**TRABAJO PRACTICO Nro 1**

**INTRODUCCIÓN**

*Alumno:* ***Mara Leandro David***

*Fecha de entrega de enunciado: miércoles 24 de agosto de 2016*

*Fecha de entrega del práctico resuelto:* ***miércoles 14 de septiembre de 2016***

**Ejercicio 1:**

El valor de= puede computarse usando la siguiente expansión en series:

Use =1.1 y compare los valores deobtenidos con una cantidad diferente de términos en la serie.

**Resultado de **

>> x=1.1;

log(x+1)-log(x-1) ans = **3.0445**

**Resultado de la serie para 5 términos.**

**>> x=1.1; s=0; impar=1;**

**for i=1:5**

**s=(1/(impar\*(x^impar)))+s;**

**impar=impar+2;**

**end**

**salida=s\*2 salida = 2.8083**

**Resultado de la serie para 20 términos.**

**>> x=1.1; s=0; impar=1;**

**for i=1:20**

**s=(1/(impar\*(x^impar)))+s;**

**impar=impar+2;**

**end**

**salida=s\*2 salida = 3.0398**

**Resultado de la serie para 100 términos.**

>> x=1.1; s=0; impar=1;

for i=1:100

s=(1/(impar\*(x^impar)))+s;

impar=impar+2;

end

salida=s\*2 salida = **3.0445**

Para que la expansión de la serie se aproxime al valor de la **** en =1.1, se debe computar la serie con más de 20 términos.

**Ejercicio 2:**

La constante de amortiguación *c* de la suspensión mostrada en la figura está dada por:



Donde µ es la viscosidad del fluido.

Sugiera un procedimiento para encontrar la influencia de pequeños errores en *a*, *h*, *r* y *l* sobre *c* para los valores de referencia: µ=0.3445 Pa.s, *l*=10 cm, *h*=0.1 cm, *a*=2cm, y *r*=0.5 cm.

**procedimiento** = @(l,h,a,r,u) (6 \* pi \* u \* l / (h^3))\*(((a - h/2)^2) - (r^2))\*((((a^2)-(r^2)) / (a - (h/2)))-h);

resultado=procedimiento (10, 0.1, 2, 0.5,0.3445)

**resultado =4.2056e+05**

Utilice el procedimiento para predecir el valor de *c* bajo las siguientes condiciones:

1. *l*=9.999 cm, *h*=0.009 cm, *a*=1.999 cm y *r*=0.499 cm

*resultado1=procedimiento (9.999, 0.009, 1.999, 0.499,0.3445);*

***resultado1 =6.2098e+08***

*errorA=resultado1-resultado*

***errorA =6.2056e+08***

*errorR=errorA/resultado*

***errorR = 1.4756e+03***

1. *l*=10.001 cm, *h*=0.101 cm, *a*=2.001 cm y *r*=0.501 cm

resultado2=procedimiento (10.001, 0.101, 2.001, 0.501,0.3445)

**resultado2 =4.0835e+05**

errorA=resultado2-resultado

**errorA =-1.2209e+04**

errorR=errorA/resultado

**errorR =-0.0290**

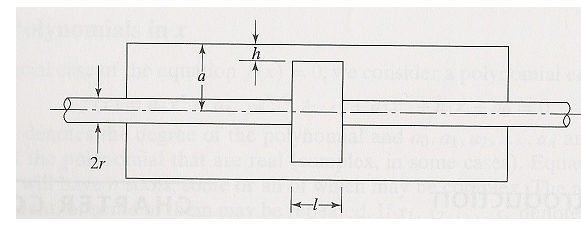
1. *l*=9.999 cm, *h*=0.101 cm, *a*=2.001 cm y *r*=0.499 cm
2. Extraiga conclusiones

**Ejercicio 3:**

Considere los números de Fibonacci , definidos como = 1 y = + i=1,2,…. Estos números se usan en muchas aplicaciones, incluyendo la minimización de una función. Se sabe que la razón: = converge a medida que x→∞ . Esto se conoce como la razón dorada. Determine el error relativo involucrado en aproximar mediante para para n=1,2,…10.

**Ejercicio 4:**

Usando MATLAB que provee una rutina para encontrar raíces, compute las 20



Usando MATLAB que provee una rutina para encontrar raíces, compute las 20 raíces  computadas para *k*=1,2, … 20 del polinomio P donde

P(x)= x20 – 210 x19 + 20615 x18 – 1256850 x17 + 53327946 x 16

–1672280820 x15 + 40171771630 x14 – 756111184500 x13

+11310276995381 x12 – 135585182899530 x11

+1307535010540395 x10 – 10142299865511450 x9

+63030812099294896 x8 – 311333643161390640 x7

+1206647803780373360 x6 – 3599979517947607200 x5

+8037811822645051776 x4 – 12870931245150988800 x3

+13803759753640704000 x2 – 8752948036761600000 x

+2432902008176640000

Use una rutina que sea capaz de computar raíces complejas en doble precisión.

La fórmula dada para P es la forma expandida del polinomio de Wilkinson p(x):

p(x)= (x-20)(x-19)(x-18)(x-17)(x-16)(x-15)(x-14)(x-13)(x-12)(x-11)(x-10)

(x-9)(x-8)(x-7)(x-6)(x-5)(x-4) (x-3)(x-2)(x-1)

Chequee la calidad de las raíces  computadas para k=1,2…20 calculando: ,  y .Explique

>> lista = 1:20; factores = [lista\*0+1; -lista]';

polinomio = [1];

for ind=1:20,

polinomio = conv (polinomio, factores(ind,:));

end

roots (polinomio),

ans =

20.0003

18.9981

18.0065

16.9875

16.0139

14.9920

13.9996

13.0046

11.9963

11.0012

10.0001

8.9997

8.0001

7.0000

6.0000

5.0000

4.0000

3.0000

2.0000

1.0000