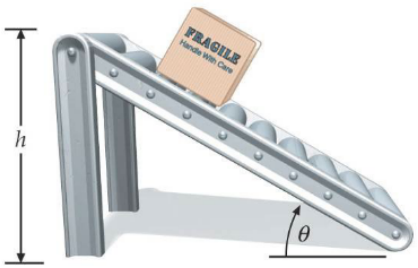
SW 1: Introduction

Model's three properties

- **Mapping**: models act as a representation of natural or artificial originals and can be models in turn;
- **Reduction**: models function as abstraction. They do not capture every attribute of the original; instead, they isolate and retain only those attributes relevant to the specific objective, intentionally omitting detail to manage complexity and focus on the problem at hand;
- **Pragmatic**: models function as utilitarian substitutes. They do not replace the original universally but serve as a representative for a specific user (subject), within a defined time frame, and for a particular purpose or operation.

Example



- **Generaliz.**: point mass sliding down an inclined plane;
- **Mapping**: box as mass, conveyor slope as an angle θ, vertical drop as height h, gravity;
- **Reduction**: no structure flexibility, no air movement, no friction, no rollers $\to$ flat plane;
- **Pragmatic**: it allows a, v_f, t of the box to be calculated, it enables the prediction of how to build a belt mockup.

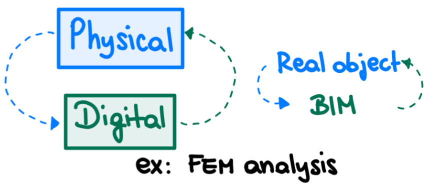
Digital representation

- - - - - - - - - - - ► Manual Data Flow (Offline)

———————————► Automatic Data Flow (Real-time)

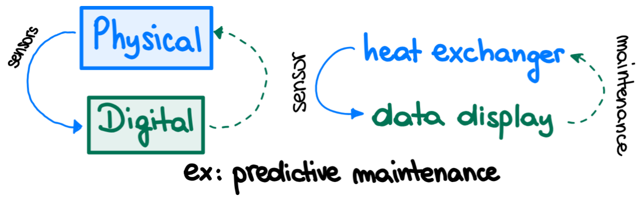
Digital model (simulation)

No direct connection between digital and physical object:



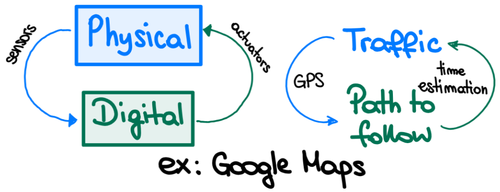
Digital shadow

Unidirectional, automated data flow from physical object to digital model:



Digital twin

Automated data exchange between physical object and model:



Role of time

Stationary behavior

Steady-state operation: $\dot{m}_\alpha = \dot{m}_\omega$

Dynamic behavior

Non stationary/transient/unsteady: $\dfrac{dm}{dt} = \dot{m}_\alpha - \dot{m}_\omega$

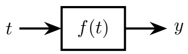
Governing dynamics

Empirical (black box)

Data based, without direct physics link. (ex: machine learning, fitting of functions)

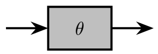
Physics-based (white box)

Based on physical laws.
(ex: conservation of mass)



Grey-box (hybrid)

Combining physics and data parameters.



Role of space

Point model (0D)

Assumes the whole system is perfectly mixed. (ex: ideal mixer with isotropic distribution). Software: Excel, MATLAB

Linked point

Connects several simple models together to create a basic network or layout. (ex: space shown via linking of 0D-models). Software: Simulink, Modelica

Spatial model (1-3D)

Considers real position of state variables or entities; spatial relationships affect the dynamics. (ex: real mixer with anisotropic, heterogeneous distribution). Software: COMSOL, ANSYS, AutoCAD, REVIT

Example with a heat pump

- **Purpose**: digital shadow $\to$ automated data;
- **Governing dynamics**: physics-based $\to$ based on thermodyn. laws;
- **Time**: time dependent, dynamic behavior $\to$ heating load, power of the hp, on/off cycles;
- **Space**: linked point $\to$ el. inputs, thermal energy exchange, 4 components to monitor.

Solvability of models

Analytical

Closed formula as solution. Only for simple problems.

$$A = \frac{x^3}{3}\Big|_0^2 = \frac{8}{3}$$

Numerical

Numerical approximation. For complex problems.

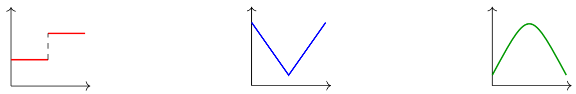
$$A \approx \sum_{i=1}^{n} f(x_i)dx \approx 2.6667$$

Further modelling properties

Linear vs Non-linear

Linear Non-Linear

Continuity vs Differentiability



Non-Cont Cont/Non-Diff Differentiable

Deterministic vs Stochastic

Deterministic Stochastic

Modelling approaches

Top-down

Largest components broken down into smaller. ex: marble block sculpture, railway network.
⊕ Efficient model, ⊖ Misses details

Bottom-up

Individual components combined into larger. ex: LEGO model, human body.
⊕ Detailed model, ⊖ Complex

SW2: How to model a system

1. Problem formulation
2. Mathematical representation
3. Mathematical analysis
4. Interpretation and evaluation of results

Problem formulation

Task 1 - Defining goals

What do we want to achieve?
How well/closely does our model need to represent reality?
What could be the goals for this specific system?

Task 2 - Characterize the system

What are the relevant parameters and variables of the system?
What are the system boundaries?
What are the inputs and outputs of the system?

Task 3 - Simplify and idealize the system

Still reproduce the significant behaviors of the system, while reducing complexity.
Reduce model to the main parameters and variables (ex. for hp: COP? Max. power? Avg power? Yearly values? Temperature levels?).

Mathematical formulation

Task 1 - Identify fundamental theories and laws

If no laws are available, use ad-hoc or empirical data to derive relationships:
Thermodynamic laws, material properties, ad-hoc
$P_{out} = COP(T_{amb}) \cdot P_{in}$

Task 2 - Derivation of relationships

Transfer system into a mathematical formulation.
Top-down (black/grey box): Use generic relationship, data from measurement to determine parameters. For more complex systems, add more parameters. Use techniques such as machine learning.
Bottom-up: Detailed physical modelling of the device. Physical laws to describe each component. Exact geometry, material properties, boundary conditions.

Task 3 - Reduce to standard mathematical problem

Simple algebra, linear programming, differential equation, diffusion problem, wave propagation, FEM problem, using suitable methods and software/programming tools.

Interpretation and evaluation of the results

Task 1 - Calibration of results

Use existing data to calibrate the model.

Task 2 - Validation

Check underlying physics law, such as energy or mass conservation, compare to known solutions, look at extreme cases, compare to measured data.
→ What is it and why do we have to do it?

Before the modelling:
What do we model how?:

a) Aims: does the model describe the process under test?
b) Output: does the model provide the required output to describe the process?
c) Type: is the type of the model suitable to describe the process?

During modelling:
Can we reproduce the measurements?
Does the model behave like to system under study?

d) Fitting data: does the model reproduce the fitting data? How to measure accuracy?
e) Reproducing novel data: does the model also predict novel measurement data correctly?
f) Sensitivity analysis: does the model predict the behavior of the system correctly when system parameters are changed?

After modelling:
Does the model also work with new data?

g) System potentially changed.
h) Differences in system behavior is only manifest in new experiments.

SW 3: Data-based modeling

Linear regression

Used to find a linear function $y = f(x) = a + bx$ that best fits a dataset (x_i, y_i).

Least squares method

Minimize the sum of squared errors (SSE):

$$S = \sum_{i=1} (y_i - (a + bx_i))^2$$

If measurement uncertainties Δy_i exist, weight the error:

$$S_i = \frac{y_i - y(x)}{\Delta y_i}$$

Optimal parameter formulas

Finding a and b when S is minimal:

$$\frac{\partial S}{\partial a} = 0 \quad ; \quad \frac{\partial S}{\partial b} = 0$$

Slope b:

$$b = \frac{\sum_i x_i y_i - \frac{1}{n}\left(\sum_i x_i\right)\left(\sum_i y_i\right)}{\sum_i x_i^2 - \frac{1}{n}\left(\sum_i x_i\right)^2}$$

Intercept a:

$$a = \bar{y} - b\bar{x}$$

where:

$$\bar{x} = \frac{\sum_i x_i}{n} \quad ; \quad \bar{y} = \frac{\sum_i y_i}{n}$$

Quality of fit ($\mathbf{R^2}$)

The coefficient of determination R^2 indicates the percentage of variation explained by the model:

$$R^2 = \frac{\sum_i \left(y(x) - \bar{y}\right)^2}{\sum_i \left(y_i - \bar{y}\right)^2}$$

- $R^2 = 1$ (100%): the model explains all data;
- $R^2 = 0$ (0%): the model doesn't (random).

Multilinear regression

Used when the target depends on multiple variables:

$$y(x_1, \dots, x_n) = a + b_1 x_1 + \cdots + b_n x_n = a + \sum_{j=1}^{n} b_j x_j$$

Non-linear regression

The goal is to fit data using non-linear functions when the underlying process is not linear.

Linearization techniques

Function	Equation	Trasformation	Variables
Exp	$y = ae^{bx}$	$\ln y = \ln a + bx$	x vs $\ln y$
Power	$y = ab^x$	$\ln y = \ln a + x \ln b$	x vs $\ln y$
Inverse	$y = \frac{a}{x}$	$\frac{1}{y} = \frac{x}{a}$	x vs $\frac{1}{y}$
Square offset	$y = ax^2 + b$	$y = a(x^2) + b$	x^2 vs y
Root / Cubic	$y = \sqrt{ax^3 + b}$	$y^2 = ax^3 + b$	x^3 vs y^2

Maximum likelihood method (MLE)

Determines the parameters of a probability distribution that best describes a dataset, independent of histogram binning.

Likelihood function

Defines as the product of probability densities for all data points:

$$L(\sigma, \mu) = \prod_i f(x_i, \sigma, \mu)$$

Log-likelihood

To simplify calculation and avoid small numbers, minimize the negative logarithm:

$$-\log L = -\sum_i \log(f(x_i, \sigma, \mu))$$

Common distribution

Normal distribution:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

Weibull distribution (Reliability):

$$f(x) = \begin{cases} \lambda k(\lambda x)^{k-1} e^{-(\lambda x)^k}, & x > 0 \\ 0 & \text{else} \end{cases}$$

Weibull cumulative distribution function

$$F(x) = \int_{-\infty}^{x} f(u)\,du = \begin{cases} 1 - e^{-(\lambda x)^k} & for x > 0 \\ 0 & \text{else} \end{cases}$$

SW4: Modelling with ODEs

Fundamentals of ODEs

An ODE contains functions of one independent variable and their derivatives.

Ordinary (ODE)

Involves one independent variable:

$$\frac{d^2x}{dt^2} = -g$$

Partial (PDE)

Involves multiple independent variables:

$$\frac{d^2u}{dt^2} = c^2 \frac{d^2u}{dx^2}$$

Analytical solution method

Separation of variables

Used when terms involving y and x can be moved to opposite sides.

Variation of parameters

Used for inhomogeneous linear ODEs. General solution is the sum of the homogeneous solution and a particular solution.

Numerical solution methods

Euler method

A simple iterative method to approximate ODEs defined as $\frac{df}{dx} = g(x)$.

The approximation uses the finite difference slope:

$$\frac{df}{dx} \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

Iterative steps:

$$f(x_0 + \Delta x) = f(x_0) + g(x_0)\Delta x$$

Modelling principles

Balance equations

Based on the conservation principle:

$$\frac{d}{dt} f(t) = f(t_\alpha) - f(t_\omega)$$

Example in a capacitor

$$U_0 = U_R + U_C \Rightarrow U_0 = RI + \frac{Q}{C} = R\frac{dQ}{dt} + \frac{Q}{C}$$

Mechanics and forces

Equation of motion is derived from Newton's second law $F_{net} = ma$.

Example of a falling drop with drag

$$m\dot{v} = mg - bv \Rightarrow v(t) = \frac{mg}{b}\left(1 - e^{-bt/m}\right)$$

Growth and decay

Describes processes where a quantity increases or decreases over time.

$$\frac{dN}{dt} = kN \Rightarrow N(t) = N_0 e^{kt}$$

with half-time / doubling factor τ:

$$\tau = \left|\frac{\ln 2}{k}\right|$$

Example of logistic growth

$$\frac{dN}{dt} = KN(t) - \frac{K}{K}N^2 \Rightarrow N(t) = \frac{L}{1 + \left(\frac{L}{N_0} - 1\right)e^{-kt}}$$

Recipe to derive the equation of motion

1. Make a sketch of the situation;
2. Define the coordinate system and select variables of interest;
3. Identify all forces and momenta;
4. Formulate the equation of motion;
5. Solve it.

Linear algebra and systems of ODEs

Matrix representation

System of equations:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

Matrix form ($Ax = b$):

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

If $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, then $\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix}$

Inversion and diagonalization

Inverse matrix R^{-1}

$R \cdot R^{-1} = I$ (Identity matrix).

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Diagonalization Special matrices can be rewritten as:

$$A = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

This transforms the matrix into a diagonal matrix containing eigenvalues λ.

Why is it called linear algebra

Linearitazion

Complex, non-linear functions can be approximated by linear functions in a small neighborhood of a point a:

$$f(x) \approx f(a) + f'(a)(x - a)$$

Benefit of solving ODEs

If A were a number, $\dot{x} = Ax$ would solve to $x(t) = ke^{At}$. Since A is a matrix, if we diagonalize it using eigenvalues λ, the solution becomes a mixture of exponentials:

$$x(t) = R^{-1} \begin{pmatrix} k_1 e^{\lambda_1 t} & 0 & 0 \\ 0 & k_2 e^{\lambda_2 t} & 0 \\ 0 & 0 & k_3 e^{\lambda_3 t} \end{pmatrix} R$$

Solvability of linear systems

Geometric interpretation:

Solving $Ax = b$ is finding the intersenction of lines/planes.

- **Case 1**, consistent: lines intersect at exactly one point;
- **Case 2**, inconsistent: lines are parallel and distinct, there is no solution;
- **Case 3**, infinite solutions: lines are identical and overlap completely.

Determinant

A scalar value derived from a square matrix that tells us if it is invertible. If det $\boldsymbol{A} = 0$, the matrix is not invertible.

2x2 formula: For $\boldsymbol{A} = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$, det $\boldsymbol{A} = ad - bc$.

3x3 formula: For $\boldsymbol{A} = \left[\begin{smallmatrix} a & b & c \\ d & e & f \\ g & h & i \end{smallmatrix}\right]$,

$$\det \boldsymbol{A} = a_{11} \det \begin{bmatrix} e & f \\ h & i \end{bmatrix} - a_{12} \det \begin{bmatrix} d & f \\ g & i \end{bmatrix} + a_{13} \det \begin{bmatrix} d & e \\ g & h \end{bmatrix}$$

$$\det \boldsymbol{A} = \sum_{j}^{n} a_{1j} C1j, \quad \underbrace{C1j = (-1)^{1+j} \det \boldsymbol{A}_{ij}}_{\text{Cofactors}}$$

The Eigenvalue problem

For a square n×n matrix $\boldsymbol{A}$, we look for a Eigenvector x and a Eigenvalues a such that:

$$\boldsymbol{A}x = \lambda x$$

Calculation method:

1. Solve the characteristic equation det$(\boldsymbol{A} - \lambda I) = 0$
2. This result in an n-th order polynomial $(a_1 \lambda^n + \cdots = 0)$
3. The roots of this polynomial are the Eigenvalues.