Review list for CS542G fall 2007

Themes
- Taylor series expansions (or at least linearization)
- Solve a big problem approximately by restriction to a smaller subspace (pick a basis)
- Use simpler model problems to analyze, debug, understand
- Reduce to core operations, use well respected libraries

Topics

<u>floating point</u>
- absolute vs. relative error
- cancellation error
- numerically unstable methods (vs. well posed problems)

<u>interpolation</u>
- polynomials, piecewise polynomials / splines, problems extending to higher dimensions
- radial basis functions - derivation as an optimally smooth interpolant, reduction to a linear system, pros and cons
- moving least squares - how to derive it, reduction to linear least squares, pros and cons

<u>dense linear algebra</u>
- vector and matrix norms
- know roughly what BLAS and LAPACK cover
- condition number - how it's derived from backwards error analysis
- 2-norm condition number - know that it's the ratio of max over min eigenvalues for a symmetric matrix
- how to solve a linear system with LU factorization
- how to derive different variations of LU (up-looking, left-looking, block versions)
- what the Cholesky factorization is
- row partial pivoting
- least squares from the normal equations - pros and cons
- QR - how to use it for least squares, Modified Gram-Schmidt and Householder, pros and cons
- weighted least squares
- PCA and the SVD: low rank approximation property (though not the derivation)
- SVD: connection to eigenproblems, "cool things about the SVD" - e.g. 2-norm, rank, pseudo-inverse
- symmetric eigenproblem: power method (why and how fast it converges)
- symmetric eigenproblem: shift-and-invert (why it's faster, if you can solve the linear systems)
- symmetric eigenproblem: orthogonal iteration (why it works, implementation with QR)
- symmetric eigenproblem: Rayleigh-Ritz

optimization
- nonlinear optimization: convexity, difficulty of knowing when converged
- steepest descent: the algorithm, line search, convergence problems
- Newton: derivation as solving a model (quadratic) problem via the Hessian, pros and cons
- nonlinear least squares: Gauss-Newton (pros and cons relative to full Newton)

time integration
- reduction of second order problems to first order systems
- truncation (local) error vs. global error
- test equation - how to do a linear stability analysis
- Forward Euler: algorithm, accuracy, stability
- Adams-Bashforth: underlying idea of explicit multistep method, how to derive an AB method, pros and cons
- Runge-Kutta: know roughly what they do (no need to memorize specific schemes), pros and cons
- techniques for validating (estimating the error of) time integration - exact solutions, real data, running with different dt
- stiffness: what is it, why is it a problem for some methods
- implicit vs. explicit
- Backwards Euler: algorithm, accuracy, stability (pros and cons)
- Trapezoidal Rule: algorithm, accuracy, stability (pros and cons - e.g. monotonicity problems)
- know of the BDF family, why/where it's useful
- Verlet - directly tackling F=ma

Nonlinear solvers
- in 1D: bisection, secant, Newton
- Newton in higher dimensions: derivation, potential problems

Barnes-Hut
- derivation of the algorithm
- know of Fast Multipole Method, application to RBF's and other problems with radial kernel functions, without getting into any details

PDE's
- what the Poisson equation and the Laplace equation are, in 1D and higher dimensions
- finite difference approach
- boundary conditions, implementing for finite differences (e.g. ghost values)
- what a "stencil" is
- pros and cons of finite differences in more than one dimension
- Galerkin FEM: derivation for Poisson problem (ignoring boundary conditions)
- FEM spaces: piecewise linear nodal basis functions, notion of generalizing to higher degree polynomials etc.
- mass matrix, lumped mass matrix (why?)
- assembling global stiffness matrix from local stiffness matrices
- basic idea of quadrature - midpoint formula
- notion (without proof) that Galerkin FEM is finding "best" solution from subspace

mesh generation
- basic idea behind four strategies (multiblock, advancing front, tile-based, Delaunay) any why we focused on Delaunay
- Delaunay: empty circumcircle property, O(n log n) cost, maximizes minimum angle, idea of edge flipping
- Predicates: idea of how to control floating-point rounding errors in geometric computations
- Delaunay refinement - for quality, and to make mesh adaptive to errors in PDE

solving sparse linear systems
- Jacobi, Gauss-Seidel, SOR
- analyzing performance on model problem (finite differences) with complex exponentials (guess for eigenvectors)
- Conjugate Gradient: global optimality over Krylov space property, but only needs O(n) storage and O(n) work per iteration
- Idea of preconditioners
- direct solvers: fill, the graph model, idea of fill-reducing orderings, the gist of Minimum Degree and Nested Dissection, high efficiency through use of BLAS/LAPACK