Grumbles of a Numerical Analyst

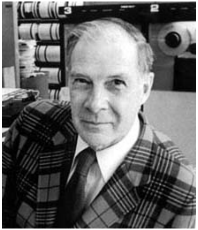
Some Ideas, Questions, and Wishes

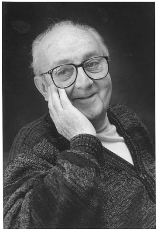
David Bindel

Department of Computer Science
Cornell University

Dagstuhl Seminar 15491:
Approximate and Probabilistic Computing
1 Dec 2015

The Application Interface





The purpose of computing is insight, not numbers.

Richard Hamming

All models are wrong but some are useful.

George E. P. Box

Digging down to error analysis

If rounding errors vanished, 95% of numerical analysis would remain.

Nick Trefethen

$$\begin{aligned}\text{Computational science} &\neq \text{Numerical methods}\\ \text{Numerical methods} &\supset \text{Numerical error analysis}\\ \text{Numerical error analysis} &\supset \text{Floating point analysis}\end{aligned}$$

Floating point is important, but let's start higher level.

Sources of error

- Floating point errors
- Truncation of approximation formulas
- Termination of iterations
- Statistical error in Monte Carlo
- What do I call approximate accelerator error?

Also: model error and input error (out of our control)

Forward and backward

- Forward error: Is my solution almost right?
- Backward error: Did I solve almost the right problem?
- Residual error: How close am I to satisfying equations?

Often residual error *is* a backward error

Forward and backward

- Small backward error $\neq$ small forward error
  - Sometimes small backward error is all you need!
- Condition number connects forward/backward error
  - Well-conditioned transformation don't amplify error
  - Orthogonal transformations a building block for robustness
- Depends implicitly on a *class of problems*
- Maintaining structural features enables finer analysis!

Issue: Fastest algorithms may *not* be backward stable!

Approximation and fixed point

Goal: Find fixed point of $F(x)$ with Lipschitz const $\alpha < 1$

$$\begin{array}{r} x_{k+1} = F(x_k) + d_k \\ x = F(x) \\ \hline e_{k+1} = (F(x_k) - F(x)) + d_k \end{array}$$

Take bounds:

$$|e_{k+1}| \leq \alpha |e_k| + |d_k|$$

Approximation and fixed point

From $|e_{k+1}| \leq \alpha |e_k| + |d_k|$, get

$$E[|e_k|] \leq \alpha^k |e_0| + \sum_{j=1}^{k} \alpha^{k-1} |d_{k-j}|.$$

For a uniform bound $E[|d_k|] < \epsilon$, we have

$$E[|e_k|] \leq \alpha^k |e_0| + \epsilon/(1-\alpha)$$

For $E[|d_k|] \to 0$, have $E[|e_k|] \to 0$.

Convergence beyond fixed point iteration

- Usual use case: $Mx_{k+1} = Kx_k + b$

$$x_{k+1} = Rx_k + c, \quad R = M^{-1}K, c = M^{-1}b$$

- Convergence rate depends on $\|R\|$. Limit bwd stable (mod bad M)
- Alternate: extrapolated estimate

$$\bar{x}_k = \sum_{j=0}^{k} \alpha_{jk} x_k$$

- Equivalent: look for solutions over *Krylov subspace*

$$\mathcal{K}_k(R, c) = \operatorname{span}(c, Rc, R^2c, \ldots, R^{k-1}c) = \{p(R)c : p \in \mathcal{P}_{k-1}\}$$

- Krylov acceleration *less forgiving* of varying approximation

Errors beyond floating point

Possible error model:

- Take a correct (in FP) step
- Get an entry or two wrong with low probability

Likely gives *better results* than norm bounds alone would imply.

And I might know how to make this work with Krylov acceleration.

Questions and comments

Interface: Deterministic error bd + nondeterministic performance

- Alternate: expensive step with exact residual
- Does this cover enough ground? (I think yes)
- What abstractions help PL tools do high-level reasoning?

Diving downward

Standard model

$$\mathsf{fl}(x * y) = (x * y)(1 + \delta)$$

Neither sound nor complete, but useful.

Linearized $1 + \delta$

Simple analysis $f = x + \sqrt{y - z}$

$$\text{fl}(f) = (x + \sqrt{(y - z)(1 + \delta_1)}(1 + \delta_2))(1 + \delta_3)$$

$$= f + \sqrt{y - z}(\delta_1/2 + \delta_2) + f\delta_3 + O(\epsilon^2)$$

Teaching tool idea:

- Form AST for $1 + \delta$ model
- Differentiate with respect to δ vars
- Analyze partials at “interesting” points

Currently mostly implemented thanks to Humam Alwassel.

More Grumbles

See the associated commentary!