October 23, 2015

Computational Thinking

Southern University New Orleans Workshop

Steven Gordon (sgordon@osc.edu)

XSEDE

Extreme Science and Engineering Discovery Environment

NSF

Computational 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
- Inquiry-based education approach engages students in learning

Acquiring the Appropriate Skills

- 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"

Steps Toward Competency

- 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 in writing and orally

Progression of Technical Skills

- Using modeling tools on a personal computer
- Learning programming skills
- Advancing applied math skills
- Applying skills to the student's academic major
  - Starting with simple models on personal computers
  - Expanding to large-scale applications on supercomputers

Integrating Materials into the Curriculum

- Model competencies
- http://hpcuniversity.org/educators/competencies/

Goals for this Session

- Demonstrate the pedagogy for computational science education
- Progression of possible activities
  - Using complete models to demonstrate principles
  - Running models to gain insights into system behavior
  - Modifying models to relax assumptions
  - Building new models

What We Will Cover

- Introduce materials and models that can be incorporated for classroom use
- Introduce simple tools that can be used to build and demonstrate modeling techniques
- Provide a list of resources you can explore in detail later

Reference Materials

- https://www.osc.edu/~sgordon
  - Choose Workshop Materials
  - Then Links to other materials

Starting with Simple Models and Tools

- Can use simple models to illustrate modeling principles and definitions
- Models of change in space and/or time:

HAVE = HAD + CHANGE

- Which phenomena employ this basic concept?
- Example – go to datasets folder
- Open SimplePopulation.xls

Mostly hidden Excel Capabilities

- Slider bar
  - Access via Developer menu
  - File/options/customize ribbon
  - Choose All tabs – move Developer and turn it on
- Insert Scroll Bar
- Associate its value with a cell D11
- Value of 50
- Value of D8 is D11/100
- Close this spreadsheet

Another simple model

- I am the average of my neighbors
- Representation of a space by a matrix of values representing location and adjacency
  - 2D or 3D
- What is modeled in this way?

More Excel tricks

- Open saltdiffusion.xlsx
  - Note the matrix of numbers
  - Each cell is calculated as the average of its neighbors
  - In D5 insert 50 or = B5
  - Note the change of color
  - Now use the F9 key to incrementally calculate the results
- Turned calculation to manual
- Added formatting to cells based on value

More Examples

- Examples from several of the tools we will be using in this workshop along with lesson plans
- http://www.shodor.org/talks-new/

Getting Started

- Point your browser here:

- https://www.osc.edu/~sgordon
  - Choose Workshop Materials
  - Then Links to other materials

Built Models You Can Use in Classroom

- Scroll down the list to Java Applets
  - Choose histogram
  - Ability to examine the impact of categorization on the description of a distribution
- Now scroll to Resources for Computational Modeling
  - Find Shared science instructional modules - PHET

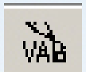
Explore Other Relevant Examples

- Look at sites relevant to your discipline
  - Computational physics
  - Computational chemistry
  - More Shodor examples
  - Engineering
  - Economics
  - And so on

Systems Model Tools

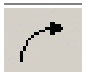
- There are several systems modeling packages that can provide similar learning experiences
- iThink; Berkeley Madonna; Stella
- Vensim
  - Free education version
  - Graphical user interface to modeling
- Open bunnycomparison.mdl

Some Sketch Tools

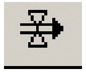


Auxiliary Variable (constant)

Box Variable (Level)



Arrow (connects cause and effect)



Rate

More Vensim Examples

- Other examples
  - Advanced SIR
  - Pharma model
- Can save runs under different names
- Compare runs on the same graph
- Interactively change parameters to find a target

Explore Other Built Models

- http://www.shodor.org/talks-new/vensim/

Not So Secret Agent

- What is an agent?
  - An autonomous entity that acts according to a set of rules or constraints
  - Multiple agents are involved in complex systems, each acting in a particular way
  - Agents that “meet” then interact to produce another set of outcomes
  - The resulting outcomes are often different than one would expect due to the complexity of the interactions
  - Most agent-based models introduce the idea of randomness in the interaction rules – i.e. Monte Carlo simulations

Some Modeling Conveniences

- Agents can act both in space and in time
  - Explicit spatial movement is often important to accurately represent some phenomena
  - More difficult to do with other approaches
  - More realistic representations of spatial phenomena are possible

Some Examples

- Spatially explicit models of the spread of disease
- Growth of urban areas
- Supply chain optimization
- Human cell and immune system models
- Biochemical processes
- Consumer behavior and economics models

Tools

- AgentSheets
- Netlogo
- StarLogo
- Repast (with a supercomputer version)
- Swarm

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?

Let’s Demonstrate

- Open AgentSheets
  - File Open Project
  - Work your way to the Datasets Folder
  - Open SimpleSick
- Opens several windows
  - Open Worksheet – graphical representation of the space being modeled and the agents in that space
  - Simulation properties – default settings for important variables
  - Gallery – Agents used in this simulation

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

Second Example

- Open Access fire (in datasets folder)
- Note there are trees with three states
- Story behind this model

Running the Model

- Click on the finger in the pallet on the Worksheet window then a tree and Run
  - Observe the behavior – what is the result?
  - Clck Stop then Reset
  - Now change the burnprob on Simulation Properties to 5.0
  - How do the results compare?

Agents and Behavior

- Agents can have several states
  - E.G. Tree – green, on fire, burnt over
  - Each state has an editable depiction
- Right click on the tree or click the tree and use the Gallery – Edit Behavior
  - Simple graphical programming environment
  - Click on red border for top component then Explain
  - Notice the other changes of state which produce a change in the agent Tree

Changing the Model

- Let's suppose that the forest we are working in is most prone to major fires when there are strong winds from a particular direction
  - E.G. California Santa Anna winds
- How would we change the agent behavior to reflect this idea?

Modifying an Existing Model

- Open Access Fire
  - Same as current model
  - Let's add some directional information by changing the behavior of the tree
  - Right click on the tree – Edit Behavior
  - Change the reaction of the tree to only seeing trees from the West.
  - Change the burnprob name to burnprobw
  - Edit the simulation properties to change burnpro to burnprobw and set an initial value
  - Try it out and check the logic

Moving to Supercomputer Scale

- Once students understand the basic principles of a particular type of model – can scale models to run on supercomputers
- Use community codes or science gateways
- Approach the problems in a similar way
  - Make multiple runs to understand system behavior
  - Pose problems that change a finite set of model parameters to answer specific questions

Other Education and Training Resources

- hpcuniversity.org
- Online training
- XSEDE collaborative online courses
- Journal of Computational Science Education
- ACM SIGHPC Chapter

Questions and Discussion

Reference Materials Reminder

- https://www.osc.edu/~sgordon
  - –Choose Workshop Materials
  - –Then Links to other materials

Session evaluation

- http://bit.ly/xsedesuno


Our reach will forever
exceed our grasp, but,
in stretching our horizon,
we forever improve our world.
XSEDE
Extreme Science and Engineering
Discovery Environment