Computational Thinking

Kate Cahill
Ohio Supercomputer Center

Outline

- Bootcamp Outcomes
- Skills for Computational Science Projects
- Creating a Model
- Model Examples
- Steps for Problem Solving

Bootcamp Outcomes

- Understanding of Energy Justice Concepts
- Familiarity with High-Performance Computing (HPC) Tools
- Hands-on Experience with HPC Applications
- Collaboration and Networking
- Enhanced Data Analysis Skills
- Presentation and Communication

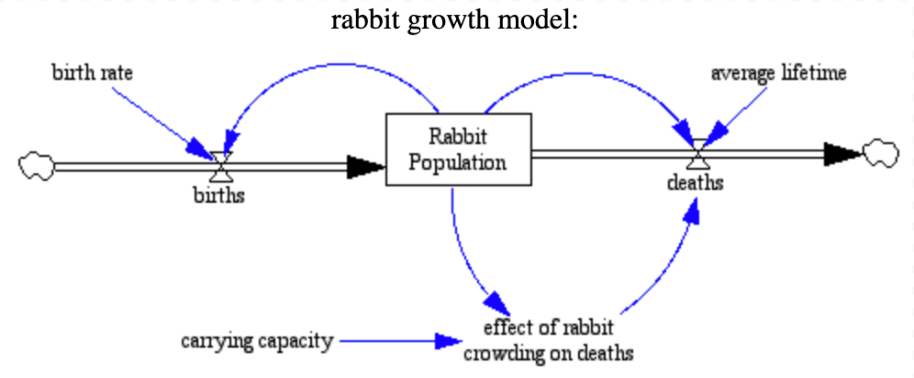
Computational & Data Science Skills

- Computational science provides skills needed in the present and future workforce
  - Understanding of modeling techniques that are used in research and business
  - Data management skills
  - Analytical skills
  - Teamwork skills
  - Communications skills

Based on Introduction to Modeling and Simulation with MATLAB and Python, Gordon & Guilfoos

Creating a Model

- Begin with basic modeling skills
  - What is a model?
    - Models of physical systems
    - Models of social systems
  - How do you create a model?
    - Understanding cause and effect
    - Representing the relationships in mathematical terms
  - How do you implement the model on the computer
  - How to you know if the model is "right"

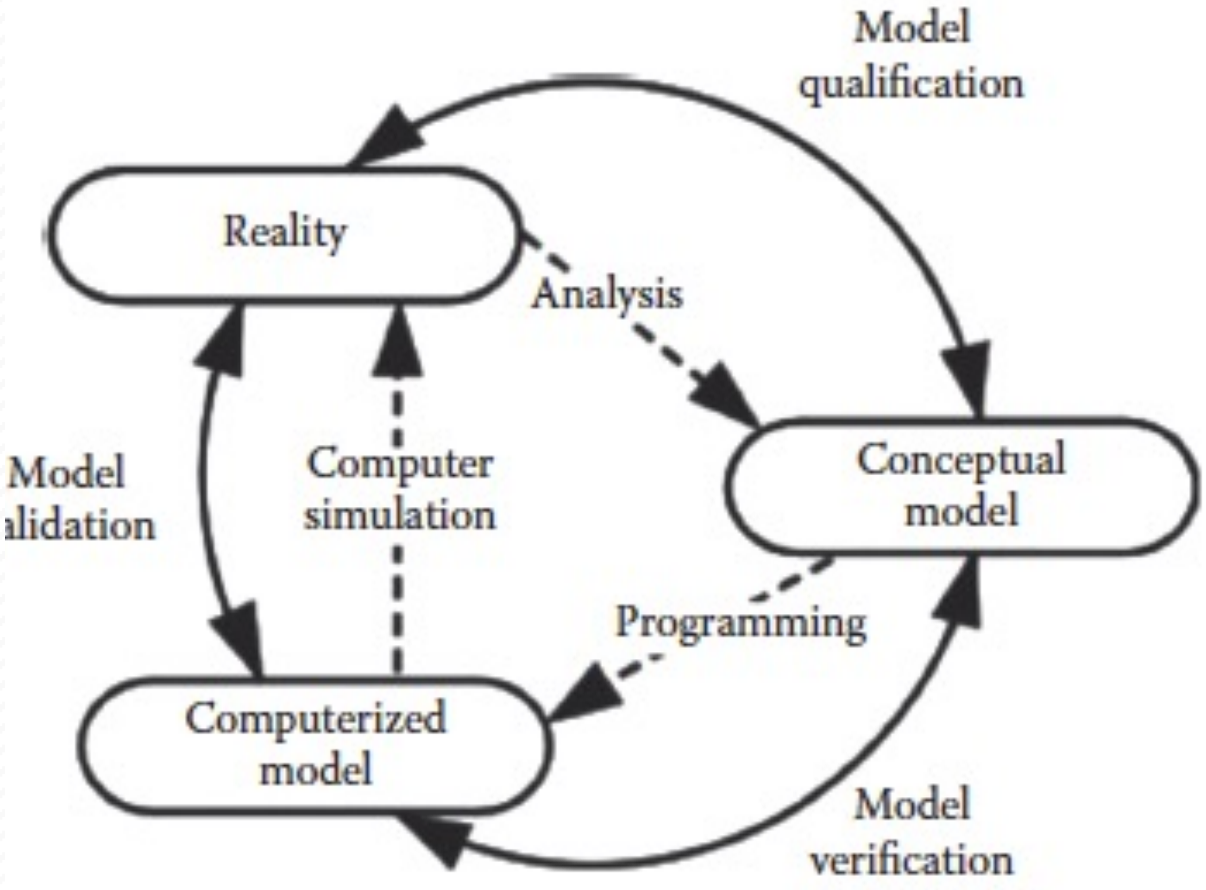


Gaining Skills

- Investigate how models have been used to gain insights about complex systems
  - Observe and manipulate built models on personal computers
- Use modeling tools to add new components to existing models
- Build new models of interesting systems
- Use the model to explore the system
- Present results – reports, presentations

Examples of Computational Modeling

- Some examples of modeling and simulation include parametric studies, systems dynamics models, and models of disease spread.
- Types of models:
  - Static/dynamic
  - Deterministic/stochastic (probabilistic)
  - Steady-state



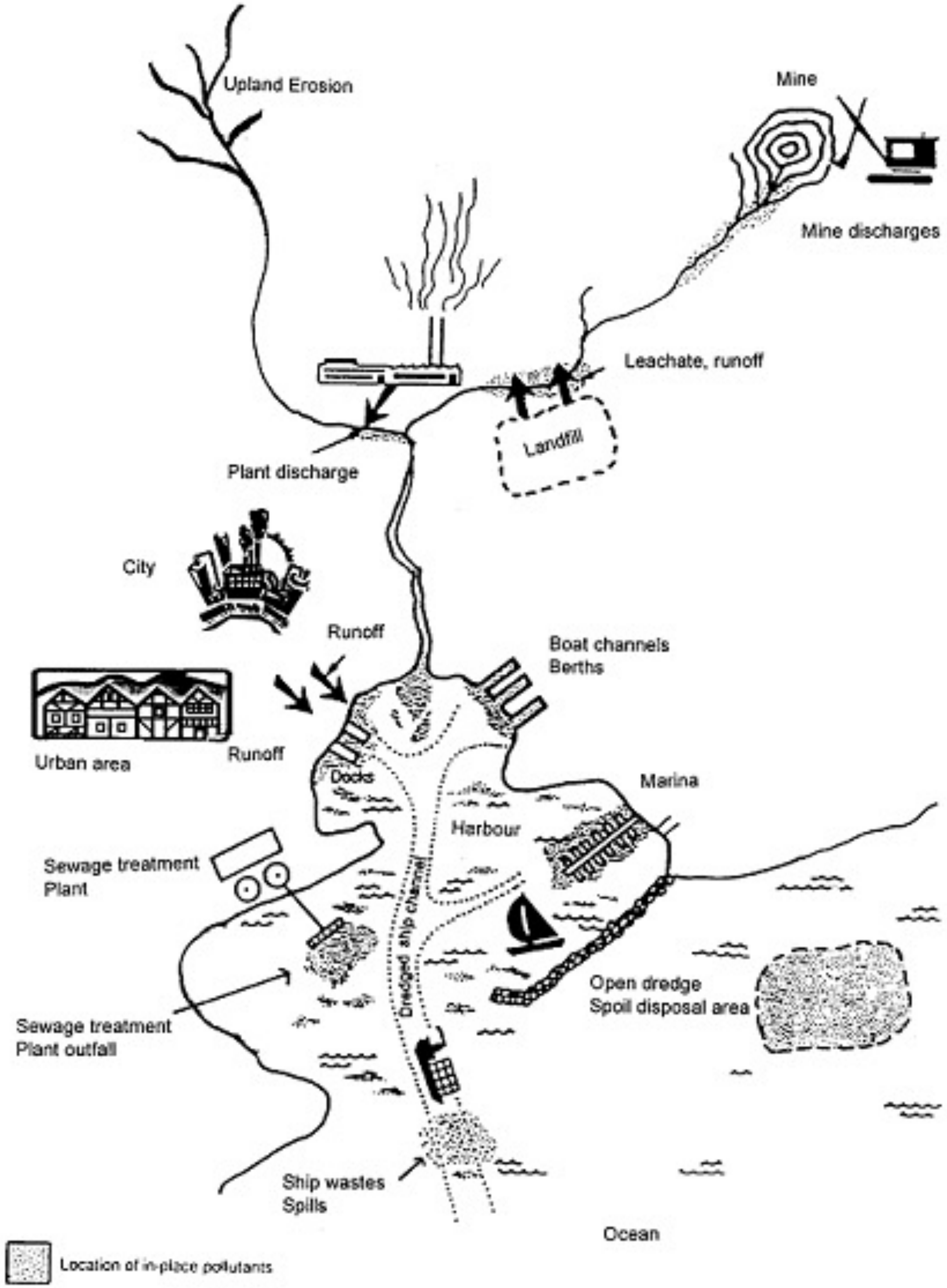
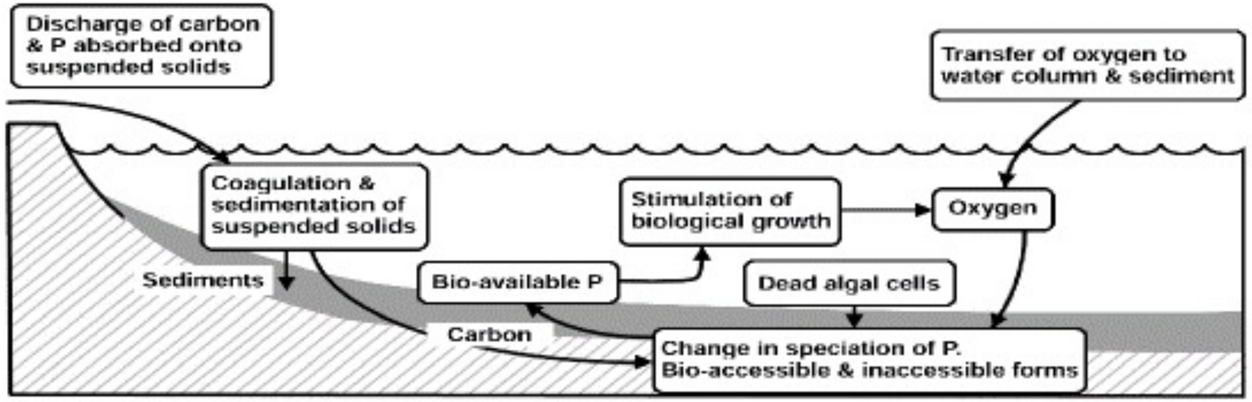
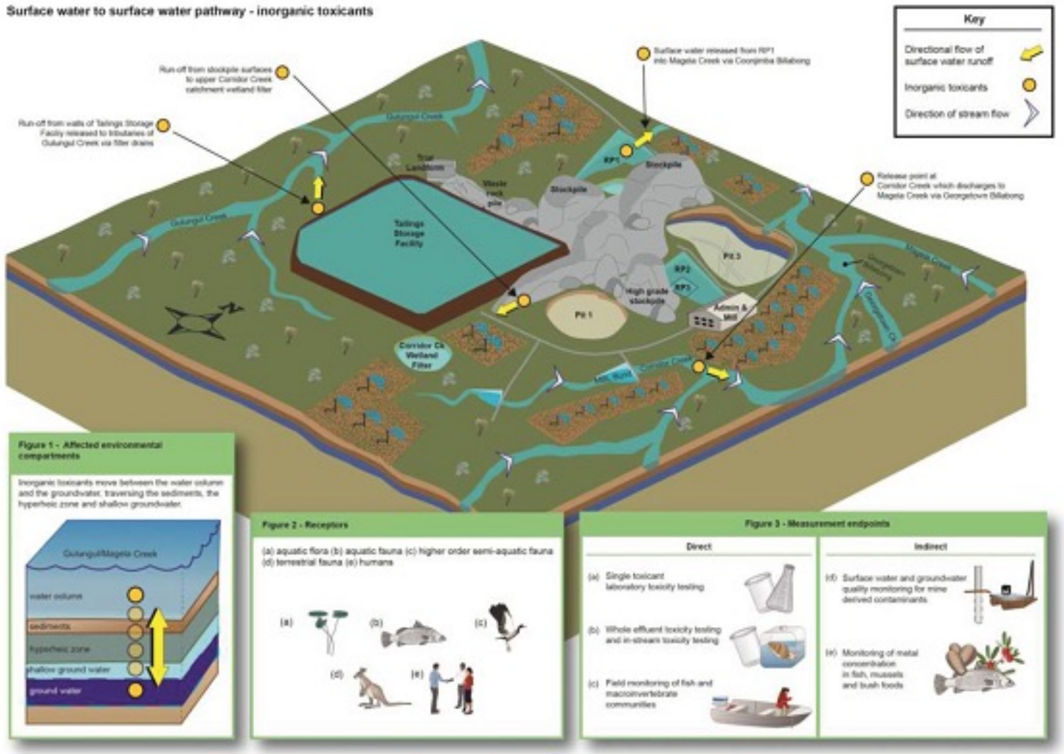


Figure 4 Conceptual model of in-lake or in-stream nutrient pathways



P = phosphorus



ps://www.waterquality.gov.au/anz-
idelines/resources/key-concepts/conceptual-models

Computational Modeling

- Models are ***simplifications or abstractions*** of real-world objects or phenomena that help us gain insights into complex systems.
  - A mathematical model is a representation of a phenomenon or system that is used to provide insights and predictions about system behavior.
  - Simulation is the application of a model to imitate the behavior of the system under a variety of circumstances.

- Creation of models is a science as well as art.
  - Abstract thinking,
  - visualization,
  - creativity

  are all part of the process

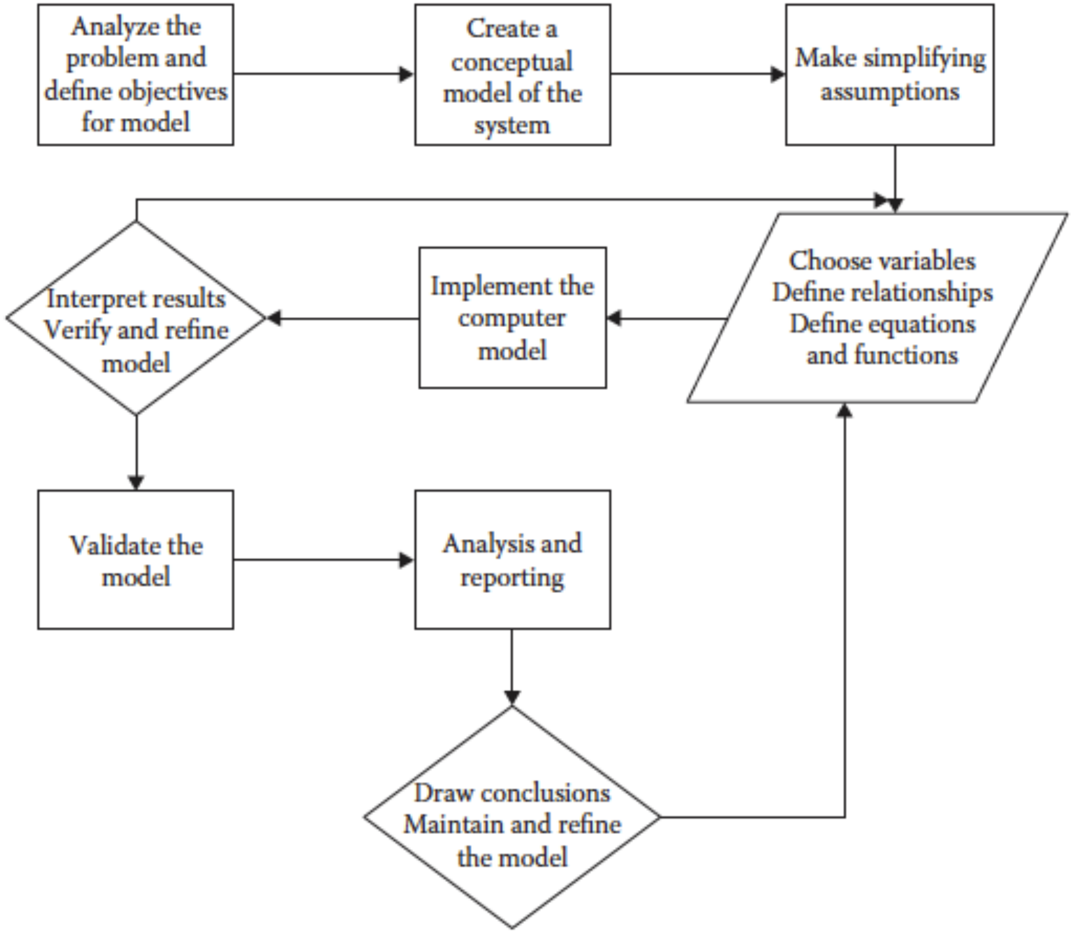


FIGURE 1.2 Major steps in the modeling process.

Social Network Analysis

- Social media allows the connection of billions of people across the globe
- Social network analysis examines and analyzes those connections
  - Email exchanges
  - Blog posts
  - Twitter
  - Wikis, etc.

Analyzing Social Media

- Behavior patterns on many issues
  - Who are the key leaders?
  - What are the key questions?
  - How does the discussion change over time?
  - Are there significant subgroups?
  - What are the strengths of the relationships?

Large Scale Example

- Six Degrees of Francis Bacon
  - Analysis of 16th – 17th Century social networks
  - Visualization tool built with JSON

http://www.sixdegreesoffrancisbacon.com/?ids=10003747&min_confidence=60&type=network

Built Models

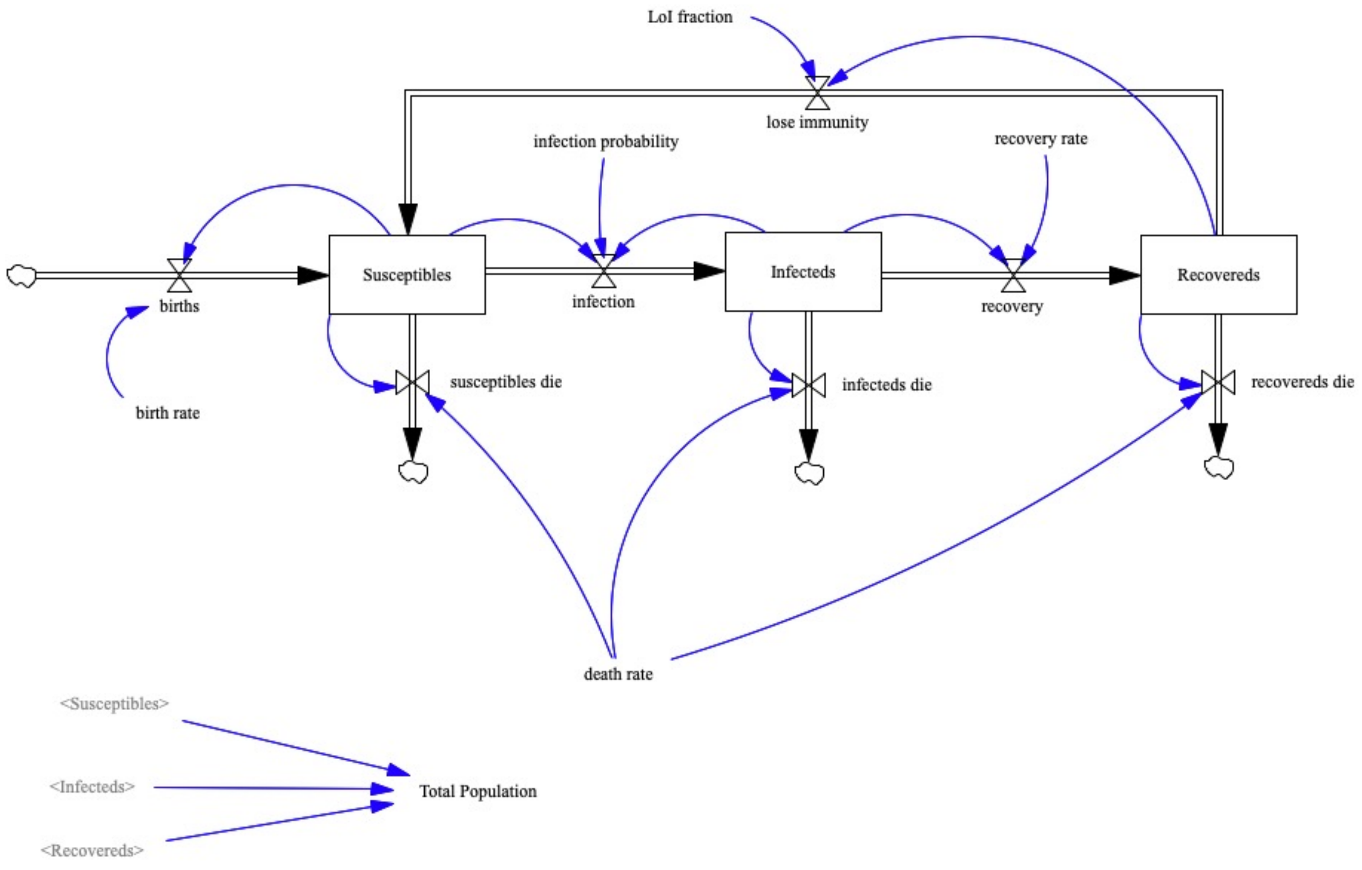
- Resources for Computational Modeling
  - https://phet.colorado.edu/
  - Find Shared science instructional modules - PHET

- Choose a model to investigate
- Work with the model to see what it is modeling
- What questions could you test with this model?

Example of Disease model

Describe system

- Relationships
- Equations



Agent Models Tell A Story

- Should describe the behaviors before model building
- Example of simple disease model
  - Agents: People
  - People are either healthy or sick
  - For a contagious disease, what is the story of the interaction of healthy and sick people?

Story behind this model

- Agents with two states
- If they “meet” there is some probability that a healthy person will become sick
- Examine the program syntax

- https://colab.research.google.com/drive/1SRhdhDHRzznd4OgeQpNOo1Et2p5NPtsB?usp=sharing

Problem Solving

Problem solving requires a combination of science and art. On the technical side, we have math, chemistry, physics, mechanics, and so on, whereas on the artistic side, we have things such as judgment, prior experience, common sense, know-how, and so on.

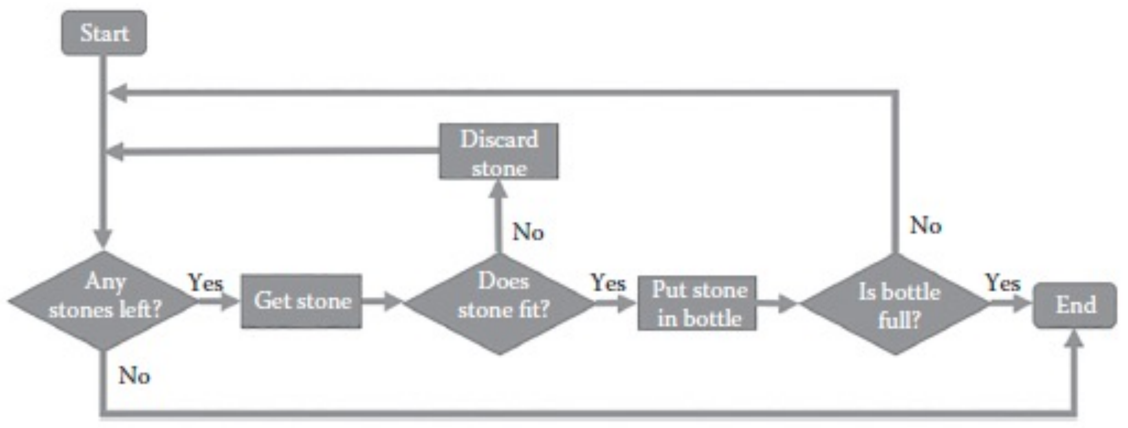
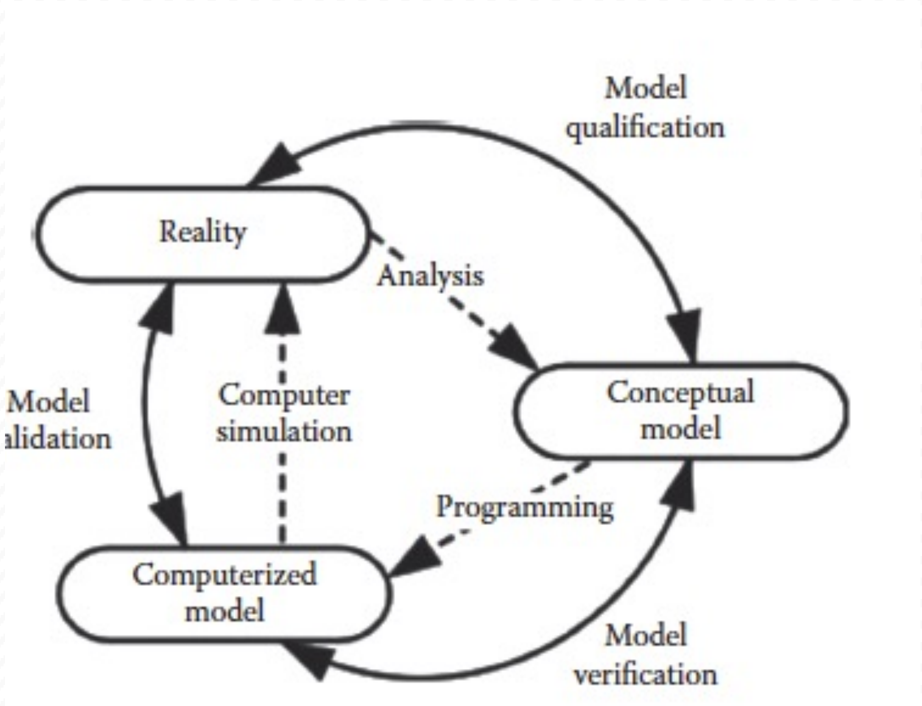


FIGURE 6.2 Bottle filling example flowchart.

1. ***Identify the problem***: You should be able to create a clear written statement of the problem to be solved. If you cannot do this, you will struggle to come up with a good solution.
2. ***Determine what is required for the solution***: What is known? What is unknown? Are there any restrictions or limitations? Are there any special cases?
3. ***Develop a step-by-step plan (algorithm)***: How are we going to solve the problem? What steps does our program need to take?
4. ***Outline the solution in a logic diagram***: It is usually helpful to outline the solution to a programming problem in a logic diagram. This step will likely create something that could be translated into a high-level version of the program used to solve the problem.
5. ***Execute the plan***: Write the code. Keep track of what works, and what does not (source control, such as git, is often helpful). It can be helpful to write and test small portions of the solution independently to make sure that those parts work as expected.
6. ***Analyze the solution***: Revise the plan and re-execute as needed. Keep the good parts of the plan, and discard the not so good parts
7. ***Report/document the results***: Share the results with collaborators and others in

Conclusion



- Models are used to gain insights into real world behavior
- Creating a model is an iterative process – build, assess, alter, reassess, validate
- Computational and data science involve creativity, collaboration, communication as much as math and coding