Single variable case $f(x) = 0$

fzero (fun, x0)

fun: function handle

x0: initial guess

Example: $\sin(x^2) = 0 \rightarrow x^2 = \pm n\pi$

Nonlinear equation systems

fsolve (fun, x0)

fun: function that returns to residual at $\underline{x}$

x0: initial guess vector

Example: $\underline{f}(\underline{x}) = \underline{0}$

$$f_1 = x_1^2 + x_1 x_2 + x_1 - 1 = 0$$

$$f_2 = x_1 x_2 + x_2 + x_3^2 - \frac{1}{4} = 0$$

$$f_2 = x_1 x_2 + x_2 + x_3^2 - \frac{1}{4} = 0$$

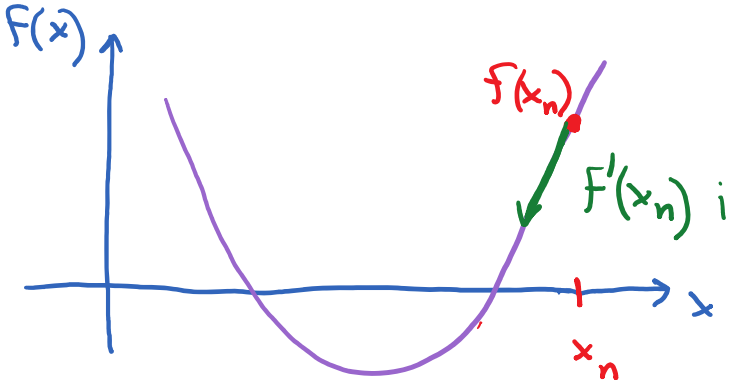
$$f_3 = x_1^2 + x_2^2 - 4x_3 = 0$$

Previously, considered single variable minimization algorithms:

Brent's method ~ fit quadratic to f using three points; minimize quadratic; iterate

Newton's method ~ find zeros of first derivatives of f

Gradient descent method



$$x_{n+1} = x_n - \alpha_n f'(x_n)$$

Note: Negative sign for gradient descent

α_n is chosen such that

$$f(x_{n+1}) < f(x_n) \longrightarrow \text{Line search}$$

Multidimensional Minimization

Let $\underline{x} \in \mathbb{R}^m$; scalar function

Let $\underline{x} \in \mathbb{R}^m$;

scalar function

Consider m dimensional minimization of $f(\underline{x})$

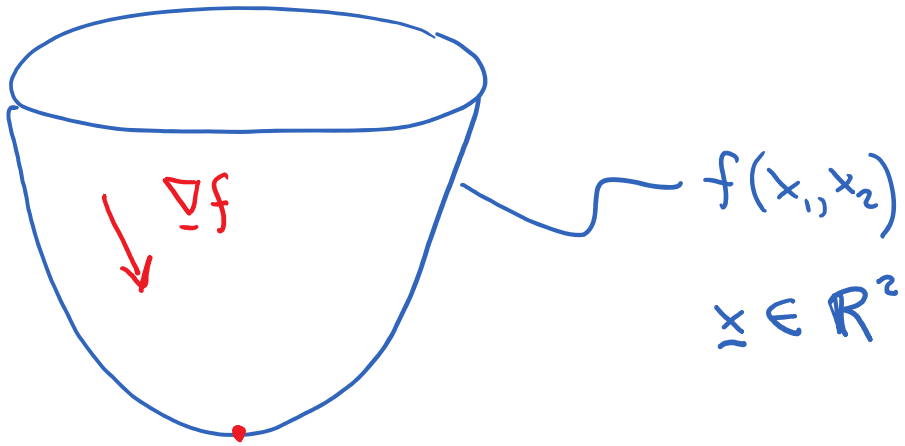
(1) Gradient Descent

Extended to higher dimensions

$$\underline{\nabla} f = \left\{ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_m} \right\}^T$$

$$\underline{x}_{n+1} = \underline{x}_n - \alpha_n \underline{\nabla} f(\underline{x}_n) \quad \text{at iteration } n$$

Choose α_n such that $f(\underline{x}_{n+1}) < f(\underline{x}_n)$



One can show that if α_n satisfies some Wolfe Conditions (not to be discussed), then method will converge to a local minimum.

Example: $\alpha_n = \dfrac{(\underline{x}_n - \underline{x}_{n-1})^T \left(\underline{\nabla} f(\underline{x}_n) - \underline{\nabla} f(\underline{x}_{n-1}) \right)}{2}$

Example: $\alpha_n = \dfrac{(\underline{x}_{n-1} - \underline{x}_n)^T \left[\nabla f(\underline{x}_{n-1}) - \nabla f(\underline{x}_n)\right]}{\left\| \nabla f(\underline{x}_n) - \nabla f(\underline{x}_{n-1}) \right\|_2^2}$

α_n: scalar

Other methods might have faster convergence

If $f(\underline{x})$ is convex, then the local minimum is the global minimum

(2) Newton's method for minimization

Recall in one-dimension

$$f'(x_n + \Delta x) = f'(x_n) + \Delta x\, f''(x_n) + \frac{1}{2}\Delta x^2 f'''(x_n) + \mathcal{O}(\Delta x^3)$$

$\Delta x \Rightarrow 0$

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

In multidimensions

$$\nabla f(\underline{x} + \underline{\Delta x}) = \nabla f(\underline{x}) + \underline{H}\,\underline{\Delta x} + \text{H.O.T.}$$

H.O.T. = Higher Order Terms

Hessian $\underline{H} = \nabla \nabla f(\underline{x})$ (Symmetric matrix)

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_m} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & & & \frac{\partial^2 f}{\partial x_2 \partial x_m} \\ \frac{\partial^2 f}{\partial x_m \partial x_1} & & & \frac{\partial^2 f}{\partial x_m^2} \end{bmatrix}$$

To find the minimum, solve

$$\frac{\partial f}{\partial \underline{\Delta x}} = 0 \text{, ignoring } \mathcal{O}(\underline{\Delta x}^2) + \text{higher terms}$$

$$\underline{0} \approx \underline{\nabla} f(\underline{x}_n) + \underline{\underline{H}}(\underline{x}_n)\,\underline{\Delta x}$$

$$\underline{\Delta x} = -\underline{\underline{H}}^{-1}\,\underline{\nabla} f \quad \text{with} \quad \underline{\Delta x} = \underline{x}_{n+1} - \underline{x}_n$$

$$\Rightarrow \boxed{\underline{x}_{n+1} = \underline{x}_n - \underline{\underline{H}}_n^{-1}\,\underline{\nabla} f_n}$$

1d: $x_{n+1} = x_n - \frac{f_n'}{f_n''}$

Issues: What if $\underline{\underline{H}}_n^{-1}$ does not exist or is ill-conditioned? Or very expensive to evaluate?

③ Quasi-Newton Methods: Methods that use the concept of Newton method, but do not evaluate the true Hessian

evaluate the true Hessian

Let $f(\underline{x})$ be the function to minimize, having $\underline{x}_n$ as an approximation of the true minimum $\underline{x}^*$

Write an approximate Taylor series

$$\underline{\nabla} f(\underline{x} + \underline{\Delta x}) \cong \underline{\nabla} f(\underline{x}) + \underline{B}_n \underline{\Delta x}$$

with $\underline{B}$ as an approximation of $\underline{H}$ that retains some of the nice properties of $\underline{H}$, such as positive definiteness

An iteration then can be written

$$\underline{x}_{n+1} = \underline{x}_n - \alpha_n \underline{B}_n^{-1} \underline{\nabla} f(\underline{x}_n)$$

where α_n is chosen to minimize $f(\underline{x}_{n+1})$

How should $\underline{B}_n$ be determined?

Basic method:

Let $\underline{x}_0$ & $\underline{B}_0$ be given

Loop over n

$n^{-1} = c(\ )$.

$$\Delta \underline{x}_n = -\alpha_n \underline{B}_n^{-1} \nabla f(\underline{x}_n)$$

$$\underline{x}_{n+1} = \underline{x}_n + \Delta \underline{x}_n$$

} line search

$$\underline{y}_n = \nabla f(\underline{x}_{n+1}) - \nabla f(\underline{x}_n)$$

$\underline{B}_{n+1}$ is then a function of $\underline{B}_n, \Delta \underline{x}_n, \underline{y}_n$

End loop when convergence is achieved

Different algorithms have different approaches to establish $\underline{B}_{n+1}$

One good choice for $\underline{B}_0$

Let $\Delta \underline{x}_0 = -\alpha_0 \nabla f(\underline{x}_0)$

$$\underline{y}_0 = \nabla f(x_0 + \Delta x_0) - \nabla f(\underline{x}_0)$$

$$\underline{B}_0 = \left(\frac{\underline{y}_0^T \underline{y}_0}{\underline{y}_0^T \Delta \underline{x}_0} \right) \underline{I}$$

h

Scalar factor

Davidon - Fletcher - Powell (DFP)

$$\underline{\underline{B}}_{n+1} = \left(\underline{\underline{I}} - \frac{\underline{y}_n \underline{\Delta x}_n^T}{\underline{y}_n^T \underline{\Delta x}_n}\right) \underline{\underline{B}}_n \left(\underline{\underline{I}} - \frac{\underline{\Delta x}_n \underline{y}_n^T}{\underline{\Delta x}_n^T \underline{y}_n}\right) + \frac{\underline{y}_n \underline{y}_n^T}{\underline{y}_n^T \underline{\Delta x}_n}$$

$$\underline{\underline{B}}_{n+1}^{-1} = \underline{\underline{B}}_n^{-1} + \frac{\underline{\Delta x}_n \underline{\Delta x}_n^T}{\underline{\Delta x}_n^T \underline{y}_n} - \frac{\underline{\underline{B}}_n^{-1} \underline{y}_n \underline{y}_n^T \underline{\underline{B}}_n^{-1}}{\underline{y}_n^T \underline{\underline{B}}_n^{-1} \underline{y}_n}$$

Broyden-Fletcher- Goldfarb-Shanno (BFGS)

$$\underline{\underline{B}}_{n+1} = \underline{\underline{B}}_n + \frac{\underline{y}_n \underline{y}_n^T}{\underline{y}_n^T \underline{\Delta x}_n} - \frac{\underline{\underline{B}}_n \underline{\Delta x}_n \underline{\Delta x}_n^T \underline{\underline{B}}_n}{\underline{\Delta x}_n^T \underline{\underline{B}}_n \underline{\Delta x}_n}$$

$$\underline{\underline{B}}_{n+1}^{-1} = \left(\underline{\underline{I}} - \frac{\underline{\Delta x}_n \underline{y}_n^T}{\underline{\Delta x}_n^T \underline{y}_n}\right) \underline{\underline{B}}_n^{-1} \left(\underline{\underline{I}} - \frac{\underline{y}_n \underline{\Delta x}_n^T}{\underline{y}_n^T \underline{\Delta x}_n}\right) + \frac{\underline{\Delta x}_n \underline{\Delta x}_n^T}{\underline{\Delta x}_n^T \underline{y}_n}$$

Notice duality between DFP and BFGS

$$\underline{\Delta x}_n \leftrightarrow \underline{y}_n$$

$$\underline{\underline{B}}_n \leftrightarrow \underline{\underline{B}}_n^{-1}$$

Other minimization algorithms:

- Levenberg - Marquardt
- Nedler - Mead
- Trust Region
- ⋮

Nocedal + Wright

Numerical

Optimization

Heuristic optimization:

- Genetic Algorithms
- Simulated Annealing
- Particle Swarm
- ⋮