TRABAJO PRÁCTICO N°4
Solución de ecuaciones no-lineales

Método de Bisección

1. *Que el código se ejecute sin errores no implica que devuelva un resultado correcto*

   Especialmente cuando se trata de programas que toman datos numéricos, los procesan, y devuelven un resultado numérico, es muy común que la compilación y ejecución se den sin avisos ni errores y, aún así, el resultado sea completamente incorrecto. También puede ser que el resultado sea inesperado, en cuyo caso la revisión de la *calidad* del resultado se vuelve evidente. Pero cuando el resultado es erróneo, y el error no es evidente, las consecuencias suelen ser negativas.

   Es importante que el código sea legible y que, en lo posible, sea “auto documentado”, es decir, que contenga los comentarios necesarios para entender con facilidad su funcionamiento. De esta manera se facilita el trabajo con otros, y permite que uno entienda el código con facilidad en el futuro.

   En este caso se analizan dos curvas, una de tenacidad y otra de tensión de fluencia, ambas en función de la temperatura de revenido de un acero inoxidable. Dado que para la selección de un material y un tratamiento térmico pueden evaluarse varias curvas, contar con la implementación de un método numérico puede simplificar un análisis innecesariamente largo y dar una justificación suficientemente buena de la selección que se realice. Para esta aplicación particular, se requiere que la energía de impacto (Fig 1) sea superior a 14,5 J y que la tensión de fluencia supere los 850 MPa.

   a) Analizar el código siguiente. Comentarlo. Ejecutarlo para encontrar los puntos en que la curva 1 cruza los 14,5 J.

```
function zero =  biseccion(f, a, b, errAbs, maxiter)
    err = 0;
    niter = 0;
    if f(a)*f(b) > 0
        err = 1;
        zero = NaN;
    endif
    if(a == 0 && b == 0)
        err = 2;
        zero = NaN;
    endif
    if err == 0
        c = (a + b)/2;
        do
            if f(a)*f(c) < 0
```

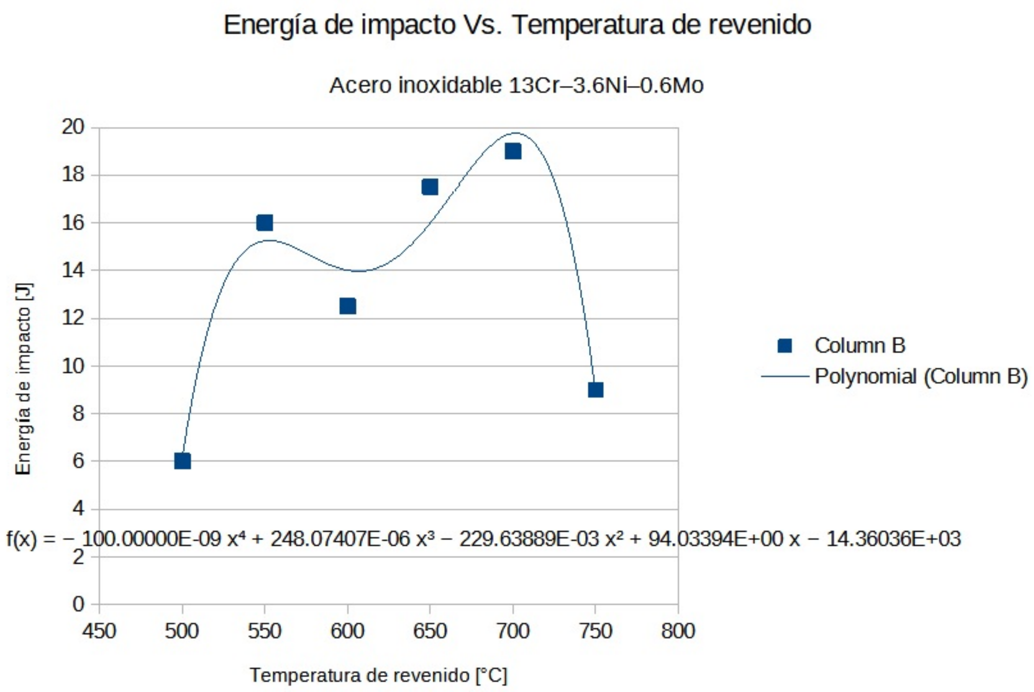


Figura 1: Energía de impacto vs. Temperatura de revenido del acero inoxidable 13Cr-3.6Ni-0.6Mo

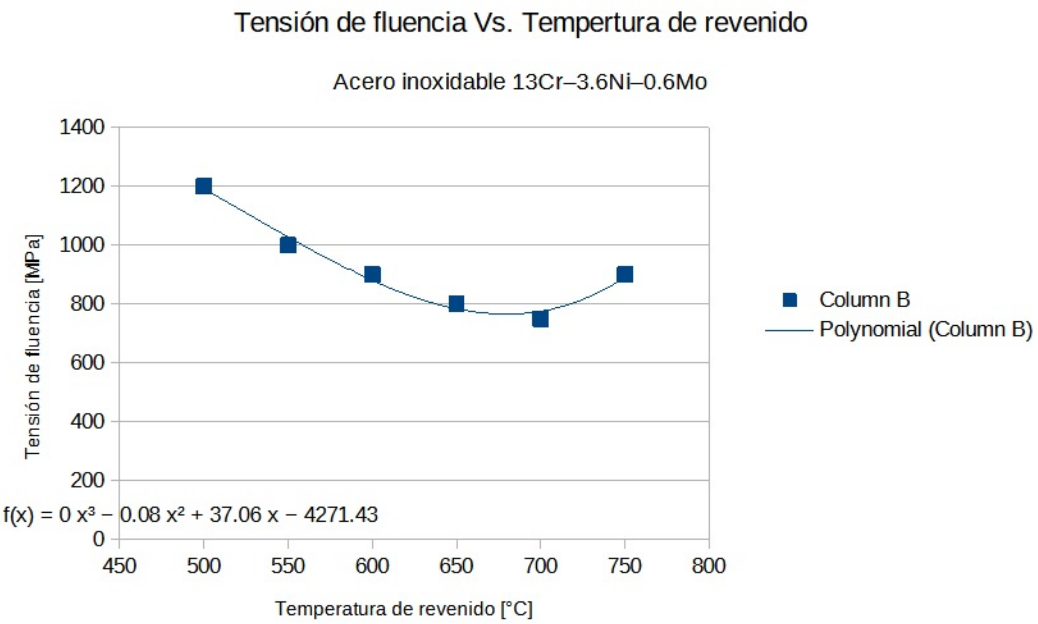


Figura 2: Tensión de fluencia vs. Temperatura de revenido del acero inoxidable 13Cr-3.6Ni-0.6Mo

```
            b = c;
        elseif f(b)*f(c) < 0
            a = c;
        else
            a = c;
```

```
                b = c;
            endif
            c = (a + b)/2;
            zero = c;
            niter = niter + 1;
        until f(c) <= errAbs || niter > maxiter
    endif
niter
endfunction
```

b) El código del punto anterior dejó de usarse y cayó en el olvido. ¿Por qué (acaso tendrá algún error)? En caso de encontrar errores, corregirlos.

c) ¿Qué agregaría al código anterior para depurarlo con facilidad?

Método de Punto fijo

2. *Un amigo de un amigo del tío de mi primo inventó la pólvora*

   "Inventar la pólvora" (o "inventar la rueda") no siempre es algo malo; tener la tendencia a inventar la pólvora para luego darse cuenta de que alguien ya la inventó primero puede indicar que uno sigue caminos similares a los que otros ya probaron y dieron por válidos. Eso parece indicar que lo que otros hicieron servirá para resolver nuestro problema, y esto suele ser cierto con una salvedad no necesariamente menor: casi todo código ajeno necesita ser comprendido y adaptado antes de ser útil a los fines que nos interesan.

   a) El código siguiente obtiene ceros de funciones por método iterativo de punto fijo. Fue obtenido de Stack Overflow. Sin embargo, carece de los obligatorios chequeos de consistencia de método, que deberán agregarse.

```
function y = punto_fijo(f, x1, errAbs, maxIter)
    x2 = f(x1);
    iter = 0;
    while(abs(x2-x1) > errAbs && iter < maxIter)
        iter = iter + 1;
        x1 = x2;
        x2 = f(x1);
    endwhile
    y = x1;
endfunction
```

      i) ¿Cuáles son los límites del método?

      ii) Agregar al código los controles necesarios para que siempre que sea posible se obtenga una solución y que, cuando no se pueda, se lo informe como salida (puede ser salida tipo NaN, un valor numérico particular o un mensaje de error; la elección del método deberá justificarse).

   b) Modificar el código para que informe la cantidad de iteraciones necesarias para alcanzar la convergencia.

      i) Ejecutar este código, y el del ejercicio anterior, para encontrar los puntos en los que la curva 1 cruza la línea de 14,5 J.

      ii) Comparar la cantidad de iteraciones necesarias en uno y otro caso.

iii) **Bonus**. Repetir el análisis con la curva 2 para una tensión de fluencia de $850 MPa$. ¿Cuál es el rango en el que se obtiene la mejor combinación de propiedades mecánicas?

Método de Newton-Raphson

3. *Uno aprender a programar, no aprende un lenguaje de programación.*

   El código siguiente fue escrito por un profesor de la cátedra.

```
function zero = newton(f, f1, x0, errAbs, maxiter)
    #Calculo de ceros de funciones por metodo de Newton-Raphson
    x0;
    fx0 = f(x0);
    f1x0 = f1(x0);
    x = x0 - fx0/f1x0;
    fx = f(x);
    zero = x;
    niter = 0;
    while abs(fx) > errAbs && niter < maxiter
        x0 = x;
        fx0 = f(x0);
        f1x0 = f1(x0);
        x = x0 - fx0/f1x0;
        fx = f(x);
        zero = x;
        niter = niter + 1;
    endwhile
    niter
endfunction
```

   El código siguiente fue hallado en internet (Stack Overflow), pero no está escrito en Octave/Matlab, sino en C.

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>

double F(double x){
    return asin(x) - M_PI/6;
}

#define Tk (double)0.0000001
double Fx(double x, double (*F)(double), double k){
    return (double)((F(x+k) - F(x))/k);
}
main(){
    double x0 = 100, x, eps = 0.001, ep;
    while(1){
        x = x0 - F(x0)/Fx(x0, F, Tk);
```

```
            ep = fabs((x-x0)/x0)*100;
            x0 = x;
            if(fabs(ep) < eps) break;
        }
        printf("%lf", x0);
        getch();
    }
```

Salvo porque el alcance de los bloques (loops, bloques "if", definición de funciones, etc) están encerrados entre llaves "{}", el lenguaje se parece bastante a Octave/Matlab. Es muy común que se necesiten soluciones en un lenguaje pero se tengan disponibles en otro. O bien, que se conozca un lenguaje pero haya que programar la solución en uno diferente. De ahí la importancia de aprender a reconocer las partes de un programa y ser capaz de traducirlo a otro lenguaje.

a) Analizar el primer código del ejercicio, comentarlo, agregarle los controles necesarios.

b) Traducir el segundo código de C a Octave/Matlab. Comentarlo y agregarle los controles necesarios. Asegurarse de que el código resultante devuelva, al menos como salida en pantalla, la cantidad de iteraciones realizadas hasta alcanzar la convergencia.

c) Ejecutar ambos códigos para encontrar los puntos en que la curva 1 cruza los 14,5 J. Comparar los resultados y el número de iteraciones realizadas. Comparar este número con el de los otros dos métodos.

4. *¿Y para qué sirve todo esto? La idea de este ejercicio es juntar todo lo anterior, y desarrollar las herramientas necesarias para resolver prácticamente cualquier problema que requiera resolver una ecuación no lineal.*

   a) En el ejercicio 2 se desarrolló una función que permite hallar ceros de funciones con el método del punto fijo, pero, ¿Qué pasa si no se cuenta con una función, sino con un conjunto de puntos? Modifiquemos un poco el programa resultante del ejercicio 2 para que:

      i) Tome dos conjuntos de puntos, uno de x y otro de y.

      ii) Verifique que ambos conjuntos (que deberán ser dados a la función como vectores) tengan la misma longitud. ¿Por qué es necesario este chequeo?

      iii) Busque cuáles son los puntos entre los que se pasa por cero. Tal vez el mismo algoritmo de bisección se pueda usar para esto...

      iv) Utilice dos puntos más, uno a izquierda y otro a derecha de los seleccionados en iii) para construir dos parábolas (puede usarse la función **polyfit** para esto). Contemplar si esto se puede hacer en los extremos del conjunto de puntos.

      v) Encuentre los ceros de estas dos parábolas. Para esto, deberá construirse una versión modificada de la función programada en el ejercicio 1 para que, en lugar de resolver para una función en forma de handle, resuelva para un polinomio. Para eso, la función tiene que tomar como argumentos un vector (con los coeficientes del polinomio), el intervalo de búsqueda de la raíz, la tolerancia y el mayor número de iteraciones permitido.

      vi) Devuelva el promedio de estos dos valores como "cero" del conjunto de puntos.

      vii) Si no se puede encontrar un cero, se devuelva NaN como resultado y se muestre un mensaje de error en pantalla.

   b) Los items del inciso anterior son una descripción bastante precisa de lo que debe hacer la función. Es una especie de "diagrama de flujo, pero escrito". Ahora, construya funciones similares usando los otros métodos, repitiendo los ítems anteriores. Se permiten modificaciones a la lista para adaptar este desarrollo a los otros métodos. Si hay modificaciones, cambiar también la lista de ítems. ¡Cuidado al adaptar la función que resuelve por punto fijo!

c) Modificar las funciones programadas en los incisos a) y b) para que halle la posición en que "el conjunto de puntos pasa por un $y = \text{cte}$".

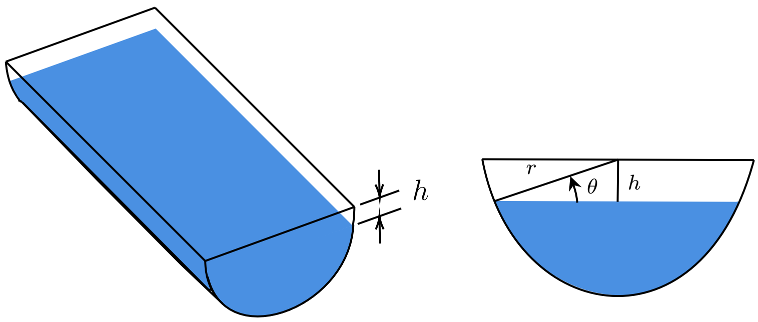
d) Repetir el inciso anterior pero para las funciones programadas en los ejercicios 1, 2 y 3. El resultado tendría que ser tres funciones, una para *bisección*, otro para *punto fijo*, y otro para *Newton-Raphson*, que pueda resolver el problema $f(x) = \text{cte}$.

Problemas

1. Encuentre una aproximación de $\sqrt{3}$ con error menor a 10^{-4} utilizando el método de bisección. Pista: considere $f(x) = x^2 - 3$

2. Un canal de largo L posee una sección transversal en forma de semicírculo de radio r. Cuando el canal está lleno hasta una distancia h del tope, el volumen de agua V es:

$$V = L\left(0{,}5\pi r^2 - r^2 \arcsin\left(\frac{h}{r}\right) - h(r^2 - h^2)^{\frac{1}{2}}\right)$$



Suponga $L = 3$ m$, r = 0{,}3$ m y $V = 0{,}351$ m^3. Encuentre la profundidad de agua en el canal con un error menor a $3 \cdot 10^{-3}$.

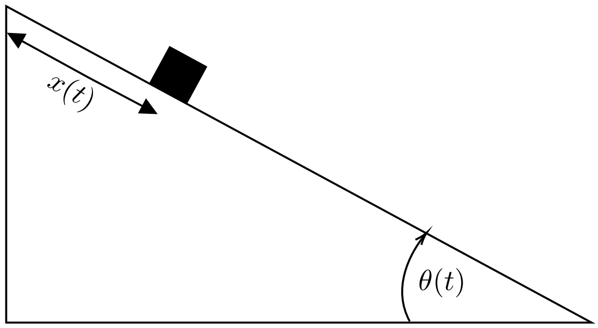
3. Un objeto puntual comienza a desplazarse sobre un plano inclinado cuyo ángulo θ cambia a velocidad constante:

$$\frac{d\theta}{dt} = \omega < 0.$$

Luego de t segundos, la posición del objeto está dada por:

$$x(t) = -\frac{g}{-2\omega^2}\left(\frac{e^{\omega t} - e^{-\omega t}}{2} - \sin(\omega t)\right)$$

Suponga que el objeto se desplazó 0,52 m en 1 segundo. Encuentre, con un error menor a 10^{-5}, la velocidad ω con la que varía θ. Asuma $g = 9{,}8\frac{m}{seg^2}$

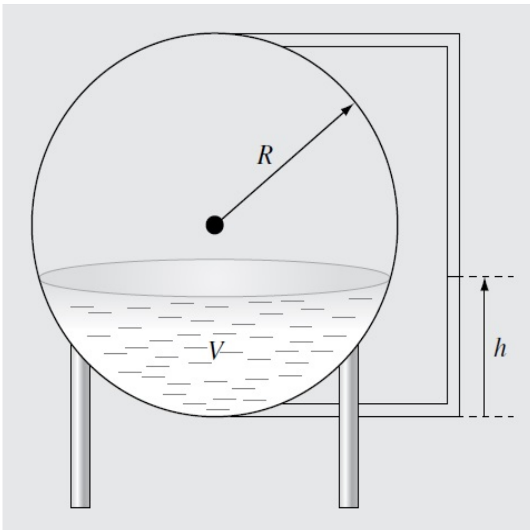


4. Se debe diseñar un tanque esférico (vease la Figura adjunta) para almacenar agua para un poblado pequeño. El volumen de líquido que puede contener se calcula con:

$$V = \pi h^2 \frac{3R - h}{3},$$

donde V es el volumen medido en metros cúbicos, h es la profundidad del agua en el tanque medida en metros, y R es el radio del tanque medido en metros.

Si $R = 3$ m. ¿A qué profundidad debe llenarse el tanque de modo que contenga 30m^3?



5. Manipule las siguientes expresiones para mostrar que cada una de las siguientes expresiones posee un punto fijo en p cuando $f(p) = 0$, donde $f(x) = x^4 + 2x^2 - x - 3$.

   a) $g_1(x) = (3 + x - 2x^2)^{1/4}$

   b) $g_2(x) = \left(\dfrac{x + 3 - x^4}{2}\right)^{1/2}$

   c) $g_3(x) = \left(\dfrac{x + 2}{x^2 + 2}\right)^{1/2}$

   d) $g_4(x) = \dfrac{3x^4 + 2x^2 + 3}{4x^3 + 4x - 1}$

6. Un objeto cayendo de forma vertical en aire está sometido a resistencia viscosa y a la fuerza de gravedad. Asumiendo que el objeto posee masa m cae desde una altura s_0 y que la altura respecto del suelo luego de t segundos es:

$$s(t) = s_0 - \frac{mg}{k}t + \frac{m^2 g}{k^2}(1 - e^{-kt/m})$$

donde $g = 9{,}8\mathrm{m/s}^2$, y k es el coeficiente de la resistencia del aire en $\frac{\mathrm{kg} \cdot \mathrm{seg}}{\mathrm{m}}$. Suponga que $s_0 = 91{,}44$ m, $m = 0{,}11$ kg, y $k = 0{,}15\,\frac{\mathrm{kg} \cdot \mathrm{seg}}{\mathrm{m}}$. Encuentre el tiempo que tarda el objeto en caer al suelo con un error menor a 0,01 seg, utilizando el método de punto fijo.

7. El balance de masa de un contaminante completamente diluido en un lago se expresa como:

$$V\frac{dc}{dt} = W - Qc - kV\sqrt{c}$$

Dados los valores de parámetros $V = 10^6$ m^3, $Q = 10^5\,\frac{\mathrm{m}^3}{\mathrm{año}}$, $W = 10^6\,\frac{\mathrm{g}}{\mathrm{año}}$ y $k = 0{,}25\frac{\mathrm{m}^{0{,}5}}{\mathrm{g}^{0{,}5}\mathrm{año}}$, use el método del punto fijo para resolver la concentración de estado estable/estado estacionario.

8. Utilice el método de Newton-Raphson para aproximar, con un error menor que 10^{-4}, el valor de abscisa x que define el punto de menor distancia entre la parábola $y = x^2$ y el punto $(1, 0)$. Pista: Minimice $(d(x))^2$, donde $d(x)$ es la función distancia entre el par de puntos definido por la parábola y el punto dado.

9. Una droga administrada a un paciente produce una concentración en el torrente sanguíneo dado por $c(t) = Ate^{-t/3}$ miligramos por mililitro, t horas después de haber inyectado A unidades. La concentración segura máxima es de $1\,\frac{\mathrm{mg}}{\mathrm{ml}}$

   a) ¿Qué cantidad debería ser inyectada para alcanzar la concentración segura máxima? Este máximo ¿Cuándo ocurre?

   b) Se administra una nueva dosis de la droga en el paciente luego de que la concentración cayera a $0{,}25\frac{\mathrm{mg}}{\mathrm{ml}}$. Determine, redondeado al minuto más cercano, cuándo debería realizarse la segunda inyección. Asuma que la concentración de inyecciones consecutivas es aditiva.

   c) Considere que el 75 % de la dosis inyectada inicialmente es administrada en la segunda inyección. ¿Cuándo debería administrarse la tercera inyección?

Referencias

[1] Burden, R. L. *Numerical Analysis*, 2019

[2] RP, C. S. C. *Numerical methods for engineers*, 2002