FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

Análisis Numérico I
[75.12/95.04]
Curso 3
Trabajo Práctico 1
Aproximación de funciones

Grupo 5
Primer cuatrimestre de 2019

Integrantes del grupo		
Santa María Tomás	92797	tomasisantamaria@gmail.com
Hemmingsen Lucas	76187	lhemmingsen@fi.uba.ar
Huenul Matías	102135	matias.huenul.07@gmail.com

1. Introducción

El objetivo del presente trabajo práctico es obtener aproximaciones de dos funciones y utilizarlas para calcular su valor en distintos puntos, estimando los errores cometidos en ambos casos y analizando las causas de los mismos en base a los conceptos teóricos vistos en el curso.

2. Conceptos teóricos

Teorema de Taylor Este teorema permite obtener aproximaciones polinomiales de funciones diferenciables en un cierto entorno, así también como una cota para el error de aproximación. Sea $k \geq 1$ un número entero y $f(x)$ una función k veces diferenciable en x_0, su polinomio de Taylor de orden k en torno a x_0 se define como:

$$P_k(x) = \sum_{i=0}^{k} \frac{f^{(i)}(x_0)}{i!}(x - x_0)^i \tag{1}$$

y el error que se comete aproximando $f(x)$ a través de $P_k(x)$ es:

$$R_k(x) = \frac{f^{(k+1)}(c)}{(k+1)!}(x - x_0)^{k+1} \tag{2}$$

donde $c \in [x_0, x]$ o $c \in [x, x_0]$.

3. Parte 1

El segundo polinomio de Taylor de la función $f(x) = e^x cos(x)$ alrededor de $x_0 = \frac{\pi}{6}$ es:

$$P_2(x) = \frac{1}{2}\sqrt{3}e^{\frac{\pi}{6}} + \frac{1}{2}(\sqrt{3} - 1)e^{\frac{\pi}{6}}(x - \frac{\pi}{6}) - \frac{1}{2}e^{\frac{\pi}{6}}(x - \frac{\pi}{6})^2 \tag{3}$$

Utilizando este polinomio para aproximar $f(x)$ en $x = 0{,}5$ se obtiene:

$$f(0,5) \approx P_2(0,5) = 1,4469 \tag{4}$$

El error estará dado por:

$$R_2(x) = \frac{-2e^c(sen(c) + cos(c))}{6}(x - \frac{\pi}{6})^3 \tag{5}$$

Luego el error cometido al realizar la aproximación es:

$$R_2(0,5) = -4,3807 * 10^{-6} * e^c(sen(c) + cos(c))$$

Sabiendo que $c \in [0,5; \frac{\pi}{6}]$ podemos tomar $c = \frac{\pi}{6}$ para obtener una cota superior de este error:

$$|R_2(0,5)| \leq 4,3807 * 10^{-6} * e^c(|sen(c)| + |cos(c)|)$$

$$|R_2(0,5)| \leq 4,3807 * 10^{-6} * e^{\frac{\pi}{6}}(|sen(\frac{\pi}{6})| + |cos(\frac{\pi}{6})|)$$

$$|R_2(0,5)| \leq 1,0102 * 10^{-5} \tag{6}$$

Por otro lado, la cota superior del error de aproximación de $f(x)$ al usar $P_2(x)$ en el intervalo $[0, 1]$, tomando $c = 1$, es:

$$|R_2| \leq |\frac{-2e^1(sen(1) + cos(1))}{6}(1 - \frac{\pi}{6})^3|$$

$$|R_2| \leq 0,19594 \tag{7}$$

Al evaluar en Octave se obtiene $f(0,5) = 1,4469$. Este resultado coincide con el calculado mediante la aproximación, lo cual es esperable ya que de acuerdo a (6), $P_2(x)$ logra aproximar a $f(x)$ hasta cuatro decimales sin error y por lo tanto, al redondear simétricamente como lo hace Octave, se llega al mismo valor.

A continuación se encuentra el código en *Octave* de la implementación particular del segundo polinomio de Taylor para la función representada en el problema:

```
function result = taylor_ej1(x)
  x0 = pi / 6
  f = e**x0 * cos(x0)
  # primer derivada
  f1 = e**x0 * (cos(x0) - sin(x0))
  # segunda derivada
  f2 = (-2 * e**x0 * sin(x0))
  f3 = (-2 * e**x0 * (sin(x) + cos(x)))

  p0 = f
  p1 = f1 * (x - x0)
  p2 = (f2 / factorial(2)) * (x - x0)**2
  error = (f3 / factorial(3)) * (x - x0)**3

  result = p0 + p1 + p2
endfunction
```

Este código imprime los siguientes resultados cuando se corre desde Octave:

```
 >> taylor_ej1 (0.5)
  x0 =  0.52360
  f =  1.4619
  f1 =  0.61788
  f2 = -1.6881
  f3 = -4.5815
  p0 =  1.4619
  p1 = -0.014581
  p2 = -0.00047005
  error =  0.000010035
  ans =  1.4469
```

4. Parte 2

(a) El primer método que utilizaremos para aproximar la función error[1] será desarrollar la serie de Taylor de e^{-x^2} alrededor de $x = 0$ y luego aplicaremos la integral definida a cada término de la serie y lo multiplicaremos por $2/\sqrt{\pi}$.
Sabiendo que el desarrollo de Taylor de e^x alrededor del cero es:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ... = \sum_{k=0}^{\infty} \frac{x^k}{k!} \tag{8}$$

Sustituyendo para e^{-x^2}:

$$e^{-x^2} = 1 - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + ... = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{k!} \tag{9}$$

Integrando cada término:

$$\int_0^x e^{-t^2} dt = x - \frac{x^3}{3} + \frac{x^5}{5 \cdot 2!} - \frac{x^7}{7 \cdot 3!} + ... = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1) \cdot k!} \tag{10}$$

Y finalmente, multiplicando por $2/\sqrt{\pi}$ obtenemos:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = \frac{2}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1) \cdot k!} \tag{11}$$

Como calcularemos un número limitado de términos de esta serie, la dejamos expresada para una cantidad de n pasos:

$$P_n(x) = \frac{2}{\sqrt{\pi}} \sum_{k=0}^{n} \frac{(-1)^k x^{2k+1}}{(2k+1) \cdot k!} \tag{12}$$

y el error de truncamiento de la serie estará dado por:

$$R_n(x) = \frac{2}{\sqrt{\pi}} \frac{f^{(n+1)}(c)}{(n+1)!} x^{n+1} = \frac{2}{\sqrt{\pi}} \frac{(-2)^{n+1} e^{-2c}}{(n+1)!} x^{n+1} \tag{13}$$

donde $c \in [0, x]$ o $c \in [x, 0]$. Para $x \geq 0$, $e^{-2c} \leq 1$, entonces:

$$R_n(x) = \frac{2}{\sqrt{\pi}} \frac{(-2)^{n+1}}{(n+1)!} x^{n+1} \tag{14}$$

Como segundo método, aproximaremos la integral de la función error con el método del trapecio[2]. En particular, utilizaremos el método del trapecio compuesto[3], con el que podremos ir variando la precisión del resultado según la cantidad de subintervalos utilizados para calcular la integral.

La regla del trapecio compuesta para n subintervalos para una función $f(x)$ en un intervalo $[a, b]$ está dada por:

$$\int_a^b f(x)dx = \frac{h}{2} \left[f(a) + 2 \sum_{j=1}^{n-1} f(x_j) + f(b) \right] - \frac{b-a}{12} h^2 f''(\mu) \tag{15}$$

donde $h = (b-a)/n$ es el tamaño de cada subintervalo y $x_j = a + jh$ para cada $j = 1, 2, .., n$. El último término corresponde al error del método donde $\mu \in (a, b)$.

Para nuestra función error, siendo $f(x) = e^{-x^2}$ tenemos:

$$\int_0^b e^{-x^2} dx = \frac{h}{2} \left[1 + 2 \sum_{j=1}^{n-1} e^{-x_j{}^2} + e^{-b^2} \right] - \frac{b}{12} h^2 f''(\mu) \tag{16}$$

Finalmente, multiplicando por $2/\sqrt{\pi}$ obtenemos nuestra segunda aproximación de la función error:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \simeq \frac{h}{\sqrt{\pi}} \left[1 + 2 \sum_{j=1}^{n-1} e^{-x_j{}^2} + e^{-x^2} \right] \tag{17}$$

donde $h = x/n$ y $x_j = jh$.

(b) A continuación se encuentra el código en *Octave* de la aproximación de la función error mediante polinomios de Taylor:

```
function result = erf_taylor( x, n )
  x2 = x * x;
  ak = x;
  S = x;
  for k = 1 : n
    ak *= -x2/k;
    S += ak / (2*k + 1);
  end
    result = 2*S / sqrt(pi);
endfunction
```

La función para la aproximación utilizando el método de los trapecios implementada en *Octave* es:

```
function result = erf_trapecios(x, n)
  h = x/n;
  xj = h;
  S = 0;
  for j = 1 : n - 1
    xj2 = xj * xj;
    S += exp(-xj2);
    xj += h;
  end
  S = 1 + 2*S + exp(-1);
  result = h*S / sqrt(pi);
endfunction
```

Se pueden encontrar ambos archivos, *erf_taylor.m* y *erf_trapecios.m*, en la presente entrega.

El módulo principal, *ejercicio2.m*, realiza llamadas a ambos métodos variando el número de iteraciones para cada método entre uno y diez, y calcula el error absoluto con respecto a la función *erf* de *Octave*. Como la aproximación por el método de los trapecios converge muy lentamente, se implementó una función que calcule el número de iteraciones necesarias para lograr un número de cifras significativas, que es pasada por parámetros. Todos estos resultados se escriben en un archivo, *resultados.txt*.

```
1;

function iteraciones = cifras_significativas_trapecios(file_id, d)
  n = 1;
  erf1 = erf(1);
  do
    resultado_trapecios = erf_trapecios(1, n);
    error_trapecios = erf1 - resultado_trapecios;
    n++;
  until (abs(error_trapecios) < 0.5*10^(-d));
  iteraciones = --n;
  fprintf(file_id, ...
      "Iteraciones de erf_trapecios para %d cifras significativas: %d\n resultado=%.10f error=%.10e\n\n"...
      , d, iteraciones, resultado_trapecios, error_trapecios);
endfunction

file_id = fopen('resultados.txt', 'w');
erf1 = erf(1);
fprintf(file_id, "erf(1) = %.10f (octave) \n\n", erf1);

fprintf(file_id, "%3s %17s %17s %17s %17s\n","n","res_taylor","error_taylor","res_trapecios","error_trapecios");
for n = 1:10
  resultado_taylor = erf_taylor(1,n);
  error_taylor = abs(erf1 - resultado_taylor);
  resultado_trapecios = erf_trapecios(1,n);
  error_trapecios = abs(erf1 - resultado_trapecios);
  fprintf(file_id, "%3d %17.10f %17.10e %17.10f %17.10e\n", n, resultado_taylor, error_taylor, resultado_trapecios, error_trapecios);
end

fprintf(file_id, "\n\n");
for cifras_significativas = 5:7
  cifras_significativas_trapecios(file_id, cifras_significativas);
end
fclose(file_id);
```

(c) A continuación se muestran los resultados de la corrida del módulo *ejercicio2.m* generados en el archivo *resultados.txt*:

```
erf(1) = 0.8427007929 (octave)

  n        res_taylor       error_taylor      res_trapecios    error_trapecios
  1      0.7522527781   9.0448014886e-02     0.7717433323   7.0957460692e-02
  2      0.8650906948   2.2389901824e-02     0.8252629556   1.7437837353e-02
  3      0.8382245241   4.4762688216e-03     0.8349853226   7.7154703151e-03
  4      0.8434485018   7.4770880382e-04     0.8383677774   4.3330155085e-03
  5      0.8425936691   1.0712389852e-04     0.8399297272   2.7710657149e-03
  6      0.8427142224   1.3429431295e-05     0.8407772220   1.9235709583e-03
  7      0.8426992967   1.4962190631e-06     0.8412879032   1.4128897987e-03
  8      0.8427009429   1.4999237341e-07     0.8416192212   1.0815717049e-03
  9      0.8427007793   1.3666073384e-08     0.8418463110   8.5448193686e-04
 10      0.8427007941   1.1411194212e-09     0.8420087166   6.9207633289e-04


Iteraciones de erf_trapecios para 5 cifras significativas: 118
 resultado=0.8426958242 error=4.9687409162e-06

Iteraciones de erf_trapecios para 6 cifras significativas: 372
 resultado=0.8427002930 error=4.9994652174e-07

Iteraciones de erf_trapecios para 7 cifras significativas: 1177
 resultado=0.8427007430 error=4.9940905522e-08
```

Las siguientes tablas muestran los resultados y las iteraciones necesarias para lograr una precisión de cinco, seis y siete cifras significativas con cada método de aproximación.

Cifras significativas	Resultado	Iteraciones
5	0.84270	7
6	0.842700	8
7	0.8427008	9

Cuadro 1: Resultados aproximando con polinomios de Taylor.

Cifras significativas	Resultado	Iteraciones
5	0.84270	118
6	0.842700	372
7	0.8427008	1177

Cuadro 2: Resultados aproximando con el método de los trapecios.

(d) Como puede observarse a simple vista aproximando la función error con polinomios de Taylor resultó mucho más eficiente que con el método de los trapecios. Se puede apreciar que en la primera iteración ambos métodos arrojaron resultados parecidos y con cotas de error similares.

Sin embargo, con solo siete iteraciones con la aproximación con Taylor se logró una precisión de cinco cifras significativas, mientras que con el método de los trapecios en la misma cantidad de iteraciones se logran solo 2 cifras significativas.

Aproximando por el polinomio de Taylor con nueve iteraciones se logra una precisión de siete cifras significativas. Es decir, va logrando una precisión de una cifra significativa por iteración. En cambio, se necesitaron 118 iteraciones para poder lograr cinco cifras significativas aproximando con el método de los trapecios y 1177 para poder lograr una precisión de siete cifras significativas.

Podemos concluir que, aunque ambos métodos convergen al resultado, la aproximación por polinomios de Taylor tiene una velocidad de convergencia mucho mayor que la aproximación por el método de los trapecios, por ende, la aproximación por polinomios de Taylor resulta un muy buen método para aproximar la función error.

5. Parte 3

(a) Tanto en la parte 1 como en la parte 2 del presente Trabajo Práctico nos encontramos con distintas fuentes de error; a continuación se describirán sus instancias de ocurrencia según la fuente de la cual estemos hablando.

- Errores Inherentes (EI): Son errores que existen en los datos de entrada del problema. En nuestro caso, al encontrarnos con números irracionales en los parámetros de entrada (π y e), estos deben ser aproximados a un número finito de dígitos, ya que no pueden representarse exactamente con la cantidad de dígitos significativos con los que nos encontramos trabajando. En nuestro caso particular, Octave por defecto trabaja con 5 dígitos significativos.
- Errores por truncamiento (ED): Una fuente de error que se origina en la utilización de un algoritmo determinado. En el caso del presente TP, los algoritmos utilizados en la parte 2 son una fuente de este tipo de error, ya que se utilizan una cantidad finita de iteraciones para aproximar el resultado. Las iteraciones omitidas introducen un error de truncamiento; en el caso de algoritmos con convergencia más lenta el error introducido será mayor (si la cantidad de iteraciones se mantiene).
- Errores por redondeo (ER): Error que se introduce al hacer cuentas con representaciones numéricas finitas. Cada operación aritmética que realicemos en Octave puede incurrir en un error de redondeo debido a la metodología de redondeo que utiliza y la cantidad de dígitos significativos empleados. El punto flotante es inherentemente una representación aproximada de un valor decimal y por ello acarreará nativamente un error de redondeo. Además, cuantas más operaciones realicemos, mayor será el arrastre del error; es por ello que los algoritmos empleados en la parte 2 corren un riesgo mayor con respecto a esta fuente de error que el Polinomio de Taylor en la parte 1 del presente TP.

(b)
- Problema Matemático (PM): En la parte 1, el problema matemático es la definición del Polinomio de Taylor presentada en la ecuación *(1)*. En cuanto a la segunda parte, la instancia es la función error presentada en el enunciado.
- Problema Numérico (PN): Las aproximaciones presentadas en *(3)*, *(11)* y *(14)* representan las instancias de problema numérico del trabajo práctico.
- Método Numérico (MN): Los pasos necesarios para resolver el problema matemático e implementarlo en una computadora.
- Algoritmo: Los algoritmos presentados en *erf_taylor.m* y *erf_trapecios.m*, además de la resolución particular del polinomio existente en *taylor_ej1.m*, ya expuestos en el presente Trabajo Práctico.

Referencias

[1] Error Function - Wikipedia, https://en.wikipedia.org/wiki/Error_function

[2] Richard L. Burden, J. Douglas Faires, *Numerical Analysis (9th edition)* Brooks/Cole, Cengage Learning, 2010, Sección 4.3, página 194

[3] Richard L. Burden, J. Douglas Faires, *Numerical Analysis (9th edition)* Brooks/Cole, Cengage Learning, 2010, Sección 4.4, página 206

[4] Taylor Series - Wolfram MathWorld, http://mathworld.wolfram.com/TaylorSeries.html