

پروژه اول درس محاسبات عددی پیشرفته
انواع روش‌های حل معادلات غیرخطی

نویسنده : امیرحسین فروزنده نژاد

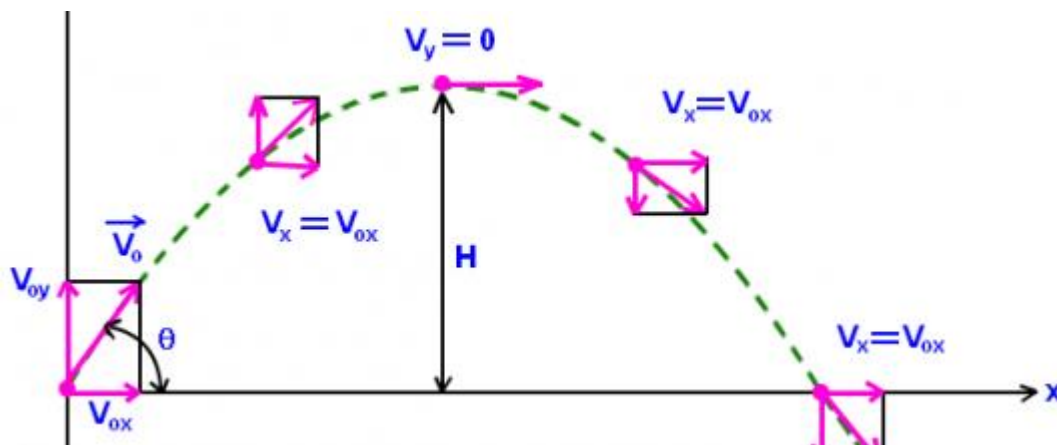
عنوان پروژه : انواع روش‌های حل معادلات غیرخطی

مقدمه

هدف از انجام این پروژه تفهیم بهتر مباحث تدریس شده در این فصل و همچنین مقایسه انواع روش‌های حل معادلات غیرخطی از دو منظر همگرایی و سرعت همگرایی و رسیدن به جواب است. که در ادامه به حل یک مسئله با جواب معلوم با 5 روش متفاوت خواهیم پرداخت.

معادله حاکم

در این پروژه معادله حرکت پرتابی به منظور پیدا کردن حداکثر برد پرتاب یک گوی با سرعت و زاویه معلوم از سطح زمین بررسی خواهد شد لازم به ذکر است که از اصطکاک هوا صرف نظر شده است.



شکل 1 : نمایی از یک حرکت پرتابی

$$y = \frac{-g \cdot x^2}{2 \cdot v_0^2 \cdot (\cos \theta)^2} + x \cdot \tan(\theta) \quad (\text{معادله 1})$$

که در معادله 1 :

V_0 : سرعت اولیه ($\frac{m}{s}$)

θ : زاویه پرتاب (deg)

g : شتاب جاذبه ($\frac{m}{s^2}$) برابر 9.81

که با فرض سرعت اولیه 10 متر بر ثانیه و زاویه پرتاب 45 درجه جواب مسئله 0 و 10.19367 متر خواهد شد .

روش‌های حل

1. Bisection
2. linear interpolation
3. Newton Raphson
4. $g(x)=x$
5. Muller

نتایج

i	Bisection	linear interpolation	Newton Raphson	$g(x)=x$	Muller
1	10.5	10.17923	10.19751	-10.815525	10.19368
2	10.25	10.19262	10.19368	-11.4753045	10.19368
3	10.125	10.1936	10.19368	-12.91806437	0
4	10.1875	10.19367	0	-16.37057357	0
5	10.21875	10.19368	0	-26.2903761	0
6	10.20313	0	0	-67.80513819	0
7	10.19531	0	0	-451.0183566	0
8	10.19141	0	0	-19955.26244	0
9	10.19336	0	0	-39064646.16	0
10	10.19434	0	0	-1.49705E+14	0
11	10.19385	0	0	-2.20E+27	0
12	10.1936	0	0	-4.74E+53	0
13	10.19373	0	0	-2.21E+106	0
14	10.19366	0	0	-4.77E+211	0
15	10.1937	0	0	inf	0
16	10.19368	0	0	inf	0
17	10.19369	0	0		0

همان طور که در شکل بالا قابل مشاهده است سرعت حل روش های استفاده شده به شرح زیر است :

Bisection < linear interpolation < Newton Raphson < Muller

دلیل واگرایی روش $g(x)=x$

شرط لازم برای همگرایی این روش به شرح زیر است :

$$|g(x)'| < 1$$

که برای این مسئله داریم :

$$g(x) = -0.0981 * x^2$$

$$g(x)' = -0.1962 * x$$

که این متغیر برای x های بزرگتر از 5.09 از بازه‌ی همگرایی خارج میشود پس همگرا نشدن جواب موضوعی همانند انتظار است.

کد

```
clc;clear;
syms f x;
f = -0.0981*x^2+x;
ff = -0.0981*x^2;

result = zeros(30,6);
for i =1:30
    result(i,1) = i;
end

a = 10; %primary guess
b = 11;

%% Bisection
aa = a;
bb = b;
cc = 0;
dd = 1;
i=0;

while abs(cc - dd)>0.00001
    i = i+1;
    dd = cc;
    f1 = subs(f,aa);
    f2 = subs(f,bb);
    cc = (aa+bb)/2;
    f3 = subs(f,cc);
    if f3*f1 < 0
        bb = cc;
    elseif f3*f1 > 0
        aa = cc;
    else
        break;
    end
    result(i,2) = cc;
end

%% linear interpolotion
aa = a;
bb = b;
cc = 0;
dd = 1;
i=0;

while abs(cc - dd)>0.00001
    i = i+1;
    dd = cc;
```

```
f1 = double(subs(f,aa));
f2 = double(subs(f,bb));
cc = aa - f1*(bb-aa)/(f2-f1);
f3 = subs(f,cc);
if f3*f1 < 0
    bb = cc;
elseif f3*f1 > 0
    aa = cc;
else
    break;
end
result(i,3) = cc;
end

%% Newton Raphson
aa = a;
bb = b;
i=0;
diff_f = diff(f);

while abs(bb - aa)>0.00001
    i = i+1;
    bb = aa;
    f1 = double(subs(f,aa));
    diff_f1 = double(subs(diff_f,aa));
    aa = aa - (f1/diff_f1);
    result(i,4) = aa;
    if i>30
        break
    end
end

%% modified newton
% aa = b;
% bb = b;
% cc = a;
% i=0;
%
% while abs(bb - cc)>0.01
%     i = i+1;
%     aa = bb;
%     bb = cc;
%     f1 = double(subs(f,aa));
%     f2 = double(subs(f,bb));
%     cc = bb - (bb-aa)/(f2-f1);
%     if i > 29
%         break;
%     end
%
```

```
%      result(i,5) = cc;
% end
% %javab cc ast
%

%% g(x)=x
x2 = (a+b)/2;
x1 = x2+1;
i = 0;
while abs(x2-x1)>0.01
    i= i+1;
    x1 = x2;
    x2 = double(subs(ff,x2));
    if i>30
        break
    end

    result(i,5) = x2;
end

%% Muller

x1 = max(a,b);
x2 = min(a,b);
x0 = (x1+x2)/2;
xr = x0 ;
s = 1;
i = 0;

while abs(s)>0.00001

    i=i+1;
    f0 = double(subs(f,x,x0));
    f1 = double(subs(f,x,x1));
    f2 = double(subs(f,x,x2));

    h1 = abs(x1-x0);
    h2 = abs(x2-x0);
    gama = h2/h1;

    aa = (gama*f1-f0*(1+gama)+f2)/(gama*h1^2*(1+gama));
    bb = (f1-f0-aa*h1^2)/h1;
    cc = f0;

    xr1 = (-2*cc)/(bb+(bb^2-4*aa*cc)^0.5);
    xr2 = (-2*cc)/(bb-(bb^2-4*aa*cc)^0.5);
    s = min(abs(xr1),abs(xr2));
    if abs(xr1)==s
        ss = sign(xr1);
```

```
else
    ss = sign(xr2);
end
xr = x0 + s*ss;

if xr>x0
    x2 = x0;
    x0 = xr;
else
    x0 = xr;
end
if i>30
    break;
end
result(i,6) = xr;

end
```