placed more emphasis on simulation and intelligent devices.

- Political concerns that have precluded certain activities and
come close to algorithmic ambitons
- Engineering applications where computational capabilities have

Progress in the fields,
- replace, and even allow experimentation to support further
- scientific and engineering needs for simulation to augment,
- complicated functions;
- increasing economic markets for consumer devices with more

- computational devices at many economic levels;
- an increase in the performance of computers and other

produced has increased tremendously in recent years due to:
The use of computational devices and the resulting information

What is Computational Science and Engineering?
latter is for the most part the concern of computer scientists.

The implementation of the infrastructure necessary to do so. The computers, e.g., new algorithms and models, but also implement

It includes not only inventive and implementing ways to use

computer architecture and numerical algorithms.

Science; and multiple specialties within a discipline, e.g.,
multiple high-level disciplines, e.g., meteorology and computer

CSE is multidisciplinary at many levels, i.e., it requires

form of technical constraint.

Science and engineering that are currently not solved due to some
computation and computational devices to solve problems in

DEFINITION: CSE addresses the issues involved in using
device.

only relative to a highly constrained machine e.g., a hand-held
problems and how to handle computational rates that are large
around determining what work is unnecessary in large

• Many of the most interesting recent problems have centered

methods.

complex (in terms of number of computations) numerical
require very high computational rates and increasingly
The traditional view centers on simulations of phenomena that

natural phenomena.

example, CSE is using computers to simulate fluids or other
area of specialization of the person or group defining it. For
Many different paradigmatic definitions that concentrate on the
II. Engineering: Develop algorithms and systems to solve problems and their interaction for basic science.

modelling the various parts (real or imagined) of the universe.

• to analyze.

models of phenomena that require computing to evaluate and

I. Science and Mathematics: Develop and interpret

What does what?
- Architecture and construction
- Manufacturing and robotics
- Aerospace design
- Sensing and imaging
- Design of circuits and devices
- Communications
- Signal processing
- Control of systems
- Electromagnetic simulation
III. Computer Science and Engineering

- Use tools: visualization, databases, problem solving
- Restructuring compilers, performance evaluation systems
- Design tools: prototyping environments and codes
- Functional libraries: numerical and non-numerical utility
- Compilers/code generators
- System runtime libraries, OS, languages
- Software: •
- Algorithms: numerical and non-numerical
- Architecture: general purpose, special purpose, hybrid

environments
III is mostly concerned with the tools needed to do I and II

- black hole behavior
- collision of galaxies
- are too dangerous?
- interaction of chemicals that do not exist; are too precious?

allows experimentation •

catastrophe simulation

nuclear weapon simulation

car crash simulation

replaces experimentation due to economic or political expense •

- predicts results to guide the design of experiments
- extends range of parameters – high speed air flow

augments experimentation •

In I and II computation
What do we want?

- Better tools to do all of above (computer science and engineering, applied math)
- Better machines: big and small (computer science and engineering)
- Better algorithms: numerical and nonnumerical (applied math and computer science)
- Better models: continuum and particle (science and applications)
prototyping, performance tuning etc. is still a very active area.

- The development of software to support this activity, e.g.,
  complexity, e.g., multipole algorithms
  algorithms and exposing structure that can be used to reduce
  more interesting issues are developing more sophisticated
  for problems constrained by the speed of the machine, the
  architectural ideas
  revolutionary developments are dependent on new
  performance parallel processors is a mature field and
  mapping large scale simulation problems on to high
Software people tend to observe algorithm designers and implementors to try and generalize and automate their interaction is key concern.

Observation: Simply building new machines is not necessarily CSE; architecture/application/algorithm performance. Observation: Examples of constructs can be built to improve to see if new machine constructs can be built to improve justify activity.

Hardware people tend to observe the type of algorithms used with hardware, making their process look like a nail. Avoid misapplication of technology to avoid hardware mistakes.

Danger: If you have a hammer, everything starts to look like a nail. Avoid misapplication of technology to transfers to software and hardware people in computer science applications looking for common problems for algorithm engineering and adaptation and improvement; and technology transfer, adaptation and improvement; and technology stretches.
techniques. They also watch how applications people use their codes and the results to develop data management systems.
Sytems.

their codes and the results to develop data management techniques. They also observe how application people use
implementations to try and generalize and automate their
software people tend to observe algorithmic designs and

What is done in practice?
algorithms to Application 2.
leads to real progress e.g. Application 1 to numerical
It is most often the interaction of more than one of these that

activity currently
applications into a single, hopefully useful, system is a major
algorithms to customize. Combining models from multiple
done science and engineering while watching for „new”
Applications people concentrate on building better models and

Application 3.
Illinois, J. Davidson - Virginia

- Incorporate in classroom (with D. Whalley, D. Jones -
  exploiting architecture and algorithm knowledge;
system for code generation for signal processors
  
Embedded System Code Generation System: Interactive

  (with R. Van Engelen)

- Compiler support for composition of programs developed

  (with R. Van Engelen)

- Application-sensitive compilation: FALCON, CTADEL

  (with R. Van Engelen)
matrix reasoning
parallel nonsymmetric sparse solver based on

tradesoffs.

count low, parallelism, and stability all have

If $A$ is large and sparse then keep the operation

to solve in a finite number of steps.

Transform $A$ into a simpler form that is easy to

direct methods:

- Linear System Solving: $Ax = q$, $A$ and $q$ given, find $x$

Numerical Linear Algebra
Incomplete Cholesky factorization for symmetric systems

- squares

Incomplete Gram-Schmidt-based preconditioning for least squares and efficiency are key issues.

- existence

Solve \( W = xAV \) via iterative method

\( W \approx V^{-1}V \)

Solve \( q_{W} = xAV \) via iterative method

\( q_{W} = xAV \) via iterative method

\( q_{W} = xAV \) via iterative method

- preconditioning required

- iteration, parallelization, stability are all involved in tradeoffs

- “fast” convergence (number of iterations), complexity of solution

Produce a sequence \( x^{0}, x^{1}, \ldots, x^{1} \cdot \) that approach the solution

- \( A \) is kept in original form

- Iterative methods

**Numerical Linear Algebra**
- Efficient methods for large problems, e.g., sparse systems.
- Solve systems of eigenvalues and eigenvectors.
- Singular values and vectors (dominant spaces).
- Robust hybrid nonsymmetric system solver: combination of direct/iterative methods.

Quasi-Newton methods compute approximations to $A^{-1}$ and $\alpha$ on each step. Recently rehabilitated.
It is an open and active area for large scale systems

- Generically called model reduction

- Mechanism – Information retrieval, latent semantic indexing

- Images, compress the representation into an efficient recovery

Given static data, e.g., documents/key words or a sequence of

- Dominant subspace tracking

- Approximations of previous evaluations of the model

- Captures “essential” behavior that can be used to recover
dynamical system(s), create a smaller more efficient one that

- Given observations of a model’s evolution (or of an actual

Keyl court family

- Efficient one that captures “essential” behavior – Rational

given a dynamical system as a model, create a smaller more

Efficient Information Representation and Recovery
multichannel

deconvolution, standard, blind, multiobserveration

linear algebra and computational statistics

removal of noise, blurs, jammer

system identification

image processing, communication, signal processing

Information Extraction and Recovery
Standoff or broadside mode – autofocus problem unification

Munson & Williamson, "Plumes – Wake Forest"

Synthetic Aperture Radar
High squint angle – conditioning/regularization problem

Synthetic Aperature Radar

safe zone

runway

safe zone
multiple phase shifts to recover PSF then solve

blind deconvolution via phase diversity blur with known

guide stars as point sources: natural and artificial

degraded image

find PSF consistent system and solve to recover image from

simplest model

convolution of input image and point spread function is

noise or turbulence

Suppose you are observing an image through a medium with

\((\text{Plemmons - Wake Forest})\)

Image Restoration
Adaptive control of phase mask – requires metric

Design combined phase mask and restoration algorithm

More sophisticated deburring techniques – phase diversity

Image given known PSF of optical system

Blur via cubic phase mask lens (one image only) then restore

In focus lose depth information to an extent

Can a combination of optical and digital processing put both

Beyond/before the focal length so seriously out of focus

Suppose you have an image with one object in focus and one

Premmons – Wake Forest, Vander Gucht – ART

Depth of field extension