Lecture 4: Sci. Comp. for DPhil Students

Nick Trefethen, Thursday 24.10.19

Today

- I.8 The definition of numerical analysis
- I.9 Overview of matrix iterations
- I.10 Lanczos iteration
- I.11 Numerical software tools and information sources

Handouts

- "Distracting digits"
- Quiz 3 (Matlab syntax)
- `m06_lanczos.m` and `lanczos.m`
- Essay "Numerical Analysis" from *Princeton Companion to Mathematics*
- Numerical software tools and information sources
- List of students in the course

Announcements

- Assignment 1 returned at end of lecture
- Assignment 2 posted online tomorrow
- Please do the quiz

I.8 The definition of Numerical Analysis

Definition of Numerical analysis/scientific computing:

> *The study of algorithms for the problems of continuous mathematics.*

Our PCG discussion touches upon one of the fascinating aspects of these algorithms nowadays. The best algorithms for large problems are often iterative, even when there exists a noniterative algorithm that in principle would get the exact answer in a finite number of steps. For other examples and discussion, see the appendix to Trefethen & Bau, "The definition of numerical analysis". Also see this essay "Numerical Analysis" from the *Princeton Companion to Mathematics.*

In particular, see Section 7 of the *PCAM* essay, especially its final paragraph.

The essay is a decade years old. The biggest thing that strikes me that has happened since then is the rise of randomness: new questions with a random aspect to them; and new algorithms for very large problems exploiting randomness.

I.9 Overview of matrix iterations

Good reading: Trefethen & Bau Lecture 32

The two biggest problems in large-scale matrix computation are:

$Ax = b$ (linear system of equations)

$Ax = \lambda x$ (eigenvalue problem)

And there are two big classes of matrices: symmetric or nonsymmetric. Here's an outline of the famous methods for these problems:

```
                     Ax = b                       Ax = lam x
          _________________________________________________________
         |                              |                           |
         |  CG  (if SPD)                |  Lanczos                  |
        T|                              |                           |
 A   = A |  MINRES                      |  Implicitly restarted L.  |
         |                              |                           |
         |  SYMMLQ                      |  J-D (Jacobi-Davidson)    |
         |______________________________|___________________________|
         |                              |                           |
         |  CGN / LSQR / LSMR           |  Arnoldi                  |
        T|                              |                           |
 A  ~= A |  GMRES                       |  Implicitly restarted A.  |
         |                              |                           |
         |  QMR,BiCG,CGS,BiCGSTAB,...   |  J-D                      |
         |______________________________|___________________________|

            see Saad 2 & Templates 2       see Saad 1 & Templates 1
```

For $Ax = b$, the crucial thing is getting a good preconditioner. If you've got one, all methods are good. If you don't. . . .

Maybe BiCGSTAB is the best? The 1992 BiCGSTAB paper by van der Vorst was the most heavily cited maths paper in the 1990s! — 5936 hits on Google Scholar as of today.

But CGN is the simplest: convert $Ax = b$ to $A^T Ax = A^T b$ and use CG. A bonus is that this also works for least-squares (rectangular A).

$Ax = b$ in MATLAB: PCG, MINRES, SYMMLQ, LSQR, GMRES, QMR, BICG, CGS, BICGSTAB

$Ax = \lambda x$ in MATLAB: EIGS

I.10 Lanczos iteration

Read: Trefethen & Bau Lec. 36.

History

- Lanczos 1950.
- Closely related to CG, though people didn't fully see that at first. "Rediscovery" - Paige 1971.

Given: symmetric matrix A. Whether A is SPD doesn't matter — the eigenvalue problems for A and $A + cI$ are equivalent.

What Lanczos does is construct an orthogonal similarity transformation of A to a tridiagonal matrix $T = QAQ^T$:

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \end{pmatrix}$$

Instead of going all the way (which wouldn't be accurate anyway because of rounding errors) we stop after a smallish number of rows and columns n and find that the eigenvalues of T_n are often good approximations of some of those of A.

Watch out: there are often some spurious repeats — "ghosts".

As with CG, we will state the iteration but not derive it. Also as with CG, A enters only via a matrix-vector multiplication at each step, so the method takes advantage of sparsity.

For a vector x, $\|x\|$ denotes the **norm** or **2-norm** defined by

$$\|x\| = (x^T x)^{1/2} = (\sum_j x_j^2)^{1/2}.$$

$\beta_0 = 0, \quad q_0 = 0, \quad b = \text{arbitrary}, \quad q_1 = b/\|b\|$

for $n = 1, 2, 3, \ldots$

$v = Aq_n$

$\alpha_n = q_n^T v$

$v = v - \beta_{n-1} q_{n-1} - \alpha_n q_n$

$\beta_n = \|v\|$

$q_{n+1} = v/\beta_n$

Compute the eigenvalues of the $n \times n$ matrix

$T_n = \text{tridiag}(\alpha_1, \ldots, \alpha_n; \beta_1, \ldots, \beta_{n-1})$

end

In MATLAB the `eigs` code does this and more:
"implicitly restarted Lanczos/Arnoldi" (from Fortran package ARPACK).

`[lanczos.m and m06_lanczos.m]`

There are some video interviews of Cornelius Lanczos (1893–1974) on YouTube.
You might enjoy watching a few minutes to get a sense of a remarkable man.

I.11 Numerical software tools and information sources

An online tour of the offerings at our course Web page under "Numerical Software Tools and Information Sources"

This site is a valuable resource. I recommend that you explore it in your browser, and read over the hardcopy sheet at leisure.

[handout: `people.maths.ox.ac.uk/trefethen/tools.html`]