Lecture 24

Ordinary Differential Equations

Owen L. Lewis

Department of Mathematics and Statistics
University of New Mexico

November 21, 2024

Outline

- Review ODEs
- Single Step Methods
  - Euler's method (1st order accurate)
  - Runge-Kutta methods (higher order accuracy)
  - Accuracy and stability of the methods

Differential Equations

- Its hard to model the state of a system
  - Temperature in the room
  - Pressure on the wing of an airplane
  - Concentration of ions in a solvent
  - $\cdots$ and a million more examples $\cdots$
- However, it's much simpler to model the rate of change of the state:
  $\Rightarrow$ Differential equations
- Could involve change w.r.t to one variable (e.g. time):
  $\Rightarrow$ Ordinary Differential Equations
- Could involve change w.r.t multiple variables (e.g. x and y axis):
  $\Rightarrow$ Partial Differential Equations

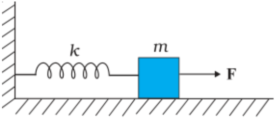
Ordinary Differential Equations (ODEs)

An ODE involves

- One function $u(t)$
- Its derivatives $u^{'}(t)$, $u^{''}(t)$, $u^{'''}(t)$, etc
- Order of ODE is the higest order derivative

Examples:

1. Generic first-order ODE
$$u' = f(t, u(t))$$
2. Harmonic Oscillator (Second-order ODE)
$$mu^{''}(t) + cu'(t) + ku(t) = f(t)$$



m: mass, c: damping coeff., k: elastic stiffness coeff, $f(t)$: external force

Notes

To the Board!

Initial Value Problems (IVPs)

- ODEs in previous examples have infinitely many solutions
- Get unique solution by specifying initial conditions (ICs)

  IVP = ODE + IC

  Examples:

  1.

$$\begin{cases} u' = f(t, u(t)), & t \in (a, b] \\ u(a) = u_a \end{cases}$$

  2.

$$\begin{cases} mu''(t) + cu'(t) + ku(t) = f(t), & t \in (0, T] \\ u(0) = A, u'(0) = B \end{cases}$$

- Furthermore for Example 1, a continuous solution is guaranteed if the functions $f(t, u(t))$ and $\partial f / \partial u$ are continuous over a sufficiently large domain of t and u values. (See Theorem 6.2, page 291, 1st Ed.)

Numerical Solution of IVPs

- Many important ODEs have no closed form solution
- Analytical methods of limited use in such scenarios
- Approximate the solution using numerics
- Consider first order ODE

$$\begin{cases} u' = f(t, u(t)), \quad t \in (a, b] \\ u(a) = u_a \end{cases}$$

- Discretize $[a, b]$ into equi-spaced points:

$$a = t_0 < t_1 < t_2 < \cdots < t_{n-1} < t_n = b$$

with $h = t_i - t_{i-1} = (b - a)/n$

Notes

Euler's method

- Simplest numerical methods to solve IVPs
- Taylor's theorem

$$u(t+h) = u(t) + hu'(t) + (h^2/2)u''(c), \quad t < c < t+h$$

- Let u_i be approximation to exact solution $u(t_i)$
- Drop $O(h^2)$ terms and utilize $u'(t) = f(t,u)$

Listing 1: Euler Algorithm

```
u_0 = u_a
for i = 0 to n-1
      u_{i+1} = u_i + h f(t_i, u_i)
```

Euler Example

Apply Euler's method with $h = 0.2$ to the IVP

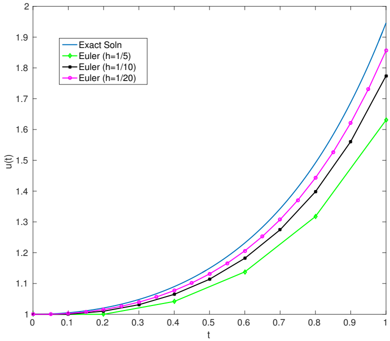
$$\begin{cases} u' = tu + t^3, & t \in [0,1] \\ u(0) = 1 \end{cases}$$

- $h = (b-a)/n \implies n = 5.$

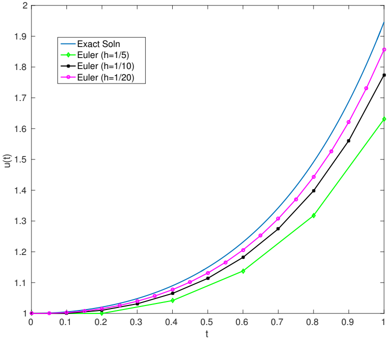
Step i	t_i	u_i	$f(t_i, u_i)$	$u_{i+1} = u_i + hf(t_i, u_i)$
0	0	1	0	1
1	0.2	1	0.2080	1.0416
2	0.4	1.0416	0.4806	1.1377
3	0.6	1.1377	0.8986	1.3175
4	0.8	1.3175	1.5660	1.6306
5	1	1.6306		

Euler Example

- Exact solution is
$$u(t) = 3e^{t^2/2} - t^2 - 2$$
- Euler method with $h = 0.2, 0.1, 0.05$
- As h decreases, approximation *converges* to exact solution



Error?

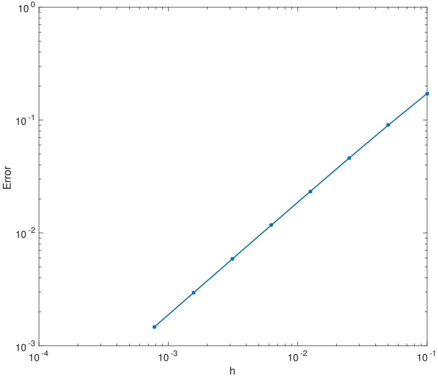


How do I measure error between a function and a collection of data points?

In a way that doesn't depend on h?

Accuracy of Euler's method

- Compute the error at final time, $t = 1$
- Error, $e = O(h^p)$
- `loglog` plot



$\Rightarrow$ p appears to be 1

Accuracy of Euler's method

- Compute the error at final time, $t = 1$
- Error, $e = O(h^p)$

$$p \approx \frac{\log(e_k/e_{k-1})}{\log(h_k/h_{k-1})}$$

Experiment k	e_k	e_{k-1}	h_k	h_{k-1}	p
0	0.1718	-	1/10	-	-
1	0.08991	0.1718	1/20	1/10	0.93
2	0.04603	0.08991	1/40	1/20	0.96
3	0.02329	0.04603	1/80	1/40	0.98
4	.01172	0.02329	1/160	1/80	0.99

- $p \approx 1$
- Euler's method is **first-order** accurate method
- Can theoretically show this too

Accuracy of Euler's method: Some Theory

- Recall the derivation using Taylor's thm
- Exact solution

$$u(t+h) = u(t) + hf(t,u) + O(h^2)$$

- Numerical soln

$$u_{i+1} = u_i + hf(t_i, u_i)$$

- Assume $u(t) = u_i$. Then the "local error", $|u(t+h) - u_{i+1}|$ is

$$|u(t+h) - u_{i+1}| = O(h^2)$$

- "Global error" is the error at final time: $u(t_n) - u_n$
- Local errors accumulate to give $O(nh^2) = O(h)$ accuracy since $nh = (b-a)$.

Systems of ODEs

A first-order system of m ODEs has the form

$$\begin{cases} u_1' = f_1(t, u_1, u_2, \ldots, u_m) \\ u_2' = f_2(t, u_1, u_2, \ldots, u_m) \\ \quad\vdots \\ u_m' = f_m(t, u_1, u_2, \ldots, u_m) \end{cases} \qquad t \in (a, b]$$

In IVPs, each variable has its own initial condition:

$$\begin{cases} u_1(a) = u_{1a} \\ u_2(a) = u_{2a} \\ \quad\vdots \\ u_m(a) = u_{ma} \end{cases}$$

Euler method for first-order system of ODEs

- Apply scalar Euler's method to each component
- Example: apply Euler's method to

$$\begin{cases} u_1' = -2u_1 + u_2^2 = f_1(t, u_1, u_2), & u_1(0) = 0 \\ u_2' = -tu_1^2 \qquad\quad = f_2(t, u_1, u_2), & u_2(0) = 1 \end{cases}$$

- Euler's method:

$$\begin{cases} u_{1,i+1} & = u_{1,i} + h\, f_1(t, u_1, u_2) \\ & = u_{1,i} + h(-2u_{1,i} + u_{2,i}^2) \\ u_{2,i+1} & = u_{2,i} + h\, f_2(t, u_1, u_2) \\ & = u_{2,i} + h(-t_i u_{1,i}^2) \end{cases}$$

Higher order equations

- Convert higher order ODE to system of first order ODEs
- Consider

$$u'' + au' + btu = c$$

- Introduce

$$\begin{cases} y_1 = u \\ y_2 = u' \end{cases}$$

- This leads to the following first-order ODE system

$$\begin{cases} y_1' = y_2 \\ y_2' = -ay_2 - bty_1 + c \end{cases}$$

- Now apply Euler's method! We get approximation to $y_1 = u$.

Notes

Example: Euler's method for higher order equations

Solve the following IVP using Euler's method:

$$\begin{cases} u''' = (u'')^2 - uu' + \sin t, \\ u(0) = 1, u'(0) = 0, u''(0) = 2 \end{cases}$$

Convert to first-order system: $y_1 = u, y_2 = u', y_3 = u''$

$$\begin{cases} y_1' = y_2, \qquad y_1(0) = 1 \\ y_2' = y_3, \qquad y_2(0) = 0 \\ y_3' = y_3^2 - y_1 y_2 + \sin t, \qquad y_3(0) = 2 \end{cases}$$

Euler's method:

$$\begin{cases} y_{1,i+1} = y_{1,i} + h y_{2,i} \\ y_{2,i+1} = y_{2,i} + h y_{3,i} \\ y_{3,i+1} = y_{3,i} + h(y_{3,i}^2 - y_{1,i} y_{2,i} + \sin t_i) \end{cases}$$

Runge-Kutta Methods (RK)

- Runge-Kutta methods are popular methods to solve IVPs
- Euler method is a “single-stage” RK method (RK1)
- Higher order methods derived using higher order Taylor series, higher order integration techniques etc.
- Higher order methods have more “stages”

Notes

Two-stage RK methods

- Integrate both sides of $u'(t) = f(t, u(t))$ from t_i to t_{i+1}

$$u(t_{i+1}) - u(t_i) = \int_{t_i}^{t_{i+1}} f(t, u(t))\, dt$$

- Now apply (simple) trapezoidal rule on the RHS

$$u(t_{i+1}) - u(t_i) = \frac{h}{2}[f(t_i, u(t_i)) + f(t_{i+1}, u(t_{i+1}))] + O(h^3)$$

- Let $u_i \approx u(t_i)$

$$u_{i+1} = u_i + \frac{h}{2}[f(t_i, u_i) + f(t_i + h, u_{i+1})] + O(h^3)$$

Approx u_{i+1} on RHS by Euler's method

$$u_{i+1} = u_i + \frac{h}{2}\left[f(t_i, u_i) + f\left(t_i + h, u_i + hf(t_i, u_i) + O(h^2)\right)\right] + O(h^3)$$

RK2a method

- Drop $O(h^3)$ terms. Assume $O(h^2)$ combines with $h/2$.

$$u_{i+1} = u_i + \frac{h}{2}\left[f(t_i, u_i) + f\left(t_i + h, u_i + hf(t_i, u_i) + O(h^2)\right)\right] + O(h^3)$$

- Get method called RK2a

$$u_{i+1} = u_i + \frac{h}{2}[f(t_i, u_i) + f(t_i + h, u_i + hf(t_i, u_i))]$$

- Written as a two-stage method

RK2a

$$\begin{aligned} K_1 &= f(t_i, u_i) \\ K_2 &= f(t_i + h, u_i + hK_1) \\ u_{i+1} &= u_i + \frac{h}{2}(K_1 + K_2) \end{aligned}$$

- K_1 and K_2 are called stages.
- RK2a has has local error of $O(h^3)$ for a global error of $O(h^2)$.

RK2b method

- Integrate both sides of $u'(t) = f(t, u(t))$ from t_i to t_{i+1}

$$u(t_{i+1}) - u(t_i) = \int_{t_i}^{t_{i+1}} f(t, u(t))\, dt$$

- Now apply midpoint rule on the RHS

$$u(t_{i+1}) - u(t_i) = h[\, f(t_i + h/2,\ u(t_i + h/2))\,] + O(h^3)$$

- Let $u_i \approx u(t_i)$ and $u_{i+1/2} \approx u(t_i + h/2)$

$$u_{i+1} - u_i = h[\, f(t_i + h/2,\ u_{i+1/2})\,] + O(h^3)$$

- Use Euler's method to approx $u_{i+1/2} = u_i + (h/2)\, f(t_i, u_i) + O(h^2)$
- Get RK2b method by dropping $O(h^3)$ error terms

$$u_{i+1} = u_i + hf(t_i + h/2,\ u_i + (h/2)\, f(t_i, u_i))$$

RK2b method

- Two-stage RK2b method

RK2b

$$
\begin{aligned}
K_1 &= f(t_i, u_i) \\
K_2 &= f(t_i + h/2,\ u_i + (h/2)\, K_1) \\
u_{i+1} &= u_i + hK_2
\end{aligned}
$$

- Second order accurate as well

RK4 method

- Four stages
- Derived using Simpson's rule with two subintervals $[t_i, t_i + h/2]$ and $[t_i + h/2, t_i + h]$

RK4

$$
\begin{aligned}
K_1 &= f(t_i, u_i) \\
K_2 &= f(t_i + h/2,\ u_i + (h/2)\,K_1) \\
K_3 &= f(t_i + h/2,\ u_i + (h/2)\,K_2) \\
K_4 &= f(t_i + h, u_i + hK_3) \\
u_{i+1} &= u_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)
\end{aligned}
$$

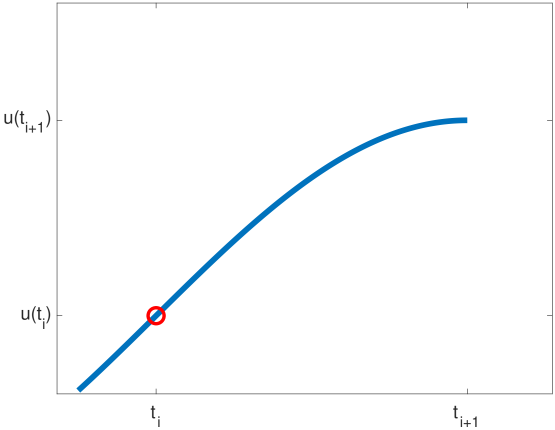
- Fourth order accurate
- Arguably the most widely use ODE integrator

Geometric Interpretation of RK methods

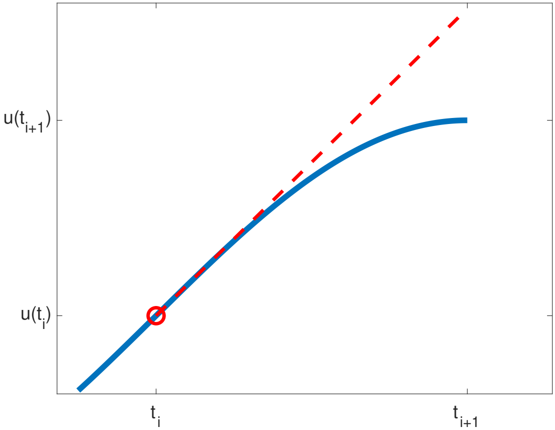
- Euler (or RK1) uses the slope at t_i: $u_i^{'} = f(t_i, u_i)$
- RK2a uses the slopes at t_i and t_{i+1} which are K_1 and K_2
- RK2b uses the slopes at t_i and $t_{i+\frac{1}{2}}$ which are K_1 and K_2
- RK4 uses the slopes

$$\begin{array}{ccc} t_i & \rightarrow & K_1 \\ t_{i+\frac{1}{2}} & \rightarrow & \frac{K_2+K_3}{2} \\ t_{i+1} & \rightarrow & K_4 \end{array}$$

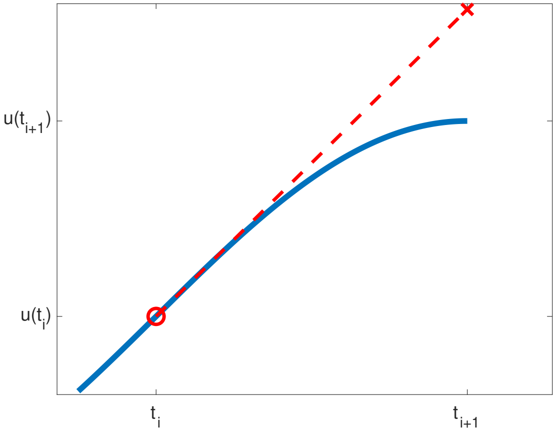
Euler



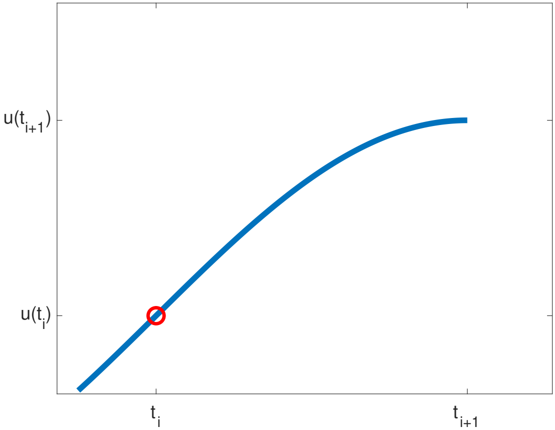
Euler



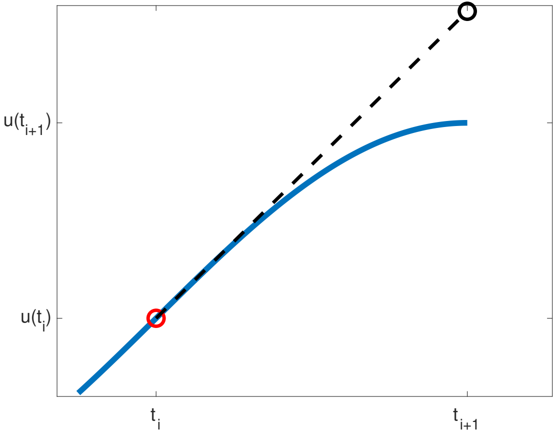
Euler



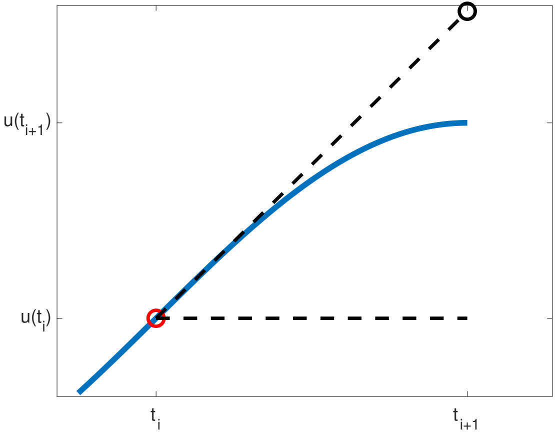
RK2a



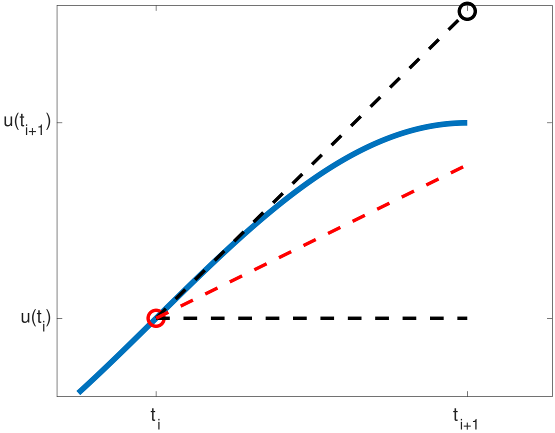
RK2a



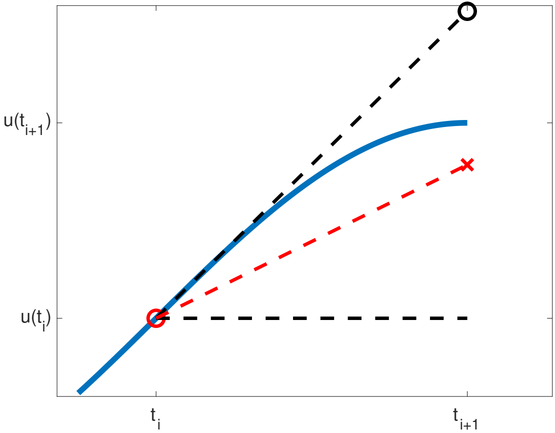
RK2a



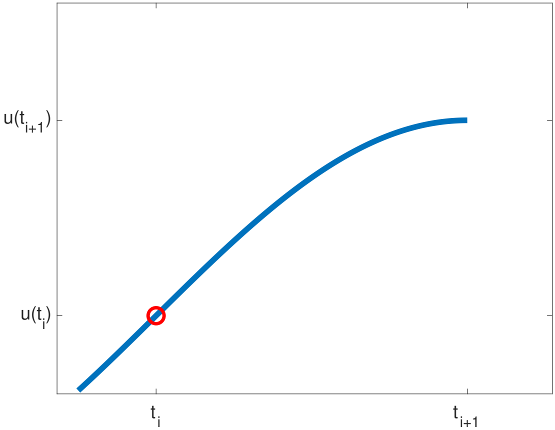
RK2a



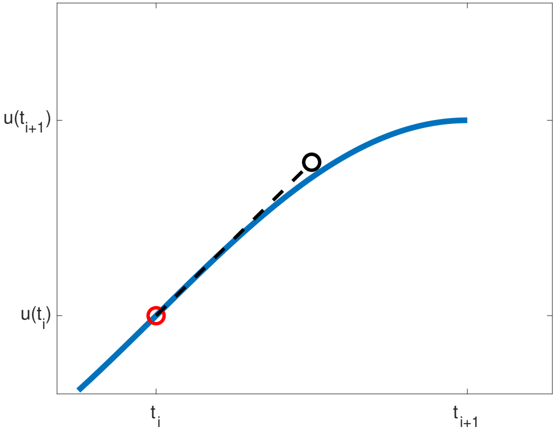
RK2a



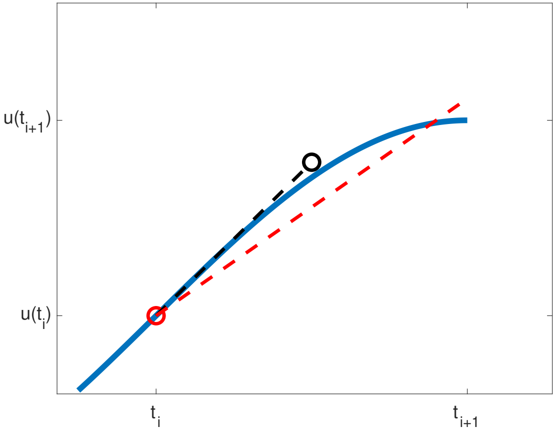
RK2b



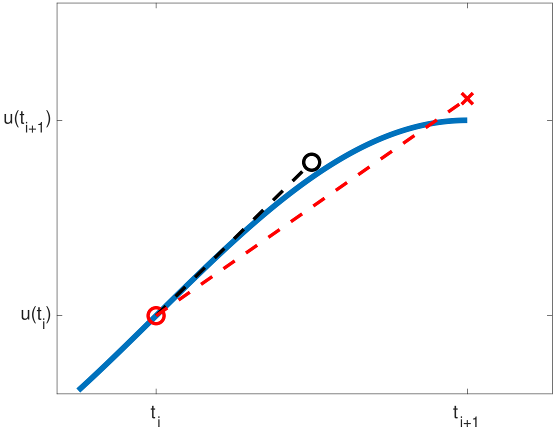
RK2b



RK2b



RK2b



Geometric Interpretation of RK methods

- Euler (or RK1) uses the slope at t_i: $u_i^{'} = f(t_i, u_i)$
- RK2a uses the slopes at t_i and t_{i+1} which are K_1 and K_2
- RK2b uses the slopes at t_i and $t_{i+\frac{1}{2}}$ which are K_1 and K_2
- RK4 uses the slopes

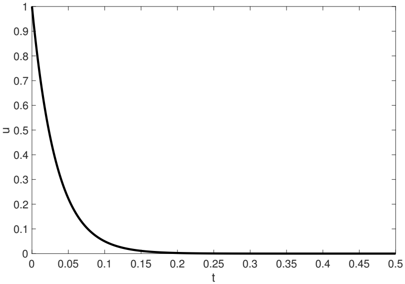
$$\begin{array}{ccc} t_i & \rightarrow & K_1 \\ t_{i+\frac{1}{2}} & \rightarrow & \frac{K_2+K_3}{2} \\ t_{i+1} & \rightarrow & K_4 \end{array}$$

Stability: Euler method on a model problem

- Model problem (Dahlquist model problem)

$$u'(t) = -\lambda u(t), \qquad \lambda > 0, \quad t \in [0, T]$$
$$u(0) = u_0$$

Exact solution is: $u(t) = u_0 e^{-\lambda t}$



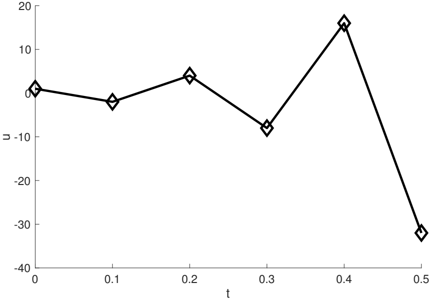
- Solution u decays exponentially as t increases

Stability: Euler method on a model problem

- Euler method for the model problem:

$$u_{i+1} = u_i + h(-\lambda u_i) = (1 - h\lambda)u_i, \qquad i = 0, 1, 2, \ldots$$

- For example, choose $\lambda = 30$, $u_0 = 1$, $h = 0.1$



- Instead of exponential decay, the approximate solution has a growing oscillatory behavior!

What went wrong?

- Euler method converges when h is small enough
- What about when h is not “sufficiently” small?
- “Local accuracy” of Euler method is 2nd order from Taylor series
- But these local errors can **propagate and increase exponentially**!

- **Stability** quantifies the propagation of errors
- In the example above, we chose a step size for which Euler method was not stable

Notes

Amplification factor

- Euler method was $u_{i+1} = (1 - h\lambda)u_i$

$$\frac{u_{i+1}}{u_i} = 1 - h\lambda \qquad \Longrightarrow \qquad G_i = \left|\frac{u_{i+1}}{u_i}\right| = |1 - h\lambda|$$

- $G_i = |u_{i+1}/u_i|$ is called the **amplification factor** at time step i
- For Euler method, $G_i = |1 - h\lambda|$
- If the amplification factor is less than one, then the numerical solution does not grow without bound. We say the numerical solution is **stable**.

Amplification factor of Euler Method

- For Euler method, we need for stability

$$G_i < 1 \implies |1 - h\lambda| < 1 \implies -1 < 1 - h\lambda < 1 \implies 0 < \lambda h < 2$$

- This gives a step size restriction $0 < h < 2/\lambda$
- In the earlier example $h = 0.1$ and $\lambda = 30$
  So, $h\lambda = 3 > 2 \implies$ unstable

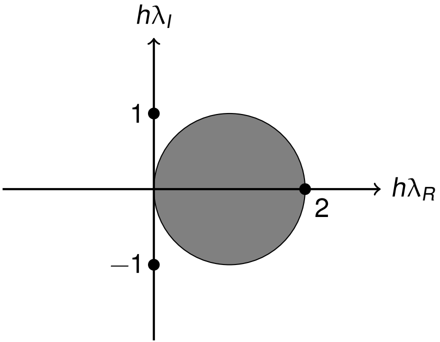
Region of Stability

- Let
$$\lambda = \lambda_R + i\lambda_I$$
with $\lambda_R > 0$
- Why complex number?
For systems of ODEs, λ represents the eigenvalues of the coefficient matrix. And, remember that the eigenvalues can be complex even if the matrix is real.
- **Region of stability:** Region in the complex plane, where λh implies method is stable.
$\Rightarrow$ Plot $h\lambda_R$ on the x-axis, and $h\lambda_I$ on the y-axis
- That is, the region in which the amplification factor $G_i < 1$

Region of Stability for Euler's method

$$G_j = |1 - h\lambda| = |1 - h(\lambda_R + i\lambda_I)|$$
$$= |(1 - h\lambda_R) - ih\lambda_i| = \sqrt{(1 - h\lambda_R)^2 + (h\lambda_I)^2} < 1$$

- Region of stability is circle with center $(0, 1)$ and radius 1:



Region of Stability for RK2b

- For the model problem we get

$$K_1 = -\lambda u_j, \qquad K_2 = -\lambda(u_j - \frac{h}{2}\lambda u_j)$$

$$u_{j+1} = u_j(1 - h\lambda + \frac{1}{2}(h\lambda)^2)$$

- Amplification factor

$$G_j = \left|\frac{u_{j+1}}{u_j}\right| = \left|1 - (h\lambda) + \frac{1}{2}(h\lambda)^2\right| < 1$$

- Want all $z = h\lambda$ in the complex place such that

$$\left|1 - z + \frac{1}{2}z^2\right| < 1$$

- Or, find all z such that for all $\phi \in [0, 2\pi]$

$$1 - z + \frac{1}{2}z^2 < e^{i\phi}$$

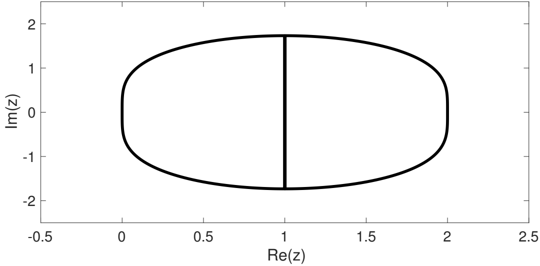
Region of Stability for RK2b

- Solve the quadratic equation

$$\frac{1}{2}z^2 - z + (1 - e^{i\phi}) = 0$$

- Two solutions

$$z_{1,2} = 1 \pm \sqrt{ze^{i\phi} - 1}$$



- Region of stability is the region inside the curves

Region of Stability for RK4

- After similar derivations as earlier we get

$$G_j = \left| \frac{u_{j+1}}{u_j} \right| = \left| 1 - (h\lambda) + \frac{1}{2}(h\lambda)^2 - \frac{1}{6}(h\lambda)^3 + \frac{1}{24}(\lambda h)^4 \right|$$

- Want $z = h\lambda$ such that

$$G(z) = \left| 1 - z + \frac{1}{2}z^2 - \frac{1}{6}z^3 + \frac{1}{24}z^4 \right| < 1$$

- Plot $G_j(z)$ for all possible z values and then plot the 1-contour of G_j

Listing 2: RK4 Stability

```
[x,y] = meshgrid(linspace(-4,4,1000), linspace(-4,4,1000));
z = x + 1i*y;
contour(x,y,abs(1 - z + (1/2)*z.^2 - (1/6)*z.^3 + (1/24)*z.^4 )
    ,[1, 1], 'k-'  );
```

Region of Stability for RK4