

# Virtual Telemetry for DDDAS

**Craig C. Douglas**

**University of Kentucky and Yale University**

**[douglas-craig@cs.yale.edu](mailto:douglas-craig@cs.yale.edu)**

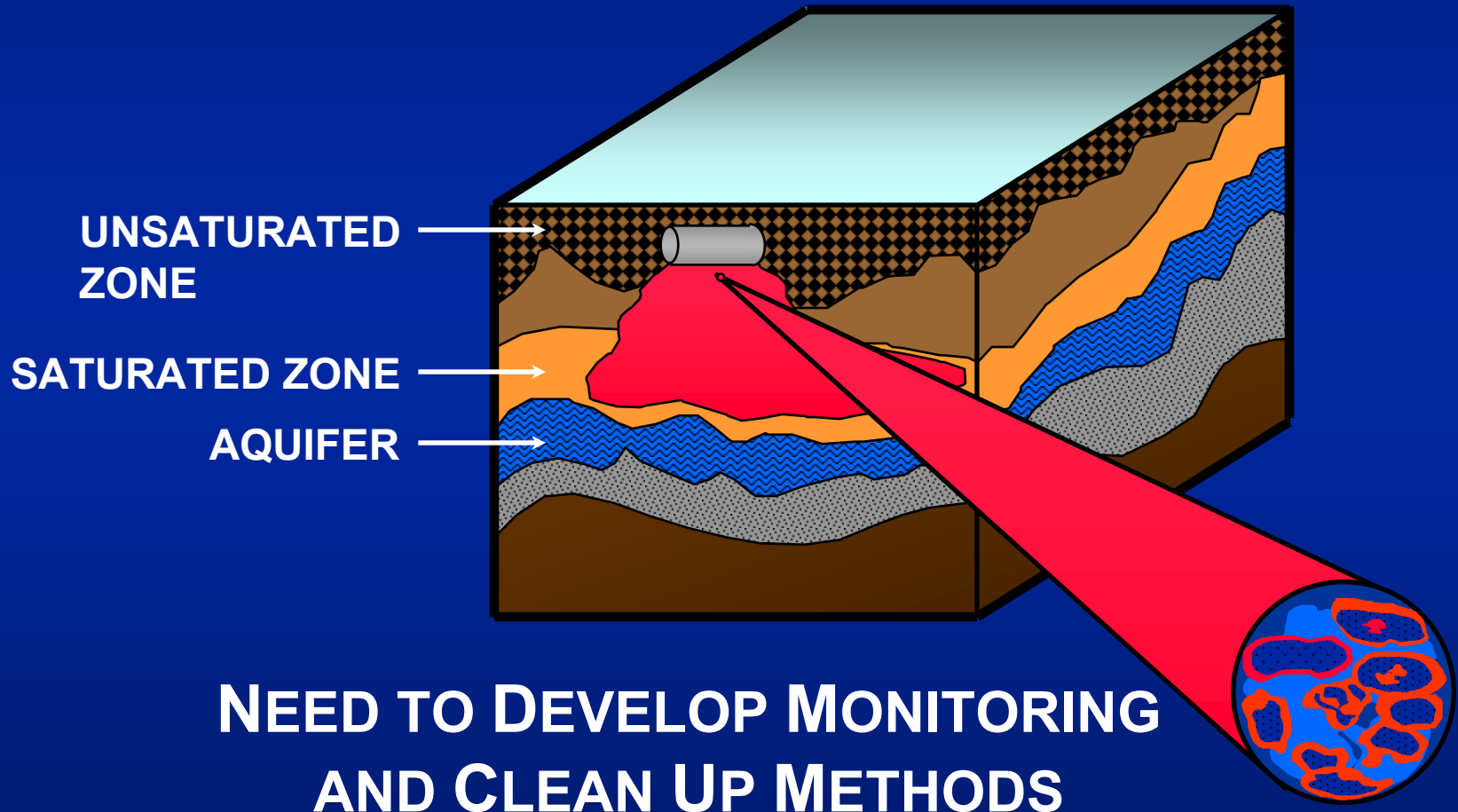
**<http://www.dddas.org>**

**Martin Cole, Yalchin Efendiev, Richard Ewing, Victor Ginting, Chris Johnson, Greg Jones, Ratcho Lazarov, Chad Shannon, and Jenny Simpson**

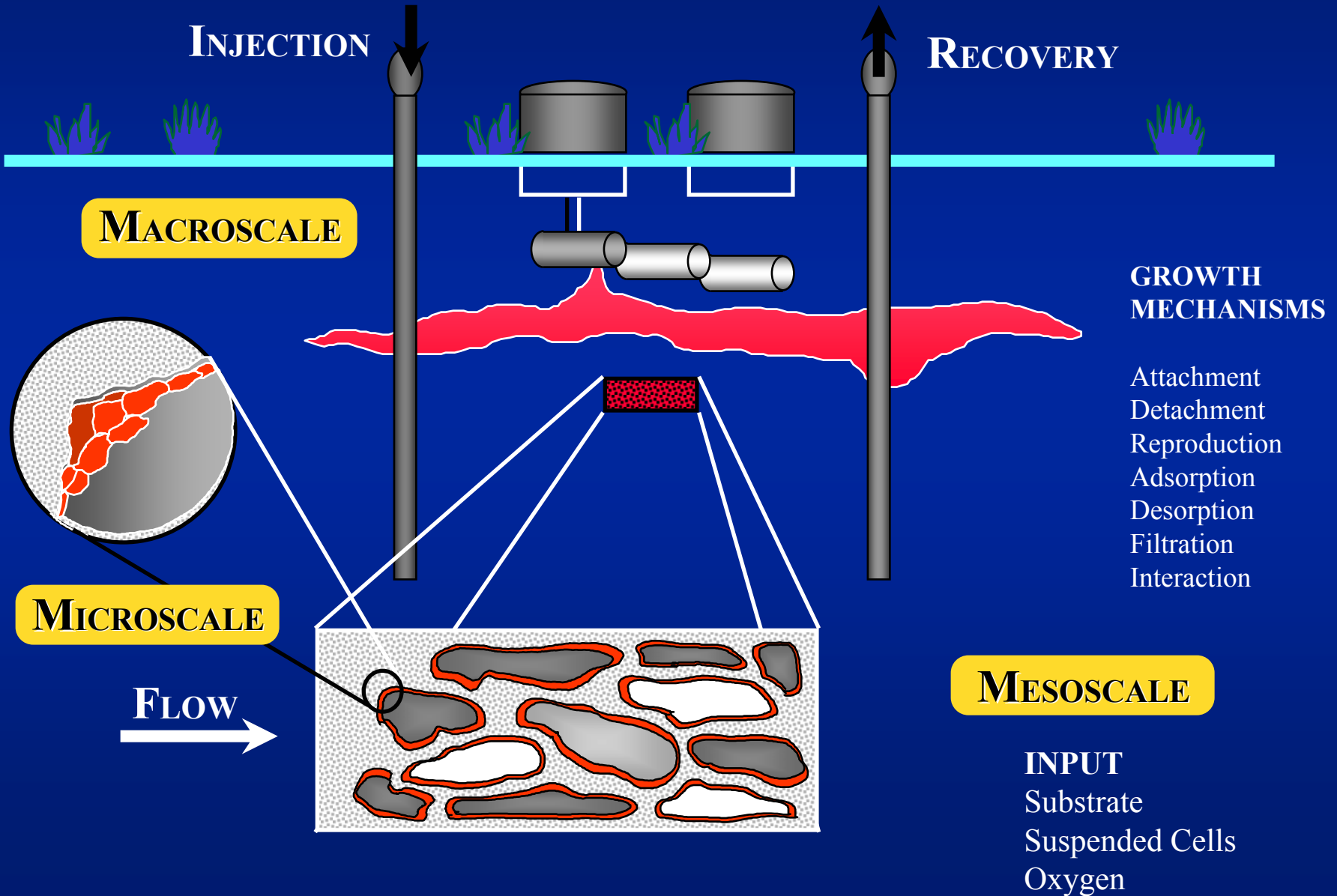
**Joint project of the University of Kentucky,  
Texas A&M, and the University of Utah**

***Supported in part by the National Science Foundation (ITR-DDDAS)***

# Leaky Underground Storage Tanks



# Bioremediation Strategies



# Forward versus Backwards in Time

## ◆ Forward

- Standard pollutant tracking problem in some sense
- Start by knowing who is the polluter and where it began

## ◆ Backward

- Might not know who the polluter is or where it began
- Determine it correctly
  - ◆ Report it so remediation can be started
  - ◆ Get sued
    - ◆ Spend 5-10 years dealing with lawyers
- Virtual telemetry solves legal problems until software used in the field by others who are used to lawsuits

# Data to Drive Application

- ◆ Where is the contaminant?
  - Use remote sensing data to locate media, update positions, and find new spread directions.
    - ◆ Open water bodies
      - ◆ Buoys and/or submerged stations
        - ◆ Chemical sensors
        - ◆ Visible, near IR, and IR scanning
        - ◆ Radiation detection
    - ◆ Underground
      - ◆ Well sensors

## Data to Drive Application (cont.)

- ◆ What is the terrain like in that area? What small-scale features are there?
  - New topography sets give world topography at 30 arcsec (~ 1 km), US at 3 arcsec (~100 m).
  - Better local sources might be available.
- ◆ What are the changing weather conditions?
  - Large-scale data (current analyses or forecasts) used for initial conditions and for updating boundary conditions.

## How a DDDAS Might Work (Research Mode)

- ◆ Use simulations: first use all available data for past (and eventually current) experimental contaminant tracking to direct collection at crucial times and places.
- ◆ Attempt to prove that the prediction of major contaminant behavior can be far more effective than the traditional method of tracking and intuition.

## How a DDDAS Might Work (Operational Mode)

- ◆ Human or a sensor (possibly on a buoy or in a well) determines a contaminant has been located near position X.
- ◆ Need to determine severity and possible expansion.
- ◆ Produce a prediction and post it on a public, known web site.
  - While running model at large-scale over a region...
  - Use latest data to locate contaminant boundary.
- ◆ Determine communication methods for remediation workers.



## How a DDDAS Might Work (Operational Mode; cont.)

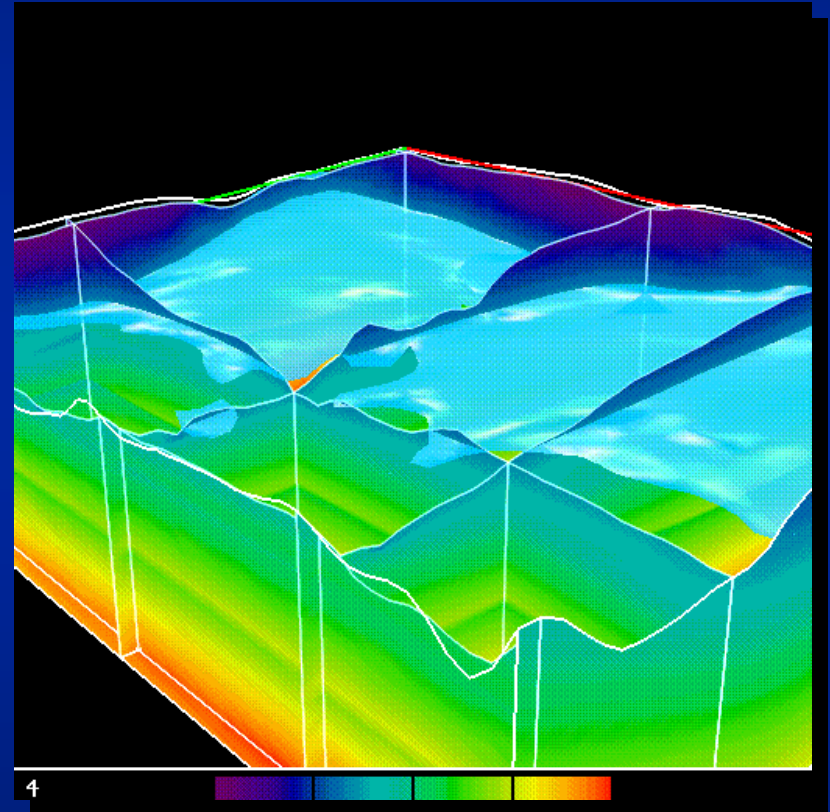
### ◆ Have application

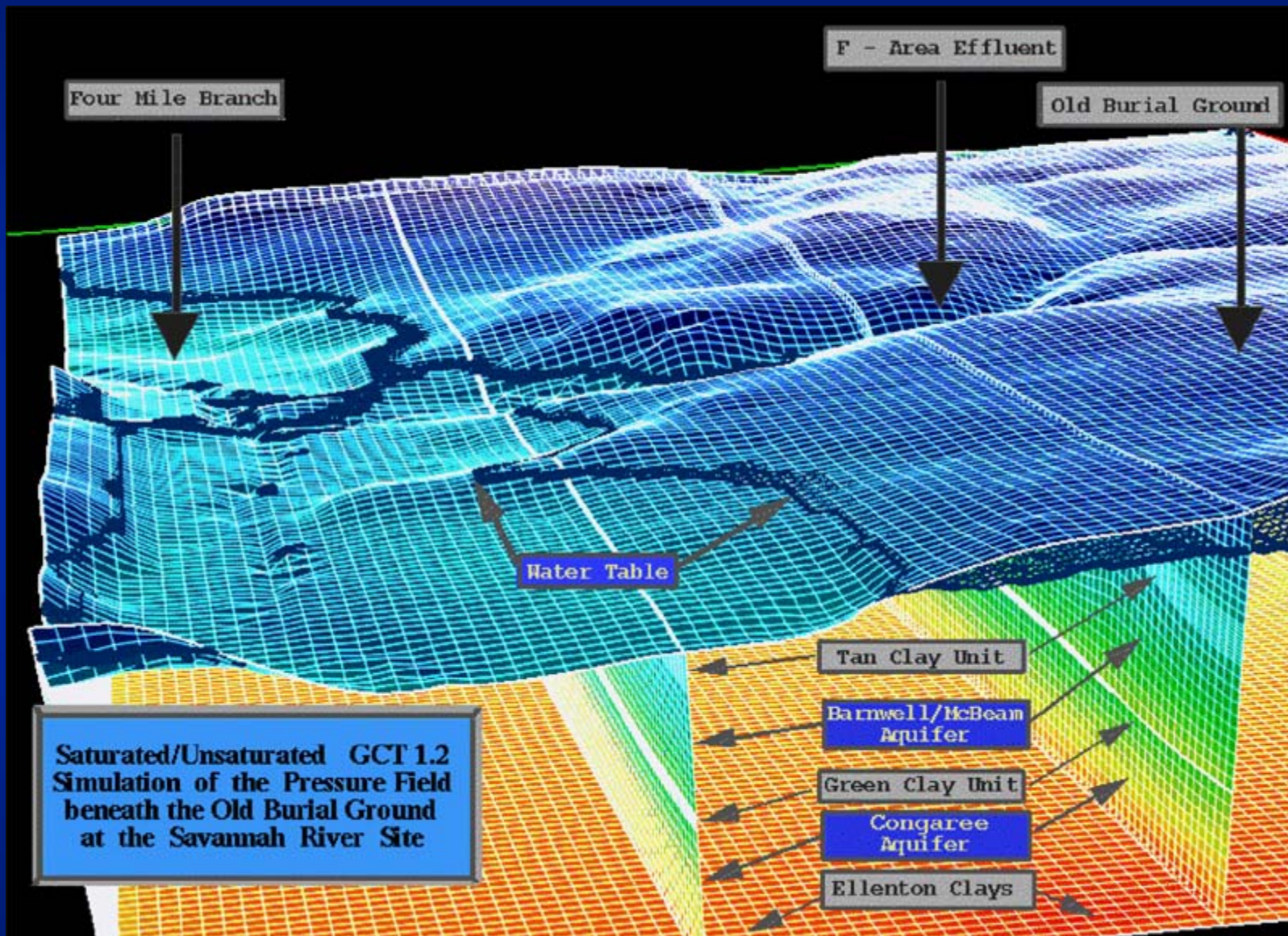
- Seek out contaminant classification data
- Collect recent large-scale data (analyses and forecast) for contaminant model initial and boundary conditions.
- Initialize and spawn smaller-scale domains, telescoping down to the contaminant area.
- Ignite contamination in the model at observed location.
- Simulate the next Y hours/months/years of spread behavior.
- Dispatch forecast to Web site.

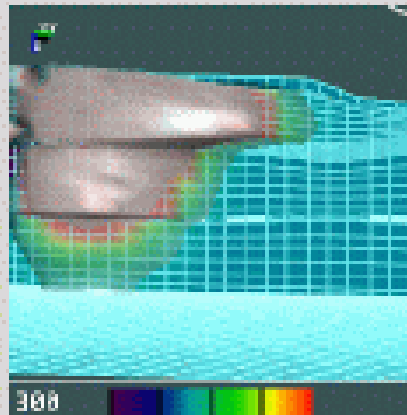
### ◆ Easier and cheaper to write a new version of the application using modern programming environments and liberal code re-use.

# Savannah River Site

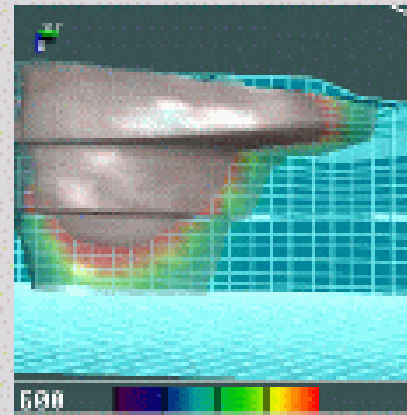
- ◆ Difficult topography
- ◆ Highly Heterogeneous Soils
- ◆ Saturated and Unsaturated Flows
- ◆ Reactions with disparate time scale
- ◆ Transient/Mixed Boundary Conditions



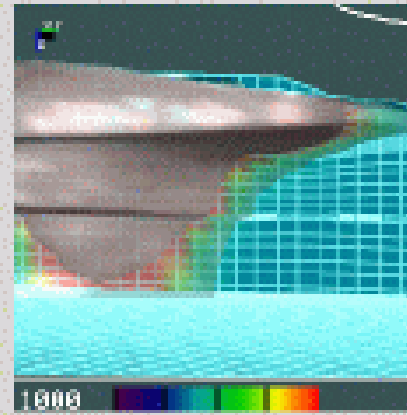
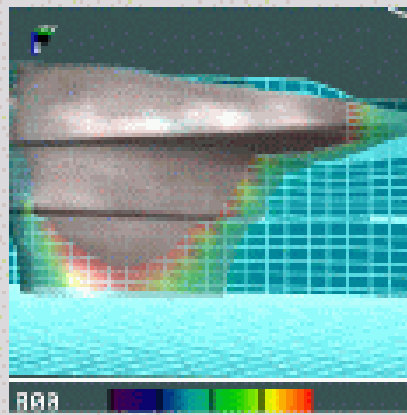




(a)



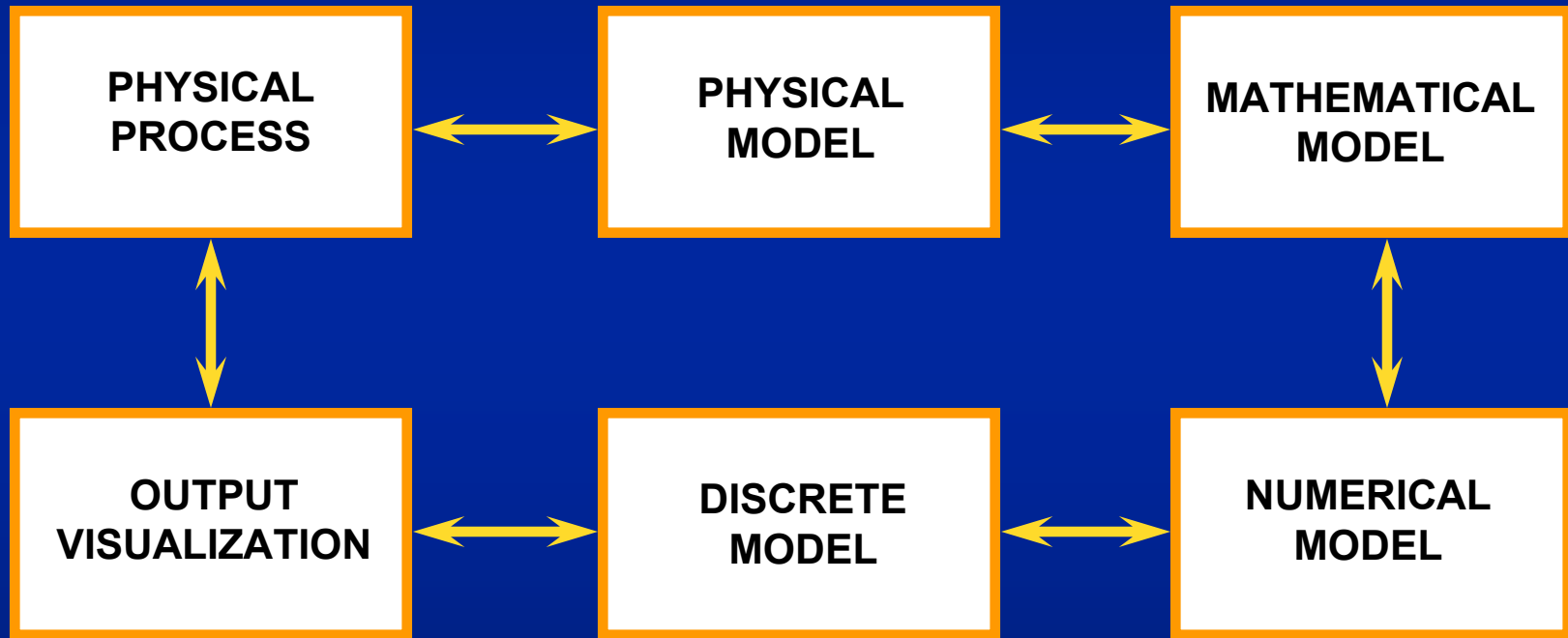
(b)



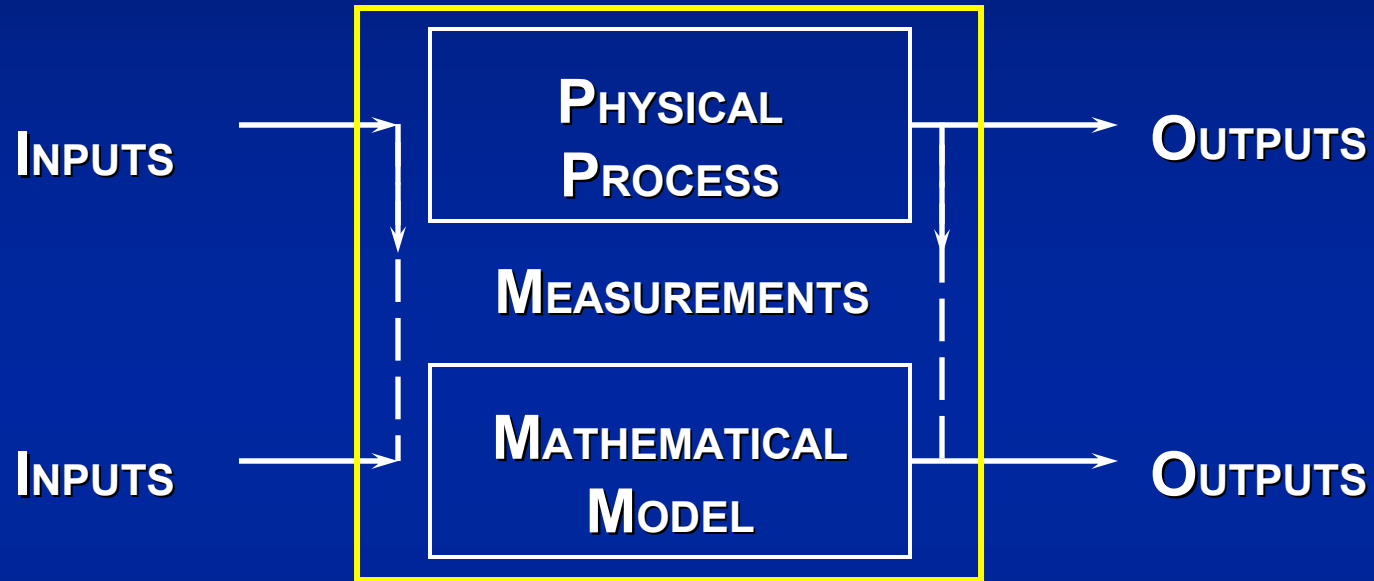
# Need for Simulation

- ◆ Develop better understanding of nonlinear behavior
  - Computational laboratory  $\longleftrightarrow$  Experiments
  - Understand sensitivities of parameters
  - Isolate phenomena then combine
- ◆ Scale-Up information and understanding
  - Microscale  $\longrightarrow$  Laboratory  $\longrightarrow$  Field
- ◆ Obtain bounding calculations
- ◆ Develop predictive capabilities
  - Optimization and control

# Modeling Process



# Identification (Inverse) Problem



- ◆ Determine suitable mathematical model
- ◆ Estimate parameters within mathematical model

# Large Scale Interactive Applications on Remote Supercomputers

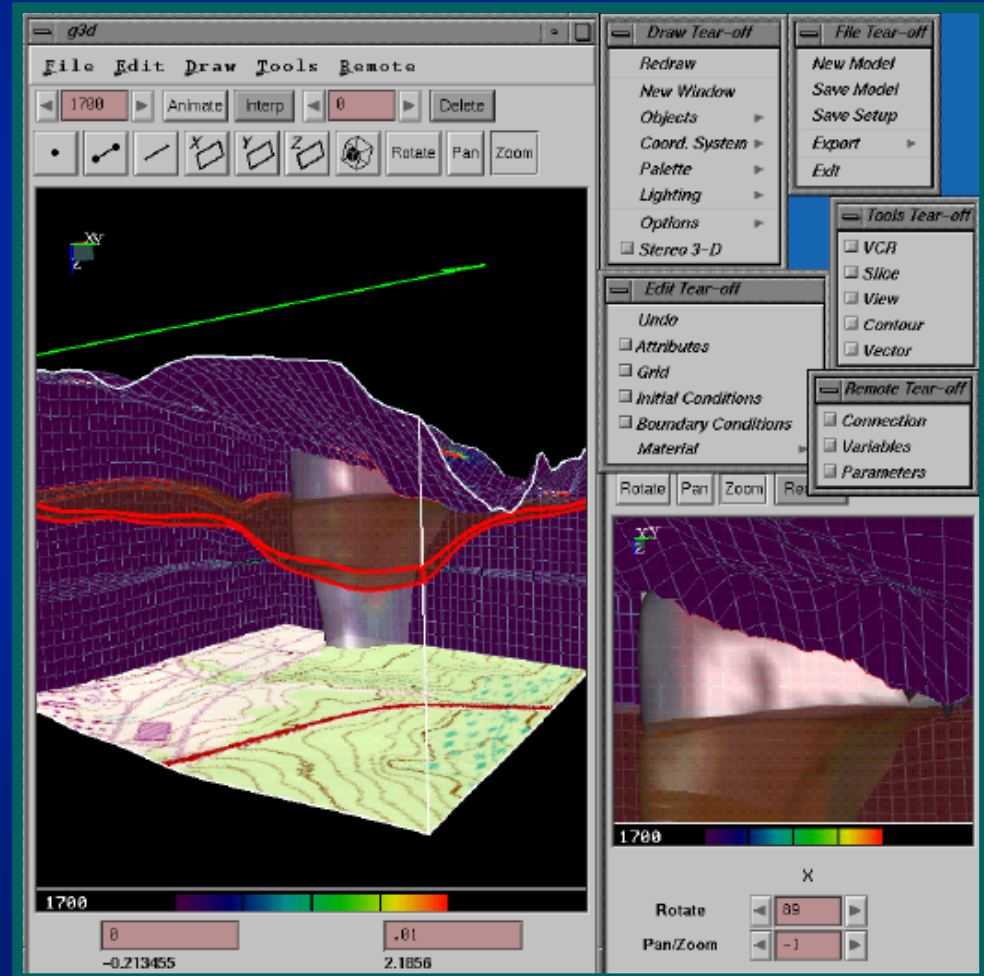
- ◆ Model Development and Formulation
- ◆ Coupled Codes with Complex Boundary Conditions
- ◆ Numerical Discretization and Parallel Algorithm Development
- ◆ MPP Code Development
- ◆ Field Testing and Production Runs
- ◆ User Environments and Visualization Tools

**Need for *interactive tracking and steering* and possibly  
elimination of *human in the loop***



# Graphics Pre-Processing

- ◆ 3D grid creation and editing
- ◆ Material properties
- ◆ Initial conditions
- ◆ Time dependent boundary conditions
- ◆ Multiple views



# Graphics Post-Processing

- ◆ Multiple vector/scalar fields
- ◆ Time animation
- ◆ Multiple slices/Iso-surfaces
- ◆ Stereo rendering, lighting models
- ◆ Overlay images for orientation
- ◆ Volume rendering

Hierarchical Representations

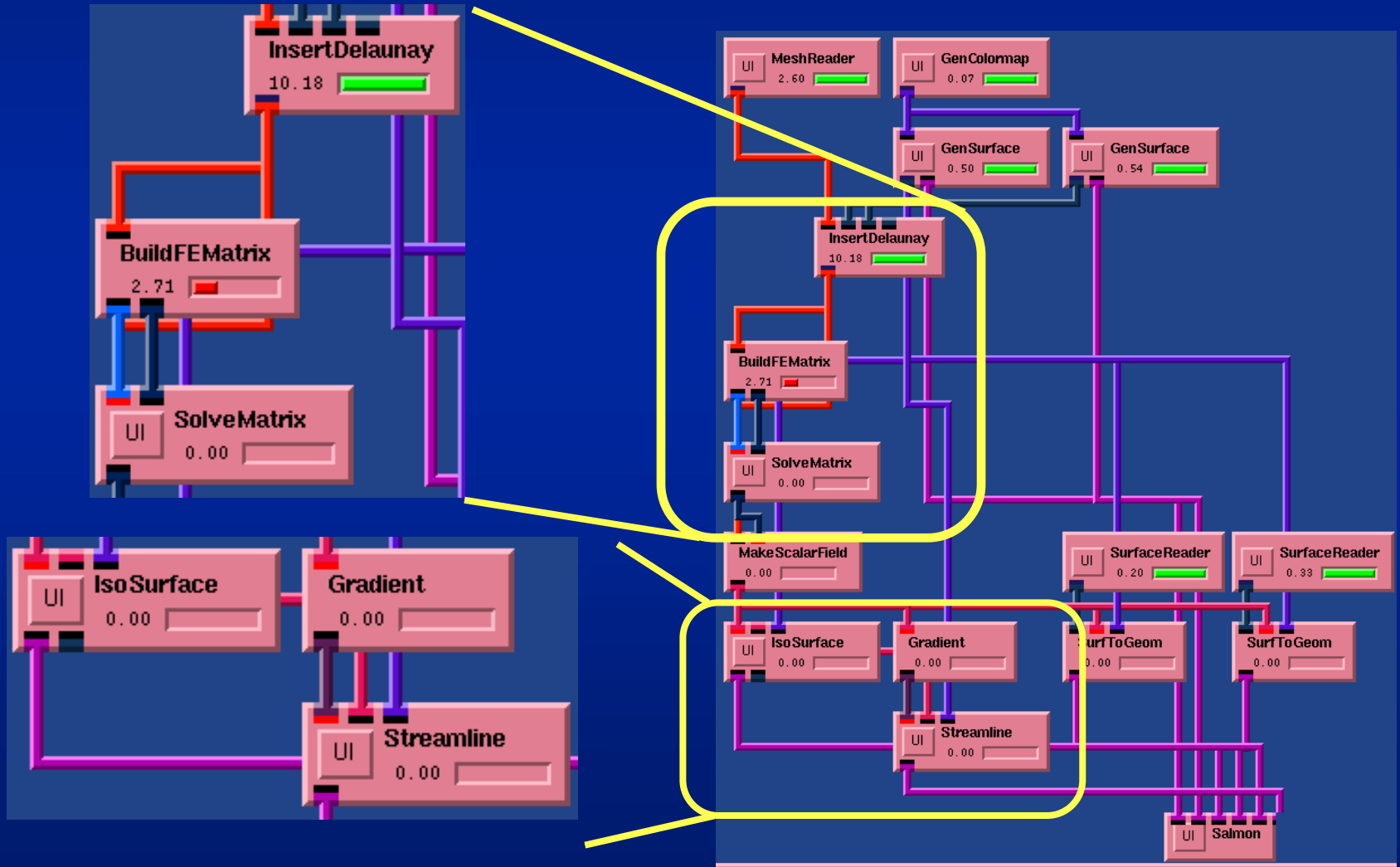
# Issues of Perturbations from On-Line Data Inputs

- ◆ Solve:  $F(x + \Delta x(t)) = 0 \leftrightarrow$  Choice of new approximation for  $x$ 
  - Do not need a precise solve of equation at each step
    - ◆ Incomplete solves of a sequence of related models
    - ◆ Effects of perturbations (either data or model)
    - ◆ Convergence questions?
  - Premium on quick approximate direction choices
    - ◆ Lower-rank updates
    - ◆ Continuation methods
  - Interchanges between algorithms and simulations
  - *Local* boundaries and conditions not known *a priori*
- ◆ Fault-tolerant algorithms

# Many Different Components that We Want to Vary *Quickly* during Project

- ◆ We need graphics, solvers, and specialized codes
  - Matlab inappropriate though it has all of the features we need
    - ◆ Expensive (becoming more so as a result of antitrust suit) and keeps going dense instead of sparse
    - ◆ Matlab compiler does not solve speed issues completely
  - SCIRun has great potential and will be used
    - ◆ Extensible since owners of code are project members
    - ◆ Easy to construct complicated codes using cut and paste plus connect the boxes techniques

# SCIRun Methodology



# SCIRun and Telemetry

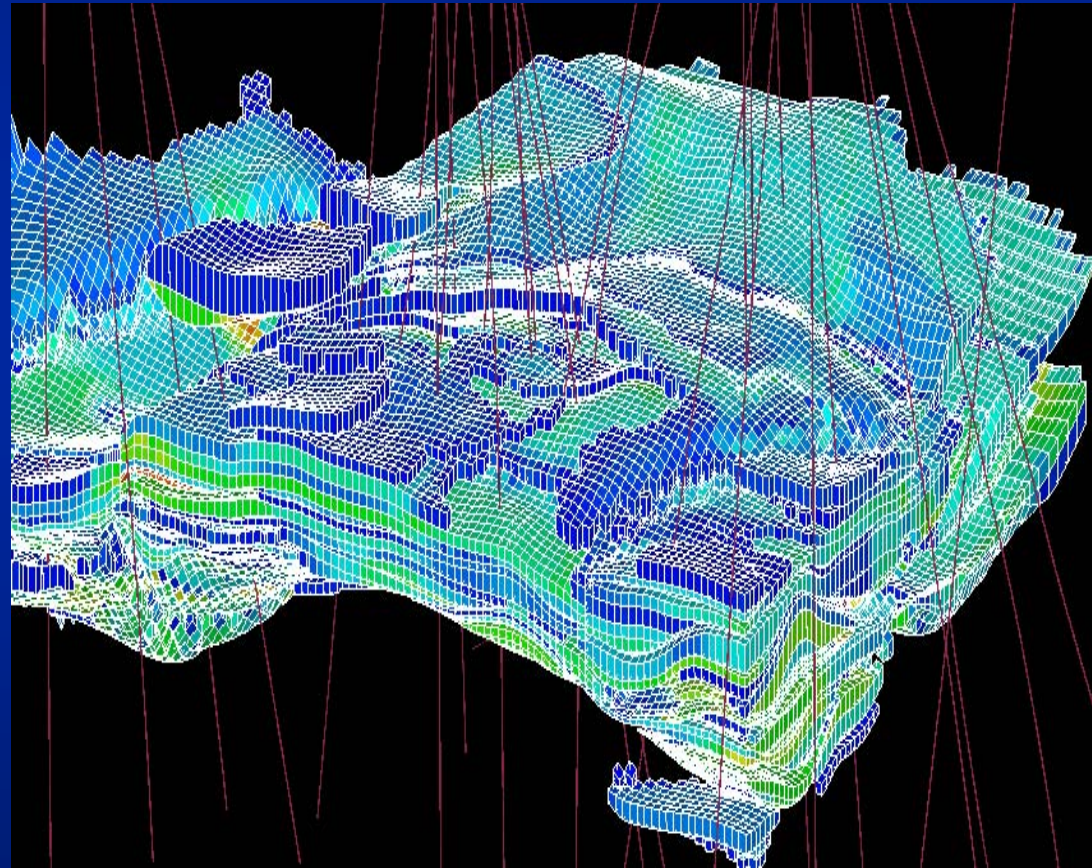
- ◆ **Streaming paradigm very useful**
  - Add/delete remote clients trivially
  - Well understood for audio and somewhat less for video
- ◆ **Streaming implementations problematic for DDDAS**
  - Standard formats lossy (filter out frequencies that cannot be heard or seen)
  - Implementations of lossless formats (e.g., Ogg Vorbis) are lossy, too ☹

# SCIRun and Telemetry

- ◆ **Adding streaming modules**
  - Line points to a stream, either incoming or outgoing, instead of another SCIRun module
  - Multicasting is an alternative, but it is banned by almost all Internet providers and may be outlawed shortly
- ◆ **Fill in data (interpolation as needed) as data comes in**
- ◆ **Timestamp data so that running backwards in time possible and reverting to an earlier time step (warm restart) is possible (locally or globally)**

# Why Interpolation Is Required

- ◆ Data available only at subset of mesh points
- ◆ Data not available on mesh at all





# Why Virtual Telemetry?

- ◆ We are doing contaminant tracking. Suppose you have real telemetry data to offer us...
  - Would you *really* admit you are a significant polluter?
  - Do you want the extent of your pollution *known*?
- ◆ Oil simulation experts in academia rarely have access to long term actual data from reservoirs. The data is usually sanitized and only for starting simulations from simple datasets.
- ◆ ***It is hard to beat the cost of virtual telemetry.***
- ◆ We can vary quantity and quality of data using multiple streams.

# Out of Country Experience

- ◆ Offered real telemetry from Rio de Janeiro.
- ◆ Cannot afford the phone bill to get the data on current grants.
- ◆ Internet delivery *not* an option.
- ◆ Personal pickup *is* an option.



# Virtual Telemetry: The Other Code

- ◆ **We have another code running in real time with small time steps for possibly duration of project**
  - **Known to be very accurate from past history matching.**
  - **Probably not close to state of the art in terms of runtime.**
  - **Computes everything each time step.**
  - **Is computing on a very different mesh than our DDDAS code is.**
  - **Code streams out data to whoever is interested (and authorized).**
  - **Data lands on the floor when no one is listening to it or can be saved.**

# Virtual Telemetry Meets Telemetry

- ◆ Design SCIRun modules for telemetry so that real telemetry data can be used in the future.
  - This is a data representation problem
  - Requires a hook to do interpolation correctly into computational mesh
    - ◆ Both virtual and real telemetry need hook
    - ◆ Requires looking at several applications to do right
      - ◆ Volunteers needed to tell us what telemetry data looks like for your DDDAS applications
        - ◆ Greg McRae, ... ?

# Conclusions

- ◆ Designing general purpose virtual telemetry system that will be useful by DDDAS community.
- ◆ Inexpensive to set up and use.
- ◆ Goals:
  - To make it transparent when switching between real and virtual telemetry.
  - Allow us to investigate easily DDDAS aspects that are not in static dataset formulations (e.g., how to analyze situation like not knowing boundary conditions or where the local boundary is *a priori*).
- ◆ Dissemination through <http://www.dddas.org>
  - Links to other projects there, too – more needed.