What SAGE Needs to be Useful for Applied Mathematics

(in my corner of this world, at least)

Randall J. LeVeque
Department of Applied Mathematics
University of Washington

or....

Can SAGE be

Software for Applied mathematics,
Graphics, and Engineering?

Outline

- Teaching — replace Matlab, Maple, etc.?
- Research — e.g., Numerical analysis
- Scientific computing, data manipulation, visualization
- `chebfun` — filling the gap between symbolic and numerical computing?

Advantages of SAGE

- Free and open source
  - Students can install at home, on laptop, iPhone, etc.
  - Students can use after leaving university,
  - Researchers many places can't afford Matlab, Maple, etc.

Advantages of SAGE

- Free and open source
  - Students can install at home, on laptop, iPhone, etc.
  - Students can use after leaving university,
  - Researchers many places can't afford Matlab, Maple, etc.
- Single interface for symbolic and numerical computing, data manipulation, visualization, etc.
  - Many applications require combinations of different techniques
  - Also supports latex, webpages, etc.

Advantages of SAGE

- Free and open source
  - Students can install at home, on laptop, iPhone, etc.
  - Students can use after leaving university,
  - Researchers many places can't afford Matlab, Maple, etc.
- Single interface for symbolic and numerical computing, data manipulation, visualization, etc.
  - Many applications require combinations of different techniques
  - Also supports latex, webpages, etc.
- Python scripting is popular for scientific computing
  - Manipulating large data sets in various formats,
  - Coupling together diverse software in different languages,
  - Creating modern interfaces and GUIs for old codes,
  - Organizing benchmark tests, validations, parameter studies, etc.

Python for numerics and scientific computing

Many packages and modules already available for

- Numerical methods: `NumPy, SciPy, MatPy, Pysparse, Signaltools, ...`
- Graphics and visualization: `Gnuplot, PythonPlot, MayaVi, gracePlot.py, NURBS, ...`
- Interfacing with other languages: `f2py, swig, pymat, ...`

Reference: Hans Petter Langtangen, *Python Scripting for Computational Science*, Springer, 2004.

Matlab functionality

Sparse matrix operations essential for many applications.

Example: Solve $u''(x) = f(x)$ on $0 \leq x \leq 1$
with boundary conditions $u(0) = u(1) = 0$.

Let $U_j \approx u(x_j), \quad j = 1,\ 2,\ \ldots,\ m$
where $x_j = jh$ with $h = 1/(m+1)$.

Replace ODE by finite difference equations

$$\frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1}) = f(x_j), \quad j = 1,\ 2,\ \ldots,\ m.$$

This is a tridiagonal linear system of m equations.

Matlab spdiags and backslash

```
x = h*(1:m)';
e = ones(m,1);
A = 1/h^2 * spdiags([e -2*e e], [-1 0 1], m, m);
b = f(x);
u = A\b;
```

The tridiagonal system is solved in $O(m)$ operations,
not $O(m^3)$ as needed for a dense $m \times m$ matrix.

2D Poisson problem

Consider $u_{xx}(x,y) + u_{yy}(x,y) = f(x,y)$ on unit square

with $u = 0$ on boundaries.

Finite difference method with "5-point stencil":

$$\frac{1}{h^2}\left[(U_{i-1,j} - 2U_{ij} + U_{i+1,j}) \right.$$
$$\left. + \ (U_{i,j-1} - 2U_{ij} + U_{i,j+1})\right], \quad i,\ j = 1:m.$$

2D Poisson problem

Consider $u_{xx}(x,y) + u_{yy}(x,y) = f(x,y)$ on unit square

with $u = 0$ on boundaries.

Finite difference method with “5-point stencil”:

$$\frac{1}{h^2}\left[(U_{i-1,j} - 2U_{ij} + U_{i+1,j}) \right.$$
$$\left. + (U_{i,j-1} - 2U_{ij} + U_{i,j+1})\right], \quad i,\ j = 1:m.$$

Natural row-wise ordering of unknowns:
(not the best for Gaussian elimination!)

$$U = \begin{bmatrix} U^{[1]} \\ U^{[2]} \\ \vdots \\ U^{[m]} \end{bmatrix}, \quad \text{where } U^{[j]} = \begin{bmatrix} U_{1j} \\ U_{2j} \\ \vdots \\ U_{mj} \end{bmatrix}.$$

Gives sparse $m^2 \times m^2$ matrix with bandwidth m.

2D Poisson problem

$$\frac{1}{h^2}\left[(U_{i-1,j} - 2U_{ij} + U_{i+1,j}) + (U_{i,j-1} - 2U_{ij} + U_{i,j+1})\right]$$

Gives sparse $m^2 \times m^2$ matrix with bandwidth m (block $m \times m$):

$$A = \frac{1}{h^2}\begin{bmatrix} T & I & & & \\ I & T & I & & \\ & I & T & I & \\ & & \ddots & \ddots & \ddots \\ & & & I & T \end{bmatrix}, \qquad U = \begin{bmatrix} U^{[1]} \\ U^{[2]} \\ U^{[3]} \\ \vdots \\ U^{[m]} \end{bmatrix},$$

Each block T or I is itself an $m \times m$ matrix,

$$T = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & 1 & -4 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -4 \end{bmatrix}$$

2D Poisson problem

```
x = h * (1:m)';   y = x;    [X,Y] = meshgrid(x,y);

I = eye(m);  e = ones(m,1);
T = spdiags([e -2*e e], [-1 0 1], m, m);
A = 1/h^2  * (kron(I,T) + kron(T,I));

b = f(X,Y);
bvec = reshape(b,m^2,1);
uvec = A\bvec;
u = reshape(uvec,m,m);
```

Matlab uses Gaussian elimination with smart ordering.

For Poisson problem, better approach is FFT.

For general sparse systems, iterative methods often used.

Other numerical software needs and issues

- Proper use of IEEE arithmetic, including NaN's
  In Matlab: 1/0 = Inf, 0/0 = NaN

Other numerical software needs and issues

- Proper use of IEEE arithmetic, including NaN's
  In Matlab: 1/0 = Inf, 0/0 = NaN
- Backward compatibility: Old SAGE code should continue to work in the future.

  Crucial for software development, reproducibility, textbook use.

Other numerical software needs and issues

- Proper use of IEEE arithmetic, including NaN's
  In Matlab: 1/0 = Inf, 0/0 = NaN
- Backward compatibility: Old SAGE code should continue to work in the future.

  Crucial for software development, reproducibility, textbook use.
- Parallel computing: Crucial for many large problems, even as processors get faster (especially now since most are multi-core).

  Grid computing, GPU computing?

Taylor series

Need symbolic manipulation of Taylor series in several variables for general functions, e.g. in Maple:

```
> mtaylor(u(x+h,t+k),[h,k],3);

 u(x, t) + D[1](u)(x, t) h + D[2](u)(x, t) k

                                  2
     + 1/2 D[1, 1](u)(x, t) h   + h D[1, 2](u)(x, t) k

                                  2
     + 1/2 D[2, 2](u)(x, t) k
```

Graphics and Visualization

Matlab is so popular largely because it combines

- simple programming capability,
- interfaces to high quality software,
- data manipulations tools,
- *and simple and powerful graphics*.

Numerical results often consist of approximations to functions at millions of grid points — graphics is the only way to view and interpret.

Graphics and Visualization

Some issues to keep in mind...

- Need to support 1, 2, and 3 space dimensions. Often the grids are nonuniform and domain is complicated.
- Often solution is time dependent and one wants to easily make an animation of how it evolves.

Graphics and Visualization

Some issues to keep in mind...

- Need to support 1, 2, and 3 space dimensions. Often the grids are nonuniform and domain is complicated.
- Often solution is time dependent and one wants to easily make an animation of how it evolves.
- Many approaches to 3D visualization, e.g., isosurfaces, contour lines on slices, volume rendering (voxel graphics), velocity vectors, streamlines, quantities on bounding surfaces, etc.

Graphics and Visualization

Some issues to keep in mind...

- Need to support 1, 2, and 3 space dimensions. Often the grids are nonuniform and domain is complicated.
- Often solution is time dependent and one wants to easily make an animation of how it evolves.
- Many approaches to 3D visualization, e.g., isosurfaces, contour lines on slices, volume rendering (voxel graphics), velocity vectors, streamlines, quantities on bounding surfaces, etc.
- Adaptive mesh refinement may be used — some regions are covered by multiple grids.

Graphics and Visualization

Some issues to keep in mind...

- Need to support 1, 2, and 3 space dimensions. Often the grids are nonuniform and domain is complicated.
- Often solution is time dependent and one wants to easily make an animation of how it evolves.
- Many approaches to 3D visualization, e.g., isosurfaces, contour lines on slices, volume rendering (voxel graphics), velocity vectors, streamlines, quantities on bounding surfaces, etc.
- Adaptive mesh refinement may be used — some regions are covered by multiple grids.
- Lots of graphics packages exist — don't want to reinvent all this! Need basic graphics well integrated and perhaps front end to more sophisticated packages.

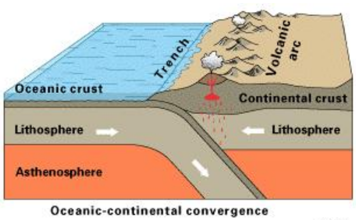
Sumatra event of December 26, 2004

Magnitude 9.1 quake near Sumatra, where Indian tectonic plate is being subducted under the Burma platelet.

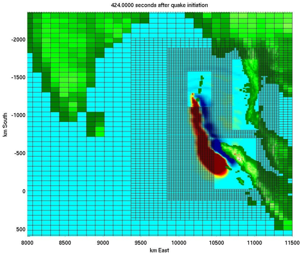
Rupture along subduction zone
$\approx$ 1200 km long, 150 km wide

Propagating at $\approx$ 2 km/sec (for $\approx$ 10 minutes)

Fault slip up to 15 m, uplift of several meters.
(Fault model from Caltech Seismolab.)



www.livescience.com



Tsunami simulations

Adaptive mesh refinement is essential

Zoom on Madras harbors with 4 levels of refinement:

- Level 1: 1 degree resolution ($\Delta x \approx 60$ nautical miles)
- Level 2 refined by 8.
- Level 3 refined by 8: $\Delta x \approx 1$ nautical mile (only near coast)
- Level 4 refined by 64: $\Delta x \approx 25$ meters (only near Madras)

Factor 4096 refinement in x and y.

Less refinement needed in time since $c \approx \sqrt{gh}$.

Runs in a few hours on a laptop.

For animation and other related results, please visit
`http://www.amath.washington.edu/~rjl/talks/hilo06/`

chebfun

On-going research problem led by Nick Trefethen (Oxford),

with Zachary Battles, Ricardo Pachón,

Toby Driscoll (Delaware).

References:

Z. Battles and L. N. Trefethen, *An extension of Matlab to continuous functions and operators*, SIAM J. Sci. Comp. 25 (2004), pp. 1743–1770.

`http://web.comlab.ox.ac.uk/projects/chebfun/`

chebfun

Example: $f(x) = \exp(-x^2)\sqrt{x+2}$.

Suppose we need to work with $g(x) = \int_{-1}^{x} f(y)\,dy$.

Symbolic manipulation fails.

chebfun

Example: $f(x) = \exp(-x^2)\sqrt{x+2}$.

Suppose we need to work with $g(x) = \int_{-1}^{x} f(y)\,dy$.

Symbolic manipulation fails.

Numerical approach:
For example, $g(0) \approx$ `f.nintegrate(x, -1, 0)`

Example: $f(x) = \exp(-x^2)\sqrt{x+2}$.

Suppose we need to work with $g(x) = \int_{-1}^{x} f(y)\,dy$.

Symbolic manipulation fails.

Numerical approach:

For example, $g(0) \approx$ `f.nintegrate(x, -1, 0)`

What if we need to ...

- compute $\|g\| = \left(\int_{-1}^{1} |g(x)|^2\,dx\right)^{1/2}$?
- compute $(g, f) = \int_{-1}^{1} g(x) f(x)\,dx$?
- work with $h(x) = \int_{-1}^{x} g(z)\,dz$?

chebfun

```
>> x = chebfun('x');
>> f = exp(x.^2) .* sqrt(2+x);
>> g = cumsum(f);

>> f2 = diff(g);
>> norm(f-f2)
ans =
     2.760689672717827e-14

>> h = cumsum(g);
>> g'*f
ans =
     8.316267551154510e+00

>> disp([size(f)   size(g)   size(h)]
   -21  1   -22  1   -23  1
```

Approximation by polynomials

Weierstrass approximation theorem: If $f \in C[-1,1]$ and p_n^* is the best approximation to $f(x)$ by a polynomial of degree n, then

$$\|p_n^* - f\|_\infty = \max_{-1\le x\le 1} |p_n^*(x) - f(x)| \to 0 \text{ as } n \to \infty.$$

Finding $p_n^*(x)$ is hard (too slow).

Approximation by polynomials

Weierstrass approximation theorem: If $f \in C[-1,1]$ and p_n^* is the best approximation to $f(x)$ by a polynomial of degree n, then

$$\|p_n^* - f\|_\infty = \max_{-1 \le x \le 1} |p_n^*(x) - f(x)| \to 0 \text{ as } n \to \infty.$$

Finding $p_n^*(x)$ is hard (too slow).

Interpolating polynomial: Given any $n+1$ points there exists a unique polynomial $P_n(x)$ of degree $\le n$ satisfying

$$P_n(x_j) = f(x_j), \quad j = 0,\ 1,\ \ldots,\ n.$$

Many ways to compute, barycentric interpolation is best.

Polynomial Interpolation

Choice of interpolating points $x_0, \ldots, x_n$.

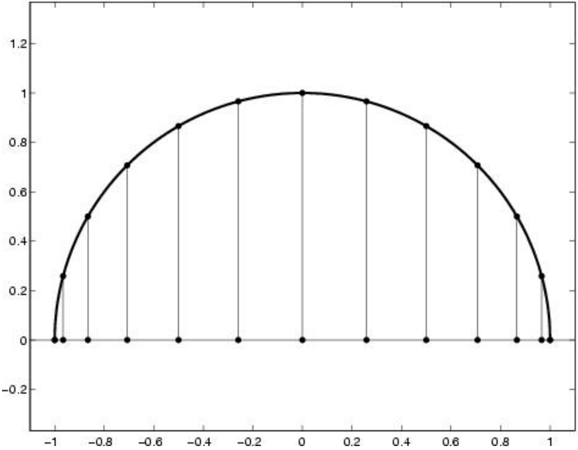
Bad choice: Equally spaced, $x_j = -1 + jh$ with $h = 2/n$.

Runge phenomenon, $\|P_n - f\|_\infty$ may blow up as $n \to \infty$

Polynomial Interpolation

Choice of interpolating points $x_0, \ldots, x_n$.

Good choice: Chebyshev points $x_j = \cos\left(\frac{\pi j}{n}\right)$



Polynomial Interpolation at Chebyshev points

Suppose $f(x)$ is analytic in ellipse enclosing $[-1,\ 1]$ with major and minor axes of length L and ℓ.

Then

$$\max_{-1\leq x\leq 1} |P_n(x) - f(x)| \leq CK^{-n}$$

with $K = L + \ell$.

Polynomial Interpolation at Chebyshev points

Suppose $f(x)$ is analytic in ellipse enclosing $[-1,\ 1]$ with major and minor axes of length L and ℓ.

Then

$$\max_{-1\le x\le 1} |P_n(x) - f(x)| \le CK^{-n}$$

with $K = L + \ell$.

Moreover,

$$\|P_n - f\|_\infty < 10\|p_n^* - f\|_\infty$$

for $n < 10^5$ (and within a factor of 100 for $n < 10^{66}$).

Polynomial Interpolation at Chebyshev points

Suppose $f(x)$ is analytic in ellipse enclosing $[-1,\ 1]$ with major and minor axes of length L and ℓ.

Then

$$\max_{-1\leq x\leq 1} |P_n(x) - f(x)| \leq CK^{-n}$$

with $K = L + \ell$.

Moreover,

$$\|P_n - f\|_\infty < 10\|p_n^* - f\|_\infty$$

for $n < 10^5$ (and within a factor of 100 for $n < 10^{66}$).

Spectral accuracy: error goes to zero faster than n^{-p} for all p. (and some finite p depending on smoothness of f if it's not analytic near interval).

chebfun

On-going research including:

- Backward error analysis,
- Extension to piecewise continuous functions,
- Continuous analogues of Householder and LU factorizations,
- Global optimization,
- Extension to 2D and 3D,
- Krylov space iterative methods for operators,
- Spectral methods for PDEs

`http://web.comlab.ox.ac.uk/projects/chebfun/`