Scientific Computing 2013

Maastricht Science Program

Week 1

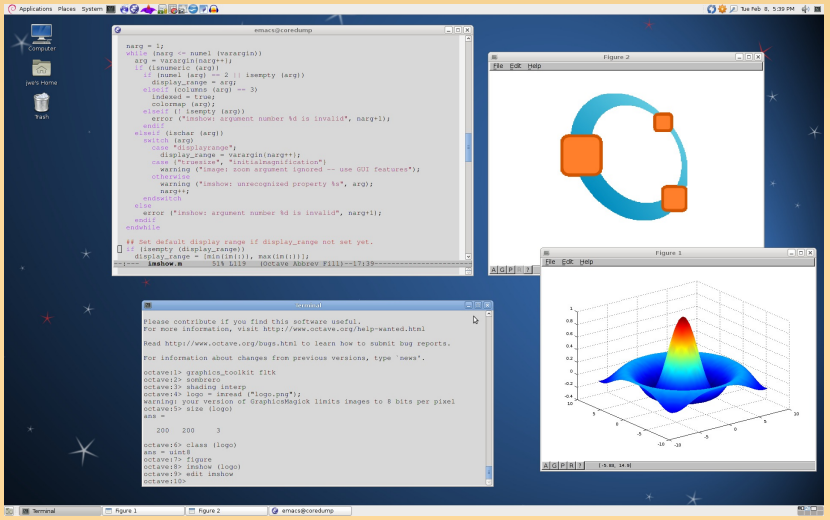
Frans Oliehoek
<frans.oliehoek@maastrichtuniversity.nl>

Good Choice!

- Let me start: Congratulations!
- There is virtually no branch of science that can do without scientific computations...
- Exact science require a way of thinking that is closely linked with math and programming

Scientific Computing: What is it about?

- **Computing**: we will learn to 'program'
  - Really: make the computer do what you want.
  - In this course we will work with
    - Matlab, or
    - (free software) Octave.



- **Scientific**:
  - We will deal with scientific problems.
  - Mostly based on calculus and linear algebra.

Scientific Computing - Goals

- accustomed with Matlab and Mathematica.
- familiar with basics of programming
- overview of some topics scientific computation:
  - (non-)linear systems, numerical and symbolic integration, differential equations and simulation.

- who likes math?
  - why you should care about it!

Why Scientific Computing?

- Why mathematical models?
  - precise understanding!
- Why use computers?
  - by hand: only very simple models...
  - Usually: no closed form solution.
    - E.g., $x^5 – x + 1 = 0$
      (solving a polynomial equation of degree > 4)
  - But can get numerical approximations!
  - Make them do what you want: programming

Alright, so what is programming?

- Programming is about making a machine (computer) do what you want it to.
  - difference with a oven or other machines?

Alright, so what is programming?

- Programming is about making a machine (computer) do what you want it to.
  - difference with a oven or other machines?
  - → a computer can do many tasks

    and programming let's you do that!
- We focus on scientific computations.

- Example: how many km is 1 light year?

How many km in a light year?

- 299792458 * 365 * 24 * 60 * 60 / 1000 = 9.4543e+12

- These computations become difficult to interpret!
  - How about if we could name parts of this computation?

How many km in a light year?

- 299792458 * 365 * 24 * 60 * 60 / 1000 = 9.4543e+12

- These computations become difficult to interpret!
  - How about if we could name parts of this computation?

```
speed_of_light = 299792458
secs_per_year = 365 * 24 * 60 * 60
m_per_lyear = speed_of_light * secs_per_year
km_per_lyear = m_per_year / 1000
```

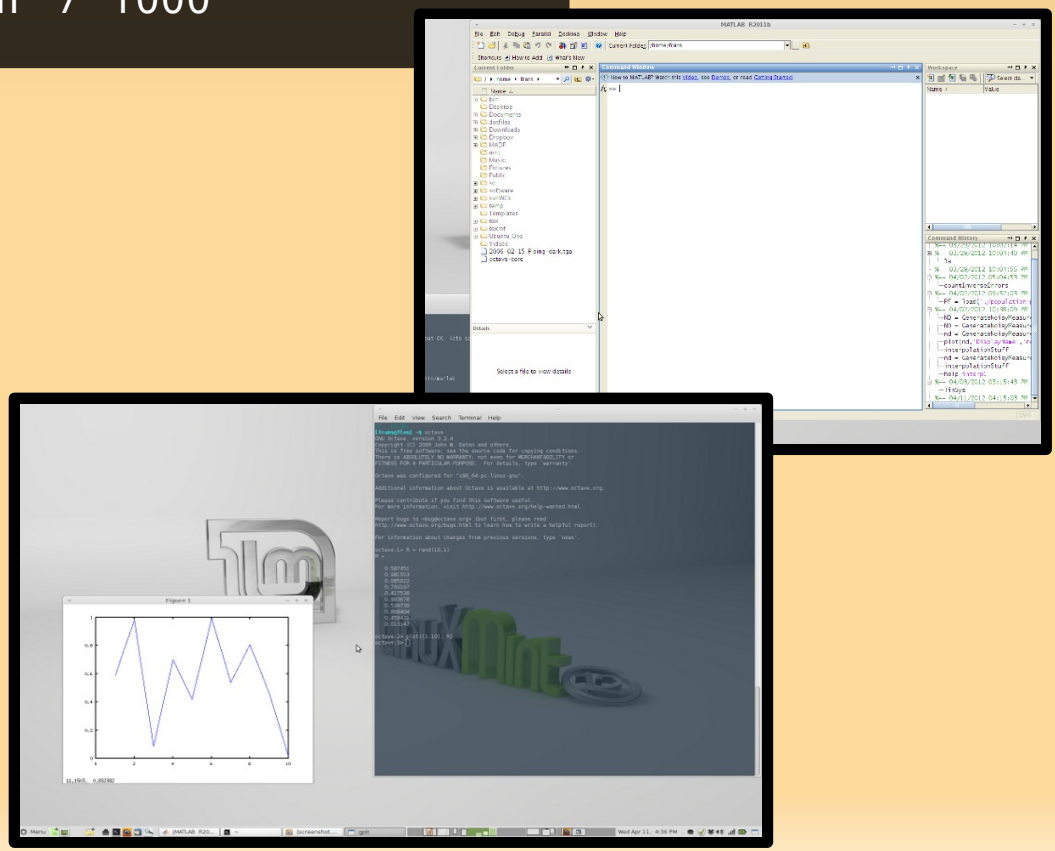
  - meaning of '='
  - the names are called 'variables'

Our first Matlab/Octave code!

- This is our first Matlab code!

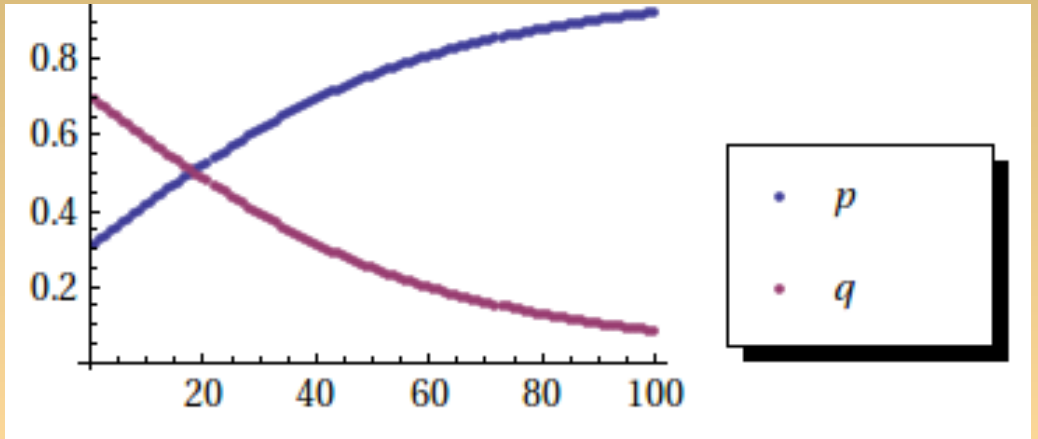
```
speed_of_light = 299792458
secs_per_year = 365 * 24 * 60 * 60
m_per_lyear = speed_of_light * secs_per_year
km_per_lyear = m_per_year / 1000
```

- Matlab (Octave) is like a convenient calculator.

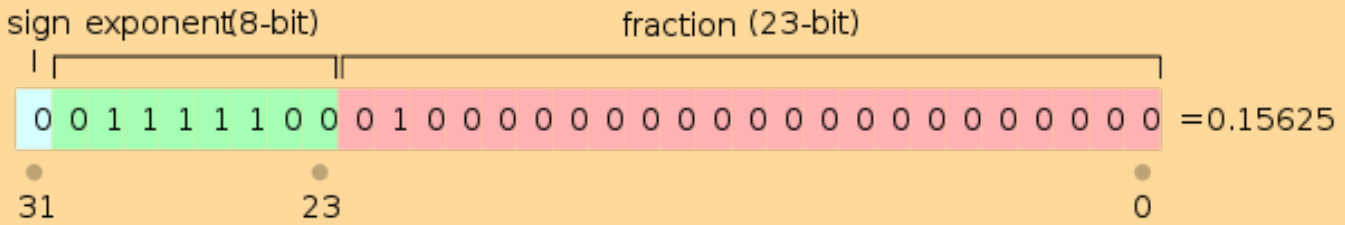


Overview

- Population models
  - functions
  - plotting



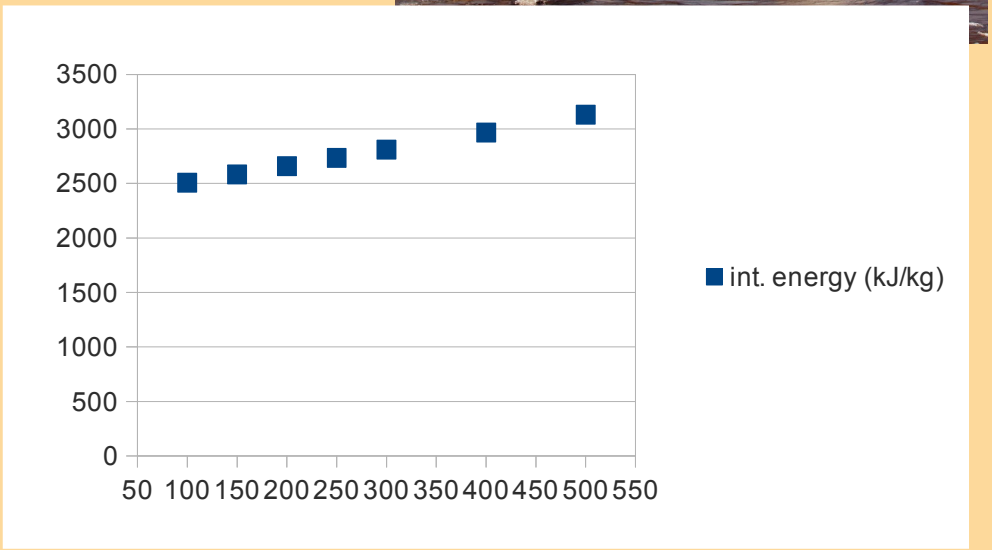
- How numbers are represented



Overview – Interpolation

- In the study of Geysers, an important quantity is the internal energy of steam.

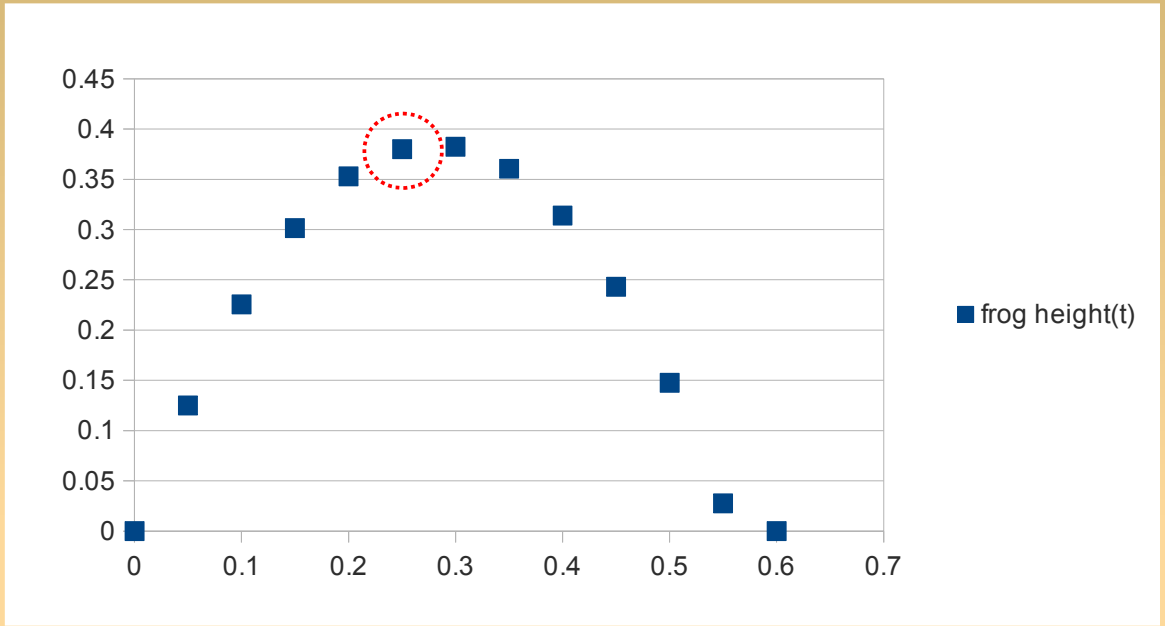
Temp. (Celsius)	int. energy (kJ/kg)
100	2506.7
150	2582.8
200	2658.1
250	2733.7
300	2810.4
400	2967.9
500	3131.6



(from Etter, 2011, Introduction to MATLAB)

Overview – differentiation/integration

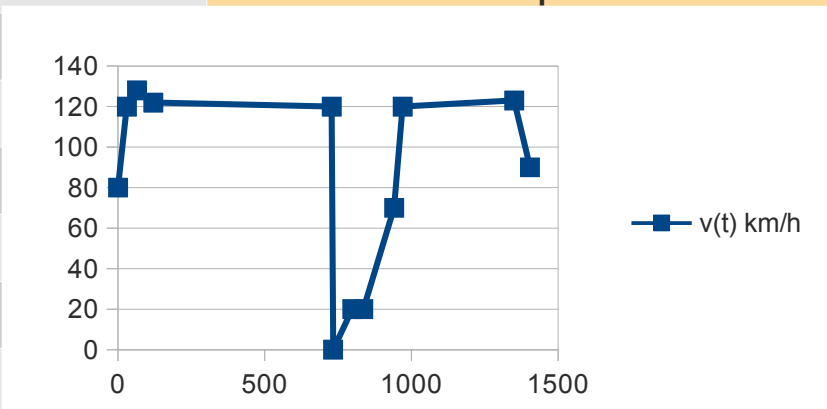
- Differentiation
  - Determine the vertical speed



- Integration
  - how far did we go?

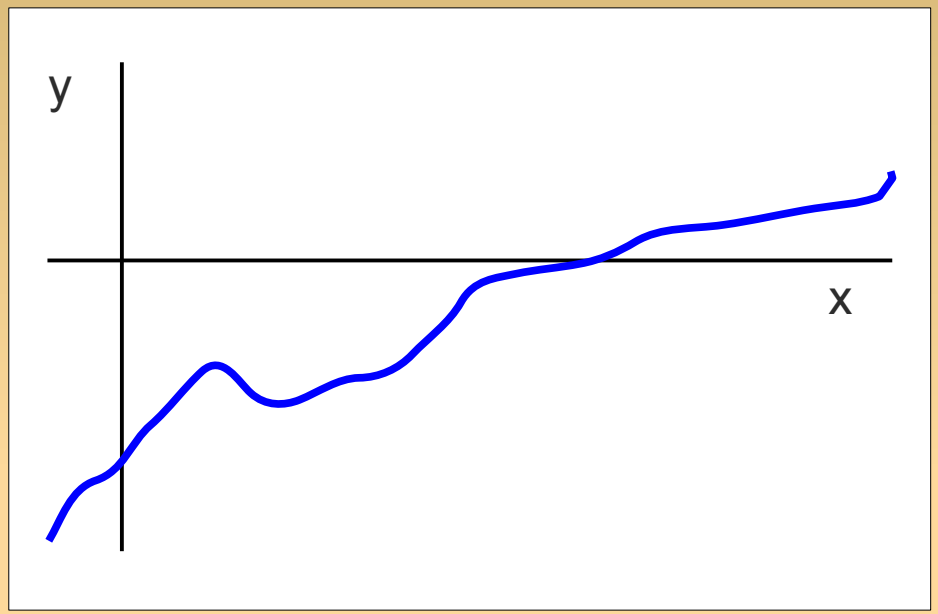
t	v(t) km/h
0	80
30	120
65	
120	
728	
733	
798	
836	
941	

enter highway ramp



Overview – algorithm

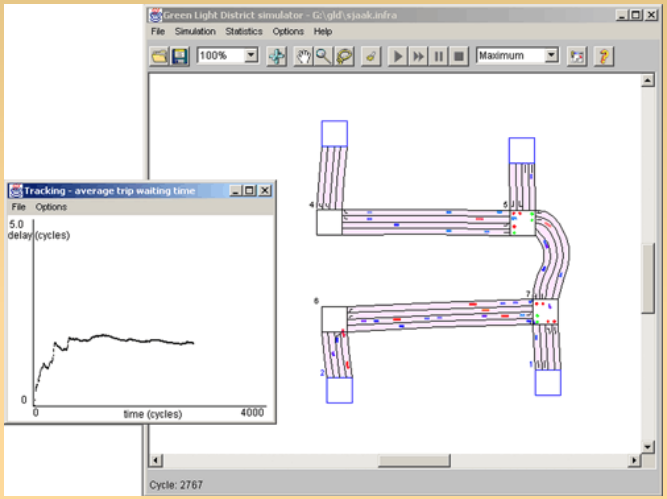
- Find the root...?

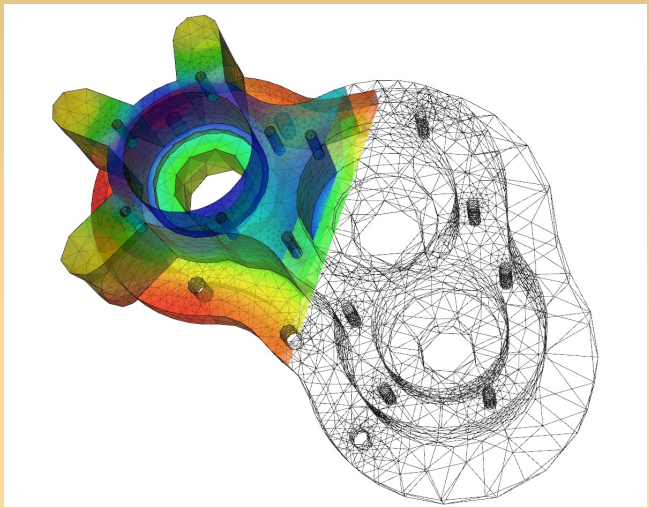


- To solve this problem: **numerical algorithm**.
  - algorithm = cook-book recipe
  - an algorithm can be **implemented** (converted to code in a programming language).

Overview – simulation

- Also basic steps: simulation!





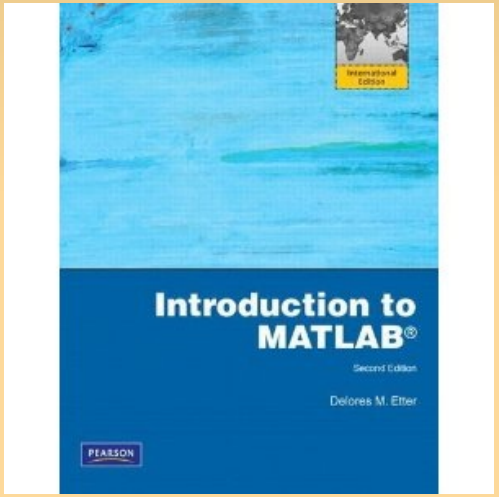
- we will keep it simple, though...
  - difference equations (e.g., population models)
  - differential equations (e.g., physics)

Practicalities


```
      Name: Frans Oliehoek
Department: DKE (RAI group)
  Location: SSK 39, room 2.001
      Tel.: +31 43 3883485
     Email: frans.oliehoek@maastrichtuniversity.nl
       WWW: http://people.csail.mit.edu/fao/
```


- About me
  - Computer Science / AI

- Book:
  - Introduction to MATLAB.
    Delores M. Etter. 2nd ed.



- Course manual on Eleum and my website.
- All information on my website (under 'teaching'):

  http://people.csail.mit.edu/fao/

Practicalities


```
      Name: Frans Oliehoek
Department: DKE (RAI group)
  Location: SSK 39, room 2.001
      Tel.: +31 43 3883485
     Email: frans.oliehoek@maastrichtuniversity.nl
       WWW: http://people.csail.mit.edu/fao/
```


- Attendance
  - standard MSC rules - 85%
- Grades based on:
  - Show your work at beginning of next lab.
  - Pop-quiz questions
  - hand-in assignments (40%)
    - **follow the instructions!**
- Work individually...
  - helping each other: great!
  - do **not** copy

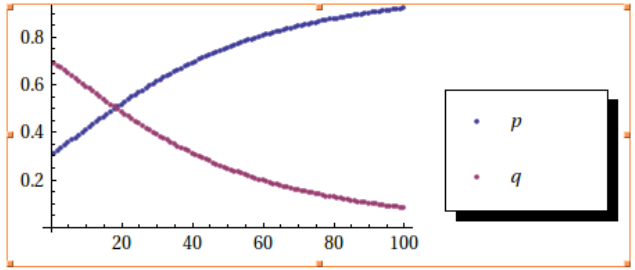
Let's get started

- Today: Mathematica

we get a list that contains 2 lists: one of p's and one of q's. The first of these (the list of p's) we can extract us
(the double brackets denote an operation that extracts an element from a list)

These can now be put in listplot:

```
Needs["PlotLegends`"]
ListPlot[{tt[[1]], tt[[2]]}, PlotLegend → {p, q}, LegendPosition → {1.1, -0.4}]
```



As expected, we see that since *P* leads to an advantage, its frequency increases over time.

Hand-in Assignment: Predator-Prey Dynamics

- Assignments are posted on my website.

  http://people.csail.mit.edu/fao

  - download the notebook
  - open it in Mathematica, and work through it