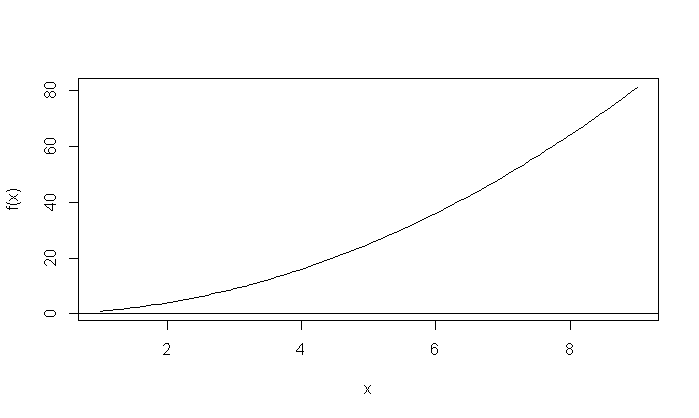
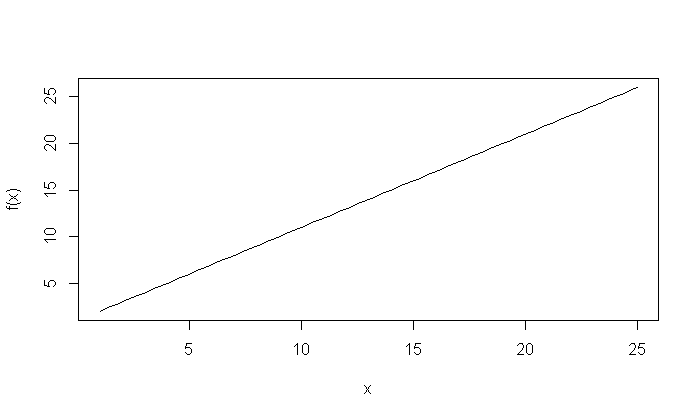
José nicolas garcia pineda parcial 1 analisis numerico

1).Sea f(n) la eﬁciencia del algoritmo, medida como el nu´mero m´ınimo de operaciones requeridas para resolver el problema

1. Implemente en R o Python un algoritmo que le permita sumar u´nicamente los elementos de la sub matriz triangular superior o triangular inferior, dada la matriz cuadrada An. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notacio´n O() con una gra´ﬁca que muestre su orden de convergencia

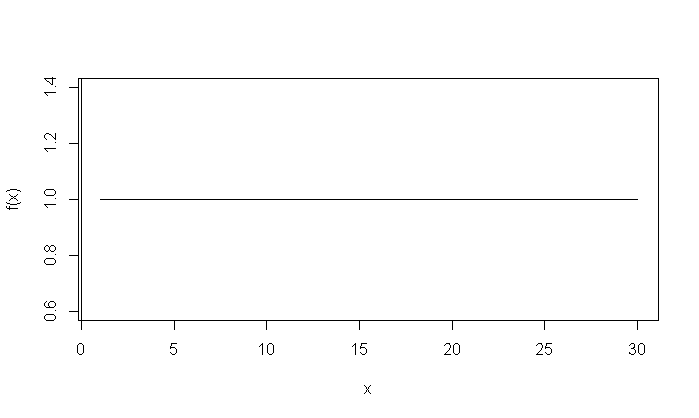
#*la función tomada es gracias al orden de convergencia por O(n+1) lo que nos dice que nuestro algoritmo es de complejidad O(n)*

1. . b) Implemente en R o Python un algoritmo que le permita sumar los elementos de una matriz cuadrada An. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notacio´n O() con una gra´ﬁca que muestre su orden de convergencia.



*#Teniendo en cuenta que R soporta algebra vectorial, la libreria SUM cuenta con cnotación O(n^2)*

1. c) Implemente en R o Python un algoritmo que le permita sumar los n2 primeros nu´meros naturales al cuadrado. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notacio´n O() con una gr´aﬁca que muestre su orden de convergencia.



*#Como en el algoritmo creado no usamos ningún ciclo y sólo operaciones algebraicas tenermos una complejidad*

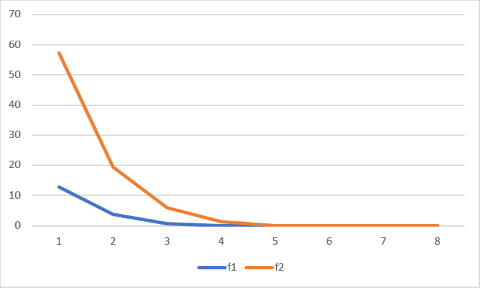
*# de O de una constante O(k) lo que podemos expresar en una complejidad constante o función*

2. En R:Sean f(x) = ln(x + 2) y g(x) = sin(x) dos funciones de valor real.

|  |
| --- |
| ----------------------------------------------------------  x1 x2  ----------------------------------------------------------  -1.0000000 -1.4557541  -1.455754140 -1.455754140  -1.4557541 -1.5856224  -1.585622437 -1.585622437  -1.5856224 -1.6194091  -1.619409121 -1.619409121  -1.6194091 -1.6282683  -1.628268284 -1.628268284  -1.6282683 -1.6306046  -1.630604586 -1.630604586  -1.6306046 -1.6312218  -1.631221818 -1.631221818  -1.6312218 -1.6313850  -1.631384967 -1.631384967  -1.6313850 -1.6314281  -1.631428097 -1.631428097  -1.6314281 -1.6314395  -1.631439499 -1.631439499  -1.6314395 -1.6314425  -1.631442514 -1.631442514  -1.6314425 -1.6314433  -1.631443311 -1.631443311  -1.6314433 -1.6314435  -1.631443521 -1.631443521  -1.6314435 -1.6314436  -1.631443577 -1.631443577  -1.6314436 -1.6314436  -1.631443592 -1.631443592  -1.6314436 -1.6314436  -1.631443596 -1.631443596  -1.6314436 -1.6314436  -1.631443597 -1.631443597  -1.6314436 -1.6314436  -1.631443597 -1.631443597  ----------------------------------------------------------  Cero de f es approx: -1.6314435968711 iteraciones 17 |
|  |
| |  | | --- | | > | |

3. En cada siguiente ejercicio solucionar por el metodo indicado.Implemente en R o Python, debe determinar el nu´mero de iteraciones realizadas,una graﬁca que evidencie el tipo de convergencia del m´etodo y debe expresarla en notacio´n O()

a) Newton: Determine el valor de los coeﬁcientes a y b tal que f(1) = 3 y f(2) = 4 con f(x) = a + (ax + b)eax+b. Obtenga la respuesta con E = 10−6



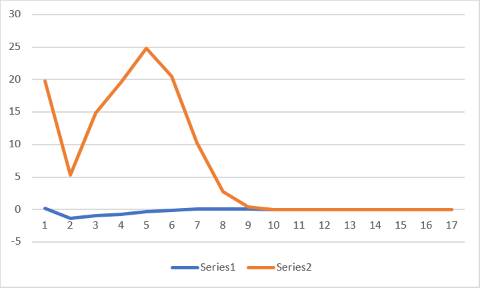
convergence!  6.96812285587e-09  <  1e-06   nre iter: 7

soluion con newton: [ a=0.15787718

b=0.86452203]

O(n^2) el método de newton converge cuadráticamente

b) Newton mejorado: Determine el valor de los coeﬁcientes a y b tal que f(1) = 3 y f(2) = 4 con f(x) = a + (ax + b)eax+b. Obtenga la respuesta con E = 10−



convergence!  -6.675615926710066e-07  <  1e-06   nre iter: 16

soluion con newton\_mejorado: [ a=3.1830518086366233,

b=5.869108380233252]

Cuando se tiene existencia de raíces múltiples, tanto el método de Newton-Raphson como el de la secante convergen linealmente.  
El metodo de Newton-Raphson modificado el cual se describe acontinuacion consiste en aplicar el metodo de Newton-Raphson univariable dos veces(para el caso de un sistema de n ecuaciones no lineales con n incógnitas, se aplicara n veces), una para cada variable.

O(4n) convergencia de 2n funciones por paso (cuatro para el caso de dos ecuaciones que se esta manejando)