A New Technique to Compute Complex Roots

Sasan A. Muheadden
Department Of Mathematics
College of Science - University of Kirkuk

Received:1/2/2010, Accepted:1/6/2010

Abstract

In this paper we modified Newton – Raphson and Secant methods to Complex form, then from that we get new iterative formula to them, named Newton (Secant) real iteration formula and Newton (Secant) imaginary iteration formula, real iteration used to find real roots while imaginary iteration one will be used to find complex roots of non-linear equations such as ($x^2 + a = 0$). To find real roots, we start with a real initial point in real iteration getting a sequence of real numbers, to find complex roots start with a complex initial point in complex formula getting a sequence of complex numbers, also we write a new algorithm for this technique and write the program by using Matlab application system version 7.8 for this new method such that it will determine the real roots when we enter the initial real point but when we enter the initial complex point it will determine the complex roots.

Introduction

Complex analysis can roughly be though of as a subject that applies the theory of calculus to imaginary numbers. But what is exactly are imaginary numbers ? we can take the square root of a negative number, but let pretend we can and being by using the symbol $i = \sqrt{-1}$. Also numerical analysis is concerned with the mathematical derivation description and analysis of methods of obtaining numerical solutions of mathematical problems; such that find complex roots of linear and non – linear equations by using iteration methods. (Abbas & Sasan, 2009)
If R is the real field and C is the Complex field then consider the following definitions:
Newton – Raphson method can be used to approximate the root of any non-linear equation of any degree by the following iterative formula (Atkinson , 1998)

$$x_{k+1} = x_k - \frac{f(x_k)}{f^{\backslash}(x_k)} \qquad k = 0,1,2..$$

If the function $f(x)$ has no derivative then we have difference approximation in this case, Newton – Raphson method is called Secant method (Steven, 2008).

know we will show comparison between Newton and Secant method (Abbas&Sasan, 2008):

Newton – Raphson method	Secant method
1.Iterative formula $x_{k+1}=x_k-\frac{f(x_k)}{f^{\backslash}(x_k)}$ $k=0,1,2..$	1.Iterative formula $x_{k+1}=x_k-\left[f(x_k).\frac{x_k-x_{k-1}}{f(x_k)-f(x_{k-1})}\right]$, $k=1,2,3..$
2.Need for one initial point .	2. Need for two initial points.
3.Unstable .	3. Stable .
4. Need to derivative .	4. No need to derivative.
5.Faster .	5. Slow .
6. Quadratic speed .	6. Linear speed .

Complex iterative formula

1-Complex Newton – Raphson formula (Complex Iteration) :

In Newton formula Replace the variable x with the variable z in both side then we get the following form

$$z_{k+1}=z_k-\frac{f(z_k)}{f^{\backslash}(z_k)} \quad , \quad k=0,1,2,\ldots \qquad \ldots(1)$$

then we substitute $z=(x+i\,y)$ in the above form we get the formula

$$(x+i\,y)_{k+1}=(x+iy)_k-\frac{f(x+iy)_k}{f^{\backslash}(x+iy)_k} \qquad k=0,1,2\ldots \qquad \ldots(2)$$

$$x_{k+1}+iy_{k+1}=(x_k+iy_k)-\left[\frac{f(x_k+iy_k)}{f^{\backslash}(x_k+iy_k)}\right] \qquad k=0,1,2,\ldots \qquad \ldots(3)$$

then we get from that

$$x_{k+1}+iy_{k+1}=\left[x_k-\frac{f(x_k)}{f^{\backslash}(x_k)}\right]+i\left[y_k-\frac{f(y_k)}{f^{\backslash}(y_k)}\right] \qquad k=0,1,2,\ldots \qquad \ldots(4)$$

know equate the real and the imaginary parts in (4) we get two iterative formula named the real iterative formula(Newton– Raphson) (R.I.F)(5)and the imaginary iterative formula (I.I.F)(6):

$$x_{k+1}=x_k-\frac{f(x_k)}{f^{\backslash}(x_k)} \quad , \qquad k=0,1,2,\ldots. \qquad \ldots(5)$$

$$y_{k+1}=y_k-\frac{f(y_k)}{f^{\backslash}(y_k)} \quad , \qquad k=0,1,2,\ldots.. \qquad \ldots(6)$$

R.I.F and I.I.F were used to evaluate real and imaginary roots of non – linear equations respectively.

2- Complex Secant formula:

In secant method if we replace the variable x by the variable z then we get the complex secant formula which can separated into two formulas, the first called the real iterative formula starts with real initial point (x_0) the second called is the imaginary iterative formula starts with complex initial point (y_0), and where used to find real and imaginary roots of non – linear equations respectively.

Now

$$z_{k+1}=z_k-\left[f(z_k).\frac{z_k-z_{k-1}}{f(z_k)-f(z_{k-1})})\right] \qquad k=1,2,3,\ldots \qquad \ldots(7)$$

putting $z=(x+i\,y)$ in the above form we get the formula

$$(x+iy)_{k+1}=(x+iy)_k-\left[f(x_k+iy_k).\frac{(x+iy)_k-(x+iy)_{k-1}}{f(x_k+iy_k)-f(x_{k-1}+iy_{k-1})}\right] \quad ,k=1,2,3,.. \quad \ldots(8)$$

$$x_{k+1}+iy_{k+1}=x_k+iy_k-\left[f(x_k)+if(y_k).\left(\frac{(x_k+iy_k)-(x_{k-1}+iy_{k-1})}{f(x_k+iy_k)-f(x_{k-1}+iy_{k-1})}\right)\right],k=1,2,3,.. \quad \ldots(9)$$

$$x_{k+1}+iy_{k+1}=x_k-\left[f(x_k).\frac{x_k-x_{k-1}}{f(x_k)-f(x_{k-1})}\right]+i\left[y_k-\left(f(y_k).\left(\frac{y_k-y_{k-1}}{f(y_k)-f(y_{k-1})}\right)\right)\right], \qquad \ldots(10)$$

$k=1,2,3.....$

Now equating the real and the imaginary parts of both sides we get that

$$x_{k+1}=x_k-f(x_k).\left[\frac{x_k-x_{k-1}}{f(x_k)-f(x_{k-1})}\right] \quad \text{, k=1,2,3,…} \qquad \ldots(11)$$

Equation (11) called the real secant iterative starting with two real initial points .

$$y_{k+1} = y_k - f(y_k).\left[\frac{y_k - y_{k-1}}{f(y_k) - f(y_{k-1})}\right] \qquad , k = 1,2,3,\ldots \qquad \ldots(12)$$

Equation (12) called the imaginary secant iterative starting with two imaginary initial points .

3-A new modified Newton – Raphson algorithm :

For computing complex and real root of non – linear equations f(x)=0 depended on given initial point, we write the real iterative formula and the imaginary iterative formula of Newton – Raphson in the following procedure such that it is computing all roots of equations step by step and the number of iterations depended on the value of tolerance .

Now

Input informatios : The initial point z and tolerance $\in$

Output informations: The complex root w_1 or real root w_2

Procedure :

Step1 : If z is real , Go to step9 .

Step2 : Replace $y_0 = z$.

Step3 : Evaluate new value by

$$y_1 = y_0 - \frac{f(y_0)}{f^{\backslash}(y_0)}$$

Step4 : If $|y_1 - y_0| < \in$, go to step7 .

Step5 : Replace $y_0 = y_1$.

Step6 : Back to step3 .

Step7 : $w_1 = y_1$.

Step8 : Print the complex root w_1 .

Step9 : Replace $x_0 = z$.

Step10: Evaluate new value by

$$x_1 = x_0 - \frac{f(x_0)}{f^{\backslash}(x_0)}$$

Step11: If $|x_1 - x_0| < \in$, go to step14 .

Step12: Replace $x_0 = x_1$.

Step13: Back to step10 .

Step14: $w_2 = x_1$

Step15: Print the real root w_2 .

Step16: Stop.

4-A new modified Secant algorithm:

For compute the complex and the real roots of non – linear equations f(x)=0 where f(x) has no derivative we write the following procedure step by step depended on two initial points and tolerance.

Input informations: the initial points z_1, z_0 and tolerance $\in$.

Output informations: complex root w_1 or real root w_2 .

procedures :

Step1: If z_1, z_0 are real , Go to step9 .

Step2: Replace $y_0 = z_0$ & $y_1 = z_1$.

Step3: Evaluate new value by $y_2 = y_1 - f(y_1)\left[\dfrac{y_1 - y_0}{f(y_1) - f(y_0)}\right]$

Step4: If $|y_2 - y_1| <\in$, go to step7 .

Step5: Replace $y_0 = y_1$ & $y_1 = y_2$.

Step6: Back to step3 .

Step7: $w_1 = y_2$.

Step8: Print the complex root w_1 .

Step9: Replace $x_0 = Z_0$ & $x_1 = Z_1$.

Step10: Evaluate new value by

$$x_2 = x_1 - f(x_1)\left[\frac{x_1 - x_0}{f(x_1) - f(x_0)}\right]$$

Step11: If $|x_2 - x_1| <\in$, go to step14 .

Step12: Replace $x_0 = x_1$ & $x_1 = x_2$.

Step13: Back to step10 .

Step14: $w_2 = x_2$

Step15: Print the real root w_2 .

Step16: Stop.

5-The program by using Matlab application system version 7.8 for new modified Newton algorithm :

Before writing the program script , we must define a function and its derivative for the given non – linear equation and save them as a function m – file .Then we define these functions as funcnew01 and funcnew02 and save them as funcnew01.m and funcnew02 .m

Now :

```
Z=input('the complex value of Z=');
tol=input('the value tolerance  tol=');
k=0;
if real(Z)==0
y=Z;
```

```
disp('y      funcnew01(y)     funcnew02(y)      k')
disp('.......................................')
while abs(feval('funcnew01',y))>tol
y=y-(feval('funcnew01',y)/feval('funcnew02',y))
k=k+1;
end;
else
x=Z;
disp('x      funcnew01(x)     funcnew(x)      k')
disp('.....................................')
while abs(feval('funcnew01',x))>tol
    x=x-
(feval('funcnew01',x)/feval('funcnew02',x))
    k=k+1;
  end;
  end;
```

6-The program by using Matlab application system version 7.8 for new modified Secant algorithm :

```
Z1=input('the complex value Z1=');
Z2=input('the complex value Z2=');
tol=input('the value tolerance tol=');
k=0;
if real(Z1)==0 & real(Z2)==0
y1=Z1;
y2=Z2;
disp('y  f1  f2 k')
disp('..........')
while abs(y2-y1)>tol
f1=feval('funsct01',y1)
f2=feval('funsct01',y2)
y=y2-((f2)*((y2-y1)/(f2-f1)))
y1=y2;
y2=y;
k=k+1;
end;
else
x1=Z1;
x2=Z2;
disp('x f1  f2 k')
disp('.........')
```

```
 while abs(x2-x1)>tol
        f1=feval('funsct01',x1)
        f2=feval('funsct01',x2)
        x=x2-((f2)*((x2-x1)/(f2-f1)))
        x1=x2;
        x2=x;
        k=k+1;
    end;
end;
```

Numerical Computation to evaluate complex roots

Example 1 :
Find the root of the following non – linear equation by using the new modified Newton and Secant algorithm :

$f(x)=x^2+1$ where (tol = 0.00001) .

Solution :

1 – By the new modified Newton algorithm:
We know that the equation f(x)=0 has no real roots
if we use real Newton iterative formula $x_{k+1}=x_k-\frac{f(x_k)}{f^{\backslash}(x_k)}$ with real initial point (x_0) we get no solution .
If we use the new modified algorithm (imaginary iterative formula) $y_{k+1}=y_k-\frac{f(y_k)}{f^{\backslash}(y_k)}$ starting by complex initial point (y_o) then after some iteration we get a sequence of complex number $\{y_k\}_{k=0}^{\infty}$ converges to the complex root on the y – axis
Now start with the complex initial point $y_0=0.5i$,we get the following table

table No.1

K	y_k	$f(y_k)$	$f^{\backslash}(y_k)$
0	0+0.5i	0.7500	0+10000i
1	0+1.2500i	-0.5625	0+2.5000i
2	0+1.0250i	-0.0506	0+2.0500i
3	0+1.0003i	-6.0985e-004	0+2.0006i
4	0+1.0000i	-9.2922e-008	0+2.0012i

after 5 – iteration we will be end with complex root **y = 0 + 1.0000i**. If we start with the initial point $y_0=-0.5i$ we get another root **y = 0 – 1.0000i** also when the distance between the initial point and the exact root is small again if we start with $y_0 = 1.1i$ then by 2 – iteration we get the complex root .

2–By the new modified secant algorithm starting with initial point $y_0 = 0.5i$ in imaginary iterative $y_3 = y_2 - (f(y_2) \cdot \frac{(y_2 - y_1)}{f(y_2) - f(y_1)})$ we get

table No.2

k	y_k	y_{k+1}	$f(y_k)$	$f(y_{k+1})$
1	0.0000+0.5000i	0.0000+0.8000i	0.7500	0.3600
2	0.0000+0.8000i	0.0000+1.0769i	0.3600	-0.1598
3	0.0000+1.0769i	0.0000+0.9918i	-0.1598	0.0163
4	0.0000+0.9918i	0.0000+0.9997i	0.0163	6.0948e-004
5	0.0000+0.9997i	0.0000+1.0000i	6.0948e-004	-2.5089-006

then after 5 – iterations we have the complex root y = 0.0000+1.0000i

Example 2 :

Find all roots of the following non – linear equation by using the new modified Newton algorithm .

$f(x) = x^5 - x^4 + 7x^3 - 5x^2 + 4x - 4$ when (tol = 0.00001)

Solution:

we know that the above function has 5 – roots $x = \pm i$, $x = \pm 2i$ and $x = 1$ in this algorithm if we start by the real initial point $x_0 = 0.5$, we get a sequence of real numbers after some iterations using real Newton Iteration formula we get the real root $x = 1$.Now to find Complex roots let y_0 be any complex initial point we get a sequence of complex numbers tends to the complex root . As shown in table No.3 :

table No.3

K	y_k	$f(y_k)$	$f^{\backslash}(y_k)$
0	0.0000+0.5000i	-2.8125+1.4063i	0.5625-4.5000i
1	0.3846+1.0769i	-2.2112-2.2944i	-8.2107-3.6410i
2	0.0560+0.9432i	-0.6447+0.0009i	-5.3867-5.4629i
3	-0.0029+1.0031i	0.0362-0.0013i	-6.0372-6.0225i
4	-0.0000+1.0000i	0.0000	-6.0000-6.0000i

In table No.2 we used the new modified Newton algorithm starting with initial point $y_0 = 0.5i$ to get the complex root y = 1.0000i after 4 – iterations .If take $y_0 = 1.5i$ we will get another root y = 2.0000i again if we take $y_0 = -1.5i$ we will get the negative roots.

Conclusion

In this paper we modified Newton – Raphson and Secant algorithms to Compute the complex roots of non – linear equations, then we wrote down their programs by Matlab applications version 7.8 by taking complex initial point to get the complex roots if it exists if not we real initial point to get real root .In this paper we study initial point have only real or imaginary part , as future work we modified these algorithm to compute the complex roots if the initial point is complex and real and imaginary part not equal to zero .

References

- Abbas Y.Al-bayati and Runak M. Abdullah, (2008): A New Family Of Spectral Cg Algorithm, al–rafiden Journal, published by college of Computer Science and Mthematics Mosul university, Vol.5, No.1.
- Abbas Y.Al-bayati and Sasan A. Al-Shwani, (2008): A New Numerical Procedure To Compute the residue of a Complex Functions, al – rafiden Journal ,published by college of Computer Science and Mthematics Mosul university, Vol.5,No.1.
- Abbas Y. Al – Bayati and Sasan A. Al – Shwani, (2009): Residue of Complex Functions With Definite and Infinite Poles On X – axis ,Journal of Mathematics and Statistics, Science publication new York USA, Vol.5,No.3, pp.152 – 158. Web Site : www.scipub.org
- Atkinson, K. E. (1998): An Introduction To Numerical Analysis, Second Edition, John and wiley and Sons, Inc.
- Steven C Chapra, (2008): Applied Numerical Methods with Matlab For Engineering and science, 2nd edition tata McGraw–Hill Publishing Company.

تقنية جديدة لحساب الجذور العقدية

ساسان عادل محي الدين
قسم الرياضيات
كلية العلوم ــ جامعة كركوك

تاريخ الاستلام:١/٢/٢٠١٠، تاريخ القبول:١/٦/٢٠١٠

الخلاصة

في هذا البحث تم تطوير خوارزمية نيوتن – رافسون وخوارزمية القاطع وذالك بكتابة الصيغة العقدية لهما ومن خلال ذالك حصلنا على صيغتين جديدتين لهما سميا بصيغة نيوتن (القاطع) الحقيقية وصيغة نيوتن (القاطع) الخيالية واستخدمنا الصيغة التكرارية الحقيقية لإيجاد الجذور الحقيقية والصيغة التكرارية الخيالية لإيجاد الجذور الخيالية للمعادلات الغير الخطية وتم كتابة الخوارزمية المطورة في فقرتين (2.4 (2.3)) حيث أن الخوارزمية المطورة يعتمد على النقطة الابتدائية أذا كان حقيقية يستخدم الصيغة التكرارية الحقيقية ويحسب الجذور الحقيقية واذا كانت عقدية يستخدم الصيغة التكرارية الخيالية ويحسب الجذور العقدي وتم تطبيق الخوارزمية المطور في فقرتين (3.2 & 3.1). وتم برمجة الخوارزمية المطورة بأستخدام نظام ماتلاب 7.8 .