Numerical methods/ introduction to scientific computing

Applied Mathematics 111

(Spring 2006)

Instructors: Eli Tziperman, Zhiming Kuang.

TF: Mike Hohensee, <hohensee@fas.harvard.edu>; tel: 617-496-7733; sections are 2 to 3pm on Fridays, in Maxwell-Dworkin 123. TF office hour: 6 to 7pm on Mondays, in Maxwell-Dworkin 319.

Day and time of course: Tue-Thu 10:00-11:30;

Location: Maxwell Dvorkin 125;

Announcements Last updated: November 24, 2015.

Feel free to write, call or visit us with any questions:

Eli Tziperman (eli AT eps.harvard.edu); Zhiming Kuang (kuang at fas.harvard.edu).

Sample Matlab programs: Please see supporting materials directory for all demo programs used in class and some more.

Supporting material: here,

Final project description: here, (please check for updates in the near future).

Homework: HW-01.pdf, HW-02.pdf, HW-03.pdf, HW-04.pdf, HW-05.pdf, HW-06.pdf, HW-07.pdf, HW-08.pdf, HW-09.pdf,

Homework solutions: see course home page.

(What's the point of optional/ extra credit problems: apart from the fun of doing them, they will count instead of homework problems in which you may have missed an answer...)

1 Textbooks:

- (**CM**) *Computing with Matlab* by Cleve Moler, also here.
- The Matlab demos for the *Computing with Matlab* textbook are at http://www.mathworks.com/moler/ncmfilelist.html.
- (**BF**) *Numerical Analysis* (8th edition) by Burden and Faires. see also here.

2 Outline

Many complex physical problems defy simple analytical solutions or even accurate analytical approximations. Scientific computing can address certain of these problems successfully, providing unique insight. This course introduces some of the widely used techniques in scientific computing through examples chosen from physics, chemistry, and biology. The purpose of the course is to introduce methods that are useful in applications and research and give the students hands-on experience with these methods. The course will introduce and use the Matlab software, which allows easy access to sophisticated (and fun...) mathematical and graphics capabilities.

3 Prerequisites

Applied Mathematics 21a and 21b, or Mathematics 21a and 21b or permission of instructor. Knowledge of some programing language would be quite helpful.

3.1 An evolving syllabus

Topics to cover, with a *sample motivation* for each, ordered by Lecture number (L1,L2, etc). This list will evolve and become more specific and detailed during the course. Files mentioned here may be found under the above supporting material link.

- L1: Introduction, overview
- Solving linear equations (**CM** 53-82): *solving for resistor networks, Google's PageRank algorithm.*
  - L2: motivation: resistor network; Permutations and triangular matrices; LU factorization (motivation: BF 388); pivoting, why is pivoting necessary (CM 2.6);
  - L3: LU derivation: (CM 2.5), note comment below, use `linear_eqns01.m`; effects of roundoff errors (CM 2.8); norms and condition numbers (first couple of pages of CM 2.9): vectors norms, matrix norm, equivalent matrix norm definition based on BF eqn 7.2 p 424 and the two preceding lines; corollary 7.10 in BF p 424; eigenvector demo from ncmgui; prove form of $\|A\|_\infty$ using BF theorem 7.11 p 426; note also (MathWorld entry on matrix norms) that $\|A\|_2^2 \leq \|A\|_1\|A\|_\infty$;
  - L4: forms of $\|A\|_\infty$ and $\|A\|_1$; ill conditioned 2×2 numerical example from CM 2.9 (`linear_eqns02.m`); estimating error bounds: theorem 7.27 from BF p 455, condition number definition (BF definition 7.28); PageRank (CM 2.11, although a clearer presentation is given in `Investigating-Googles-PageRank-algorithm.pdf`, p 1-7; the theoretical background, proving that there is a PageRank and that it is unique is the Perron-Frobenius Theorem given in section 6.1 of the file; the power method is explained in BF 9.2 p 557-558.
    `Gill-Prystowsky-PageRank-explained.pdf`);
  - L5: Mention briefly some special matrices (CM 2.10, BF 6.6 398-409): strictly diagonally dominant (BF definition 6.18), symmetric positive definite (BF definition 6.20) and LDL^T and Cholesky (LL^T) factorizations (BF p 404), band matrices (BF definition 6.28), tridiagonal matrices and forward elimination/ backward substitution being simply loops)

  Comment: CM p. 6 says "it turns out that this equation can be rewritten...". I think this refers to the following: Suppose $n=2$, then

  $$U = M_2P_2M_1P_1A = M_2(P_2M_1P_2^T)P_2P_1A$$

  (remember that for the P matrices $P^{-1}=P^T$) so that we have $L_2 = M_2^{-1}$ and $L_1 = (P_2M_1P_2^T)^{-1} = P_2M_1^{-1}P_2^T$. For $n=3$ this is

  $$U = M_3P_3M_2P_2M_1P_1A = M_3(P_3M_2P_3^T)(P_3P_2M_1(P_3P_2)^{-1})(P_3P_2P_1)A$$

so that $L_1 = (P_3 P_2 M_1 P_2^T P_3^T)^{-1}$ etc. As for M_n^{-1}, because of the form of these matrices, their inverse is simply obtained by changing the sign of the terms under the diagonal.

- Interpolation (**CM** 93-110): *finding the total US population at a given date*

  - L5 (continued): Lagrange full degree polynomial interpretation (BF Fig 3.5, p 106), Vandermonde matrix, importance of scaling and shifting and bad conditioning (CM Ex 3.18, `interp01.m`). Oscillatory nature of global polynomials and global sensitivity to local changes (BF Fig 3.13, p 152; polynomial *ncmgui* demo);
  - L6: A final word on polynomial interpolation: if you need to use it, don't use equidistant data; example: Ex 3.9 from CM with the interpolation of $f(x) = 1/(1+25x^2)$ in $-1 \le x \le 1$, show `rungeinterp.m` from the ncm Matlab files; introduce Chebyshev polynomials: $T_n(cos(\theta)) = \cos(n\theta)$ (see much more in wikipedia, copied in supporting material); show Chebyshev (`chebyshev_plot.m`); note that knots at zeros of Chebyshev polynomials results in uniform convergence: show chebyshev zeros (`chebydiag.m`), then demonstrate uniform convergence by running `ginterp.m` and typing in the input box `runge(10)`, `runge(20)` etc for uniformly sampled data, and then `rungec(10)`, `rungec(20)` etc for sampling at Chebyshev zeros. Condition number of Vandermonde matrix for uniform vs Chebyshev sampling (`interp02_HW03.m`). Finally, the Chebyshev sampling also works for functions that are not symmetric about $x = 0$, see `runge_asymmetric.m`. Links: wikipedia entries on "Lebesgue_constant_(interpolation)". Piecewise linear interpolation and discontinuous derivatives at breakpoints (CM 3.2). To get a more continuous behavior at knots: piecewise cubic Hermit/ osculatory (kissing) interpolation (CM 3.3). Shape-preserving piecewise cubic (CM 3.4).
  - L7: Cubic splines, what's a spline, tridiagonal matrix equations (CM 3.5); the only deviation from CM is the use of "natural" or "free" boundary conditions on edges: $P''(x_0+) = P''(x_n-) = 0$ on first page of CM 3.5; this leads to a diagonally dominant matrix equation: $2d_0 + d_1 = 3\delta_k$ and $2d_n + d_{n-1} = 3\delta_{n-1}$; note that BF write the tridiagonal equation for the polynomial coefficients rather than for the slopes d_j as in CM). Mention briefly: parametric curves (BF 3.5 p 157-8).

- Zero finding, solution of nonlinear equations (**CM** 117-135): *everything's nonlinear! Finding birth rate of a population; fractals and chaos*

  - L7 (continued): motivational example: population dynamics, logistic map, finding birth rate (BF 2, p 45-46). Bisection and the calculation of $\sqrt{2}$ (CM 4.1).
  - L8: Newton method (CM 4.2). Convergence rate in Newton's method: let $x_{i+1} = x_i - f(x_i)/f'(x_i)$; subtract the actual solution x from both sides to find $\varepsilon_{i+1} = \varepsilon_i - f(x_i)/f'(x_i)$ where $\varepsilon_i \equiv x_i - x$; expand $f(x_i)$ and derivative in Taylor series around x, noticing that $f(x) = 0$:

$$\begin{aligned}\varepsilon_{i+1} &= \varepsilon_i - \frac{f(x)+f'(x)\varepsilon_i}{f'(x)+f''(x)\varepsilon_i} = \varepsilon_i\left(1 - \frac{f'(x)}{f'(x)+f''(x)\varepsilon_i}\right) \\ &= \varepsilon_i\left(\frac{f''(x)\varepsilon_i}{f'(x)+f''(x)\varepsilon_i}\right) \approx \varepsilon_i^2\left(\frac{f''(x)}{f'(x)}\right)\end{aligned}$$

Quadratic vs linear convergence (BF table 2.7, or `zero_finding01.m`); Pathology: when $f'(x) = 0$ we may get slower convergence; when $f'(x) = \infty$ Newton iterations may not converge: a limit cycle of Newton iterates (CM 4.3). **A couple of digressions:** First, Newton iterates and fractal attraction basins, see here, or in wikipedia, also in supporting materials. Note sensitivity of Newton iterates to initial conditions when not close to final root! Define fractals and give two examples: Cantor set and Koch triangle; fractal box-dimension in general and for these two examples; (Strogatz, "Nonlinear dynamics and chaos", 10.2 p 353-357, available in supporting materials directory).

- L9: Secant method (CM 4.4); add this: given $e_{n+1} = e_n e_{n-1}$, assume $e_{n+1} = e_n^\alpha$, then $e_{n+1} = e_n^\alpha = e_n e_{n-1}$ so that $e_{n-1} = e_n^{\alpha-1}$; at the same time we have $e_n = e_{n-1}^\alpha$, or $e_{n-1} = e_n^{1/\alpha}$; therefore $e_{n-1} = e_n^{\alpha-1} = e_n^{1/\alpha}$, or $1/\alpha = \alpha - 1$ so that $\alpha = (1+\sqrt{5})/2 = \phi$, the golden ratio. Practical zero finding routines typically combine Newton or secant method with bisection, depending on behavior of the iterates (CM 4.6). Second digression: chaos in the logistic map: from the general form of these iterative methods, $x_{n+1} = g(x_n)$, proceed to the logistic map $x_{n+1} = rx_n(1-x_n)$ and population dynamics/ carrying capacity; fixed points, periodic behavior, period doubling and chaos, orbit diagram; sensitivity to initial conditions and weather prediction; demonstrate using `logistic_map.m`, (Strogatz, "Nonlinear dynamics and chaos", 11.2 p 401-402, 11.3 p 404-405, 11.4 p 409-410, available in supporting materials directory).
- L10: Back to subject: Optimization: golden section search, aided by fitting a parabola (CM 4.10). Mention briefly: finding zeros of polynomials requires specialized methods, much more appropriate than the general ones above (BF 2.6 starting at p 87).

- Least squares (**CM** 53-82): *predicting the stock market?*
  - L11: Motivation: stock/ oil prices; uncorrelated white noise random numbers, 1st order Markov process, autocorrelation of a time series, or a white noise, of a Markov process (least-squares-notes.pdf with scanned hand written notes in supporting materials)
  - L12: geometric Brownian motion (`plot_GBM.m`; reducing the stock price prediction to a least-squares problem for a few coefficients in a Markov process (above hand written notes). Another motivation: fitting a set of data points with a low-order polynomials (BF 8.1 p 482-484). Solving least-square problems: the textbook solution which is the wrong way to go (CM 5.5 first page, and above notes).
  - L13: the right way to solve least squares: Householder reflections (CM 5.4 and above notes); QR factorization (CM 5.5, above notes, `least_squares01.m`);
  - L14: comparison of wrong and right ways to solve a least squares problem for both well and ill conditioned cases (`least_squares01.m`); Pseudo-inverse (CM 5.6); rank deficient problems and the non unique basic solution vs the unique minimum norm solution (CM 5.7);

- Numerical Integration (quadrature, **CM** 53-82); *building corrugated roofs; global warming, CO_2 emission rate and the missing sink*

- L15: Motivation: building roofs (BF introduction to Chapter 4, p 167). More motivation: global warming, CO_2 emissions and the missing sink! (numerical-integration-missing-co2-sink.ppt)
- L16: adaptive quadrature (CM 6.1), basics: trapezoid, mid-point and Simpson rules (CM 6.2 and hand written integration-notes.pdf), error bounds and adaptive function evaluations (CM 6.2 and `quadgui.m` from the textbook demos) integrating discrete data (CM 6.6).

- Solving ordinary differential equations (**CM** 53-82): *chaos and the butterfly effect*
  - L17: introduction to initial value problems (ivp) with ode: Euler's method, forward and backward; explicit vs implicit methods; stability, accuracy; (Mike's notes); Scripts used in class, available in supporting material directory here: `euler_method.m`, `euler_accuracy.m`, `euler_stability.m`,
  - L18: Taylor methods, multi-stage single-step methods (Mike's notes).
  - L19: error control (adaptive step size); multi-step methods; comparison of different methods available in Matlab: `ode23, ode45, ode113`. (Also, Mike's notes). Scripts used in class, available in supporting material directory here: `single_pendulum.m, lorenz.m, euler_stiff.m, back_euler_stiff.m`.
  - L20: stiff problems, definition of stiffness (CM 7.9); ode solvers for stiff problems in Matlab: `ode23s, ode15s`; boundary value problems (BVPs): shooting methods (BF 11.2), finite difference methods (BF 11.3), finite difference methods for nonlinear problems, requiring iterations (BF 11.4). (Also, Mike's notes). Scripts used in class, available in supporting material directory here: `Schrodinger.m, bvp.m`.

- Eigenvalues and singular values (**CM** 269-298): *love affairs*
  - L21-22: *Motivation (1):* love affairs; Romeo and Juliet and various combinations of love/ hate relationships. how to navigate the beginning of your relationship in order to guarantee explosive lasting love: 2d phase plane, fixed points, stability of fixed points, stable node, unstable saddle, eigenvalues and eigenvectors, all used to analyze relationships (Strogatz 5.3, pp 138-140, and exercises 5.3.1-5.3.6, page 144). Note role of eigenvectors separating different attraction basins. *Motivation (2):* string vibrations (Derivation-of-the-wave-equation.pdf and then BF, introduction to chapter 9, p 547).
  - L23: eigenvalue and singular value decompositions (CM 10.1); example (see comment 1 below) and Matlab gui demo `eigshow.m` of both eigenvectors and SVD (CM 10.2, 10.3). See more on SVD in wikipedia. Characteristic polynomial and why it is not the right way to calculate eigenvalues (CM 10.4). Symmetric and Hermitian matrices (CM 10.5).
  - L24 Eigenvalues sensitivity and accuracy (CM 10.6). Demos in `my_eigenvalues_demos.m` in supporting materials. Singular value sensitivity and accuracy (CM 10.7). Singular values and L_2 matrix norm which we still owe from the linear equations chapter (comment 2 below). SVD and generalized inverse, SVD minimum norm solution to an under-determined least squares problem (`SVD.pdf` in supporting materials from here).

• *Comment 1:* CM 10.2 p 272 shows the characteristic equation for singular values. It can be derived directly from the definition of singular values/ vectors: $Av = \sigma u$, $A^H u = \sigma v$; multiply the first equation by A^H on the left, $A^H Av = \sigma A^H u = \sigma^2 v$, which implies $(A^H A - I\sigma^2)v = 0$, which provides a characteristic equation for the singular values: $det(A^H A - I\sigma^2) = 0$.
• *Comment 2:* the matrix norm under L_2 is the maximum of $\|Ax\|$ subject to $\|x\| = 1$. Use Lagrange multiplier λ to write this maximization problem as

$$\max_x((Ax)^T(Ax) + \lambda(x^T x - 1)) = \max_x(x^T A^T Ax + \lambda(x^T x - 1)).$$

at the max point, derivative with respect to x is zero, leading to $A^T Ax + \lambda x = 0$, so that λ is simply the squared of the largest singular value and the vector maximizing the above expression is the corresponding singular vector. We showed previously that the matrix norm is $\|Ax\|/\|x\| = \sqrt{x^T A^T Ax/x^T x} = \sqrt{\lambda}$, so that the norm is the largest singular value. (**CM** 307-319): *pollution dispersion in groundwater or ocean*

4 Course requirements

Homework will be given throughout the course. The best 80% of the assignments will constitute 50% of the final grade. A final project will constitute another 50%.