Solving computational problems

(in environmental engineering)

Computational methods

Reasons for computer methods
(*iterative calculations*)

- Modeling
  - Forward modeling (numerical integration)
  - Inverse modeling (parameter estimation)
- Data analysis

Classifications of models

- Mechanistic / Statistical
- Deterministic / Stochastic
- Analytical / Computational

Reasons for modeling

- Prediction / Forecasting
- Understanding

Simulation → generate data

Data analysis → reduce amount of data

Computational methods

Forward modeling

$$\boldsymbol{y} = F(\boldsymbol{x})$$

Inverse modeling

$$\hat{\boldsymbol{x}} = F^{-1}(\boldsymbol{y})$$

Data analysis

https://ch.mathworks.com/products/matlab/data-analysis.html
https://ch.mathworks.com/discovery/data-visualization.html

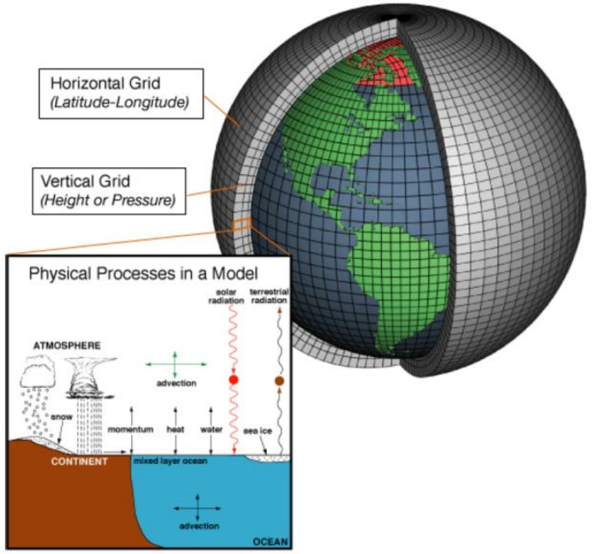
Numerical methods

- Integration
- Differentiation
- Solving system of equations (root-finding in the univariate case)
- …

Models in environmental engineering

"All models are wrong, but some are useful."
- George P. Box

Mechanistic models
continuum models



source: NOAA

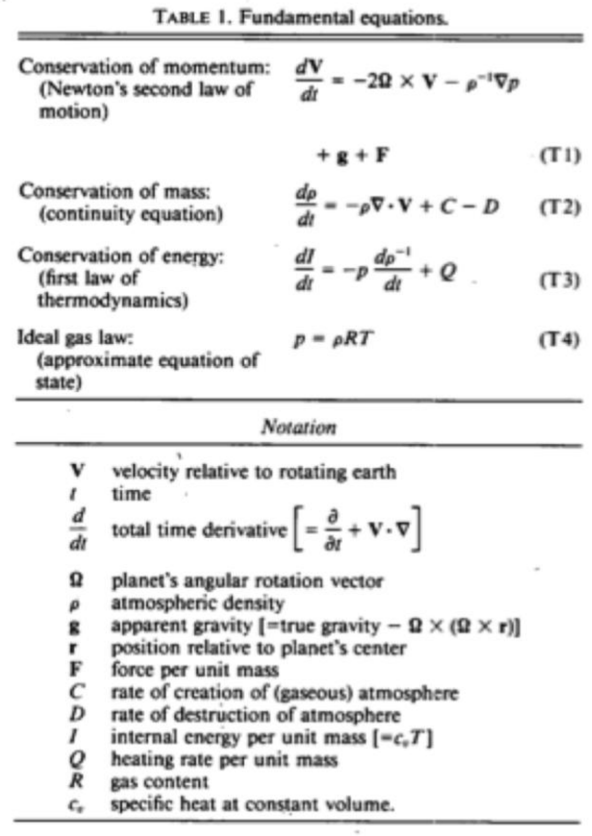


TABLE 1. Fundamental equations.

Conservation of momentum: (Newton's second law of motion)	$\frac{d\mathbf{V}}{dt} = -2\mathbf{\Omega} \times \mathbf{V} - \rho^{-1}\nabla p$ $+ \mathbf{g} + \mathbf{F}$	(T1)
Conservation of mass: (continuity equation)	$\frac{d\rho}{dt} = -\rho\nabla \cdot \mathbf{V} + C - D$	(T2)
Conservation of energy: (first law of thermodynamics)	$\frac{dI}{dt} = -p\frac{d\rho^{-1}}{dt} + Q$	(T3)
Ideal gas law: (approximate equation of state)	$p = \rho RT$	(T4)

Notation

$\mathbf{V}$	velocity relative to rotating earth
t	time
$\frac{d}{dt}$	total time derivative $\left[= \frac{\partial}{\partial t} + \mathbf{V} \cdot \nabla\right]$
$\mathbf{\Omega}$	planet's angular rotation vector
ρ	atmospheric density
$\mathbf{g}$	apparent gravity [=true gravity $- \mathbf{\Omega} \times (\mathbf{\Omega} \times \mathbf{r})$]
$\mathbf{r}$	position relative to planet's center
$\mathbf{F}$	force per unit mass
C	rate of creation of (gaseous) atmosphere
D	rate of destruction of atmosphere
I	internal energy per unit mass [$=c_v T$]
Q	heating rate per unit mass
R	gas content
c_v	specific heat at constant volume.

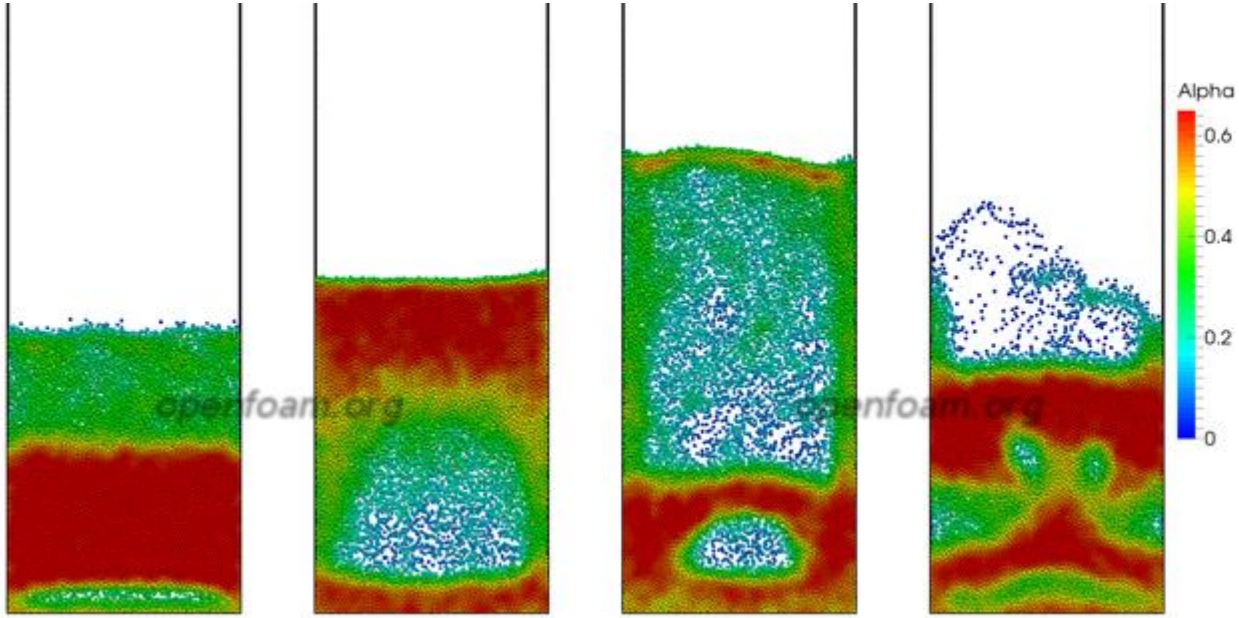
Hansen et al., 1983

Numerically integrate over time/space

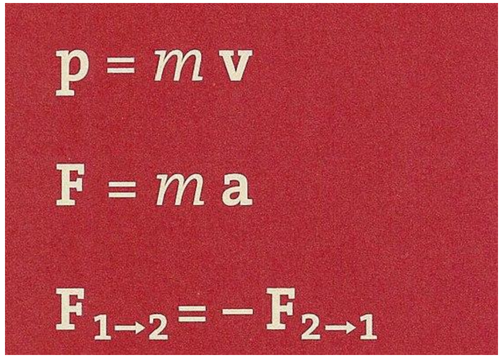
- Integration scheme
- Initial values
- Boundary values

Mechanistic models

discrete models



https://openfoam.org/



$$\mathbf{p} = m\,\mathbf{v}$$

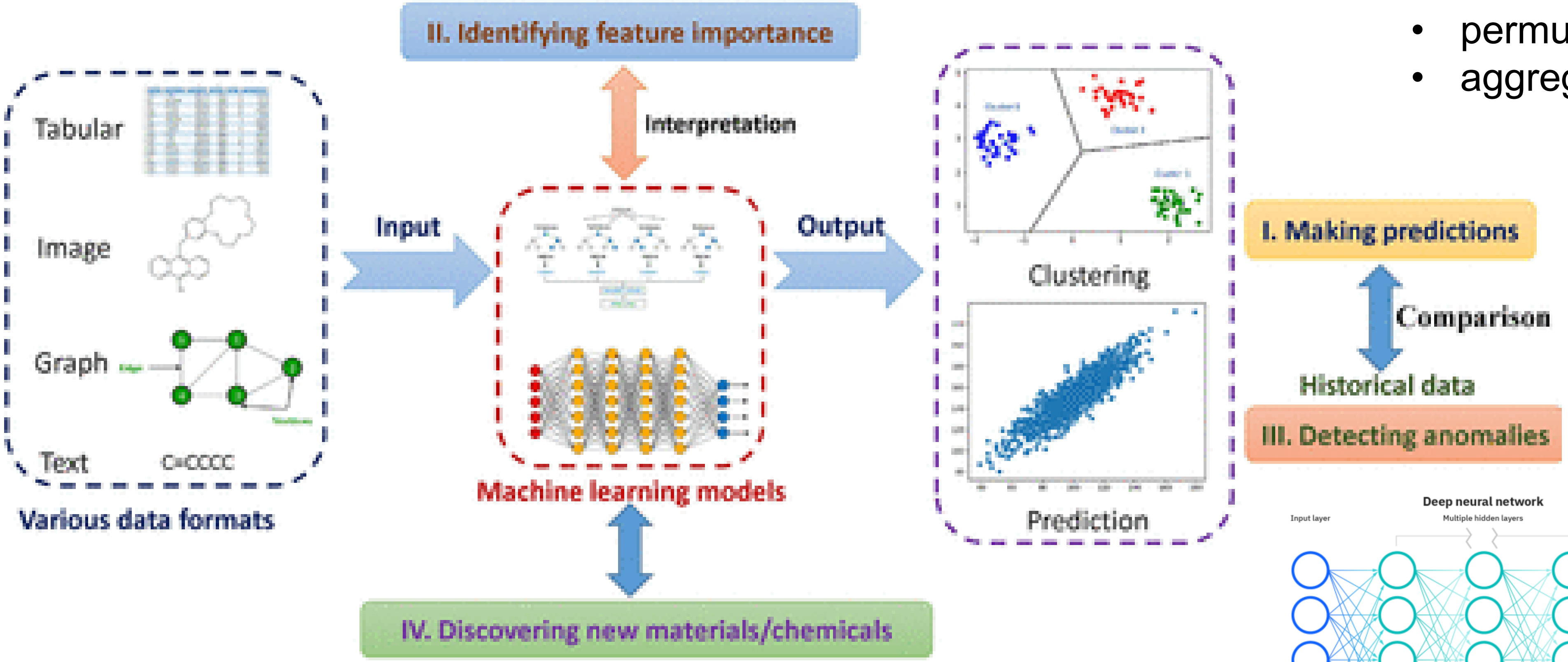
$$\mathbf{F} = m\,\mathbf{a}$$

$$\mathbf{F}_{1\rightarrow 2} = -\,\mathbf{F}_{2\rightarrow 1}$$

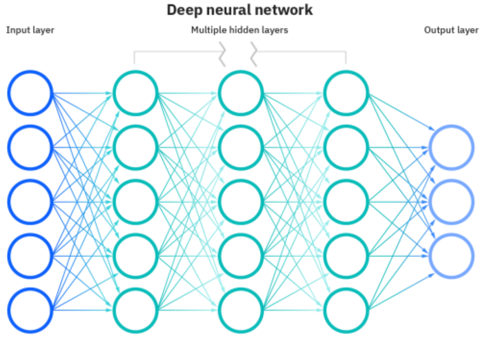
source: Wikipedia

Statistical models

Data preparation:
- cleaning
- curation
- permutation
- aggregation



Zhong et al., *ES&T*, 2021



$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

$$output = f(x) = \begin{cases} 1 \text{ if } \sum w_1 x_1 + b \geq 0 \\ 0 \text{ if } \sum w_1 x_1 + b < 0 \end{cases}$$

https://www.ibm.com/cloud/learn/neural-networks

Statistical time series forecasting

Model present values based on previous values

$$X_t = \varphi_0 + \varphi_1 X_{t-1} + \ldots + \varphi_p X_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \ldots - \theta_q \varepsilon_{t-q}$$

Mancini et al., *J. Phys.: Conf. Ser.*, 2022

Atmos. Chem. Phys., 20, 9051–9066, 2020
https://doi.org/10.5194/acp-20-9051-2020

Atmospheric Chemistry and Physics
Open Access
EGU

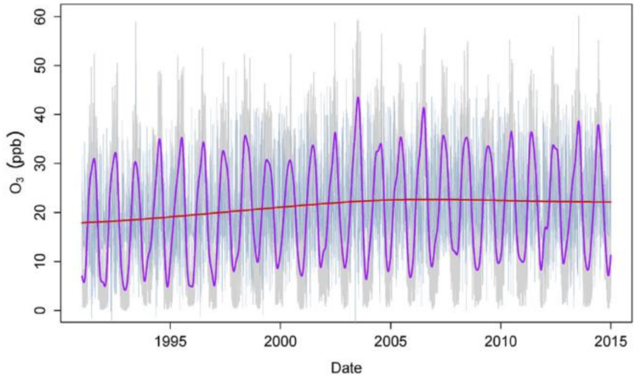
Temporal and spatial analysis of ozone concentrations in Europe based on timescale decomposition and a multi-clustering approach

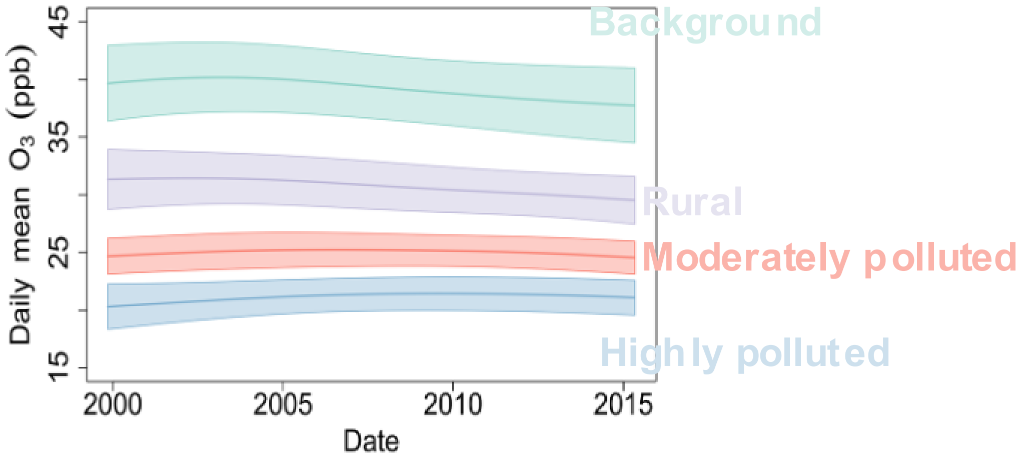
Eirini Boleti[1,2], Christoph Hueglin[1], Stuart K. Grange[1,4], André S. H. Prévôt[3], and Satoshi Takahama[2]
[1]Empa, Swiss Federal Laboratories for Materials Science and Technology, Überlandstrasse 129,
8600 Dübendorf, Switzerland
[2]EPFL, École Polytechnique Fédérale de Lausanne, Route Cantonale, 1015 Lausanne, Switzerland
[3]PSI, Paul Scherrer Institute, 5232 Villigen, Switzerland
[4]Wolfson Atmospheric Chemistry Laboratories, University of York, York, YO10 5DD, UK

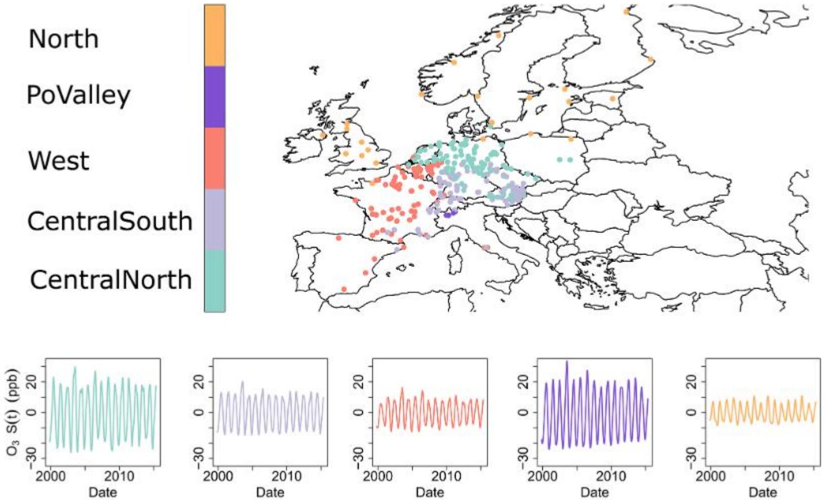
Correspondence: Christoph Hueglin (christoph.hueglin@empa.ch)

Received: 7 October 2019 – Discussion started: 30 October 2019
Revised: 16 March 2020 – Accepted: 30 March 2020 – Published: 30 July 2020

$$y(t) = LT(t) + S(t) + W(t) + E(t)$$







nature COMMUNICATIONS

ARTICLE

Check for updates

https://doi.org/10.1038/s41467-021-27484-1 OPEN

Secondary organic aerosol association with cardiorespiratory disease mortality in the United States

Havala O. T. Pye[1✉], Cavin K. Ward-Caviness[2], Ben N. Murphy[1], K. Wyat Appel[1] & Karl M. Seltzer[3]

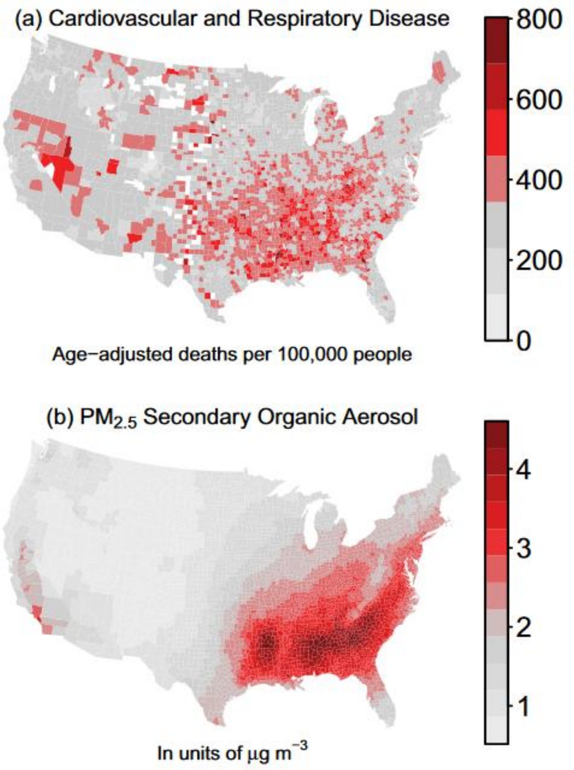
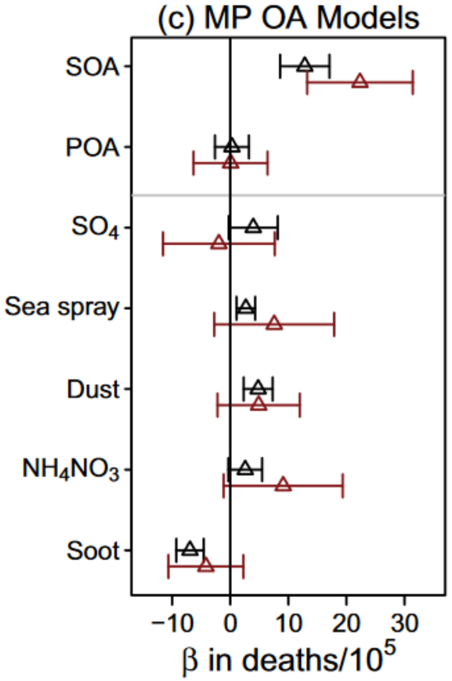


Fig. 1 Cardiorespiratory disease mortality rates and secondary organic aerosol concentrations. County-level, year 2016 (**a**) cardiovascular and respiratory disease age-adjusted death rates (per 100,000 in population) are from CDC and (**b**) PM$_{2.5}$ secondary organic aerosol concentrations are predicted by CMAQ. White in (**a**) indicates no death rate data while light gray indicates low reported rates.

$$D = \beta_0 + \{\beta_1 PM_{SOA_{BVOC}} + \beta_2 PM_{SOA_{AVOC}} + \beta_3 PM_{POA}\} + \beta_4 PM_{Seaspray} + \beta_5 PM_{Soot} + \beta_6 PM_{NH_4NO_3} + \beta_7 PM_{SO_4} + \beta_8 PM_{Dust} + \sum_{j=1}^{N} \beta'_j C_j$$

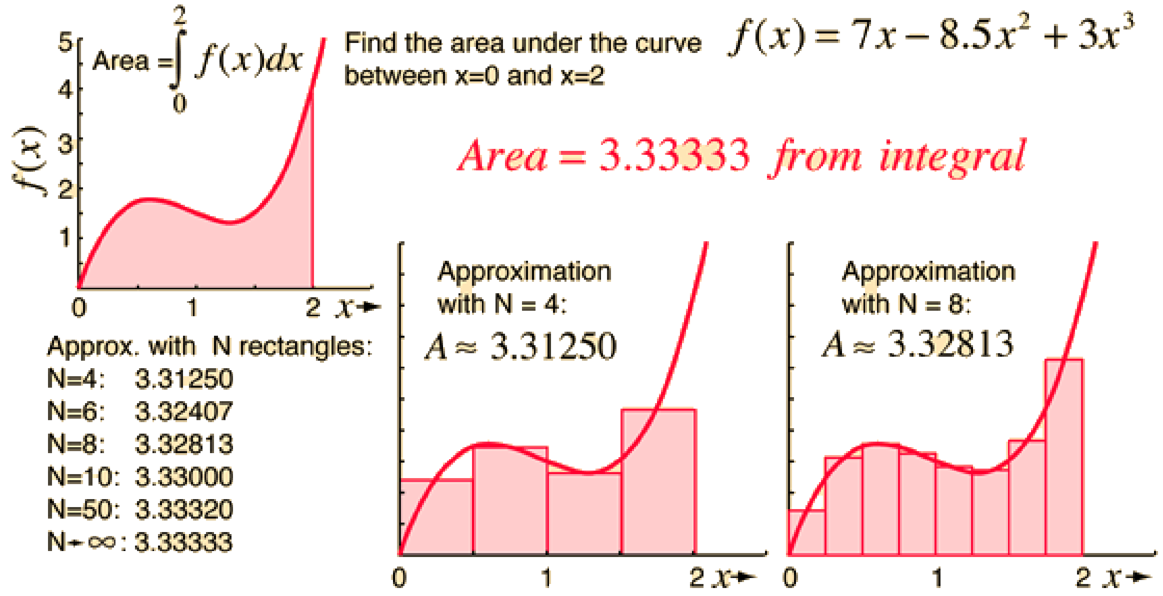


Numerical methods

(Simplest examples selected)

Numerical integration

area under curve



Find the area under the curve between x=0 and x=2 $f(x) = 7x - 8.5x^2 + 3x^3$

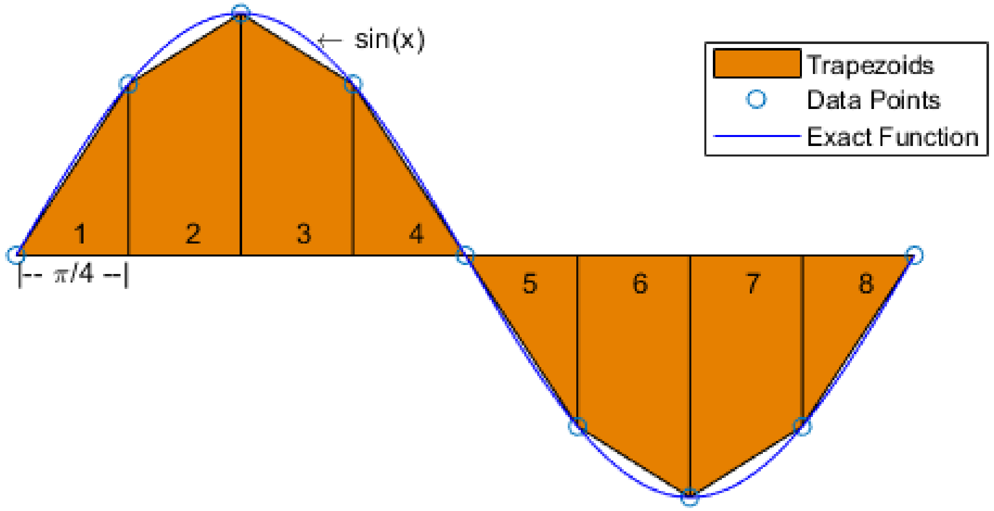
Area = 3.33333 from integral

Approx. with N rectangles:

N=4:	3.31250
N=6:	3.32407
N=8:	3.32813
N=10:	3.33000
N=50:	3.33320
$N \to \infty$:	3.33333

Approximation with N = 4: $A \approx 3.31250$

Approximation with N = 8: $A \approx 3.32813$



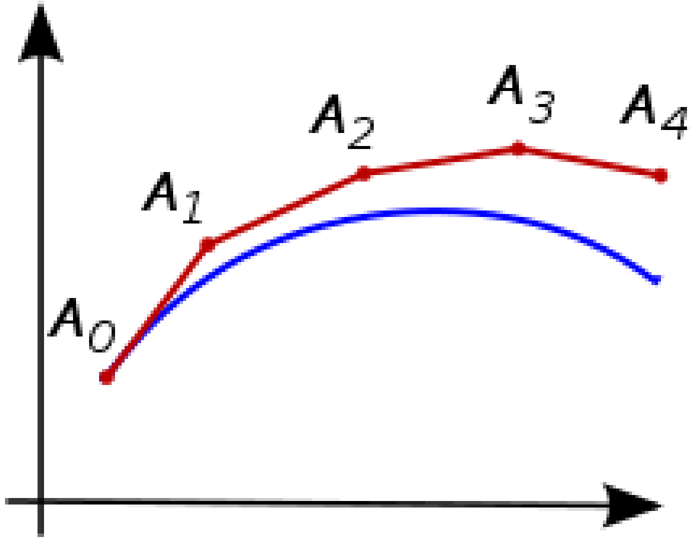
http://hyperphysics.phy-astr.gsu.edu/hbase/integ.html

https://ch.mathworks.com/help/matlab/ref/trapz.html

Numerical integration

Initial value problem

$$y_{i+1} = y_i + \frac{dy}{dx}\Delta x$$



numerical approximation

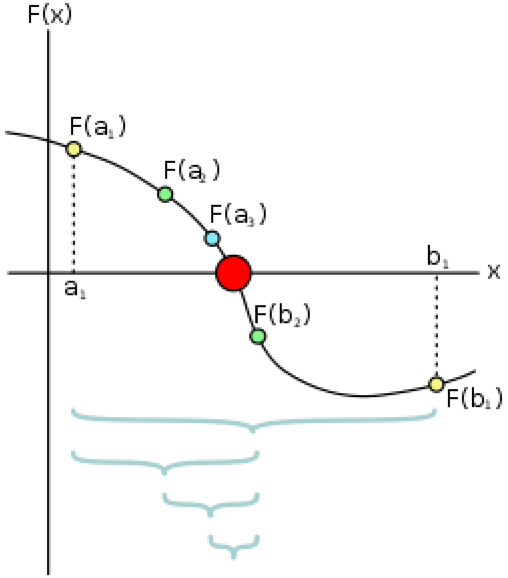
analytical solution

https://en.wikipedia.org/wiki/Euler_method

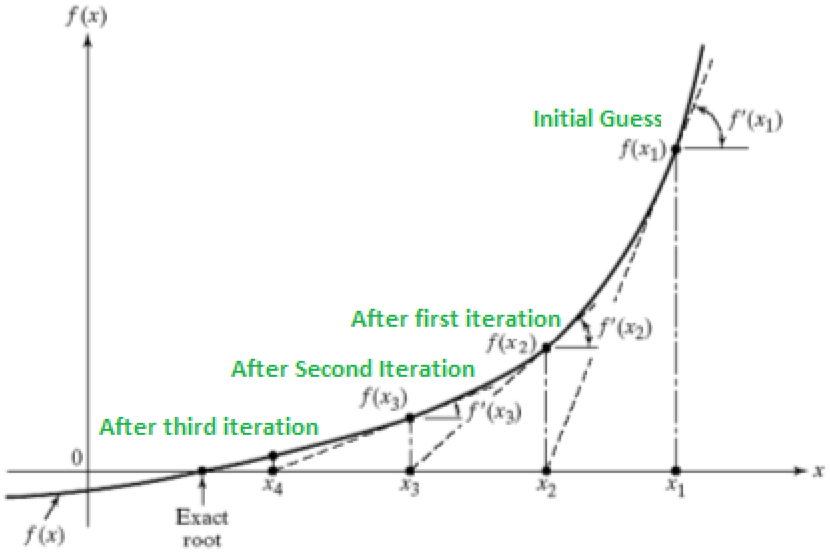
Constraint satisfaction

Root finding

Bisection method

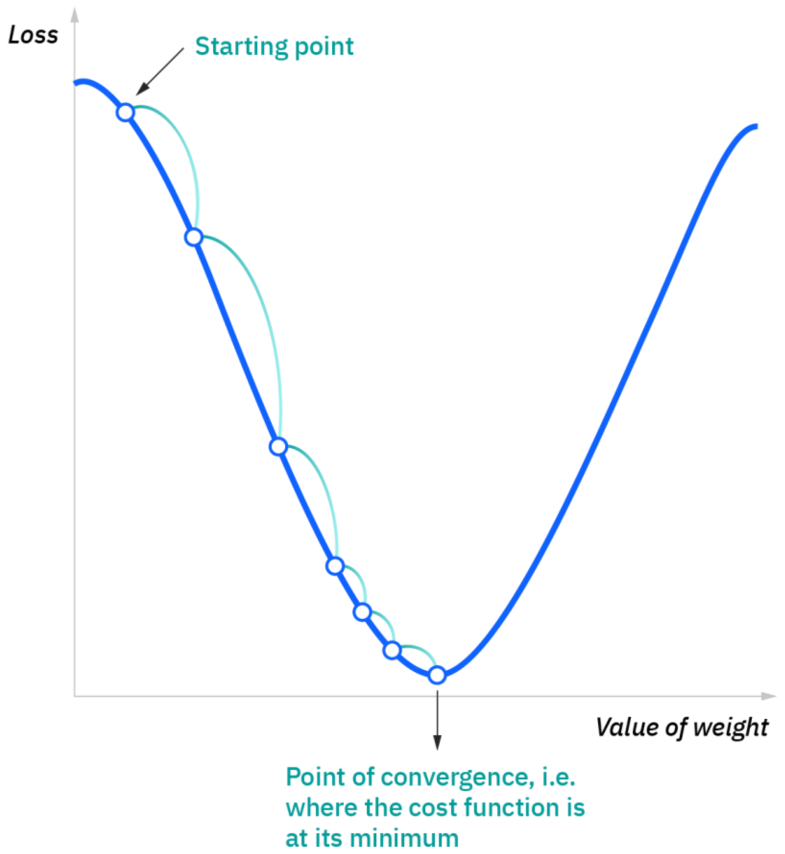


Newton-Raphson method



https://en.wikipedia.org/wiki/Bisection_method
https://www.geeksforgeeks.org/program-for-newton-raphson-method/

Parameter estimation
Minimization



$$J = (\boldsymbol{y} - F(\boldsymbol{x}))^T (\boldsymbol{y} - F(\boldsymbol{x}))$$

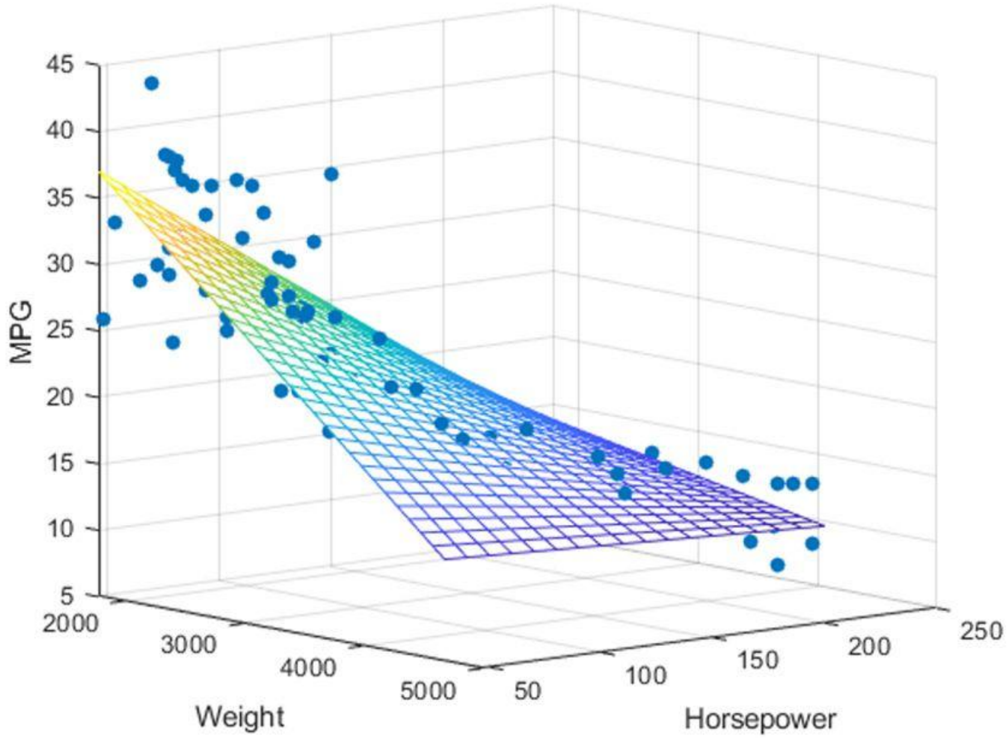
$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \eta \nabla J(\boldsymbol{x}_i)$$

(there are other ways to solve a least squares problem)

https://www.ibm.com/cloud/learn/gradient-descent

Regression

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1q} \\ 1 & x_{21} & x_{22} & \cdots & x_{2q} \\ 1 & x_{31} & x_{32} & \cdots & x_{3q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nq} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_q \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{pmatrix}$$



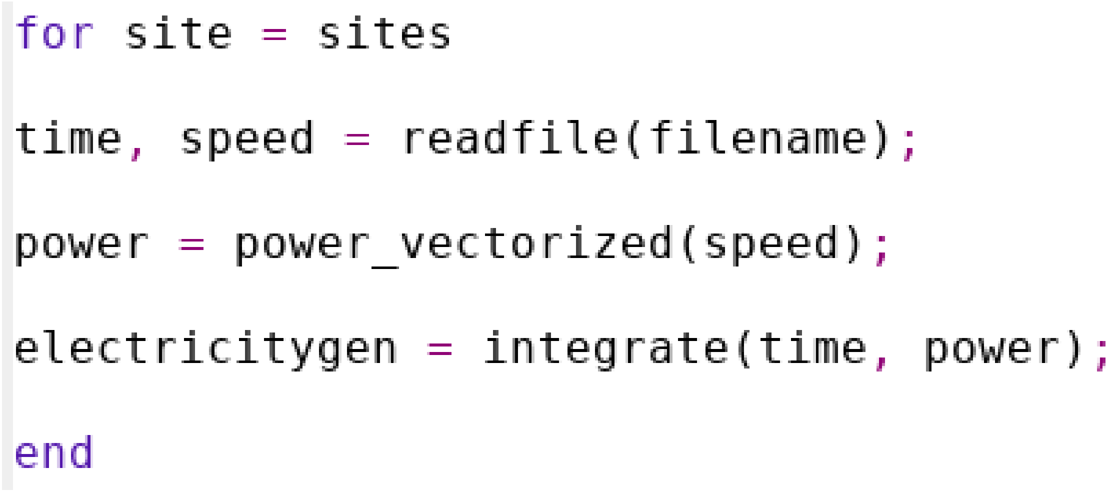
https://ch.mathworks.com/discovery/linear-regression.html

Coding

“Wishful programming”

- To solve the problem, think about how you want to transform the data and what operations you need to perform to transform them
- Write out the operations without filling in the details (pseudo-code)
- Often programming can be done without the computer
- (90% of coding is spent debugging)

Example for Éolienne



```
for site = sites
time, speed = readfile(filename);
power = power_vectorized(speed);
electricitygen = integrate(time, power);
end
```

Organizing code

- MANY ways to solve the same problem

- Some are preferred over others
  - Readability*
    - parsimonious – no more variables and functions defined than necessary
    - short - also less place to introduce bugs
    - closer to mathematical formulae
    - declarative (describes *what* is being computed rather than *how* it's being computed)
  - Efficiency
    - vectorization
    - avoid defining same operation in multiple places

- *There are two different ways in which code can be readable, and they can be orthogonal:
  - easy to read and understand what the whole program or sections of the program is doing at a glance
  - easy to read and understand what each step is doing

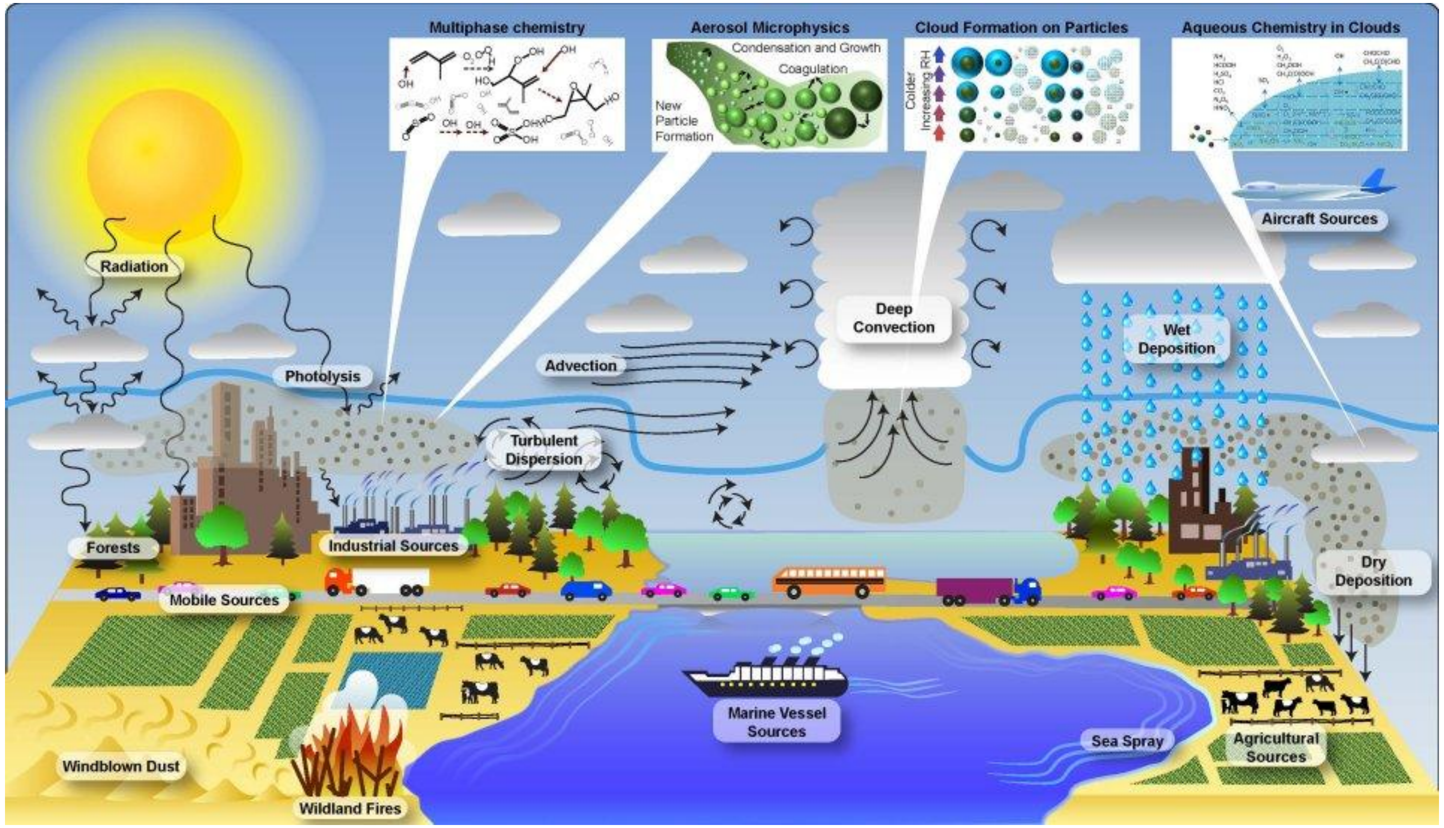
Finding a solution to your project problem

The value of simple models

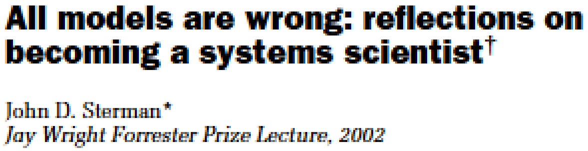
(but don’t make it too simple)

“Develop a computational model (e.g., simulation) of a real or hypothetical system to study its behavior.”

Managing complexity



https://www.epa.gov/sciencematters/cmaq-celebrating-25-years-air-quality-modeling-excellence

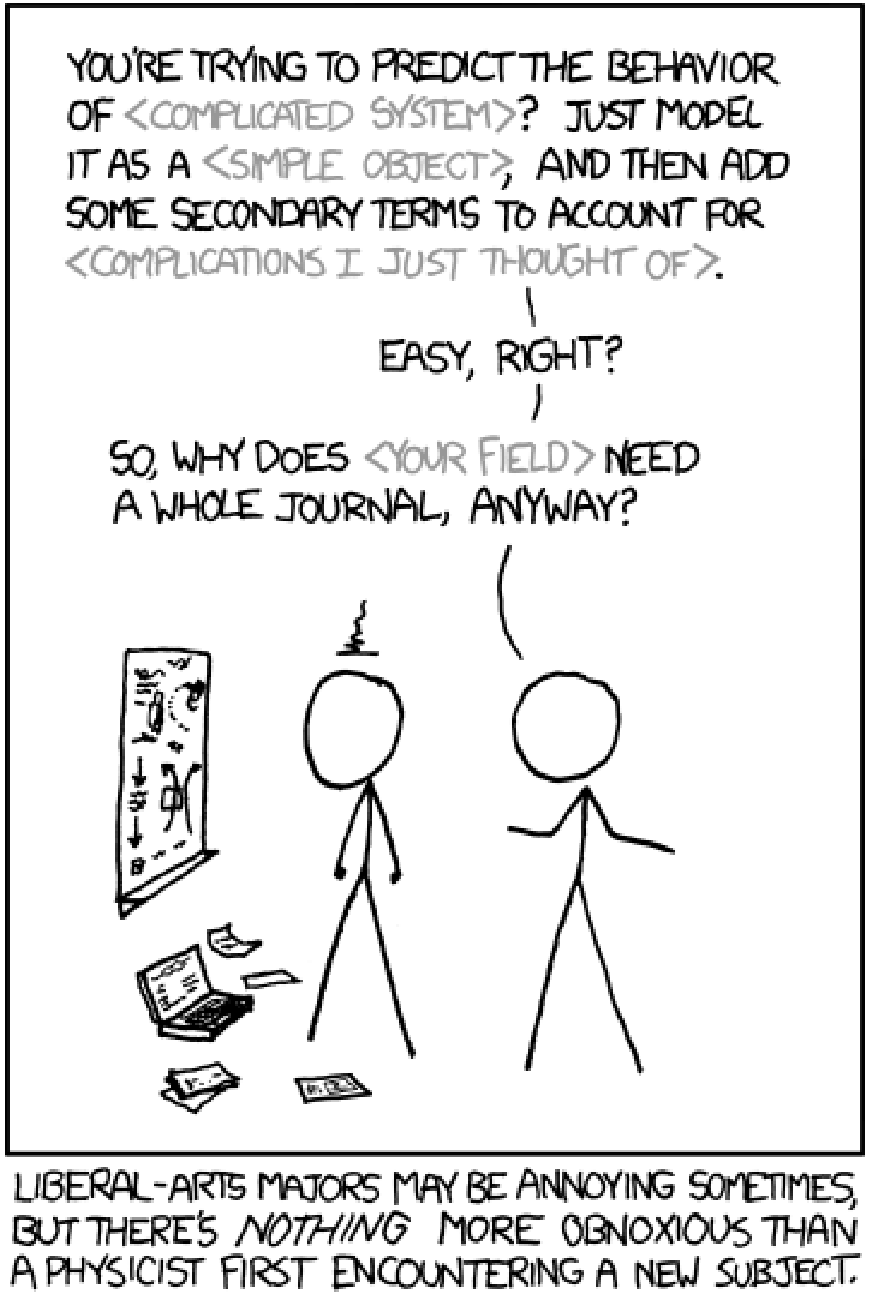


All models are wrong: reflections on becoming a systems scientist†

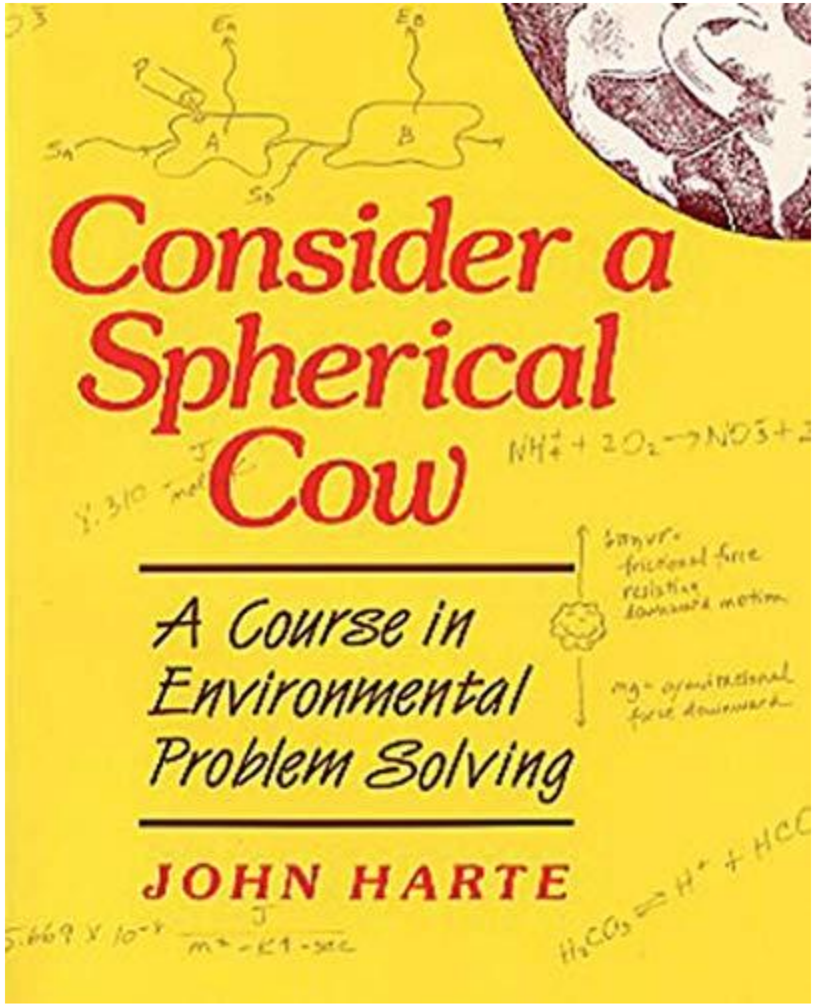
John D. Sterman*

Jay Wright Forrester Prize Lecture, 2002

- All models are wrong in that they are simplifications of reality – but they can be useful and we can learn from them.
- We should be aware of their limitations


YOU'RE TRYING TO PREDICT THE BEHAVIOR OF <COMPLICATED SYSTEM>? JUST MODEL IT AS A <SIMPLE OBJECT>, AND THEN ADD SOME SECONDARY TERMS TO ACCOUNT FOR <COMPLICATIONS I JUST THOUGHT OF>.
EASY, RIGHT?
SO, WHY DOES <YOUR FIELD> NEED A WHOLE JOURNAL, ANYWAY?
LIBERAL-ARTS MAJORS MAY BE ANNOYING SOMETIMES, BUT THERE'S *NOTHING* MORE OBNOXIOUS THAN A PHYSICIST FIRST ENCOUNTERING A NEW SUBJECT.

Source: XKCD



Consider a Spherical Cow

A Course in Environmental Problem Solving

JOHN HARTE

https://en.wikipedia.org/wiki/Spherical_cow

Tips

Start with the question

Identify the simplest model / data set to answer this question

- You should be able to (roughly) justify your simplifications – sometimes it is a matter of time constraints of the project – speculate on its impact on your results, and improving it can be mentioned in your outlook
- Are you capturing the important physics / mechanisms / features of your system?

State whether the answer is within the expected order of magnitude

- Does it make sense?
- Can you compare it to any reference value?

Note that you can change/redefine your project after the project proposal