Overview

- Motivation
- Euler's method
- Runge-Kutta method
- Applications:
  - SIR model
  - Van der Pol oscillator
  - Double pendulum

- Many differential equations of practical interest are nonlinear and do not have a closed-form analytical solution.
- Examples:

  Van der Pol oscillator $x'' - \mu(1-x^2)x' + x = 0$

  Double pendulum

$$\dot{\theta}_1 = \frac{6}{m l^2} \frac{2 p_1 - 3\cos(\theta_1-\theta_2)\, p_2}{16 - 9\cos^2(\theta_1-\theta_2)} \qquad \dot{\theta}_2 = \frac{6}{m l^2} \frac{8 p_1 - 3\cos(\theta_1-\theta_2)\, p_2}{16 - 9\cos^2(\theta_1-\theta_2)}$$

$$\dot{p}_1 = -\frac{m l^2}{2}\left(\dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1-\theta_2) + 3\frac{g}{l}\sin\theta_1\right) \quad \dot{p}_2 = \frac{m l^2}{2}\left(\dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1-\theta_2) - \frac{g}{l}\sin\theta_1\right)$$

- Though apparently complicated such systems are of the general form $\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y})$
- Methods for systems are essentially the same as those for the scalar DE $y' = f(x, y)$
- We'll study a few key concepts on numerical solution of $y' = f(x, y)$

Basic idea

- IVP: $y' = f(x, y)$, $y(x_0) = y_0$, find solution over interval $[x_0, x_n]$
- Discretize the interval with *step size* $h = (x_n - x_0)/n$, $x_k = x_0 + kh, k = 0, \ldots, n$
- The slope is sampled at points (x_k, y_k), notation $f_k = f(x_k, y_k)$
- The derivative is approximated, e.g., Euler's method $y_k \cong y(x_k)$, $y'_k \cong y'(x_k)$

$$y'(x_k) = f(x_k, y(x_k)) \cong y'_k = \frac{y_{k+1} - y_k}{h} = f(x_k, y_k) \Rightarrow$$

$$y_{k+1} = y_k + h f(x_k, y_k)$$

- An approximate solution is built up iteratively

$$\begin{aligned}
y_1 &= y_0 + h f(x_0, y_0) \\
y_2 &= y_1 + h f(x_1, y_1) \\
&\vdots \\
y_n &= y_n + h f(x_n, y_n)
\end{aligned}$$

- *Truncation error*. Since $\frac{y_{k+1}-y_k}{h}$ is an approximation of y'_k, an error is introduced at each step of the iteration. The truncation error can be estimated from a Taylor series

$$
\begin{aligned}
& y_{k+1} = y_k + h\,f(x_k, y_k) \Rightarrow y(x_{k+1}) = y(x_k) + h\,f(x_k, y_k) + \tau_k \\
& \text{Error: } e_k = y(x_k) - y_k = \text{exact} - \text{approximation (Note: } e_0 = 0) \\
& \text{Since } x_{k+1} = x_k + h\text{:} \\
& y(x_{k+1}) = y(x_k) + y'(x_k)\,h + \frac{1}{2}y''(\xi_k)h^2 \\
& e_{k+1} = e_k + \frac{1}{2}y''(\xi_k)h^2
\end{aligned}
$$

- Cummulative error (global truncation error)

$$
\left.
\begin{aligned}
& e_1 = e_0 + \frac{1}{2}y''(\xi_0)h^2 \\
& e_2 = e_1 + \frac{1}{2}y''(\xi_1)h^2 \\
& \vdots \\
& e_n = e_{n-1} + \frac{1}{2}y''(\xi_{n-1})h^2
\end{aligned}
\right.
\quad \Rightarrow e_n = \sum_{i=0}^{n-1} \frac{1}{2}y''(\xi_i)h^2 \leqslant \max|y''|\frac{x_n - x_0}{2}h = K h
$$

Basic analysis of Euler's method: stability

- Numerical methods use floating point numbers, a finite-precision approximation of reals
- A numerical method that amplifies inherent *rounding error* is unstable, e.g.,$\pi - 3.141592 = e$
- Consider $y' = f(y) \cong \lambda y$ approximated by Euler's method

$$\left.\begin{array}{l} y_{k+1} - y_k = \lambda h y_k \Rightarrow y_{k+1} = (1 + \lambda h)\, y_k \\ y(x_{k+1}) = y(x_k) + \lambda h y(x_k) + \frac{1}{2} y''(\xi_k) h^2 \end{array}\right. \Rightarrow e_{k+1} = (1 + \lambda h) e_k + \frac{1}{2} y''(\xi_k) h^2$$

- Euler's method error

$$\left.\begin{array}{l} e_1 = (1 + \lambda h) e_0 + \frac{1}{2} y''(\xi_0) h^2 \\ e_2 = (1 + \lambda h) e_1 + \frac{1}{2} y''(\xi_1) h^2 \\ \vdots \\ e_n = (1 + \lambda h) e_{n-1} + \frac{1}{2} y''(\xi_{n-1}) h^2 \end{array}\right. \Rightarrow e_n = (1 + \lambda h)^{n-1} e_1 + \dots$$

- If $|1 + \lambda h| > 1$ errors get amplified, Euler's method is unstable, $|1 + \lambda h| \leqslant 1$ stable

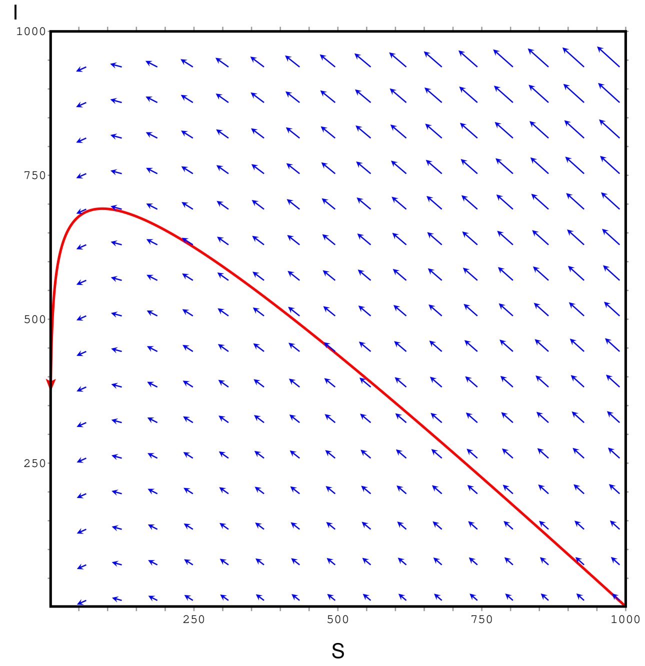
- Euler's method approximates the average slope over interval $[x_k, x_{k+1}]$ by $f(x_k, y_k)$, i.e., at the start of the interval. $y(x_{k+1}) = y(x_k) + y'(\xi_k)h$
- Runge-Kutta idea: approximate the average slope by a weighted average

$$\begin{aligned}
k_1 &= h\,f(x_i, y_i)\\
k_2 &= h\,f\left(x_i + \frac{h}{2}, y_i + \frac{1}{2}k_1\right)\\
k_3 &= h\,f\left(x_i + \frac{h}{2}, y_i + \frac{1}{2}k_2\right)\\
k_4 &= h\,f(x_i + h, y_i + k_3)\\
y_{i+1} &= y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}$$

- The above method is one particular example of a large class of Runge-Kutta algorithms
- The (global) truncation error is $e_n = Kh^4$, much more accurate than that for Euler's method ($e_n = Lh$) since h can be considered small (subunitary)

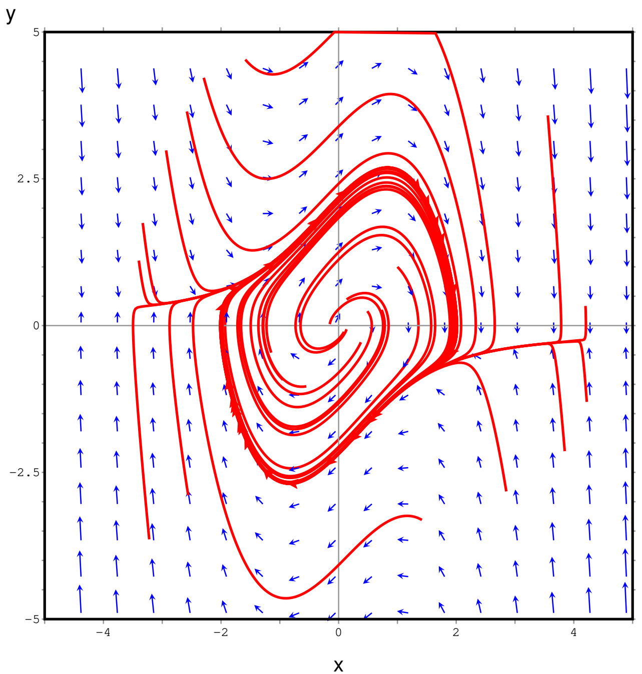
- Recall the SIR epidemic model: $S' = -\beta I S$, $I' = \beta I S - \gamma I$, $R' = \gamma I$

```
(%i2) plotdf([-beta*I*S,beta*I*S-gamma*I],[S,I],
      [trajectory_at,1000,1],
      [S,1,1000],[I,1,1000],
      [direction,forward],
      [parameters,"beta=0.0001,gamma=0.01"],
      [sliders,"beta=.00001:.01,gamma=0.01:.1"],
      [versus_t,1])$
(%i3)
```



- $x'' - \mu(1-x^2)x' + x = 0 \Rightarrow$

$$x' = y,\ y' = \mu(1-x^2)y - x$$




```
(%i4) plotdf([y,mu*(1-x^2)*y-x],[x,y],
      [trajectory_at,1,1],
      [x,-5,5],[y,-5,5],
      [direction,forward],
      [parameters,"mu=1"],[sliders,"mu=-1:1"],
      [versus_t,1])$
(%i5)
```

Application 3: Double pendulum

- $$\dot{\theta}_1 = \frac{6}{m\,l^2}\frac{2\,p_1 - 3\cos(\theta_1-\theta_2)\,p_2}{16 - 9\cos^2(\theta_1-\theta_2)} \quad \dot{\theta}_2 = \frac{6}{m\,l^2}\frac{8\,p_1 - 3\cos(\theta_1-\theta_2)\,p_2}{16 - 9\cos^2(\theta_1-\theta_2)}$$
- Choose $m\,l^2 = 6$

```
(%i6) plotdf([
        (2*p1-3*cos(th1-th2)*p2)/
        (16-9*(cos(th1-th2))^2),
        (8*p1-3*cos(th1-th2)*p2)/
        (16-9*(cos(th1-th2))^2)],[th1,th2],
       [trajectory_at,1,1],
       [th1,-4*%pi,4*%pi],
       [th2,-4*%pi,4*%pi],
       [direction,forward],
       [parameters,"p1=1,p2=1"],[sliders,"p1=-
       1:1,p2=-1:1"],[versus_t,1])$
(%i7)
```