=1}^{i-1} \alpha_{ij} k_j\Big) + \tau A \sum_{j=1}^{i} \gamma_{ij} k_j \end{cases}$$

$i-1 \leftarrow$ explicit

$i \leftarrow$ implicit

where $A \equiv A^n = \frac{\partial F(w^n)}{\partial w}$ (Jacobian)

Example: Advective term in Navier-Stokes is $(v \cdot \nabla) v$. But one can approximate with $(v^n \cdot \nabla) v$, giving a linear system.

Rosenbrock methods are related to fully nonlinear methods if Newton's method is used to solve the nonlinear systems.

Often $\gamma_{ii} = \gamma = \text{const.}$

$$I - \tau \gamma A = M = \text{const.}$$

Examples

$$\begin{cases} w^{n+1} = w^n + k_1 \\ k^1 = \tau F(w^n) + \frac{\tau}{2} A^n k_1 \end{cases}$$

← second order

$$A = \frac{\partial F}{\partial w}(w^n) + O(\tau)$$

← one can approximate

Order Reduction

Higher-order RK methods sometimes exhibit a reduction in the order of (global) accuracy when there are certain inhomogeneous boundary conditions.

In fact, the same applies for stiff-problems (analysis is hard)

Conclusion of analysis (see Hundsdorfer & Verwer)

Any explicit or implicit RK method of classical order $\boxed{p \geq 3}$ and stage order $\boxed{q < p-1}$ may suffer an order reduction to order $\boxed{q+1} \leq p$ for stiff problems or when (non-trivial) boundary conditions are present

(h plays the role of stiffness)

There are some mitigation techiques, but the point here is to watch out with high-order methods

Linear Multistep methods

$$\sum_{j=0}^{k} \alpha_j w^{n+j} = \tau \sum_{j=0}^{k} \beta_j F(t_{n+j}, w^{n+j})$$

$$\left\{ \begin{array}{l} k = \text{number of } \underline{\text{past}} \text{ values used} \\ \qquad \text{(memory)} \\ \\ \beta_k = 0 \quad = \text{explicit} \\ \alpha_k = 1 \end{array} \right.$$

Note that this is as efficient as a <u>single</u> stage of RK.

But one must <u>store</u> $w^n, \dots, w^{n+k}$

these methods are <u>not</u> self-starting

Typically a single-step method is used to initialize.

<u>Local order</u>: of <u>consistency</u>: p

$$\left\{ \begin{array}{l} \displaystyle\sum_{j=0}^{k-1} \alpha_j w(t_{n+j}) + \alpha_k w_*^{n+k} = \tau \sum_{j=0}^{k-1} \beta_j F^{n+j} + \tau \beta_k F_*^{n+k} \\ \boxed{w(t_{n+k}) - w_*^{n+k} = O(\tau^{p+1})} \end{array} \right.$$

[plug exact values for past steps and do Taylor series analysis]

[<u>Note</u>: Starting values $w_1, \dots, w_{k-1}$ must also be computed with convergence order p

Examples: Leap-frog method

$$w^{n+2} - w^{n} = 2\tau F(t_{n+1}, w^{n+1}).$$

(for wave equations)
second-order

Adams-Bashforth methods

2-step:
$$\boxed{w^{n+2} - w^{n+1} = \frac{3}{2}\tau F^{n+1} - \frac{\tau}{2} F^{n}}$$

often used to treat advection!

3-step:
$$w^{n+3} - w^{n+2} = \frac{23}{12}\tau F^{n+2} - \frac{16}{2}\tau F^{n+1} + \frac{5}{12}\tau F^{n}$$

Backward Differentiation (BDF)

$\beta_k = 1, \quad \beta_j = 0$ otherwise

$$\left.\frac{3}{2} w^{n+2} - 2 w^{n+1} + \frac{1}{2} w^{n} = \tau F^{n+2} \right\} \begin{array}{l}\text{2-step}\\ \text{BDF}\end{array}$$

(implicit)

These have very good stability properties for stiff problems

The best approach often is to use a <u>mixed implicit-explicit</u> approach, e.g.

IMEX RK
(implicit-explicit Runge-Kutta)

or <u>splitting approaches</u>

For example:

$$u_t + a u_x = d u_{xx}$$

$$\boxed{u_t + g(u) = Lu}$$

(L - linear, g - nonlinear or difficult due to linear algebra)

Treat $g(u)$ using Adams-Bashforth-2 but Lu using implicit midpoint (Crank-Nicolson):

$$\frac{u^{n+1} - u^n}{\Delta t} + \underbrace{\left[\frac{3}{2}\tau g^n - \frac{1}{2}\tau g^{n-1}\right]}_{\text{advection}} = \underbrace{L\,\frac{u^n + u^{n+1}}{2}}_{\substack{\text{diffusion}\\ \text{viscosity}}}$$

Doing this for higher-order methods is tricky.

Monotonicity

Monotone ODE system
$$\begin{cases} w'(t) = A w(t) \\ a_{ij} \geq 0 \quad \text{for } i \neq j \\ a_{ii} \geq -\alpha, \quad \alpha > 0 \\ A \text{ has no eigenvalue on positive real axes} \end{cases}$$

Take a linear <u>one-step</u> method

$$w^{n+1} = \underbrace{R(\tau A)}_{\text{stability function}} w^n$$

(see old lectures)

Theorem

$$\left\{\begin{array}{l} R(\tau A) \geq 0 \ \text{iff} \\ \alpha\tau \leq \gamma_R \end{array}\right.$$

where γ_R is the largest γ for which $R(\cancel{x})$ and all of its derivatives are positive on $[-\gamma, 0]$

So here γ_R is a ~~▒~~ monotonicity limit on the timestep, similar to stability limits but distinct.

$\boxed{\text{Theorem: Any unconditionally positive method } (\gamma_R = \infty) \text{ has order } 1}$ yet another order barrier

So one cannot do better than Backward Euler if we want to take very large time steps and preserve non-negativity!

Often we can construct spatial discretizations such that <u>forward Euler is monotone under a condition</u> $\alpha \tau \leq 1$ ($\gamma_k = 1$).

This knowledge can then be used to construct higher-order methods that <u>preserve</u> this property

Nonlinear Positivity

$$w' = F(t, w)$$

Assume: forward Euler is positive:

$$\begin{cases} v + \tau F(\tau, v) \geq 0 \\ \text{if } v \geq 0 \text{ and } \alpha\tau \leq 1 \end{cases}$$

then one can prove that diagonally-implicit RK methods where the final update is a convex combination of forward and backward Euler steps is also positive if $\boxed{\alpha\tau \leq S}$

wher $S = O(1)$ depends on method

Such RK methods are called Strong - Stability Preserving (SSP) methods (Osher & Shu 1988)

(generalites to TVD or in fact any convex functional)

E.g.

Explicit Trapezoidal Rule (RK2)

$$\begin{cases} w^{*} = w^{n} + \tau F^{n} & \text{(first Euler)} \\ w^{**} = w^{*} + \tau F^{*} & \text{(second Euler)} \\ w^{n+1} = \frac{1}{2}(w^{n} + w^{**}) & \text{(convex combo)} \\ \quad = w^{n} + \frac{1}{2}(\tau F^{n} + \tau F^{*}) & \end{cases}$$

RK3 $\left\{\begin{matrix}\text{TVD}\\ \text{SSP}\end{matrix}\right\}$ scheme (explicit)

$$
\begin{cases}
w^{*} = w^{n} + \tau F^{n} & \text{(first Euler)} \\[2ex]
w^{**} = \dfrac{3}{4} w^{n} + \dfrac{1}{4}\Big[\underbrace{w^{*} + \tau F^{*}}_{\text{second Euler}}\Big] & \text{(convex combo)} \\[2ex]
w^{n+1} = \dfrac{1}{3} w^{n} + \dfrac{2}{3}\Big[\underbrace{w^{**} + \tau F^{**}}_{\text{third Euler}}\Big] & \text{(convex combo)}
\end{cases}
$$

this scheme is SSP, and its stability function includes a portion of the imaginary axes. Also third-order. Great for advection!