

Parcial 1

Análisis Numérico

Ejercicio escogido para realizar: **3.C**

3. En los siguientes ejercicios aplicar el Teorema de Taylor para aproximar la función $f(x)$ con un polinomio de Taylor alrededor de $a=0$ (de menor error), estimar el error para cada x , realizar una gráfica que muestre el polinomio de aproximación. Implemente en R o Python, utilizar 9 cifras significativas

c) $f(x) = \ln(1+x)$ en $[-0.5, 0.5]$ para $x=0.005, 0.0001, 0.499999999$

La implementación del algoritmo fue realizada en Python y a continuación se muestran los resultados:

Para $x=0.005$ (El resultado se muestra al final de la imagen)

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
1: Python Debug Cons... +  [ ]  ^  x

(n=0) (aprox.-0.0) (error=0.00498754151103897)
(n=1) (aprox.-0.005) (error=0.00498754151103897)
(n=2) (aprox.-0.0049875) (error=0.0000124504089183222)
(n=3) (aprox.-0.004987541056666666) (error=4.15110389881908E-8)
(n=4) (aprox.-0.004987541510416666) (error=1.55627698228411E-10)
(n=5) (aprox.-0.004987541511041666) (error=6.22301689345850E-13)
(n=6) (aprox.-0.0049875415110390625) (error=2.69836236688192E-15)
(n=7) (aprox.-0.004987541511039074) (error=9.45424284487268E-17)
(n=8) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=9) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=10) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=11) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=12) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=13) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=14) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=15) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=16) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=17) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=18) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=19) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=20) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=21) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=22) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=23) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=24) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=25) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=26) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=27) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=28) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=29) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=30) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=31) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=32) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=33) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=34) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=35) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=36) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=37) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=38) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=39) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=40) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=41) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=42) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=43) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=44) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=45) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=46) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=47) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=48) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
(n=49) (aprox.-0.004987541511039074) (error=1.05818132034585E-16)
La mejor aprox. al valor 0.005 con el menor error posible es: 0.004987542

Ln 20, Col 8  Spaces: 4  UTF-8  CR/LF  Python  8:22 AM  3/5/2021
```

Comprobación de respuesta con Wolfram:



↪

= 0.004987541511039

Extended Keyboard

Upload

Alumno: Daniel Ricardo Ramírez Molina

Para $x=0.0001$ (El resultado se muestra al final de la imagen)

[illegible]

Comprobación de respuesta con Wolfram:



Alumno: Daniel Ricardo Ramírez Molina

Para $x=0.049999999$ (El resultado se muestra al final de la imagen)

```

TERMINAL      PROBLEMS      OUTPUT      DEBUG CONSOLE
(=0) (aprox.-0.0) (error=0.4054651074414908)
(=1) (aprox.-0.499999999) (error=0.4054651074414908)
(=2) (aprox.-0.37499999949999996) (error=0.0945348915585023)
(=3) (aprox.-0.416666669166666) (error=0.034051079414977)
(=4) (aprox.-0.40104166684166663) (error=0.012015558731690)
(=5) (aprox.-0.4072916659701666) (error=0.0042344139383103)
(=6) (aprox.-0.4046874993437499) (error=0.00182655853766894)
(=7) (aprox.-0.4058035707566964) (error=0.00077608977447740)
(=8) (aprox.-0.40531528951458085) (error=0.000336463131598712)
(=9) (aprox.-0.405532339394915) (error=0.00019017026088010)
(=10) (aprox.-0.4054346471514446) (error=0.000071959579938153)
(=11) (aprox.-0.40547983635501353) (error=0.0000304602900530471)
(=12) (aprox.-0.4054586913034185) (error=0.0000130289135158704)
(=13) (aprox.-0.4054488813727128) (error=0.0000046133097016830)
(=14) (aprox.-0.405453721673317) (error=0.0000027308571512998)
(=15) (aprox.-0.405465761784043) (error=0.00000138576818065982)
(=16) (aprox.-0.40546480250417843) (error=6.48736966646357E-7)
(=17) (aprox.-0.4054652512920768) (error=3.049371310228760E-7)
(=18) (aprox.-0.40546508393445355) (error=1.4385957915154E-7)
(=19) (aprox.-0.4054513975121240) (error=6.807709319170427E-8)
(=20) (aprox.-0.4054650200753097) (error=1.23072722270009E-8)
(=21) (aprox.-0.40546511477404135) (error=1.53739866948222E-8)
(=22) (aprox.-0.40546510393683366) (error=7.33254368334713E-9)
(=23) (aprox.-0.405465109110946) (error=3.50466400345084E-9)
(=24) (aprox.-0.40545186363137524) (error=1.67038030919779E-9)
(=25) (aprox.-0.40546510782041205) (error=8.85178423934905E-10)
(=26) (aprox.-0.4054651072552905) (error=3.80914389416404E-10)
(=27) (aprox.-0.40546510753123793) (error=1.86207160801644E-10)
(=28) (aprox.-0.40546510739818106) (error=9.5748207700223E-11)
(=29) (aprox.-0.405465107402421) (error=4.33083003072307E-11)
(=30) (aprox.-0.4054651074313769) (error=2.4923318632369E-11)
(=31) (aprox.-0.40546510744639824) (error=1.01207375813317E-11)
(=32) (aprox.-0.4054651074391222) (error=4.90857994184667E-12)
(=33) (aprox.-0.40546510744205) (error=2.3753772323670E-12)
(=34) (aprox.-0.405451074093080) (error=1.12325598904960E-12)
(=35) (aprox.-0.4054651074417066) (error=5.5967051562010E-13)
(=36) (aprox.-0.4054651074413654) (error=2.71949129891932E-13)
(=37) (aprox.-0.405465107441562) (error=1.32283073384087E-13)
(=38) (aprox.-0.40546510744146624) (error=6.43374242770270E-14)
(=39) (aprox.-0.4054651074415129) (error=3.1493115600919E-14)
(=40) (aprox.-0.4054651074414901) (error=1.52100554373646E-14)
(=41) (aprox.-0.4054651074415012) (error=7.54951656745106E-15)
(=42) (aprox.-0.4054651074414958) (error=3.5527136708005E-15)
(=43) (aprox.-0.4054651074414984) (error=1.8873791410277E-15)
(=44) (aprox.-0.40546510744149710) (error=7.77155117237610E-16)
(=45) (aprox.-0.4054651074414978) (error=4.9950036188120E-16)
(=46) (aprox.-0.40546510744149744) (error=1.11022302462516E-16)
(=47) (aprox.-0.4054651074414976) (error=2.20446049259316E-16)
(=48) (aprox.-0.40546510744149755) (error=5.5511515123157E-17)
(=49) (aprox.-0.4054651074414976) (error=1.11022302462516E-16)
La mejor aprox. al valor 0.999999999 con el menor error posible es: 0.405465107

```

Comprobación de respuesta con Wolfram:



Alumno: Daniel Ricardo Ramírez Molina

La grafica de la función de aproximación se programó en Python y se muestra a continuación (También se puede visualizar al correr el código)

