Computer Simulation

An overview of simulation principles and technologies

DI Stefan Hattinger BSc

2.1 - Computer Simulation Overview

Core Concepts:

- **2.2 Static Simulation**: Time-invariant systems and models.
- **2.3 Dynamic Simulation**: Systems that change over time.
    - Continuous and Discrete approaches.
    - Hybrid methods.

Learning Goals:

- Understand the differences between static and dynamic simulations.
- Explore real-world applications and computational techniques.

2.2 - Static Simulation

2.2.1 Definition and Characteristics

Models systems that do not change over time. The system has no state.

- **Pros**: Simpler models, faster computation.
- **Cons**: Limited to time-invariant systems, lacks insight into dynamic behavior.

2.2.2 Example Applications

- **Application 1**: Evaluating system reliability (e.g., server downtime prediction).
- **Application 2**: Predicting steady-state performance of systems (e.g., a power grid under constant load).
- **Example Tools**: MATLAB, Excel Solver, LINGO.

2.3 - Dynamic Simulation

Core Types:

- **2.3.1 Continuous Simulation**: Models continuous changes in system state.
- **2.3.2 Discrete Simulation**: Models systems where changes occur at discrete points.
- **2.3.3 Hybrid Simulation**: Combines continuous and discrete modeling approaches.

2.3.1 - Continuous Simulation

2.3.1.1 Continuous Simulation Overview

Simulates system behavior over time, tracking changes continuously.

- **Characteristics**: Focuses on systems described by differential equations (ODEs, PDEs).
- **Pros**: Models systems with smooth changes over time.
- **Cons**: Requires complex mathematical formulations and computational resources.

2.3.1.2 Ordinary Differential Equations (ODE)

Models the rate of change in a system based on its current state.

- **Example**: Newton's Law of Cooling, population growth models.
- **Pros**: Precise modeling of smooth dynamic systems.
- **Cons**: Difficult for large-scale systems or nonlinear models.
- **Example Software**: MATLAB, Simulink, Mathematica.

2.3.1.3 Finite Element Method (FEM)

FEM Breaks a complex system into smaller parts to approximate solutions to PDEs.

- **Pros**: Highly accurate for mechanical, structural simulations.
- **Cons**: High computational cost, complex setup.

Applications

- Structural analysis, heat transfer, fluid dynamics.
- **Example Software**: ANSYS, COMSOL, Abaqus.

2.3.1.4 Computational Fluid Dynamics (CFD)

CFD Simulates fluid flow and the interaction with physical objects.

- **Pros**: Provides detailed insights into fluid systems.
- **Cons**: Computationally expensive, requires specialized knowledge.

Applications

- Turbomachinery, aerodynamics, HVAC systems.
- **Example Software**: ANSYS Fluent, OpenFOAM.

2.3.1.5 Analytic Methods

Analytic Methods solve equations mathematically for exact solutions.

- **Pros**: Yields precise solutions, no computational errors.
- **Cons**: Only feasible for simple systems, cannot handle real-world complexity.

Example Application

- Simple harmonic motion, ideal gas law.

2.3.1.6 Numeric Methods

Numeric Methods provide approximate solutions for complex systems when analytical solutions are impractical.
- **Pros**: Applicable to a wide range of real-world problems.
- **Cons**: Accuracy depends on step size, introduces computational errors.

Methods

- **Euler Method**: Basic approach for solving ODEs.
  - **Pros**: Simple, fast.
  - **Cons**: Inaccurate for large step sizes.

2.3.1.6 Numeric Methods

- **Runge-Kutta Method**: More accurate numerical solution for ODEs.
  - **Pros**: Higher accuracy, stable.
  - **Cons**: Requires more computational resources.

Example Software

- **Example Software**: MATLAB, GNU Octave, Python (SciPy).

2.3.2 - Discrete Simulation

2.3.2.1 Discrete Simulation Overview

Models systems where changes occur at specific discrete points in time.
- **Pros**: Suitable for systems with distinct events.
- **Cons**: Less effective for continuous changes in state.

Applications

- Manufacturing systems, queue management, traffic systems.

Tools

- **Example Software**: AnyLogic, Simio, Arena.

2.3.2.2 Time-Driven Simulation

- **Time-Driven Simulation**: Advances the system state at fixed time intervals.
  - **Pros**: Simple to implement, effective for systems with uniform behavior.
  - **Cons**: Inefficient for event-heavy systems, can miss significant events.

Example Application

- Assembly lines, conveyor systems.

2.3.2.3 Event-Driven Simulation

- **Event-Driven Simulation**: Advances the system based on the occurrence of events.
    - **Pros**: Efficient for systems with distinct events, provides accurate tracking.
    - **Cons**: Requires detailed event modeling, complexity increases with system size.

Example Applications

- **Discrete Event Simulation (DES)**: Models systems like manufacturing where distinct events occur.
- **Queuing Theory**: Analyzes waiting lines or queues in systems such as call centers or traffic flow.
- **Example Software**: JaamSim, Arena, Simio, FlexSim.

2.3.3 - Hybrid Simulation

2.3.3.1 Hybrid Simulation Overview

Hybrid Simulation combines continuous and discrete methods to model complex systems.
- **Pros**: Models systems with both continuous processes and discrete events.
- **Cons**: More complex to set up and solve.

- **Example Application**: Manufacturing process with continuous machine operation and discrete human interventions.
- **Example Software**: AnyLogic, MATLAB/Simulink, FlexSim.

2.3.3.2 Application Areas

- **Manufacturing**: Combining continuous material flow with discrete production events.
- **Healthcare**: Continuous monitoring of patient vitals combined with discrete medical events.
- **Energy Systems**: Continuous energy consumption with discrete operational failures.
- **Example Software**: AnyLogic, FlexSim.

2 - Summary and Closing

- **Computer Simulation**: Covers static, dynamic, and hybrid approaches to modeling systems.
- **Key Techniques**: Analytic vs. numeric methods, ODEs, FEM, CFD, DES.
- **Applications**: Widely used in manufacturing, healthcare, energy, logistics.

Questions?