Métodos Numéricos

http://www.famaf.unc.edu.ar/∼serra/metodos_numericos.html

Guía 2

Abril de 2016

Problema 1: Cree un directorio dentro de su directorio metnum, con nombre guia2. Luego copie el archivo *guia2-2016.pdf* que se encuentra dentro del /tmp, al directorio creado.

Comentario: todo archivo que se cree dentro del /tmp (temporal) dura un tiempo finito. En general se setea el sistema para que borre todo lo que está dentro del temporal, cada vez que se rebootea (reinicia) la máquina. Conclusión, cópielo ahora porque la próxima clase no estará ahi.

Problema 2: Interprete el resultado del siguiente programa en virtud de la representación de punto flotante de los números reales.

```
program test_igualdad
implicit none
integer, parameter      :: pr=kind(1.d0)      ! pr puede ser simple o doble

  if (19.08_pr + 2.01_pr == 21.09_pr) then
    write(*,*) '19.08 + 2.01 = 21.09 '
   else
     write(*,*) '19.08 + 2.01 /= 21.09 '
  endif

end program test_igualdad
```

Piense el mensaje de este ejercicio, el cual debe tener presente en toda la materia.

Problema 3: Implementar un programa Fortran para evaluar la suma (en precisión simple)

$$\sum_{n=1}^{10\,000\,000} \frac{1}{n}$$

primero, en el orden usual, y luego, en el orden opuesto. Explique las diferencias obtenidas e indique cuál es más preciso y su justificación.

Problema 4: Efectúe con un programa en Fortran en simple precisión los siguientes cálculos, matemáticamente equivalentes,

a) $1\,000\,000 \times 0.1$

b) $\sum_{n=1}^{1\,000\,000} 0.1$

c) $\sum_{m=1}^{1\,000} \left(\sum_{n=1}^{1\,000} 0.1 \right)$

En los puntos b) y c), vaya guardando con un write, las sumas parciales en un archivo de datos con primera columna los valores de los índices de las sumas y segunda columna las sumas parciales. Idem para el punto c), pero en archivo con otro nombre. Grafique usando *gnuplot*, las sumas parciales de b) y c) superpuestas, en función del número de términos, en un archivo“prob4.ps” en color, con título de gráfica“Problema 4, Guía 2”, con nombres adecuados en los ejes y con leyendas adecuadas para cada set de datos.

La columna 2 de b), graficarla con línea de grosor 0.75 (el default es 1), y la columa 2 de c) con puntos de tamaño 1.2 (el default es también 1). Luego de ver cómo le queda la gráfica en escala lineal, cambie el eje x, a escala logarítmica. Según su criterio, en qué escala se aprecia mejor el resultado del problema para su análisis ? Puede al eje y darle escala logarítmica? Pruebe qué pasa si lo hace.

Finalmente deje el archivo grabado en escala lineal en y, y logarítmica en x, y diga el tamaño del mismo tanto en bytes como kbytes. Aprecie las dimensiones y sea conciente de cuanto espacio de disco podemos llegar a usar rápidamente (recuerde en su cuenta puede tener solo 200MB, utilice el df -h para ver el uso de su home), asique luego de usarlo lo borra, o escribe algunos términos nada mas, por ej. cada 100 ó 500 y vea como se reduce. *Es buen hábito de programador, estimar el tamaño de los archivos de salida, antes de correr.*

Si le gusta usar otro graficador, se incentiva a que lo usen, siempre y cuando logren el archivo final con mismos requisitos que le pedimos.

Explique las diferencias obtenidas entre resultados finales de a), b) y c) y muestre que el error relativo en b) es del orden del 1%, pero es mucho menor en c). Resalte la conclusión de este ejercicio.

Problema 5: La fórmula cuadrática nos dice que las raíces de $ax^2 + bx + c = 0$ son

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \qquad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

Si $b^2 \gg 4ac$, entonces, cuando $b > 0$ el cálculo de x_1 involucra en el numerador la sustracción de dos números casi iguales, mientras que si $b < 0$, esta situación ocurre para el cálculo de x_2. “Racionalizando el numerador” se obtienen las siguientes fórmulas alternativas que no sufren este problema:

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}, \qquad x_2 = \frac{2c}{-b + \sqrt{b^2 - 4ac}},$$

siendo la primera adecuada cuando $b > 0$, y la segunda cuando $b < 0$. Escriba un programa en precisión simple que utilice la fórmula usual y la “racionalizada” para calcular las raíces de

$$x^2 + 6210x + 1 = 0.$$

Interprete los resultados.

Problema 6: Considere las siguientes integrales

$$y_n = \int_0^1 \frac{x^n}{x + 10} dx$$

para $n = 1, 2, \ldots, 30$. Muestre que

$$y_n = \frac{1}{n} - 10 y_{n-1} \ ,$$

y que $y_0 = \ln(11) - \ln(10)$. Note que empleando esta fórmula de recursión, se obtienen los resultados exactos de las integrales.

a) Escriba un programa en precisión simple que a partir de y_0, calcule recursivamente y_i para $i = 2, \cdots, 30$. Explique los resultados obtenidos (note que $0 < y_n < 1$).

b) Derive una fórmula para evaluar y_{n-1} dado y_n. Escriba un programa que utilice esta recursión para calcular y_n, aproximando y_{n+k} por 0. Explique por qué este algoritmo es estable. Encuentre el valor de k, para que el programa calcule y_7 con un error absoluto menor a 10^{-6} (note que $y_7 \approx 0.0114806$).

c) Modifique el programa para que tome como entrada n, y el error absoluto deseado, ϵ, y luego estimando el error absoluto en el calculo de y_n como $Err = |\hat{y}_n(y_{n+k} = 0) - \hat{y}_n(y_{n+k-1} = 0)|$ ($\hat{y}_n(y_{n+k} = 0)$ es el valor de y_n obtenido partiendo de $y_{n+k} = 0$)), determine y_n con un error absoluto (aproximado) menor que ϵ.

Problema 7: *Problema matemáticamente inestable.* Considere la suceción

$$x_n = \frac{13}{3}\, x_{n-1} - \frac{4}{3}\, x_{n-2}\,. \tag{1}$$

a) Demuestre que, eligiendo $x_0 = 1$, $x_1 = 1/3$ tenemos que $x_n = 1/3^n \ \forall n \geq 0$ (sugerencia: use inducción).

b) Haga un código fortran que calcule x_n y su error relativo hasta $n = 15$ y discuta el resultado comparando reales de 4 y 8 bytes.

c) Defina $y_n = 1/x_n$ y encuentre la relación de recurrencia para y_n. Imponga la condición inicial $y_0 = 1,\ y_1 = 3$. Calcule ahora $x_n = 1/y_n$ y compare con lo obtenido en el punto anterior. Es este algoritmo estable? discuta.

d) Demuestre que la solución general de la ecuación (1) con x_0, x_1 arbitrarios es

$$x_n = \frac{A}{3^n} + B\,4^n\,,$$

y discuta en base a esto los resultados numéricos obtenidos.

Problema 8: Desarrolle un programa para encontrar la raíz de una función f utilizando el método de la bisección, dando como datos de entrada el intervalo inicial $[a, b]$ y la tolerancia ε. f debe definirse como una función dentro del programa. La salida debe ser

- archivo con tres columnas: N, x_N y $f(x_N)$ y $|x_N - x_{N-1}|/|x_N|$, con 12 cifras significativas para los reales.
- la aproximación final x_N (en pantalla)
- el valor final de $f(x_N)$. (en pantalla)
- el número de iteraciones realizadas (en pantalla)

Utilice el programa para
a) encontrar la menor solución positiva de la ecuación $2x = \tan(x)$ con un error menor a 10^{-5}. Cuántos pasos son necesarios si se comienza con el intervalo $[0.8, 1.4]$?
b) encontrar una aproximación a $\sqrt{3}$ con un error menor a 10^{-5}. Note que $\sqrt{3}$ es la raíz positiva de la ecuación $f(x) = x^2 - 3$.

Problema 9: Desarrolle un programa para encontrar la raíz de una función f utilizando el método de Newton–Raphson, dando como datos de entrada una estimación inicial x_0, la tolerancia `tol` y un número máximo de iteraciones `MAX_ITE`. El programa debe finalizar cuando se satisfaga una de las siguientes condiciones:

$$\frac{|x_N - x_{N-1}|}{|x_N|} < \varepsilon, \qquad |f(x_N)| < \varepsilon, \qquad \text{Número de iteraciones} = \texttt{MAX\_ITE}$$

El programa debe retornar (en pantalla) el número de iteraciones realizadas, el valor final de la aproximación x_N, el error relativo estimado, y el valor de $|f(x_N)|$. f y f' deben ser funciones del programa. Además, en un archivo, debe escribir en cada iteración: N, x_N, $f(x_N)$ y $|x_N - x_{N-1}|/|x_N|$, con 13 cifras significativas para los reales.

Utilice este programa para resolver los incisos a) y b) del problema 8. Compare la cantidad de evaluaciones de la función y su derivada en los dos métodos.

Problema 10: Grafique el error relativo y el error relativo estimado ($|x_k - x_{k-1}|/|x_k|$) de la aproximación k-ésima de $\sqrt{3}$ (pensada como raíz de la función $f(x) = x^2 - 3$) en función de k, empleando el método de bisección (use el intervalo inicial $[0., 2.5]$ y el de Newton–Raphson (use $x_0 = 2.5$). Escriba los programas en doble precisión, y fije una tolerancia de 10^{-10} como criterio de detención. Compare los resultados en un único gráfico en escala *log – log*. Cree un archivo *postscript color* con el gráfico. Recuerde de poner título, nombre a los ejes y leyendas para las distintas curvas.

Problema 11: Adapte el programa de Newton–Raphson para calcular una aproximación a la raíz cúbica de un número R positivo. La entrada debe ser el número R, la aproximación inicial x_0 y el error máximo permitido ε.

Problema 12: Dado el siguiente polinomio

$$p(x) = -10 + 5x - 12x^2 + 6x^3 - 2x^4 + x^5 \ .$$

Grafique el mismo utilizando *gnuplot* y observe que posee una única raíz real positiva, encuentre la misma utilizando:

a) El método de bisección. Elija los valores iniciales utilizando los teoremas que acotan la región del espacio complejo donde se encuentran las raíces. Evalúe el polinomio en una subrutina y utilice el algoritmo de Horner.

b) El método de Newton-Raphson. Elija el valor inicial utilizando los teoremas que acotan la región del espacio complejo donde se encuentran las raíces. Evalúe el polinomio y su derivada en una subrutina utilizando el algoritmo de Horner.

Problema 13: Un objeto en caída vertical en el aire está sujeto a la fuerza de gravedad y a la resistencia del aire. Si un objeto de masa m es dejado caer desde una altura h_0, su altura luego de t segundos está dada por:

$$h(t) = h_0 - \frac{mg}{k}t + \frac{m^2 g}{k^2}\left(1 - e^{-kt/m}\right)$$

donde $g = 9.8\,m/s^2$ y k representa el coeficiente de resistencia del aire en $kg \cdot s/m$. Suponga que $h_0 = 10m$, $m = 0.1\,kg$, y $k = 0.149\,kg\,s/m$. Grafique $h(t)$ usando *gnuplot* para analizar su comportamiento. Encuentre, con una precisión de $0.01\,s$, el tiempo que le toma a este objeto llegar al suelo. Utilice el método de bisección y el de Newton–Raphson.

Ejercicios Complementarios

Problema 14: Aplicar algoritmo de Newton-Raphson generalizado al siguiente sistema de ecuaciones no lineales:

$$\begin{aligned} 2x + \cos(y) &= 0 \\ 2y + \sin(x) &= 0 \end{aligned}$$

Grafique primero las funciones, y encuentre la solución gráficamente. Implemente un programa para resolver este sistema que tome como valor incial el par $(0, 0)$, y escriba los valores sucesivos de los pares (x_k, y_k) obtenidos en la iteración k y la estimación del error absoluto, definido commo $||\mathbf{e}_k||_\infty = ||\mathbf{x}_k - \mathbf{x}_{k-1}||_\infty \equiv \texttt{max}(|x_k - x_{k-1}|, |y_k - y_{k-1}|)$. El programa debe detenerse cuando $||\mathbf{e}_k||_\infty < 10^{-6}$.