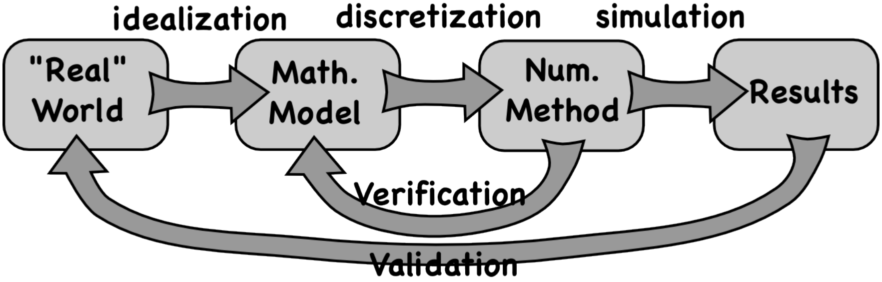
The Modeling and Simulation Process

The Full Loop

Melanie Ganz-Benjaminsen
Kenny Erleben
Department of Computer Science

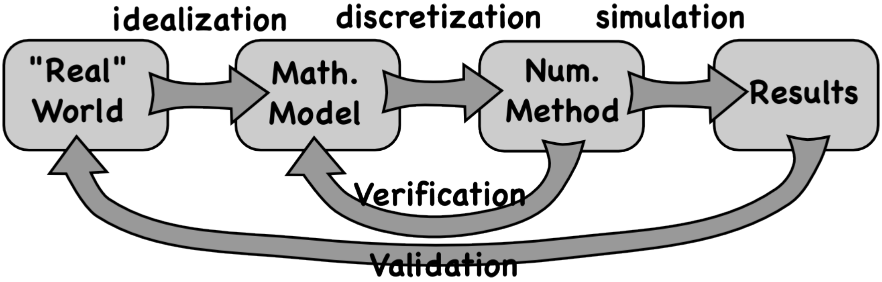
University of Copenhagen

A Conceptual Data Flow of the Process



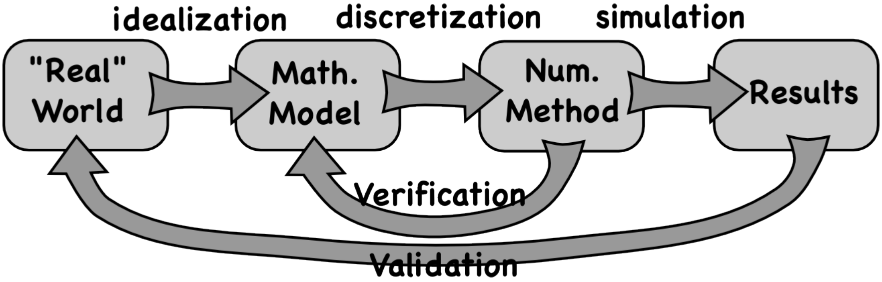
Data flow diagram of the modeling and simulation process.

Idealization



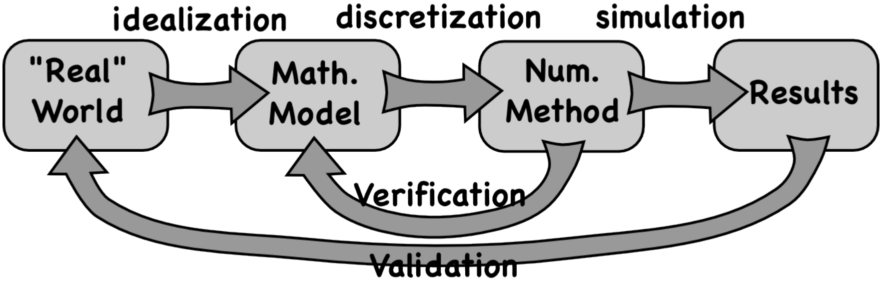
Stating a mathematical model using for instance the laws of physics.

Discretization



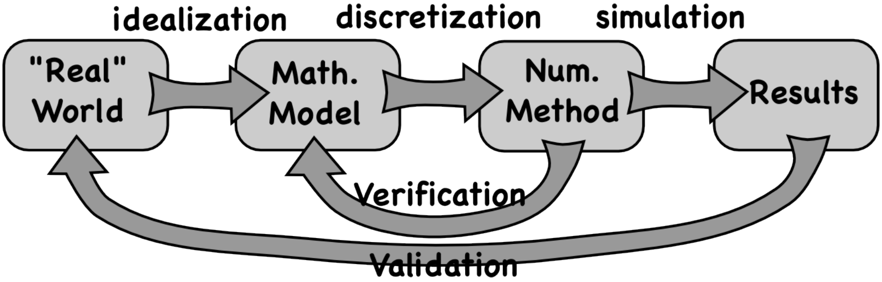
Converting the mathematical model into a discrete model and finally applying a numerical method to the discrete model.

Simulation



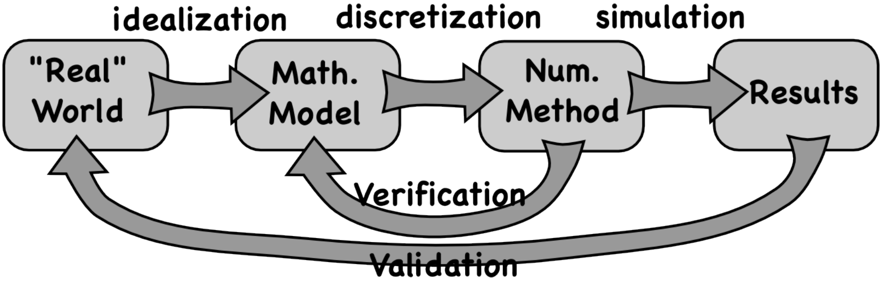
Using the numerical method to compute simulation results.

Verification



Ensures the numerical method (ie. discrete model) agrees with the mathematical model
(Are you computing solutions to the problem?)

Validation



Concerns whether the simulation model can be applied to a real-world problem (Are you solving the right Problem?)

Classification – A Quick Overview

One way to subclass a paradigm/method is to discriminate between types of **Computational Meshes**

- Mesh Free Methods (Particles)
- Regular Meshes (Uniform Regular Grids)
- Irregular Meshes (Tetrahedral Meshes)

Another way is the **Discretization Methods**

- Mass Spring Systems/Smoothed Particle Hydrodynamics
- Finite Difference Methods
- Finite Volume Methods
- Finite Element Methods
- Explicit/Implicit etc.

This course will cover many of these details.

Simulation Example of Deformable Model

Typically one has to solve the task/problems

- Pre-processing
  - (Data Acquisition)
  - Meshing
- Run-time
  - Computing Deformation
  - Performing Collision Detection
  - Computing Contact/Constraint Forces
- Post-processing
  - Visualizing the simulation results

A Typical Simulation Loop

1 : **algorithm simulation loop**
2 : **while running**
3 : $\mathcal{C}_{\text{contacts}} \leftarrow$ **collision detection**$(\mathbf{x}^t)$
4 : $\mathbf{f} \leftarrow$ **compute contact forces**$(\mathcal{C}_{\text{contacts}})$
5 : $\mathbf{v}^{t+\Delta t} \leftarrow$ **velocity-update**$(\mathbf{f}^t, \mathbf{v}^t)$
6 : $\mathbf{x}^{t+\Delta t} \leftarrow$ **position-update**$(\mathbf{x}^t, \mathbf{v}^{t+\Delta t})$
7 : **draw**$(\mathbf{x}^{t+\Delta t})$
8 : $t \leftarrow t + \Delta t$
9 : **end**
10 : **end algorithm**

Notices: Line 5-6 is time integration of equations of motion (Semi-Implicit Euler method)

In This Course

- We mainly focus on the discretization and simulation processes from a Computer Science perspective.
- We will NOT go deep into physics but some details will be touched upon.
- There will be math in this course. First-year undergraduate level in linear algebra and calculus should suffice.