comp20005 Engineering Computation

Additional Notes

Numeric Computation, Part B

Lecture slides prepared by Alistair Moffat

Root finding

Given a continuous function $f()$, determine the set of values x such that $f(x) = 0$.

Why? Function often represents a point of balance between to opposing constraints (for example, gravity and frictional resistance on falling object at terminal velocity).

Also required to find maxima and minima. Turning points have zero derivative.

Three main methods:

- graphical methods;
- bracketing methods; and
- open methods.

Graphical methods

Plot the graph, and look for the roots.

Can use a graphing calculator, even if it can't identify the exact roots.

Knowledge of general shape of function, and imprecise knowledge of roots is of benefit to more principled methods.

Bisection method

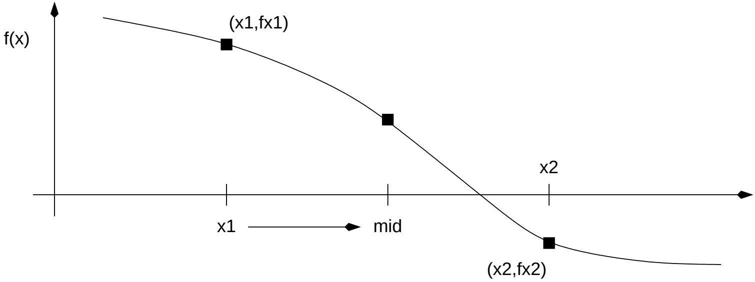
A better algorithm uses bisection to rapidly locate an approximation of the root.

Start with a pair of values x_1 and x_2 for which $f(x_1) \times f(x_2) < 0$.

At each step the mid-point $x_m = (x_1 + x_2)/2$ of the current range is tested. Keep root sandwiched between two x values at which $f(x)$ has opposite signs, by moving either x_1 or x_2 to x_m.

- `bisection.c`

Bisection method



Bisection method

The function judges termination by two different conditions:

- Whether the root has been found to the required precision (and not by a test of the form `f(x)==0.0`), and
- Whether some preset number of iterations has been exceeded.

The latter is necessary to prevent endless non-converging computation caused by a too-tight tolerance on the approximation.

In floating point arithmetic, there are no continuous functions, and every function is discrete.

Bisection example

If $f(x) = x^2 - 10$, and $x_0 = 0$ and $x_1 = 10$, the sequence of midpoint estimates x_m is

$$5, 2.5, 3.75, 3.125, 3.4375, 3.28125, 3.203125, 3.1640625$$

The correct value is $x = \sqrt{10} = 3.16227766016$.

This form of calculation can also be easily done with a spreadsheet.

Bisection convergence

Each iteration adds one bit of accuracy, or a little under one third of a decimal digit. This observation allows the number of loop iterations to be bounded.

So 32 iterations will give an accurate `float`, and 64 iterations an accurate `double`.

Bisection is robust and reliable, provided an initial interval is available. Can do this via iterative search through through x values, stepping with some suitable interval.

False position

Rather than use midpoint, could also use proportions and triangles to get a "better" next estimate,

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

This is the method of false position.

Need to be a bit more careful with termination test, best bet is to use `fabs(f(xm))<EPSILON`, because `x2-x1` might not converge.

May require fewer loop iterations than bisection to reach similar level of precision.

Convergence

In practice convergence speed is not a problem, 50 or 100 loop iterations is nothing.

But both bisection and false position are unreliable (or may fail completely) when multiple roots fall within preliminary inspection interval.

So it make sense to also look at options that do not require an initial bracketing interval.

Secant method

If the "bracketing" requirement of the false position method is relaxed, we get the secant method, with two points on the curve used to fit a straight line, and the intersection of that line with the x-axis computed to provide a better estimate of the root:

$$x_{k+1} = \frac{f(x_k)x_{k-1} - f(x_{k-1})x_k}{f(x_k) - f(x_{k-1})}$$

When x_k and x_{k-1} are close together, the gradient of the line through them approximates the tangent $f'(x_k)$.

Newton-Raphson

If the derivative of the function in the vicinity of the root is known, it can be used to directly compute the gradient of the tangent to the curve.

Starting with x_0 as an estimate of the true value x, compute a sequence of better estimates via:

$$x_{i+1} = \mathrm{NR}(x_i) = x_i - \frac{f(x_i)}{f'(x_i)} \, .$$

This approach requires that f is differentiable, and that the derivative is well-behaved over the interval in question.

Newton-Raphson

The number of correct digits approximately doubles at each iteration, provided certain conditions are met.

Gives faster convergence than bisection method, but also has greater risks – when $f'(x) \approx 0$, or when $f'(x_i) = 0$ for any x_i, may have serious problems.

Newton-Raphson

Suppose that $f(x) = x^2 - 10$, and $\sqrt{10}$ is to be computed.

In this case, $f'(x) = 2x$, and hence

$$\mathrm{NR}(x) = x - \frac{x^2 - 10}{2x} = \frac{1}{2}\left(x + \frac{10}{x}\right).$$

This yields the sequence (assuming $x_0 = 1$)

$$1.0, 5.5, 3.66, 3.196, 3.16246, 3.1622777, 3.16227766\,.$$

(And the correct value is $x = \sqrt{10} = 3.162277660168379$.)

Multiple simultaneous equations

Systems of equations also sometimes arise:

$$
\begin{aligned}
f_1(x_1, x_2, x_3, \ldots, x_n) &= 0 \\
f_2(x_1, x_2, x_3, \ldots, x_n) &= 0 \\
f_3(x_1, x_2, x_3, \ldots, x_n) &= 0 \\
&\cdots \\
f_n(x_1, x_2, x_3, \ldots, x_n) &= 0
\end{aligned}
$$

where the "roots" are vectors of n values.

Multiple simultaneous equations

Example:

$$\begin{aligned} \sin x_1 - \cos x_2 &= 0 \\ x_1 + x_2 - 1 &= 0 \end{aligned}$$

has solution $(x_1, x_2) = (-1.85619, 2.85619)$.

A multidimensional version of NR can be used in a similar iterative approach, making use of partial derivatives and hyperplanes.

Root finding – summary

For simple well-behaved situations, use NR or bisection, both are fine. NR is faster, but requires that the derivative be available. Difference in speed is likely to be inconsequential.

Where there may be multiple roots: beware, both methods may have issues.

Turning points near the root being sought can give divergent outcomes with NR; and multiple roots can affect both approaches. So make sure you know the nature of the function first.

Numerical integration

Given some continuous function $f(x)$ and two limits, x_1 and x_2, compute

$$\int_{x_1}^{x_2} f(x)dx\,.$$

Useful in a range of situations where volumes of objects are required, total energy in dynamic systems, and etc.

One trapezoid

As a crude approximation, assume that $f(x)$ is linear between the points $(x_1, f(x_1))$ and $(x_2, f(x_2))$, and compute the area of the trapezoid that is defined:

$$I = (x_2 - x_1)\frac{f(x_1) + f(x_2)}{2} .$$

But the error can be arbitrarily large. Consider $f(x) = -x^2 + k$, with $x_1 = -\sqrt{k}$ and $x_2 = \sqrt{k}$.

Trapezoidal method

To get a better estimate, break the interval up into n steps each of size $h = (x_2 - x_1)/n$. Then the area of each small trapezoid is worked out, and added to a total.

$$
\begin{aligned}
I &= \sum \text{trapezoids} \\
&= \sum_{i=0}^{n-1} h \cdot \frac{f(x_1 + h \cdot i) + f(x_1 + h \cdot (i+1))}{2} \\
&= \frac{h}{2} \left(f(x_1) + 2 \sum_{i=1}^{n-1} f(x_1 + h \cdot i) + f(x_1 + h \cdot n) \right) .
\end{aligned}
$$

Trapezoidal method

The greater the value of n, the smaller the value of h, and the better the approximation.

In theory, that is.

In practice, rounding errors arising from summation of many values of comparable magnitude needs to be monitored. Techniques for progressive aggregation might be helpful.

And note that multiplication by h is factored out and only performed once, avoiding another possible source of rounding error.

Simpson's method

The trapezoidal approach fits a linear polynomial to pairs of points. If n is even (or is doubled to make it even), can fit a quadratic to consecutive triples of points.

It turns out that if x_1, $x_1 + h$, and $x_1 + 2h$ are evenly spaced, and if $f(x) = ax^2 + bx + c$ is fitted through $(x_1, f(x_1))$, $(x_1 + h, f(x_1 + h))$, and $(x_1 + 2h, f(x_1 + 2h))$, then

$$\int_{x_1}^{x_1+2h} f(x)dx = \frac{2h}{6}\left(f(x_1) + 4f(x_1 + h) + f(x_1 + 2h)\right) .$$

Simpson's method

Hence, Simpson's method for numerical integration:

$$
\begin{aligned}
I &= \frac{h}{3}\Bigg(f(x_1) + f(x_1 + h \cdot n) + \\
&\qquad 4 \sum_{i=1,3,5,\ldots}^{n-1} f(x_1 + h \cdot i) + 2 \sum_{i=2,4,6,\ldots}^{n-2} f(x_1 + h \cdot i)\Bigg) .
\end{aligned}
$$

The improved approach has significantly better convergence properties. For a given level of mathematical error it allows use of a smaller n, meaning that rounding errors are less likely to intrude.

Integration – summary

Both trapezoidal and Simpson's method are easy to implement, but Simpson's method is more accurate.

Watch out for little numbers being added to big numbers, arrange the summation so that the little numbers don't get swamped.