NUMERICAL METHODS WEEK 7

SOLUTIONS TO NONLINEAR EQUATIONS

Learning outcomes:

- Understand and use Root Finding algorithms to solve nonlinear equations.
- Understand optimization problems.

MATT WATKINS MWATKINS@LINCOLN.AC.UK

NON-LINEAR EQUATIONS

These are all the equations that are not linear. A linear function satisfies

$$f(x+y) = f(x) + f(y)$$

and

$$f(\alpha x) = \alpha f(x)$$

which imply the superposition principle

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

An equation written

$$f(x) = \mathrm{C}$$

is linear if $f(x)$ is a linear function. Otherwise it is a nonlinear equation.

$$x^2 + x - 1 = 0$$

The operations that map x to $f(x)$ can be very general, including, for instance differentiation.

ROOT FINDING ALGORITHMS

In the simplest case we want to find the solutions of

$$f(x) - \mathrm{C} = 0$$

In a few cases this can be done analytically - but generally we want an $iterative$ algorithm that will converge to a root.

OPTIMIZATION

If we want to find minima (or maxima) of a function f(x) we need to find solutions of

$$f'(x) = 0$$

This is clearly the same problem as before - but we are now working with the derivative of the function.

ROOT FINDING ALGORITHS

There are many algorithms, often small modifications of each other.

Most well known are

- Bisection
- Newton-Raphson
- Secant

Others are often combinations of the previous, aiming to keep their strengths but mitigate weaknesses. For instance the Brent algorithm is a combination of Bisection, Secant and interpolations.

BISECTION

Simplest possible algorithm.

Let f be a continuous function. Find an interval $[a, b]$ such that $f(a)$ and $f(b)$ have opposite signs (a bracket).

Let $c = (a + b)/2$ be the middle of the interval.

Then either $f(a)$ and $f(c)$, or $f(c)$ and $f(b)$ have opposite signs, and one has divided by two the size of the interval.

Rinse and repeat

Although the bisection method is robust, it gains one and only one bit of accuracy with each iteration. Other methods, under appropriate conditions, can gain accuracy faster.

NEWTON'S METHOD

This looks just like our linear interpolation (https://mattatlincoln.github.io/teaching/numerical_methods/lecture_4/#/5). If we have a continuous curve $y = f(x)$ the equation of the tangent to the curve at the point x_n is:

$$y = f'(x_n)\,(x - x_n) + f(x_n),$$

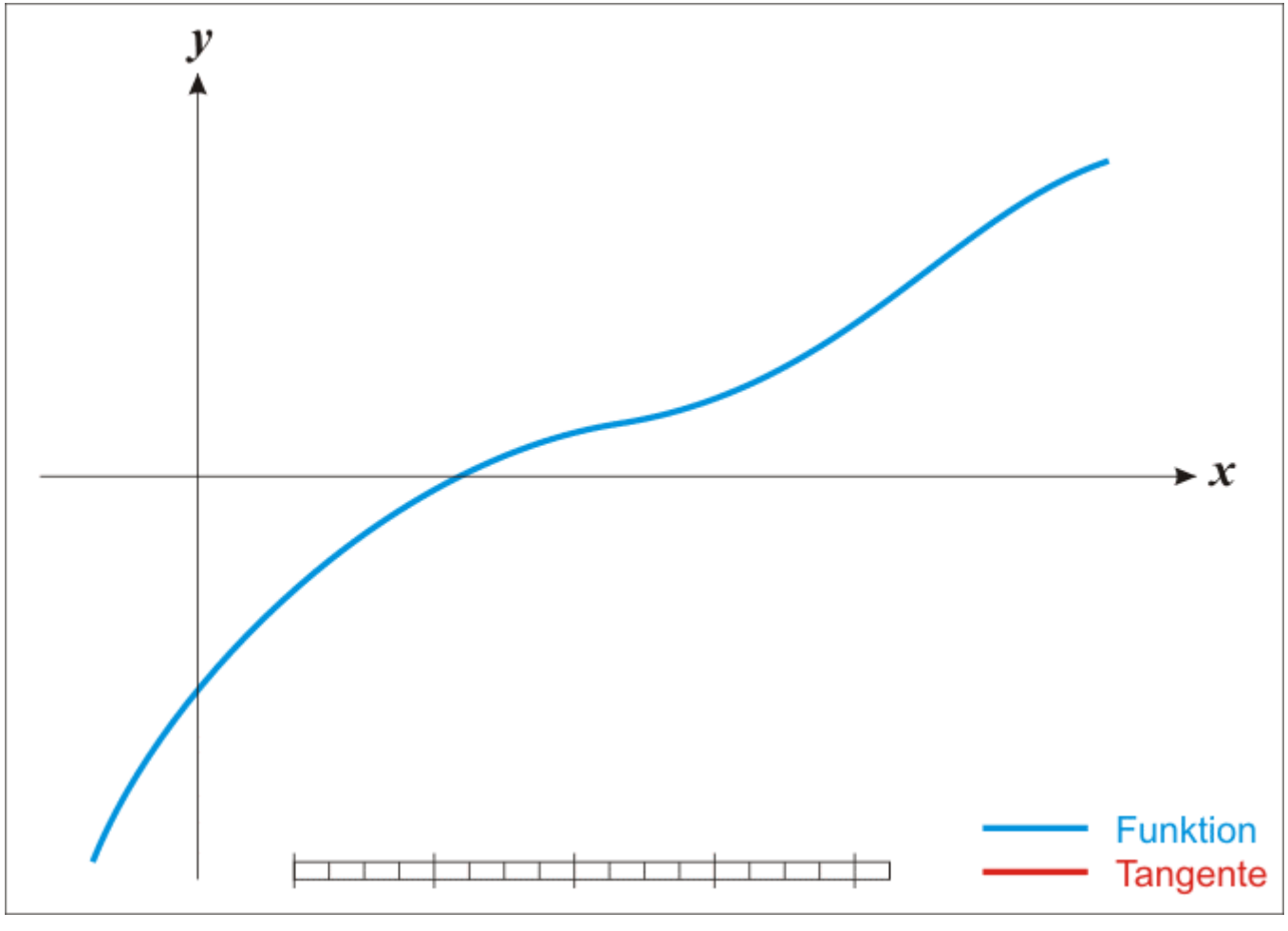
If we want an estimate of the value where this cuts the x-axis we look for the value of $x = x_{n+1}$ that satisfies

$$0 = f'(x_n)\,(x_{n+1} - x_n) + f(x_n)$$

Which implies that

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

NEWTON'S METHOD



Animation taken from wikipedia (https://en.wikipedia.org/wiki/Newton%27s_method)

ROOT FINDING

- Implement Newton's Method and use it to find the following.
- Find solutions of $x^2 = 612$
- Find solutions of $x^3 = \cos x$

SECANT METHOD

The Secant method is essentially the same as Newton's method - but instead of calculating $f'(x)$ analytically, it is approximated by finite difference $f'(x) \approx \frac{f(x+\Delta x)-f(x)}{(x+\Delta x)-x}$

Plugging into Newton's method we get

$$x_n = x_{n-1} - f(x_{n-1})\frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})} = \frac{x_{n-2}f(x_{n-1}) - x_{n-1}f(x_{n-2})}{f(x_{n-1}) - f(x_{n-2})}$$

Note that you need to choose two (close) starting guesses, x_0 and x_1 before you can iterate the algorithm.

WHAT IS THE POINT?

Sometimes it is difficult or expensive to evaluate $f'(x)$

Related 'quasi-Newton' methods are very common for multidimensional optimization.

- Repeat the previous questions using the Secant method. How do the number of iterations compare?

OPTIMIZATION

Newton's method for optimization is found by minimizing the quadratic approximation

$$q(x_{k+1}) = f(x_k) + f'(x_k) \cdot (x_{k+1} - x_k) + \frac{1}{2} f''(x_k) \cdot (x_{k+1} - x_k)^2$$

where $q(x_{k+1})$ is the estimated value of $f(x_{k+1})$ and we are currently at x_k.

- Derive Newton's Method for optimization by finding the value of x (call it $x^{(k+1)}$) that minimizes $q(x)$
- Implement Newton's Method for optimization and use it to find minima of the following functions.
  - $f(x) = x^2 - \cos x$,
  - $f(x) = x^4 - 14x^3 + 60x^2 - 70x$

ROOT FINDING IN HIGHER DIMENSIONS

Newton's method or modified Newton schemes can be readily extended to higher dimensions.

One may also use Newton's method to solve systems of k (nonlinear) equations, which amounts to finding the zeroes of continuously differentiable functions $F : \mathbb{R}^k \to \mathbb{R}^k$ - i.e. a vector is mapped to a vector.
So we have $\mathbf{x}^T = (x_0, x_1 + \ldots + x_{k-1})$ and for each $\mathbf{x}$ we have $\mathbf{F}(\mathbf{x})^T = (F_0, F_1 + \ldots + F_{k-1})$.

So instead of our linear approximation for the 1D case

$$f(x + \Delta x) = f(x) + f'(x)\,(\Delta x) = 0,$$

we have a similar equation for each component of, F_i

$$F_i(\mathbf{x} + \mathbf{\Delta x}) = F_i(\mathbf{x}) + \sum_{i=0}^{i=k-1} \frac{\delta F_i(\mathbf{x})}{\delta x_i} \Delta x_i = 0,$$

We can write these k equations in matrix form:

$$\begin{pmatrix} F_0(\mathbf{x} + \mathbf{\Delta x}) \\ F_1(\mathbf{x} + \mathbf{\Delta x}) \\ \vdots \\ F_{k-1}(\mathbf{x} + \mathbf{\Delta x}) \end{pmatrix} = \begin{pmatrix} F_0(\mathbf{x}) \\ F_1(\mathbf{x}) \\ \vdots \\ F_{k-1}(\mathbf{x}) \end{pmatrix} + \begin{pmatrix} \frac{\delta F_0(\mathbf{x})}{\delta x_0} & \frac{\delta F_0(\mathbf{x})}{\delta x_1} & \cdots & \frac{\delta F_0(\mathbf{x})}{\delta x_{k-1}} \\ \frac{\delta F_1(\mathbf{x})}{\delta x_0} & \frac{\delta F_1(\mathbf{x})}{\delta x_1} & \cdots & \frac{\delta F_1(\mathbf{x})}{\delta x_{k-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\delta F_{k-1}(\mathbf{x})}{\delta x_0} & \frac{\delta F_{k-1}(\mathbf{x})}{\delta x_1} & \cdots & \frac{\delta F_{k-1}(\mathbf{x})}{\delta x_{k-1}} \end{pmatrix} \begin{pmatrix} \Delta x_0 \\ \Delta x_1 \\ \vdots \\ \Delta x_{k-1} \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

This is just a set of equations of the form $\mathbf{A}x = b$ which we can solve to find Δx. Then we find our new $\mathbf{x}$ vector as $\mathbf{x} + \mathbf{\Delta x}$.

The matrix of derivatives is called the Jacobian, $J(\mathbf{x})$. You can also solve the equations by finding the inverse of J.

OPTIMIZATION IN HIGHER DIMENSIONS

Newton's method or modified Newton schemes for optimization can be readily extended to higher dimensions.

You can find minima of a scalar field, $f(\mathbf{x})$,using the same multidimensional Newton algorithm.

If you have a scalar field $f(x_1, x_2, \ldots, x_n)$ then $\mathrm{grad}(f(\mathbf{x})) = \nabla f(\mathbf{x})$ is a series of equations that should be equal be equal to zero at a minimum (extrema). You can then use the algorithm on the previous slide to find the zeros of the grad of f.

$$x_{n+1} = x_n - J_{\nabla f}(x_n)^{-1} \nabla f(x_n)$$

The Jacobian of the grad of a scalar field is often called the Hessian and contains the mixed 2nd order partial derivatives of the field.

so to minimize $f(\mathbf{x})$ we have to solve

$$\begin{pmatrix} \frac{\delta^2 f(\mathbf{x})}{\delta x_0 \delta x_0} & \frac{\delta^2 f(\mathbf{x})}{\delta x_0 \delta x_1} & \cdots & \frac{\delta^2 f(\mathbf{x})}{\delta x_0 \delta x_{k-1}} \\ \frac{\delta^2 f(\mathbf{x})}{\delta x_1 \delta x_0} & \frac{\delta^2 f(\mathbf{x})}{\delta x_1 \delta x_1} & \cdots & \frac{\delta^2 f(\mathbf{x})}{\delta x_1 \delta x_{k-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\delta^2 f(\mathbf{x})}{\delta x_{k-1} \delta x_0} & \frac{\delta^2 f(\mathbf{x})}{\delta x_{k-1} \delta x_1} & \cdots & \frac{\delta^2 f(\mathbf{x})}{\delta x_{k-1} \delta x_{k-1}} \end{pmatrix} \begin{pmatrix} \Delta x_0 \\ \Delta x_1 \\ \vdots \\ \Delta x_{k-1} \end{pmatrix} = \begin{pmatrix} -\frac{\delta f(\mathbf{x})}{\delta x_0} \\ -\frac{\delta f(\mathbf{x})}{\delta x_1} \\ \vdots \\ -\frac{\delta f(\mathbf{x})}{\delta x_{k-1}} \end{pmatrix}$$

SUMMARY AND FURTHER READING

You should be reading additional material to provide a solid background to what we do in class

All the textbooks contain sections on root finding and solving non-linear equations, for instance chapter 9 of Numerical Recipes (http://www.nrbook.com/a/bookcpdf.php).