Informe I:
Trabajo Práctico 2

Cipolatti Edgardo
edgardocipolatti@hotmail.com

I. EJERCICIO 9

Considere una red hidráulica como la representada en la figura 1, la cual es alimentada por un reservorio de agua que se encuentra a la presión constante $p_r = 10$bar. Para la j-ésima cañería, es válida la siguiente relación entre el caudal volumétrico Q_j , expresado en [m3/s], y la diferencia de presión Δp_j , dada en [bar], entre los extremos de la cañería

$$Q_j = kL\Delta p_j \tag{1}$$

donde k es la resistencia hidráulica en [m2/(bar s)] y L es la longitud del conducto expresada en [m]. El agua fluye hacia los extremos abiertos de los conductos que se encuentran a la presión atmosférica (indicados con puntos negros en la figura 1). Se pide determinar los valores de presión en cada uno de los nodos internos del sistema de cañerías (indicados como 1, 2, 3 y 4) si se tienen los siguientes datos del mismo:

Conducto	k	L
1	0.01	20
2	0.005	10
3	0.005	14
4	0.005	10
5	0.005	10
6	0.002	8
7	0.002	8
8	0.002	8
9	0.005	10
10	0.002	8

Nota 1: tenga en cuenta que los valores de presión se refieren a la diferencia entre la presión real y la presión atmosférica.

Nota 2: tenga en cuenta que la suma algebraica de los caudales en los conductos que se encuentran en el nodo-j es cero.

De acuerdo a la figura 1 planteo los flujos entrantes (positivos) y salientes (negativos) al punto sabiendo que la suma de los mismos debe ser cero.
Entonces en los 4 puntos tenemos:

$$\begin{aligned}
\text{punto 1:} \quad & Q_1 - (Q_2 + Q_3 + Q_4) = 0 \\
\text{punto 2:} \quad & Q_2 - (Q_9 + Q_1 0) = 0 \\
\text{punto 3:} \quad & Q_4 - (Q_5 + Q_6) = 0 \\
\text{punto 4:} \quad & Q_9 + Q_3 + Q_5 - (Q_8 + Q_7) = 0
\end{aligned}$$

Tomando en consideración la ecuación 1 sabemos que cada tramo de la cañería tiene un caudal volumétrico y podemos calcularlo.
Reemplazando en las ecuaciones de los puntos 1, 2, 3 y 4

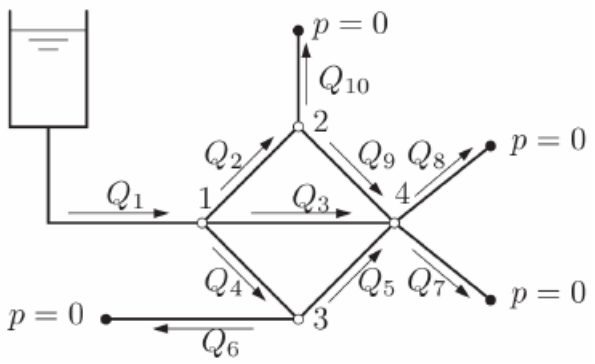

Fig. 1
CIRCUITO HIDRÁULICO EJERCICIO 9.

obtenemos un sistema en donde las variables p_1, p_2, p_3 y p_4 que son las que queremos averiguar (Estas corresponden a la presión en ese punto). Se muestra a continuación el reemplazo de cada caudal.

- Para el punto 1:
  $-2 + 0{,}2P_1 - (0{,}05P_2 - 0{,}05P_1 + 0{,}07P_4 - 0{,}07P_1 + 0{,}05P_3 - 0{,}05P_1) = 0$
  $\boxed{0{,}37P_1 - 0{,}05P_2 - 0{,}05P_3 - 0{,}07P_4 = 2}$
- Para el punto 2:
  $0{,}05P_2 - 0{,}05P_1 - (0{,}05P_4 - 0{,}05P_2 - 0{,}016P_2) = 0$
  $\boxed{-0{,}05P_1 + 0{,}116P_2 - 0{,}05P_4 = 0}$
- Para el punto 3:
  $0{,}05P_3 - 0{,}05P_1 - (0{,}05P_4 - 0{,}05P_3 - 0{,}016P_3) = 0$
  $\boxed{-0{,}05P_1 + 0{,}116P_3 - 0{,}05P_4 = 0}$
- Para el punto 4:
  $0{,}04P_4 - 0{,}05p_2 + 0{,}07P_4 - 0{,}07P_1 + 0{,}05P_4 - 0{,}05P_3 + 0{,}016P_4 + 0{,}016P_4) = 0$
  $\boxed{-0{,}07P_1 - 0{,}05P_2 - 0{,}05P_3 + 0{,}202P_4 = 0}$

Finalmente tenemos el sistema:

$$\begin{array}{cccccc}
0{,}37P_1 & -0{,}05P_2 & -0{,}05P_3 & -0{,}07P_4 & = & 2 \\
-0{,}05P_1 & 0{,}116P_2 & 0P_3 & -0{,}05P_4 & = & 0 \\
-0{,}05P_1 & 0P_2 & 0{,}116P_3 & -0{,}05P_4 & = & 0 \\
-0{,}07P_1 & -0{,}05P_2 & -0{,}05P_3 & 0{,}202P_4 & = & 0
\end{array}$$

Lo cual expresado como matrices para resolver el sistema Ax=b nos queda:

$$\begin{bmatrix} 0{,}37 & -0{,}05 & -0{,}05 & -0{,}07 \\ -0{,}05 & 0{,}116 & 0 & -0{,}05 \\ -0{,}05 & 0 & 0{,}116 & -0{,}05 \\ -0{,}07 & -0{,}05 & -0{,}05 & 0{,}202 \end{bmatrix} \times \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Con el sistema ya planteado paso a la resolución del problema con el código que se muestra a continuación. El mismo realiza la eliminación de gauss y nos devuelve, en caso de ser posible, un vector b con la solución a nuestro sistema Ax=b.

```
function [x] = gauss2(A,b)
   [m,n] = size(A);
   tol = 1e-9;
   indx = [1:n];

   for k = 1 : n-1
      aux = [k:n];

      [max_pivot,indxmax] =
          max(abs(A(indx(aux),k)));

      % control de pivot
      if(max_pivot < tol)
         disp('No se puede seguir con el
             algoritmo');
         return;
      end

      % hago intercambio
      if(indx(k) ~= indx(aux(indxmax)))
         m = indx(k);
         indx(k) = indx(aux(indxmax));
         indx(aux(indxmax)) = m;
      end

      % calculo multiplicadores
      A(indx(k+1:n),k) = A(indx(k+1:n),k) /
          A(indx(k),k);

      % eliminación
      for j = k+1:n
         A(indx(k+1:n),j) =
             A(indx(k+1:n),j) -
             A(indx(k+1:n),k) * A(indx(k),j);
      end

      b(indx(k+1:n)) = b(indx(k+1:n)) -
          A(indx(k+1:n),k) * b(indx(k));
   end

   x = backsub2(A,b,indx);
end
```

El algoritmo anterior usa la función backsub2 que es la encargada de hacer la sustitución hacia atrás. El código de esa función se detalla a continuación.

```
function [x] = backsub2(U,b,indx)
   [m,n] = size(U);
   U1 = zeros(n,n);
   b1 = zeros(n,1);

   for i = 1 : n
      U1(i,:) = U(indx(i),:);
      b1(i) = b(indx(i));
   end

   tol = 1e-9;
   x = zeros(n,1); %// vector columna de
       nx1 ceros

   if(abs(U1(n,n)) < tol) %si Unn es muy
       chico
      disp('No se puede continuar con el
          algoritmo');
      return;
   end

   x(n) = b1(n) / U1(n,n);

   for i = n-1: -1 : 1 % para i = n-1 hasta
       1 decrementando de a 1
      x(i) = (b1(i) - U1(i,i+1:n) *
          x(i+1:n)) / U1(i,i);
   end
end
```

Con el uso del algoritmo planteado llegamos a la resolución del ejercicio y obtenemos las presiones P_1, P_2, P_3 e P_4. Las cuales resultan ser:

$P_1 = 8{,}117249154$
$P_2 = 5{,}989289741$
$P_3 = 5{,}989289741$
$P_4 = 5{,}777903044$

II. EJERCICIO 10

Realice la factorización LU de la siguiente matriz siguiendo el orden de Doolittle, con y sin pivoteo parcial (con lo cual, si P es distinta de la identidad, en realidad se tiene PA = LU). Luego, calcule las matrices residuales A-LU y PA-LU y **justifique las diferencias que ocurren.**

$$\begin{bmatrix} 1 & 1+0{,}5e-15 & 3 \\ 2 & 2 & 20 \\ 3 & 6 & 4 \end{bmatrix}$$

Quiero hallar la factorización LU tal que A=LU, entonces utilizando el algoritmo de Doolittle sin pivoteo parcial obtengo las matrices L y U que muestro a continuación

$$U = \begin{bmatrix} 1 & 1 & 3 \\ 0 & -8{,}882e-16 & 14 \\ 0 & 0 & 4{,}729e+16 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & -3{,}378e+15 & 1 \end{bmatrix}$$

El código de Doolittle sin pivoteo ejecutado en Matlab es el siguiente.

```
function [L,U]=doolittle(A)
   %creo las matrices
   L = zeros(size(A));
   U = zeros(size(A));
   [m,n] = size(A);

   % Recorremos en orden de columnas
   for i=1:n
     for j=1:m
   % Estamos arriba de la diagonal buscamos
       U
      if i<=j
          U(i,j) = A(i,j);
          for k=1:i-1
            U(i,j) = U(i,j) - L(i,k)*U(k,j);
          end
      end
   % Estamos abajo de la diagonal buscamos L
      if j<=i
         L(i,j) = A(i,j);
```

```
      for k=1:j-1
        L(i,j) = L(i,j) - L(i,k)*U(k,j);
      end
      L(i,j) = L(i,j)/U(j,j);
    end
  end
end
```

El resultado de hacer A-LU es la matriz residual y se espera que tenga todas sus entradas en cero. Si no son todas cero como se muestra a continuación, significa que hay errores por utilizar una representación de dígitos finitos para los números y por la aritmética de dígitos finitos empleada en las operaciones, se generan acumulaciones de errores en los redondeos y truncamientos.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

Ahora utilizando Gauss con pivoteo parcial obtengo las siguientes matrices P (Permutación), L (Triangular inferior) y U (Triangular superior) que son las matrices de la factorización de Doolittle (PA=LU).

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0{,}6666667 & 1 & 0 \\ 0{,}3333333 & 0{,}5 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 3 & 6 & 4 \\ 0 & -2 & 17{,}333333 \\ 0 & 0 & -7 \end{bmatrix}$$

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Estas matrices fueron obtenidas por el siguiente algoritmo.

```
    function [P,L,U] = gauss2_lu(A)
  [m,n] = size(A);
  tol = 1e-9;
  indx = [1:n];
  for k = 1 : n-1
    aux = [k:n];
    [max_pivot,indxmax] =
        max(abs(A(indx(aux),k)));

    % control de pivot
    if(max_pivot < tol)
      disp('No se puede seguir con el
          algoritmo');
      return;
    end

    % hago intercambio
    if(indx(k) ~= indx(aux(indxmax)))
      m = indx(k);
      indx(k) = indx(aux(indxmax));
      indx(aux(indxmax)) = m;
    end

    % calculo multiplicadores
    A(indx(k+1:n),k) = A(indx(k+1:n),k) /
        A(indx(k),k);

    % eliminación
    for j = k+1:n
      A(indx(k+1:n),j) =
          A(indx(k+1:n),j) -
          A(indx(k+1:n),k) * A(indx(k),j);
    end
  end
  I = eye(n,n); % Crea la matriz identidad
       de nxn
  P = I(indx,:); % matriz de permutación
  [L,U] = lu_sep(P*A);
end
```

El algoritmo gauss2_lu utiliza en su interior otro algoritmo (lu_sep) que se encarga de separar una matriz cuadrada en L y U. Se muestra el algoritmo a continuación

```
function [L,U] = lu_sep(A)
  [~,n] = size(A);
  L=eye(n,n);
  U=zeros(n,n);

  if n == 1 then
    U = A;
  end

  for i=1 : n
    for j=i : n
      U(i,j) = A(i,j);
    end
  end
  for i = 2 : n
    for j = 1 : i-1
      L(i,j) = A(i,j);
    end
  end
end
```

Ahora por la igualdad PA=LU puedo obtener el error de cálculo haciendo PA-LU, ya que dispongo de todas las matrices. El resultado de la operación antes mencionada es:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -8 - 882D - 16 \end{bmatrix}$$

Como conclusión se deduce que el algoritmo de Gauss con pivoteo parcial es más conveniente en este caso porque el error es más pequeño. Ambos métodos deberían llegar al mismo resultado pero ya que se trabaja con pivotes muy cercanos a cero el error de cálculo se magnifica y es ahí donde se hace notar la diferencia entre un algoritmo y otro. Estos errores de cálculo se producen por falta de precisión en la representación en punto flotante de los números y la consecuente acumulación de errores de redondeo truncamiento y redondeo en las operaciones aritméticas.

REFERENCIAS

[1] R. L. Burden, *Análisis Numérico*, 7th ed. Thomson Learning, 2002.
[2] V. Sonzogni, "Cálculo numérico - apuntes de cátedra," 2015.