ODEs: Systems of Equations

October 22, 2021

Higher Order ODEs

- Most differential equations we encounter are of 2nd order or higher.

$$y'' - \lambda^2 y = 0$$

- Our numerical algorithms only work on 1st order ODEs.
- We can convert a 2nd order ode into a 1st order ode by making the substitution

$$z = y'$$

so that we have the system

$$\begin{pmatrix} y' \\ z' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \lambda^2 & 0 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix}$$

Systems of ODEs

- We can also encounter systems of ODEs naturally. Consider the following problem:

Systems of ODEs

- We can also encounter systems of ODEs naturally. Consider the following problem:
- Suppose we are modelling the spread of a disease. We might have a susceptible population (S), an infected population (I), and a recovered population (R).
- The susceptible population gets infected at some rate α, while the infected population gets cured at some rate β.

Systems of ODEs

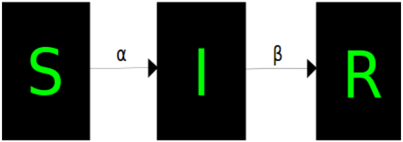
- We can also encounter systems of ODEs naturally. Consider the following problem:

- Suppose we are modelling the spread of a disease. We might have a susceptible population (S), an infected population (I), and a recovered population (R).

- The susceptible population gets infected at some rate α, while the infected population gets cured at some rate β.

- The ODE modelling such an event might be

$$\begin{pmatrix} S' \\ I' \\ R' \end{pmatrix} = \begin{pmatrix} -\alpha \frac{SI}{S+I+R} \\ \alpha \frac{SI}{S+I+R} - \beta I \\ \beta I \end{pmatrix}$$



Systems of ODEs

- We can use the same numerical methods for systems of ODEs. Consider the following ODE:

$$\begin{pmatrix} y_1' \\ y_2' \\ \vdots \\ y_n' \end{pmatrix} = \begin{pmatrix} f_1(y_1, y_2, \ldots, y_n, t) \\ f_2(y_1, y_2, \ldots, y_n, t) \\ \vdots \\ f_n(y_1, y_2, \ldots, y_n, t) \end{pmatrix}$$

- Forward Euler's method for this equation is

$$\begin{pmatrix} y_1^{m+1} \\ y_2^{m+1} \\ \vdots \\ y_n^{m+1} \end{pmatrix} = \begin{pmatrix} y_1^m + f_1(y_1^m, y_2^m, \ldots, y_n^m, t_m) \\ y_2^m + f_2(y_1^m, y_2^m, \ldots, y_n^m, t_m) \\ \vdots \\ y_n^m + f_n(y_1^m, y_2^m, \ldots, y_n^m, t_m) \end{pmatrix}$$

- We just apply forward Euler to each ODE independently.

Systems of ODEs

- The same principle applies to any of the methods we talked about.
- *Implicit* methods are even more frustrating. It now involves solving a *nonlinear system* of equations.
- So for now, we'll stick to using *explicit* methods.