Solving Nonlinear Equations using MATLAB

An Introduction

Characteristics of Nonlinear Equations

A comparison of linear and nonlinear equations

- Linear Equations
  - Many methods to solve
  - Analytic solutions are always possible (if the system is well posed)
  - Have a single solution (if the system is well posed)
- Nonlinear Equations
  - *Much* more difficult to solve
  - Analytic solutions are often impossible
  - Can have multiple solutions
    - Which one do you want?

Solving for the Roots of Polynomials

Any polynomial may be written as: $f(x)=\sum_{i=0}^{n} a_i x^i$

- ROOTS(coeff)
  - Returns a vector containing the roots of the polynomial
  - coeff is a vector of length n+1 containing the a_i values in descending order
    - Coefficient of highest power first
    - Constant last
- Example: find the roots of $y=3x^3+x^2-5$
  - What do we expect?

Nonlinear Equations with *one* independent variable

- FZERO(fun, x_o)
  - Finds a zero of the function referred to by ‘fun’ near x_o
  - fun is a text string which gives the name of an m-file. This file accepts a value, x, and returns the function value f(x).
  - x_o is a starting guess for the solver. This guess should be made intelligently!
- Could we use this for our previous example?
  - Which root(s) would it give?
  - Is it sensitive to the starting position?
- Example: find roots of $f(x) = \sin(10x)\, e^{-\sqrt{x}}$ using starting guesses of 0.8, 1.0, 1.2.

Solving equations with more than one variable

Using inline functions

- fun = inline(‘function’, ‘arg$_1$’, ‘arg$_2$’, ...);
  - Defines an expression
  - arg1...argn are variables passed into this function
- FZERO(fun, x$_o$, options, arg$_1$, arg$_2$, ..., arg$_n$)
  - fun and x$_o$ are defined as before
  - options allows you to customize the solver
    - Use [ ] for the default options
  - arg$_1$ ... arg$_n$ are arguments passed to the function, which *do NOT change during the solve*.
- Example: generalized version of previous example

$$f(x)=\sin(ax)\, e^{-b\sqrt{x}}$$

Multidimensional Minimization

The concept

Consider the system of equations:

$$\begin{array}{c} f_1(x_1, x_2 \cdots x_n) \\ f_2(x_1, x_2 \cdots x_n) \\ \vdots \\ f_m(x_1, x_2 \cdots x_n) \\ with\ (m \geq n)\ in\ general \end{array}$$

- How to solve this system?
  - Write each equation in the form $f_i(x_1 \ldots x_n) = 0$
  - If we pick a set of xi, the equations will not be satisfied in general
    - Re-write in the form $f_i(x_1 \ldots x_n) = r_i$
  - Now try to minimize $\sum (r_i)^2$
  - This gives the “best” solution

Multidimensional Minimization

In MATLAB

- FMINSEARCH(fun, x_o, options, arg1, arg2,...)
  - Finds the *local* minimum of 'fun' near the guess x_o
  - fun is a text string which gives the name of an m-file (function). This function accepts a value, x, and returns the function value f(x).
  - x_o is a starting guess for the solver. This guess should be made intelligently!
    - See previous example.
  - Don't mess with options
  - arg1, arg2, ... argn can be passed into fun
    - x_o is the first argument passed into fun