THE UNIVERSITY OF WESTERN ONTARIO
LONDON ONTARIO

Paul Klein
Office: SSC 4028
Phone: 661-2111 ext. 85227
Email: paul.klein@uwo.ca
URL: www.ssc.uwo.ca/economics/faculty/klein/

Economics 613/614

Advanced Macroeconomics I & 2

Fall 2007

Numerical optimization

In these notes we consider some methods of numerical *minimization* since that is what engineers are mostly up to. To maximize f, minimize $-f$.

Broadly speaking, optimization methods can be divided into gradient-base methods and non-gradient based methods. Gradient-based methods are typically faster but less robust. Which is better depends on the problem at hand. If the mimimand is smooth and you have a good initial guess, gradient-based methods are superior.

1 Golden section search

Golden section search finds a minimizer in one dimension of a single-troughed function.

1.1 The golden section

$$\frac{a+b}{a} = \frac{a}{b} = \varphi.$$
$$\frac{1+\varphi}{\varphi} = \varphi.$$
$$\varphi^2 - \varphi - 1 = 0.$$
$$\varphi = \frac{1+\sqrt{5}}{2} \approx 1.61803$$

1.2 Finding three points

Golden section search starts with three points x_1, x_2, x_3 such that $f(x_1) > f(x_2)$ and $f(x_3) > f(x_2)$. If f is single-troughed, we can be sure that the minimizer lies between x_1 and x_3. But how to find three such points?

Suppose we have not yet bracketed the minimum, i.e. either $f(x_1) < f(x_2)$ or $f(x_3) < f(x_2)$.

If $f(x_1) < f(x_2)$ then we can toss out x_3 and create a new point between x_1 and x_2. If $f(x_3) < f(x_2)$ we can toss out x_1 and create a new point between x_2 and x_3.

This will eventually work provided that the minimizer is in between the original x_1 and x_3, i.e. we have already *bracketed* the minimum. If not, we need some way of bracketing the minimum, somehow you need to successively expand the interval $[x_1, x_3]$ and then look for a suitable x_2 as above.

1.3 Updating: case 1

At the end of the previous computation, we have three points with the desired property and preferably also with the following "golden section" property.

$$\frac{x_3 - x_2}{x_2 - x_1} = \varphi.$$

and the next point where we evaluate f should be chosen in this way too, i.e.

$$\frac{x_3 - x_4}{x_4 - x_2} = \varphi.$$

(This is illustrated in Figure 1.) Why is this desirable? Well, the next point where we evaluate f should be in the larger subinterval (x_2, x_3). Depending on whether $f(x_4)$ is greater than or less than $f(x_2)$, our new interval has a width of either $x_4 - x_1$ or $x_3 - x_2$. To minimize the significance of bad luck, the length of these two intervals should be equal. In the notation of Figure 1, we should have

$$a + c = b.$$

Also, the spacing between the points should be the same before and after the revising of the triplet of points, i.e.

$$\frac{c}{a} = \frac{a}{b}.$$

These two equations together imply

$$\frac{b}{a} = \frac{a}{b} + 1$$

which leads to the conclusion that

$$\frac{b}{a} = \varphi.$$

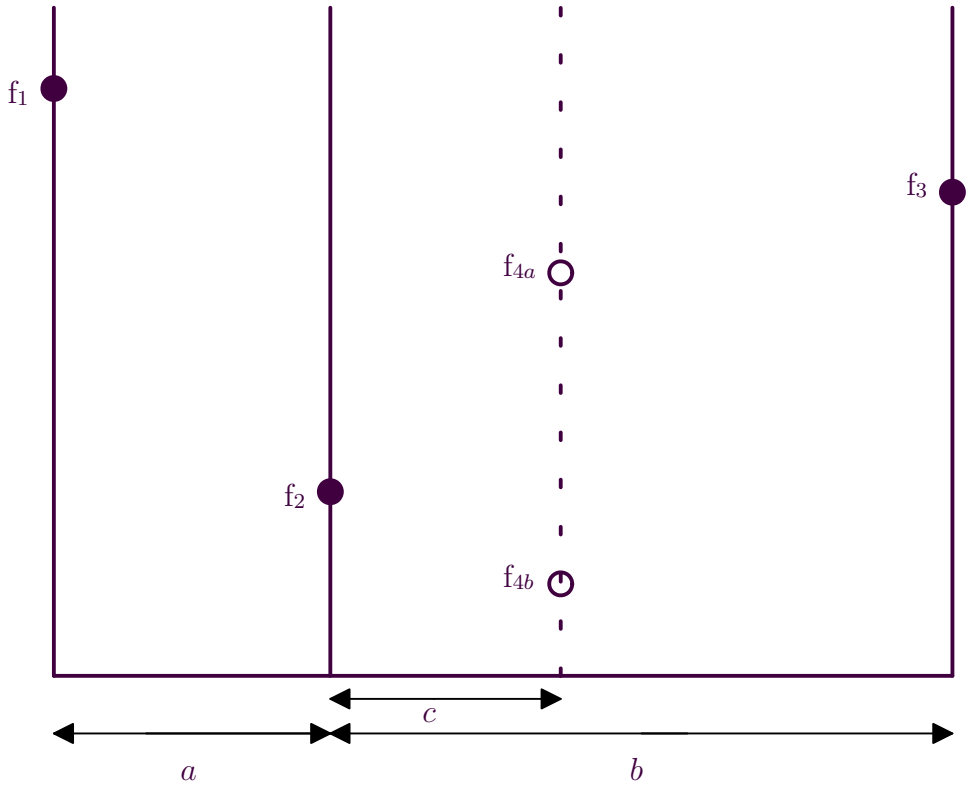

Figure 1: Golden section search: case 1

1.4 Updating: case 2

What if it turns out that the new interval is in fact $[x_1, x_4]$? Then we will have the bigger subinterval first and the smaller subinterval second. The code has

to allow for that possibility. In fact we should draw a picture for that case, too. See Figure 2. In this case, we get

$$a = b + c$$

and

$$\frac{a}{b} = \frac{a - c}{c}$$

and the conclusion is that

$$\frac{a}{b} = \varphi.$$

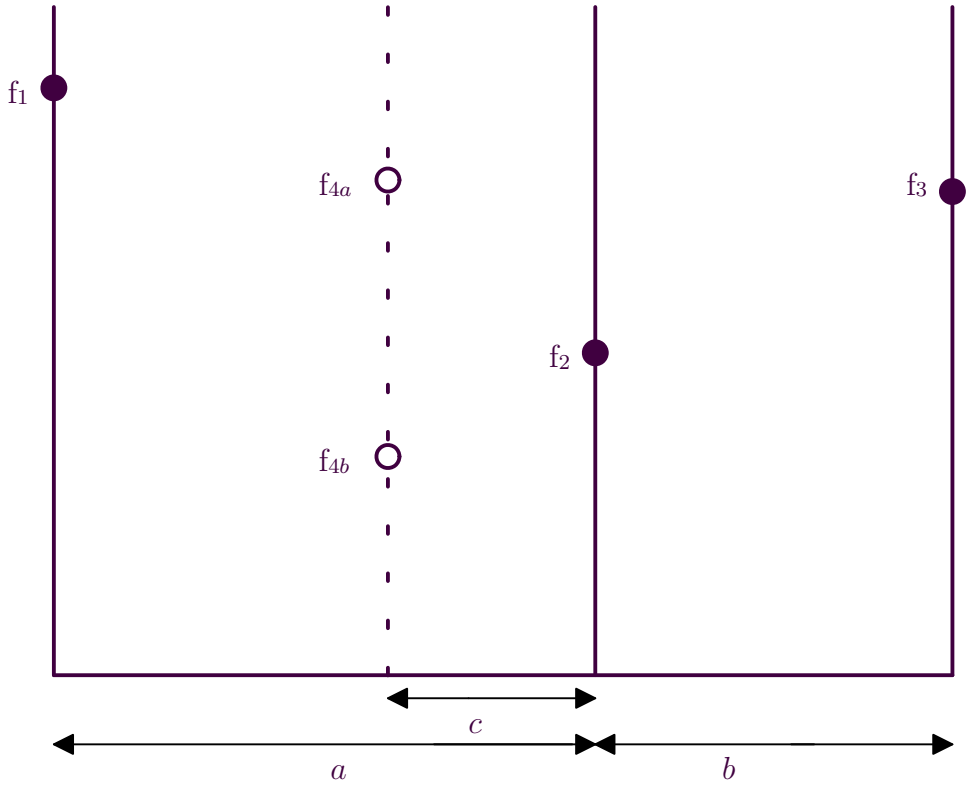

Figure 2: Golden section search: case 2

1.5 When to stop

It is tempting to think that you can bracket the solution x^* in a range as small as $(1-\epsilon)x^* < x^* < (1+\epsilon)x^*$ where ϵ is machine precision. However, that is not so!

$$f(x) \approx f(x^*) + f'(x^*)(x-x^*) + \frac{1}{2}f''(x^*)(x-x^*)^2$$

The second term is zero, and the third term will be negligible compared to the first (that is, will be a factor ϵ smaller and so will be an additive zero in finite precision) whenever

$$\frac{1}{2}f''(x^*)(x-x^*)^2 < \epsilon f(x^*)$$

or

$$\frac{x-x^*}{x^*} < \sqrt{\epsilon}\sqrt{\frac{2|f(x^*)|}{(x^*)^2 f''(x^*)}}.$$

Therefore, as a rule of thumb, it is hopeless to ask for bracketing with a width of less than $\sqrt{\epsilon}$!

2 Brent's method

If the function to be minimized is smooth (is continuously differentiable) then approximating it by a parabola and taking as the new approximation of the minimizer the minimizer of the parabola.

Let the minimizer be bracketed by the three points a, b and c. Then the following number minimizes the parabola that goes through the points $(a, f(a))$, $(b, f(b))$ and $(c, f(c))$.

$$x = b - \frac{1}{2}\frac{(b-a)^2[f(b)-f(c)] - (b-c)^2[f(b)-f(a)]}{(b-a)[f(b)-f(c)] - (b-c)[f(b)-f(a)]}.$$

A pure inverse parabolic interpolation method proceeds as follows. If $a < x < b$, then the new three points are a, x and b. If on the other hand $b < x < c$, then the new three points are b, x and c.

This method is fast but could easily become numerically unstable. So a good algorithm cannot be based just on this. Brent's method switches between inverse parabolic interpolation as described above and golden section search.

Inverse parabolic interpolation is used to update a, b and c if it is "acceptable" (in the current interval and represents a noticeable improvement over the previous guess.) If not, then the algorithm falls back to an ordinary golden section step. Notice that golden section search doesn't *have* to start with two intervals with the "right" ratio of lenghts. All you need to know is which is the bigger interval and then you choose your new point there.

3 Newton's method

Newton's method of optimization is to apply Newton's method of root finding to the equation $f'(x) = 0$. The first-order approximation of this function around the nth approximation x_n of the true solution x^* is

$$f'(x^*) \approx f'(x_n) + f''(x_n) * (x^* - x_n).$$

where we interpret $f'(x_n)$ as the $(1 \times n)$ gradient of f at x_n and $f''(x_n)$ as the $(n \times n)$ Hessian at x_n. Evidently $f'(x^*) = 0$ so we can solve for $\Delta x_n = x^* - x_n$ by solving

$$f''(x_n)\Delta x_n = -f'(x_n)$$

and then defining

$$x_{n+1} + \gamma \Delta x_n$$

where $0 < \gamma \leq 1$.

4 The method of steepest descent

A nice fact about the gradient $\bigtriangledown f(x)$ of a function is that it points in the direction of steepest ascent at x. To descend, we of course want to move in the opposite direction. The only question is how far. Let that be an open question for now, denoting the distance travelled in each iteration by α_k.

$$x_{k+1} = x_k - \alpha_k \bigtriangledown f(x_k).$$

But how to choose α_k? A natural option would seem to be to minimize f along the line $x = x_k - \alpha \bigtriangledown f(x_k)$ and that is precisely what is typically done, i.e. α_k solves

$$\min_{\alpha} f(x_k - \alpha \bigtriangledown f(x_k)).$$

This is a one-dimensional minimization problem that can be solved using, for example, the method of golden section or Brent's method.

5 The Nelder-Mead method

Golden section is fine in one dimension, and steepest descent is good in arbitrarily dimensions provided we are comfortable with computing gradients. But what to do in higher dimensions with functions that are not necessarily

differentiable? On popular alternative is the Nelder-Mead algorithm. We will illustrate it in two dimensions, but the algorithm can fairly easily be extended to arbitrarily many dimensions.

The method starts with three (or $n+1$ in n dimensions) points B, G and W that do not lie on a line (or hyperplane in more than two dimensions). The notation is there to capture the assumption that

$$f(B) < f(G) < f(W).$$

Think "Best", "Good", "Worst".

We will now describe the set of points constructed from B, G and W that might be computed at some step along the decision tree. We then complete our description of the algorithm by describing the decision tree.

5.1 Definition and illustration of new points

The points R (reflection point) and M (midpoint of the good side) are defined as follows and are illustrated in Figure 3.

$$M = \frac{1}{2}(B + G).$$

$$R = 2M - W.$$

The extension point E is defined as follows and is illustrated in Figure 4.

$$E = 2R - M.$$

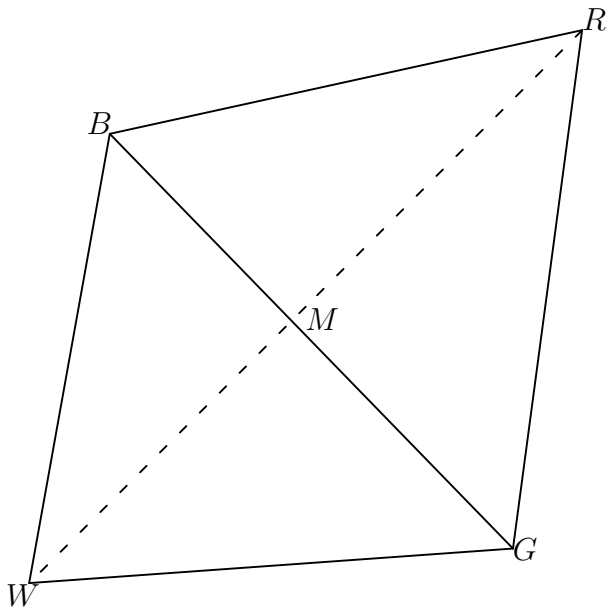

Figure 3: The triangle BGW and the points R and M.

The contraction points C_1 and C_2 are is defined as follows and is illustrated in Figure 5.

$$C_1 = \frac{1}{2}(M + W).$$

$$C_2 = \frac{1}{2}(M + R).$$

Whichever of the points C_1 and C_2 delivers the lowest value of f is called C.

The shrinkage point S is defined as follows and is illustrated in Figure 6.

$$S = \frac{1}{2}(B + W).$$

5.2 Decision tree

Compute R and $f(R)$.

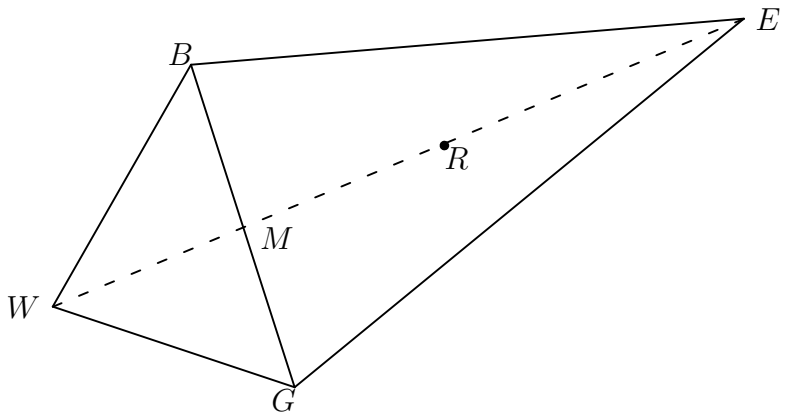

Figure 4: The triangle BGW and the points R, M and E.

Case I: If $f(R) < f(G)$ then either reflect or extend.

Case II: If $f(R) > f(G)$ then reflect, contract or shrink.

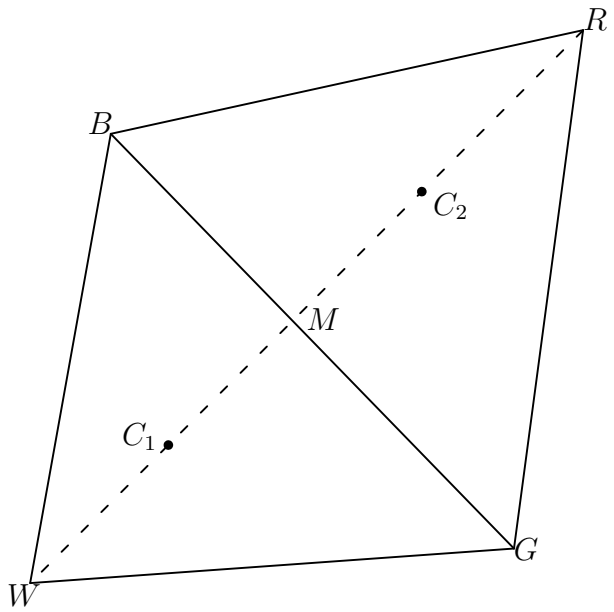

Figure 5: The triangle BGW and the points R, M, C_1 and C_2.

Case I

```
if f(B) < f(R)
  replace W with R
else
  compute E and f(E)
  if f(E) < f(B)
   replace W with E
  else
   replace W with R
  endif
endif
```

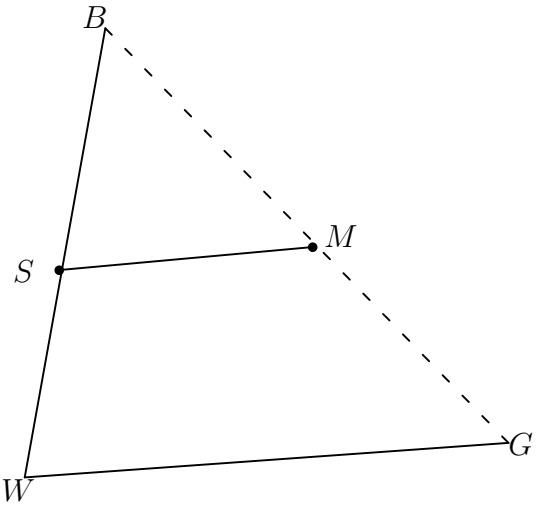

Figure 6: The triangle BGW and the points M and S.

Case II

```
if f(R) < f(W)
  replace W with R
else
  compute C and f(C)
  if f(C) < f(W)
   replace W with C
  else
   compute S and f(S)
   replace W with S
   replace G with M
  endif
endif
```

Notice that when we say "replace W with R" that doesn't mean that R is the new W, it just means that the new points are B, G, and R. They may need to be renamed appropriately.