Bisection:   
 c = (a+b)/2; if(F(a) F(c) < 0){b = c;}else{a = c}

Error = O(2^^-n)

Ralph newson:   
 x\_n+1 = x\_n + f(x\_n)/f’(x\_n)

Lagrange:   
 f = f(x\_1) \* x-x\_2 / x\_1 - x\_2 \* x-x\_3 / x\_1 - x\_3

Least square fit:   
 c\_0 = Ey\_j \* E(x\_j^^2) - Ex\_j \* E(x\_j \* y\_j) / N \* E(x\_j^^2) - (Ex\_j)^^2   
 c\_1 = N \* E(x\_j \* y\_j) - Ey\_j \* Ex\_j / N \* E(x\_j^^2) - (Ex\_j)^^2

Rectangular rule:   
 I = E(n=0, n-1)h\*x\_i  
 Local Error = O(h^^2), E(n=0, n-1) O(h^^2) = n\*O(h^^2) = b-a/h \* O(h^^2) = O(h)

Trapezoid rule:   
 I ≅ E(i=0, n-1) (h \* [f(x\_i) + f(x\_i+1)] ) / 2 = h/2 \* [f(x\_0) + 2\*f(x\_1) + 2\*f(x\_2) .. f(x\_n)]

Local error = O(h^^2), global error = O(h^^3)

Simpson's rule:

I ≅ E(i=0,2... , n-2) h/3 [f(x\_i) + 4\*f(x\_i+1) + f(x\_i+2) ]  
 I ≅ h/3 \* [f(x\_0) + 4\*f(x\_1) + 2\*f(x\_2) + 4\*f(x\_3) + 2\*f(x\_4) + … 4\*f(x\_n-1) + f(x\_n]

Local error = O(h^^3), global error = O(h^^4)

Taylor series:   
 f(x+h) = f(x) + h\*f’(x) + h^^2/2! \* f’’(x) + h^^n/n! \* f^^n(x) + O(h^^n+1)

Forward/backward differencing:  
 f’(x) lim(h -> 0) = f(x+h) - f(x) / h, f’(x) lim(h -> 0) = f(x) - f(x-h) / h

f(x+h) = f(x) + h\*f’(x) + O(h^^2) -> f’(x) = f(x+h) - f(x) / h - O(h^^2) / h = O(h)

Central differencing:   
 f’(x) ≅ f(x+h) - f(x-h) / 2h + O(h^^2)

Double derivative:   
 f’’(x) ≅ f(x+h) - 2f(x) + f(x-h) / h^^2 + O(h^^2)

Euler method:   
 x(t\_0 + ∆t) = x(t\_0) + ∆t \* x’(t\_0) + O(∆t^^2)

x(t\_0 + ∆t) ≅ x(t\_0) + ∆t \* f(t\_0, x(t\_0)), Use iteratively with initial conditions in first.

global error ≅ N × local error = (T/∆t) ×O(∆t2) = O(∆t).

When x’(t) = sqrt(2 (E−V( x )) /M), f(t,x) =sqrt(2 (E−V( x )) /M)

2.nd order ODE:   
 x’’(t) = f(t, x(t), x’(t)), sol x’(t) = v(t).   
 x\_i+1 = x\_i + ∆t \* v\_i  
 v\_i+1 = v\_1 + f(t\_i, x\_i, v\_i)   
Improved euler:   
 xp\_i+1 = x\_i + ∆t \* f(t\_i, x\_i) - Gennemsnit af hældning i x\_i og predicted værdi

x\_i+1 = x\_i + 1/2 ∆t \* (f(t\_i, x\_i) + f(t\_i, xp\_i+1))

RK4:   
 x\_i+1 = x\_i + ∆t/6 \* (f\_0 + 2\*f\_1 + 2\*f\_2 + f\_3)

f\_0 = f(t\_i, x\_i)

f\_1 = f(t\_i \* ∆t/2, x\_i + ∆t/2 \* f\_0)

f\_2 = f(t\_i \* ∆t/2, x\_i + ∆t/2 \* f\_1)

f\_2 = f(t\_i \* ∆t, x\_i + ∆t \* f\_2)

RK4 has a local error = O(∆t5) and global error = O(∆t4).

Shooting method:   
 Problem: Find initial velocity or position at t\_0 that give a certain position or velocity at t\_1

Method: Use bisection method to find the v\_0 or x\_0 that gives the correct value of euler method at t\_1

Heat flow in 1D   
 u’’(x) = 0, u(0) = y, u(L) = z.   
 Central differencing with shooting method.   
 u’’(x) = u(x+∆x) + u(x-∆x) - 2\*u(x)/ ∆x^^2 + O(∆x^^2)

Gauss seidel:

u[i][j]=1./4.\*(u[i+1][j]+u[i-1][j]+u[i][j+1]+u[i][j-1]);