Systems of non-linear equations

Say we have 2 equations :- $f(x,y) = 0$, $g(x,y) = 0$

This is basically a root finding problem, and we will employ Newton's method.

$\therefore$ We will use iteration to solve for $x_0, x_1, \dots x_{n+1}$

$y_0, y_1, \dots y_{n+1}$

where x_{n+1}, y_{n+1} have converged to the roots.

<u>Take Taylor series :</u>

$$f(x_{n+1}, y_{n+1}) = f(x_n, y_n) + \overset{\Delta x_n}{(x_{n+1} - x_n)} f_x(x_n, y_n) + \underset{\Delta y_n}{(y_{n+1} - y_n)} f_y(x_n, y_n)$$

partial derivative w.r.t x evaluated at x_n, y_n

partial derivative w.r.t x evaluated at x_n, y_n

[higher order terms ignored]

and, $$g(x_{n+1}, y_{n+1}) = g(x_n, y_n) + \overset{\Delta x_n}{(x_{n+1} - x_n)} g_x(x_n, y_n) + \underset{\Delta y_n}{(y_{n+1} - y_n)} g_y(x_n, y_n)$$

[higher order terms ignored]

but since we need to find roots, $f(x_{n+1}, y_{n+1}) = 0$ and $g(x_{n+1}, y_{n+1}) = 0$

$$\therefore (x_{n+1} - x_n) f_x(x_n, y_n) + (y_{n+1} - y_n) f_y(x_n, y_n) = -g(x_n, y_n)$$

$$(x_{n+1} - x_n) g_x(x_n, y_n) + (y_{n+1} - y_n) g_y(x_n, y_n) = -g(x_n, y_n)$$

or, $$\Delta x_n f_x(x_n, y_n) + \Delta y_n f_y(x_n, y_n) = -f(x_n, y_n)$$

$$\Delta x_n g_x(x_n, y_n) + \Delta y_n g_y(x_n, y_n) = -g(x_n, y_n)$$

In matrix form :

$$\begin{bmatrix} f_x(x_n, y_n) & f_y(x_n, y_n) \\ g_x(x_n, y_n) & g_y(x_n, y_n) \end{bmatrix} \begin{bmatrix} \Delta x_n \\ \Delta y_n \end{bmatrix} = - \begin{bmatrix} f(x_n, y_n) \\ g(x_n, y_n) \end{bmatrix}$$

evaluated at x_n, y_n

start with initial values and solve for $\begin{bmatrix} \Delta x_n \\ \Delta y_n \end{bmatrix}$ and update x_n, y_n to x_{n+1}, y_{n+1} and redo the same process for solution convergence.

For n-unknowns and n-equations the matrices become:

$$\begin{bmatrix} f^1_{x_1}(x^1_n, x^2_n \cdots x^n_n) & f^1_{x_2}(x^1_n, x^2_n \cdots x^n_n) & \cdots & f^1_{x_n}(x^1_n, x^2_n \cdots x^n_n) \\ f^2_{x_1}(x^1_n, x^2_n \cdots x^n_n) & \cdots & \cdots & f^2_{x_n}(x^1_n, x^2_n \cdots x^n_n) \\ \vdots & & \ddots & \\ f^n_{x_1}(x^1_n, x^2_n \cdots x^n_n) & & \cdots & f^n_{x_n}(x^1_n, x^2_n \cdots x^n_n) \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} f^1 \\ f^2 \\ \vdots \\ f^n \end{bmatrix}$$

evaluated at $x^1, x^2, \ldots x^n$

Case study - Lorentz equations

The lorentz equations are:

$\dot{x} = \sigma(y-x)$, $\dot{y} = x(r-z)-y$, $\dot{z} = xy - \beta z$ → where σ, r, β are fixed parameters

Roots of these solutions mean $\dot{x}=0, \dot{y}=0, \dot{z}=0$

∴ the equations become: $\sigma(y-x)=0$ $f(x,y,z)$

$x(r-z)-y=0$ $g(x,y,z)$

$xy-\beta z=0$ $h(x,y,z)$

Jacobian:

∴ $f_x = -\sigma$, $f_y = \sigma$, $f_z = 0$

$g_x = r-z$, $g_y = -1$, $g_z = -x$

$h_x = y$, $h_y = x$, $h_z = -\beta$

∴ Matrix:

$$\begin{bmatrix} -\sigma & \sigma & 0 \\ r-z & -1 & -x \\ y & x & -\beta \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\sigma(y_n - x_n) \\ -x_n(r-z_n)+y_n \\ -x_n y_n + \beta z_n = 0 \end{bmatrix}$$

∴ Now we solve for ΔMatrix and find x_1, y_1, z_1 and so on.

For lorentz fractals we need to grid up $x-z$ plane, set $y = 3\sqrt{z}$ for every iteration and run newton's method for all the grid points. Each and every grid point must converge to one of the 3 possible roots and are thus color coded depending on which root they converge to. This gives the fractals.