P. Grohs
W. Wu

Spring Term 2016

Numerical Analysis II

ETH Zürich
D-MATH

Problem Sheet 9

Problem 9.1 Stability Domain of a Rational Single Step Method

Consider the rational function

$$R(z) = \frac{2 - z^2}{2(1-z)}.$$

(9.1a) Determine the maximal $p \in \mathbb{N}$ such that

$$|\exp(z) - R(z)| = \mathcal{O}(|z|^{p+1}) \qquad \text{for } z \to 0.$$

HINT: Compute the first three derivatives of $R(z)$ and use them to compare the Taylor series of $\exp(z)$ and $R(z)$ around the point 0.

Solution: The consistency order of the approximation to the exponential function can be inferred by comparing the Taylor series of $R(z)$ and $\exp(z)$ about $z = 0$. Taylor expansion yields

$$\begin{aligned} R(z) &= R(0) + z\dot{R}(0) + \frac{z^2}{2}\ddot{R}(0) + \frac{z^3}{6}R^{(3)}(0) + \frac{z^4}{24}R^{(4)}(0) + \mathcal{O}(z^5) \\ &= 1 + z + \frac{z^2}{2} + \frac{z^3}{2} + \frac{z^4}{2} + \mathcal{O}(z^5), \end{aligned}$$

and thus

$$\begin{aligned} R(z) - e^z &= R(z) - \sum_{i=0}^{\infty} \frac{z^i}{i!} \\ &= \frac{z^3}{3} + \frac{11z^4}{24} + \mathcal{O}(z^5). \end{aligned}$$

The approximation is therefore of second order.

One can obtain the series expansion of $R(z)$ more quickly by considering the partial fraction decomposition of $R(z)$:

$$\begin{aligned} R(z) = \frac{2 - z^2}{2(1-z)} &= \frac{1+z}{2} + \frac{1}{2(1-z)} \\ &= \frac{1}{2}\left(1 + z + \sum_{i=0}^{\infty} z^i\right) \\ &= 1 + z + \frac{z^2}{2} + \frac{z^3}{2} + \frac{z^4}{2} + \mathcal{O}(z^5). \end{aligned}$$

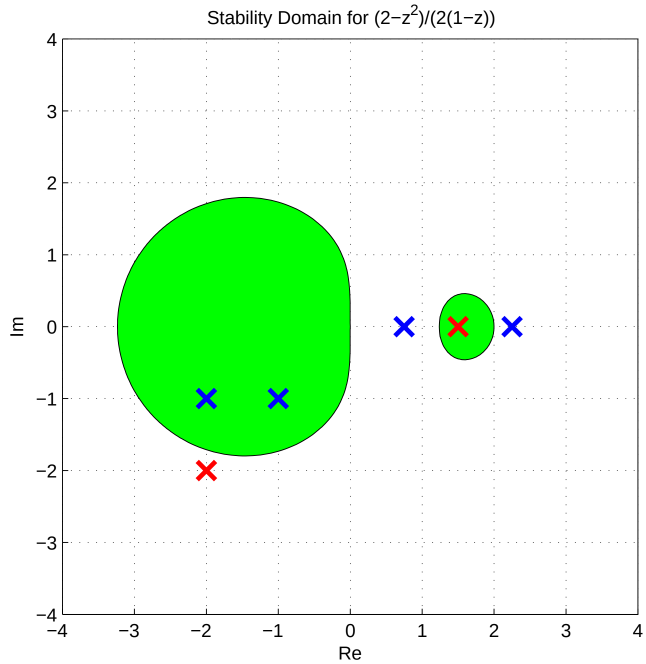

Figure 9.1: Stability domain for the stability function $R(z)$.

(9.1b) Consider $R(z)$ as a stability function of a Runge-Kutta single step method and plot its stability domain in MATLAB by completing the template `StabilityRegion.m`.

Solution: For the code, see Listing 9.1.

Listing 9.1: Plot the stability domain of $R(z)$.

```
function StabilityRegion
% STABDOM Stability Domain of (2-z^2)/(2(1-z))

[X, Y] = meshgrid(-4:0.05:4);
Z = X + 1i*Y;

R = ( (2 - Z.^2)./(2.*(1-Z)) );

figure
contourf(X, Y, abs(R), [1 1]);
title('Stability Region of (2-z^2)/(2(1-z))')
xlabel('Re')
ylabel('Im')
axis square
grid on
```

The stability domain is shown in Figure 9.1.

(9.1c) Show that a Runge-Kutta method with stability function $R(z)$ is of convergence order 2 when applied to linear ODEs, that is, to problems of the form $\dot{y} = \lambda y,\ y(0) = y_0$.

Solution: It suffices to establish that the given method is of *consistency* order 2. The conclusion then follows by using Theorem 2.1.16 from the lecture notes. The continuous flow of the ODE satisfies $\Phi^{t,t+h}y = \exp(\lambda h)y_0$. We have

$$|\Phi^{t,t+h}y - \Psi^{t,t+h}y| = |R(\lambda h) - \exp(\lambda h)||y_0| \leq C\frac{|\lambda h|^3}{3},$$

where the last inequality follows by (9.1a). Hence, we can conclude that the method is of consistency order 2, and by Theorem 2.1.16 it follows that it is also of convergence order 2.

(9.1d) Write down (in detail) the discrete evolution of the single step method (whose stability function is $R(z)$), when applied to the autonomous linear differential equation

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y}, \qquad \mathbf{A} \in \mathbb{R}^{d\times d}. \tag{9.1.1}$$

Solution: The method equation for the system

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y}, \qquad \mathbf{A} \in \mathbb{R}^{d\times d}, \quad \mathbf{y}(0) = \mathbf{y}_0$$

with step size h is given by

$$\mathbf{y}_{k+1} = R(h\mathbf{A})\mathbf{y}_k = [2(I - h\mathbf{A})]^{-1}\big(2I - (h\mathbf{A})^2\big)\mathbf{y}_k.$$

(9.1e) Implement the method (in MATLAB) for the approximate solution of (9.1.1) by completing the template `RationalSSM.m` to solve the initial value problem

$$\dot{\mathbf{y}} = \begin{pmatrix} \alpha & \beta \\ -\beta & \alpha \end{pmatrix}\mathbf{y}, \quad \mathbf{y}(0) = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

for $t \in [0, 10]$ with the values

	(i)	(ii)	(iii)	(iv)	(v)	(vi)
α	-2	-2	-2	1.5	1.5	1.5
β	-1	-2	-2	0	0	0
h	1	1	0.5	0.5	1	1.5

where h is the step size. Plot your results and compare them with the exact solution. Explain the behaviour of the method with the help of the stability domain of $R(z)$.

Solution: For the implementation, see Listing 9.2.

Listing 9.2: Solve the IVP for various values.

```
function RationalSSM
%  Implementation of the 'exotic' scheme
%   y(k+1) = (2-z^2)/(2(1-z)) * y(k)

% given data
alpha   = [-2, -2, -2, 1.5, 1.5, 1.5];
beta    = [-1, -2, -2,  0,    0,    0 ];
h       = [ 1,   1, 0.5, 0.5, 1,   1.5];
```

```
T = [0 10];
y0 = [2; 2];

tExact = T(1):0.05:T(2);

% loop over all parameter configurations
for p = 1:length(alpha)

      % set up matrix A
      A = [ alpha(p), beta(p);
          -beta(p),  alpha(p)];

      % evaluate stabilty function at hA
      R = ( 2*eye(2) - (h(p)*A)^2 )/( 2*(eye(2) - h(p)*A) );

      % define time steps
      t = T(1):h(p):T(2);

      % allocate memory for results
      y = nan(2,length(t));

      % store initial value
      y(:,1) = y0;

      % time stepping
      for k=1:length(t)

            y(:,k+1) = R*y(:,k);
      end

      % plot results
      figure;

      % limit axis in case of blow-up in analytical solution
      if ( real(eig(A)) > 0)

            axis([-1,4,-1,4]);
      else

            axis equal;
      end

      hold on;
      grid on;

      % compute exact solution
```

```
    yExact = sol(tExact,y0,alpha(p),beta(p));

    % plot computed solution
    plot(y(1,:), y(2,:), 'r', 'LineWidth', 2.5);

    % plot exact solution
    plot(yExact(1,:), yExact(2,:), 'g', 'Linewidth', 1.2);

    % plot initial value
    plot(y0(1), y0(2), 'gh', 'LineWidth', 2.5, 'MarkerSize',
       12);

    % plot final value of computed solution
    plot(y(1,end), y(2,end), 'mx', 'LineWidth', 2.5,
       'MarkerSize',16);

    % plot final value of exact solution
    plot(yExact(1,end), yExact(2,end), 'b.', 'LineWidth',
       2.5, 'MarkerSize', 18)

    legend('Numerical',...
          'Exact',...
          'y0',...
          ['y_h(', num2str(T(2)), ')'],...
          ['y(',   num2str(T(2)), ')'],...
          'Location','SouthEast');

    title(['Solution for',...
          ' \alpha=', num2str(alpha(p)),...
          ', \beta=', num2str(beta(p)),...
          ' with step size h=', num2str(h(p))]);

    hold off
end

function yExact = sol(t,y0,alpha,beta)
% implementation of exact solution

  yExact = nan(2,length(t));
  yExact(1,:) = y0(1)*exp(t*alpha).*( cos(t*beta) +
     sin(t*beta) );
  yExact(2,:) = y0(2)*exp(t*alpha).*( cos(t*beta) -
     sin(t*beta) );
```

From the plots in Figure 9.2, we can see that the behaviour of the method for the parameter values b and e is qualitatively wrong.

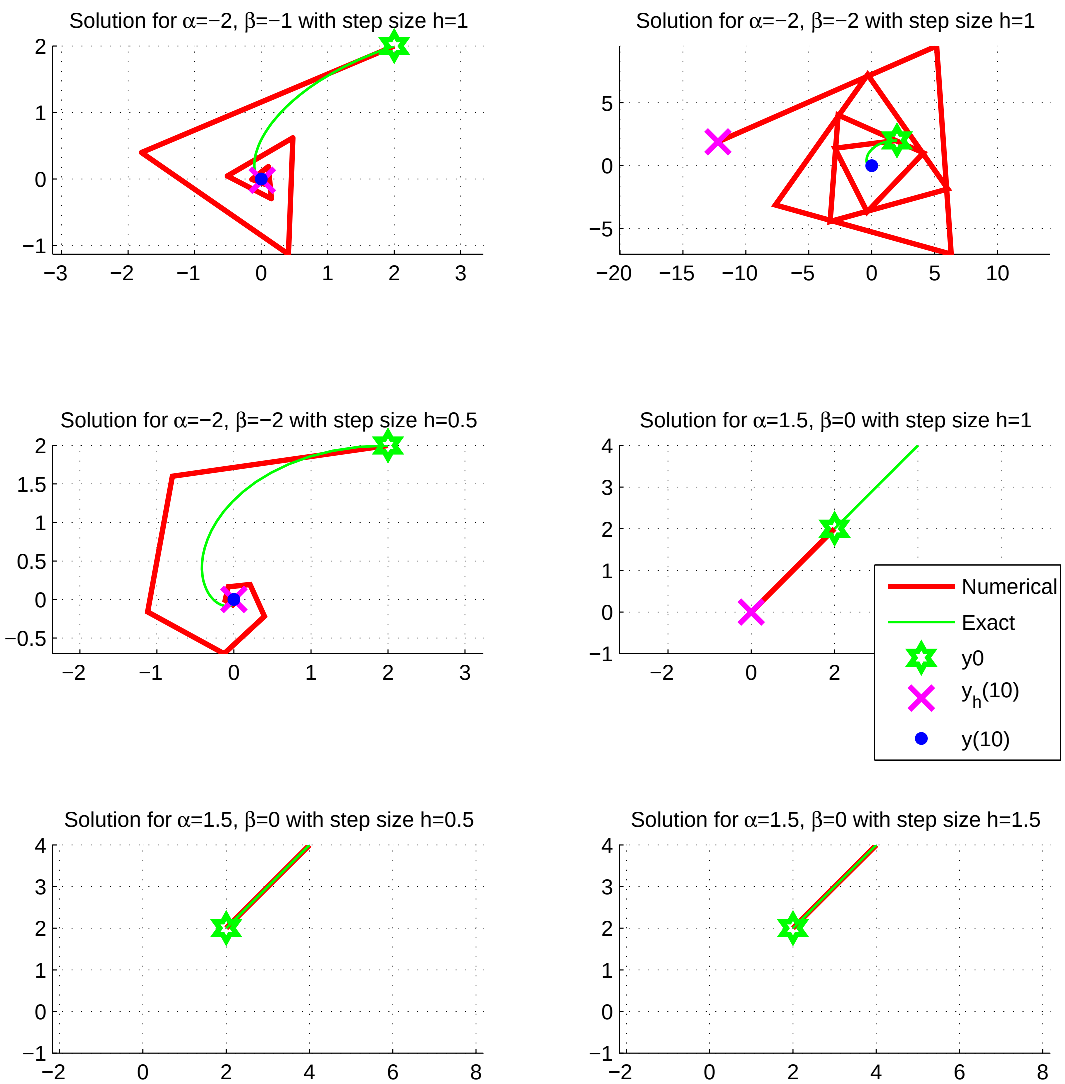

Figure 9.2: Exact and numerical solution for the given parameter values.

We can explain this behaviour based on the eigenvalues of the matrix A and on the stability domain $\mathcal{S}(R(z))$ (Figure 9.1). First, we diagonalize the matrix A (see also [NUMODE, Sect. 1.3.2]). The eigenvalues of A are given by $\lambda_{1,2} = \alpha \pm i\beta$, the corresponding eigenvectors are $(i, 1)^\top$ und $(-i, 1)^\top$. We define the matrix S consisting of the eigenvectors and the diagonal matrix Λ built from the eigenvalues:

$$S := \begin{pmatrix} i & -i \\ 1 & 1 \end{pmatrix}, \qquad \Lambda := \begin{pmatrix} \alpha + i\beta & 0 \\ 0 & \alpha - i\beta \end{pmatrix}.$$

Then, $A = S\Lambda S^{-1}$. We introduce the new variable $\widehat{\mathbf{y}} := S^{-1}\mathbf{y}$ and obtain the equivalent, decoupled system

$$\dot{\widehat{\mathbf{y}}} = \Lambda \widehat{\mathbf{y}}.$$

The method defined by $R(z)$ for the transformed system is therefore given by

$$\widehat{\mathbf{y}}_{k+1} = R(h\Lambda)\widehat{\mathbf{y}}_k = R\left(h \begin{pmatrix} \alpha + i\beta & 0 \\ 0 & \alpha - i\beta \end{pmatrix}\right)^k \widehat{\mathbf{y}}_0.$$

The behaviour of the numerical solution now becomes clear: The numerical solution behaves asymptotically stable, if $|R(h\lambda_{1,2})| < 1$. This is the case, precisely if $h\lambda_{1,2} \in \mathcal{S}(R(z))$. The behaviour of both the numerical as well as the analytical solution is summarized in the following table:

	parameter			behaviour	
	α	β	h	analytical	numerical
a	-2	-1	1	asymp. stable	asymp. stable
b	-2	-2	1	asymp. stable	unstable
c	-2	-2	0.5	asymp. stable	asymp. stable
d	1.5	0	0.5	unstable	unstable
e	1.5	0	1	unstable	asymp. stable
f	1.5	0	1.5	unstable	unstable

Problem 9.2 Stability-Induced Bound on Step Size.

The stability domains of explicit Runge-Kutta methods are necessarily bounded, see [NODE, Lemma. 3.1.10]. This leads to a stability-induced bound on the step size in the vicinity of asymptotically stable fixed points.

Consider the logistic differential equation $\dot{y} = \lambda y(1-y)$, $\lambda > 0$, with asymptotically stable fixed point $y = 1$.

Determine numerically an optimal bound on the step size (in dependence of λ) for the classical Runge-Kutta method and the embedded method of order 4(5) of Merson (see [NUMODE, Ex. 2.6.14]) in such a way, that the stability of the fixed point $y = 1$ is only just preserved by the discretization.

Complete the template `stabfn.m`, `stpRestrict.m`. Test your bound in a numerical experiment and plot with `stabdomRK.m`.

HINT: [NODE, Thm. 3.2.8].

Solution: According to [NODE, Thm. 3.2.8], $h\sigma(f'(1)) \subset S_{\Phi}$ must be contained in the stability domain. For the logistic ODE, we have $f'(1) = -\lambda$, i.e. we are looking for the intersection of the stability domain with the negative real axis. This point will give us the maximal step size. We use an iterative bisection algorithm. For the MATLAB code, see Listing 9.4.

Listing 9.3: Implementation of `stabfn.m`

```
function S = stabfn(A,b,z)
s = length(b);

e = ones(s,1);
S = zeros(size(z));
for i=1:length(z)
     d = (eye(s)-z(i)*A)\e;
     S(i) = 1+z(i)*dot(b,d);
 end
```

Listing 9.4: Implementation of `stpRestrict.m`

```
function z0=stepRestrict(A,b,z0,z1)

% RK4
% A=[ 0 0 0 0;...
%       0.5 0 0 0;...
%       0 0.5 0 0;...
%       0 0 1 0];
% b= [1/6 2/6 2/6 1/6];
% stepRestrict(A,b,-2.5,-3.5);

%RK4(5)
% A=[ 0 0 0 0 0;...
%       1/3 0 0 0 0;...
%       1/6 1/6 0 0 0;...
%       1/8 0 3/8 0 0; ...
%       1/2 0 -3/2 2 0 ];
% b= [1/6 0 0 2/3 1/6];
% stepRestrict(A,b,-3,-4);
% b= [1/10 0 3/10 2/5 1/5];
% stepRestrict(A,b,-3,-4);

l=abs(z1-z0);
sign=-1;
nsteps=100;
i=1;

while (i<nsteps && abs(stabfn(A,b,z0)-1)>eps)
    z0=z0+sign*2^-(i)*l;
    if abs(stabfn(A,b,z0))>1
        sign=1;
```

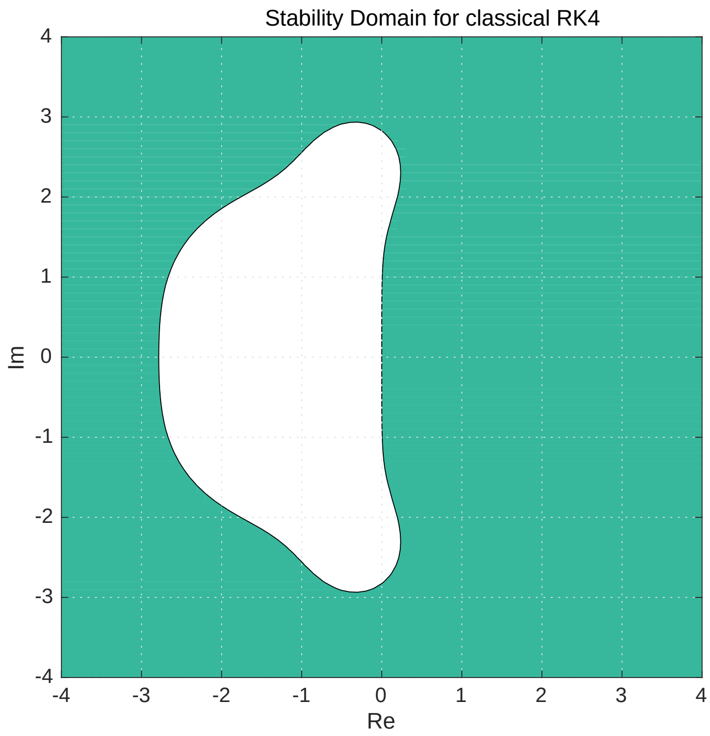

Figure 9.3: Stability domain for classical RK4 Method.

```
    else
        sign=-1;
    end
    i=i+1;
end
[i z0]
```

The results are

- RK4: $h\lambda < 2.7853$
- RK4(5) 4. order: $h\lambda < 3.5483$
- RK4(5) 5. order: $h\lambda < 3.2170$

Problem 9.3 Diagonalizable matrices are dense in $\mathbb{C}^{d\times d}$

Prove that the set of all diagonalizable $d \times d$ complex matrices is dense in $\mathbb{C}^{d\times d}$.

HINT: Use the Schur decomposition and the fact that a matrix with pairwise distinct eigenvalues is diagonalizable.

Solution: Using Schur decomposition, any $d \times d$ complex matrix $\mathbf{A}$ can be decomposed as

$$\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^*,$$

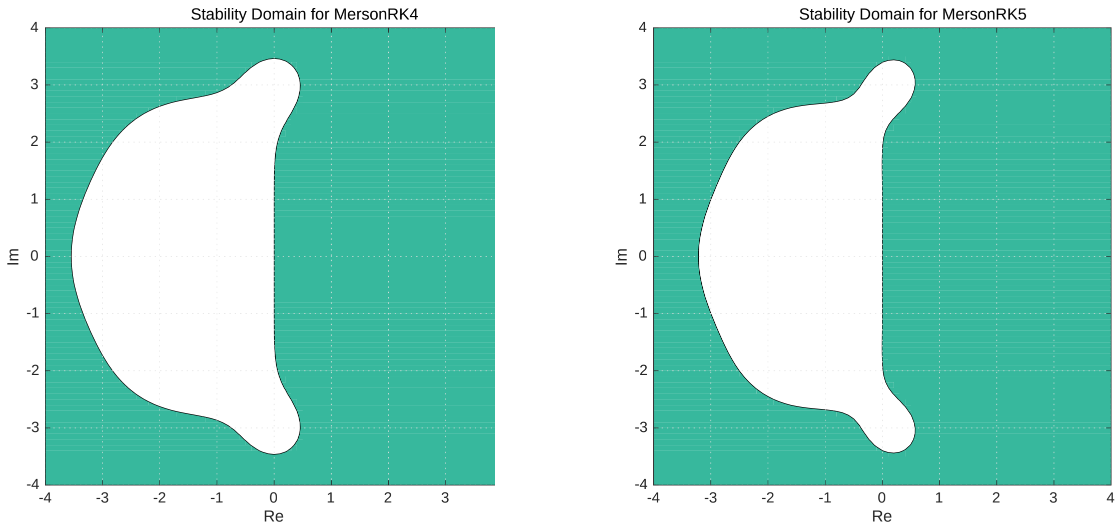

Figure 9.4: Stability domain for RK4 Merson and RK5 Merson methods.

where $\mathbf{U}$ is a unitary matrix, $\mathbf{T}$ an upper triangular matrix and $\mathbf{U}^*$ denotes the conjugate transpose. For any upper triangular matrix $\mathbf{T} \in \mathbb{C}^{d \times d}$, there exists a diagonal matrix $\mathbf{D}$ with arbitrary small diagonal entries such that $\mathbf{T} + \mathbf{D}$ has pairwise distinct diagonal entries. Thus $\mathbf{T} + \mathbf{D}$ is diagonalizable, so is $\mathbf{U}(\mathbf{T} + \mathbf{D})\mathbf{U}^H$ because it also has pairwise distinct eigenvalues, which finishes the proof.

Published on 25 April 2016.
To be submitted by 3 May 2016.

References

[NODE] Lecture Notes for the course "Numerical Methods for Ordinary Differential Equations".

[NUMODE] Lecture Slides for the course "Numerical Methods for Ordinary Differential Equations", SVN revision # 52913.

Last modified on May 4, 2016