Non-Linear Problems

Systems of Nonlinear Equations

- Given a function $F : R^n \to R^n$, find a point x such that

$$F(x) = 0 .$$

- The 1-D case is covered in undergraduate classes.
- The best software for this problem is some variant on Richard Brent's zeroin, available in Netlib. In Matlab, it is called fzero.
- Note: Solving nonlinear equations is a close kin to solving optimization problems.
  - Easier than optimization, since a "local solution" is just fine.
  - Harder than optimization, since there is no natural merit function f(x) to measure progress.

Special case: Polynomial systems

- If F is a polynomial in the variables x, use special purpose software
- Enables finding all solutions reliably
- Example of a polynomial system:

$$x^2y^3 + xy = 2$$

$$2xy^2 + x^2y + xy = 0$$

  - Traub algorithm
  - Homotopy methods specialized for polynomials

Newton's method

- Instead of the Hessian matrix, we have the Jacobian matrix of first derivatives:

$$J_{ik} = \partial F_i / \partial x_k$$

- Matrix is generally not symmetric
- Use Newton's method

$$F_i(x_k + \alpha p_k) = F_i(x_k) + \alpha\, J_{ik}\, p_k + O(\alpha^2)$$

- However line searches are more difficult to guide, since we can't measure progress using a scalar function $f(x)$.
- Some attempts have been made to use $||\mathbf{F}(\mathbf{x})||$ as a merit function
- However there are difficulties with this approach.
- Convergence will not be quadratic in general

Newton's method

- Derivation: By Taylor series,

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) + \mathcal{O}(\|\mathbf{x} - \mathbf{x}_k\|^2).$$

- For $\mathbf{x} = \mathbf{x}^*$, also $\mathbf{f}(\mathbf{x}) = \mathbf{0}$.
- Neglect nonlinear term and define method by

$$\mathbf{0} = \mathbf{f}(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k).$$

- This is conceptually identical to the procedure (Chapter 3) for one function in one variable.
- Algorithm: Given an initial guess $\mathbf{x}_0$;
  for $k = 0, 1, \ldots,$ until convergence

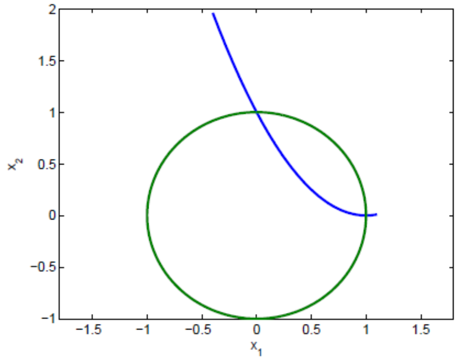
$$\text{solve } J(\mathbf{x}_k)\mathbf{p} = -\mathbf{f}(\mathbf{x}_k),$$

$$\text{set } \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}.$$

  end

Example: a parabola meets a circle

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \mathbf{f}(\mathbf{x}) = \begin{pmatrix} x_1^2 - 2x_1 - x_2 + 1 \\ x_1^2 + x_2^2 - 1 \end{pmatrix}.$$



Two solutions: $(0,1)^T$ and $(1,0)^T$.

$$J(\mathbf{x}) = \begin{pmatrix} 2x_1 - 2 & -1 \\ 2x_1 & 2x_2 \end{pmatrix}$$

1. Starting at $\mathbf{x}_0 = (0,0)^T$ is bad because $J(\mathbf{x}_0)$ is singular!
2. Starting at $\mathbf{x}_0 = (1,1)^T$ obtain root $(0,1)^T$ in 5 iterations. Observe quadratic convergence.
3. Starting at $\mathbf{x}_0 = (-1,1)^T$ obtain root $(1,0)^T$ in 5 iterations. Observe quadratic convergence.

Example: two-point boundary value ordinary differential equation

- Consider the differential problem

$$u''(t) + e^{u(t)} = 0, \quad 0 < t < 1,$$
$$u(0) = u(1) = 0.$$

- Discretize on a uniform mesh (grid) $t_i = ih$, $i = 0, 1, \ldots, n+1$, where $(n+1)h = 1$:

$$\frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} + e^{v_i} = 0, \qquad i = 1, 2, \ldots, n.$$
$$v_0 = v_{n+1} = 0.$$

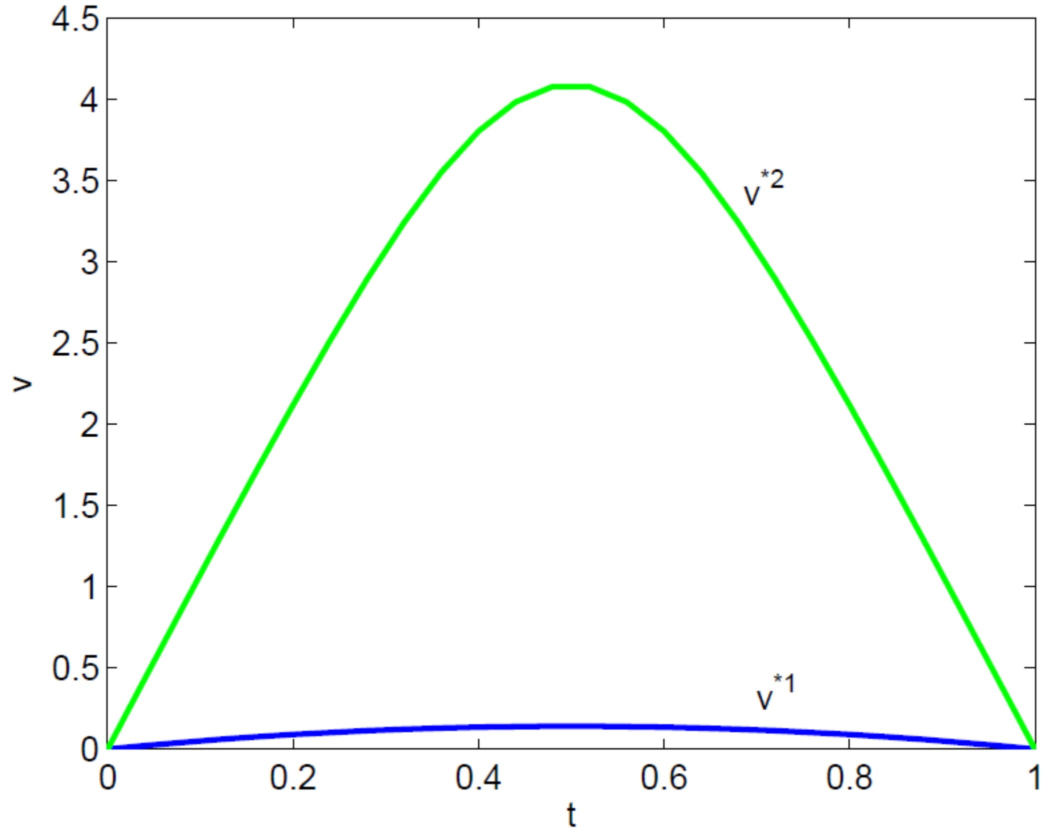
- This is a system of nonlinear equations, with $\mathbf{x} \leftarrow \mathbf{v}$ and $f_i(\mathbf{v}) = \frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} + e^{v_i}$.

- Jacobian matrix is possibly large but tridiagonal

$$J = \frac{1}{h^2}\begin{pmatrix} -2+h^2e^{v_1} & 1 & & & \\ 1 & -2+h^2e^{v_2} & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2+h^2e^{v_{n-1}} & 1 \\ & & & 1 & -2+h^2e^{v_n} \end{pmatrix}.$$

- Initial guess $\mathbf{v}_0 = \alpha \left(t_1(1-t_1), \ldots, t_n(1-t_n)\right)^T$.
- Take various values of α and see what happens, setting `tol` = 1.e-8, $n = 24$:
  1. $\alpha = 0 \;\Rightarrow$ converges in 4 iterations
  2. $\alpha = 10 \;\Rightarrow$ converges in 6 iterations
  3. $\alpha = 20 \;\Rightarrow$ converges in 6 iterations to another solution
  4. $\alpha = 50 \;\Rightarrow$ diverges

Two solutions



Non linear least squares

- Can we use the mechanism of unconstrained optimization?
- Form a scalar function $||\mathbf{F}(x)||^2$
- Then we can use optimization algorithms for this cost function
- Easily generalizes to case where we have overdetermined systems
- Gradient: $\mathbf{g}(\mathbf{x}) = 2\mathbf{J}^t\mathbf{F}(x)$
- Hessian: $\mathbf{H}(\mathbf{x}) = 2\mathbf{J}^t \partial F_i / \partial x_k + \mathbf{Z}$
- **Complicated term:** **Z**=Sum of matrix products involving 2^{nd} derivatives
- Ignoring the Z term and solving system is called "Gauss Newton"

Newton-Like Methods

- Return to original system

$$J_{ik}\, p_k = -F_i(x_k)$$

- **Finite Difference Newton method**: If J is not available, approximate it using finite differences. not recommended
- **Inexact Newton method**: Instead of solving the linear system $J(x(k))p(k) = -F(x(k))$ exactly, use an iterative method to obtain an approximate solution.
- Usual choice of iterative method is GMRES,
- matrix-vector products are evaluated by differencing F in the guess direction

Quasi-Newton methods:

- If storage is not a problem, store and update an approximation to J.
- The usual formula is Broyden's method:

$$\mathsf{B}^{(k+1)} = \mathsf{B}^{(k)} + \frac{(\mathsf{y} - \mathsf{B}^{(k)}\mathsf{s})\mathsf{s}^T}{\mathsf{s}^T\mathsf{s}}$$

where $\mathsf{y} = \mathsf{F}(\mathsf{x}^{(k+1)}) - \mathsf{F}(\mathsf{x}^{(k)})$ and $\mathsf{s} = \mathsf{x}^{(k+1)} - \mathsf{x}^{(k)}$. Then

$$\mathsf{B}^{(k+1)}\mathsf{s} = \mathsf{y} \quad \text{Secant condition}$$

and if $\mathsf{s}^T\mathsf{v} = 0$, then

$$\mathsf{B}^{(k+1)}\mathsf{v} = \mathsf{B}^{(k)}\mathsf{v}.$$

- superlinear convergence if the direction is (asymptotically) close enough to the Newton direction

Continuation methods

- Want to solve $\mathbf{F}(\mathbf{x}) = 0$.
- Know the solution to some easy problem $\mathbf{F}_a(\mathbf{x}) = 0$.
- Example: $\mathbf{F}_a(\mathbf{x}) = \mathbf{x} - \mathbf{a}$
  for some constant vector **a**.
- Formulate the problem
- $\rho_a(\lambda, \mathbf{x}) = \lambda\, \mathbf{F}(\mathbf{x}) + (1 - \lambda)\mathbf{F}_a(\mathbf{x})$
- where λ is a real number in the interval [0, 1].
- The solution to $\rho_a(0, \mathbf{x}) = 0$ is known (for our example, it is just **a**),
- The solution to $\rho_a\,(1, x) = 0$ is the solution to our desired problem.

Algorithm

- Initialize $\lambda = 0$ and $\mathbf{x}$ = solution to $\mathbf{F}_a(\mathbf{x}) = 0$.
- While $\lambda < 1$,
  - Increase λ by a small amount.
  - Solve $\rho_a(\lambda, \mathbf{x}) = 0$ using the previous solution vector as a starting point to solve the new problem.
- Hope that solution exists for intermediate problems
- Hope solution is well-behaved as λ changes
- Issues:
  - Turning points, bifurcations, non-existence, divergence
- For continuous functions **F**, and with $\mathbf{F_a}$ having a unique solution, then ρ_a will exist for almost all values of λ
- If we have issues, it turns out that choosing another value of **a** in $\mathbf{F_a}$ will lead to convergence
- Issues
  - choosing the stepsize for λ.
  - choosing the tolerance in the nonlinear equation solver.

Differentiating along the curve

- Parametrize λ and **x** along the solution path arc length as λ(s) and **x**(s)
- s is the arc length

$$\rho\mathbf{a}(\lambda(s), \mathbf{x}(s)) = 0\,,$$

so

$$\frac{d}{ds}\rho\mathbf{a}(\lambda(s), \mathbf{x}(s)) = 0$$

and we have the intial conditions

$$\lambda(0) = 0\ ,\ \ \mathbf{x}(0) = \mathbf{a}\,.$$

For uniqueness, we normalize so that

$$\left\|\left(\frac{d\lambda}{ds}, \frac{d\mathbf{x}}{ds}\right)\right\| = 1\,.$$

- Use ideas from ODE solution to solve the system
- Hompack, by Watson and co-workers, is a high-quality system for solving nonlinear equations by continuation algorithms.