

CS381: Numerical Computation & Softwares

Numerical Integration

Jakramate Bootkrajang

Department of Computer Science

Chiang Mai University

Implementing midpoint rule and trapezoid rule for integration

Midpoint rule

การประมาณพื้นที่ใต้กราฟโดยใช้วิธี midpoint rule เมื่อแบ่งส่วนของกราฟออกเป็น N ช่วงย่อย

$$\int_a^b f(x) dx \approx \sum_{i=1}^N f\left(\frac{x_{i+1} + x_i}{2}\right) (x_{i+1} - x_i)$$

หากกำหนดให้ทุกช่วงย่อยมีความกว้างเท่ากันหมด $h = (b - a)/n$ การหาพื้นที่ใต้กราฟสามารถเขียนได้เป็น

$$\int_a^b f(x) dx \approx \sum_{i=1}^N f\left(\frac{x_i + x_i}{2}\right) h$$

How to choose h ?

- ทางทฤษฎี h ควรมีค่าน้อยๆ เพื่อให้ค่า error ต่ำ แต่ในทางปฏิบัติเราไม่สามารถเลือก h ที่น้อยได้ตามตั้งใจ เนื่องจากอาจเกินขีดความสามารถของคอมพิวเตอร์
- ฉะนั้นเราอาจเปลี่ยนจากการกำหนดค่า h โดยตรง มาเป็นกำหนดค่า error ที่เรายอมรับได้ แล้วค่อยๆ ลดค่า h ไปเรื่อยๆ
- และหยุดลดค่า h เมื่อค่า error ต่ำกว่า ค่า error ที่เรายอมรับได้

The algorithm (1/2)

Algorithm 11.29. Suppose the function f , the interval $[a, b]$, the length n_0 of the initial partition, a positive tolerance $\epsilon < 1$, and the maximum number of iterations M are given. The following algorithm will compute a sequence of approximations to $\int_a^b f(x) dx$ by the midpoint rule, until the estimated relative error is smaller than ϵ , or the maximum number of computed approximations reach M . The final approximation is stored in I .

```
 $n := n_0; \quad h := (b - a) / n;$   
 $I := 0; \quad x := a + h / 2;$   
for  $k := 1, 2, \dots, n$   
     $I := I + f(x);$   
     $x := x + h;$   
 $j := 1;$   
 $I := h * I;$   
 $abserr := |I|;$   
while  $j < M$  and  $abserr > \epsilon * |I|$   
     $j := j + 1;$   
     $I_p := I;$ 
```

The algorithm (2/2)

```
n := 2n;  h := (b - a) / n;  
I := 0;  x := a + h / 2;  
for k := 1, 2, ..., n  
    I := I + f(x);  
    x := x + h;  
I := h * I;  
abserr := |I - Ip|;
```

Notations

ใน pseudocode สามารถแปลงเป็น syntax ของ Julia ได้ดังนี้

In pseudocode	ความหมาย	In Julia
<code>:=</code>	assignment	<code>=</code>
<code> x </code>	ค่าสัมบูรณ์ของ x	<code>abs(x)</code>
<code>ε</code>	อักษร epsilon	พิมพ์ <code>\epsilon</code> แล้วกด Tab

Our function

จาก algorithm เราสามารถออกแบบฟังก์ชันของเราให้รับพารามิเตอร์ ดังนี้

- 1 Function ที่ต้องการจะ integrate f
- 2 ค่า error ที่ยอมรับได้ ϵ
- 3 ขอบเขตบน b และขอบเขตล่าง a
- 4 จำนวนครั้งที่ algorithm สามารถวนซ้ำเพื่อลดระยะของช่วง M
- 5 จำนวนช่วงเริ่มต้น n

```
function integrate(f, a, b, n, epsilon, M)  
  
end
```


The initial length of interval h

- คำนวณความกว้างของช่วง จากจำนวนช่วงเริ่มต้น

```
function integrate(f, a, b, n, epsilon, M)
    h = (b-a)/n
end
```

Integration loop

- คำนวณพื้นที่ใต้กราฟ โดยหาผลบวกของพื้นที่ของทุกช่วง ให้ I แทนพื้นที่ใต้กราฟ

```
function integrate(f, a, b, n, epsilon, M)
    h = (b-a)/n
    I = 0    # initial area is zero
    x = a + h/2    # x is midpoint of 1st interval
    for i=1:n
        I = I + f(x)    # summing all sub-areas
        x = x + h    # change x to next midpoint
    end
end
```

Integration loop

- คำนวณ relative error (ส่วนต่างของค่าประมาณเก่ากับค่าประมาณใหม่)

```
function integrate(f, a, b, n, epsilon, M)
...
for i=1:n
    I = I + f(x)    # summing all sub-areas
    x = x + h      # change x to next midpoint
end

I = I * h    # because h is common factor
error = abs(I - 0)    # relative error

end
```

Halving the interval

```
function integrate(f, a, b, n, epsilon, M)
    ...
    j = 1 # keep track of allowed loops
    while j < M && error < epsilon * abs(I)
        n = 2 * n # increase number of intervals
        h = (b-a)/n # h is roughly halved
        Itmp = I # store previous area value
    end
end
```

end

Calculate integral based on new interval

```
function integrate(f, a, b, n, epsilon, M)
    while ...
        x = a + h/2
        for j=1:n    # same loop as above
            I = I + f(x)
            x = x + h
        end
    end
end
```

Updating error

```
function integrate(f, a, b, n, epsilon, M)
    while ...
        for j=1:n    # same loop as above
            ...
        end
        I = I * h
        error = abs(I - Itmp)
    end
    return I
end
```

Testing your function

```
f(x) = sin(x)
```

```
println(integrate(f,0,1,1,1e-8,10000))
```

```
0.4597257553496738
```

- Implement trapezoid rule (เปลี่ยนแค่สองบรรทัดของโค้ด midpoint rule)
- But which lines ???

Homework

- Implement Simpson's rule for numerical integration.
- Instruction will be uploaded shortly.