

Notes on Scientific Computing

Nilay Kumar

Last updated: January 30, 2013

0 Administrativia

A standard reference on numerics is *Numerical Recipes*. For a standard programming book on C, Kernighan and Ritchie is an excellent place start to start.

There will be 6 to 8 homework sets and a final project. There will be a review session every Friday at 3pm.

0.1 Tentative topics

- Introduction to basic numerics and numerical stability
- Simple differential equation solvers
- Statistical mechanics of liquid argon; simulate $10^3 - 10^4$ interacting argon atoms; compute equation of state $f(P, V, T) = 0$
- Monte Carlo techniques for a canonical ensemble; MC used to simulate thermal fluctuations; random number generation
- General discussion of topics in statistics; applications to the above two topics
- Linear algebra algorithms; large systems (sparse matrices of order $n = 10^9$)
- Linear algebra on parallel computers
- **Time permitting:** General relativity; quantum Monte Carlo

1 Back recurrence

The Bessel functions are solutions to a partial differential equation in cylindrical coordinates. They satisfy the following recurrence relation:

$$J_{n+1}(x) = \frac{2n}{x}J_n(x) - J_{n-1}(x)$$

where n is an integer and x is real. Does knowing $J_0(x)$ and $J_1(x)$ allow you to know $J_n(x)$ for $n \geq 2$?

Let us write this relation as a 2 component vector

$$\begin{pmatrix} J_n \\ J_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & \frac{2n}{x} \end{pmatrix} \begin{pmatrix} J_{n-1} \\ J_n \end{pmatrix}$$

What are the eigenvalues of this matrix?

$$\begin{vmatrix} -\lambda & 1 \\ -1 & \frac{2n}{x} - \lambda \end{vmatrix} = 0$$

which leads to

$$\lambda = \frac{n}{x} \pm \sqrt{\frac{n^2}{x^2} - 1}.$$

If $\frac{n}{x} > 1$, then $\lambda_+ > 1$ and $\lambda_- < 1$. Note that if we apply this recurrence relation m times, we will have λ_+ exponentially growing and λ_- exponentially decaying. Note also that the Bessel and Neumann functions are known to exponentially decay and grow respectively. So if we start with a $\tilde{J}_0(x)$ and $\tilde{J}_1(x)$ to some machine precision, we know that these can be written as

$$\begin{aligned}\tilde{J}_0(x) &= \alpha J_0(x) + \beta N_0(x) \\ \tilde{J}_1(x) &= \alpha J_1(x) + \beta N_1(x)\end{aligned}$$

From this, we can determine α and β . As we iterate, however, the Neumann functions will dominate over the Bessel functions - i.e. we have numerical instability, as the Bessel function will get lost in the noise.

If we iterate backwards, however, with some random choice of the precision numbers, the resulting quantity will essentially be $\alpha J_0(x)$, as the growing term will be suppressed by the back-recurrence. We can make an algorithm out of this using

$$1 = J_0(x) + J_2(x) + J_4(x) + \dots$$

which determines the constant α . We are putting complete garbage in... and getting good results out!