MÉTODO DE NEWTON-SIMPSON

TRABAJO FINAL

Andres Felipe Vargas Londoño
Universidad Nacional de Colombia
Departamento de Física
anvargasl@unal.edu.co

30 de junio de 2020

ABSTRACT

El presente trabajo se enfoca en exponer el origen e importancia de la conocida ecuación recursiva (1), a menudo atribuida erróneamente a Sir Isaac Newton y Joseph Raphson, siendo realmente Thomas Simpson quien innovó al emplear el cálculo de fluxiones para el desarrollo de un método que permitiera aproximar la solución numérica de ecuaciones no solo polinómicas.

1. Historia

El desarrollo de lo que hoy comúnmente se conoce como el método de Newton-Raphson es interesante y usualmente mal acreditado. En principio el método desarrollado por Newton era una mejora del creado por Viète, el ejemplo dado en el libro *De analysi per aequationes numero terminorum infinitas* [1] de Newton es la ecuación cúbica $x^3 - 2x - 5 = 0$. La idea empleada era asumir $r + \epsilon$ como solución de la ecuación, donde r es una primera aproximación de la raíz y ϵ el error que r pudiera tener, al reemplazar se obtiene una nueva ecuación cúbica en términos de ϵ, asumiendo ϵ pequeño sus potencias se consideraban nulas y se llega a una expresión lineal de la forma $c\epsilon = d$ para dos constantes reales c y d; el proceso se repite hasta obtener un resultado satisfactorio. Esto es a grandes rasgos lo que podría llamarse “el método de Newton”, un argumento que no emplea el cálculo diferencial, o cálculo de fluxiones como se le conocía entonces, sin ninguna relación aparente a la ecuación recursiva que normalmente se le asocia.

Posteriormente Joseph Raphson trabajaría sobre los cimientos de Viète y Newton para desarrollar un nuevo algoritmo semejante al de Newton, sin embargo en este caso se despeja ϵ después de reemplazar en la ecuación. Hoy es claro que este proceso lo llevaba siempre a una expresión de la forma $\epsilon = \frac{p(x)}{p'(x)}$, con $p(x)$ un polinomio y $p'(x)$ su derivada, es importante aclarar que Raphson no usó esta ni ninguna otra notación que implicara una noción de derivada a pesar de que el cálculo de fluxiones ya se había dado a conocer, además siempre se limitó a ecuaciones polinómicas. No fue sino hasta 1740 que Thomas Simpson implementó el cálculo de fluxiones en expresiones algebraicas, para así finalmente llegar a un resultado que se asemeja mucho más a la hoy famosa expresión iterativa

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

usando su método no solo en polinomios sino también en raíces cuadradas y funciones exponenciales, es por esto que de ahora en adelante se referirá al método iterativo representado por (1) como Newton-Simpson. En *Thomas simpson and ‘newton’s method of approximation’: an enduring myth* [2] el lector puede encontrar una versión mucho más detallada del desarrollo de los métodos así como sus distintas cualidades.

2. Importancia

El método de Newton-Simpson se ha mostrado muy útil recientemente gracias al desarrollo de computadores capaces de realizar múltiples iteraciones del algoritmo en poco tiempo, Simpson lo llamó en su momento “A new Method for the Solution of Equations”, que es el uso que comúnmente se le da, a simple vista esto no resulta demasiado atractivo, sin embargo el carácter iterativo del método lo hace relativamente sencillo de implementar en términos de programación. Algunos programas gráficos lo emplean para el cálculo de raíces inversas debido a su precisión para buenos valores iniciales de la iteración, esto permite determinar ángulos de incidencia de la luz, reflexiones y sombreado; un ejemplo de este uso es el juego Quake III Arena del año 1999 [3].

Es comúnmente usado en análisis numérico y es allí donde puede verse que el método es bastante sensible a la elección inicial del estimado de la raíz de un polinomio $p(x)$ por ejemplo, asociando un punto del plano complejo a dicha elección es posible ver entonces si este converge o no a alguna de las raíces de p, qué tan rápido lo hace y a cuál de ellas. Mediante programación es posible entonces visualizar dichas características para múltiples puntos del plano, a esto se le llama fractales de Newton, en [4] se explica detalladamente el proceso necesario para graficar uno y las conclusiones a las que se puede llegar a partir de ellos para ecuaciones de una y múltiples variables. Actualmente existen múltiples programas que permiten realizar este tipo de visualizaciones y su implementación es relativamente sencilla.

3. Implementación computacional de los Fractales de Newton

Esta sección esta dedicada a mostrar los resultados arrojados por un programa sencillo desarrollado en el lenguaje de programación C++ inspirado en su mayoría por el ejemplo dado en [5]

Con el propósito de comprender mejor la información que guarda un Fractal de Newton se creó un código capaz de imprimir en formato .ppm los resultados de la convergencia de un punto en el plano imaginario. El desarrollo completo del código junto con los resultados se encuentra en el siguiente repositorio de github.

La idea es tomar un píxel en la imagen y asignarle un punto en el plano imaginario, para ello se crean dos loops que recorren cada fila y columna de píxeles, se toma x (columna) como la parte real, y (fila) como la parte imaginaria, usando las funciones `map_to_real` y `map_to_imaginary` se mapean x y y en los intervalos $(-2{,}5, 1)$ y $(-1, 1)$ respectivamente.

```
for(int y = 0; y < height; y++){
  for(int x = 0; x < width; x++){
    complex z(map_to_real(x), map_to_imaginary(y)); // map the pixel to the imaginary plane
```

Lo siguiente es darle un color al píxel dependiendo del número de iteraciones del método de Newton-Simpson requeridas para que su imagen sea menor a `tolerance = 0.00001`, si no se logra en 100 iteraciones el píxel se pinta de negro.

```
      for(int i = 0; i < maxiter; i++){
  if(std::abs(f(z)) < tolerance){ // Color the pixel if z converges fast enough to a root
    int red = 255 - i*i % 255;
    int green = 255 - i*i % 255;
    int blue = 255 - i*i % 255;
    img << red << " " << green << " " << blue << std::endl;
    break;
    }
  z = z - f(z)/deriv(z); // Newton's Method
      }
      if(std::abs(f(z)) > tolerance){
  img << 0 << " " << 0 << " " << 0 << std::endl;
      }
```

Las operaciones que determinan el color del píxel arrojan escalas de grises, a mayor velocidad de convergencia más claro es el píxel. El código puede mejorarse haciendo que el color que se imprime sea distinto dependiendo de la raíz a la que converge el punto, además de solamente sombrear de acuerdo a la velocidad de convergencia.

Estos son los resultados arrojados para tres funciones de variable compleja a valor complejo distintas

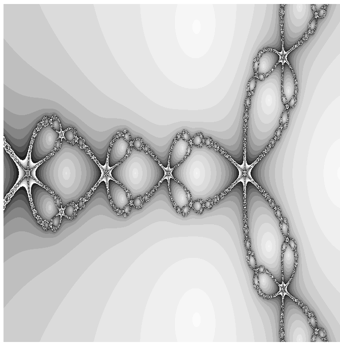

Figura 1: $f(z) = z^3 - 1$

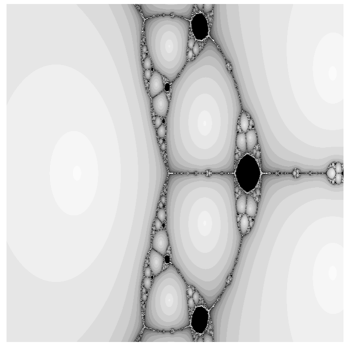

Figura 2: $f(z) = z^3 - 2z + 1$

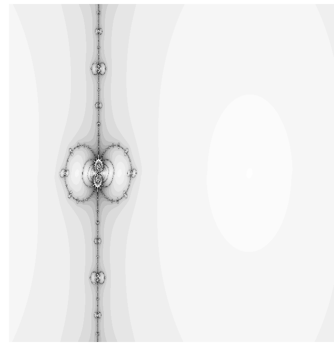

Figura 3: $f(z) = sin(z)$

Por lo mencionado antes es fácil ver que las partes más claras de las imágenes son aquellas regiones del plano que convergen más rápidamente a alguna de las raíces de su respectiva función, por otro lado las zonas oscuras son aquellos puntos cuya imagen no es inferior a la tolerancia dada después de 100 iteraciones. Nótese que en el caso de la figura 1 $f(z)$ solo tiene tres raíces, sin embargo son más las regiones del plano que convergen rápidamente a alguna de ellas.

4. Conclusiones

En la opinión del autor el nombre "Newton-Raphson"no hace justicia al ingenio de Thomas Simpson para emplear el cálculo de fluxiones en expresiones algebraicas y así llegar a un resultado tan satisfactorio del algoritmo. Es cierto que ambos Newton y Raphson realizaron aportes relevantes y necesarios para el progreso del método, sin embargo nunca se dieron a la tarea de llevarlo a otras áreas de las matemáticas, encerrando sus desarrollos en ecuaciones polinómicas, es por esto que este trabajo se refiere al método como Newton-Simpson.

El presente escrito solo expone un par de las aplicaciones de Newton-Simpson, sin embargo es clara su versatilidad, maleabilidad y belleza, siendo relativamente fácil de implementar e interpretar, además de presentar rápida convergencia para condiciones adecuadas del valor inicial, lo que implica que solo un pequeño número de pasos es requerido para lograr un resultado satisfactorio de la aproximación.

Referencias

[1] Isaac Newton. *De analysi per aequationes numero terminorum infinitas*. 1711.

[2] Nick Kollerstrom. Thomas simpson and 'newton's method of approximation': an enduring myth. *The British journal for the history of science*, 25(3):347–354, 1992.

[3] Wikipedia contributors. Fast inverse square root — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Fast_inverse_square_root&oldid=963980486, 2020. [Online; accessed 29-June-2020].

[4] Zoltán Kovács. Understanding convergence and stability of the newton-raphson method. 2011.

[5] Wikipedia contributors. Newton fractal — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Newton_fractal&oldid=951517872, 2020. [Online; accessed 28-June-2020].