Other topics in brief:

Finite Difference Methods for Nonlinear BVPs

- Consider $y'' = f(x, y, y')$ on $[a, b]$ with $y(a) = \alpha$ and $y(b) = \beta$.
- Then $\frac{y(x_{i+1})-2y(x_i)+y(x_{i-1})}{h^2} \approx f(x_i, y(x_i), \frac{y(x_{i+1})-y(x_{i-1})}{2h})$
- Guess a solution: $y(x_0), \ldots, y(x_n)$ at all points.
- Bring all terms to the left and you get a system of nonlinear equations. It can be solved using Newton's Method for systems.
- The Jacobian is tridiagonal and there is a unique solution for $h < \frac{2}{L}$ where $L = max_{x,y,y'}|f_{y'}(x, y, y')|$.

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - J^{-1}(\mathbf{w}^{(k)})F(\mathbf{w}^{(k)})$$

Eigenvalue Problems – from ODEs

These problems arise in coupled mass-spring systems and in beam deflection problems as well as in other applications, especially from using separation of variables on partial differential equations.

For example, consider $y'' + p^2 y = 0$ with $y(0) = y(L) = 0$.

This ODE, of course, has the trivial solution $y \equiv 0$, but we would like to know when there are other solutions. Physically, that may correspond to buckling of the beam.

We can write the system using finite differences. There will only be non-trivial solutions of the linear system of equations that arises if the determinant of the matrix is 0. The eigenvalues depend on p. We can find the values of p for the system in which there is 1 equation in 1 unknown, then change the step size to make it 2 equations in 2 unknowns, etc. As h decreases and the size of the system increases, our approximations to the eigenvalues gets better and better. The true eigenvalues p^2 of the original ODE are $\left(\frac{np}{L}\right)^2$ for $n = 1, 2, \ldots$. There are an infinite number of eigenvalues.

Approximating Eigenvalues

Gershgorin's Theorem - Every eigenvalue of a matrix $\mathbf{A}$ is in at least one of the circles $C_1, C_2, \ldots, C_n$ where C_i has its center at the diagonal entry a_{ii} and its radius $r_i = \sum_{j \neq i} |a_{ij}|$ is the sum of the absolute values along the rest of the row. This fact can be used to estimate some of the eigenvalues and make statements about their magnitude. For example, the proof of why Jacobi's method always converges for diagonally dominant matrices becomes simple using Gershgorin's Theorem.

Power Method (see part 2 of notes) and Inverse Power Method - to find the eigenvalue closest to a given value. The Inverse Power Method finds the smallest eigenvalue (in magnitude). Rather than making a guess $\mathbf{x_0}$ and computing $\mathbf{x_1} = \mathbf{Ax_0}/\|\mathbf{Ax_0}\|_\infty$ and so forth, we guess $\mathbf{x_0}$ and solve $\mathbf{Ax_1} = \mathbf{x_0}$ for $\mathbf{x_1}$ and then scale $\mathbf{x_1}$ by its entry of largest magnitude. Convergence is linear (with coefficient $\frac{\lambda_n}{\lambda_{n-1}}$ and we can speed it up using Aitken's Method).

The Inverse Power Method may also be used to find the eigenvalue closest to a given value. This simply involves applying the Inverse Power Method on the matrix $\mathbf{A} - r\mathbf{I}$ to find the eigenvalue closest to r.

Annihilation - Once λ_1 is known, applying the Power Method with $\mathbf{A} - \lambda_1\mathbf{I}$ can be used to find an eigenvalue.
$\mathbf{x^{(0)}} = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$
$\mathbf{Ax^{(0)}} = \alpha_1\lambda_1 v_1 + \alpha_2\lambda_2 v_2 + \ldots + \alpha_n\lambda_n v_n$
$(\mathbf{A} - \lambda_\mathbf{1}\mathbf{I})\mathbf{x^{(0)}} = \alpha_2\lambda_2 v_2 + \ldots + \alpha_n\lambda_n v_n$
This will give a new eigenvalue, call it μ_2 of $(\mathbf{A} - \lambda_1 I)$. Then, there is an eigenvalue $\lambda_2 = \mu_2 + \lambda_1$. Another eigenvalue can then be found by applying the Power Method with $(\mathbf{A} - \lambda_1 I)(\mathbf{A} - \lambda_2 I)$ and so on. The round-off errors may build up and cause the method to become unstable.

Deflation - The idea of deflation is that once one eigenvalue and eigenvector are known, this information can be used to find another eigenvector.

- *Hotelling's method for symmetric matrices.* If λ_1 is the largest eigenvalue of $\mathbf{A}$ and $\mathbf{v_1}$ is the corresponding unit eigenvector, then $\mathbf{B} = \mathbf{A} - \lambda_1\mathbf{v_1}\mathbf{v_1}^T$ has eigenvalues $0, \lambda_2, \ldots, \lambda_n$ with the same eigenvectors as $\mathbf{A}$. Here, round-off error again can build up so only a few eigenvalues can be expected to be found accurately.
- *Wielandt deflation.* Let $\mathbf{B} = \mathbf{A} - \lambda_1\mathbf{v_1}\mathbf{x}^T$ where $\mathbf{x}$ is related to $\mathbf{A}$ and λ_1.

Householder's method- Using orthogonal matrices, the original matrix is transformed

to turn symmetric matrices into tridiagonal matrices and to turn non-symmetric matrices into Upper Hessenberg (upper triangular but with just one subdiagonal) matrices, without changing the eigenvalues.

QR algorithm- Using rotations on symmetric tridiagonal matrices, the QR algorithm eliminates the off diagonal elements and yields the eigenvalues. The QR algortihm turns Upper Hessenberg matrices into upper triangular.

Continuous Function Approximation

Suppose we want to approximate the function $f(x)$ by a polynomial. We can apply the Least Squares idea in order to minimize $\int_a^b [f(x) - P_n(x)]^2 dx$. If we have reason to believe $f(x)$ can be well approximated by a polynomial function of a certain form, we might have success using this approach.

Just as with polynomial interpolation, the computation gets messy and unstable. The Hilbert matrix arises. In addition, (as in polynomial interpolation) the degree k results don't help us compute the degree $k+1$ results. Orthogonal polynomials can be used instead of fitting an arbitrary polynomial. The best type to use may depend on the problem at hand. Some examples of orthogonal polynomials are Legendre polynomials, Chebyshev polynomials, and Laguerre polynomials. (Some of these have weight functions). Fourier Series, Pade approximants and Fast Fourier Transforms are other related topics.

Optimization

Mostly from Chapra

Applications:

- Designing aircraft for minimum weight or maximum strength
- Space vehicle trajectory
- Minimum cost for building
- Analyzing structural behavior through minimizing potential energy
- Finding the shortest sales route

Methods for One Variable Problems

- Our purpose is to find the minimum or maximum of a continuous curve. At this point, if the curve is smooth, the first derivative is 0. If we can take the derivative of the function, we can apply the root-finding methods we've discussed earlier.
- **Golden Section Search** This method can be applied when we know that the desired maximum or minimum lies between two points x_a and x_b where $l_0 = x_b - x_a$.
  - We choose a point in between, x_c, such that $\frac{l_1}{l_0} = \frac{l_2}{l_1}$ where $l_1 = x_c - x_a$ and $l_2 = x_b - x_c$.
  - Since $l_0 = l_1 + l_2$, we find $\frac{l_1}{l_0} = \frac{\sqrt{5}-1}{2} = 0.61803...$ or the golden ratio.
  - We do the same thing from the other end of the interval to find a fourth point, x_d.
  - Now, $x_a < x_d < x_c < x_b$.
  - From the function values at these points, we can tell whether the maximum or minimum must lie on the interval $[x_a, x_c]$ or $[x_d, x_b]$.
  - We use this new interval and repeat the process.
  - The advantage is that one of the discarded points (x_c or x_d) is used in the next step since it will work out to be a distance l_1 from the end of the new interval.
- **Quadratic Interpolation** If we know the maximum or minimum is bracketed, we can choose a point in the middle and fit a quadratic through the 3 points. We can find the maximum or minimum of the quadratic to use as our new guess, eliminating one of the previous endpoints. We repeat the procedure and note that convergence may be slow and one sided for certain functions, in a similar way to the one sided convergence that sometimes occurs with Regula Falsi.
- **Newton's method on $f'(x)$.** As noted in the introduction to this section, this is an obvious choice. Note that in order to apply this idea, $f'(x)$ should be easy to compute. The method may diverge. It should be most useful if you are already close to the root.

Multi-Variable Problems

- *Random Search* - This brute force method should get you close to the maximum or minimum on the region. It should even work for functions that are discontinuous.

However, the convergence is very slow. Simply divide each dimension into, say, n pieces and evaluate the function at each of the n^p grid values, for a function of p variables.

- *Univariate search* - Vary one variable at a time to find a maximum or minimum, while holding the other variables fixed. This turns the problem into a sequence of one-dimensional searches. Viewing the progress with contour plots adds insight into the method. It is better to find more appropriate directions, *pattern directions*, in which to search, rather than going in the x direction, then the y direction, then the z direction, then the x again etc. One way to do this is called Powell's Method. Here, by choosing several directions and searching along them, one can choose better new directions in which to search. It is an efficient method which is quadratically convergent and requires no derivatives to be taken.

- *Gradient Search / Steepest Descent* - Search in the direction of ∇f since this is the direction of maximum increase. (The negative of this direction is the direction of maximum decrease). Determine how far to go in the desired direction by considering $f(x_0 + f_x h, y_0 + f_y h)$ and finding h using one dimensional methods. Then repeat in a new direction using the value of ∇f at the new point. This method converges quickly since the directions of greatest decrease are considered first. *Conjugate Gradient* is a variant of Gradient search in which each new direction should be orthogonal to each previous direction.

- *Newton's Method for Multiple Variables* - Writing out a Taylor series through second order for functions of more than one variable gives

$$f(x) \sim f(x_i) + \nabla f^T(x_i)(x - x_i) + \frac{1}{2}(x - x_i)^T H_i (x - x_i)$$

  where H_i is the Hessian matrix (of second partial derivatives of f). At the minimum, x_*, $\frac{\partial f(x_*)}{\partial x_j} = 0$ for $j = 1, 2, ..., n$. Thus,

$$\nabla f(x_*) = \nabla f(x_i) + H_i(x_* - x_i) = 0$$

  and if H_i is nonsingular, we find $x_{i+1} = x_i - H_i^{-1} \nabla f(x_i)$

- *Marquardt's Method* combines Steepest Descent and Newton's Method.

- *Quasi-Newton Methods* use approximate Hessians, H, without using second derivatives. Quadratic convergence is lost, but convergence is still superlinear. One example is Broyden's Method, another is BFGS.

Linear Programming

For linear optimization problems, the maximum must occur at a corner of the feasible region.

- Graphical method - Plot the constraints. Find the corners. Plug into the objective function. Take the maximum or minimum value as desired.
- Simplex Method - Move from corner to corner in an intelligent way while staying feasible.

Partial Differential Equations

Elliptic PDEs

Applications:

- Temperature distribution in a heated plate
- Electric field

Poisson's Equation: 2-D steady state problems:

$$u_{xx} + u_{yy} = 0 \text{ with } u(x,y) = g(x,y) \text{ onthe boundary}$$

- Set up grid lines (for a rectangular region) and use finite differences.
- We obtain a band matrix with width $\approx n$ and n^2 points (for a region with n grid points in each direction, an $n \times n$ grid).
- Gauss-Seidel or SOR are often used in solving the linear system of equations.
- It is more difficult for irregular shaped regions or complicated boundary conditions, such as having part of the rod insulated and part not insulated.
- Another numerical method for solving such problems is called Control Volumes. The method is similar to finite differences. Around each grid point a rectangle is drawn and the flux through each boundary is considered. This leads again to a system of linear equations. One can discretize non-uniformly for irregular regions but it still can get messy. One can also refine the grid near complicated boundaries.

Parabolic PDEs

Application: Heat equation $u_t = ku_{xx}$

Explicit approach: Discretize the PDE to obtain the scheme

$$T_i^{j+1} = T_i^j + \frac{k\Delta t}{(\Delta x)^2}[T_{i+1}^j - 2T_i^j + T_{i-1}^j]$$

where the subscripts denote the spatial step and the superscripts denote the time step.

- For stability, we need $\Delta t \leq \frac{(\Delta x)^2}{2k}$.
- To avoid oscillatory behavior, we need $\Delta t \leq \frac{(\Delta x)^2}{4k}$.
- Halving Δx requires reducing the time step by a factor of 4. So the amount of work goes up very quickly when reducing the mesh spacing. (Work increases by a factor of 8 to get to the same end time when the mesh spacing is halved).
- For 2 dimensions, the explicit approach requires $\Delta t \leq \frac{1}{8k}[(\Delta x)^2 + (\Delta y)^2]$ for stability.

Implicit approach: In order to avoid the stability issues of the explicit approach, implicit approaches are considered. Once again, we discretize the PDE, this time obtaining the scheme

$$T_i^{j+1} = T_i^j + \frac{k\Delta t}{(\Delta x)^2}[T_{i+1}^{j+1} - 2T_i^{j+1} + T_{i-1}^{j+1}]$$

- This leads to a tridiagonal system of linear equations.
- The method is stable and convergent.
- The time discretization is only first order, Δt, while the step discretization has second order error $(\Delta x)^2$.
- This method can also be used for finding steady-state behavior since the intermediate time values are not needed.
- To achieve second order error in both space and time, we average the explicit and implicit approaches to obtain the **Crank-Nicolson method**.

$$T_i^{j+1} = T_i^j + \frac{k\Delta t}{2(\Delta x)^2}[T_{i+1}^j - 2T_i^j + T_{i-1}^j + T_{i+1}^{j+1} - 2T_i^{j+1} + T_{i-1}^{j+1}]$$

- For the 2 dimensional heat equation, we can apply the **ADI** or Alternating Direction Implicit method, we break the PDE into two steps (each with time step $\frac{\Delta t}{2}$) to turn it into 2 one-dimensional problems

$$\begin{aligned} T_{i,j}^{l+1/2} &= T_{i,j}^{l} + \frac{k\Delta t}{2(\Delta x)^2}[T_{i+1,j}^{l} - 2T_{i,j}^{l} + T_{i-1,j}^{l}] + \frac{k\Delta t}{(\Delta y)^2}[T_{i,j+1}^{l+1/2} - 2T_{i,j}^{l+1/2} + T_{i,j-1}^{l+1/2}] \\ T_{i,j}^{l} &= T_{i,j}^{l+1/2} + \frac{k\Delta t}{2(\Delta x)^2}[T_{i+1,j}^{l+1} - 2T_{i,j}^{l+1} + T_{i-1,j}^{l+1}] + \frac{k\Delta t}{(\Delta y)^2}[T_{i,j+1}^{l+1/2} - 2T_{i,j}^{l+1/2} + T_{i,j-1}^{l+1/2}] \end{aligned}$$

  These are equivalent to solving (by discretizing), the one dimensional PDEs

$$\begin{aligned} T_t &= k(T_{xx}^{j} + T_{yy}^{j+1}) \quad \text{For the first half of the time step} \\ T_t &= k(T_{xx}^{j+1} + T_{yy}^{j}) \quad \text{For the second half of the time step} \end{aligned}$$

  The result is 2 tridaigonal systems.

Finite Elements

Finite difference methods are difficult for irregular geometry. The Finite Element method uses simple functions (often piecewise linear or bilinear) as basis functions and solves the PDEs using a variational formulation to find the sum of the basis functions that best approximates the solution to the PDE.