Informe Laboratorio 4 Computación Científica

Josue Peña Atencio

Mayo 2020

Funciones de aproximación de derivadas

Las funciones para realizar las aproximaciones y comparaciones fueron definidas manualmente mediante el manejador de funciones de matlab.
Para todos los casos, se usó el rango de números $[10, 110]$ y $h = 0.001$ para calcular la derivada de cada punto. La cantidad de puntos a aproximar es 100.

En las tablas de valores aproximados y analíticos, sólo se muestran los valores para los primeros 5 puntos de la función $f_1'(x)$ y $f_1''(x)$ (ya que estas fueron las que mostraron mayor error). Sin embargo, en el cálculo de error para cada método se emplean los 100 puntos al igual que el resto de funciones a aproximar.

Para facilitar el cálculo de todos los puntos, se empleó la función de Matlab "`arrayfun`" junto con funciones anónimas, de tal forma que las aproximaciones e incluso cálculos de los valores analíticos se pudieran hacer en una sóla línea. Esto fue usado tanto para las derivadas como para las integrales.

Método de aproximación	**Función implementada en MATLAB**
Fórmula de diferencias finitas hacia adelante	`dif_adelante.m`
Fórmula de diferencias finitas hacia atrás	`dif_atras.m`
Fórmula de diferencias centrada	`dif_centrada.m`
Fórmula de segunda derivada	`seg_derivada.m`

Funciones usadas y sus derivadas

$f_1(x) = ln(x) + x^5 + 4x^2$
$f_1'(x) = \frac{1}{x} + 5x^4 + 8x$
$f_1''(x) = -\frac{1}{x^2} + 20x^3 + 8$

$f_2(x) = e^{-x} + sin(x) + 3x$
$f_2'(x) = -e^{-x} + cos(x) + 3$
$f_2''(x) = e^{-x} - sin(x)$

De orden 1

Tiempo

Función Matlab	**Tiempo (s)**	**Función parámetro**
`dif_adelante.m`	0.00561300000000000	$f_1(x)$
`dif_atras.m`	0.00464400000000000	$f_1(x)$

`dif_adelante.m`	0.00487100000000000	$f_2(x)$
`dif_atras.m`	0.00445000000000000	$f_2(x)$

Valores calculados

Función Matlab: `dif_adelante.m`		
Función aproximada: $f_1'(x)$		
X	**Derivada aproximada**	**Derivada analítica**
10	50090.1049950189	50080.1000000000
11	73575.7830282382	73562.4311198144
12	104493.564553763	104476.191692625
13	144267.626896151	144245.497206426
14	194447.069925722	194419.385695086

Función Matlab: `dif_atras.m`		
Función aproximada: $f_1'(x)$		
X	**Derivada aproximada**	**Derivada analítica**
10	50070.0970049365	50080.1000000000
11	73549.0816357196	73562.4311198144
12	104458.821721084	104476.191692625
13	144223.370912310	144245.497206426
14	194391.705406830	194419.385695086

Error cuadrático medio

Función Matlab	**ECM**	**Función parámetro**
`dif_adelante.m`	28454982.8776500	$f_1(x)$
`dif_atras.m`	28453755.1170438	$f_1(x)$
`dif_adelante.m`	1.24446457843256e-07	$f_2(x)$
`dif_atras.m`	1.24445930601077e-07	$f_2(x)$

De orden 2

Tiempo

Función Matlab	Tiempo (s)	Función parámetro
`dif_centrada.m`	0.004536000000000	$f_1(x)$
`seg_derivada.m`	0.002009000000000	$f_1'(x)$
`dif_centrada.m`	0.001937000000000	$f_2(x)$
`seg_derivada.m`	0.001858000000000	$f_2'(x)$

Valores calculados

Función Matlab: `dif_centrada.m`		
Función aproximada: $f_1'(x)$		
X	**Derivada aproximada**	**Derivada analítica**
10	50080.1009999777	50080.1000000000
11	73562.4323319789	73562.4311198144
12	104476.193137423	104476.191692625
13	144245.498904231	144245.497206426
14	194419.387666276	194419.385695086

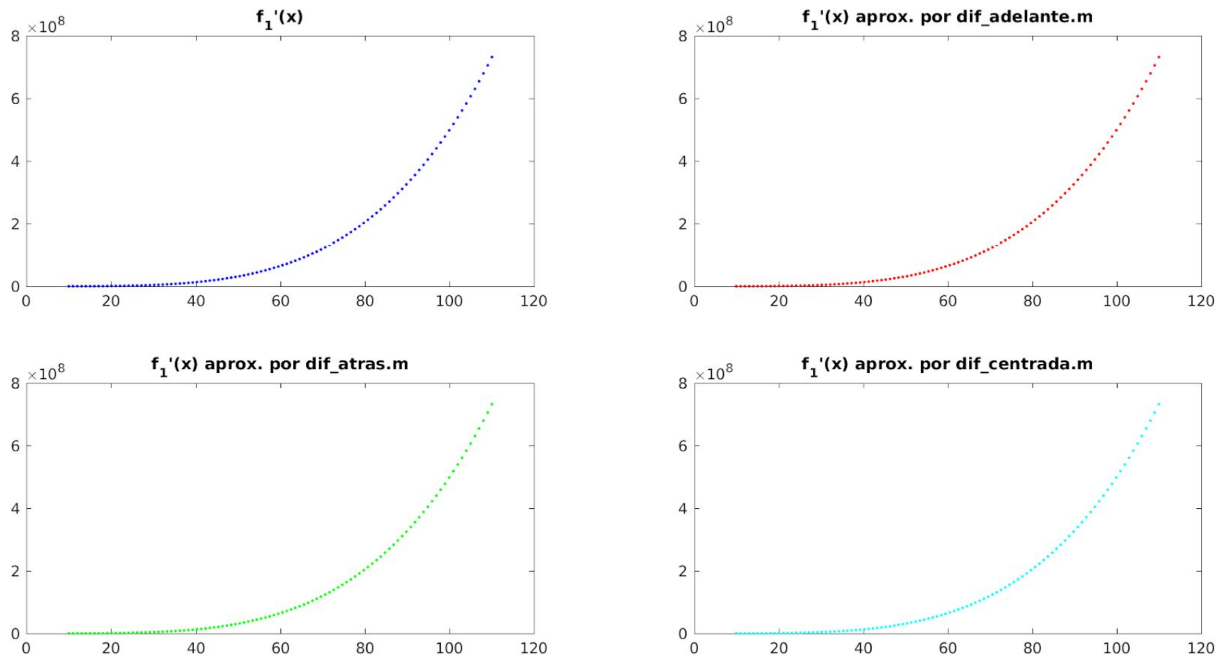
Función Matlab: `seg_derivada.m`		
Función aproximada: $f_1''(x)$		
X	**Derivada aproximada**	**Derivada analítica**
10	20007.9900823766	20007.9900000000
11	26701.3925185893	26701.3924447073
12	34742.8326786030	34742.8325458883
13	44255.9838411398	44255.9836475757
14	55364.5188920200	55364.5190502577

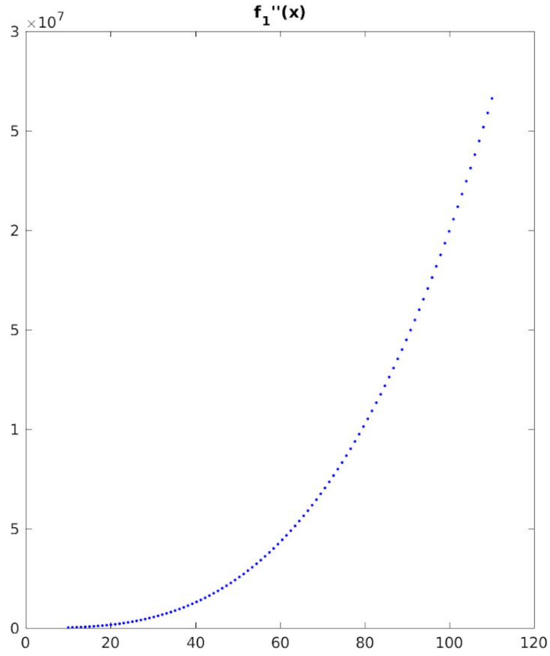
Error cuadrático medio

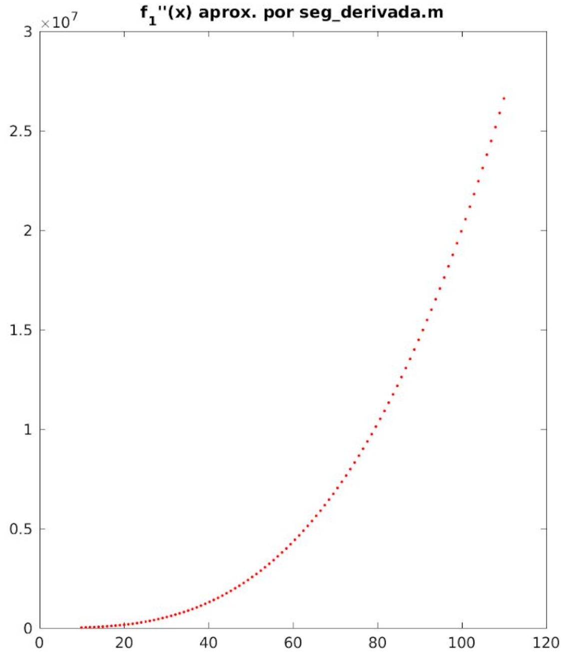
Función Matlab	ECM	Función parámetro
`dif_centrada.m`	0.00339704043458646	$f_1(x)$
`seg_derivada.m`	1.23989476611968	$f_1(x)$
`dif_centrada.m`	1.39505652514597e-14	$f_2(x)$
`seg_derivada.m`	1.00316054971283	$f_2(x)$

Gráficas

En la siguiente página se anexan las gráficas de la derivada analítica de la función $f_1'(x)$ y $f_1''(x)$ haciendo una comparación con los diferentes métodos usados.







Funciones de aproximación de integrales mediante regla de la cuadratura

Para todos los casos, se usó el rango de números $[0.1,\ 2.0]$ para calcular la integral. Para el caso de los métodos de cuadratura compuesta, la cantidad de sub-intervalos o paneles es 500 en cada uno.

Funciones usadas y sus integrales indefinidas

$f_3(x) \ = e^{-2x} + cos(x)$
$F_3(x) \ = -\frac{1}{2}e^{-2x} + sin(x)$

$f_4(x) \ = \frac{1}{x} + sin(x)$
$F_4(x) \ = ln(|x|) - cos(x)$

Método de aproximación	**Función implementada en MATLAB**
Regla del rectángulo	`rectangulo.m`
Regla del trapezoide	`trapezoide.m`
Regla de simpson	`simpson.m`
Regla del rectángulo compuesta	`rectangulo_comp.m`
Regla del trapezoide compuesta	`trapezoide_comp.m`
Regla de simpson compuesta	`simpson_comp.m`

Simples

Tiempo

Función Matlab	Tiempo (s)	Función parámetro
`rectangulo.m`	0.000805000000000000	$f_3(x)$
`trapezoide.m`	0.00355900000000000	$f_3(x)$
`simpson.m`	0.00145100000000000	$f_3(x)$
`rectangulo.m`	0.000338000000000000	$f_4(x)$
`trapezoide.m`	0.000251000000000000	$f_4(x)$
`simpson.m`	0.00102500000000000	$f_4(x)$

Cálculo de valores

Función Matlab	Integral aproximada	Integral analítica	Función
`rectangulo.m`	1.17805220467495	1.20967156727348	$f_3(x)$
`trapezoide.m`	1.34510853466272	1.20967156727348	$f_3(x)$
`simpson.m`	1.23373764800420	1.20967156727348	$f_3(x)$
`rectangulo.m`	3.45762793815244	4.40688327537916	$f_4(x)$
`trapezoide.m`	10.9336743012989	4.40688327537916	$f_4(x)$
`simpson.m`	5.94964339253459	4.40688327537916	$f_4(x)$

Error cuadrático medio

Función Matlab	ECM	Función parámetro
`rectangulo.m`	0.000999784091137342	$f_3(x)$
`trapezoide.m`	0.0183431721355945	$f_3(x)$
`simpson.m`	0.000579176241737865	$f_3(x)$
`rectangulo.m`	0.901085695253409	$f_4(x)$
`trapezoide.m`	42.5990010960262	$f_4(x)$

simpson.m	2.38010877908544	$f_4(x)$

Compuestas

Tiempo

Función Matlab	Tiempo (s)	Función parámetro
rectangulo_comp.m	0.00652300000000000	$f_3(x)$
trapezoide_comp.m	0.00161400000000000	$f_3(x)$
simpson_comp.m	0.00631100000000000	$f_3(x)$
rectangulo_comp.m	0.000665000000000000	$f_4(x)$
trapezoide_comp.m	0.00110400000000000	$f_4(x)$
simpson_comp.m	0.00512900000000000	$f_4(x)$

Cálculo de valores

Función Matlab	Integral aproximada	Integral analítica	Función
rectangulo_comp.m	1.40002985706945	1.20967156727348	$f_3(x)$
trapezoide_comp.m	1.30035926928709	1.20967156727348	$f_3(x)$
simpson_comp.m	1.36680632780867	1.20967156727348	$f_3(x)$
rectangulo_comp.m	6.41182180303295	4.40688327537916	$f_4(x)$
trapezoide_comp.m	4.91199373788387	4.40688327537916	$f_4(x)$
simpson_comp.m	5.91187911464992	4.40688327537916	$f_4(x)$

Error cuadrático medio

Función Matlab	ECM	Función parámetro
rectangulo_comp.m	0.0362362784940486	$f_3(x)$
trapezoide_comp.m	0.00822425929650991	$f_3(x)$

`simpson_comp.m`	0.0246913329684512	$f_3(x)$
`rectangulo_comp.m`	4.01977849967054	$f_4(x)$
`trapezoide_comp.m`	0.255136579331719	$f_4(x)$
`simpson_comp.m`	2.26501247622230	$f_4(x)$

Análisis y conclusiones

Como se pudo observar en las aproximaciones de derivadas mediante las fórmulas de diferencias finitas hacia atrás y hacia adelante y con la de diferencias centrada (las cuales demostraron resultados más o menos variados), la precisión de la aproximación puede depender grandemente con respecto a la función que se esté aproximando. En el caso de $f_1(x)$, se obtuvieron resultados de error muchísimo más elevados que con respecto a $f_2(x)$.

La razón de esto puede ser debido a que $f_1(x)$ cuenta con un polinomio de grado más o menos alto (5), mientras que $f_2(x)$ tiene un polinomio de grado 1. Al ser un polinomio de tal grado, el error en la aproximación se propaga con mayor facilidad.

De todas las aproximaciones de derivadas, la que fue más precisa fue la de Fórmula de diferencias centrada, con un valor de error de tan sólo 0.003.

Algo destacable de los métodos para las derivadas es lo eficientes que son computacionalmente. Calcularlos toma tiempo constante, por lo cual aproximar n puntos tomaría tiempo $O(n)$.

Los métodos simples de aproximación de integrales también gozan de una ventaja de rendimiento computacional bastante grande. Con tan sólo la función base y el rango, se puede computar la aproximación a una integral de forma constante.

Curiosamente, los métodos de cuadratura simples fueron los métodos que demostraron mejor precisión con respecto al rango en el que se calcularon las integrales. El método con menor error fue el de la regla del punto medio o rectángulo, con tan sólo 0.0009 de error cuadrático.

Lo anterior no quiere decir que los métodos compuestos sean imprecisos a pesar de haber usado 500 particiones (la mayoría tuvo cálculos de error por debajo de 0.03). Experimentalmente se descubrió en el laboratorio de que si se aumenta el tamaño del rango para aproximar la integral, los métodos simples empiezan a ser mucho más imprecisos (con valores de error de muchos órdenes de magnitud más grandes), mientras que los compuestos mantienen su precisión. Esto pagando el precio de rendimiento computacional, ya que estos deben calcular n valores diferentes dependiendo del número de particiones que se establezca.