UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INGENIERÍA MECÁNICA

TAREA N° 2
COMPUTACIÓN CIENTÍFICA IPM458
Magíster en Ciencias de la Ingeniería Mecánica
1er. semestre 2017

<u>Antecedentes</u>
Fecha de publicación: jueves 04 de mayo
Fecha de entrega: lunes 15 de mayo, hasta las 17:00 hrs.
Lugar de entrega: en Secretaría del Departamento de Ingeniería Mecánica.

1. Demostrar que la factorización LU de una matriz A, puede ser utilizada para calcular su matriz inversa A^{-1}
2. Una *norma* es una función que toma valores reales y que proporciona una "medida" del tamaño o longitud de entidades matemáticas, como los vectores y las matrices. Investigue, defina y explique la *norma de Frobenius* y la *norma infinita ó norma matricial uniforme ó norma renglón suma*. ¿Se pueden utilizar las normas anteriores para evaluar el número de condición de una matriz?, comente.
3. Utilizando el programa disponible en el moodle para resolver un sistema lineal de ecuaciones mediante la factorización LU, determinar:
    a. las fuerzas y reacciones sobre armadura de la figura N°1.
    b. determinar la el número de condición y la inversa del sistema.

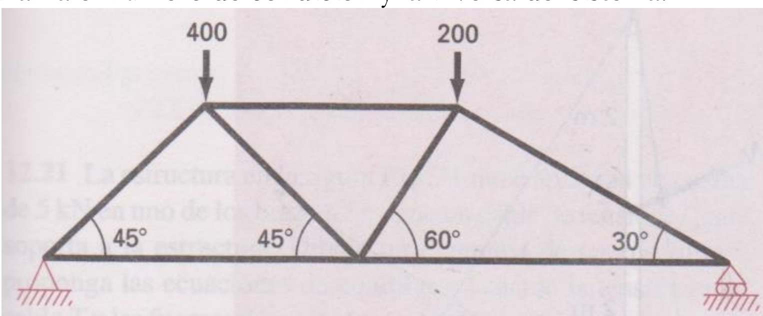

Figura N°1

4. Calcular las fuerzas y reacciones sobre la armadura de la figura N°1, considerando adicionalmente la acción de un viento de 1000 [N] cuando sopla:
    a. en dirección +x
    b. en dirección –x
    c. en cuál caso es más desfavorable?

Nota: se puede considerar la acción del viento como cargas puntuales sobre los nodos

5. Desarrolle un programa computacional en FORTRAN, amigable para el usuario, que permita resolver la pregunta N°3, utilizando el método de eliminación de Gauss con pivoteo parcial escalado. Compare sus resultados. ¿Es importante en este caso la técnica de pivoteo parcial escalado? Comente.

Nota: Para el desarrollo del programa se puede utilizar como guía el pseudocódigo de la figura N°2.

```
SUB Gauss (a, b, n, x, tol, er)
  DIMENSION s (n)
  er = 0
  DO i = 1, n
    s_i = ABS(a_{i,1})
    DO j = 2, n
      IF ABS(a_{i,j})>s_i THEN s_i = ABS(a_{i,j})
    END DO
  END DO
  CALL Eliminate(a, s, n, b, tol, er)
  IF er ≠ −1 THEN
    CALL Substitute(a, n, b, x)
  END IF
END Gauss

SUB Eliminate (a, s, n, b, tol, er)
  DO k = 1, n − 1
    CALL Pivot (a, b, s, n, k)
    IF ABS (a_{k,k}/s_k) < tol THEN
      er = −1
      EXIT DO
    END IF
    DO i = k + 1, n
      factor = a_{i,k}/a_{k,k}
      DO j = k + 1, n
        a_{i,j} = a_{i,j} − factor*a_{k,j}
      END DO
      b_i = b_i - factor * b_k
    END DO
  END DO
  IF ABS(a_{k,k}/s_k) < tol THEN er = −1
END Eliminate

SUB Pivot (a, b, s, n, k)
  p = k
  big = ABS(a_{k,k}/s_k)
  DO ii = k + 1, n
    dummy = ABS(a_{ii,k}/s_ii)
    IF dummy > big THEN
      big = dummy
      p = ii
    END IF
  END DO
  IF p ≠ k THEN
    DO jj = k, n
      dummy = a_{p,jj}
      a_{p,jj} = a_{k,jj}
      a_{k,jj} = dummy
    END DO
    dummy = b_p
    b_p = b_k
    b_k = dummy
    dummy = s_p
    s_p = s_k
    s_k = dummy
  END IF
END pivot

SUB Substitute (a, n, b, x)
  x_n = b_n/a_{n,n}
  DO i = n − 1, 1, −1
    sum = 0
    DO j = i + 1, n
      sum = sum + a_{i,j} * x_j
    END DO
    x_i = (b_i − sum) / a_{i,i}
  END DO
END Substitute
```

Figura N°2

6. Desarrolle un programa computacional en FORTRAN, amigable para el usuario, que permita aproximar la solución del sistema lineal resultante de la pregunta N°3, de forma iterativa, con un criterio de parada en el error relativo porcentual de 0.001, utilizando:

   a. el método de Gauss Seidel con relajación, utilizando el pseudocódigo de la figura N°3. Resuelva el sistema utilizando un valor de lambda:
      i. entre 0 y 1 (sub-relajación)
      ii. entre 1 y 2 (sobre-relajación simultánea ó SOR)
   b. el método de Jacobi
   c. realice una comparación entre ambos métodos

```
SUBROUTINE Gseid (a, b, n, x, imax, es, lambda)
 DO i = 1,n
   dummy = a_{i,i}
   DO j = 1,n
     a_{i,j} = a_{i,j}/dummy
   END DO
   b_i = b_i/dummy
 END DO
 DO i = 1, n
  sum = b_i
  DO j = 1, n
   IF i≠j THEN sum = sum - a_{i,j}*x_j
  END DO
  x_i=sum
 END DO
 iter=1
 DO
  centinela = 1
  DO i = 1,n
   old = x_i
   sum = b_i
   DO j = 1,n
    IF i≠j THEN sum = sum - a_{i,j}*x_j
   END DO
   x_i = lambda*sum +(1.-lambda)*old
   IF centinela = 1 AND x_i ≠ 0. THEN
      ea = ABS((x_i - old)/x_i)*100.
      IF ea > es THEN centinela = 0
   END IF
  END DO
  iter = iter + 1
  IF centinela = 1 OR (iter ≥ imax) EXIT
 END DO
END Gseid
```

Figura N°3

Nota importante:

Para comprobar la correcta implementación de los programas, se deberá enviar al ayudante del curso, en un mail, el(los) archivo(s) fuente (*.f90) y ejecutable desarrollados en la tarea.