Integrating Computational Science into the Curriculum

Steven I. Gordon
sgordon@osc.edu

XSEDE

Extreme Science and Engineering Discovery Environment

NSF

The Need for a Modeling Savvy Workforce

- Documenting the Need
- How science and engineering (and social science and humanities) research is done
- What should our students know?
- Barriers and strategies for Implementing changes to the curriculum
- XSEDE and related resources and services

Crucial Tools for Manufacturing

- Toyota Avalon – no "mules."
- Each mule cost about $500K, and about 60 are needed to develop a new model.

http://detnews.com

Automakers trade 'mules' for computers

The advanced computer design brings cars to market faster and cheaper.

By Eric Mayne / The Detroit News

For decades, automotive engineers had one place to test the vehicles they designed -- where the rubber meets the road.

But software capability is advancing so rapidly that the time-honored practice of building prototype vehicles -- "mules" in auto industry jargon -- to test safety equipment, manufacturing tools, and aerodynamics is fading fast.

Super-powerful computers can now simulate vehicle appearance, functions and performance so well that engineers send few if any prototypes to the wind tunnel, proving grounds or crash test lab.

Max Ortiz / The Detroit News;Toyota

The Toyota Avalon built at the company's plant in Kentucky was constructed without using conventional prototypes.; Phil Martens

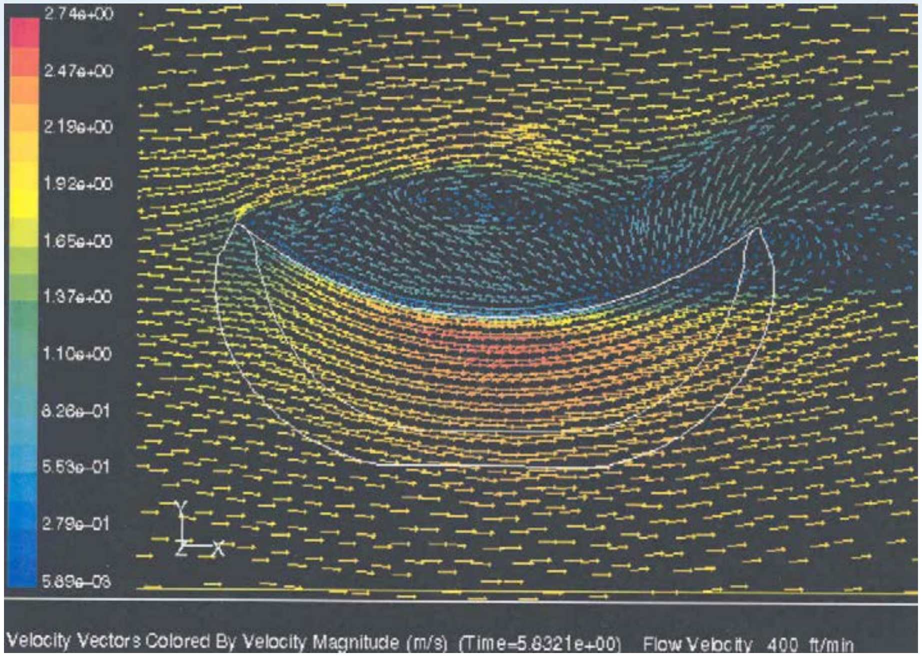
- Comment on this story
- Send this story to a friend
- Get Home Delivery

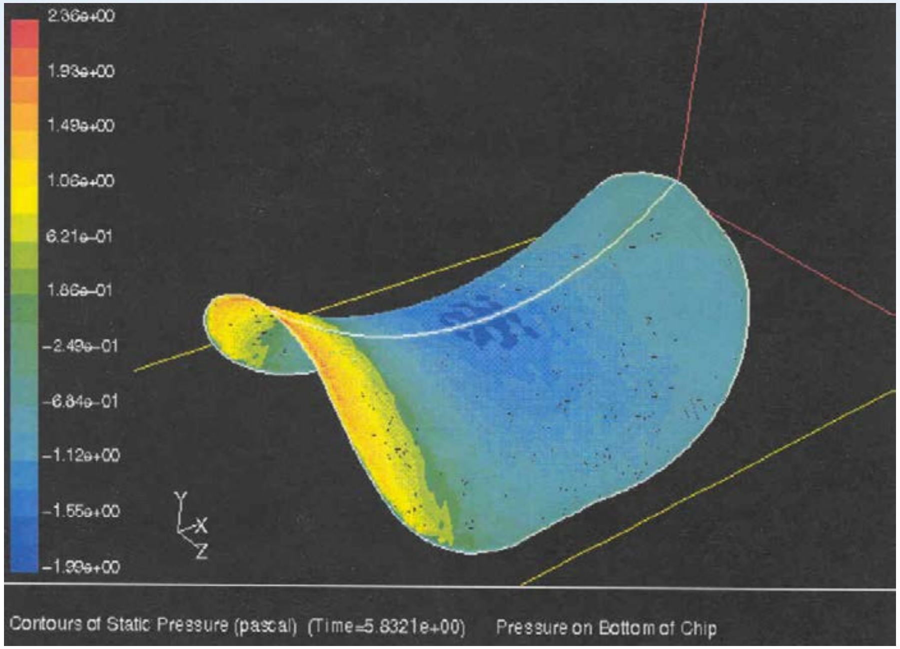
It's a sea change that has the potential to save millions and bring vehicles to the market faster in the highly competitive auto industry. A single prototype vehicle can cost up to $500,000 and as many as 60 prototypes have been used to develop new models.

"The cost savings to go digitally are enormous," said Phil Martens, Ford Motor Co.'s group vice president of North America product creation. "It reduces the complexity of having to worry about building the prototype in the first place."

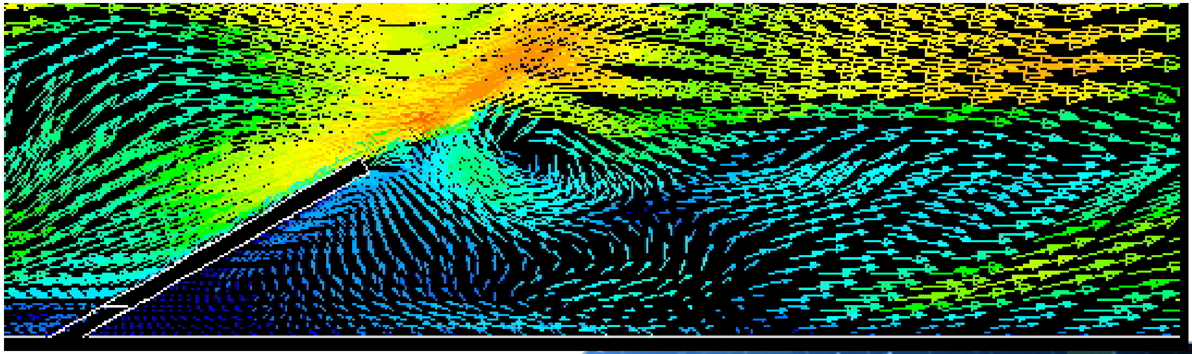
One in three vehicles on the market today is designed with minimal use of prototypes. And in five years, the number will be closer to two-thirds. And in designing the 2005 Toyota Avalon, the Japanese automaker said it used no prototypes at all.

Will Pringles Fly?



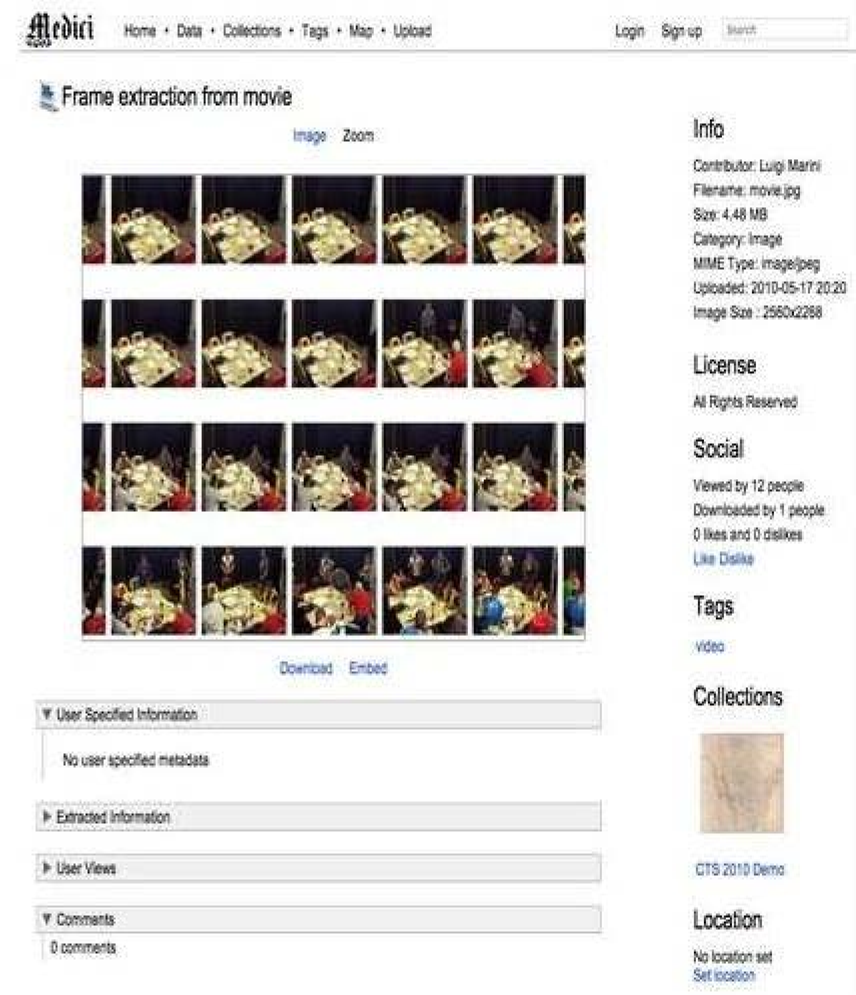


High Speed Conveying
Create Vortices
Shedding…
…'Rocking Chips'
NOT GOOD!



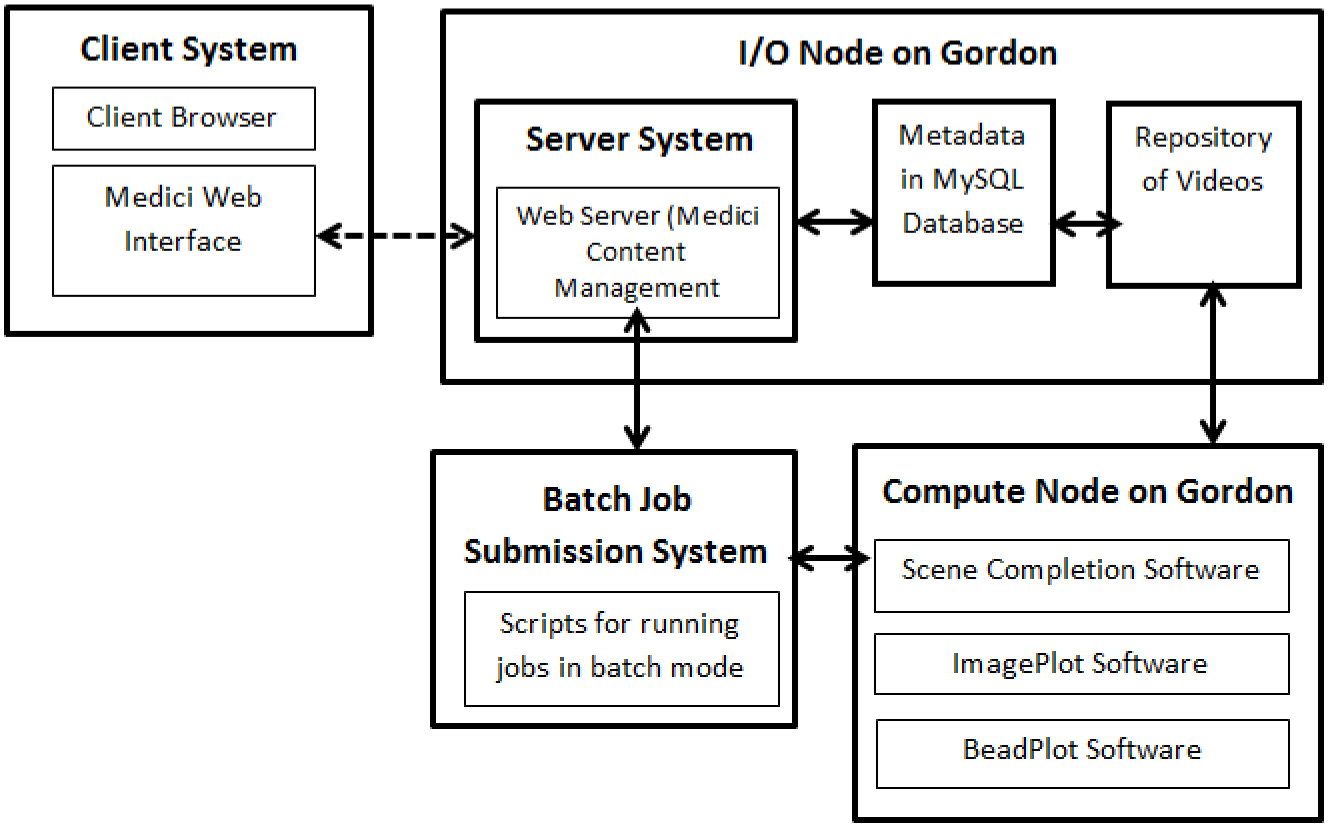
Large Scale Video Analytics

- Creation of human-machine hybrid process to analyze and archive video images
- Allow exploration of cultural trends through analysis of video archives

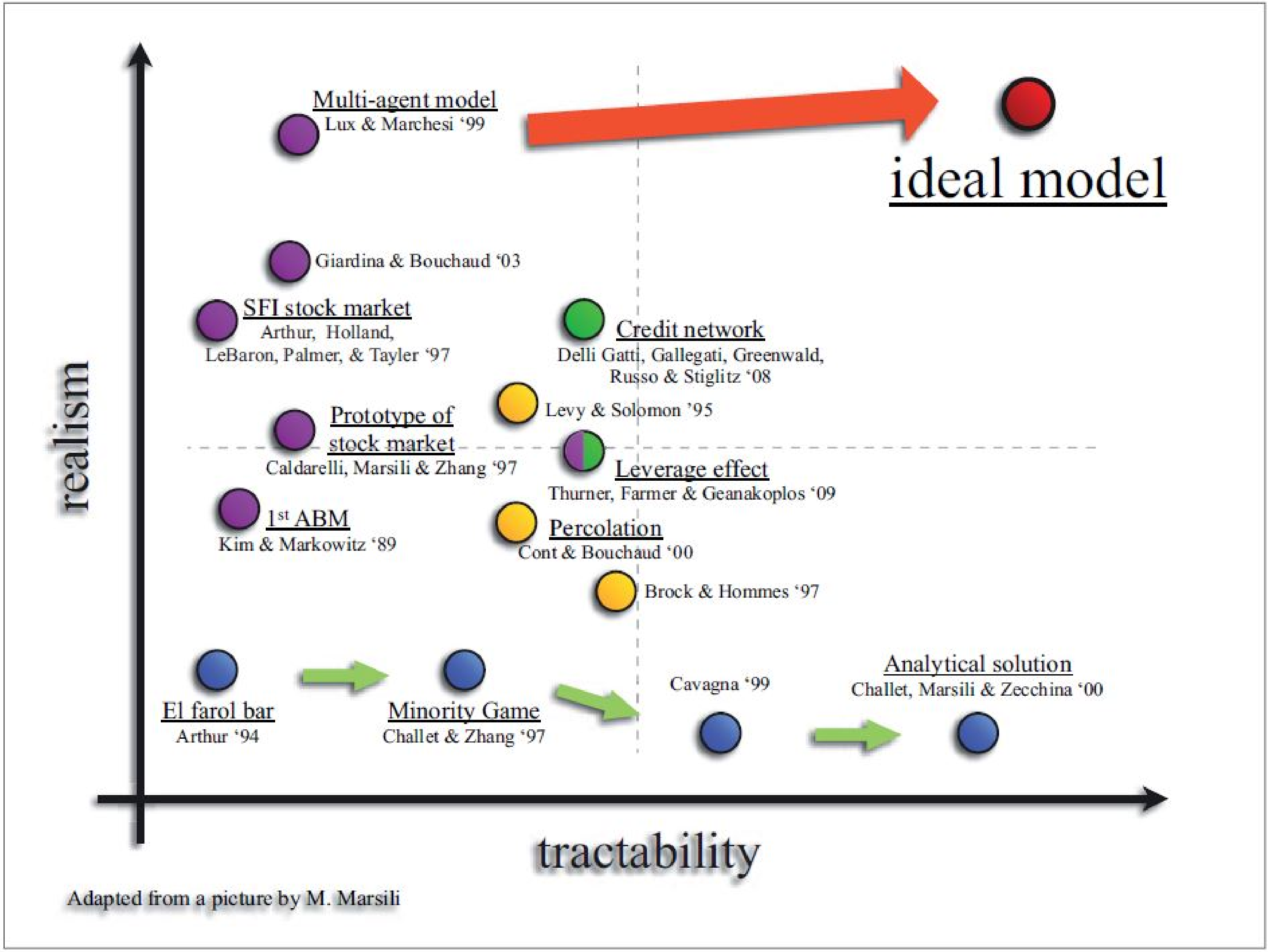


NSF XSEDE

Analysis Process



Source: http://www.academia.edu/2653762/Large_Scale_Video_Analytics_On-demand_iterative_inquiry_for_moving_image_research



Adapted from a picture by M. Marsili

M. Cristelli, L. Pietronero, and A. Zaccaria

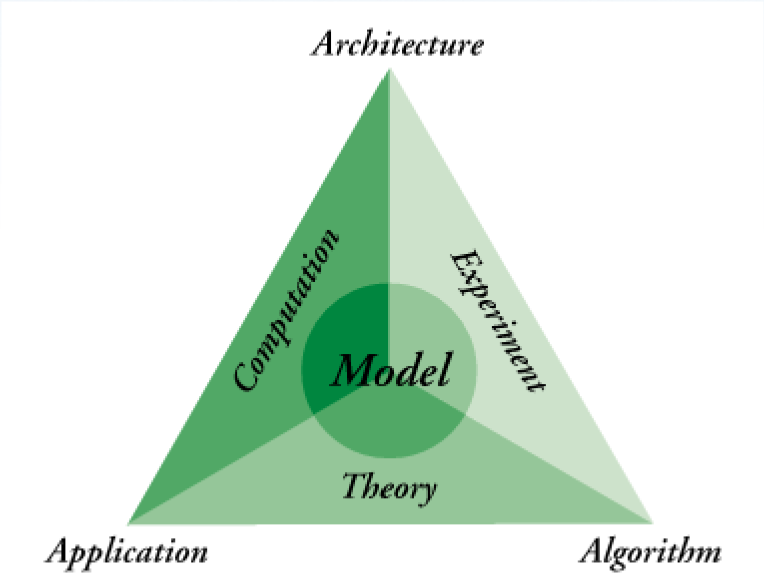
Making Progress in Science

- A number of studies document the need for computational scientists
  - …" computer modeling and simulation are the key elements for achieving progress in engineering and science." NSF Blue Ribbon Panel on Simulation-Based Engineering Science
  - "Unfortunately, the translation of systems biology into a broader approach is complicated by the innumeracy of many biologists" Cassman et al. Barriers to Progress in Systems Biology, Nature Vol. 438|22/29 December 2005
  - Nearly 100% of the respondents indicated that HPC tools are indispensable, stating that they would not exist as a viable business without them or that they simply could not compete effectively. IDC Study for Council on Competitiveness of Chief Technology Officers of 33 Major Industrial Firms

Computation is how science is done

Marketing Computational Science

Computation is Central to How Science is Done



- Computation lets us explore phenomena that are too big or complex to experiment, too small, or changes too fast or too slowly.
- Computation allows us to explore more options more quickly.

How we teach is just as important as what we teach

Seymour and Hewitt: Talking About Leaving

Students switching to non-science majors

- Over 90% indicated poor instruction among reasons for switching
- 26% had trouble learning the basic concepts

How Do We Go From the Abstract to the Applied?

F = −kx

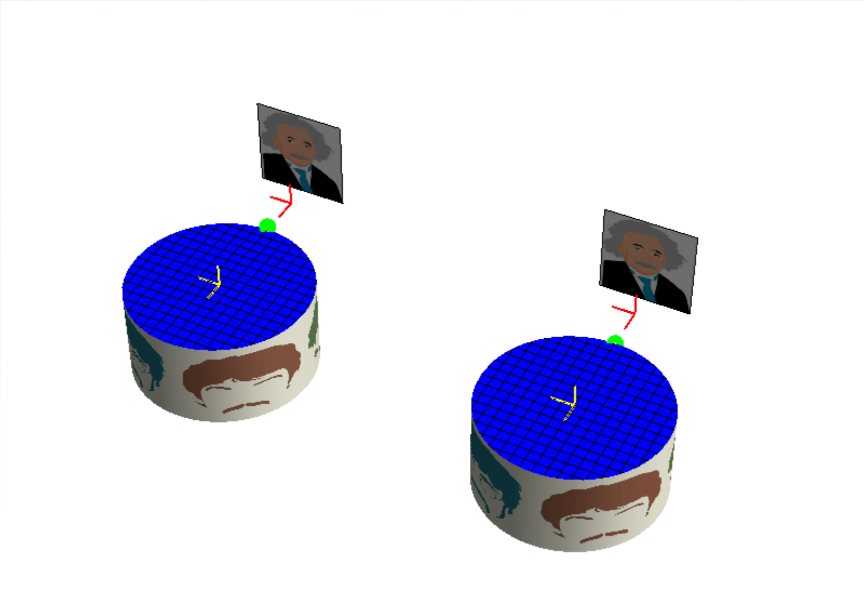
$$U = \frac{1}{2}kx^2$$

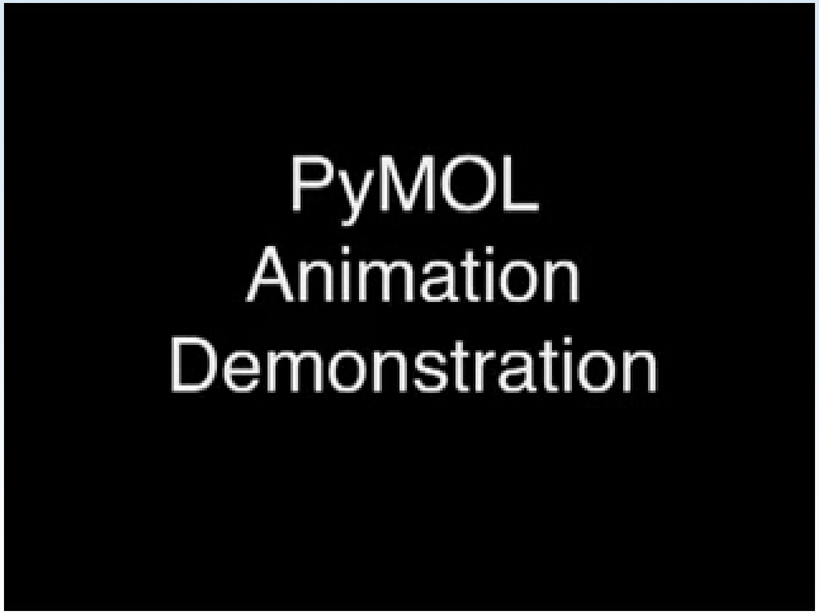
Wc **= −Δ**U

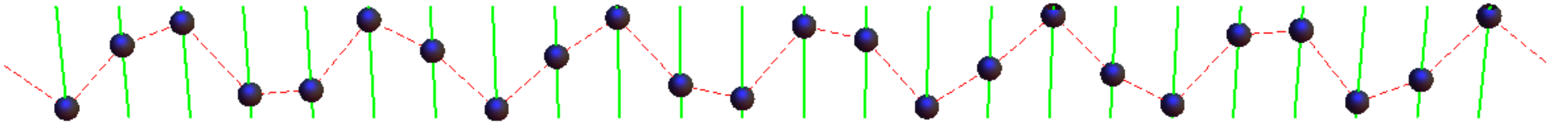
$$E = K + U$$

$$m\frac{dv}{dt} = F_{net}$$

$$x(t) = x_0 + v_0 t + \frac{1}{2}\left(\frac{F_c}{m}\right)t^2$$




PyMOL
Animation
Demonstration



Challenges to Changing How and What We Teach

- We tend to teach in the way we were taught
- Computational science is interdisciplinary
  - Faculty workloads fixed on disciplinary responsibilities
  - Coordination across departments is superficial
  - Expertise at universities is spotty
- Major time commitments are required to negotiate new programs and develop materials
- Curriculum requirements for related fields leave little room for new electives
- Change is hard

Strategies for Change

- Integrate computational examples into basic science and math courses
- Create general education courses that introduce simulation and modeling concepts and applications
- Combine those efforts to create formal concentrations, minors, or certificates in computational science
- XSEDE is working with institutions to assist with those activities

What Do Students Need to Know?

- Considerable discussion across many disciplines
  - Difficulty working from general conceptual ideas to specific skills and knowledge
  - Need to bridge disciplinary boundaries and terminology
- Using a competency based model to arrive at consensus of the essential knowledge base
- Competencies reviewed by both academic and non-academic experts
- See http://hpcuniversity.org/educators/competencies/

Ohio Minor Program Example

- Undergraduate minor program
  - 6-8 courses
  - Varies based on major
- Faculty defined competencies for all students
- Reviewed by business advisory committee
- Program started in Autumn 2007
- Agreements to share students at distance, instructional modules, revenues, and teaching responsibilities

Competencies for Undergraduate Minor
Simulation and Modeling
Programming and Algorithms
Differential Equations and Discrete Dynamical Systems
Numerical Methods
Optimization
Parallel Programming
Scientific Visualization
One discipline specific course
Capstone Research/Internship Experience
Discipline Oriented Courses

Example Competencies Simulation and Modeling

- Explain the role of modeling in science and engineering
- Analyze modeling and simulation in computational science
- Create a conceptual model
- Examine various mathematical representations of functions
- Analyze issues in accuracy and precision
- Understand discrete and difference-based computer models
- Demonstrate computational programming utilizing a higher level language or modeling tool (e.g. Maple, MATLAB, Mathematica, Python, other)
- Assess computational models
- Build event-based models
- Complete a team-based, real-world model project
- Demonstrate technical communication skills

Detailed Descriptors

Explain the role of modeling in science and engineering
Descriptors:
Discuss the importance of modeling to science and engineering
Discuss the history and need for modeling
Discuss the cost effectiveness of modeling
Discuss the time-effect of modeling (e.g. the ability to predict the weather)
Define the terms associated with modeling to science and engineering
List questions that would check/validate model results
Describe future trends and issues in science and engineering
Identify specific industry related examples of modeling in engineering (e.g., Battelle; P&G, material science, manufacturing, bioscience, etc.)
Discuss application across various industries (e.g., economics, health, etc.)

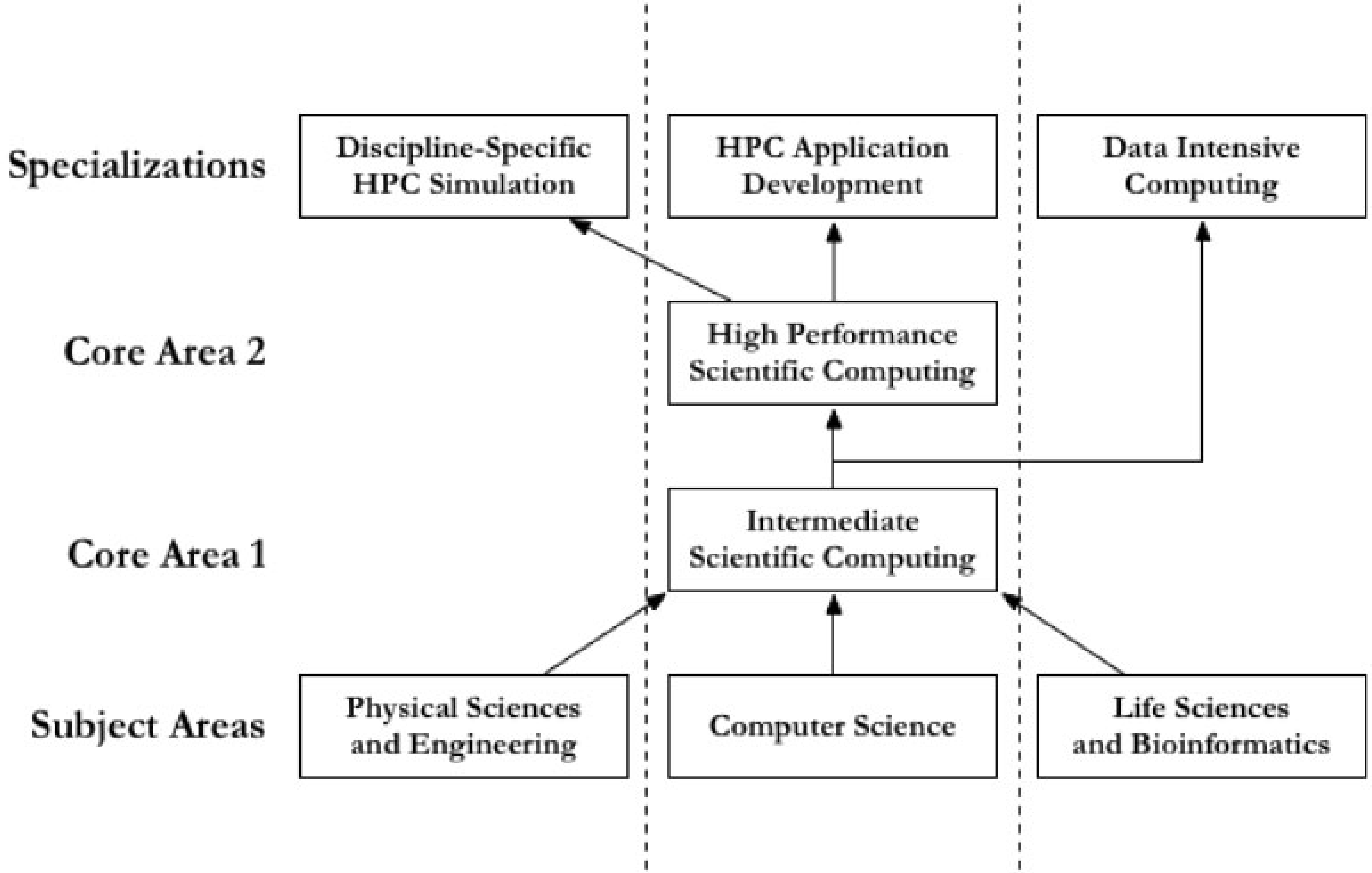
Flexibility in Implementation

- Adapt existing courses by adding computationally oriented modules
- Discipline oriented courses dependent on existing faculty expertise and interests
- Different subsets of required and optional competencies tied to major, required math, and example projects

Graduate Level Competencies

- Assumes some of the background of an undergraduate
- Focus more on research skills
- Core areas focus on the computer science and related modeling skills
- Need to branch into a wider array of specializations based on the nature of the graduate program

Graduate Competencies



http://hpcuniversity.org/educators/gradCompetencies/

Intermediate Scientific Computing

Area 1: Intermediate-Level Scientific Computing [-]

- Prereqs:[+]
- Ability to contribute code in the programming language of greatest importance in the research domain (Fortran or C):[+]
- Students will be introduced to basic debugging techniques:[+]
- Understand number representation:[+]
- Understand numerical errors:[+]
- Students will be introduced to software engineering best practices:[+]
- Ability to identify efficient file formats:[+]
- Create a program that uses at least one widely used numerical library:[+]
- Students will be introduced to the concepts of algorithm complexity:[+]
- Students will be introduced to serial program optimization concepts and techniques:[+]
- Students will understand verification and validation principles:[+]
- Students will be introduced to Monte Carlo methods:[+]

High Performance Scientific Computing

Area 2: Intro to High Performance Scientific Computing [-]

- Prereqs - Intermediate-Level Scientific Computing:[+]
- Students will be introduced to parallel architectures and execution models:[+]
- Students will be introduced to memory models for parallel programming:[+]
- Students will understand the principles of how to match algorithms, applications, and architectures:[+]
- Students will understand the concept of application scalability:[+]
- Students will understand code performance metrics:[+]
- Students will be introduced to parallel programming methods and concepts:[+]
- Data Intensive Computing:[+]
- Data management:[+]
- Understanding fault tolerance:[+]
- Students will understand basic scientific visualization concepts[+]

Getting Started

- Analyze current curriculum
  - Existing computer modeling content
  - Opportunities to illustrate science, math, engineering concepts with modeling examples
  - Comparison with competencies
- Review program models
- Devise plans for curriculum changes
  - Integration into existing courses
  - New courses

Discussion Questions

- How is computer modeling used in existing courses here?
- Which areas would benefit from additional modeling experience?
- What are the possible ties to research and internship experiences for students?
- Who should be involved in future curriculum discussions?