Numerical Solutions to Differential Equations

Lecture Notes #11 — Runge-Kutta Methods for Stiff ODEs

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Spring 2015

Outline

1. Introduction
   - Review: Stability for Explicit Runge-Kutta Methods
   - Stability of Semi-Implicit RK-Methods
2. Approximations of e^x
   - Optimal Polynomial Approximations
   - Optimal Rational (Padè) Approximations
   - Rational Approximations: Classification and Properties
3. Implicit RK-Methods for Stiff Problems
   - Examples: Gauss-Legendre Methods
   - Wishing for L-stability... The Radau Methods

By applying the RK-methods to the scalar test-problem $\mathbf{y}'(\mathbf{t}) = \lambda \mathbf{y}(\mathbf{t}),\ \mathbf{y}(\mathbf{t_0}) = \mathbf{y_0}$ we will find the regions of stability for the methods.

E.g. Heun's Method

$$\begin{array}{c|cc} c_1 & a_{1,1} & a_{1,2} \\ c_2 & a_{2,1} & a_{2,2} \\ \hline & b_1 & b_2 \end{array} \quad = \quad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

Hence

$$\begin{aligned} k_1 &= f(t_n, y_n) = \lambda y_n \\ k_2 &= f(t_n + h, y_n + hk_1) = \lambda(y_n + hk_1) = \lambda y_n + h\lambda^2 y_n \\ y_{n+1} &= y_n \left[1 + \tfrac{h}{2}\left[2\lambda + h\lambda^2\right]\right] = y_n \underbrace{\left[1 + h\lambda + \frac{(h\lambda)^2}{2}\right]}_{\mathbf{R}(\mathbf{h}\lambda)} \end{aligned}$$

The iteration is given by

$$y_{n+1} = R(h\lambda)y_n,$$

and the stability region is given by

$$|R(h\lambda)| = \left|1 + h\lambda + \frac{(h\lambda)^2}{2}\right| \leq 1.$$

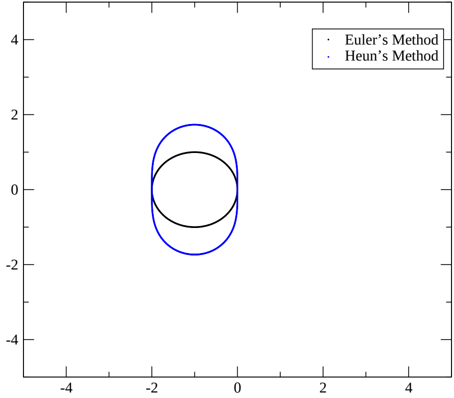
We find the boundary of the region by find the complex roots of

$$1 - e^{i\theta} + h\lambda + \frac{(h\lambda)^2}{2} = 0,$$

for all values of $\theta \in [0, 2\pi)$.

We find the boundary of the region by find the complex roots of

$$1 - e^{i\theta} + h\lambda + \frac{(h\lambda)^2}{2} = 0, \quad \forall \theta \in [0, 2\pi).$$



For notational convenience we absorb $h\lambda \to \widehat{h}$.

Using the A from the Butcher array, we can write the k_i's

$$\tilde{\mathbf{k}} = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_s \end{bmatrix} = y_n \tilde{\mathbf{1}} + \widehat{h} A \tilde{\mathbf{k}}, \quad \text{where } \tilde{\mathbf{1}} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix},$$

thus, we can solve for $\tilde{\mathbf{k}}$:

$$\tilde{\mathbf{k}} = (I - \widehat{h} A)^{-1} \tilde{\mathbf{1}} y_n.$$

Further,

$$y_{n+1} = y_n + \widehat{h} \tilde{\mathbf{b}}^T \tilde{\mathbf{k}} = y_n + \widehat{h} \tilde{\mathbf{b}}^T (I - \widehat{h} A)^{-1} \tilde{\mathbf{1}} y_n.$$

We have

$$y_{n+1} = y_n + \widehat{h}\tilde{\mathbf{b}}^T\tilde{\mathbf{k}} = y_n + \widehat{h}\tilde{\mathbf{b}}^T(I - \widehat{h}A)^{-1}\tilde{\mathbf{1}}y_n.$$

Thus, the stability function is

Stability Function, $R(\widehat{h})$

$$R(\widehat{h}) = 1 + \widehat{h}\tilde{\mathbf{b}}^T(I - \widehat{h}A)^{-1}\tilde{\mathbf{1}}.$$

As usual, the method is stable for $\widehat{h}$ such that $|R(\widehat{h})| \leq 1$.

For explicit methods, A strictly lower triangular, the quantity

$$\tilde{\mathbf{d}} = (I - \widehat{h}A)^{-1}\tilde{\mathbf{1}},$$

is easily computable using forward substitution.

The Stability Function $R(\widehat{h})$

As we have seen, the stability functions for explicit RK-methods are **polynomials...** Lets consider the stability analysis for a **semi-implicit** method defined by the following Butcher array:

$$\begin{array}{c|cc} c_1 & a_{1,1} & 0 \\ c_2 & a_{2,1} & a_{2,2} \\ \hline & b_1 & b_2 \end{array}$$

We get

$$\begin{aligned} k_1 &= f(t_n + c_1 h, y_n + h k_1 a_{1,1}) &&= \lambda y_n + \widehat{h} a_{1,1} k_1 \\ k_2 &= f(t_n + c_2 h, y_n + h k_1 a_{2,1} + h k_2 a_{2,2}) &&= \lambda y_n + \widehat{h}(a_{2,1} k_1 + a_{2,2} k_2) \end{aligned}$$

$$k_1 = \left[\frac{1}{1 - \widehat{h} a_{1,1}} \right] \lambda y_n, \quad k_2 = \left[\frac{1 - \widehat{h} a_{1,1} - \widehat{h} a_{2,1}}{(1 - \widehat{h} a_{1,1})(1 - \widehat{h} a_{2,2})} \right] \lambda y_n.$$

The Stability Function $R(\widehat{h})$ — Semi Implicit RK

With these values of k_1, k_2:

$$k_1 = \left[\frac{1}{1-\widehat{h}a_{1,1}}\right]\lambda y_n, \quad k_2 = \left[\frac{1-\widehat{h}a_{1,1}-\widehat{h}a_{2,1}}{(1-\widehat{h}a_{1,1})(1-\widehat{h}a_{2,2})}\right]\lambda y_n$$

the final step becomes

$$\begin{aligned} y_{n+1} &= y_n\left[1 + hb_1k_1 + hb_2k_2\right] \\ &= y_n \underbrace{\left[1 + \widehat{h}\left[\frac{b_1}{1-\widehat{h}a_{1,1}} + \frac{b_2(1-\widehat{h}a_{1,1}-\widehat{h}a_{2,1})}{(1-\widehat{h}a_{1,1})(1-\widehat{h}a_{2,2})}\right]\right]}_{R(\widehat{h})} \end{aligned}$$

Clearly, $R(\widehat{h})$ is a rational function.

We have seen that when we apply an RK-method to the test equation $y'(t) = \lambda y(t)$, we get the discrete iteration

$$y_{n+1} = R(\widehat{h})y_n, \quad \widehat{h} = h\lambda, \quad \lambda \in \mathbb{C},$$

where

- for explicit RK-methods $R(\widehat{h})$ is a polynomial, and
- for semi- (and fully) implicit RK-methods it is a rational function.

The exact solution to the test equation is

$$\mathbf{y}(\mathbf{t}) = \mathbf{K}\mathbf{e}^{\lambda\mathbf{t}}, \quad K \text{ constant (initial conditions)}$$

hence, the exact solution to the iteration is

$$y_{n+1}^{*} = e^{\lambda h} y_n = e^{\widehat{h}} y_n.$$

We can express the truncation error as:

$$\mathrm{LTE}(\widehat{h}) = \frac{y_{n+1}^{*} - y_{n+1}}{h} = \frac{1}{h}\left[e^{\widehat{h}} - R(\widehat{h})\right] y_n = \mathcal{O}\left(\widehat{h}^{p}\right),$$

for a p^{th} order method.

Clearly the truncation error

$$\mathrm{LTE}(\widehat{h}) = \frac{1}{h}\left[e^{\widehat{h}} - R(\widehat{h})\right] y_n = \mathcal{O}\left(\widehat{h}^p\right)$$

only depends on how well $R(\widehat{h})$ approximates the exponential $e^{\widehat{h}}$!!!

Hence, if we know how to find a good approximation to the exponential, we can back-track and build a high-order scheme (hopefully with good stability properties).

The optimal polynomial approximations come directly from the **Taylor expansion** of $e^{\widehat{h}}$:

$$e^{\widehat{h}} = \sum_{k=0}^{\infty} \frac{1}{k!} \widehat{h}^k.$$

We are now motivated to look at **Rational Approximations to the Exponential**[Math 541].

Value-Add (Strong Connection to Stability)

The value-add is that we are working directly with the stability function. Once we find high-order approximations to $e^{\widehat{h}}$ with desirable stability properties we go back and identify coefficients to build the corresponding finite-difference scheme.

Let

$$R_T^S(\widehat{h}) = \left[\sum_{i=0}^{S} a_i \widehat{h}^i\right] \Big/ \left[\sum_{j=0}^{T} b_j \widehat{h}^j\right] \quad a_0 = b_0 = 1, a_S \neq 0, b_T \neq 0$$

denote a rational approximation of $e^{\widehat{h}}$.

The maximum order of approximation of the exponential for a rational function $R_T^S(\widehat{h})$ is $T+S$:

$$e^{\widehat{h}} - R_T^S(\widehat{h}) = \mathcal{O}(\widehat{h}^{p+1}), \quad p \leq T+S$$

if $p = S+T$ then $R_T^S(\widehat{h})$ is called a **Padé Approximation** of $e^{\widehat{h}}$.

Butcher (1987) figured out what the coefficients for the Padé approximations (of e^x) are:

$$a_i = \frac{S!}{(S+T)!}\frac{(S+T-i)!}{i!(S-i)!}, \; i = 1, 2, \ldots, S$$

$$b_j = (-1)^j \frac{T!}{(S+T)!}\frac{(S+T-j)!}{j!(T-j)!}, \; j = 1, 2, \ldots, T$$

Examples: Some Padé Approximations — Order 3

$$R_3^0(\widehat{h}) = \frac{1}{1 - \widehat{h} + \frac{1}{2}\widehat{h}^2 - \frac{1}{6}\widehat{h}^3}$$

$$R_2^1(\widehat{h}) = \frac{1 + \frac{1}{3}\widehat{h}}{1 - \frac{2}{3}\widehat{h} + \frac{1}{6}\widehat{h}^2}$$

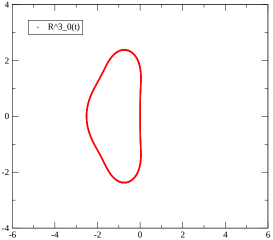
$$R_1^2(\widehat{h}) = \frac{1 + \frac{2}{3}\widehat{h} + \frac{1}{6}\widehat{h}^2}{1 - \frac{1}{3}\widehat{h}}$$

$$R_0^3(\widehat{h}) = 1 + \widehat{h} + \frac{1}{2}\widehat{h}^2 + \frac{1}{6}\widehat{h}^3$$

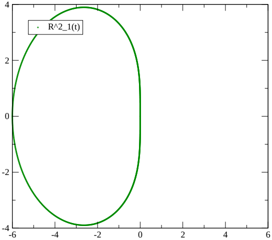
As usual, the boundaries of the stability regions are given by

$$R_T^S(\widehat{h}) = e^{i\theta}, \quad \theta \in [0, 2\pi)$$

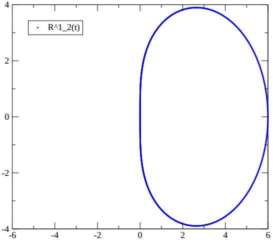
The Associated Stability Regions



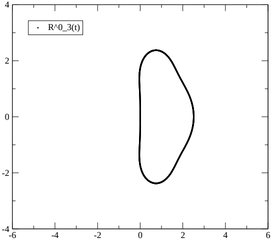
$R_0^3(\widehat{h})$ — interior



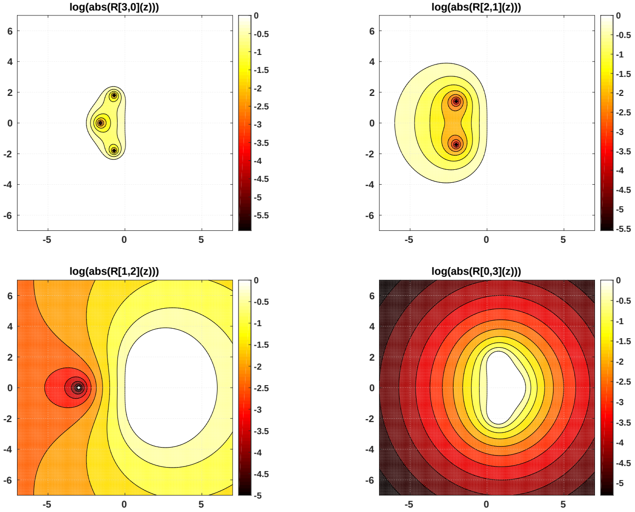
$R_1^2(\widehat{h})$ — interior



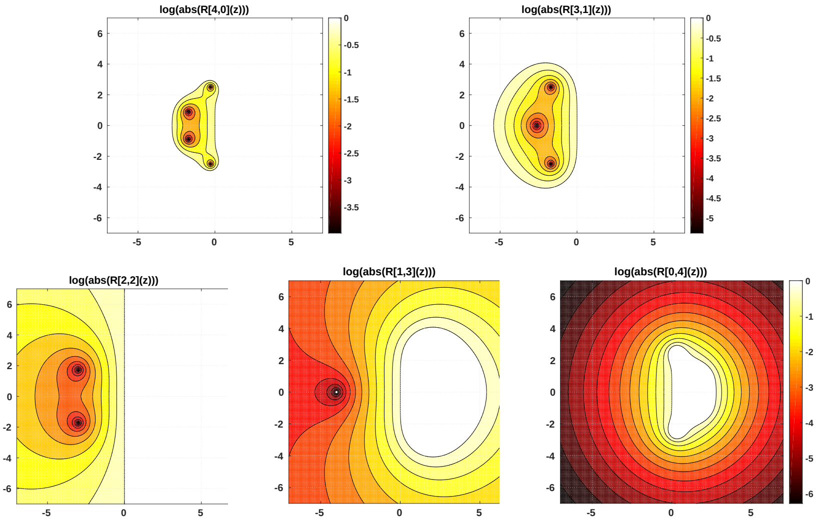
$R_2^1(\widehat{h})$ — exterior



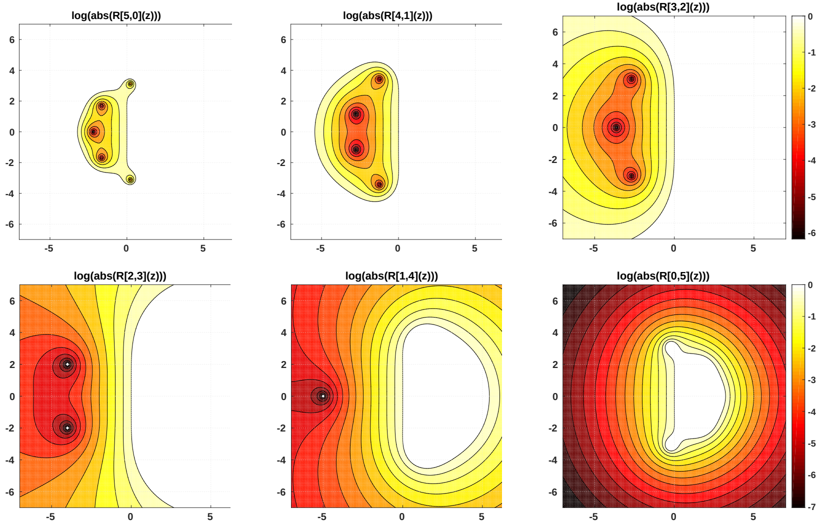
$R_3^0(\widehat{h})$ — exterior


log(abs(R[3,0](z)))
log(abs(R[2,1](z)))
log(abs(R[1,2](z)))
log(abs(R[0,3](z)))

The Associated Stability Regions with Magnitude



The Associated Stability Regions with Magnitude — Order 5



Definition (Ehle, 1969)

A rational approximation $R(\widehat{h})$ to $e^{\widehat{h}}$ is said to be:

1. **A-acceptable** if $|R(\widehat{h})| < 1$ whenever $Re(\widehat{h}) < 0$.
2. **$\mathbf{A}_0$-acceptable** if $|R(\widehat{h})| < 1$ whenever $\widehat{h}$ is real and negative.
3. **L-acceptable** if it is A-acceptable, and $|R(\widehat{h})| \to 0$ as $Re(\widehat{h}) \to -\infty$.

Clearly the associated numerical methods are A-stable, A_0-stable, and *L*-stable.

Theorems: Acceptability of Padé Approximations

Theorem (Varga, 1961)

If $T \geq S$, then $R_T^S(\widehat{h})$ is A_0-acceptable.

Theorem (Birkhoff and Varga, 1965)

If $T = S$, then $R_T^S(\widehat{h})$ is A-acceptable.

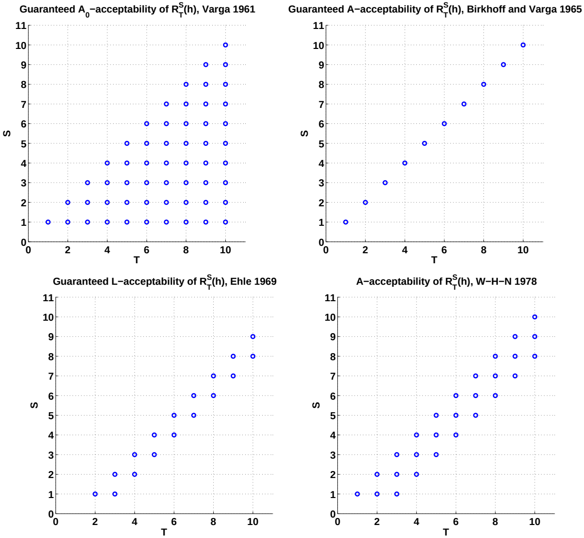
Theorem (Ehle, 1969)

If $T = S + 1$, or $T = S + 2$ then $R_T^S(\widehat{h})$ is L-acceptable.

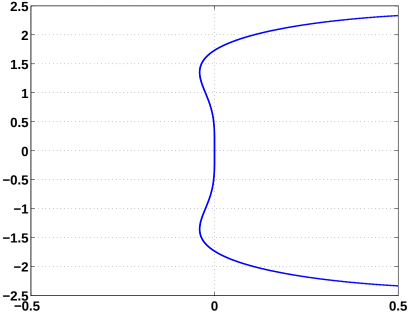
Theorem (Wanner, Hairer, Nørsett, 1978)

$R_T^S(\widehat{h})$ is A-acceptable if and only if $T - 2 \leq S \leq T$.
("The Ehle Conjecture" 1965)

Theorems — Visualized



Theorems — Note



Note: Even though $\lim_{\mathrm{Real}(\widehat{h})\to-\infty} |R_3^0(\widehat{h})| \to 0$, $R_3^0(\widehat{h})$ is `not` L-acceptable, since it is **not** A-acceptable; — The left-half-plane of the region of absolute stability has two small "cut-outs." It is $A(\alpha)$-acceptable, where $\alpha \approx \frac{\pi}{2} - 0.031$.

Given the preceding detour into approximation of the exponential, we are now ready to take another look at RK-methods.

Given an RK-method, with its associated Butcher array

$$\begin{array}{c|c} \tilde{\mathbf{c}} & A \\ \hline & \tilde{\mathbf{b}}^T \end{array}$$

we recall that we can express the stability function as

$$R(\widehat{h}) = 1 + \widehat{h}\tilde{\mathbf{b}}^T(I - \widehat{h}A)^{-1}\tilde{\mathbf{1}},$$

or

$$R(\widehat{h}) = \frac{\det[I - \widehat{h}(A - \tilde{\mathbf{1}}\tilde{\mathbf{b}}^T)]}{\det[I - \widehat{h}A]}.$$

Finding the RK-method from $R(\widehat{h})$

- Whereas it is possible, in some cases (but extremely tedious, in all cases) to take a rational function $R(\widehat{h})$ and *"reverse engineer"* a numerical method, this is not the path we will take.
- We are going to look at the fully implicit **Gauss** or **Gauss-Legendre Methods**:
- By **optimally** selecting the points where f is evaluated (the entries in the matrix A which occurs in the Butcher array), an s-stage Gauss method achieves order $2s$.

Note: The optimal placement of the (time, $\vec{c}$) points comes directly from the analysis for Gaussian numerical integration[Math 541].

Since there is a **unique** $R_S^S(\widehat{h})$ rational approximation to order $2s$ of $e^{\widehat{h}}$, namely the Padé approximation, it follows that the stability function for the Gauss methods must be the Padé approximation.

Since $S = T$ all Gauss methods are A-stable (Birkhoff-Varga).

Example ("Implicit Mid-point Rule.")

The "Implicit Mid-point Rule" is a 1-stage 2nd-order Gauss method:

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

$$y_{n+1} = y_n + hf\left(t_n + \frac{1}{2}h, \frac{1}{2}(y_n + y_{n+1})\right)$$

Example (2-stage 4th order Gauss method)

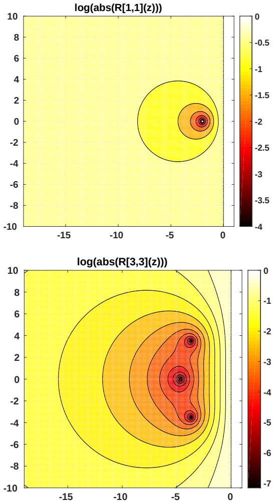
$$
\begin{array}{c|cc}
\frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\
\frac{3+\sqrt{3}}{6} & \frac{3+2\sqrt{3}}{12} & \frac{1}{4} \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}
$$

Example (3-stage 6th order Gauss method)

$$
\begin{array}{c|ccc}
\frac{5-\sqrt{15}}{10} & \frac{5}{36} & \frac{10-3\sqrt{15}}{45} & \frac{25-6\sqrt{15}}{180} \\
\frac{1}{2} & \frac{10+3\sqrt{15}}{72} & \frac{2}{9} & \frac{10-3\sqrt{15}}{72} \\
\frac{5+\sqrt{15}}{10} & \frac{25+6\sqrt{15}}{180} & \frac{10+3\sqrt{15}}{45} & \frac{5}{36} \\
\hline
 & \frac{5}{18} & \frac{4}{9} & \frac{5}{18}
\end{array}
$$

Ponder how much fun would it be to reverse engineer this 3-6 method from the Padé approximation

$$
R_3^3(\widehat{h}) = \frac{1 + \frac{1}{2}h + \frac{1}{10}\widehat{h}^2 + \frac{1}{120}\widehat{h}^3}{1 - \frac{1}{2}h + \frac{1}{10}\widehat{h}^2 - \frac{1}{120}\widehat{h}^3}
$$


log(abs(R[1,1](z)))
log(abs(R[2,2](z)))
log(abs(R[3,3](z)))
log(abs(R[4,4](z)))
10
8
6
4
2
0
-2
-4
-6
-8
-10
-15
-10
-5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-5
-5.5
-6
-7
-8

Gauss(-Legendre) Methods — The Final Wish

- If want to find something "wrong" with the Gauss methods, it would be that they are **not L-stable.**
- It turns out we can trade one order of approximation for L-stability. The **Radau I-A** and **Radau II-A** s-stage methods are order $(2s-1)$ and L-stable.
- The Radau I-A methods are derived just like the Gaussian methods, but require the left endpoint to be part of the interval ($c_1 = 0$).
- The Radau II-A methods require the right endpoint to be part of the interval ($c_1 = 1$).

Example (1-stage 1st order Radau II-A L-stable method)

$$
\begin{array}{c|c}
1 & 1 \\
\hline
 & 1
\end{array}
$$

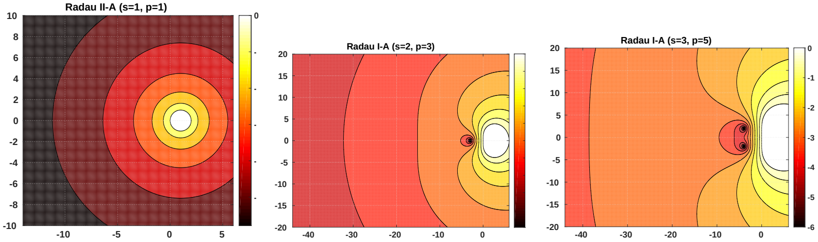
Example (2-stage 3rd order Radau I-A L-stable method)

$$
\begin{array}{c|cc}
0 & \frac{1}{4} & -\frac{1}{4} \\
\frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\
\hline
 & \frac{1}{4} & \frac{3}{4}
\end{array}
$$

Example (3-stage 5th order Radau I-A L-stable method)

$$
\begin{array}{c|ccc}
0 & \frac{1}{9} & \frac{-1-\sqrt{6}}{18} & \frac{-1+\sqrt{6}}{18} \\
\frac{6-\sqrt{6}}{10} & \frac{1}{9} & \frac{88+7\sqrt{6}}{360} & \frac{88-43\sqrt{6}}{360} \\
\frac{6+\sqrt{6}}{10} & \frac{1}{9} & \frac{88+43\sqrt{6}}{360} & \frac{88-7\sqrt{6}}{360} \\
\hline
 & \frac{1}{9} & \frac{16+\sqrt{6}}{36} & \frac{16-\sqrt{6}}{36}
\end{array}
$$

Radau Methods — Some Stability Regions Visualized



- Clearly, constructing A- or L-stable implicit RK-methods is not an insurmountable task.
- Further, implementing the methods is also quite straight-forward.
- Either with the help of Richardson Extrapolation or by RKF45-like methods we can get good error estimates, and thus construct adaptive algorithms that change the step-size h on the fly.

- These methods will work and can be designed to be very robust.
- However, in terms of **efficiency** they fall short of fine-tuned BDF (LMM) methods.
- To make RK-methods competitive, the computational handling of the implicitness must be cut down. There are a number of "tricks" — transformations that can be applied to reduce the computational burden.

Next couple of lectures...

- Linear Multistep Methods for Stiff ODEs.
- Review and examples.
- Hybrid Methods.
- Tie up loose ends.
- Start thinking about projects....