Mat3110: Runge–Kutta methods and A-stability

November 14, 2023

1 Theoretical results

Here is a short summary of topics covered in lectures that are not covered in Süli and Mayers. We use the same notation and terminology as in that textbook. We first summarize the main existence and uniqueness result for IVP and convergence of one-step methods given in the lectures (as they are not identical to the corresponding ones in SM). Section 2 describes Runge–Kutta methods and the Butcher tableau representation, and Section 3 treats A-stability for one-step methods.
Main existence and uniqueness result (proof in lecture):

Theorem 1.1 (Existence and uniqueness) Consider the IVP

$$y' = f(t,y) \qquad t \in [a,b], \quad y(a) = y_0 \in \mathbb{R}^d \tag{1.1}$$

with f Lipschitz in y. Then there exists a unique solution to (1.1) with $y \in C^1([a,b],\mathbb{R}^d)$.

Theorem 1.2 (Convergence of one-step method) Consider the IVP (1.1) with f Lipschitz in y. Let $y_{n+1} = y_n + h\Phi(t_n, y_n; h)$ with $h = (b-a)/N$ and $t_n = a + nh$, be an explicit one-step method with order of accuracy $p \geq 1$ (for the particular IVP) and that is Lipschitz continuous in y, meaning that

$$\|\Phi(t,y;h) - \Phi(t,y;h)\| \leq L_\Phi \|x - y\| \qquad \text{for all} \quad x,y \in \mathbb{R}^d, \quad t \in [a,b] \quad \& \quad h < h_0.$$

for some Lipschitz constant $L_\Phi > 0$ and $h_0 > 0$.
Then it holds that

$$\max_{n=0,1,\ldots,N} \|y_n - y(t_n)\| = \mathcal{O}(h^p).$$

2 Runge–Kutta methods

The explicit Runge-Kutta methods is a family of methods that

- do not require knowledge of partial derivatives of f to be used,
- can all be represented compactly in a Butcher tableau,
- has common features among subsets which often are easy to study and classify.

Definition 2.1 (Expicit s-stage Runge-Kutta method) For some natural number $s \geq 1$, consider $b, c \in \mathbb{R}^s$ and a strictly lower triangular matrix

$$A = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ a_{2,1} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ a_{s,1} & \cdots & a_{s,s-1} & 0 \end{bmatrix} \in \mathbb{R}^{s \times s}.$$

A stepping rule on the form

$$\Phi(t,y;h) = \sum_{i=1}^{s} b_i k_i(t,y;h), \qquad \text{where} \quad k_i = f\Big(t + c_i h,\, y + h \sum_{j=1}^{i-1} a_{ij} k_j\Big) \quad i = 1,\ldots,s$$

is called an explicit s-stage Runge-Kutta method.
The method is described by the weights $b = (b_1, \dots, b_s)$, the nodes $c = (c_1, \dots, c_s)$ and the coefficient matrix A, and it is common to arrange the information in a so-called Butcher tableau:

$$\begin{array}{c|ccc} c_1 & & & \\ \vdots & a_{2,1} & & \\ \vdots & \vdots & \ddots & \\ c_s & a_{s,1} & \cdots & a_{s,s-1} \\ \hline & b_1 & \cdots & \cdots & b_s \end{array} \qquad = \begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

Remark 2.2

- The method is explicit because each equation for k_i is explicit. This is because the sum is from $j = 0$ and only up to $i - 1$, which relates to the coefficient matrix A being strictly lower triangular.
- A more general way of expressing the system of equations for k_i, covering both explicit and implicit RK methods, is

$$k_i = f\Big(t + c_i h,\, y + h \sum_{j=1}^{s} a_{ij} k_j\Big) \qquad i = 1, \dots, s.$$

  This leads to an implicit system of equations for $k_1, \dots, k_s$ when A is not strictly lower triangular.

Example 2.3 (Explicit Euler) The 1-stage method

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

has the stepping rule

$$\Phi(t, y; h) = \sum_{i=1}^{s} b_k k_i = k_1,$$

with

$$k_1(t, y; h) = f(t + c_1 h, y + \sum_{j=1}^{0} a_{1j} k_j) = f(t, y).$$

This is the explicit Euler method

$$y_{j+1} = y_j + h\Phi(t_j, y_j; h) = y_j + h f(t_j, y_j).$$

Example 2.4 (Explicit midpoint method) The 2-stage method

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|cc} 0 & 0 & \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}$$

has the stepping rule

$$\Phi(t,y;h)=\sum_{i=1}^{s} b_k k_i = k_2,$$

with

$$k_1 = f(t+c_1 h, y) = f(t,y)$$

and

$$k_2(t,y;h) = f(t+c_2 h, y+h\sum_{j=1}^{2-1} a_{2j}k_j) = f(t+c_2 h, y+ha_{21}k_1) = f(t+\frac{h}{2}, y+\frac{h}{2}f(t,y)).$$

This is the explicit midpoint/modfied Euler method:

$$y_{j+1} = y_j + hf(t+\frac{h}{2}, y_j+\frac{h}{2}f(t_j,y_j)).$$

Example 2.5 The classic 4-stage RK method is given by

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 \end{array}$$

We obtain (recalling that $k_i = f\Big(t+c_i h,\, y+h\sum_{j=1}^{i-1} a_{ij}k_j\Big)$)

$$\begin{aligned}
k_1(t,y;h) &= f(t+c_1 h, y) = f(t,h)\\
k_2(t,y;h) &= f(t+c_2 h, y+ha_{21}k_1) = f\Big(t+\frac{h}{2}, y+\frac{h}{2}f(t,h)\Big)\\
k_3(t,y;h) &= f\Big(t+c_3 h, y+h(a_{31}k_1+a_{32}k_2)\Big) = f\Big(t+\frac{h}{2}, y+\frac{h}{2}k_2\Big)\\
k_4(t,y;h) &= f\Big(t+c_4 h, y+ha_{43}k_3\Big) = f\Big(t+h, y+hk_3\Big),
\end{aligned}$$

the stepping rule

$$\Phi(t,y;h)=\sum_{i=1}^{s} b_i k_i = \frac{k_1+2k_2+2k_3+k_4}{6},$$

and the so-called RK4 method (see also Figure 2.1)

$$y_{j+1} = y_j + \frac{h_j}{6}\Big(k_1+2k_2+2k_3+k_4\Big)(t_j,y_j;h).$$

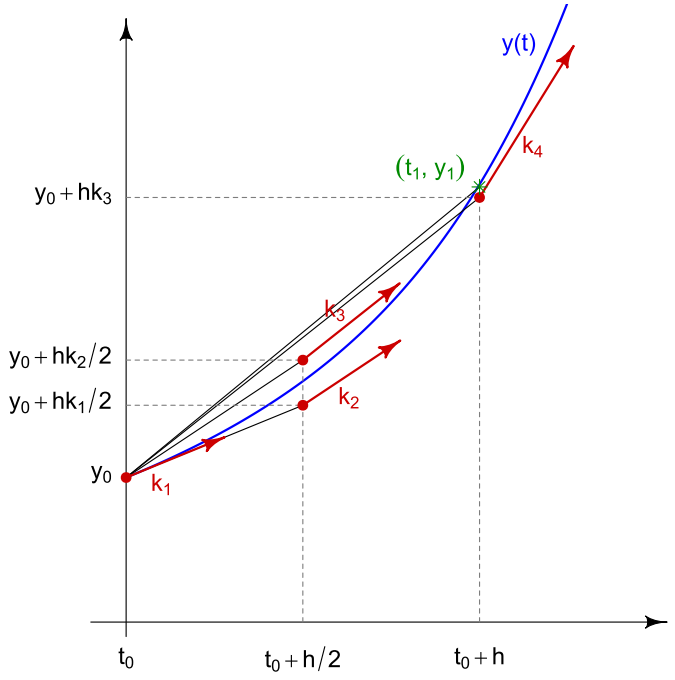

Figure 2.1: One step of the RK4 method of the ODE $y' = y + t^3$. See that k_i approximate y', in the sense that that $k_i(t_n, y_n; h) \approx y'(t_n + c_i h)$ for $i = 1, \ldots, s$. (Source: Wikipedia)

2.1 Coefficient constraints and convergence properties

We recall that a one-step method is consistent if $\lim_{h\to 0} \Phi(t, y; h) = f(t, y)$.
An $s-$stage explicit RK-method:

- is consistent iff $\sum_{i=1}^{s} b_i = 1$
- has order of accuracy $p \leq s$ and
  - $p \geq 1$ if $\sum_{i=1}^{s} b_i = 1$
  - $p \geq 2$ if additionally $c_i = \sum_{j=1}^{i-1} a_{ij}$ for $i = 1, \ldots, s$ and $\sum_{i=1}^{s} b_i c_i = 1/2$
  - $p \geq 3$ if additionally further conditions hold, etc.

2.2 Implicit Runge–Kutta methods

An s-stage RK method with the tableau (2.1)

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|ccc} c_1 & a_{1,1} & \cdots & a_{1,s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s,1} & \cdots & a_{s,s-1} \\ \hline & b_1 & \cdots & b_s \end{array} \tag{2.1}$$

is called **implicit** if $a_{i,j} \neq 0$ for at least one component with $j \geq i$.

Recall that an s-stage RK method with Butcher tableau (2.1) has the stepping rule

$$y_{n+1} = y_n + h\Phi(t_n, y_n, y_{n+1}; h)$$

with

$$\Phi(t_n, y_n, y_{n+1}; h) = \sum_{i=1}^{s} b_i k_i,$$

and system of equations

$$k_i = f\left(t_n + c_i h,\ y_n + h\sum_{j=1}^{s} a_{ij} k_j\right) \qquad i = 1, \ldots, s,$$

Solution approach: introduce $F : \mathbb{R}^s \to \mathbb{R}^s$ where

$$F_i(k_1, \ldots, k_s) = k_i - f\left(t_n + c_i h,\ y_n + h\sum_{j=1}^{s} a_{ij} k_j\right) \quad i = 1, \ldots, s$$

and solve $F(k_1, \ldots, k_s) = 0$ using e.g. Newton's method (for every timestep).

2.3 Implicit vs explicit

- Implicit methods tend to be more stable than explicit methods.
- So one can often solve problems robustly with larger $h > 0$ with implicit methods than explicit.
- Implicit methods are more suitable for **stiff problems**, involving dynamics on different timescales, like

  $$y' = \begin{pmatrix} -1 & 1/100 \\ 0 & -100 \end{pmatrix} y,$$

  where, $y_2(t) = e^{-100t} y_2(0)$ may vary on a faster timescale than

  $$y_1(t) = e^{-t} y_1(0) + \text{“small contribution from”} y_2.$$

- A drawback is that implicit methods can be more computationally costly than explicit methods, as one needs to solve implicit equation for y_{n+1} every iteration.

3 Long-time stability

For studying the long-time stability properties of numerical methods we introduce the test equation

$$y' = \underbrace{\lambda y}_{f(y)}, \qquad \text{for} \quad t \geq 0 \quad \text{and some } \lambda < 0, \tag{3.1}$$

The exact solution

$$y(t) = y_0 e^{\lambda t},$$

is asymptotically decaying in absolute value, and if we consider the solution perturbed initial data $y^\varepsilon(0) = y(0) + \varepsilon$, we observe a similar asymptotic decay in the perturbation error:

$$|y(t) - y^\varepsilon(t)| \leq e^{\lambda t} |y(0) - y^\varepsilon(0)|.$$

We would like the observed decay to carry over to numerical methods solving the test equation.

Explicit Euler method

For

$$y_{j+1} = y_j + hf(y_j) = (1 + \lambda h) y_j = (1 + \lambda h)^{j+1} y_0, \tag{3.2}$$

we observe that

$$|y_{j+1}| < |y_j| \iff -1 < 1 + \lambda h < 1 \iff h < \frac{2}{|\lambda|}.$$

Conclusion: the decay of the solution is determined by the stability function

$$R(\lambda h) = 1 + \lambda h$$

and

- For $h > 2/|\lambda|$, we have that $R(\lambda h) > 1$ and the method is unstable.
- For $h \in (1/|\lambda|, 2/|\lambda|)$, we have that $-1 < R(\lambda h) < 0$ so the solution decays, but it will have unnatural oscillations due to $R(\lambda h)$ being negative.
- For $h < 1/|\lambda|$, we have that $0 < R(\lambda h) < 1$. The numerical solution decays in a non-oscillatory manner, consistent with the exact solution.

See Figure 3.1 for a numerical verification of these observations.

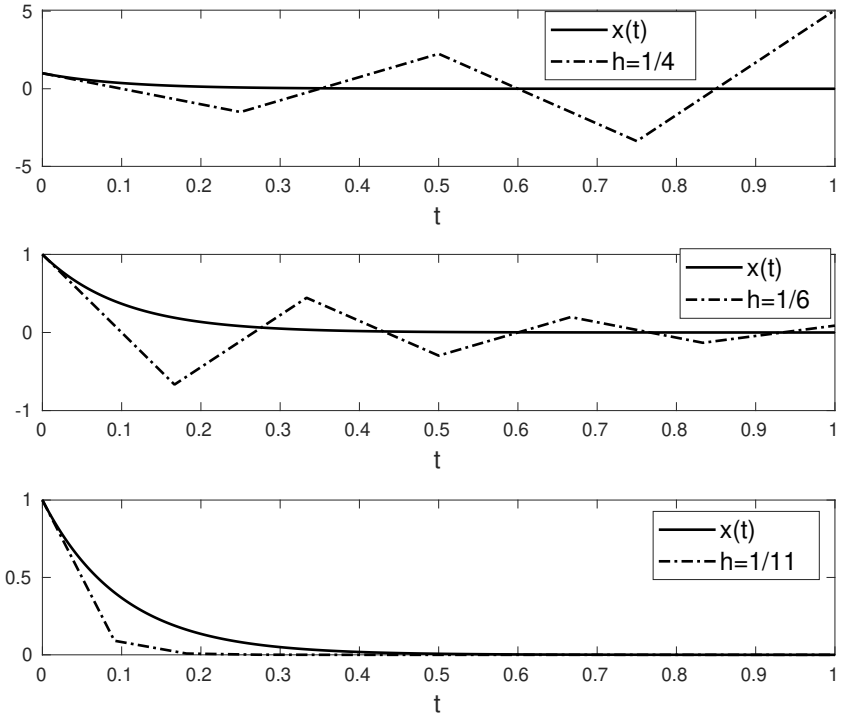

Figure 3.1: Explicit Euler solutions of the test equation (3.1) with $\lambda = -10$ and $y(0) = 1$. **Top:** unstable setting with $h > 2/10$, **middle:** stable but oscillatory solution with $h \in (1/10, 2/10)$ and **bottom:** "reasonable" non-oscillatory solution with $h < 1/10$.

Implicit Euler

The method

$$\begin{aligned} & y_{j+1} = y_j + hf(y_{j+1}) = y_j + h\lambda y_{j+1} \\ \Longrightarrow\ & y_{j+1} = \frac{y_j}{1-\lambda h} = (1-\lambda h)^{-(j+1)} y_0, \end{aligned} \tag{3.3}$$

implies that

$$|y_{j+1}| < |y_j| \iff \frac{1}{1-\lambda h} < 1.$$

This holds for any $h > 0$ since $\lambda < 0$, so unlike explicit Euler, there is no stepsize constraint for the implicit Euler. A numerical comparison of explicit- and implicit Euler is given in Figures 3.1 and 3.2.

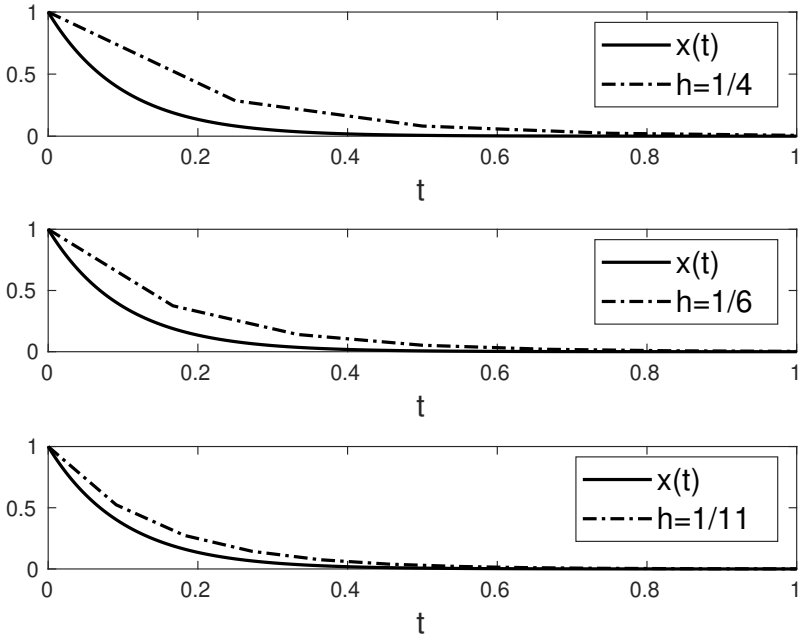

Figure 3.2: Implicit Euler solutions of the test equation (3.1) with $\lambda = -10$ and $y(0) = 1$.

A-Stability

Dahlquist's test equation is generally of the form

$$y' = \lambda y, \qquad \text{for } \lambda \in \mathbb{C}. \tag{3.4}$$

Definition 3.1 (Stability function) For a one-step method applied to (3.4) the associated stability function $R : \mathbb{C} \to \mathbb{C}$ is for $z = \lambda h$ defined as the function satisfying by

$$y_{j+1} = R(z) y_j$$

For explicit Euler, for instance,

$$y_{j+1} = (1 + \lambda h) y_j \implies R(z) = 1 + z.$$

Theorem 3.2 For all consistent one-step methods that we have considered (RK and Taylor based), the stability function can be written as rational function. That is

$$R(z) = \frac{P(z)}{Q(z)},$$

where $P : \mathbb{C} \to \mathbb{C}$ and $Q : \mathbb{C} \to \mathbb{C}$ are polynomials that satisfy

(i) $Q(z) = 1$ for explicit methods,

(ii) $R(0) = 1$ (and we set $P(0) = Q(0) = 1$),

Definition 3.3 (Region of absolute stability and A-Stability) A one-step method with stability function R has region of absolute stability

$$\mathcal{S} = \{z \in \mathbb{C} \,|\, |R(z)| < 1\},$$

and the method is called A-stable (absolutely stable) if it holds that $\mathcal{S} \supset \mathbb{C}_- = \{z \in \mathbb{C} \,|\, \mathrm{Re}(z) < 0\}$

Extension to $d > 1$: When the numerical method is applied to a linear system of ODE $y' = Ay$ using stepsize $h > 0$, it is said to be stable if

$$\lambda h \in \mathcal{S} \quad \text{for } \textbf{all} \text{ eigenvalues } \lambda \in \sigma(A).$$

Example 3.4 (Explicit Euler)

$$y_{j+1} = \underbrace{(1+\lambda h)}_{R(\lambda h)} y_j \implies R(z) = 1 + z$$

with region of absolute stability

$$\mathcal{S} = \{z \in \mathbb{C} \,|\, |1+z| < 1\}$$

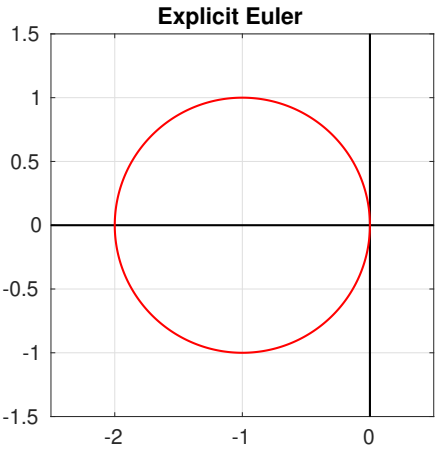

Figure 3.3: The region of absolute stability of the explicit Euler method is the **interior** of the red curve.

Example 3.5 (Implicit Euler) From (3.3) we recall that

$$y_{j+1} = \underbrace{\frac{1}{1-\lambda h}}_{R(\lambda h)} y_j, \text{ which implies that } R(z) = \frac{1}{1-z}$$

with region of absolute stability

$$\mathcal{S} = \{z \in \mathbb{C} \,|\, \frac{1}{|1-z|} \leq 1\} = \{z \in \mathbb{C} \,|\, |1-z| > 1\}$$

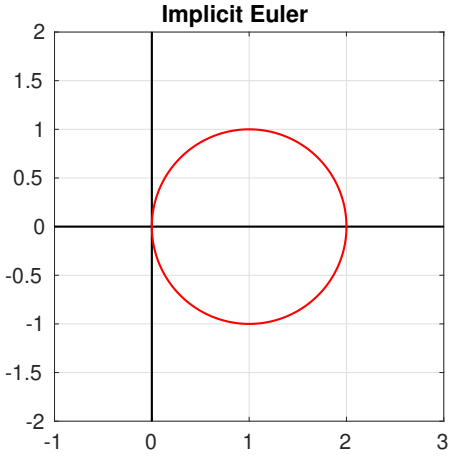

Figure 3.4: The region of absolute stability of the implicit Euler method is the **exterior** of the red curve.

Example 3.6 (Explicit Runge-Kutta 2, 3, and 4) The RK2 method is given by

$$y_{j+1} = y_j + hf\left(y_j + \frac{h}{2}f(y_j)\right) = \left(1 + \lambda h + \frac{(\lambda h)^2}{2}\right)y_j$$

which implies that

$$R_{RK2}(z) = 1 + z + \frac{z^2}{2}.$$

By similar computations, one can show that

$$R_{RK3}(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{3!}, \quad \text{and} \quad R_{RK4}(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{3!} + \frac{z^4}{4!}.$$

(RK2 here denoting any $2-$*stage* explicit RK method with order of accuracy 2, and similar for RK3 and RK4.)

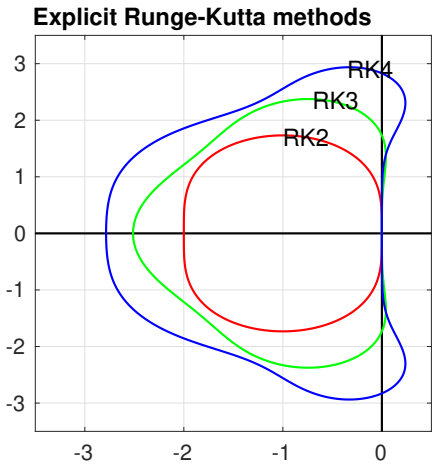

Figure 3.5: The region of absolute stability of RK 2,3 and 4 is the **interior** of the respective curves.

Example 3.7 (Explicit Euler on a system of ODE) For the linear system of ODE with $x \in \mathbb{R}^2$

$$\frac{d}{dt}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \underbrace{\begin{pmatrix} -2 & 1 \\ -1 & -2 \end{pmatrix}}_{=A} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \tag{3.5}$$

where A has complex-valued eigenvalues: $\lambda_1 = -2 - i$ and $\lambda_2 = -2 + i$. By Definiton 3.3(ii), the explicit Euler method is stable for all $h > 0$ such that

$$|1 + \lambda_1 h| < 1 \quad \text{and} \quad |1 + \lambda_2 h| < 1. \tag{3.6}$$

Since

$$|1 + \lambda_1 h|^2 = |1 + \lambda_2 h|^2 = (1 - 2h)^2 + h^2 = 1 - 4h + 5h^2$$

the condition (3.6) holds if and only if

$$-4h + 5h^2 < 0 \iff h < 4/5.$$

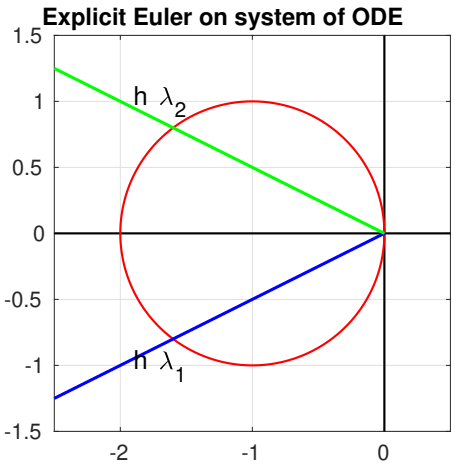

Figure 3.6: The admissible stepsizes for explicit Euler solving (3.5) are the h values along the blue and green lines which intersect with/are inside the method's red-circled region of absolute stability. That is, $h < 4/5$.

Exercise: Why cannot a consistent explicit Runge-Kutta method be A-stable? (Hint: its stability function $R(z)$ is a polynomial of degree ≥ 1 with $R(0) = 1$.)
Exercise: Describe the set of all 2-stage RK methods (b, c, A) with $a_{12} = 0$ that have order of accuracy 2 and are A-stable.