Numerical Methods in Engineering and Applied Science

Lecture 10. Initial-value problems for systems of ODEs.

Multi-step methods.

We consider the following model system of equations:

$$\mathbf{u}'(t) = A\mathbf{u}(t), \tag{1}$$

where $\mathbf{u}(t) \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m\times m}$. Let us suppose that A is diagonalizable, $V^{-1}AV = \Lambda$, where Λ is a diagonal matrix. We define $\mathbf{u}(t) = V\mathbf{y}(t)$ and we see that $\mathbf{y}(t)$ satisfies

$$\mathbf{y}'(t) = \Lambda\mathbf{y}(t) \tag{2}$$

where each line is a scalar equation of type $y(t) = \lambda y(t)$. Since $\mathbf{y}(t)$ and $\mathbf{u}(t)$ are related by a linear transform V, $\mathbf{u}$ is bounded in the limit of $t \to \infty$ if $\mathbf{y}$ is bounded.

Theorem. Let A be a diagonalizable matrix with eigenvalues λ_1,...,λ_m. The solutions of $\mathbf{u}'(t) = A\mathbf{u}(t)$ converge to zero in limit $t \to \infty$ for all initial conditions iff $\Re(\lambda_j) < 0$ for all $j = 1, 2, ..., m$.

Consider a two-step method applied to $\mathbf{u}'(t) = A\mathbf{u}(t)$,

$$\mathbf{u}_{n+1} + \alpha_1 \mathbf{u}_n + \alpha_0 \mathbf{u}_{n-1} = hA\left(\beta_2 \mathbf{u}_{n+1} + \beta_1 \mathbf{u}_n + \beta_0 \mathbf{u}_{n-1}\right). \tag{3}$$

We multiply from the left by V^{-1} and we get

$$\begin{aligned} V^{-1}\mathbf{u}_{n+1} + \alpha_1 V^{-1}\mathbf{u}_n + \alpha_0 V^{-1}\mathbf{u}_{n-1} \\ = hV^{-1}AV\left(\beta_2 V^{-1}\mathbf{u}_{n+1} + \beta_1 V^{-1}\mathbf{u}_n + \beta_0 V^{-1}\mathbf{u}_{n-1}\right). \end{aligned} \tag{4}$$

We define $\mathbf{u}_j = V\mathbf{y}_j$ to obtain

$$\mathbf{y}_{n+1} + \alpha_1 \mathbf{y}_n + \alpha_0 \mathbf{y}_{n-1} = h\Lambda\left(\beta_2 \mathbf{y}_{n+1} + \beta_1 \mathbf{y}_n + \beta_0 \mathbf{y}_{n-1}\right). \tag{5}$$

Note that the same formula (5) is obtained if we apply the multi-step method to the diagonalized system (2): the diagonalization and the multi-step method commute.

A multi-step method is *absolutely stable* for a diagonalizable system $\mathbf{u}'(t) = A\mathbf{u}(t)$ if $\lambda h \in \mathcal{A}$ (the stability region) for any eigenvalue λ of A.

Stiff problems have the following symptoms:

- The problem has two or more significantly different time scales;
- The stability constraint on the time step is more severe that the precision constraint;
- Implicit methods solve the problem more effectively than explicit methods.

In stiff problems, all eigenvalues have negative real part, but the ratio

$$\frac{\max_j(-\Re(\lambda_j))}{\min_j(-\Re(\lambda_j))} \tag{6}$$

can be very large.

If the problem is non-linear, it can be linearized near a base state $\bar{\boldsymbol{u}}(t)$ to determine if it is stiff. We decompose the solution into

$$\boldsymbol{u}(t) = \bar{\boldsymbol{u}}(t) + \boldsymbol{\varepsilon}(t), \tag{7}$$

where $\boldsymbol{\varepsilon}(t)$ is a small perturbation from the base state. We linearize $\boldsymbol{f}$:

$$\boldsymbol{f}(t, \boldsymbol{u}) = \boldsymbol{f}(t, \bar{\boldsymbol{u}}) + J(t)\boldsymbol{\varepsilon}(t) + o(||\boldsymbol{\varepsilon}||), \tag{8}$$

where $J(t)$ is the Jacobian. If we neglect $o(||\boldsymbol{\varepsilon}||)$, the equation for $\boldsymbol{u}$ becomes $\boldsymbol{u}' = \boldsymbol{f}(t, \bar{\boldsymbol{u}}) + J(t)\boldsymbol{\varepsilon}$ and, since $\bar{\boldsymbol{u}}' = f(t, \bar{\boldsymbol{u}})$, we obtain

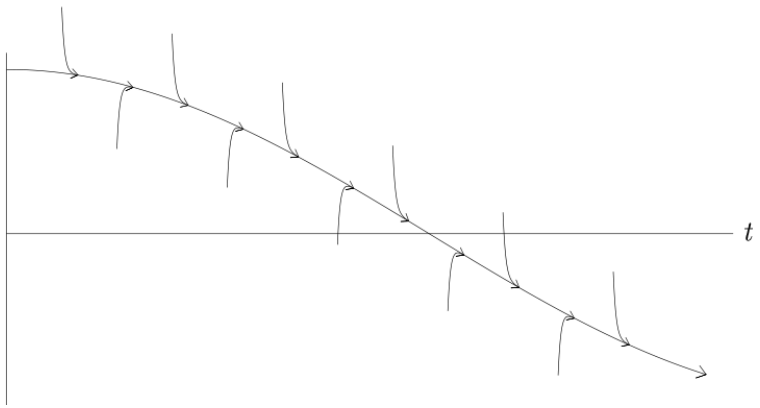
$$\boldsymbol{\varepsilon}' = J(t)\boldsymbol{\varepsilon}(t). \tag{9}$$

We fix the coefficients of J at $t = t^*$ of interest, and diagonalize the matrix $J(t^*)$.
To numerically calculate $\boldsymbol{u}(t)$ à $t \approx t^*$, h must be small enough that $h\lambda$ is inside the stability region of the numerical method for all λ, eigenvalues of $J(t^*)$.

Example. The IVP

$$\begin{cases} u' = -100 \sin(u - \cos t) - \sin t, \\ u(0) = 1 \end{cases} \tag{10}$$

has the solution $u(t) = \cos t$. It attracts neighboring solutions. We see the fast transient at a time scale ≈ 0.01. The linearized equation is $\varepsilon' = -100\varepsilon$.



Example.

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} -100 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}, \quad t \in [0,1], \quad \begin{pmatrix} u(0) \\ v(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

In fact, the two equations are decoupled. Let's say that the approximate solution is sufficiently accurate if $|u(1) - e^{-100}| < \epsilon$ and $|v(1) - e^{-1}| < \epsilon$. Suppose the numerical method is of order p. To obtain the desired precision for $v(1)$, the time step $h = \mathcal{O}(\epsilon^{1/p})$ (e.g. $h \approx 0.1$ if $\epsilon \approx 10^{-4}$ and $p = 4$). But if we use an explicit method, the time step is limited by a value $\approx 10^{-2}$ for the reason of stability of the solution of the first équation.

Example.

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} -5 & 6 \\ 4 & -5 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}, \quad t \in [0,1], \quad \begin{pmatrix} u(0) \\ v(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

The eigenvalues are, approximately, $\lambda_1 = -9.9$ et $\lambda_2 = -0.1$.

For the solution of stiff problems, it is useful to use A-stable methods. They are implicit. We will therefore consider methods of solution of the system of algebraic equations associated with implicit methods.
We rewrite the implicit multi-step method as follows:

$$\boldsymbol{u} = h\beta_s \boldsymbol{f}(t_{n+1}, \boldsymbol{u}) + \boldsymbol{g}_n, \tag{11}$$

where

$$\boldsymbol{g}_n = h\left(\beta_{s-1}\boldsymbol{f}_n + ... + \beta_0 \boldsymbol{f}_{n-s+1}\right) - \left(\alpha_{s-1}\boldsymbol{u}_n + ... + \alpha_0 \boldsymbol{u}_{n-s+1}\right) \tag{12}$$

contains only known values and where $\boldsymbol{u}$ is the unknown. Equation (11) has only one solution if $h = 0$ but it can have several solutions if $h > 0$. It is reasonable to choose the one that is close to $\boldsymbol{u}_n$.

Predictor-corrector methods. One can use an explicit method to obtain a 'prediction' $\hat{\boldsymbol{u}}$ and substitute it on the right side of (11). It turns out that these methods are neither A-stable nor A_0-stable, since they are explicit.

Fixed point iteration method.

$$\boldsymbol{u}^{[l+1]} = h\beta_s \boldsymbol{f}(t_{n+1}, \boldsymbol{u}^{[l]}) + \boldsymbol{g}_n, \quad l = 0, 1, 2, ... \tag{13}$$

- A good initial approximation can help reduce the number of iterations. An easy choice is $\boldsymbol{u}^{[0]} = \boldsymbol{u}_n$ but we can improve it if we use an explicit scheme as 'predictor'.
- To analyze the convergence of $\boldsymbol{u}^{[l]}$, we note that $\boldsymbol{u}^{[l]} = \boldsymbol{u}_{n+1} + \boldsymbol{E}^{[l]}$, J is the Jacobian of $\boldsymbol{f}$, and we expand $\boldsymbol{f}$,

  $$\boldsymbol{f}(t_{n+1}, \boldsymbol{u}^{[l]}) = \boldsymbol{f}(t_{n+1}, \boldsymbol{u}_{n+1}) + J\boldsymbol{E}^{[l]} + o(||\boldsymbol{E}^{[l]}||). \tag{14}$$

  Using (13) and (11) we obtain $\boldsymbol{E}^{[l+1]} \approx h\beta_s J\boldsymbol{E}^{[l]}$. If λ is an eigenvalue of J and $\boldsymbol{v}$ is its corresponding eigenvector, we have $\boldsymbol{E}^{[l+1]} \approx (h\beta_s\lambda)^l \boldsymbol{v}$. We obtain the condition

  $$h|\beta_s\lambda| < 1 \tag{15}$$

  similar to the condition for explicit methods. The fixed point method is not suitable for the solution of stiff problems.

Newton–Raphson method.

We solve

$$\boldsymbol{F}(\boldsymbol{u}) = \boldsymbol{0} \quad \text{où} \quad \boldsymbol{F}(\boldsymbol{u}) = \boldsymbol{u} - h\beta_s \boldsymbol{f}(t_{n+1}, \boldsymbol{u}) - \boldsymbol{g}_n. \tag{16}$$

Let $\boldsymbol{u}^{[l]} = \boldsymbol{u}_{n+1} + \boldsymbol{E}^{[l]}$. The Taylor series expansion $\boldsymbol{F}(\boldsymbol{u}_{n+1}) = \boldsymbol{0}$ gives a linear system of equations for $\hat{\boldsymbol{E}}^{[l]}$ (approximation of $\boldsymbol{E}^{[l]}$),

$$\boldsymbol{F}(\boldsymbol{u}^{[l]}) - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}(\boldsymbol{u}^{[l]}) \hat{\boldsymbol{E}}^{[l]} = \boldsymbol{0}. \tag{17}$$

We obtain a new approximation to the solution,

$$\boldsymbol{u}^{[l+1]} = \boldsymbol{u}^{[l]} - \hat{\boldsymbol{E}}^{[l]}. \tag{18}$$

The rate of convergence is quadratic. Moreover, this method preserves the stability properties of the implicit scheme.

The classical Runge–Kutta methods are explicit, but *implicit Runge–Kutta methods* are also used. We can write these methods in the form

$$\begin{aligned} \boldsymbol{g}_i &= \boldsymbol{u}_n + h\sum_{j=1}^{s} a_{ij}\boldsymbol{f}(t_n + c_j h, \boldsymbol{g}_j), \quad i = 1, ..., s, \\ \boldsymbol{u}_{n+1} &= \boldsymbol{u}_n + h\sum_{j=1}^{s} b_j \boldsymbol{f}(t_n + c_j h, \boldsymbol{g}_j). \end{aligned} \tag{19}$$

An implicit RK method of rank s can be of order $2s$ at most, but we have to solve a system of sm, where m is the number of differential equations in the system. This method applied to $\boldsymbol{u}' = \lambda \boldsymbol{u}$ gives $\boldsymbol{u}_{n+1} = M(h\lambda)\boldsymbol{u}_n$, where

$$\begin{aligned} &M(z) = 1 + z\boldsymbol{b}^T(I - zA)^{-1}\mathbf{1}, \\ &\boldsymbol{b}^T = (b_1, ..., b_s), \quad A = (a_{ij})_{i,j=1}^{s}, \quad \mathbf{1} = (1, ..., 1)^T. \end{aligned} \tag{20}$$

For any s, there exist an implicit A-stable RK method of order $2s$.

Example. Gaussian method of order 4.

$\frac{1}{2} - \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4} - \frac{\sqrt{3}}{6}$
$\frac{1}{2} + \frac{\sqrt{3}}{6}$	$\frac{1}{4} + \frac{\sqrt{3}}{6}$	$\frac{1}{4}$
	$\frac{1}{2}$	$\frac{1}{2}$

Example. Gaussian method of order 6.

$\frac{1}{2} - \frac{\sqrt{15}}{10}$	$\frac{5}{36}$	$\frac{2}{9} - \frac{\sqrt{15}}{15}$	$\frac{5}{36} - \frac{\sqrt{15}}{30}$
$\frac{1}{2}$	$\frac{5}{36} + \frac{\sqrt{15}}{24}$	$\frac{2}{9}$	$\frac{5}{36} - \frac{\sqrt{15}}{24}$
$\frac{1}{2} + \frac{\sqrt{15}}{10}$	$\frac{5}{36} + \frac{\sqrt{15}}{30}$	$\frac{2}{9} + \frac{\sqrt{15}}{15}$	$\frac{5}{36}$
	$\frac{5}{18}$	$\frac{4}{9}$	$\frac{5}{18}$

Diagonal implicit Runge–Kutta methods. These are methods where the elements on the diagonal of A are nonzero, but the elements above are equal to zero.

c_1	γ				
c_2	$a_{2,1}$	γ			
c_3	$a_{3,1}$	$a_{3,2}$			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
c_s	$a_{s,1}$	$a_{s,2}$	$\cdots$	$a_{s,s-1}$	γ
	b_1	b_2	$\cdots$	b_{s-1}	b_s

The advantage of using these methods is to reduce the size of the system of algebraic equations that must be solved at each time step: we get s systems of m equations instead of a system of sm equations.

Example. A diagonal implicit RK method of order 4 (second to last line) and order 3 (last line).

$\frac{1}{4}$	$\frac{1}{4}$				
$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{4}$			
$\frac{11}{20}$	$\frac{17}{50}$	$-\frac{1}{25}$	$\frac{1}{4}$		
$\frac{1}{2}$	$\frac{371}{1360}$	$-\frac{137}{2720}$	$\frac{15}{544}$	$\frac{1}{4}$	
1	$\frac{25}{24}$	$-\frac{49}{48}$	$\frac{125}{16}$	$-\frac{85}{12}$	$\frac{1}{4}$
	$\frac{25}{24}$	$-\frac{49}{48}$	$\frac{125}{16}$	$-\frac{85}{12}$	$\frac{1}{4}$
	$\frac{59}{48}$	$-\frac{17}{96}$	$\frac{225}{32}$	$-\frac{85}{12}$	0

These methods are L-stable (A-stable and $\lim_{z\to\infty} M(z) = 0$).

Integrating factors.

Let us consider the following IVP:

$$\begin{cases} \boldsymbol{u}' = A\boldsymbol{u} + \boldsymbol{g}(t, \boldsymbol{u}), \quad t \in [0, T], \\ \boldsymbol{u}(0) = \boldsymbol{u}_0. \end{cases} \tag{21}$$

Let us write down a relater homogeneous linear problem

$$\begin{cases} \boldsymbol{v}' = A\boldsymbol{v}, \quad t \in [0, T], \\ \boldsymbol{v}(0) = \boldsymbol{v}_0. \end{cases} \tag{22}$$

Its solution is $\boldsymbol{v}(t) = e^{At}\boldsymbol{v}_0$. We define $\boldsymbol{U}(t) = e^{A(t-t_n)}\boldsymbol{u}(t)$ (then $\boldsymbol{u} = e^{-A(t-t_n)}\boldsymbol{U}$). We obtain the following equation for $\boldsymbol{U}$:

$$\boldsymbol{U}' = \boldsymbol{G}(\boldsymbol{U}), \quad \text{où} \quad \boldsymbol{G} = e^{A(t-t_n)}\boldsymbol{g}(t, e^{-A(t-t_n)}\boldsymbol{U}). \tag{23}$$

For this problem, a Runge–Kutta method yields

$$\boldsymbol{U}_{n+1} = \boldsymbol{U}_n + h\boldsymbol{\Phi}(t, \boldsymbol{U}_n, h) \tag{24}$$

with $\boldsymbol{U}_n = \boldsymbol{u}_n$ and $\boldsymbol{u}_{n+1} = e^{-Ah}\boldsymbol{U}_{n+1}$.

Splitting.

Let us consider the following problem:

$$\begin{cases} \boldsymbol{u}' = (A+B)\boldsymbol{u}, \quad t \in [0,T], \\ \boldsymbol{u}(0) = \boldsymbol{u}_0. \end{cases} \tag{25}$$

Suppose we have two different well-adapted methods for the equations $\boldsymbol{u}' = A\boldsymbol{u}$ and $\boldsymbol{u}' = B\boldsymbol{u}$, respectively. Splitting allows to use the same methods for the solution of (25).

Lie–Trotter formula: we replace (25) with a sequence of Cauchy problems

$$\boldsymbol{x}' = A\boldsymbol{x}, \quad \boldsymbol{x}(0) = \boldsymbol{u}_0, \quad t \in [0,h], \tag{26}$$

$$\boldsymbol{y}' = B\boldsymbol{y}, \quad \boldsymbol{y}(0) = \boldsymbol{x}(h), \quad t \in [0,h], \tag{27}$$

and use $\boldsymbol{y}(h)$ as an initial condition to repeat (26) over $t \in [h, 2h]$, and so on. We obtain an approximate solution $\boldsymbol{u}_{sp}(nh) = \boldsymbol{y}(nh)$.

To analyze the error of this method, we notice that

$$\boldsymbol{u}_{sp}(h) = C_{LT}(h)\boldsymbol{u}_0, \quad \text{où} \quad C_{LT}(h) = e^{Bh}e^{Ah} \tag{28}$$

and that

$$\boldsymbol{u}(h) = e^{(A+B)h}\boldsymbol{u}_0 \tag{29}$$

Then, the local truncation error of this method is

$$\boldsymbol{u}_{sp}(h) - \boldsymbol{u}(h) = \left[e^{Bh}e^{Ah} - e^{(A+B)h}\right]\boldsymbol{u}_0. \tag{30}$$

By definition,

$$\begin{aligned}
e^{(A+B)h} &= I + (A+B)h + \frac{1}{2!}(A+B)^2h^2 + \mathcal{O}(h^3) \\
&= I + (A+B)h + \frac{1}{2!}(A^2+B^2)h^2 + \frac{1}{2!}(AB)h^2 + \frac{1}{2!}(BA)h^2 + \mathcal{O}(h^3), \\
e^{Bh}e^{Ah} &= I + (A+B)h + \frac{1}{2!}(A^2+B^2)h^2 + (BA)h^2 + \mathcal{O}(h^3).
\end{aligned}$$

We obtain

$$\boldsymbol{u}_{sp}(h) - \boldsymbol{u}(h) = \frac{h^2}{2} \left[BA - AB\right] \boldsymbol{u}_0 + \mathcal{O}(h^3). \tag{31}$$

The global discretization error is, therefore, $\mathcal{O}(h)$: it is a first-order method.

A splitting formula is called *stable* if the operator $C(h)$ is such that there exists a constant $K > 0$ such that

$$||C(h)^n|| \leq K \tag{32}$$

for all $n \in \mathbb{N}$ such that $nh \leq T$.

If the operators e^{Ah} and e^{Bh} are contractive, the Lie–Trotter splitting is stable.

Strang splitting: we replace (25) by a sequence of Cauchy problems

$$\boldsymbol{x}' = A\boldsymbol{x}, \quad \boldsymbol{x}(0) = \boldsymbol{u}_0, \quad t \in [0, h/2], \tag{33}$$

$$\boldsymbol{y}' = B\boldsymbol{y}, \quad \boldsymbol{y}(0) = \boldsymbol{x}(h/2), \quad t \in [0, h], \tag{34}$$

$$\boldsymbol{x}' = A\boldsymbol{x}, \quad \boldsymbol{x}(h/2) = \boldsymbol{y}(h), \quad t \in [h/2, h], \tag{35}$$

and use $\boldsymbol{x}(h)$ as the initial condition for the next step. We obtain an approximate solution $\boldsymbol{u}_{sp}(nh) = \boldsymbol{x}(nh)$.
The corresponding splitting operator is

$$C_S(h) = e^{Ah/2} e^{Bh} e^{Ah/2}. \tag{36}$$

This is a second-order method.
There exist higher order splitting methods.