Algoritmos

Unidad 3: Solución de Sistemas de Ecuacioens Lineales

Taller de Métodos Numéricos

Año 2018

1 Sistemas con matriz triangular superior

Algoritmo 1 Función susreg (Sustitución regresiva para matrices triangulares superiores invertibles)

Entrada: A: matriz triangular superior invertible nxn; b: matriz nx1
Salida: x: solución del sistema lineal A x = b

```
n ← número de filas de A
x ← vector numérico de largo n
x[n] ← b[n] / A[n, n]
para k desde n-1 hasta 1 cada -1 hacer
   x[k] ← (b[k] - A[k, k+1 : n] * x[k+1 : n]) / A[k, k]
fin para
devolver x
```

2 Eliminación gaussiana sin pivoteo

Algoritmo 2 Función elimGauss (Eliminación Gaussiana para matrices invertibles)

Entrada: A: matriz invertible nxn; b: matriz nx1
Salida: x: solución del sistema lineal A x = b

```
n ← número de filas de A
Aum ← A || b  (Concatenación horizontal, matriz aumentada)

Los siguientes pasos triangularizan a la matriz A
para q desde 1 hasta n-1 hacer
   para r desde q+1 hasta n hacer
      mrq ← Aum[r, q] / Aum[q, q]
      Aum[r, q:(n+1)] ← Aum[r, q:(n+1)] - mrq * Aum[q, q:(n+1)]
   fin para
fin para

Una vez triangularizada la matriz, aplicar sustitución regresiva.
La última columna de Aum es la de términos independientes
A ← Aum[, 1:n]
b ← Aum[, n+1]
x ← susreg(A, b)
devolver x
```

3 Eliminación gaussiana con pivoteo trivial

Algoritmo 3 Función elimGauss_PivTriv (Eliminación Gaussiana para matrices invertibles con Pivoteo Trivial)

```
Entrada: A: matriz invertible nxn; b: matriz nx1
Salida: x: solución del sistema lineal A x = b
  n ← número de filas de A
  Aum ← A || b  (Concatenación horizontal, matriz aumentada)

  para q desde 1 hasta n-1 hacer

    Pivoteo trivial: Fijarse que el pivote no sea 0 y si lo es buscar otra fila en la que no sea 0 para intercambiar

    si Aum[q, q] = 0 entonces
      para r desde q + 1 hasta n hacer
        si Aum[r, q] ≠ 0 entonces
          temp ← Aum[q, ]
          Aum[q, ] ← Aum[r, ]
          Aum[r, ] ← temp
        fin si
      fin para
      Si después de buscar en todas las filas sigue siendo 0 es porque no había un pivote distinto de 0 y no
      se puede resolver
      si Aum[q, q] = 0 entonces
        imprimir "A es singular. No hay solución o no es única".
        devolver  (Finalizar sin devolver resultado)
      fin si
    fin si

    Realizar reemplazos para triangularizar a la matriz (igual que antes)
    para r desde q+1 hasta n hacer
      mrq ← Aum[r, q] / Aum[q, q]
      Aum[r, q:(n+1)] ← Aum[r, q:(n+1)] - mrq * Aum[q, q:(n+1)]
    fin para
  fin para

  Una vez triangularizada la matriz, aplicar sustitución regresiva.
  La última columna de Aum es la de términos independientes
  A ← Aum[, 1:n]
  b ← Aum[, n+1]
  x ← susreg(A, b)
  devolver x
```

4 Eliminación gaussiana con pivoteo parcial

Algoritmo 4 Función elimGauss_PivParc (Eliminación Gaussiana para matrices invertibles con Pivoteo Parcial)

Entrada: A: matriz invertible nxn; b: matriz nx1
Salida: x: solución del sistema lineal A x = b

```
n ← número de filas de A
Aum ← A || b  (Concatenación horizontal, matriz aumentada)

para q desde 1 hasta n-1 hacer

  Pivoteo parcial: intercambiar por la fila que tenga pivote con mayor valor absoluto
  Crear vector con los posibles pivotes en valor absoluto:
  candidatos ← abs(Aum[q:n, q])
  Calcular r, el número de fila que usaremos en el intercambio, r ≥ q:
  r ← q - 1 + (posición de max(candidatos))
  Intercambiar filas:
  temp ← Aum[q, ]
  Aum[q, ] ← Aum[r, ]
  Aum[r, ] ← temp;
  Si después del intercambio el pivote es 0, no se puede resolver:
  si Aum[q, q] = 0 entonces
    imprimir "A es singular. No hay solución o no es única".
    devolver  (Finalizar sin devolver resultado)
  fin si

  Realizar reemplazos para triangularizar a la matriz (igual que antes)
  para r desde q+1 hasta n hacer
    mrq ← Aum[r, q] / Aum[q, q]
    Aum[r, q:(n+1)] ← Aum[r, q:(n+1)] - mrq * Aum[q, q:(n+1)]
  fin para
fin para

Una vez triangularizada la matriz, aplicar sustitución regresiva.
La última columna de Aum es la de términos independientes
A ← Aum[, 1:n]
b ← Aum[, n+1]
x ← susreg(A, b)
devolver x
```

5 Eliminación gaussiana con pivoteo parcial escalado

Algoritmo 5 Función elimGauss_PivParcEsc (Eliminación Gaussiana para matrices invertibles con Pivoteo Parcial Escalado)

Entrada: A: matriz invertible nxn; b: matriz nx1
Salida: x: solución del sistema lineal A x = b

```
n ← número de filas de A
Aum ← A || b  (Concatenación horizontal, matriz aumentada)

para q desde 1 hasta n-1 hacer

  Pivoteo parcial escalonado: intercambiar por la fila que tenga pivote con mayor tamaño relativo a los
  elementos de su fila
  Crear vector columna con el maximo de cada fila en valor absoluto:
  sr ← vector columna con el máximo valor absoluto de cada fila de Aum[q:n, q:n]
  Posibles pivotes divididos por el maximo de su fila:
  sr2 ← abs(Aum[q:n, q]) / sr
  Fila que usaremos en el intercambio porque su pivote tiene mayor tamaño relativo:
  r ← loc(sr2 = max(sr2))[1] + q - 1
  Intercambiar filas:
  temp ← Aum[q, ]
  Aum[q, ] ← Aum[r, ]
  Aum[r, ] ← temp;
  Si después del intercambio el pivote es 0, no se puede resolver:
  si Aum[q, q] = 0 entonces
    imprimir "A es singular. No hay solución o no es única".
    devolver  (Finalizar sin devolver resultado)
  fin si

  Realizar reemplazos para triangularizar a la matriz (igual que antes)
  para r desde q+1 hasta n hacer
    mrq ← Aum[r, q] / Aum[q, q]
    Aum[r, q:(n+1)] ← Aum[r, q:(n+1)] - mrq * Aum[q, q:(n+1)]
  fin para
fin para

Una vez triangularizada la matriz, aplicar sustitución regresiva.
La última columna de Aum es la de términos independientes
A ← Aum[, 1:n]
b ← Aum[, n+1]
x ← susreg(A, b)
devolver x
```

6 Eliminación de Gauss-Jordan con pivoteo trivial

Algoritmo 6 Función gaussJordan (Eliminación de Gauss-Jordan para matrices invertibles con Pivoteo Trivial)

Entrada: A: matriz invertible nxn; b: matriz nx1
Salida: x: solución del sistema lineal A x = b

```
n ← número de filas de A
Aum ← A || b  (Concatenación horizontal, matriz aumentada)

para q desde 1 hasta n hacer

  Pivoteo trivial: Fijarse que el pivote no sea 0 y si lo es buscar otra fila en la que no sea 0 para intercambiar

  si Aum[q, q] = 0 entonces
    para r desde q + 1 hasta n hacer
      si Aum[r, q] ≠ 0 entonces
        temp ← Aum[q, ]
        Aum[q, ] ← Aum[r, ]
        Aum[r, ] ← temp
      fin si
    fin para
    Si después de buscar en todas las filas sigue siendo 0 es porque no había un pivote distinto de 0 y no
    se puede resolver
    si Aum[q, q] = 0 entonces
      imprimir  "A es singular. No hay solución o no es única".
      devolver  (Finalizar sin devolver resultado)
    fin si
  fin si

  Realizar reemplazos para llegar a la matriz identidad
  para r desde 1 hasta n hacer
    si r = q entonces
      Aum[q, q:(n+1)] ← Aum[q, q:(n+1)] / Aum[q, q]
    si no
      mrq ← Aum[r, q] / Aum[q, q]
      Aum[r, q:(n+1)] ← Aum[r, q:(n+1)] - mrq * Aum[q, q:(n+1)]
    fin si
  fin para
fin para

La última columna de Aum es la solución
x ← Aum[, n+1]
devolver  x
```