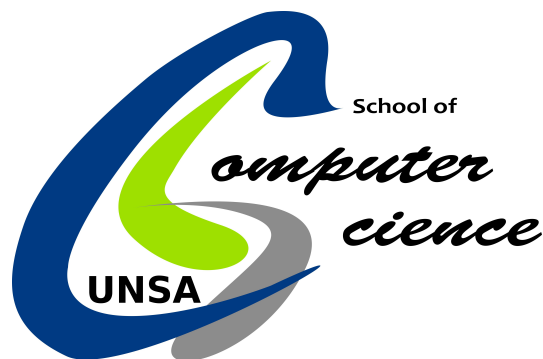


UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN



TEMA:

MÉTODO DE LAS POTENCIAS EN FORTRAN

Curso:

MATEMÁTICA APLICADA A LA COMPUTACIÓN

Presentado por:

Christofer Chávez Carazas

Arequipa - Perú
2017

1. Código

```
program potencias

  integer :: i = 0
  integer :: j = 0
  integer :: m = 2
  integer :: n = 2
  real(16), dimension(3,3) :: matrix
  real(16), dimension(3) :: ini

  matrix(1,1) = 3
  matrix(1,2) = -1
  matrix(1,3) = 0

  matrix(2,1) = -1
  matrix(2,2) = 2
  matrix(2,3) = -1

  matrix(3,1) = 0
  matrix(3,2) = -1
  matrix(3,3) = 3

  ini(1) = 1
  ini(2) = 1
  ini(3) = 1

  call Mpotencias(matrix,ini,3,10)

end program potencias

subroutine Mpotencias(M,ini,N,iter)
  integer :: i = 0
  real(16), dimension(N,N) :: M
  real(16), dimension(N) :: v
  real(16), dimension(N) :: u
  real(16), dimension(N) :: ini
  real(16) :: dominante,temp
  v = ini
  do i = 1,iter
    call mul(M,v,N,u)
    call mayor(u,N,dominante)
    temp = 1.0 / dominante
    call mulnum(u,temp,N,v)
    call printMatriz(u,N,1)
    write(*,*)
    call printMatriz(v,N,1)
    write(*,*)
  end do
return
end subroutine Mpotencias

subroutine mayor(M,N,res)
  real(16), dimension(N) :: M
  real(16) :: res,temp
  integer :: i
  res = -1
  do i = 1,N
    temp = abs(M(i))
    if((res == -1) .or. (res < temp)) then
      res = temp
    end if
  end do
return
end subroutine mayor

subroutine printMatriz(M,f,c)
  integer :: i,j
  integer :: f,c
  real(16), dimension(f,c) :: M
  do i=1,f
    do j=1,c
      write(*,'(F10.6,$)') M(i,j)
    end do
    write(*,*)
  end do
return
end subroutine printMatriz

subroutine mul(A,B,N,res)
  integer :: N,i,j,k
  real(16), dimension(N,N) :: A
  real(16), dimension(N) :: B
  real(16), dimension(N) :: res
  real(16) :: sum
  do i=1,N
    sum = 0
    do j=1,N
      sum = sum + B(j) * A(i,j)
    end do
    res(i) = sum
  end do
return
end subroutine mul

subroutine mulnum(A,x,N,res)
  integer :: N,i
  real(16) :: x
  real(16), dimension(N) :: A,res
  do i=1,N
    res(i) = A(i) * x
  end do
return
end subroutine mulnum
```

```
do i=1,N
  sum = 0
  do j=1,N
    sum = sum + B(j) * A(i,j)
  end do
  res(i) = sum
end do
return
end subroutine mul

subroutine mulnum(A,x,N,res)
  integer :: N,i
  real(16) :: x
  real(16), dimension(N) :: A,res
  do i=1,N
    res(i) = A(i) * x
  end do
return
end subroutine mulnum
```

2. Ejemplo

La matriz usada en este ejemplo es la siguiente:

$$A = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix}$$

Número de iteraciones: 5

3. Resultados

Se imprime primero la matriz u y luego la matriz v en cada iteración.

```
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/MAC/MPotencia$ ./a.out
2.000000
0.000000
2.000000

1.000000
0.000000
1.000000

3.000000
-2.000000
3.000000

1.000000
-0.666667
1.000000

3.666667
-3.333333
3.666667

1.000000
-0.909091
1.000000

3.909091
-3.818182
3.909091

1.000000
-0.976744
1.000000

3.976744
-3.953488
3.976744

1.000000
-0.994152
1.000000
```

```

integer :: N,i
real(16) :: x
real(16), dimension(N) :: A,res
do i=1,N
  res(i) = A(i) * x
end do
return
end subroutine mulnum

\end{lstlisting}
\section{Ejemplo}

La matriz usada en este ejemplo es la siguiente

\begin{array}{ccc}
3 & -1 & 0 \\
-1 & 2 & -1 \\
0 & -1 & 3
\end{array}

\textbf{Número de iteraciones: } 55

\onecolumn

\section{Resultados}

Se imprime primero la matriz $u$ y luego la matriz $v$

\begin{figure}
\centering
\includegraphics[scale = 0.5]{1.png}
\end{figure}

```

Log and Messages | Output | Konsole | PDFLaTeX | doc.tex => doc.pdf (pdflatex) | 0 errors, 0 warnings, 0 badboxes | Done!