Lecture 7: Minimization or maximization of functions (Recipes Chapter 10)

• Actively studied subject for several reasons:
  – Commonly encountered problem: e.g. Hamilton’s and Lagrange’s principles, economics problems, statistical fitting of data ($\chi^2$ or maximum likelihood)....
  – For the most interesting cases (multivarient, non-linear functions), there is no “best technique”
  – There are many competing methods each with some advantages and disadvantages
Minimization of 1-D functions

• We will search for a minimum of a function $f(x)$ on some interval $[x_1, x_2]$
Minimization of 1-D functions

• Notes:
  – No loss of generality in focusing on minimum: for maximum consider the function \(- f(x)\)
  – Global minimum need not have \(f'(x) = 0\)
How accurately can the minimum be found?

- Suppose a minimum of $f(x)$ occurs at $x = b$ (in the case where $f'(b) = 0$)

$$f(x) = f(b) + f'(b) (x-b) + \frac{1}{2} f''(b) (x-b)^2 + \ldots.$$ 

Define $\delta f = f(x) - f(b)$ as the smallest difference in FP numbers that we can distinguish:

Then $\delta f = \varepsilon f(b)$

with $\varepsilon \sim 10^{-8}$ in single precision or $\sim 10^{-16}$ in double precision
How accurately can the minimum be found?

Then
\[ \delta f = \varepsilon f(b) = \frac{1}{2} f''(b) (x-b)^2 \]

and \[ |x-b| = \sqrt{2 \varepsilon f(b) / f''(b)} \sim b \varepsilon^{1/2} \]

we typically cannot resolve minima with fractional accuracies better than \( \sim 10^{-4} \) in single precision.
Bracketing for minimization

• As in the case of root finding, the best 1-D techniques make use of bracketing. In this context, a “bracket” is defined by three points, \( a < b < c \), for which \( f(b) < f(a) \) AND \( f(b) < f(c) \)

A minimum must lie between \( a \) and \( c \)
Establishing a bracket

1) Choose two points, $x_1 < x_2$, separated by $d = |x_2 - x_1|$

2) If $f(x_1) < f(x_2)$, choose $x_3 = x_1 - d$
   else, choose $x_3 = x_2 + d$
Establishing a bracket

3) Evaluate $f(x_3)$
   
   If $f(x_3) > \min [f(x_2), f(x_1)]$, we are done
   
   else, set $x_2 = x \ (\min [f(x_2), f(x_1)])$, $x_1 = x_3$
   
   return to step (1)

Can accelerate this by allowing the step size to grow
Golden section search

- We now want to squeeze the bracket:
  - put in a new point $d$
  - New bracket is either $\{a \ d \ b\}$ or $\{d \ b \ c\}$
Golden section search

- Choose the distances shown below such that the ratios are preserved \((1-w):w = w:(1-2w)\)

Leads to self-similarity
Golden section search

• The required value of \( w \) is the solution to
  \[ w^2 - 3w + 1 = 0 \]
  \[ w = \frac{1}{2} (3 - \sqrt{5}) = 0.38197..... = 1 - \phi \]
  (need solution with \( w < 1 \))

• This procedure converges \textit{linearly}, with bracket size after \( N \) iterations given by \( (x_2 - x_1) 0.618^N \)

  cf. bisection for root finding yields bracket size \( (x_2 - x_1) 0.5^N \) after \( N \) iterations
Faster methods

• As with bisection, in the Golden section method we only ask about whether certain quantities (e.g. \( f(d) - f(c) \) are positive or negative)

• We can accelerate convergence by using more information about the values of various quantities
Brent’s method
(a.k.a. inverse parabolic interpolation)

• In Brent’s method, we expand about the true minimum, \( x^* \)

\[
f(x) = f(x^*) + \frac{1}{2} f''(x^*) (x-x^*)^2 + R(x)
\]

If \( R(x) \) were zero, we would have
three unknowns: \( x^*, f(x^*), \) and \( f''(x^*) \)

three data points: \( f(a), f(b), f(c) \)

(from the function values on our three bracket points)
Brent’s method

• The solution is

\[ x_4 = x_3 - \frac{(x_3-x_1)^2 [f_3 - f_2] - (x_3-x_2)^2 [f_3 - f_1]}{2 (x_3-x_1) [f_3 - f_2] - (x_3-x_2) [f_3 - f_1]} \]

• If \( x_4 \) is reasonable – i.e. lies in the interval \([x_1,x_2]\) and yields \( f_4 < f_3 \) (previous smallest value) use it to form a new bracket

• Otherwise, revert to Golden section
Brent’s method

For the case \( R(x) \sim 1/6 f'''(x^*) (x - x^*)^3 \),

we find that \( |x_4 - x^*| \sim \left[ \frac{2f'''(x^*)}{2f''(x^*)} \right]^{1/2} |x_3 - x^*|^{3/2} \)

\( \Rightarrow \) supralinear convergence (m=1.5) when it works
(or m = 1 when it reverts to Golden section)

Hybrid method: combines robustness (valid bracket always maintained) with increased speed when possible
Use of derivative information

• When $f'(x)$ is known explicitly, this information can be used to further improve performance
  – *Recipes* has a hybrid routine that uses the secant method to find the root of $f'(x)$ with the Golden section method to ensure that a bracket is maintained
Multi-D minimization (*Numerical Recipes*, §10.4 – 10.7)

- As with root finding, things get a lot harder when $f$ is a function of several variables
  - no analog to a “bracket”
- Overview of techniques
  - Function evaluations only $\rightarrow$ downhill simplex method
  - Function evaluation to estimate the optimum direction of motion $\rightarrow$ Powell’s method
  - Function evaluations and explicit gradient calculation $\rightarrow$ Conjugate Gradient Method
Downhill simplex

- A simplex is a hyperpolygon of $N + 1$ vertices in an $N$-dimensional space
  - $N = 2$: triangle
  - $N = 3$: tetrahedron

- If one vertex is at the origin of the coordinate system, the others are given by $N$ vectors which span the $N$-dimensional space:
  \[ V_i = P_i - P_0 \quad (i = 1, N), \] where $P_i$ is the $i$th vertex
Downhill simplex

- Downhill simplex involves moving a simplex downhill to find the minimum of a function
- Basic move: reflection in the face opposite the vertex for which $f$ is largest

Largest value here
Downhill simplex

- Additional moves:
  - Stretch to accelerate motion in a particular direction
  - Contraction, if reflection overshoots the minimum
- Press et al. name their routine AMOEBA
3–D representation
(from Recipes)

- simplex at beginning of step
- contraction
- reflection
- multiple contraction
- reflection and expansion
Direction set methods

• Basic tool of all such methods is a 1-D minimization (Golden section, Brent’s method)
• Choose a starting position \( p \), and a direction \( \hat{n} \), and minimize \( f (p + \lambda \hat{n}) \)
• Now use \( p + \lambda \hat{n} \) as the new starting position, choose a different direction, and minimize along that direction…….
• Methods differ as to how the directions are chosen
Direction set methods

- Simplest method: take \( N \) orthogonal unit vectors in turn, \( \hat{e}_i \)
- Slow convergence, unless the unit vectors are well-oriented with respect to the valley.

*Recipes*, Fig 10.5.1

Figure 10.5.1. Successive minimizations along coordinate directions in a long, narrow “valley” (shown as contour lines). Unless the valley is optimally oriented, this method is extremely inefficient, taking many tiny steps to get to the minimum, crossing and re-crossing the principal axis.
Direction set methods

• Better methods update the directions as the method proceeds, so as to
  – choose favorable directions that proceed far along narrow valleys
  – choose “non-interfering” directions, such that the next direction doesn’t undo the minimization achieved by previous steps
Steepest descent

• If you know the derivatives of $f$ (i.e. you know $\nabla f$), you might think that you would do best to choose $\hat{n} = -\nabla f / |\nabla f|$

• This is the method of steepest descent

• BUT, this means you always choose a new direction that is *orthogonal* to the previous direction

i.e. $\hat{n}_{i+1} \cdot \hat{n}_i = 0$
Steepest descent

• The performance isn’t that good, because we can only ever take a right angle turn

Recipes, Fig 10.6.1
Steepest descent: 2-D example

• Suppose step k occurred along the y-axis, and led to position $p_{k+1}$, at which $\partial f/\partial y = 0$.

• Next step is along the x-axis: that step leads to a position $p_{k+2}$, where $\partial f/\partial x = 0$.

• But if $\partial^2 f / \partial y \partial x$ is non-zero, $\partial f/\partial y$ will no longer be zero.

• We really want to move along some direction other than the x-axis, such that $\partial f/\partial y$ remains zero.