The Program of Grand Challenge Problems: Expectations and Results

John Gustafson
Ames Lab-ISU
Ames, Iowa

This work was supported by the Applied Mathematical Sciences Program of the U.S.D.O.E., contract W-7405-ENG-82

SCL

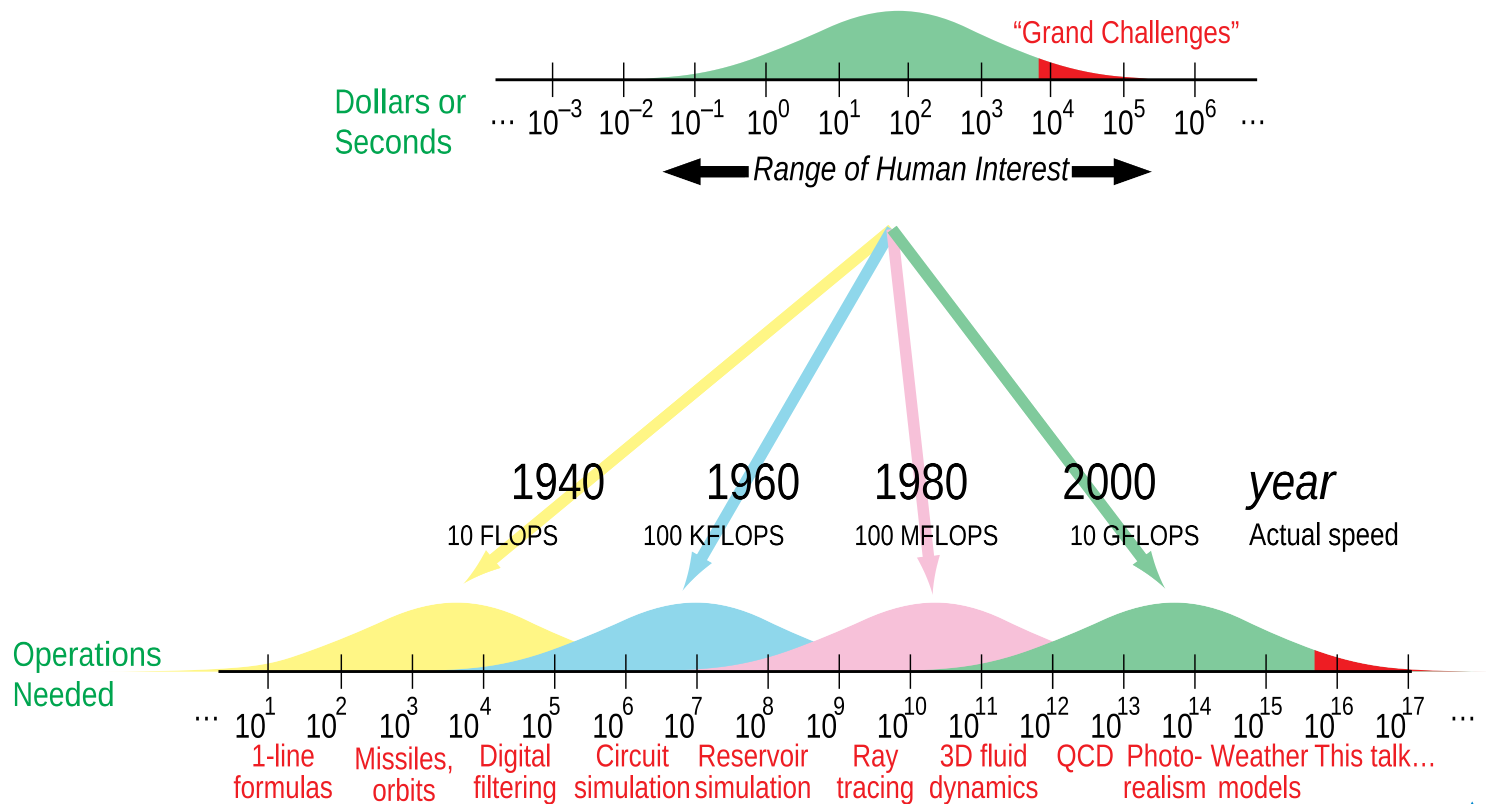
AMES LABORATORY

Thesis

"Grand Challenge" is a fundraising term.
But it makes us define goals more precisely.
Some goals are vague or irrelevant.
Rigorously-defined goals solve several long-standing shortcomings of computational science, including fair performance metrics and confidence in simulation prediction.

A History of "Grand Challenges"

Human factors limit problem size.



SCL

AMES LABORATORY

History of Concern for Validity

1940	Ignored. Just use lots of decimals.
1950	Monte Carlo debated; roundoff studied
1960	Wilkinson proves validity of linear algebra
1970	First 60-bit, 64-bit computer architectures
1980	PASCAL-SC, ACRITH, ULTRITH
1990	Ignored. Just use IEEE arithmetic.

Comparison with physical experiments is getting rarer.
Accuracy is neglected; speedup, FLOPS emphasized.
Different answers for parallel methods cause surprise.

Example: LINPACK Residuals

```
Value for n (maximum =  1160):
Please send the results of this run to:

Jack J. Dongarra
Computer Science Department
University of Tennessee
Knoxville, Tennessee 37996-1300

Fax: 615-974-8296

Internet: dongarra@cs.utk.edu

    norm. resid      resid           machep         x(1)          x(n)
  1.33497627E+01  2.96423996E-12  2.22044605E-16  1.00000000E+00  1.00000000E+00


   times are reported for matrices of order  1000
     factor     solve      total     mflops       unit      ratio
times for array with leading dimension of 580
  2.415E+01  1.844E+00  2.600E+01  2.572E+01  7.776E-02  4.642E+02
```

Which of these gets the most attention?

AMES LABORATORY

Recent Ames Lab Observations

- Most graphics rendering is grossly incorrect (but looks great).
- “Order *N* ” methods for *N* -body are nothing of the kind.
- 64-bit arithmetic is used with methods valid to less than two decimals.
- PDE error analysis is poorly done or omitted; *O*(…) notation at best.
- Monte Carlo methods beat many "exact" methods on closer examination.
- The literature equates answer quality with number of discrete variables.

Means-Based vs Ends-Based Metrics

MEANS-BASED		ENDS-BASED
Flop/s	⟷	Time to Compute Answer
Bytes of RAM	⟷	Detail, Content of Answer
Number of Processors	⟷	Feasible Problems to Attempt
Use of Commodity Parts	⟷	Cost, Availability of System
Word Size	⟷	Closeness to Actual Physics
ECC Memory	⟷	Reliability of Answer
Speedup	⟷	Product Line Range

Which Algorithm Would You Pick?

Explicit Timestepping ⟷	Implicit Timestepping
Conventional Matrix Multiply ⟷	Strassen, Winograd Methods
Cholesky Decomposition ⟷	PC Conjugate Gradient
All-to-All N-Body Methods ⟷	Barnes-Hut, Greengard
Successive Over-Relaxation ⟷	Multigrid
Time-Domain Operators ⟷	FFT's
Recompute Gaussian Integrals ⟷	Compute Once and Store
Material Property Function ⟷	Table Look-Up
HIGHER FLOP/S RATES	FASTER ANSWERS

SCL

AMES LABORATORY

Accelerated Strategic Computing Initiative (ASCI) Example

Scenario 1:
(cost of about $100M)

Parallel computer rated at 3 TFLOPS sustains 1 TFLOPS, 70% parallel efficiency modeling nuclear weapon test.
No proof of correctness, no accuracy goal.

Scenario 2:
(cost of about $0.01M)

Computer rated at 0.0002 TFLOPS sustains unknown TFLOPS and efficiency modeling nuclear weapon test.
Answers have 95% confidence, match prior physical experiments.

Which scenario has the higher performance? Is TFLOPS a valid goal?

"Grand Challenge" Examples: The Factoring of RSA 129

- 5000 MIPS years
- 0.1% of Internet used in 1994.
- 100% of Internet would have solved problem in 3 hours.
- Over 10^{17} operations.
- RIGOROUSLY DEFINED GOAL (and enthusiastic support).

“Grand Challenge” Examples: The Production of Toy Story

- Distributed over dozens of Sun workstations, ~10 MIPS per Sun
- 140,000 frames to render for full-length feature film
- 10,000 seconds per frame (!)
- About 10^{17} operations, same as RSA 129.
- Goal was defined, though not rigorously. Esthetics play a role.

SCL

AMES LABORATORY

"Grand Challenge" Examples: The Production of Toy Story

- Distributed over dozens of Sun workstations, ~10 MIPS per Sun
- 140,000 frames to render for full-length feature film
- 10,000 seconds per frame (!)
- About 10^{17} operations, same as RSA 129.
- Goal was defined, though not rigorously. Esthetics play a role.

N-Body Challenges

- Planetary position was a Grand Challenge in the 1940s.
- Size of *N* often taken as figure of merit. “Billion-particle simulation.”
- GRAPE processor project uses all-to-all method, measures ops/sec.
- Materials science, astrophysics, and fusion all require *N* -body variants.
- Greengard, Barnes-Hut et al. made force calculation take $O(Np^2)$ work

To double the physical accuracy of any *N* -body method appears to take at least four times as much work. This is not generally understood. Physical accuracy and *N* are not proportional.

Measuring Answer Quality

If F is the answer, bound it rigorously by F^+ and F^-.
Define total error as

$$E = \int\int\int\int (F^+ - F^-)\ dx\, dy\, dz\, dt$$

and define the answer quality $Q = 1 / E$.

This has several desirable consequences:

- Removes need for flops/second or instructions/second metrics
- Allows fair comparison of different algorithms and computer architectures
- Permits clear and rigorous statement of goal for Grand Challenges

One Approach: Integral Equations

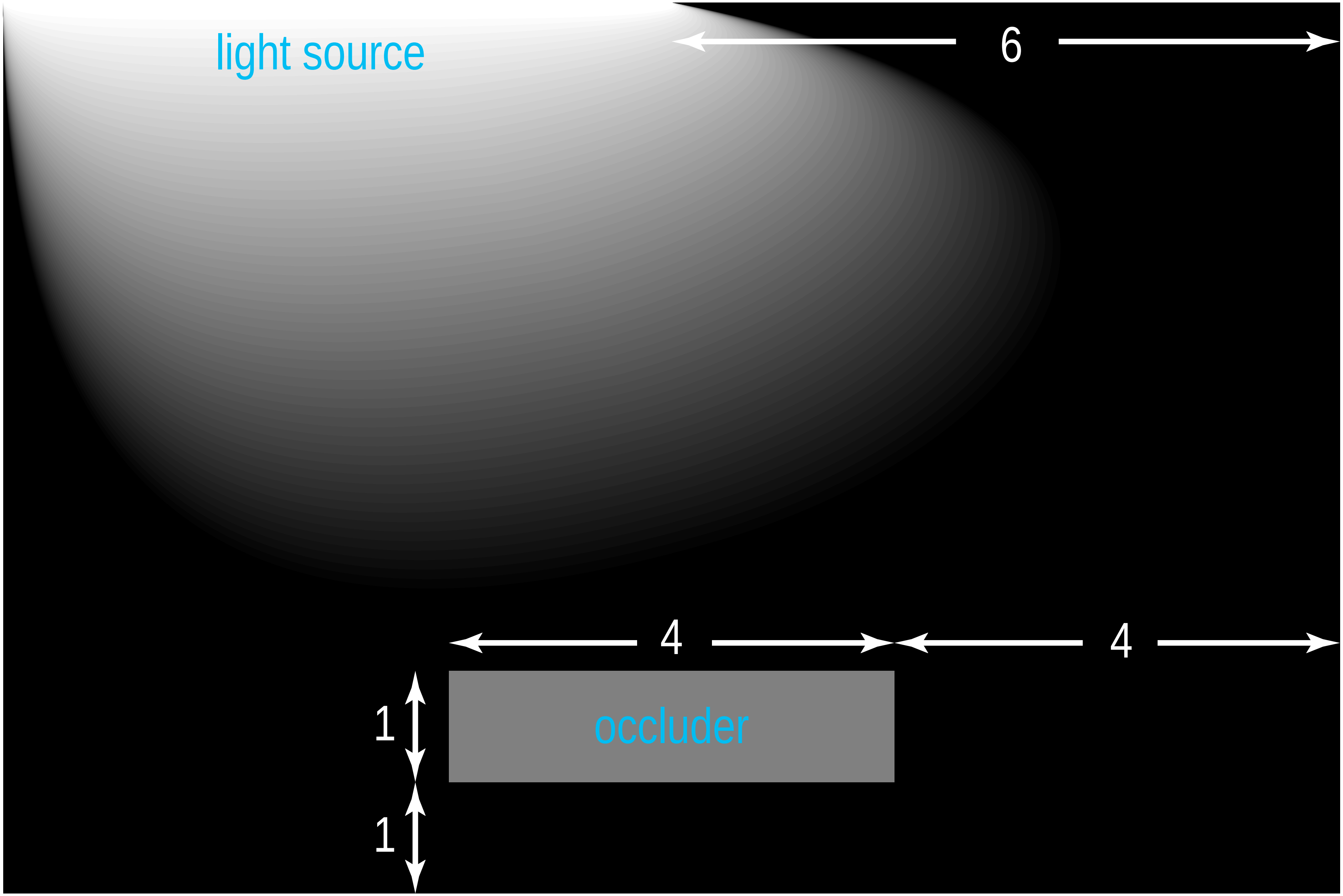
One way to find F^+ and F^- is to restate the PDE as an integral equation, if possible. Integral equations of the Second Kind are usually tractable.

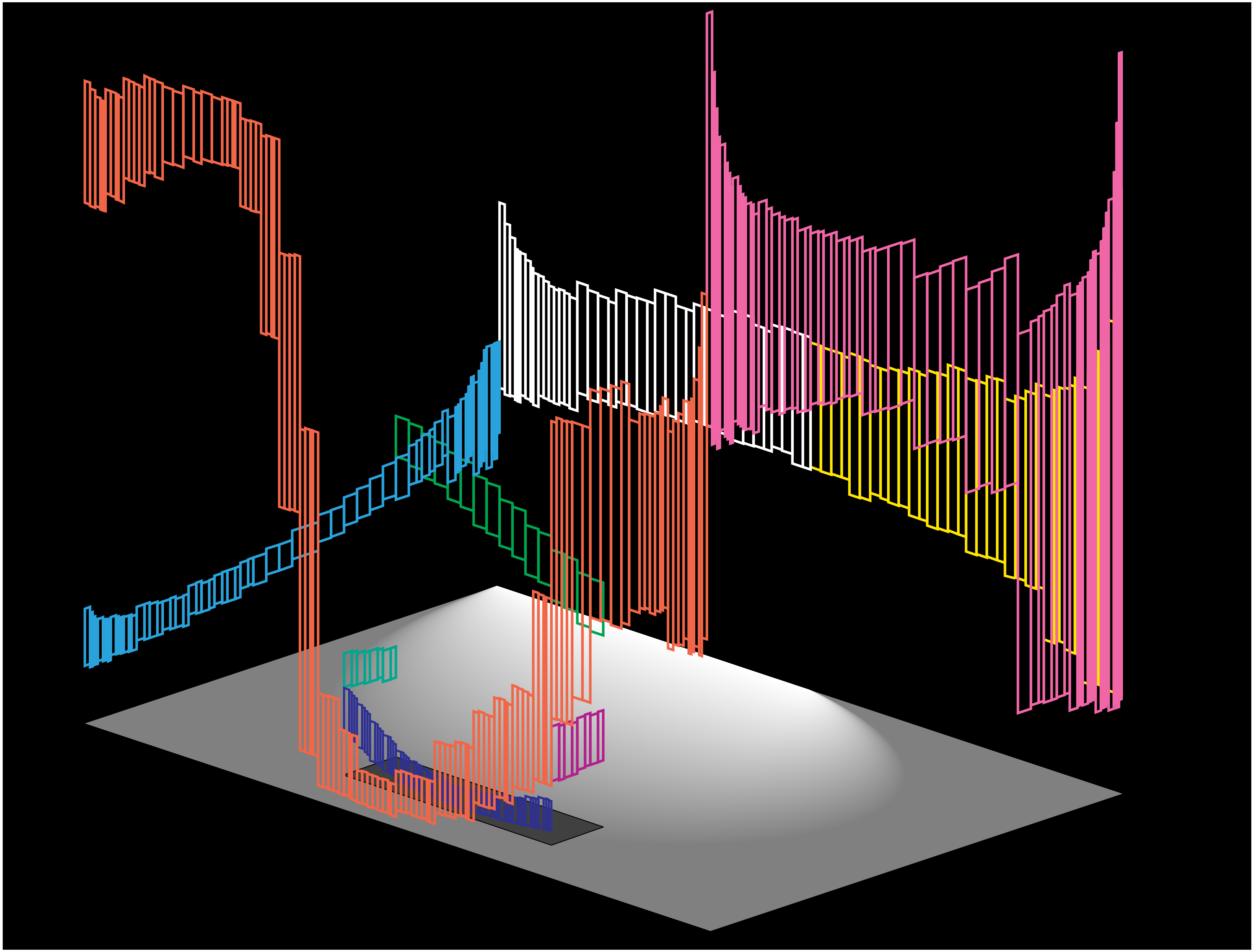
$$f(x) - \int K(x, s)\, f(s)\, ds = g(x)$$

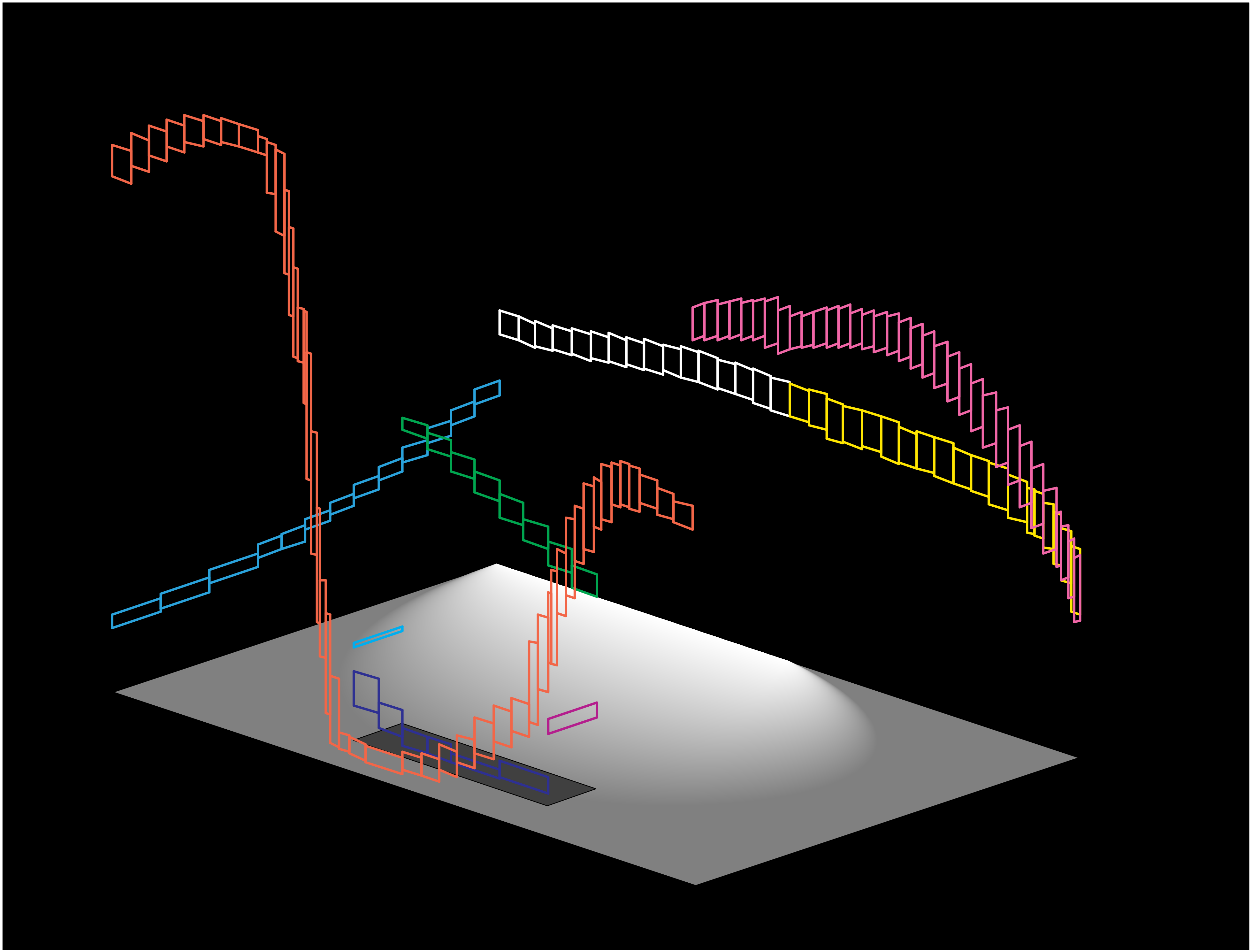
One can bound each variable on a discretization, and bound the integral. Physical reasoning may be needed to get an initial bound.

We have found quality definitions and corresponding algorithms for

- Nonlinear ODEs
- Heat transfer problems
- The *N* -body problem
- Laplace's equation

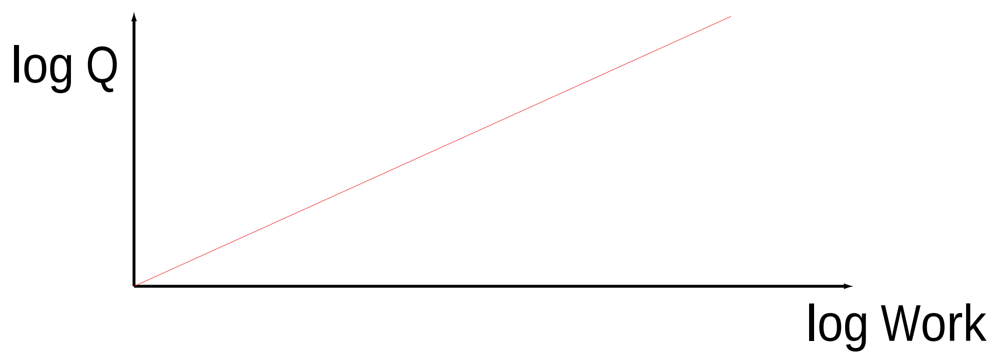

light source
6
4
4
1
occluder
1





Comparison with Monte Carlo

For examples we have studied, quality typically grows as the square root of the number of operations once quality is defined.



This is the same as for Monte Carlo methods, if one uses confidence intervals instead of rigorous bounds! Is there a sort of “conservation law” at work here? Should we return to Monte Carlo approaches?

Summary

The HPC community has undertaken many large computing efforts without defining “success” at those efforts. This is especially true for continuum simulation problems.

Successes have occurred where the goal definition was clear and rigorous. Progress in “Grand Challenges” requires this.

The use of integral equations, multidimensional integrals to define error, and quality as the reciprocal of error, can bring many continuum problems to the required level of clarity and rigor.