Optimization

1. Variants of Bisection
   1.618..., the Golden mean, etc.

2. Bracketing revisited
   make sure you have the duck cornered
   and to what tolerance

3. Parabolic method
   Brent revisited

4. And, what if we use derivatives

On to higher dimensions

1. Nelder and Mead's downhill simplex

2. Powell's method

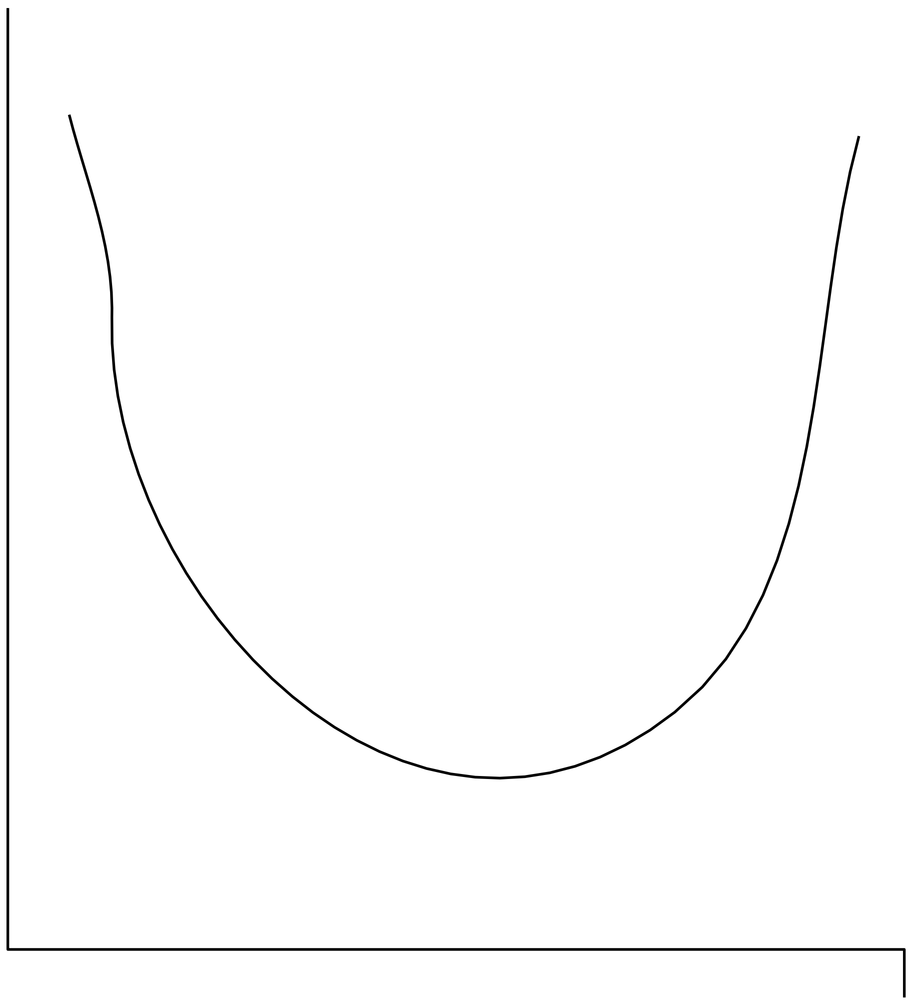
3. Conjugate directions
   Using gradients:
   - steepest descent
   - conjugate gradient methods

4. Variable metric methods

Finding the minimum

simple observation -- also works for maximum

Simplest situation



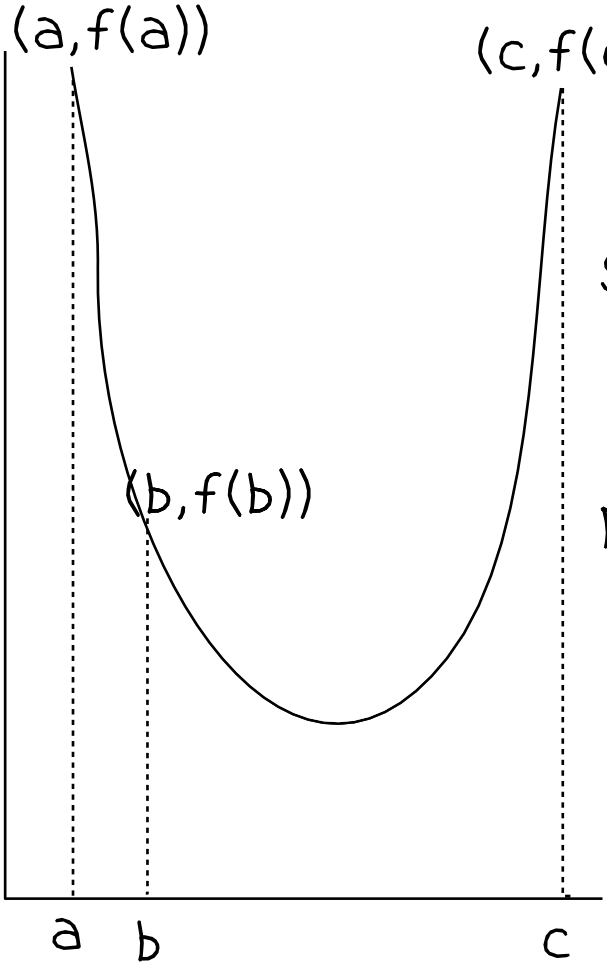
Ultimately, what you want to do is find the place where $f'(x) = 0$. You can do this in various ways.

Simplest -- estimate $f'(x)$ by sampling at $x, x+dx$
and apply Newton's method to find 0

Bisection based method

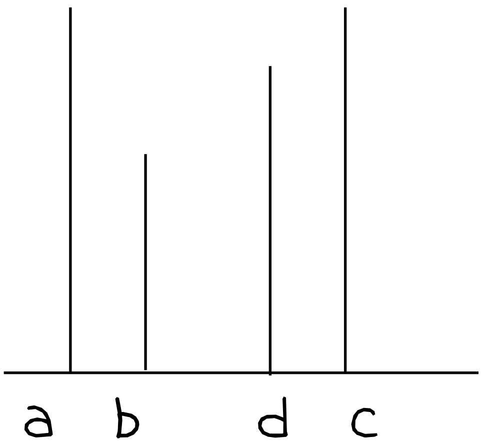
3 samples already taken (rather than 2 as before)

Where does next sample go?

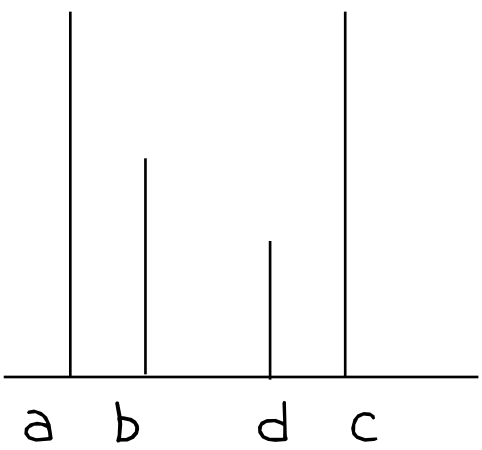


Suppose $a = 0$, $c = 1$;
where should b lie and
where should d
(next choice) lie.

Put next sample at d ($b<d<c$).



If $f(d) > f(b)$ recurse on [a,d]



else recurse on [b.c]

```
If f(d) > f(b),
        recurse on [a,d]
else
        recurse on [b.c]
```

So, want d-a = c-b;
or d = 1-b (*)

Want to maintain the invariant that d is proportionally as far from b to c and b was from a to c.

So, (d-b)/(1-b) = b (**)

combine (*) and (**) to $b^2 - 3b + 1 = 0$

So b = (3 - sqrt(5))/2 ~ .38196601125010515179

d = 1-b = .61803398874989484821;

d*d=b

Golden mean 8->

Bracketing a minimum (or a root)

How to bracket minimum (or root).

For root --> given [a,b], want to expand (or contract) the interval until it f(a)*f(b) < 0.

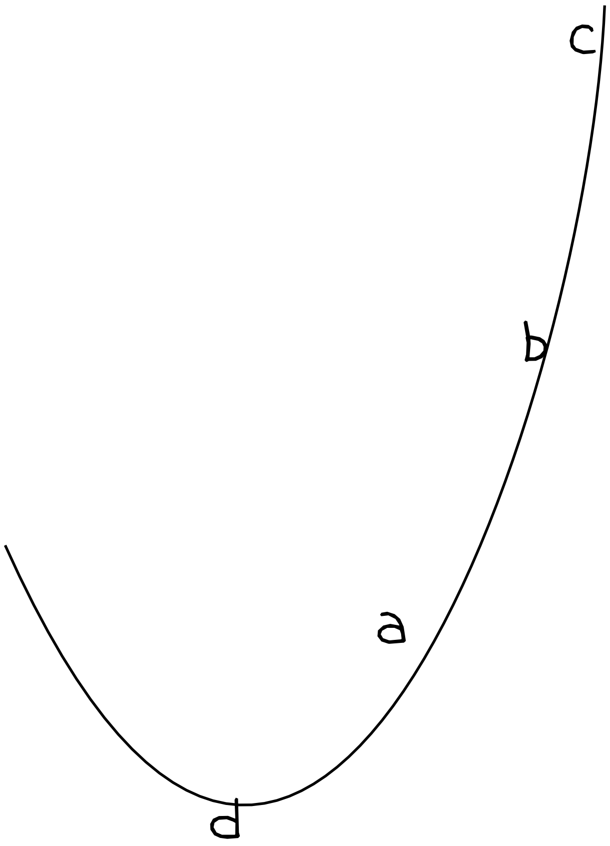
```
/* to expand interval */
while (f(a)*f(b) > 0)        {
        if (fabs(f(a)) < fabs(f(b)))
                a = a + C * (a-b)
        else
                b = b + C * (b-a)
        }


/*to contract interval, assume f(a),f(b) > 0*/
dx = (b-a)/N
for (x = a, i = 0 ; i < N ; i++ )   {
        x += dx;
        if (f(x) < 0)
                return       /* interval found */
        }
```

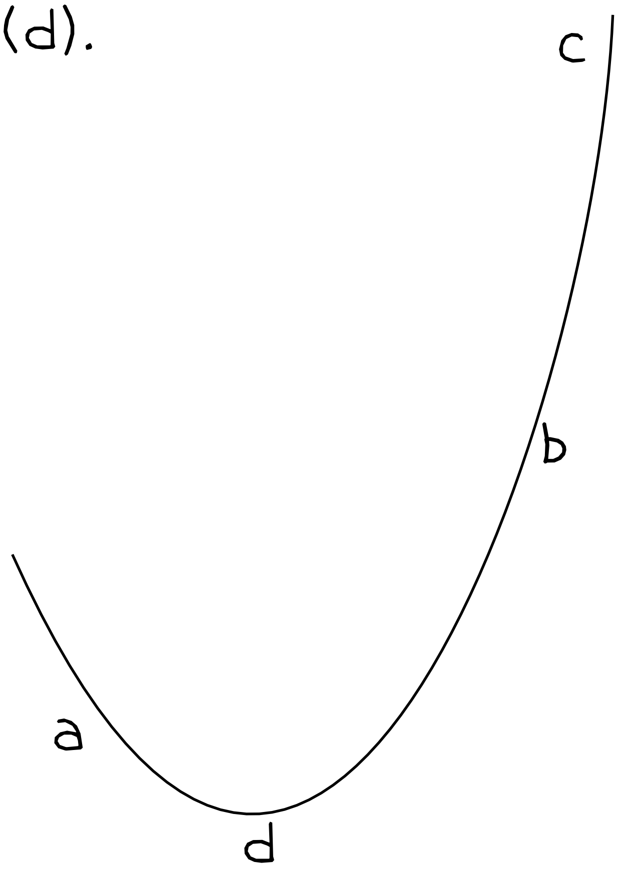
Adjusting bracketing to minimization

Now, we are looking for bracketing of a minimum, so $a < b < c$ such that

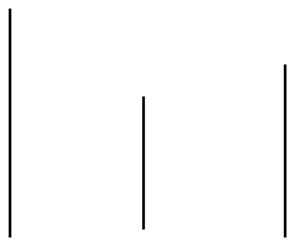
$$f(a) > f(b) < f(c)$$

Fit a parabola to the 3 points
$$(a,f(a)),\ (b,f(b)),\ (c,f(c))$$
Find it's minimum d and f(d).





Places where d can lie -- now work off relative values until you get



Brent's method revisited

Parabolic interpolation
For samples (a,f(a)), (b,f(b)), (c,f(c)),
The minimum x of the parabola thru these 3 points lies at

$$x = b - 1/2 * N/D$$

where

$$N = (b-a)^2(f(b) - f(c)) - (b-c)^2(f(b)-f(a))$$

$$D = (b-a)(f(b) - f(c)) - (b-c)(f(b)-f(a))$$

Assuming [a,b] initially brackets the minimum,

Brent does
1) find x.
2) if x in [a,b] and movement to x is a step
   less than half the movement of
   the step before last
      move to x
   else
      use the Golden Section move
3) stop when you get to an accuracy of **tol**
   (see Press for details)

Using derivatives

Wild eyed method(1) --

You know $f(a),f(b),f(c),f'(a),f''(b),f'(c)$, so interpolate a 5th degree equation, differentiate and find it's root.

Wild eyed method(2) --

Use a root finding method to find a zero of the derivative.

Fuddy-Duddy method --

Find a,b,c as before. Use $f'(c)$ to predict whether the minimum occurs in $[a,c]$ or $[c,b]$. Use the secant method (with the wisdom of regula falsi) to find the new **x**

Why the conservative approach???

On to Higher dimensions

Key idea of all algorithms...
You're walking through the mountains to find the lowest point. So, you always look for a path down.

Questions to answer

Do you always take the steepest path?

How big a step do you take?

How do you know whether you've hit bottom?

Even if you have, how do you know it's a global and not local minimum?

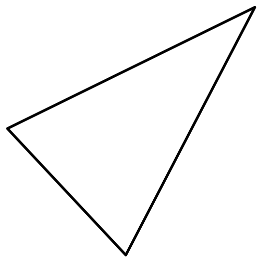
Downhill Simplex Method

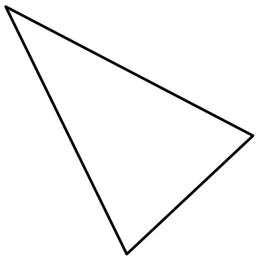
simplex is basic shape in N-dimensions

N=1 segment
N=2 triangle
N=3 tetrahedron
...

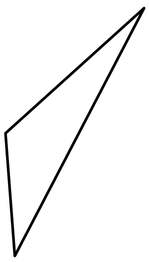
Don't start with a point, start with a simplex

Do updates to the simplex.

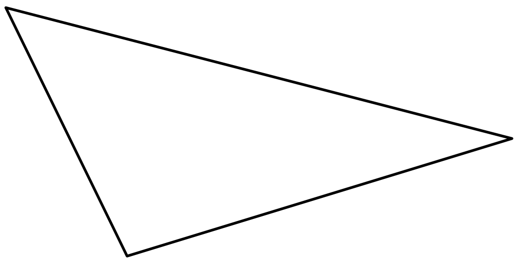




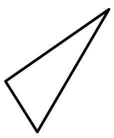
reflection



contraction



reflection and
expansion



multiple contraction

Walking methods

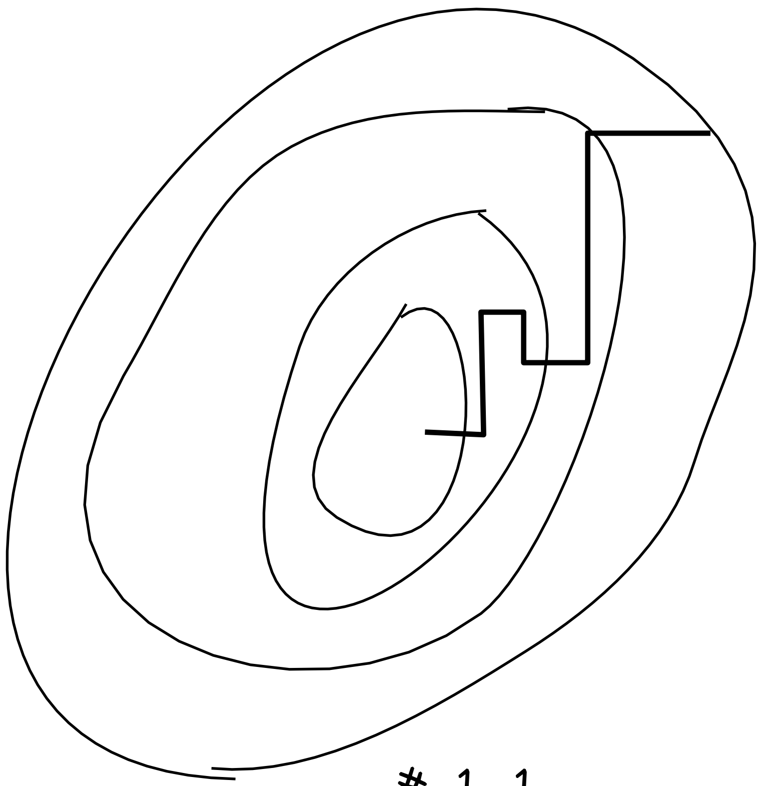
Blackbox step

linmin: Given as input the vectors $\mathbf{P}$ and $\mathbf{n}$ and the function $\mathbf{f}$, find the scalar l that minimizes $\mathbf{f}(\mathbf{P}+\lambda\mathbf{n})$. Replace $\mathbf{P}$ by $\mathbf{P}+\lambda\mathbf{n}$. Replace $\mathbf{n}$ by $\lambda\mathbf{n}$.

How to choose $\mathbf{n}$ and then λ?

Simple method:

Take the unit vectors $\mathbf{e}_1$, $\mathbf{e}_2$, ... $\mathbf{e}_n$ as a set of directions. Using **linmin**, find the minimum along $\mathbf{e}_1$. Start there and find the minimum along $\mathbf{e}_2$, ...



Conjugate (non-interfering) directions

How do you minimize along direction $\mathbf{u}$ and then minimize along $\mathbf{v}$ without affecting our minimization along $\mathbf{u}$?

Taylor series (up 1 dimension) wrt point $\mathbf{P}$

f($\mathbf{x}$) = f($\mathbf{P}$) + sum (partial(f wrt x_i)) x_i +
1/2 sum sum (partial f wrt $x_i x_j$) $x_i x_j$ + ...

$$\sim \; c - \langle \mathbf{b}, \mathbf{x} \rangle + \mathbf{x}^T \mathbf{A} \, \mathbf{x}$$

where c = f($\mathbf{P}$), b = $-\nabla$ (f) at $\mathbf{P}$, $\mathbf{A}$ is the Hessian matrix

Gradient vanishes (ie we're at an extremum) when $\nabla (f) = \mathbf{A} \, \mathbf{x} \quad \mathbf{b}$.

If we move by $\delta\mathbf{x}$, the gradient changes by

$$\delta(\nabla(f)) = \mathbf{A} \, \delta\mathbf{x}.$$

So, movement along $\mathbf{u}$ to a minimum followed by movement along $\mathbf{v}$ to a minimum preserves the minimum wrt $\mathbf{u}$ if

$$0 = \mathbf{u} \, \delta(\nabla(f)) = \mathbf{u} \, \mathbf{A} \, \mathbf{v}$$

In this case the directions $\mathbf{u}$ and $\mathbf{v}$ are said to be conjugate. If we follow n conjugate directions, we get to the minimum of a quadratic.

Powell's method

Produce N mutually conjugate directions
and get quadratic convergence.

1. Initialize a set of directions $\mathbf{u}_i = \mathbf{e}_i$, 1 = 1...N
2. Save your starting point as $\mathbf{P}_0$
3. For i = 1, ... , N, move $\mathbf{P}_{i-1}$ to the minimum along direction $\mathbf{u}_i$ and call this point $\mathbf{P}_i$
4. For i = 1, ... , N−1, set $\mathbf{u}_i = \mathbf{u}_{i-1}$
5. Set $\mathbf{u}_N = \mathbf{P}_N - \mathbf{P}_0$.
6. Move $\mathbf{P}_N$ to the minimum along direction $\mathbf{u}_n$ and call this point $\mathbf{P}_0$

When it works, it works fast,
It can end up with not enough independence
of directions.
Fix this by various heuristics.

Conjugate Gradient Methods

Simple counting...

Taylor series looks like

$$f(x) = c - \mathbf{b}\,\mathbf{x} + 1/2\ \mathbf{x}\,\mathbf{A}\,\mathbf{x} + \ldots$$

So, $O(N^2)$ variables. Previous method used N^2 line minimizations.

If we know gradient of f (vector of partials f wrt x_i's),
that's N pieces of info per step.
Can we use only O(N) line minimizations?

Sidebar: Why is O(N) better than $O(N^2)$?
Sidebar 2: Is is really N times faster?

Steepest descent method: Start at a point $\mathbf{P}_0$. As many times as needed, move from the point $\mathbf{P}_i$ to the point $\mathbf{P}_{i+1}$ by minimizing along the line from $\mathbf{P}_i$ in the direction of the local downhill gradient $-\nabla f(\mathbf{P}_i)$.

Steepest Descent

+ dates back to Cauchy
- performs many small steps to go down a long narrow valley (even for a quadratic valley)
+ can modify to find conjugate gradients

Suppose we knew $\mathbf{A}$ above.

Start with $\mathbf{g}_0 = \mathbf{h}_0$.

Update via

$$\mathbf{g}_{i+1} = \mathbf{g}_i - \lambda_i A\,\mathbf{h}_i \qquad \mathbf{h}_{i+1} = \mathbf{g}_{i+1} + \gamma_i \mathbf{h}_i$$

So $\mathbf{g}_i\,\mathbf{g}_j = 0$; $\mathbf{h}_i\,A\,\mathbf{h}_j = 0$; $\mathbf{g}_i\mathbf{h}_j = 0$; $i \mathrel{!=} j$

where λ_i and γ_i are given by

$$\lambda_i = (\mathbf{g}_i\,\mathbf{g}_i)/(\mathbf{h}_i\,A\,\mathbf{h}_i)$$

$$\gamma_i = (\mathbf{g}_{i+1}\mathbf{g}_{i+1})/(\mathbf{g}_i\,\mathbf{g}_i)$$

The directions $\mathbf{h}_i$ are the conjugate directions along which to line minimize.

If we are at $\mathbf{g}_i = -\nabla f(\mathbf{P}_i)$ for some $\mathbf{P}_i$. If we proceed in direction $\mathbf{h}_i$ to the local minimum

Suppose we are at $\mathbf{g}_i = -\nabla f(\mathbf{P}_i)$ for some $\mathbf{P}_i$.

Now, proceed in direction $\mathbf{h}_i$ to the local minimum of f located at some point $\mathbf{P}_{i+1}$

Set $\mathbf{g}_{i+1} = -\nabla f(\mathbf{P}_{i+1})$.

Now $\mathbf{g}_{i+1}$ is the same as we derived before but we got it without computing and storing $\mathbf{A}$