Intro to Scientific Computing: How long does it take to find a needle in a haystack?

Dr. David M. Goulet

Intro

Binary Sorting Suppose that you have a detector that can tell you if a needle is in a haystack, but not where in the haystack it is. You could divide the stack into two smaller stacks and use your detector to figure out which of them the needle is in. Being clever, you decide to repeat this process again and again until the stacks are small enough for you to see the needle yourself. How many times will you need to apply your stack-halving "algorithm" before you find the needle?

Root Finding Finding roots of polynomials is one of the fundamental problems in applied mathematics. You've probably seen the *quadratic* equation, $\frac{-b\pm\sqrt{b^2-4ac}}{2a}$, which gives the roots of the quadratic polynomial $p_2 = ax^2 + bx + c$. What you may never have seen are the *cubic* and *quartic* equations, which give formulae for the roots of $p_3 = ax^3 + bx^2 + cx + d$ and $p_4 = ax^4 + bx^3 + cx^2 + dx + e$. It is an interesting fact, one which required hundreds of years and the genius of Auguste Galois to prove, that there is no *quintic* formulae. In fact, Galois proved that there are no general formula for finding the roots of p_k when $k \geq 5$. Because of this, mathematicians have developed many clever ways to estimate the roots of polynomials.

The Connection It may seem odd, but a certain class of methods for finding roots of polynomials is directly related to the needle problem. Each of these problems can be approached using what are called iterative numerical methods. Below we'll develop some of these methods, show how they can be used to solve important problems, and show how they can be efficiently implemented on a computer using Matlab.

Sorting

The Bisection Method

Suppose you have 16 objects and you want to separate them. One way to do this is to split them into piles of 8 and then split each of these into piles of 4 and each of those into piles of 2, etc. In this way, you can separate your 16 objects in 4 steps.

$$(16)$$
$$(8, 8)$$
$$(4, 4, 4, 4)$$
$$(2, 2, 2, 2, 2, 2, 2, 2)$$
$$(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$$

In a similar way, you could separate a group of 2^n objects in n steps.

If you start with a group of k objects, even if k is not a power of 2, you can apply a slight variation of this algorithm to separate your objects. If the number of objects in a group is even, then you can just split the group in half. If the number is odd, then you can put an extra one in either pile. For example

$$(13)$$
$$(6, 7)$$
$$(3, 3, 3, 4)$$
$$(1, 2, 1, 2, 1, 2, 2, 2)$$
$$(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$$

In this way, a set of k objects can be separated in $1 + [\log_2 k]$ steps. Here $[x]$ is the largest integer $\leq x$.

Problems

1. How many steps does it take to separate 32 objects into groups of 4?

2. How many steps does it take to separate 23 objects into groups of less than 3?

3. If your haystack is originally a 10 meter cube, and you can't see the needle until the stack is a 1 centimeter cube, how many steps will the bisection method require?

4. If you have a set of spheres, cubes and pyramids each in three different colors, how many steps of the algorithm will guarantee that they are sorted into groups of the same geometry and color?

5. A king is placed somewhere on a chess board but you are not allowed to look at the board. You are allowed to pick a vertex and this vertex is defined as the origin of a coordinate system. You are then told which quadrant the king lies in. How many iterations of this "quadsection" method will guarantee that you find the king.

Finding Roots

The Intermediate Value Theorem

says that if you have a continuous function $f(x)$ on the interval $x \in [a, b]$ and that $f(a) < n$ and $f(b) > n$ then there is some number c in the interval (a, b) such that $f(c) = n$. In other words, if a continuous function takes on values above and below some number, n, then it also takes on the value n.

For example, $f(x) = 2x - 6$, has the properties that $f(-1) = -8$ and $f(10) = 14$, so there is some number, c, in the interval $[-1, 10]$ so that $f(c) = 2$. In this case it is easy to show that $f(4) = 2 * 4 - 6 = 2$.

Problems

1. Use the theorem to show that $f(x) = x^2 - 1$ has a root somewhere in the interval $[0, 3]$.

2. Use the theorem to show that $f(x) = e^x - (x + 2)\cos(x)$ has a root somewhere in the interval $[0, \pi/2]$.

3. If a runner finishes a 10 mile race in 60 minutes, was there a 1 mile section that the runner ran in exactly 6 minutes?

The Bisection Method

Consider the polynomial $p(x) = x^2 - 1$. Notice that $p(0) = -1$ and that $p(3) = 8$. By the intermediate value theorem, there is some $c \in [0, 3]$ so that $p(c) = 0$. We can use the bisection method to approximate this root as accurately as we want. The idea is to cut the interval in half and apply the intermediate value theorem to each half, then repeat.

So, we split the interval into $[0, 3/2]$ and $[3/2, 3]$. Notice that $p(3/2) = 5/4 > 0$, so there is a root on $[0, 3/2]$ but not necessarily on $[3/2, 3]$. So we focus on $[0, 3/2]$ and split it again, into $[0, 3/4]$ and $[3/4, 3/2]$. Because $p(3/4) = -8/16 < 0$, we know that there is a root on $[3/4, 3/2]$.

Notice that at each step the interval of interest is only half the size of the previous interval. So we are slowly closing in on the answer. If the original interval has length L, then after n steps the interval has length $2^{-n} L$. We may never arrive at the exact solution this way, but we can get closer and closer. In the example above, the interval is $[3/4, 3/2]$. The exact solution then must be close to the middle of this interval, $9/8$. How close? The error can be no more than half the length of the interval. So our estimate is $9/8 \pm 3/8$.

If we continue this process for 2 more steps, we get the estimate $33/32 \pm 3/32$. The error is now less than $1/10$ and our estimate is 1.03125. This is close to the exact answer of 1. In general, if we start with an interval of length L, then after n steps the interval will have length $2^{-n} L$ and our estimate will be the midpoint of this interval with an error no greater than $2^{n+1} L$.

Example Notice that $f(x) = x - \sin^2 x$ has $f(-\pi/2) = -\pi/2 - 1 < 0$ and $f(\pi) = \pi > 0$. So there is a root somewhere in $(-\pi/2, \pi)]$. We can use the bisection method to estimate this root accurate to 2 decimal places. The original interval has length $3\pi/2$ so we need to choose n, the number of steps, so that $2^{-n}(3\pi/2) < 0.01$. We find n=9. Below is the sequence of midpoints of the intervals.

So we know that 0.0031 is an approximation of the solution accurate to within 0.01. This agrees with the exact answer, which we know is 0.

Problems

1. If $f(0) = -1$ and $f(8) = 1$, how many steps of the bisection method

0.7854...
-0.3927...
0.1963...
-0.0982...
0.0491...
-0.0245...
0.0123...
-0.0061...
0.0031...

will be required to find an approximation to the root of $f(x)$ accurate to 0.25?

2. You've built a machine that shoots darts at a target. By aiming it at an angle $\theta = \pi/3$, your first dart lands 10cm above the center of the bulls eye. You adjust the angle to $\theta = \pi/4$, and your second dart lands 10cm below the center of the bulls eye. Can the bisection method be used to find the correct angle? Can we say with certainty how many steps will be required?[1]

Numerical Methods

The bisection method can be used to find roots of functions by following an algorithm. Consider the function $f(x)$.

1. Find two values, a and b with $a < b$, so that $f(a)$ and $f(b)$ have opposite signs.

2. Define the midpoint of the interval $m = (a + b)/2$.

3. If $f(m)=0$, then stop because you've found a root. If $f(m)$ has the same sign as $f(a)$, then there is a root on (m, b). If it has the same sign as $f(b)$, then there is a root on (a, m).

4. Redefine a and b to be the endpoints of the new interval of interest. Repeat the above steps until the interval is as small as you desire.

---

[1]This relates to solving a certain type of differential equation problem. In that context it is called, not surprisingly, a "shooting method."

We can write this as a computer algorithm in Matlab. For example, if $f(x) = 2x$ and $a = -2$ and $b = 1$, then we use the following code to find an approximation of the root accurate to 0.1.

```
a=-2;
b=1;
error=0.1;
while(b-a>2*error)
     m=(a+b)/2;
     f=2*m;
     if f=0
          a=m;
          b=m;
     elseif a*f<0
          b=m;
     else
          a=m;
     end
end
```

Or, for the example above with $f(x) = x - \sin^2 x$ on $[-\pi/2, \pi]$ if we want an approximation accurate to 0.01 we use the following code.

```
a=-pi/2;
b=pi;
error=0.01;
while(b-a>2*error)
     m=(a+b)/2;
     f=m-sin(m)^2;
     if f=0
          a=m;
          b=m;
     elseif a*f<0
          b=m;
     else
          a=m;
     end
end
```

Problems

1. Approximate a root of the polynomial $x^5 - 3x^4 - 6x^3 + 18x^2 + 8x - 24$ with an accuracy of at least 0.0001. From this, guess what the exact value is and then use this information to factor the polynomial. Then apply the algorithm again to find another root. Can you find all five roots this way?

2. Obviously $f(x) = x^4$ has a root at $x = 0$. Can the bisection method approximate this root?

3. The bisection method does not help when searching for the roots of $f(x) = x^2 + 1$. Why? Can you think of a way to modify the bisection method to find roots of $f(x)$?
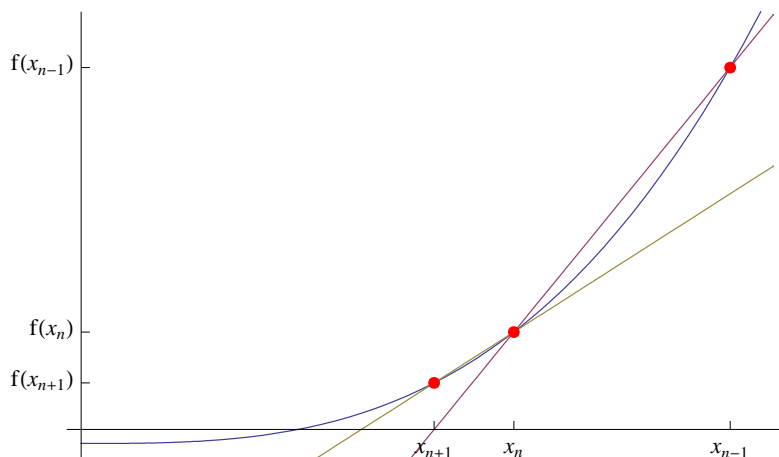
1   Newton's Method

Figure 1: The geometry of Newton's method.

Suppose that you make two initial guesses for the location of the root of $f(x) = 0$. Call these guesses x_1 and x_0. The figure suggests that if you draw a straight line through the points $(x_0, f(x_0))$ and $(x_1, f(x_1))$ then this new point x_2 will be closer to the root. The equation of the line through the first

two guesses is

$$y = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_1) + f(x_1)$$

Setting $y = 0$ allows us to solve for x_2

$$0 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_1) + f(x_1)$$

or

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}$$

This suggests an iterative scheme for finding roots.

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

Here is an example of using this algorithm to find the root of $f(x) = x^3 - 1$ with initial guesses $x_1 = 2$ and $x_2 = 0$.

```
x(1)=2;
x(2)=0;
error=1;
n=2;
while (error>0.01)&(n<100)
    x(n+1)=x(n)-(x(n)^3-1)*(x(n)-x(n-1))/(x(n)^3-x(n-1)^3);
    n=n+1;
    error=abs(x(n)^3-1)
end
```

This algorithm proceeds until $f(x) \leq 0.01$ or until it has made 100 guesses, which ever comes first. The reason for putting a limit on the number of guesses is that the algorithm may not converge and so may not terminate. Note: If you've learned Calculus, then you'll understand why

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

is another useful iterative scheme. In this form, it is known as *Newton's Method*.
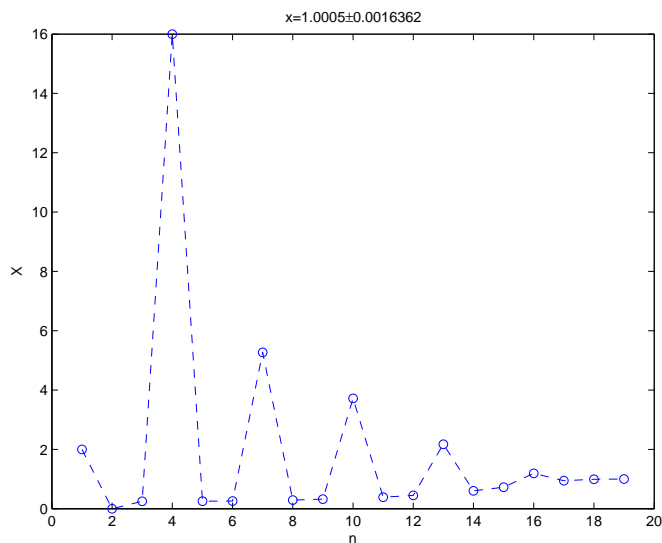
Figure 2: The convergence of Newton's method for the root of $x^3 - 1 = 0$

1. Does this method appear to converge for $f(x) = x^2 - 1$, $x_0 = 3$, and $x_1 = 2$?

2. Can you think of a way to apply these ideas to higher dimensional problems? Apply your idea to finding the root of $f(x, y) = x^2 + y^2$.

1.1 Convergence

What are some reasons that this scheme may or may not converge?

Converge Firstly, we can write

$$|x_{n+1} - x_n| = |x_n - x_{n-1}| \left| \frac{f(x_n)}{f(x_n) - f(x_{n-1})} \right|$$

If we could could somehow be sure that the second term is always < 1, then we have a contraction mapping. A decreasing sequence, bounded below, converges. So $\lim_{n \to \infty} |x_{n+1} - x_n|$ exists. Moreover,

$$|x_{n+1} - x_n| \leq r|x_n - x_{n-1}| \leq r^2|x_{n-1} - x_{n-2}| \leq \ldots \leq r^n|x_1 - x_0|$$

with $r < 1$. Therefor, the series $x_n = \sum_{i=0}^{n-1}(x_{i+1} - x_i)$ converges absolutely, and so converges. Therefor, in this case, $\lim_{n\to\infty} x_n$ exists.

Diverge

- Alternatively, suppose that this quantity is always > 1 then our mapping is an expansion, and so guaranteed to not converge.

- Suppose the algorithm gives guesses so that $f(x_n) = f(x_{n-1})$. Division by zero is not defined, so the algorithm fails.
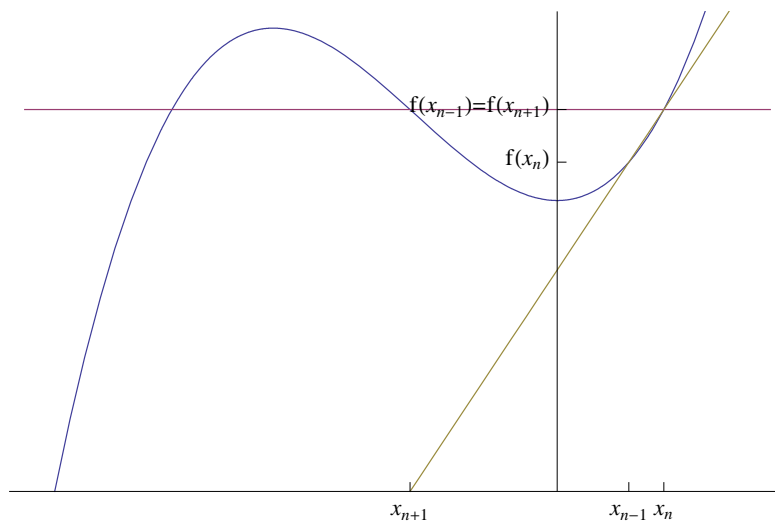

Figure 3: The failure of Newton's method.

1. Does this method appear to converge for $f(x) = x^2 - \sin^2 x$, $x_0 = -1$, and $x_1 = 2$?

2. Does this method appear to converge for $f(x) = e^{-x^{-2}}$, $x_0 = -1$, and $x_1 = 2$? What about $x_0 = 1$ and $x_1 = 2$?

1.2 If you know some Calculus and physics. . .

A pendulum problem   The equation of motion of a pendulum is

$$\frac{d^2\theta}{dt^2} + (g/l)\sin\theta = 0$$

where g is the acceleration of gravity, l is distance of the center of mass from the pivot point, and θ is the angular deviation from the downward rest position. We have neglected all types of friction and other dissipative forces.

Suppose that you are asked to give the pendulum a push from its rest position so that it reaches its highest point in exactly τ seconds. In this case you are solving

$$
\begin{aligned}
\theta'' + (g/l)\sin\theta &= 0 \\
\theta(0) &= 0 \\
\theta'(0) &= \omega
\end{aligned}
$$

and then trying to figure out what the correct value of ω should be so that $\theta'(\tau;\omega) = 0$. This again can be solved by Newton's method. Suppose that we have the solution to the pendulum equation $\theta(t,\omega)$. Then we want to know for what value of ω does $\theta'(\tau;\omega) = 0$. Calling the left hand side $f(\omega)$ we can apply Newton's method.

The problem is, that we can't solve for θ explicitly, so this proposal isn't going to work. What can we do? You'll have to take the following on faith, but the best we can do is say that when it reaches its highest point we have

$$\int_0^\theta \frac{ds}{\sqrt{\omega^2 + (2g/l)(\cos(s) - 1)}} = \tau$$

and

$$\omega^2 + (2g/l)(\cos(\theta) - 1) = 0$$

1. How can you combine these two ideas together with Newton's method to find the correct value of ω?

2. Given the plot of $f(x) = \int_0^x \frac{ds}{\sqrt{\cos(s) - \cos(x)}}$, do you think Newton's method will converge? It may help to know that $f(0^+) = \frac{149}{60\sqrt{2}} \approx 1.756$
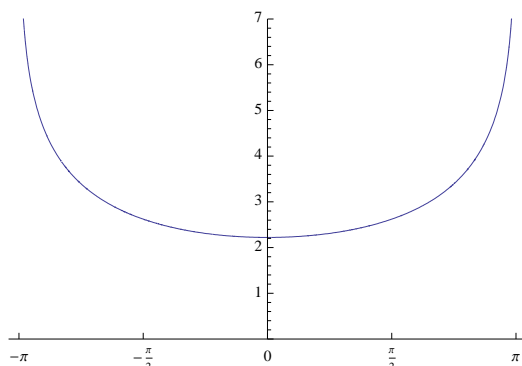
11

Figure 4: The function $f(x) = \int_0^x \frac{ds}{\sqrt{\cos(s)-\cos(x)}}$.

If you like, you can apply Newton's method to this problem using an integrator that I wrote. After creating a new m-file with the following function in it, you can use $fintegral(x, 10000)$ to get 10 digits of accuracy for the function $f(x)$. Incorporate this into your Newton's method routine. (If you understand how to use these strange computers, you can download the m-file instead of retyping it[2].

Be warned that the way this is coded, you can't choose values for x outside of $(0, \pi)$. Do you think that this will matter when using Newton's method?

```
function y=fintegral(x,n)

dt=x/n;
t=0:dt:x;
sum=0;
for i=1:length(t)-2
    sum=sum+.5*dt*( (cos(t(i))-cos(x))^(-.5) -((x-t(i))*sin(x))^(-.5) - ...
        .25*cot(x)*((x-t(i))/sin(x))^.5 + (cos(t(i+1))-cos(x))^(-.5) ...
        -((x-t(i+1))*sin(x))^(-.5) - .25*cot(x)*((x-t(i+1))/sin(x))^.5 );
end
y=sum+2*sqrt(x/sin(x))+(1/6)*cos(x)*(x/sin(x))^1.5 + ...
    .5*dt*( (cos(t(end-1))-cos(x))^(-.5) - ((x-t(end-1))*sin(x))^(-.5) - ...
    .25*cot(x)*((x-t(end-1))/sin(x))^.5);
```

[2]http://www.math.utah.edu/~ goulet/classes/mathcircle/fintegral.m