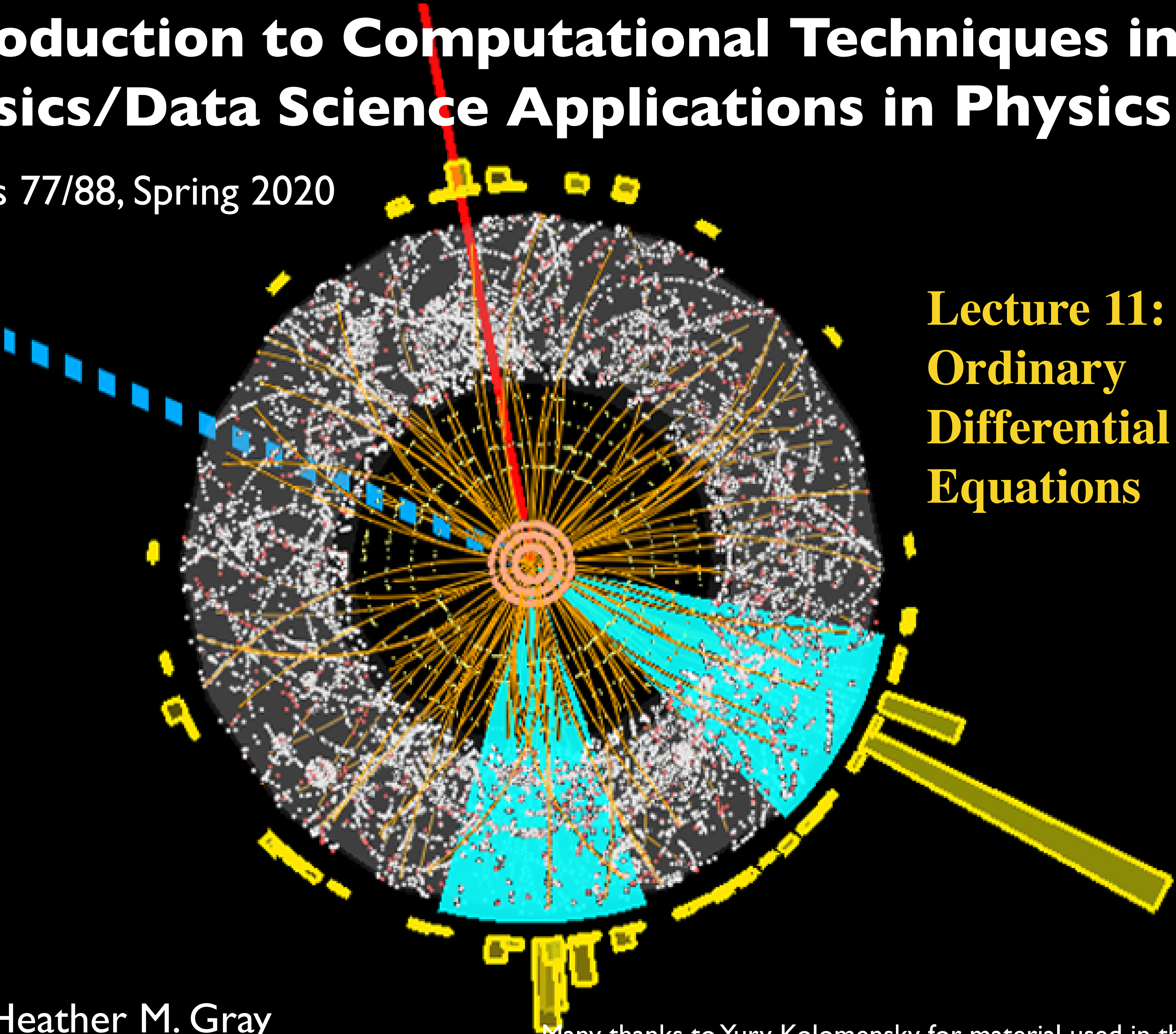
Introduction to Computational Techniques in Physics/Data Science Applications in Physics

Physics 77/88, Spring 2020



Lecture 11: Ordinary Differential Equations

Prof. Heather M. Gray

Many thanks to Yury Kolomensky for material used in this course

Announcements

- There will be no more homework assignments for the course
  - Please use the remaining time to work on your projects
- Course evaluations are available — any feedback that you have would be much appreciated; both on the regular time and the online period
  - We'll take 5 mins during next week's lecture to complete them
- This week's office hours will be on Tues from noon-1 pm in the class zoom
  - I'm happy to discuss any questions or issues you might have about the project or any questions about the class
- Everyone should have received feedback on their proposals
  - Many were excellent — well done!
  - If you got 4/8 and would like to address the issues; you're welcome to do that an resubmit the proposal
- The final presentations will be on 1/4 May
  - Please sign up for your slot in the doodle poll

Announcements: Final Presentations

- Presentations should be 10 minutes in length (advance warning -- we'll run a timer and you'll be cut off wherever you are in the presentation) and will be followed by 5 minutes of questions and discussion.
- You are free to choose the form of the presentation and how to split the time between different members of the group. For example, you might want to show slides, demonstrate your code, show a video, etc.
- All presentations will be on zoom, so you'll probably want to practice presenting on zoom (and sharing your screen) before the actual presentation.
- The presentation will be 20% of the grade for the project and common between members of the team

Announcements: Report & Code

- **Project Report**
  - Your report should be 3-4 pages in length including any diagrams, plots or references
  - Each student needs to prepare their own report and the report will be graded individually
  - Reports are due by 6 pm on 8 May
  - The report will be 40% of your project grade
- **Project Code**
  - The implementation (code) for your project is also due by 6 pm on 8 May
  - Please submit either a tar ball of the code or a link to the GitHub (or similar) repository
  - The group will receive a grade for the code
  - The code will be 30% of your project grade

Feedback from last time

- Thanks for the feedback on last weeks lecture
- Some questions about partial pivoting
    - e.g. Why do we have to use partial pivoting instead of shifting the rows within a matrix as that does not change the actual matrix itself?
        - You can definitely switch rows of a matrix, however, sometimes that can lead to problems later in Gaussian elimination (which aren't necessarily obvious early on)
        - Partial pivoting is designed to avoid such problems because it ensures that the element that we divide by is as far from zero as possible
- I did enjoy the feedback that I don't need to go over things again because the online videos mean that its possible to watch things again :-)

Outline

- Brief introduction to differential equations
- Euler Method
- Improved Euler method (Euler-Cromer)
- Runge-Kutta

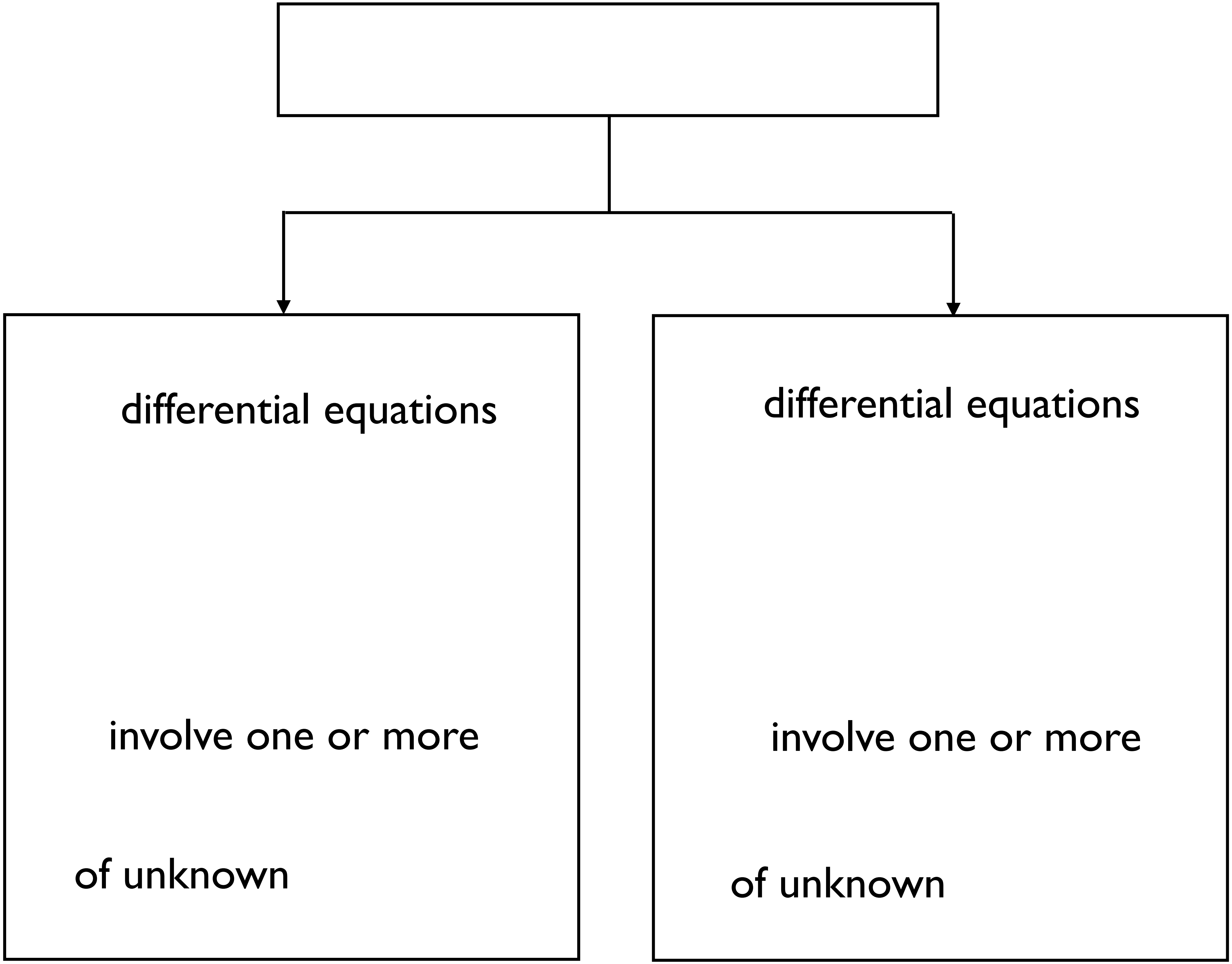
Numerical examples in notebook Lecture11.ipynb

Differential Equations

- Equations that are composed of and its are called differential equations

- importance in physics:
  - problems: in terms of of some
    - Equations of motion:
    - Thermodynamics:
    - E&M:

Differential Equations



differential equations

involve one or more

of unknown

differential equations

involve one or more

of unknown

Ordinary Differential Equations (ODEs)

- Most are described by a (set of)
  - e.g.
- Solutions are not always
  - Only for the you find in a
    - e.g.
- We will consider the most , which are solutions

Classification of ODEs

- ODE can be in ways
  -
    -
    -
    -
  -
    -
    -
  - conditions
    - value problems
    - value problems

Order of ODE

- The of an is the of the
  - Examples
    -
    -
    -

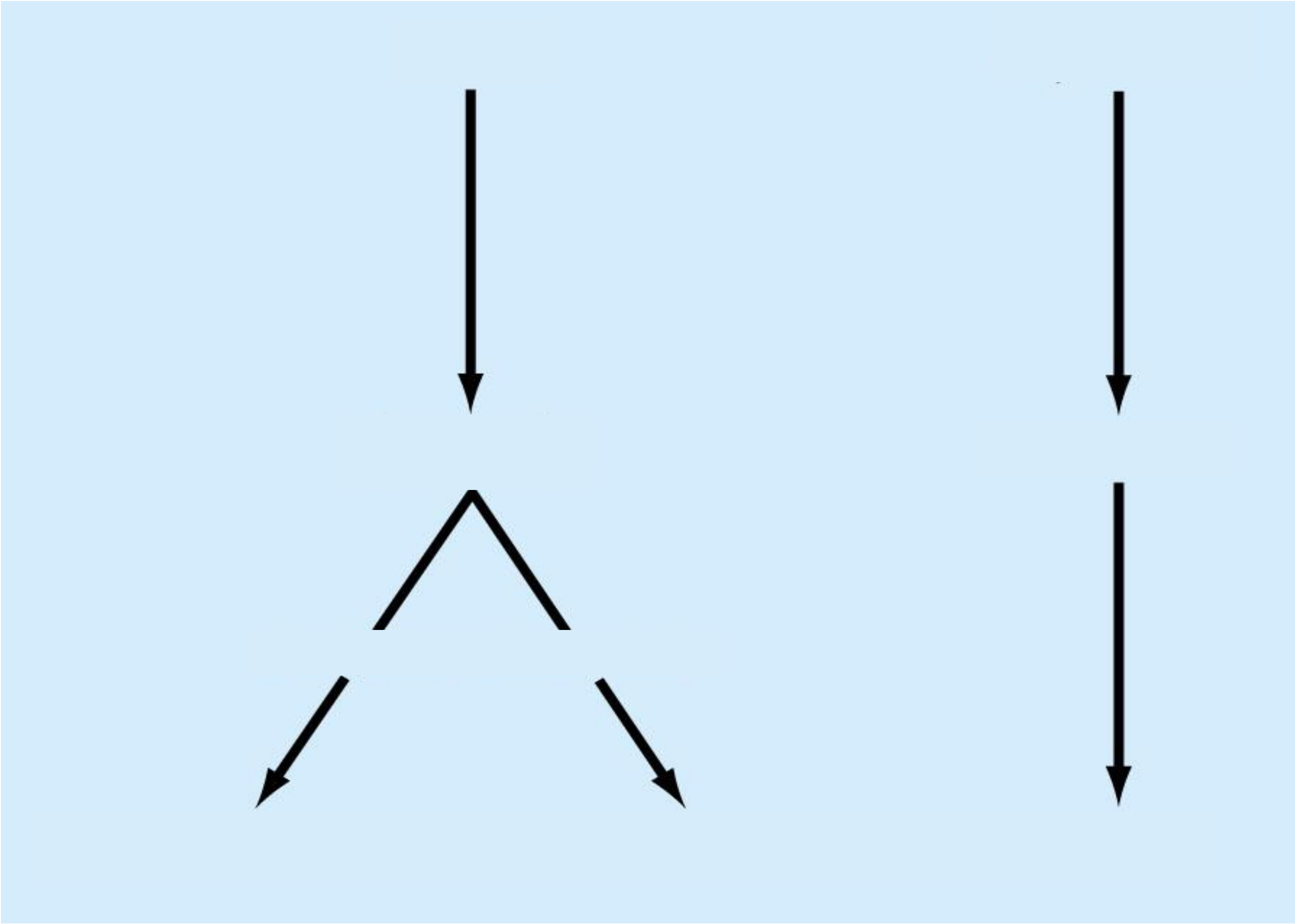
Linear vs Nonlinear ODEs

- An ODE is if the and its appear to power
- There is also of the and/or its
- i.e.
  - Examples
    -

Initial Conditions

- Problems
  - The are at of the
    - e.g.
- Problems
  - The are not at of the
    - e.g.

Analytical vs Numerical Solutions



Visualization: Direction Fields

- Consider the

- The specifies a at each in the plane where is

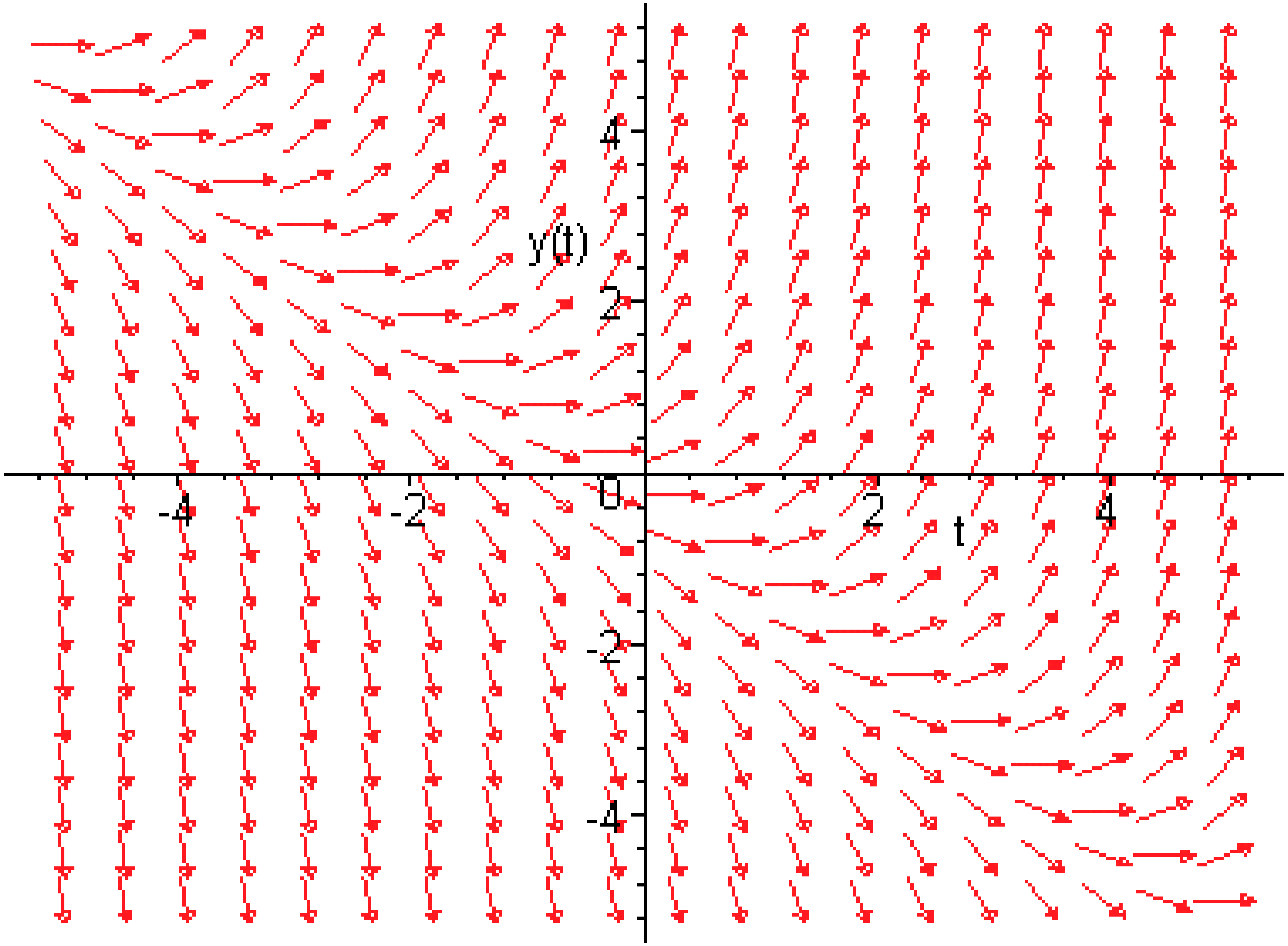
- It gives the that a to the must have at

- A plot of drawn at in the showing the of the

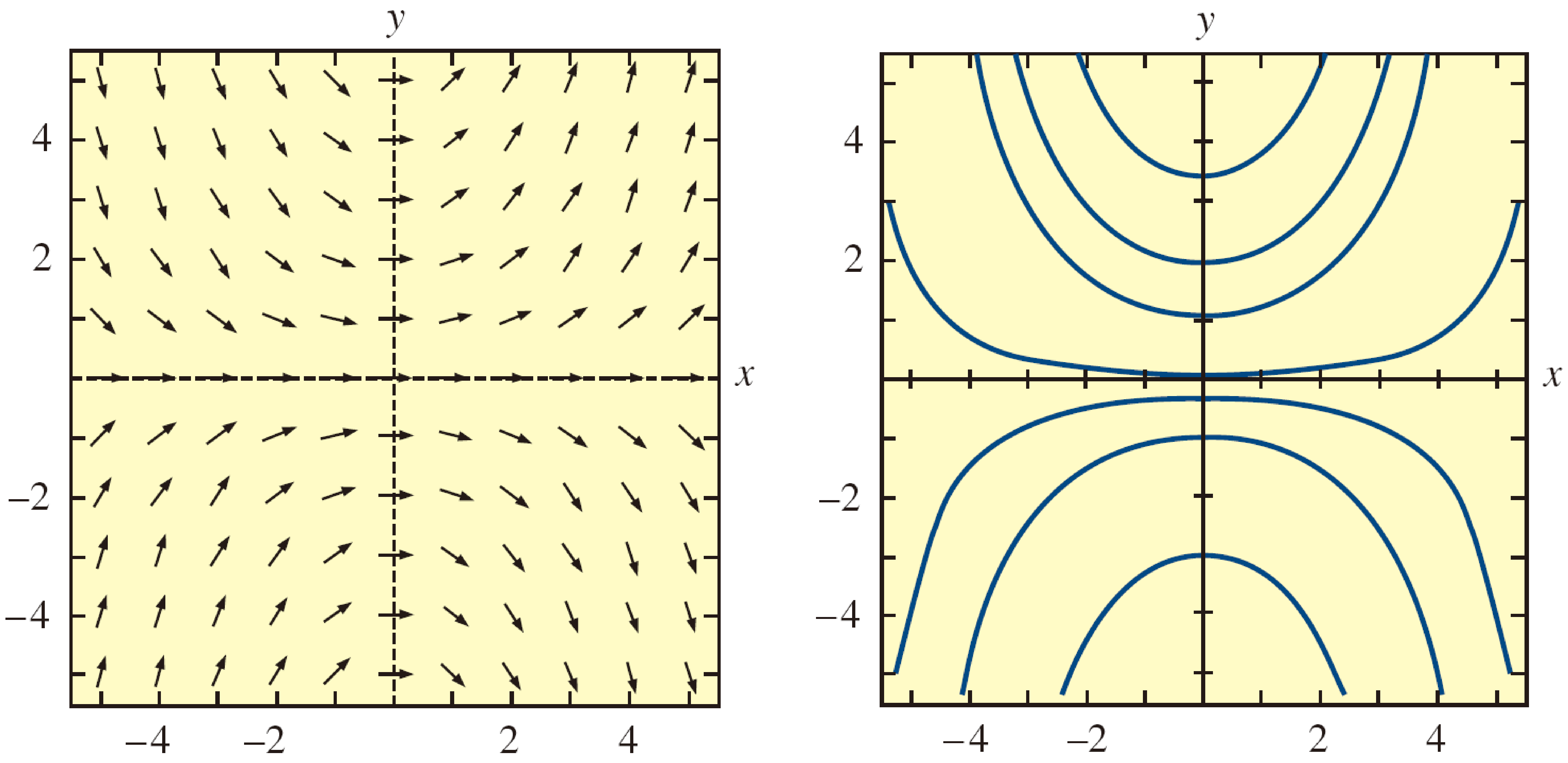
- This is called a for the

- The gives us the

Example



Example



Solving Differential Equations

- and
  - Try a , see if it works
    - reduce the set of
-
  - Reduce the to a case
-

Example: Analytical Solution

- An to a is a that satisfies the
  - Example
    -
  - Solution
    -
  - Proof
    -
    -
    -

Stability and Chaos

- Solution of an is
  - if the solutions resulting from of

    to
  - if the solutions resulting from

    back to
  - if the solutions resulting from

    away from the without

Numerical Solutions

- solution values are in moving across the in which the is sought
  - i.e. need to in a
- In from one to the , we incur some
  - next value lies on a from the one we
- or of solutions determines, in part, whether such are or with time

Euler Method

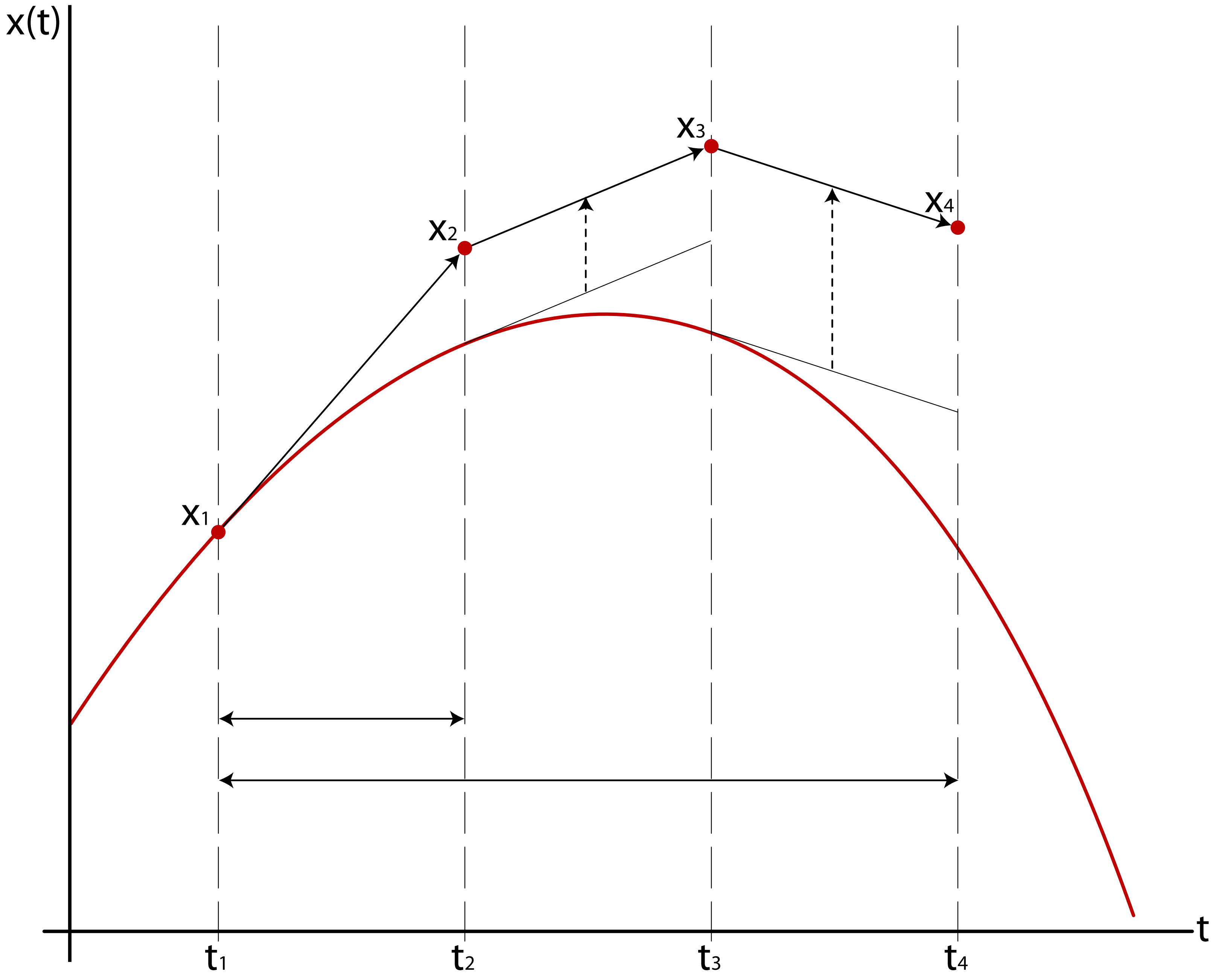
- Example: find   of a

- Rewrite with

- Implementation

- This is known as the

Euler Method



A. Ayers, Computational Physics with Python

Euler Method

- Precision limited by
    - Decrease to reduce
- Calculation time scales with the , i.e.
    -
        - where τ is the to be
- So far limited to

Euler Method for 2nd Order ODE

- Standard trick: convert a order ODE into a system of order ODEs
  - Example: free fall
    - Define
      -
      -
      -
  - Solutions follow Euler:
      -
      -

Generalize

- Rewrite in

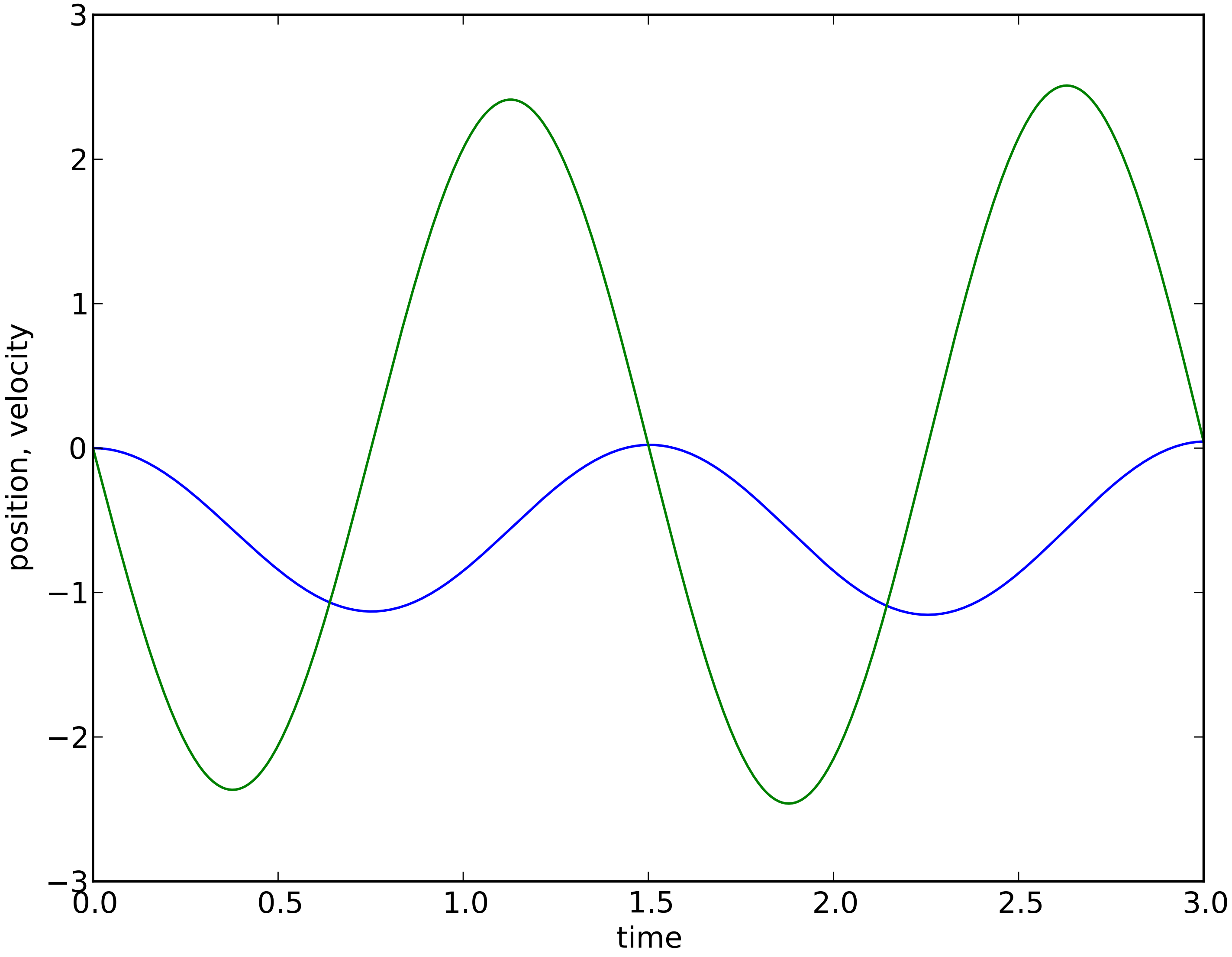
- Then the vector of is

- Euler solution:

- Implementation is straightforward (see notebook)

Solutions

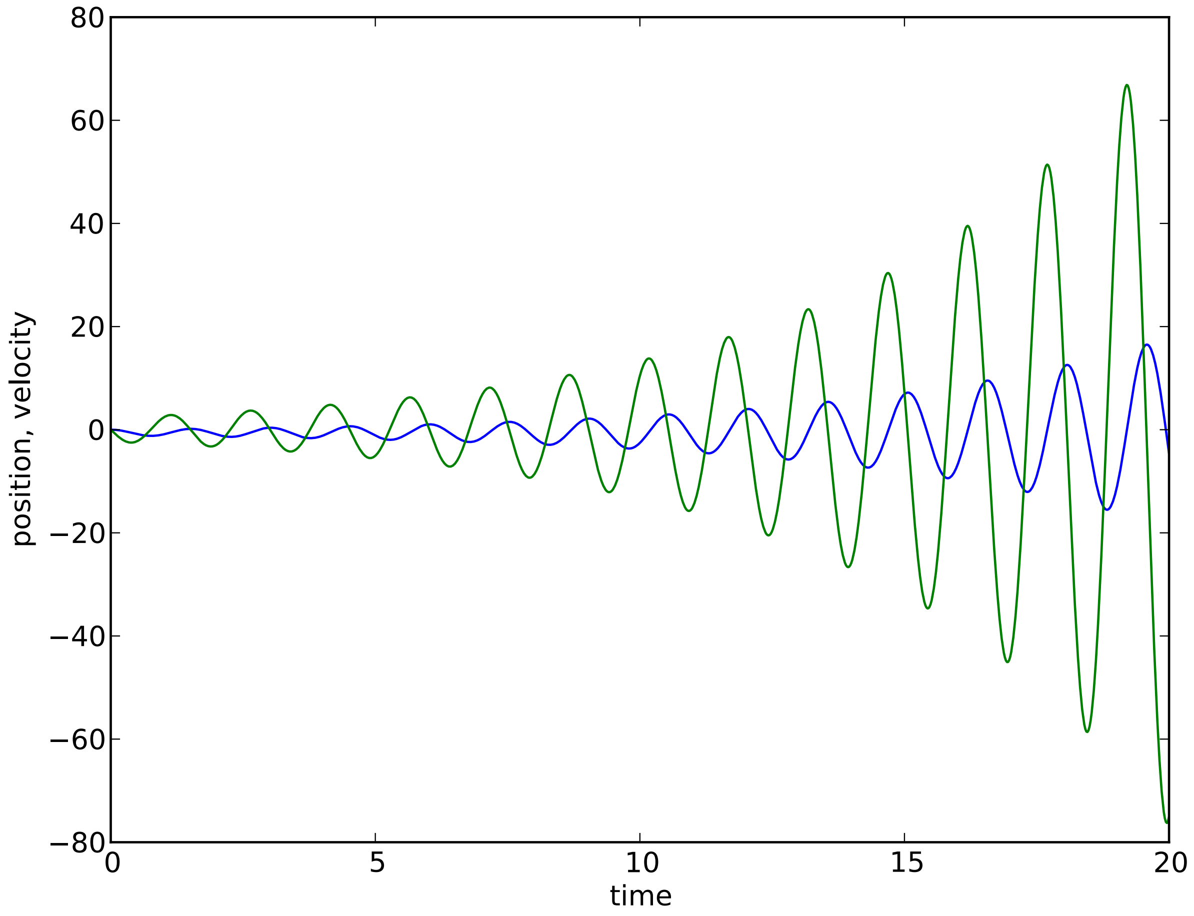
- Example:
  - Mass on a
    -



Let’s take a look at the jupyter notebook

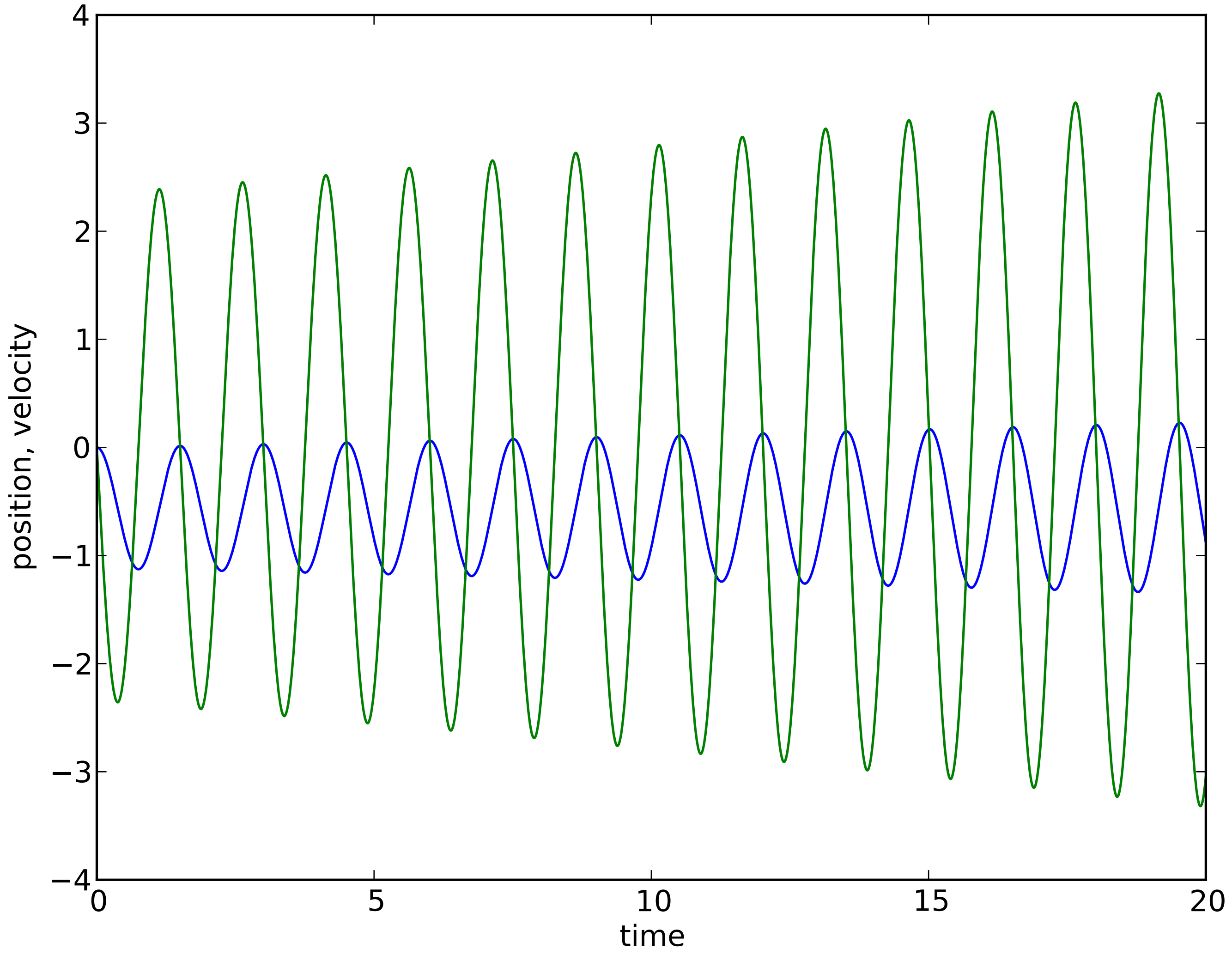
Problems

- Euler method underestimates → is not



Example with a total time of seconds and N =

Problems



Example with a total time of seconds and N =
Better, but the energy is still with

Euler-Cromer Method

- Trick that works for
    - Replace with evaluated at i.e.
    - Not a , so have to do

Let’s take go back to the jupyter notebook

Runge-Kutta Methods

- General case:
  - Find function with its $g(y, t) =$
  - Chain rule
  - Similarly:

Runge-Kutta Methods

- Taylor expansion:
  - $y(t + \Delta t) =$
- Compare to an alternative polynomial expansion:
  - $y(t + \Delta t) =$

Runge-Kutta Methods

- Polynomials:
  - $k_1 =$
  - $k_2 =$
  - $k_3 =$
  -
  - $k_n =$
- Coefficients and are determined by matching against
  - They are determined by
    - i.e. order RK, order RK, etc

2nd order RK

- $y(t + \Delta t) =$
- $k_2 =$
- Follows:
  - $y(t + \Delta t) =$
  - $\alpha_1 + \alpha_2 =$
  - $\alpha_1 \nu_{21} =$
- Standard solution:
  - $\nu_{21} = \qquad ; \alpha_1 = \alpha_2 =$
  - $t(1 + \Delta t) =$
  - $k_1 =$
  - $k_2 =$

4th Order RK

- $y(t + \Delta t) =$
  - $k_1 =$
  - $k_2 =$
  - $k_3 =$
  - $k_4 =$
- 4th order RK is a standard tool, offers good tradeoff between precision and speed

Let’s take a look at the jupyter notebook