CS381: Numerical Computation & Softwares

Gaussian Elimination

Jakramate Bootkrajang

Department of Computer Science

Chiang Mai University

Outline

Implementing Gaussian Elimination

Approach

- ต้องการแปลง $Ax = b$ ให้อยู่ในรูป $A'x = b'$
- การ implement ไม่ได้ซับซ้อนมาก เนื่องจากมีสูตรในการคำนวณค่าต่างๆของสมาชิกของเมทริกซ์ A' ไว้อยู่แล้ว

Forward elimination

- ในการแปลง $Ax = b$ เป็น $A'x = b'$
- สำหรับ pivot row $\vec{a}_k$ และ a_{ij} (สมาชิกแถวที่ i หลักที่ j)

$$a'_{ij} = \begin{cases} a_{ij} - \frac{a_{ik}}{a_{kk}} a_{kj} & \text{for } k < i, j \leq n \\ 0 & \text{for } k < i \leq n, j = k \\ a_{ij} & \text{else} \end{cases}$$

$$b'_i = \begin{cases} b_i - \frac{a_{ik}}{a_{kk}} b_k & \text{for } k < i \leq n \\ b_i & \text{else} \end{cases}$$

Backward substitution procedure

$$
\begin{aligned}
x_n &= \frac{b_n}{a_{n,n}} \\
x_{n-1} &= \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}} \\
x_{n-2} &= \frac{b_{n-2} - a_{n-2,n}x_n - a_{n-2,n-1}x_{n-1}}{a_{n-2,n-2}} \\
&\vdots \\
x_i &= \frac{b_i - \sum_{j=i+1}^{n} a_{i,j}x_j}{a_{i,i}}
\end{aligned}
$$

Starting from the formula

Ap คือ A'

```
if k < i <=n && k < j <= n
    Ap[i,j] = A[i,j] - A[i,k]/A[k,k] * A[k,j]
elseif k < i <= n && j == k
    Ap[i,j] = 0
else
    Ap[i,j] = A[i,j]
end
```

ต้องทำทุกๆค่า i และ j

นำ for loop สองอันมาครอบ if .. else ไว้ ในที่นี้เราจะติดตัวแปร k (pivot row) เอาไว้ก่อน

```
for i=1:n
    for j=1:n
        if k < i <=n && k < j <= n
            ...
        end
    end
end
```

จากสมการเราสามารถคำนวณ b' ได้เช่นกัน

bp คือ b' แต่เนื่องจาก b' เป็นเวคเตอร์จึง loop แค่ i ก็พอ

```
for i=1:n
    if k < i <= n
        bp[i] = b[i] - A[i,k]/A[k,k] * b[k]
    else
        bp[i] = b[i]
    end
end
```

สุดท้ายเราจะ loop ทุกๆ pivot row

นำส่วนของโค้ดในการอัพเดท Ap กับ bp มาประกอบกัน

```
Ap = zeros(size(A))
bp = zeros(size(b))
for k=1:n # for each pivot row
    if ... # here is code for updating Ap
    if ... # followed by code for updating bp
    A = copy(Ap) # update A for the next loop
    b = copy(bp) # update b
end
return Ap, bp
```

อย่าลืมตั้งชื่อฟังก์ชันให้การคำนวณของเราด้วย

```
function ge(A, b)
    n = size(A,1)  # how many row do we have ?
    Ap = zeros(size(A))
    bp = zeros(size(b))
    for k=1:n # for each pivot row
        ....
    end
    return Ap, bp
end
```

Labwork

- ความจริงแล้วใน Julia ก็มี operator ที่ทำ partial pivot Gaussian elimination อยู่แล้ว คือ

```
x = A \ b
```

- เราจะลองเทียบความแม่นยำและความเร็วของการแก้ระบบสมการเชิงเส้นด้วยวิธีหา inverse กับ Gaussian elimination กัน

Labwork

- เริ่มจากการแก้สมการระบบเล็กๆ

```
A = rand(5,5)
A = 10*A
A = floor.(A)
z = ones(5,1) # correct answer
b = sum(A,2) # sum along columns
```

- เราจะสร้างระบบสมการที่มีคำตอบเป็น $[1 \;\; 1 \;\; 1 \;\; 1 \;\; 1]'$

Labwork

- เปรียบเทียบความเร็วในการคำนวณ

```
tic()
x1 = inv(A) * b
toc()


tic()
x2 = A \ b
toc()
```

Labwork

- เปรียบเทียบความแม่นยำในการคำนวณ

```
maximum(abs(z-x1))  # result of inverse()

maximum(abs(z-x2)) # result of Gaussian
```

Labwork

- ลองเปลี่ยนเป็นระบบที่ใหญ่ขึ้นแล้วทดลองซ้ำ

```
A = rand(5000,5000)
A = 10*A
A = floor.(A)
z = ones(5000,1) # correct answer
b = sum(A,2) # sum along columns
```

Homework

- เขียนส่วน Backward substitution ของ Gaussian elimination ให้สำเร็จ