MA4255 Numerical Methods in Differential Equations

Chapter 4: Stiff problems

4.1 Stability of numerical methods for stiff systems

4.2 Backward differentiation methods for stiff systems

4.3 Adaptivity for stiff problems

4.1 Stability of numerical methods for stiff systems

Motivation

For $A \in \mathbb{C}^{m\times m}$, consider the IVP (system of m ODEs)

$$\mathbf{y}'(x) = A\mathbf{y}(x), \qquad \mathbf{y}(x_0) = \mathbf{y}_0, \qquad [\mathbf{y}(x) = (y_1(x), \ldots, y_m(x))^{\mathrm{T}}]$$

Apply linear k-step method $\sum_{j=0}^{k} \alpha_j \mathbf{y}_{n+j} = h \sum_{j=0}^{k} \beta_j \mathbf{f}_{n+j}$ to this IVP:

$$\sum_{j=0}^{k} (\alpha_j I_m - h\beta_j A)\mathbf{y}_{n+j} = \mathbf{0}.$$

Suppose the eig.vals $\lambda_1, \ldots, \lambda_m \in \mathbb{C}$ of A are distinct. Then, $\exists H \in \mathbb{C}^{m\times m}$ invertible s.t. $H^{-1}AH = \mathrm{diag}(\lambda_1, \ldots, \lambda_m) =: \Lambda$. Define $\mathbf{z}_{n+j} := H^{-1}\mathbf{y}_{n+j}$ for $j \in \{0, \ldots, k\}$. Then,

$$\sum_{j=0}^{k} (\alpha_j I_m - h\beta_j \Lambda)\mathbf{z}_{n+j} = H^{-1} \sum_{j=0}^{k} (\alpha_j I_m - h\beta_j A)\mathbf{y}_{n+j} = \mathbf{0}.$$

$\Longrightarrow \sum_{j=0}^{k} (\alpha_j - \lambda_i h\beta_j) z_{n+j,i} = 0$ for $i \in \{1, \ldots, m\}$. Each of these m eqns completely decoupled from others. Thus, we are in framework of Ch.3 (LMMs for a single ODE). New feature: $\bar{h} := \lambda_i h \in \mathbb{C}$.

Region of absolute stability

Definition (Region of absolute stability)

A linear k-step method is said to be **absolutely stable** in an open set $\mathscr{R}_A \subseteq \mathbb{C}$ if, for all $\bar{h} \in \mathscr{R}_A$, all roots r_s, $s \in \{1, \dots, k\}$, of the stability polynomial $z \mapsto \pi(z; \bar{h})$ associated with the method satisfy $|r_s| < 1$. The largest such $\mathscr{R}_A$ is called the **region of absolute stability** of the method.

Rk: interval of absolute stability $\subseteq$ region of absolute stability.

Example: explicit Euler $y_{n+1} - y_n = h f_n$. We have

$$\pi(z; \bar{h}) = \rho(z) - \bar{h}\sigma(z) = (z - 1) - \bar{h} = z - (1 + \bar{h}).$$

This has the unique root $r_1 := 1 + \bar{h}$. Note $|r_1| < 1$ iff $\mathrm{dist}(\bar{h}, -1) < 1$ iff $\bar{h} \in D_1(-1)$. Hence, the region of absolute stability of explicit Euler is

$$\mathscr{R}_A = D_1(-1).$$

Rk: What to do if $\pi(z; \bar{h})$ is more complicated? $\Longrightarrow$ Schur criterion.

Example of a stiff problem

For $\lambda \in \mathbb{C}^-$ with $|\lambda| \gg 1$, consider the problem

$$y''(x) + (1-\lambda)y'(x) - \lambda y(x) = 0, \qquad y(0) = 1, \quad y'(0) = -\lambda - 2.$$

Writing $\mathbf{y}(x) = (y(x), y'(x))^{\mathrm{T}}$, we can rewrite the problem as

$$\mathbf{y}'(x) = \begin{pmatrix} 0 & 1 \\ \lambda & \lambda - 1 \end{pmatrix} \mathbf{y}(x) =: A\mathbf{y}(x), \qquad \mathbf{y}(0) = \begin{pmatrix} 1 \\ -\lambda - 2 \end{pmatrix} =: \mathbf{y}_0.$$

Note that the true solution satisfies

$$\mathbf{y}(x) = \begin{pmatrix} 2e^{-x} - e^{\lambda x} \\ -2e^{-x} - \lambda e^{\lambda x} \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{as} \quad x \to \infty.$$

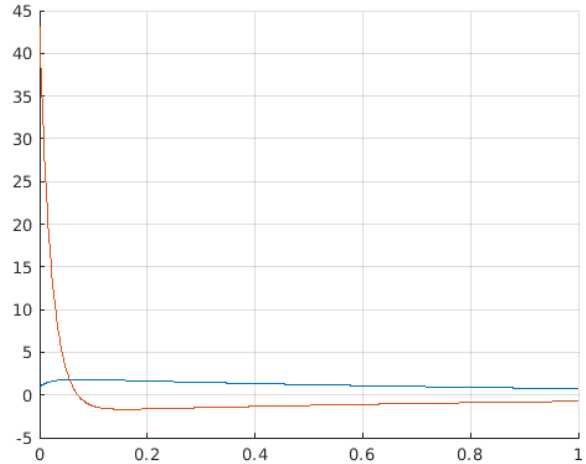
Consider explicit Euler: $\mathbf{y}_{n+1} = \mathbf{y}_n + hA\mathbf{y}_n$, i.e.,

$$\mathbf{y}_n = (I_2 + hA)^n \mathbf{y}_0.$$

Note $\lim_{n\to\infty} \mathbf{y}_n = \mathbf{0}$ iff eigvals of $I_2 + hA$ are in $D_1(0)$, i.e., iff

$$|1 - h| < 1, \qquad |1 + \lambda h| < 1,$$

i.e., iff $h \in (0, 2)$ and $\bar{h} = \lambda h \in D_1(-1)$. $\Longrightarrow$ h must be very small!



$y(x) = 2e^{-x} - e^{\lambda x}$ and $y'(x) = -2e^{-x} - \lambda e^{\lambda x}$ for $\lambda = -45$.

$\Longrightarrow$ The y' component of the soln $\mathbf{y} = (y, y')$ varies rapidly near $x = 0$ (we say that the fct has a thin layer at $x = 0$).

In order to ensure the stability of explicit Euler, h is forced to be exceedingly small, $h < -2\frac{\mathrm{Re}\,(\lambda)}{|\lambda|^2}$ (since $|1 + \lambda h| < 1$), smaller than an accurate approximation of the solution for $x \gg 1/|\lambda|$ would necessitate.

Systems of ODEs which exhibit this behaviour are called **stiff systems**.

Rk: Stiffness lacks a rigorous definition. Here is a historic "definition": stiff eqns are eqns where implicit Euler works much better than explicit Euler.

(For stiff systems, stability of eE requires h very small, much smaller than required by accuracy.)

A-stability

Definition (A-stability of LMMs)

A LMM is called A-stable if its region of absolute stability $\mathcal{R}_A$ is s.t.

$$\mathbb{C}^- \subseteq \mathcal{R}_A.$$

Ex.: implicit Euler is A-stable.

Pf: $\pi(z;\bar{h}) = \rho(z) - \bar{h}\sigma(z) = (1-\bar{h})z - 1$. If $\bar{h} \neq 1$, there is a unique root at $r_1 := \frac{1}{1-\bar{h}}$. Note $|r_1| < 1$ iff $|1-\bar{h}| > 1$, i.e., iff $\bar{h} \in \mathbb{C}\backslash\bar{D}_1(1)$. We find that $\mathcal{R}_A = \mathbb{C}\backslash\bar{D}_1(1) \supseteq \mathbb{C}^-$. □

Unfortunately, A-stability is very restrictive:

Theorem

(i) *No explicit LMM is A-stable.*

(ii) *The order of accuracy of an A-stable implicit LMM cannot exceed 2.*

(iii) *The second-order accurate A-stable LMM with smallest error constant is the trapezium rule method.*

Relaxing A-stability: $A(\alpha)$-stability

Definition ($A(\alpha)$-stability of LMMs)

For $\alpha \in (0, \frac{\pi}{2})$, a LMM is called $A(\alpha)$-**stable**, if its region of absolute stability $\mathcal{R}_A$ is s.t.

$$W_\alpha := \{\bar{h} \in \mathbb{C} : \arg(\bar{h}) \in (\pi - \alpha, \pi + \alpha)\} \subseteq \mathcal{R}_A.$$

A LMM is called $A(0)$-**stable** if it is $A(\alpha)$-stable for some $\alpha \in (0, \frac{\pi}{2})$.
A LMM is called A_0-stable if $\mathcal{R}_A$ includes the negative real axis.

Rk: for given $\lambda \in \mathbb{C}^-$, $\bar{h} = \lambda h$ either lies inside the wedge W_α or outside W_α for all $h > 0$.

Consequently, for the IVP $\mathbf{y}'(x) = A\mathbf{y}(x)$, $\mathbf{y}(x_0) = \mathbf{y}_0$, if all eigenvalues λ of A belong to W_α then an $A(\alpha)$-stable method can be used for the numerical solution of the IVP without any restrictions on h.

In particular, if all eigenvalues of A are real and negative, then an $A(0)$-stable method can be used.

Definition ($A(\alpha)$-stability of LMMs)

For $\alpha \in (0, \frac{\pi}{2})$, a LMM is called $A(\alpha)$-**stable**, if its region of absolute stability $\mathscr{R}_A$ is s.t.

$$W_\alpha := \{\bar{h} \in \mathbb{C} : \ arg(\bar{h}) \in (\pi - \alpha, \pi + \alpha)\} \subseteq \mathscr{R}_A.$$

A LMM is called $A(0)$-**stable** if it is $A(\alpha)$-stable for some $\alpha \in (0, \frac{\pi}{2})$. A LMM is called A_0 stable if $\mathscr{R}_A$ includes the negative real axis.

Theorem

(i) *No explicit LMM is $A(0)$-stable.*

(ii) *The only $A(0)$-stable linear k-step method whose order exceeds k is the trapezium rule method.*

(iii) *For each $\alpha \in [0, \frac{\pi}{2})$ there exist $A(\alpha)$-stable linear k-step methods of order p for which $k = p = 3$ and $k = p = 4$.*

A further stability concept: Stiff-stability

Motivation: for a typical stiff problem, the eigvals of A which produce the fast transition all lie to the left of a line $\{\bar{h} \in \mathbb{C} : \mathrm{Re}\,(\bar{h}) = -a\}$, $a > 0$, and those responsible for the slow transitions are clustered around 0.

Definition (Stiffly stable LMMs)

A LMM is called **stiffly stable** if $\exists a, c > 0$ s.t. its region of absolute stability $\mathscr{R}_A$ is such that $\mathscr{R}_1 \cup \mathscr{R}_2 \subseteq \mathscr{R}_A$ where

$$\mathscr{R}_1 = \{\bar{h} \in \mathbb{C} \,:\, \mathrm{Re}\,(\bar{h}) \in (-\infty, -a)\},$$
$$\mathscr{R}_2 = \{\bar{h} \in \mathbb{C} \,:\, \mathrm{Re}\,(\bar{h}) \in [-a, 0), \;\; \mathrm{Im}\,(\bar{h}) \in [-c, c]\}.$$

We have the following chain of implications:

$$A\text{-stab.} \Rightarrow \text{stiff-stab.} \Rightarrow A(\alpha)\text{-stab.} \Rightarrow A(0)\text{-stab.} \Rightarrow A_0\text{-stab.}$$

4.2 Backward differentiation methods for stiff systems

BDF methods for stiff systems

Consider a LMM with stability polynomial $\pi(z;\bar{h}) = \rho(z) - \bar{h}\sigma(z)$. If the method is $A(\alpha)$-stable or stiffly stable, the roots $r(\bar{h})$ of $z \mapsto \pi(z;\bar{h})$ lie in $D_1(0)$ when $\bar{h}$ is real and $\bar{h} \to -\infty$. Then,

$$0 = \lim_{\bar{h}\to-\infty} \frac{\rho(r(\bar{h}))}{\bar{h}} = \lim_{\bar{h}\to-\infty} \sigma(r(\bar{h})) = \sigma\left(\lim_{\bar{h}\to-\infty} r(\bar{h})\right).$$

$\implies$ the roots of $z \mapsto \pi(z;\bar{h})$ approach those of σ. Thus, it is natural to choose σ in such a way that its roots lie within $D_1(0)$.

A particularly simple choice would be to take $\sigma(z) = \beta_k z^k$; the resulting class of **backward differentiation formulae (BDF)** has the general form:

$$\sum_{j=0}^{k} \alpha_j \mathbf{y}_{n+j} = h\beta_k \mathbf{f}_{n+k},$$

where $\alpha_0, \ldots, \alpha_k$, β_k are given in the following table for $k \in \{1, \ldots, 6\}$ (also displaying a from the defn of stiff-stability, α from the defn of $A(\alpha)$-stability, the order p, and the error constant C_{p+1}).
Rk: BDF methods with $k > 6$ are not zero-stable.

List of BDF methods

$$\sum_{j=0}^{k} \alpha_j \mathbf{y}_{n+j} = h\beta_k \mathbf{f}_{n+k}$$

k	α_6	α_5	α_4	α_3	α_2	α_1	α_0	β_k	p	C_{p+1}	a_{min}	α_{max}
1						1	-1	1	1	$-\frac{1}{2}$	0	90^o
2					1	$-\frac{4}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	2	$-\frac{2}{9}$	0	90^o
3				1	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$	$\frac{6}{11}$	3	$-\frac{3}{22}$	0.1	88^o
4			1	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$	$\frac{12}{25}$	4	$-\frac{12}{125}$	0.7	73^o
5		1	$-\frac{300}{137}$	$\frac{300}{137}$	$-\frac{200}{137}$	$\frac{75}{137}$	$-\frac{12}{137}$	$\frac{60}{137}$	5	$-\frac{10}{137}$	2.4	52^o
6	1	$-\frac{360}{147}$	$\frac{450}{147}$	$-\frac{400}{147}$	$\frac{225}{147}$	$-\frac{72}{147}$	$\frac{10}{147}$	$\frac{60}{147}$	6	$-\frac{20}{343}$	6.1	19^o

4.3 Adaptivity for stiff problems

Motivation

Ideally, we would like to compute an approximate solution of the following IVP for a system of first-order ODEs:

$$\mathbf{y}'(x) = \mathbf{f}(x, \mathbf{y}(x)), \qquad \mathbf{y}(x_0) = \mathbf{y}_0,$$

for $x \in [x_0, X_M]$, and make sure that this approximation is accurate up to a certain (absolute/relative) precision.

In addition, we would like to achieve such a precision in the fastest/cheapest way possible. How should this be done?

$\Longrightarrow$ We present two attempts; the first being conceptually simpler, the second being the preferred one in practice.

Attempt 1

A simple strategy could be to:

1. choose a one-step method of order p;
2. choose $N \in \mathbb{N}$ and compute approx. soln $\{\mathbf{y}_n\}_{n=0}^{N}$ with $h = \frac{X_M - x_0}{N}$;
3. choose a large natural number $\tilde{N} \in \mathbb{N}$ with $\tilde{N} > N$ and compute approx. soln $\{\tilde{\mathbf{y}}_n\}_{n=0}^{\tilde{N}}$ with $\tilde{h} = \frac{X_M - x_0}{\tilde{N}}$.

Idea: use $\|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\|$ to estimate the error $\|\mathbf{y}(X_M) - \mathbf{y}_N\|$.
$\Longrightarrow$ If $\|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\| < \texttt{TOL}$, stop. Otherwise,

1. increase N so that $N > \tilde{N}$;
2. compute the approximate solution $\{\mathbf{y}_n\}_{n=0}^{N}$ using $h = \frac{X_M - x_0}{N}$;
3. check whether $\|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\| < \texttt{TOL}$.

If $\|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\| < \texttt{TOL}$, then stop. Otherwise, select $\tilde{N} > N$, compute $\{\tilde{\mathbf{y}}_n\}_{n=0}^{\tilde{N}}$ using $\tilde{h} = \frac{X_M - x_0}{\tilde{N}}$, and check whether $\|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\| < \texttt{TOL}$.
Repeat until convergence ...

Why is this a sensible strategy?

Why can we use $\|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\|$ to estimate $\|\mathbf{y}(X_M) - \mathbf{y}_N\|$?

Assume $\tilde{N} > N$, and set $\alpha := \frac{\tilde{h}}{h} = \frac{N}{\tilde{N}} < 1$. For h sufficiently small, have

$$\|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\| \leq \|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}(X_M)\| + \|\mathbf{y}(X_M) - \mathbf{y}_N\| \leq C(\tilde{h}^p + h^p) = (1+\alpha^p)Ch^p$$

for some constant $C > 0$, and thus,

$$\begin{aligned} \|\mathbf{y}(X_M) - \mathbf{y}_N\| &\leq \|\mathbf{y}(X_M) - \tilde{\mathbf{y}}_{\tilde{N}}\| + \|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\| \\ &\leq C\tilde{h}^p \qquad\qquad + (1+\alpha^p)Ch^p \\ &= \alpha^p Ch^p \qquad\qquad + (1+\alpha^p)Ch^p. \end{aligned}$$

For $\alpha < 1$, $\alpha^p \ll 1 + \alpha^p$ (in relative terms).
$\Longrightarrow$ $\|\mathbf{y}(X_M) - \tilde{\mathbf{y}}_{\tilde{N}}\|$ has a minor contribution.
$\Longrightarrow$ $\|\tilde{\mathbf{y}}_{\tilde{N}} - \mathbf{y}_N\|$ may be used to estimate $\|\mathbf{y}(X_M) - \mathbf{y}_N\|$.

Drawbacks of this strategy

This strategy could deliver an accurate solution, but it is computationally inefficient, because whenever the target tolerance is not met, we need to compute another solution from scratch on a finer computational mesh over the entire interval $[x_0, X_M]$.

(I.e., a global mesh-refinement needs to be performed – a new numerical approximation has to be computed on a globally refined mesh).

Attempt 2

Idea: Control consistency error (c.e.) for each mesh point. Recall: global error bounded by the maximum of the c.e. up to constant factor.

$\Longrightarrow$ We hope we can compute a sufficiently accurate soln by choosing a suitable h or, even better, by adapting the step size locally, i.e., selecting a suitable h_n for every x_n to control the c.e. locally. To estimate the c.e. at $x = x_n$, in addition to the 1-step (for simplicity) method

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\boldsymbol{\Phi}(x_n, \mathbf{y}_n; h) =: \boldsymbol{\Psi}(x_n, \mathbf{y}_n; h)$$

of order p being used, consider an additional 1-step method

$$\tilde{\mathbf{y}}_{n+1} = \tilde{\mathbf{y}}_n + h\tilde{\boldsymbol{\Phi}}(x_n, \tilde{\mathbf{y}}_n; h) =: \tilde{\boldsymbol{\Psi}}(x_n, \tilde{\mathbf{y}}_n; h)$$

of order $\tilde{p}$, with $\tilde{p} > p$, and compute

$$\mathrm{ERR}(x_n; h) := \|\tilde{\boldsymbol{\Psi}}(x_n, \mathbf{y}_n; h) - \boldsymbol{\Psi}(x_n, \mathbf{y}_n; h)\|.$$

Recall:

$$\mathrm{ERR}(x_n; h) := \|\tilde{\boldsymbol{\Psi}}(x_n, \mathbf{y}_n; h) - \boldsymbol{\Psi}(x_n, \mathbf{y}_n; h)\|.$$

The idea behind using this to estimate the c.e. T_n is that, if the error has been controlled from x_0 up until x_n, for some $n \geq 1$, then the difference between $\mathbf{y}(x_n)$ and $\mathbf{y}_n$ is "negligible", and therefore $\mathbf{y}_n$ can be assumed to be equal to $\tilde{\mathbf{y}}_n$ (both being "equal" to $\mathbf{y}(x_n)$). Hence,

$$\begin{aligned} hT_n &= \mathbf{y}(x_{n+1}) - \boldsymbol{\Psi}(x_n, \mathbf{y}(x_n); h) \\ &= \mathbf{y}(x_{n+1}) - \tilde{\boldsymbol{\Psi}}(x_n, \mathbf{y}(x_n); h) + \tilde{\boldsymbol{\Psi}}(x_n, \mathbf{y}(x_n); h) - \boldsymbol{\Psi}(x_n, \mathbf{y}(x_n); h) \\ &\approx \mathbf{y}(x_{n+1}) - \tilde{\boldsymbol{\Psi}}(x_n, \mathbf{y}(x_n); h) + \tilde{\boldsymbol{\Psi}}(x_n, \mathbf{y}_n; h) - \boldsymbol{\Psi}(x_n, \mathbf{y}_n; h) \\ &\approx Ch^{\tilde{p}+1} \qquad\qquad\qquad\qquad + \tilde{\boldsymbol{\Psi}}(x_n, \mathbf{y}_n; h) - \boldsymbol{\Psi}(x_n, \mathbf{y}_n; h). \end{aligned}$$

Since $hT_n = \mathcal{O}(h^{p+1})$ and $\tilde{p} > p$, it follows that the term $\approx Ch^{\tilde{p}+1}$ on the right-hand side is "negligible":

$$hT_n \approx \tilde{\boldsymbol{\Psi}}(x_n, \mathbf{y}_n; h) - \boldsymbol{\Psi}(x_n, \mathbf{y}_n; h).$$

Locally adaptive strategy

The strategy is as follows: at every x_n,

1. select an initial local step size h_n;
2. compute $\mathrm{ERR}(x_n; h_n) = \|\tilde{\mathbf{\Psi}}(x_n, \mathbf{y}_n; h_n) - \mathbf{\Psi}(x_n, \mathbf{y}_n; h_n)\|$;
3. if $\mathrm{ERR}(x_n; h_n) < \mathtt{TOL}$, set $\mathbf{y}_{n+1} = \mathbf{\Psi}(x_n, \mathbf{y}_n; h_n)$; otherwise, choose a smaller h_n and go to step 2.

For more efficiency: increase the step h_n every time this step has been accepted, that is, to select βh_n for a suitable $\beta > 1$.

Rk: Let $\mathtt{TOL} > 0$ be an absolute error tolerance and $\mathrm{ERR}(x_n; h_n) < \mathtt{TOL}$. Then, the "optimal" β is

$$\beta = \beta_n = \left(\frac{\mathtt{TOL}}{\mathrm{ERR}(x_n; h_n)} \right)^{\frac{1}{p+1}}.$$

Why? Let β_n s.t. $\mathrm{ERR}(x_n, \beta_n h_n) = \mathtt{TOL}$, i.e., $\beta_n h_n$ ideal step size. Then,

$$\mathtt{TOL} = \mathrm{ERR}(x_n; \beta_n h_n) \approx C(\beta_n h_n)^{p+1} = \beta_n^{p+1} C h_n^{p+1} \approx \beta_n^{p+1} \mathrm{ERR}(x_n; h_n).$$

Embedded RK methods

Improve efficiency of adaptive algorithm by using embedded RK methods:

Definition (Embedded RK methods)

Two RK methods are **embedded** if they use the same stages. The Butcher tableau of two embedded RK methods can be written as

$$\begin{array}{c|c} \mathbf{a} & \mathbf{B} \\ \hline & \begin{matrix} \mathbf{c}_2^{\mathrm{T}} \\ \mathbf{c}_1^{\mathrm{T}} \end{matrix} \end{array}, \quad \text{where} \quad \begin{array}{c|c} \mathbf{a} & \mathbf{B} \\ \hline & \mathbf{c}_2^{\mathrm{T}} \end{array} \quad \text{and} \quad \begin{array}{c|c} \mathbf{a} & \mathbf{B} \\ \hline & \mathbf{c}_1^{\mathrm{T}} \end{array}$$

are the Butcher tableaus of the two RK methods, respectively.

Ex.: The **Heun–Euler method** has the Butcher tableau

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \\ & 1 & 0 \end{array}, \quad \text{where} \quad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array} \quad \text{and} \quad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1 & 0 \end{array}$$

are the Butcher tableaus of Heun's method and explicit Euler, respectively.

End of “Chapter 4: Stiff problems”.