Ενδεικτικές Απαντήσεις $1^{\eta\varsigma}$ Εργασίας Εργαστηρίου MATLAB Γραμμικών Μαθηματικών

Δημήτριος Χριστόπουλος *

Απρίλιος 2011

Ερώτημα 1ο

Γνωρίζουμε ότι θα αντιμετωπίσουμε πρόβλημα όταν αναγκαστούμε να κάνουμε αφαίρεση δύο σχεδόν ίσων αριθμών, δηλ. όταν θα ισχύει:

$$u \approx v \Leftrightarrow \sqrt[3]{\sqrt{p^3+q^2}-q} \approx \sqrt[3]{\sqrt{p^3+q^2}+q} \tag{1}$$

Εξετάζουμε διάφορες δυνατές περιπτώσεις:

1. Εάν αντιμετωπίσουμε την ανωτέρω σχέση ως ισότητα είναι εύκολο να καταλήξουμε στην σχέση $q=0$, η οποία δεν ισχύει, διαφορετικά δεν θα είχε νόημα η δοθείσα εξίσωση ως τρίτου βαθμού, αφού θα λυνόταν άμεσα με παραγοντοποίηση. Επομένως εξετάζουμε πότε ισχύει η 1 ως έχει και όχι σαν ισότητα.

2. Μία άλλη περίπτωση είναι εκείνη κατά την οποία ισχύει $p^3 << q^2$. Τότε θα ισχύει: $\sqrt{p^3+q^2} \approx \sqrt{q^2} = |q|$, επομένως θα έχουμε ότι:

$$u = \begin{Bmatrix} \sqrt[3]{|q|-q} & = & 0 & , & q>0 \\ \sqrt[3]{|q|-q} & = & \sqrt[3]{2\,|q|} & , & q<0 \end{Bmatrix}$$

$$v = \begin{Bmatrix} \sqrt[3]{|q|+q} & = & \sqrt[3]{2q} & , & q>0 \\ \sqrt[3]{|q|+q} & = & 0 & , & q<0 \end{Bmatrix}$$

Συνεπώς ούτε σε αυτήν την περίπτωση μπορούμε να έχουμε αφαίρεση σχεδόν ίσων αριθμών, αφού πάντοτε ο ένας από τους δύο όρους θα είναι μηδενικός.

*dchristop@econ.uoa.gr

3. Μας μένει η περίπτωση κατά την οποία ισχύει:

$$p^3 >> q^2 \tag{2}$$
$$\frac{p^3}{q^2} >> 1 \tag{3}$$
$$\frac{q^2}{p^3} << 1 \tag{4}$$

Τότε θα έχουμε προσεγγιστικά ότι:

$$\begin{array}{ccc} u & = & \sqrt[3]{\sqrt{p^3\left(1+\frac{q^2}{p^3}\right)}-q} \\ & \underbrace{\approx}_{4} & \sqrt[3]{\sqrt{p^3}-q} \\ & = & \sqrt[3]{\sqrt{p^3}-sign(q)\,|q|} \\ & = & \sqrt[3]{|q|\left(\sqrt{\frac{p^3}{q^2}}-sign(q)\right)} \\ & \overbrace{\underbrace{\approx}_{sign(q)=\pm 1}}^{3} & \sqrt[3]{|q|\sqrt{\frac{p^3}{q^2}}} \\ & = & \sqrt[3]{|q|\,\frac{\sqrt{p^3}}{|q|}} \\ & = & \sqrt{p} \end{array} \tag{5}$$

Επίσης παρόμοια εργαζόμενοι έχουμε και για το v:

$$\begin{array}{ccc} v & = & \sqrt[3]{\sqrt{p^3\left(1+\frac{q^2}{p^3}\right)}+q} \\ & \underbrace{\approx}_{4} & \sqrt[3]{\sqrt{p^3}+q} \\ & \ddots & \ddots \\ & = & \sqrt{p} \end{array} \tag{6}$$

Από τις σχέσεις 5 και 6 είναι φανερό ότι στην περίπτωση όπου ισχύει $p^3 >> q^2$, τότε αφαιρούμε σχεδόν τους ίδιους αριθμούς, άρα έχουμε πρόβλημα απώλειας ακρίβειας όταν εργαζόμαστε με προγράμματα αριθμητικής κινητής υποδιαστολής όπως το MATLAB.

Ερώτημα 2ο

Χρησιμοποιούμε την ταυτότητα που μας δόθηκε και έχουμε:

$$u^3 - v^3 = (u - v)(u^2 + uv + v^2) \Leftrightarrow u - v = \frac{u^3 - v^3}{u^2 + uv + v^2}$$

Επομένως η ρίζα που ψάχνουμε υπολογίζεται τώρα με τον τύπο:

$$\begin{aligned}
\rho &= \frac{\left(\sqrt[3]{\sqrt{p^3+q^2}-q}\right)^3-\left(\sqrt[3]{\sqrt{p^3+q^2}+q}\right)^3}{\left(\sqrt{q^2+p^3}+q\right)^{\frac{2}{3}}+\left(\sqrt{q^2+p^3}-q\right)^{\frac{1}{3}}\left(\sqrt{q^2+p^3}+q\right)^{\frac{1}{3}}+\left(\sqrt{q^2+p^3}-q\right)^{\frac{2}{3}}} \\
&= -\frac{2\,q}{\left(\sqrt{p^3+q^2}+q\right)^{2/3}+\sqrt[3]{\left(\sqrt{p^3+q^2}-q\right)\left(q+\sqrt{p^3+q^2}\right)}+\left(\sqrt{p^3+q^2}-q\right)^{2/3}} \\
&= -\frac{2\,q}{\left(\sqrt{p^3+q^2}+q\right)^{2/3}+\sqrt[3]{\left(\sqrt{p^3+q^2}\right)^2-q^2}+\left(\sqrt{p^3+q^2}-q\right)^{2/3}} \qquad (7) \\
&= -\frac{2\,q}{u^2+\sqrt[3]{p^3+q^2-q^2}+v^2} \\
&= -\frac{2\,q}{u^2+\sqrt[3]{p^3}+v^2} \\
&= -\frac{2\,q}{u^2+p+v^2}
\end{aligned}$$

Βλέπουμε ότι τώρα δεν έχουμε αφαίρεση σχεδόν ίσων αριθμών, άρα προσδοκούμε βελτίωση της ακρίβειας.

Ερώτημα 3ο

Υπολογίζουμε, χρησιμοποιώντας το MATLAB, την ρίζα και με τους δύο τύπους. Για τον σκοπό αυτό δημιουργούμε μία συνάρτηση με όνομα *cardano.m* η οποία δέχεται σαν όρισμα τα p, q και αποδίδει την πραγματική ρίζα της εξίσωσης χρησιμοποιώντας και τους δύο τρόπους. Επίσης αποδίδει την τιμή που προκύπτει εάν αντικαταστήσουμε την κάθε ρίζα στην εξίσωση. Ο πλήρης κώδικας ακολουθεί:

```
function [r1,r2,t1,t2]=cardano(p,q)
%Computation of the real root of equation x^2+3*p*x+2*q =0
%with p^3+q^2>0 by two ways:
%r1=u-v
%r2=-2q/(u^2+u*v+v^2)
%and check accuracy by substitution.
format long e
while (p^3+q^2<=0)
    p=input('give new p: ');
    q=input('give new q: ');
end
disp('x^2+3*p*x+2*q =0,p^3+q^2>0')
```

```
u=(sqrt(p^3+q^2)-q)^(1/3);
v=(sqrt(p^3+q^2)+q)^(1/3);
r1=u-v;
r2=-2*q/(u^2+p+v^2);
t1=r1^3+3*p*r1+2*q;
t2=r2^3+3*p*r2+2*q;
p
q
disp('root with type: rho=u-v')
r1
disp('root with type: rho=-2q/(u^2+p+v^2)')
r2
disp('test r1 gives:')
t1
disp('test r2 gives:')
t2
end
```

Τώρα απλά πληκτρολογούμε δύο εντολές:

```
>> format compact
>> [r1,r2,t1,t2]=cardano(10^7,-3000000000000001/2000000000000);
x^2+3*p*x+2*q =0,p^3+q^2>0
p =
    10000000
q =
   -1.500000000000001e+003
root with type: rho=u-v
r1 =
    9.999999974752427e-005
root with type: rho=-2q/(u^2+p+v^2)
r2 =
    1.000000000000001e-004
test r1 gives:
t1 =
   -7.574271876364946e-006
test r2 gives:
t2 =
    1.818989403545857e-012
```

Παρατηρούμε βελτίωση της ακρίβειας υπολογισμού της ρίζας με τον δεύτερο τύπο. Το ίδιο μπορούμε να κάνουμε χρησιμοποιώντας το Octave, όπως φαίνεται από τα αποτελέσματα που ακολουθούν:

```
octave-3.2.4.exe:2>[r1,r2,t1,t2]=\
cardano(10^7,-3000000000000001/2000000000000);
x^2+3*p*x+2*q =0,p^3+q^2>0
p = 1.00000000000000e+007
q = -1.50000000000000e+003
root with type: rho=u-v
r1 = 9.99999997475243e-005
root with type: rho=-2q/(u^2+p+v^2)
r2 = 1.00000000000000e-004
test r1 gives:
t1 = -7.57427187636495e-006
test r2 gives:
t2 = 1.81898940354586e-012
```

Παρατηρούμε μικρές διαφορές στα αποτελέσματα ανάμεσα στα δύο προγράμματα με το Octave να επιτυγχάνει καλύτερη ακρίβεια, έστω και οριακά.

Ερώτημα 4ο

Στο wxMaxima[1] ορίζουμε την συνάρτηση:

$$f(x,p,q) = x^3 + 3px + 2q$$

κι έτσι μπορούμε να έχουμε την εξίσωση 3ου βαθμού με απλή τοποθέτηση των τιμών p, q που θέλουμε κάθε φορά. Λύνουμε την εξίσωση που μας δόθηκε και κρατάμε την πραγματική ρίζα. Σημειώνουμε ότι ο συμβολισμός $b-4$ σημαίνει απλά 10^{-4} για αριθμούς “big float” του wxMaxima.

(%i1)

```
f(x,p,q):=x^3+3*p*x+2*q;
sol:solve(f(x,10^7,-3000000000000001/2000000000000),x);
r:sol[3];
```

(%o1) $\mathrm{f}(x,p,q) := x^3 + 3\,p\,x + 2\,q$

(%o2) $[x = -\dfrac{\sqrt{12000000000000003}\,i+1}{20000}, x = \dfrac{\sqrt{12000000000000003}\,i-1}{20000},$
$x = \dfrac{1}{10000}]$

(%o3) $x = \dfrac{1}{10000}$

[1]Ελεύθερο λογισμικό, μπορείτε να το ‘κατεβάσετε’ από εδώ.

Εάν χρησιμοποιήσουμε το Axiom[2] έχουμε άμεσο διαχωρισμό της πραγματικής ρίζας από τις μιγαδικές ρίζες, διότι το Axiom μας δίνει στην έξοδο και το πολυώνυμο από το οποίο προκύπτουν οι μιφαδικές ρίζες, χωρίς να το λύνει:

```
→ fx:=x^3+3*10^7*x+2*-3000000000000001/2000000000000
```

(1) $$x^3 + 30000000\,x - \frac{3000000000000001}{1000000000000}$$

Type:Polynomial Fraction Integer

```
→ solve(fx,x)
```

(2) $$\left[x = \frac{1}{10000},\ \ 100000000\,x^2 + 10000\,x + 3000000000000001 = 0\right]$$

Type:List Equation Fraction Polynomial Integer

→

Εάν παρ΄ όλα αυτά εμείς θέλουμε να ταλαιπωρηθούμε κάνοντας αριθμητική κινητής υποδιαστολής, μπορούμε να το κάνουμε και αυτό με τα ανωτέρω προγράμματα.

Στο wxMaxima λύνουμε την γενική μορφή της εξίσωσης χωρίς να θέσουμε τις αριθμητικές τιμές των p, q που μας δόθηκαν:

(%i4)

```
sol:solve(f(x,p,q),x);r:sol[3];
ratsimp(%);
solf:subst([p=10^7,q=-3000000000000001/2000000000000],%);
fpprec : 16;bfloat(solf);
fpprec : 32;bfloat(solf);
fpprec : 64;bfloat(solf);
```

(%o4) $$[x = \left(-\frac{\sqrt{3}\,i}{2} - \frac{1}{2}\right)\left(\sqrt{q^2+p^3} - q\right)^{\frac{1}{3}} - \frac{\left(\frac{\sqrt{3}\,i}{2} - \frac{1}{2}\right)p}{\left(\sqrt{q^2+p^3} - q\right)^{\frac{1}{3}}},$$

$$x = \left(\frac{\sqrt{3}\,i}{2} - \frac{1}{2}\right)\left(\sqrt{q^2+p^3} - q\right)^{\frac{1}{3}} - \frac{\left(-\frac{\sqrt{3}\,i}{2} - \frac{1}{2}\right)p}{\left(\sqrt{q^2+p^3} - q\right)^{\frac{1}{3}}},$$

$$x = \left(\sqrt{q^2+p^3} - q\right)^{\frac{1}{3}} - \frac{p}{\left(\sqrt{q^2+p^3} - q\right)^{\frac{1}{3}}}]$$

[2]Ελεύθερο λογισμικό, μπορείτε να το ‘κατεβάσετε’ από εδώ ή εδώ, ανάλογα με το λειτουργικό που χρησιμοποιείτε.

(%o5) $x=\left(\sqrt{q^2+p^3}-q\right)^{\frac{1}{3}}-\frac{p}{\left(\sqrt{q^2+p^3}-q\right)^{\frac{1}{3}}}$

(%o6) $x=\frac{\left(\sqrt{q^2+p^3}-q\right)^{\frac{2}{3}}-p}{\left(\sqrt{q^2+p^3}-q\right)^{\frac{1}{3}}}$

(%o7) $x=\frac{\left(\frac{1000000000000001\sqrt{4000000000000001}}{2000000000000}+\frac{3000000000000001}{2000000000000}\right)^{\frac{2}{3}}-10000000}{\left(\frac{1000000000000001\sqrt{4000000000000001}}{2000000000000}+\frac{3000000000000001}{2000000000000}\right)^{\frac{1}{3}}}$

(%o8) 16
(%o9) $x=9.999999995550721b-5$
(%o10) 32
(%o11) $x=1.000000000000000000000000006335166b-4$
(%o12) 64
(%o13) $x=1.000000000000000000000000000000000000000000000000000$
$0000904052b-4$

Προσέξτε την αριθμητική οιασδήποτε ακρίβειας: αυξάνουμε την ακρίβεια από 16 σε 32 και μετά σε 64 δεκαδικά ψηφία και βλέπουμε πως βελτιώνεται η προσεγγιστική τιμή της ρίζας.

Στο Axiom πάλι μπορούμε να λύνουμε την εξίσωση με οιαδήποτε ακρίβεια ορίσουμε εμείς κάθε φορά με μία εντολή:

```
→ solve(fx,1.e-16)
```

$$[x=0.00010000000000000001674216] \tag{3}$$

```
Type:List Equation Polynomial Float
→ solve(fx,1.e-32)
```

$$[x=0.0001] \tag{4}$$

```
Type:List Equation Polynomial Float
→ solve(fx,1.e-64)
```

$$[x=0.0001] \tag{5}$$

```
Type:List Equation Polynomial Float
→
```

Ας δούμε τώρα τους δύο τύπους που μας δίνουν τις ρίζες για ακρίβεια 16 δεκαδικών ψηφίων:

wxMaxima

(%i14)

```
fpprec : 16;
rho[1](p,q):=(sqrt(p^3+q^2)-q)^(1/3)-(sqrt(p^3+q^2)+q)^(1/3);
rho[1](10^7,-3000000000000001/2000000000000)$
bfloat(%);
```

(%o14) 16

(%o15) $\rho_1(p,q) := \left(\sqrt{p^3+q^2}-q\right)^{\frac{1}{3}} - \left(\sqrt{p^3+q^2}+q\right)^{\frac{1}{3}}$

(%o17) $9.999999997489795b-5$

(%i18)

```
rho[2](p,q):=-2*q/((sqrt(p^3+q^2)-q)^(2/3)+p+(sqrt(p^3+q^2)+q)^(2/3));
rho[2](10^7,-3000000000000001/2000000000000)$
bfloat(%);
```

(%o18) $\rho_2(p,q) := \dfrac{-2\,q}{\left(\sqrt{p^3+q^2}-q\right)^{\frac{2}{3}} + p + \left(\sqrt{p^3+q^2}+q\right)^{\frac{2}{3}}}$

(%o20) $1.0b-4$

Axiom

```
→ r1(p,q)==(sqrt(p^3+q^2)-q)^(1/3)-(sqrt(p^3+q^2)+q)^(1/3)
Type:  VoidVoid
→ r2(p,q)==-2*q/((sqrt(p^3+q^2)-q)^(2/3)+p+(sqrt(p^3+q^2)+q)^(2/3))
Type:  VoidVoid
→

r22(p,q)==-2*q/((sqrt(p^3+q^2)-q)^(2/3)+((p^3+q^2)^(1/2)-q)^(1/3)*
((p^3+q^2)^(1/2)+q)^(1/3)+(sqrt(p^3+q^2)+q)^(2/3))
```

```
   Type:  VoidVoid
   → digits 16
```

(10) $$16$$

```
PositiveInteger
   → r1(10^7,-3000000000000001/2000000000000)::Float
   Compiling function r1 with type (PositiveInteger,Fraction Integer)
   AlgebraicNumber
```

(11) $$0.0001$$

```
Float
   → r2(10^7,-3000000000000001/2000000000000)::Float
   Compiling function r2 with type (PositiveInteger,Fraction Integer)
   AlgebraicNumber
```

(12) $$0.000099999999973662692$$

```
Float
   → r22(10^7,-3000000000000001/2000000000000)::Float
   Compiling function r22 with type (PositiveInteger,Fraction Integer)
   AlgebraicNumber
```

(13) $$0.0001$$

```
Float
```

Στο Axiom παρατηρούμε ότι υπάρχει καλύτερη ακρίβεια όταν δεν κάνουμε απλοποίηση στον παρονομαστή του δευτέρου τύπου. Είναι αναμενόμενη αυτή η διαφοροποίηση γιατί το Axiom δεν είναι πρόγραμμα αριθμητικής κινητής υποδιαστολής, αλλά είναι Σύστημα Υπολογιστικής Άλγεβρας. Βέβαια αυτό του δίνει την δυνατότητα να κάνει πράξεις με 500 δεκαδικά ψηφία:

```
→ digits 500
```

(15) $$500$$

```
PositiveInteger
   → r1(10^7,-3000000000000001/2000000000000)::Float
```

(16) $$0.0001$$

```
Float
```

→ `r2(10^7,-3000000000000001/2000000000000)::Float`

(17)
$$0.0000999999999999999999999999999999999999999999999999999\\
99999999999999999999999999999999999999999999999999999999\\
99999999999999999999999999999999999999999999999999999999\\
99999999999999999999999999999999999999999999999999999999\\
99999999999999999999999999999999999999999999999999999999\\
99999999999999999999999999999999999999999999999999999999\\
99999999999999999999999999999999999999999999999999999999\\
99999999999999999999999999999999999999999999999999999999\\
99999999999999999999999984988$$

```
Float
```

→ `r22(10^7,-3000000000000001/2000000000000)::Float`

$$0.0001 \tag{18}$$

```
Float
```

→

Το ίδιο μπορούμε να κάνουμε και με το wxMaxima:

(%i14)

```
fpprec : 500;
rho[1](p,q):=(sqrt(p^3+q^2)-q)^(1/3)-(sqrt(p^3+q^2)+q)^(1/3);
rho[1](10^7,-3000000000000001/2000000000000)$
bfloat(%);
```

(%o14) 500

(%o15) $\rho_1\left(p,q\right):=\left(\sqrt{p^3+q^2}-q\right)^{\frac{1}{3}}-\left(\sqrt{p^3+q^2}+q\right)^{\frac{1}{3}}$

(%o17) $1.000000000000000000000000000000[444 digits]0000000000$
$00000000000674388b-4$

(%i18)

```
fpprec : 500;
rho[2](p,q):=-2*q/((sqrt(p^3+q^2)-q)^(2/3)+p+(sqrt(p^3+q^2)+q)^(2/3));
rho[2](10^7,-3000000000000001/2000000000000)$
bfloat(%);
```

(%o18) 500

(%o19) $\rho_2\,(p,q) := \dfrac{-2\,q}{\left(\sqrt{p^3+q^2}-q\right)^{\frac{2}{3}}+p+\left(\sqrt{p^3+q^2}+q\right)^{\frac{2}{3}}}$

(%o21) $1.0b-4$

Παρατηρήστε ότι στο wxMaxima η ακρίβεια του πρώτου τύπου βελτιώνεται μεν, αλλά δεν εξαλείφεται. Αντιθέτως με την χρήση του δεύτερου τύπου έχουμε απόλυτη ακρίβεια.

Επομένως ακόμα και με τα CAS χρειάζεται να έχουμε γνώσεις *Αριθμητικής Ανάλυσης* για να επιτυγχάνουμε απόλυτα ακριβή αποτελέσματα, όταν αποφασίσουμε να χρησιμοποιήσουμε αυτά τα συστήματα για πράξεις αριθμών κινητής υποδιαστολής.