Lecture 17

Modeling Reality

The demands of research and operational modeling
Agenda

- Contrasting and Common Issues in Real World Modeling
  - With emphasis on the ones we will explore shortly

- Linux Spinup
Working Environment

- **Operational**
  - Focuses on Real Time Product Generation for Decision Makers
  - Example: NCEP, ECMWF, BOR
  - Speed and “Working Accuracy”
    - Can you use it and is it available fast enough

- **Research Mode**
  - Focuses on processes, and not as heavy on time-dependant products
  - Example: NCAR
  - Accuracy, Understanding and Development
    - You can be fancy, just don’t expect it to be done quickly
Time-Dependance

- **Prognostic**
  - Generating a decision making forecast based on initial conditions and estimated boundary conditions (forward in time)

- **Diagnostic**
  - Based on near-past or real-time information (including, to some degree, assimilated data)

- **Retrospective or Climatology-Based Models**
  - Past information only (the model may not be able to handle forward in time data with any reliability)
    - WEPP and the Mono Lake Models are good examples: you can’t rely on future precip fxs by the day or month for these applications but you COULD use climate data to force them.
Model Construction

☐ Empirical
  ■ Based on statistical relationships
    □ CENTURY has a lot of this
    □ Parameterizations often rely heavily on Rugged Empiricisms

☐ Physical
  ■ Often Based on the proverbial “first principles” or navigable trains of events
    □ Here you are more likely to be able to trace causality through the steps
Model Transparency

- Proprietary and Commercial Models
  - Owned by a single organization and closed to development

- Semi-Black Boxes
  - Most releases that include only DLLs or GUI executables DO have support code that *may* be available for advanced users

- Community Models
  - Open Source (for better or worse)
Model Goals

- Forecast Reality
  - Creating working and functionally credible forecasts of the real world that can be validated and even used to initialize future versions of reality
  - Examples
    - NWP Models
  - Requires accurate model dynamics and physics (and empiricism must be robust enough to handle most expected events)
Model Goals

- Impose Reality
  - Taking idealized or generalized fields and causing them to conform to localized or contextualized conditions
  - Example: Low Level Met Models, Downscaling Models
Model Goals

- Approximating Behavior and Trends
  - Captures General Behavior and Trends but may not be able to provide a reasonable forecast at any moment in time or any given spatial point.
  - Application: Climate Modeling (which often has large-scale or time-dependant averages as its output product).
  - Other Applications: Lots of Systems Modeling Approaches.
Modeling Frameworks

Plumbing 101
Model Framework

- Operating Environment
  - Windows and X-Windows
    - Many have GUIs to make pointing and clicking your way into modeling paradise...
    - ... or hell if you have repetitive runs to make
    - Often accesses DLLs or Black Box Code
    - Low Maintenance but often not as flexible
  - UNIX/LINUX/DOS (Command Line)
    - Executables and Source Code often available*
      - You may be expected to build your own executables
        - That’s very common with LINUX/UNIX envs.
    - Scripts, Shells, and *BAT files can make repetitive tasks easy on your repetitive injury insurance.
    - User interfaces less than pretty to being limited to support programs and editing tools
Model Framework

- User Interface (to start or to run completely)
  - Where the you talk to the model (and the model talks back to you)

- Initialization
  - Where the model gets its data (may or may not be part of the model-proper)

- I/O Interface
  - Where the model (initialization and solver) talks to its data (input and output)

- Solver
  - Where the ODEs (or prognostic eqs) are solved

- Post Processing
  - Turning the model output into something recognizable (often redundant for GUI based tools)
Model Framework

- User Interface (Input)
  - Where you talk to the model (and the model talks back to you)
    - Can be a GUI
      - Nice because they are often easily navigated
        - WEPP for Windows
      - Can encapsulate both starting & running the model and looking at the model output
    - Can be a script
      - Nice because it can be automated
        - WRF and MM5 (having one “namelist file” containing most settings needed to describe what you want to do)
    - Can be a suite of operating files
      - Nice because they can be modularized
        - CENTURY and MODFLOW (e.g., have a weather file, a management file, an environment file, and one file to bind them all when you type “century.exe” at the prompt)
Model Framework

- Initialization

  - Where the model gets its data (may or may not be part of the model-proper)
  - Providing initial conditions and other data fields (as opposed to marching orders from the user)
  - Some models require intensive pre-processing to obtain this data...
    - Most NWP models.
  - ...while others really on ingesting pre-prepared or “community” datasets in a more transparent fashion
    - BASINS
Model Framework

- I/O Interface
  - Where the model (initialization and solver) talks to its data (input and output)
  - Separated by the user interface (where the model reads in exogenous data as it runs)
  - Where the model outputs results
    - Can be in ASCII or Binary or SDF
    - May output static graphics on-the-fly for real-time or post-run viewing
Model Framework

 Solver

- Where the ODEs (or prognostic eqs) are solved
- This covers most endogenous modeling activities
- Just sit back and pray that things work
- Sometimes there is logfile output that you can view in real-time or on-screen “entertainment” like a percent-completion bar.
Model Framework

- Post Processing
  - Turning the model output into something recognizable (often redundant for GUI based tools)
  - GUIs often have something for this already
    - WEPP
  - Some Command Line Models have Graphical Utilities (especially NWP models)
  - Others rely on post processing software
    - IDL, GRADS, Excel, IDV
Model Framework

☐ Not included: Product Generation
   - Creating products for users
   - This is not part of the local modeling framework and often is provided after the fact using external software and resources

☐ Examples
   - Web Portals and online products
   - Fax distribution etc
Rest of the Day

An introduction to Linux at the command line

SDSU may want to go home early
The Classic Hello Program

hello.f

program hello

C     Russ Meyer/Sanrio Version of the
C     Classic Hello Program

print*, "hello kitty! Kill! Kill!"

print*, 4e0 * atan(1e0)

print*, 4d0 * atan(1d0)

end

(you need to start typing code between spaces 7-72)
| □ 1-5 Line Numbers | □ 6 Continuation Marker | □ 7-72 Code Itself | □ 72-80 Identification Reference Space |

26 March 2007
Something more worthwhile
pi_overkill.f (non-cluster)

☐ cp ~wjc/pi_overkill.f ./

☐ if you want pi, all you really need is
  • pi = 4d0*atan(1d0)
  • And some languages have their own environmental variable for pi, epsilon and other parameters.