Ristrutturo un po' l'algoritmo del solutore robustissimo (spero) di terzo grado per rendere più facile la sua traduzione in Javascript ed inoltre traduco in Mathematichese le function Javascript che avevo già programmato anni fa...

In questo modo faccio una verifica preliminare del funzionamento del solutore di quarto grado che sfrutta il solutore robustissimo di terzo grado...

http://www.elegio.it/calcolatrice/trovaradici3-4.html

Solutore equazione 3° grado convertibile in Javascript

Versione più facilmente traducibile in Javascript...

In[1]:=

```
solve3[poli_] := Block[{a, p, q, A, B, C, D, Ca, s = poli[[3]] / 3, sina, cosa, expa,
     segno, sqrt3 = Sqrt[3]}, q = poli[[1]] - s (poli[[2]]  - s (poli[[3]]  - s));
    p = poli[[2]] - s (2 poli[[3]]  - 3 s);
    If[p == 0, a = Sign[q] Abs[ q]^(1 / 3) - s; Return[{a, a, a}]];
    If[p > 0, B = Sqrt[4 p / 3]; D = - q Sqrt[27 / (4 p^3)];
     a = (Log[D + Sqrt[D^2 + 1]]) / 3;
     expa = Exp[a];
     sina = B (expa - 1 / expa) / 2;
     cosa = B (expa + 1 / expa) / 2;
     Return[
      {sina - s, (-sina +  ⅈ sqrt3 cosa) / 2 - s,  (-sina -  ⅈ sqrt3 cosa) / 2 - s}]];
    A = Sqrt[-4 p / 3];
    C = q Sqrt[-27 / (4 p^3)]; Ca = Abs[C];
    If[Ca > 1, segno = C / Ca;
     a = (Log[Ca + Sqrt[Ca^2 - 1]]) / 3;
     expa = Exp[a];
     sina = segno A (expa - 1 / expa) / 2;
     cosa = segno  A (expa + 1 / expa) / 2;
     Return[{- cosa - s,
        (cosa  - ⅈ sqrt3 sina ) / 2 - s,
        (cosa  + ⅈ sqrt3 sina ) / 2 - s} ]];
    a = ArcSin[C] / 3;
    sina = A Sin[a];
    cosa = A Cos[a];
    Return[ {sina - s, (- sina + sqrt3  cosa) / 2 - s, (- sina - sqrt3  cosa) / 2  - s}]];
```

In[2]:=

```
For[mieitest = 1, mieitest > 0, mieitest--,
  Print["=====\nConto alla rovescia: ", mieitest];
  unpo =
   {1, RandomInteger[{-999, 999}] / 500, RandomInteger[{-999, 999}] / 500, 1};
  Print["Un polinomio : ", unpo];
  radici = solve3[unpo];
  (* Print["Radici: ",radici]; *)
  (* Print["Radici in forma numerica \n",N[radici,30]]; *)
  forsezero = N[Dot[unpo, Table[x^(j - 1), {j, 4}]] /. x → N[ radici, 50], 50];
  Print["Dovrebbe essere un vettore di zeri : ", forsezero]];
```

=====
Conto alla rovescia: 1

Un polinomio : $\left\{1, \frac{1}{50}, \frac{859}{500}, 1\right\}$

Dovrebbe essere un vettore di zeri :
$\{0.\times 10^{-49}, 0.\times 10^{-50} + 0.\times 10^{-50}\, i, 0.\times 10^{-50} + 0.\times 10^{-50}\, i\}$

In Javascript lo programmo così:

```
function cubicabis(poli){
// radici di polinomio di terzo grado con
// coefficienti reali e calcolati con
// algoritmo basato su funzioni trigonometriche
// ed esponenziali
var a,p,q,s,am,bm,cm, dm, ca,sina,cosa,expa,segno,sqrt3;
sqrt3=Math.sqrt(3);
if(3>poli.length) return[ [0,1],[0,1],[0,1] ];
// Lo faccio uscire se il polinomio e' mal fatto.
s=poli[2][0]/3;
q=poli[0][0]-s*(poli[1][0]-s*(poli[2][0]-s));
p=poli[1][0]-s*(2*poli[2][0]-3*s);
if(0>=p*p){
// cubo perfetto, tre radici coincidenti
if(q>0) segno=1;
else if(0>q) segno=-1;
else segno=0;
a=segno*Math.pow(Math.abs(q),1/3)-s;
return [[a,0],[a,0],[a,0]];};
//
if(p>0){
bm=Math.sqrt(4*p/3);
dm=-q*Math.sqrt(27/(4*p))/p;
a=Math.log(dm+Math.sqrt(dm*dm+1))/3;
expa=Math.exp(a);
sina=bm*(expa-1/expa)/2;
cosa=bm*(expa+1/expa)/2;
return [[sina-s,0],
[-sina/2-s,sqrt3*cosa/2],
[-sina/2-s,-sqrt3*cosa/2]];};
//
am=Math.sqrt(-4*p/3);
cm=-q*Math.sqrt(-27/(4*p))/p;
ca=Math.abs(cm);
```

```
if(ca>1){
if(cm>0) segno=1;
else if(0>cm) segno=-1;
else segno=0;
a=Math.log(ca+Math.sqrt(ca*ca-1))/3;
expa=Math.exp(a);
sina=segno*am*(expa-1/expa)/2;
cosa=segno*am*(expa+1/expa)/2;
return [[-cosa-s,0],
[cosa/2-s,-sqrt3*sina/2],
[cosa/2-s,sqrt3*sina/2]];};
// tre radici reali
a=Math.asin(cm)/3;
sina=am*Math.sin(a);
cosa=am*Math.cos(a);
return [[sina-s,0],
[-sina/2+sqrt3*cosa/2-s,0],
[-sina/2-sqrt3*cosa/2-s,0]];
}
```

Solutore di 2° grado tradotto da quello scritto in Javascript

LO SO CHE ESISTE LA SOLVE !!! Ma qui ho bisogno di controllare se la versione programmat in Javascript funziona bene...

In[3]:=

```
solve2[r_] := Block[{a = r[[2]], b = r[[1]], de, p, q, s, t, x},
   If[0 ≥ Abs[b], Return[{0, -a}]];
   de = a^2 - 4 b ;
   p = Re[de];
   q = Im[de];
   If[p > 0, s = Sqrt[(Sqrt[p^2 + q^2] + p) / 2]; t = q / (2 s),
    t = Sqrt[(Sqrt[p^2 + q^2] - p) / 2];
    If[t == 0, Return[{-a / 2, -a / 2}]];
    s = q / (2 t)];
   If[s Re[a] + t Im[a] > 0, x = - (a + s + I t) / 2,
    x = - (a - s - I t) / 2];
   Return[{x, b / x}]
  ];
poldue[p_, x_] := p[[1]] + x (p[[2]] + x);
```

In[5]:=

```
For[mieitest = 4, mieitest > 0, mieitest--,
 Print["=====\nConto alla rovescia: ", mieitest];
 pp = {RandomInteger[{-999, 999}] / 500, RandomInteger[{-999, 999}] / 500, 1};
 Print["Polinomio di secondo grado : ", pp];
 xpp = FullSimplify[solve2[pp]];
 Print["Radici : ", xpp];
 Print["Controllo\n", FullSimplify[poldue[pp, xpp[[1]]]], "\n",
  FullSimplify[poldue[pp, xpp[[2]]]]]
]
```

=====
Conto alla rovescia: 4

Polinomio di secondo grado : $\left\{\frac{37}{125}, \frac{41}{125}, 1\right\}$

Radici : $\left\{\frac{1}{250}\left(-41+11\,i\sqrt{139}\right), \frac{1}{250}\left(-41-11\,i\sqrt{139}\right)\right\}$

Controllo
0
0

=====
Conto alla rovescia: 3

Polinomio di secondo grado : $\left\{\frac{787}{500}, \frac{431}{500}, 1\right\}$

Radici : $\left\{\frac{i\left(431\,i+\sqrt{1\,388\,239}\right)}{1000}, -\frac{i\left(-431\,i+\sqrt{1\,388\,239}\right)}{1000}\right\}$

Controllo
0
0

=====
Conto alla rovescia: 2

Polinomio di secondo grado : $\left\{\frac{323}{500}, \frac{469}{250}, 1\right\}$

Radici : $\left\{\frac{1}{500}\left(-469-\sqrt{58\,461}\right), \frac{1}{500}\left(-469+\sqrt{58\,461}\right)\right\}$

Controllo
0
0

=====
Conto alla rovescia: 1

Polinomio di secondo grado : $\left\{-\frac{4}{5}, \frac{67}{100}, 1\right\}$

Radici : $\left\{\frac{1}{200}\left(-67-\sqrt{36\,489}\right), \frac{1}{200}\left(-67+\sqrt{36\,489}\right)\right\}$

Controllo
0
0

In Javascript lo programmo così:

```
function secondogrado(r){
    //
```

```
// Radici equaz. di secondo grado a coeff. complessi:
//   (y + r[1] )*y + r[0] = 0
//
// r[0][0] rappresenta la parte reale del termine noto.
// r[0][1] è la parte immaginaria del termine noto.
// r[1][0] è la parte reale del termine di primo grado.
// r[1][1] è la parte immaginaria del termine di primo grado.
// Il termine di secondo grado è sottinteso e vale 1.
//
var p,q,s,t,x,a=r[1],b=r[0];
if(0>=b[0]*b[0]+b[1]*b[1]) return [ [0,0],[-a[0],-a[1]] ];
p=a[0]*a[0]-a[1]*a[1]-4*b[0];
q=2*a[0]*a[1]-4*b[1];
with(Math) {
    if(p>0){
        s=sqrt((sqrt(p*p+q*q)+p)*0.5);
        t=0.5*q/s;
        }
    else {
        t=sqrt((sqrt(p*p+q*q)-p)*0.5);
        // radici coincidenti
        if(0 >= t) return [[-a[0]/2,-a[1]/2],[-a[0]/2,-a[1]/2]];
        s=0.5*q/t;
        }
    // Fa direttamente la radice di maggior modulo
    if(a[0]*s+a[1]*t>0) x=[-0.5*(a[0]+s), -0.5*(a[1]+t)];
    else    x=[-0.5*(a[0]-s), -0.5*(a[1]-t)];
    // Deduce l'altra dal termine noto.
    p=1/( x[0]*x[0] + x[1]*x[1] );
    }
return [ x, [p*(b[0]*x[0]+b[1]*x[1]), p*(-b[0]*x[1]+b[1]*x[0])] ];
}
```

Soluzione dell'equazione di quarto grado

Ripeto: So che esiste la Solve ! ma per controllare la programmazione in Javascript...

Soluzione equazione di quarto grado che si basa su quella di secondo e di terzo.

In[6]:=

```
solve4[p_] := Block[{a = p[[4]], b = p[[3]],
    c = p[[2]], d = p[[1]], u, s, t, v, w, z, e, f, g, h, m, n},
   u = solve3[{4 d b - c^2 - d a^2, c a - 4 d, -b, 1 }];
  z = u[[1]];
  s = a^2 / 4 + z - b;
  t = z^2 / 4 - d;
  v = Abs[s]; w = Abs[t];
  If[w > v, w = Sqrt[w]; v = (a z - 2 c) / (4 w),
   If[v > w, t = s; v = Sqrt[v]; w = (a z - 2 c) / (4 v)]];
  If[t ≥ 0 , e = z / 2 - w; f = z / 2 + w;
   g = a / 2 - v; h = a / 2 + v,
   e = z / 2 + ⅈ w;
   f = z / 2 - ⅈ w;
   g = a / 2 - ⅈ v;
   h = a / 2 + ⅈ v];
  m = solve2[{e, g}];
  n = solve2[{f, h}];
  Return[{n[[1]], n[[2]], m[[1]], m[[2]]}]
 ]
```

In Javascript l'ho programmata così...

```
function tetra(p){
    // radici di polinomio di quarto grado con
    // coefficienti reali.
    var a,b,c,d,q,u,s,t,v,w,e,f,g,h,m,n;
    a=p[3][0];  b=p[2][0];
    c=p[1][0];  d=p[0][0];
    q=[[4*d*b-c*c-d*a*a,0],[c*a-4*d,0],[-b,0]];
    u=cubica(q);
    s=a*a/4+u[0][0]-b;
    t=u[0][0]*u[0][0]/4-d;
    with(Math){
        v=abs(s); w=abs(t);
        if(w>v) {
            w=sqrt(w);
            v=(a*u[0][0]-2*c)/(4*w);
            }
        else if(v>w) {
            t=s;
            v=sqrt(v);
            w=(a*u[0][0]-2*c)/(4*v);
            }
        }
    if(t >= 0) {
        e=[u[0][0]/2-w,0];
        f=[u[0][0]/2+w,0];
        g=[a/2-v,0];
        h=[a/2+v,0];
        }
    else {
        e=[u[0][0]/2,w];
        f=[u[0][0]/2,-w];
        g=[a/2,-v];
```

```
        h=[a/2,v];
        }
    m=secondogrado([e,g]);
    n=secondogrado([f,h]);
    return  [n[0],n[1],m[0],m[1]];
    }
```

Qualche test fisso che **evidenzia problemi interni di *Mathematica*** ma comunque la soluzione risulta giusta.....

In[7]:=

```
unpol = {0, 3, 6, 3, 1};
Print["Un polinomio con una radice nulla : ", unpol];
radici = solve4[unpol];
Print["Trovo queste radici : ", N[radici]]
forsezero = N[unpol.Table[x^(j - 1), {j, 5}] /.
   x -> N[radici, 50], 50];
Print["Controllo che le radici diano tutte 0 "];
Print["Dovrebbe essere un vettore di zeri : ",
  forsezero]
```

Un polinomio con una radice nulla : {0, 3, 6, 3, 1}

N::meprec : Internal precision limit $MaxExtraPrecision = 50.` reached while evaluating

$$\frac{1}{2}\left(-2-\frac{1}{\left(\frac{7}{2}+\frac{3}{2}\,\text{Power}[\ll 2\gg]\right)^{1/3}}-\left(\frac{7}{2}+\frac{3}{2}\,\text{Power}[\ll 2\gg]\right)^{1/3}\right)+\frac{1}{2}\left(2+\frac{1}{\left(\frac{7}{2}+\frac{3\sqrt{5}}{2}\right)^{1/3}}+\left(\frac{7}{2}+\frac{3\sqrt{5}}{2}\right)^{1/3}\right).\gg$$

N::meprec : Internal precision limit $MaxExtraPrecision = 50.` reached while evaluating

$$-\text{Abs}\left[\frac{1}{2}\left(-2-\frac{1}{\text{Plus}[\ll 2\gg]^{1/3}}-\text{Plus}[\ll 2\gg]^{1/3}\right)+\frac{1}{2}\left(2+\frac{1}{\left(\frac{7}{2}+\text{Times}[\ll 2\gg]\right)^{1/3}}+\left(\frac{7}{2}+\text{Times}[\ll 2\gg]\right)^{1/3}\right)\right].\gg$$

Trovo queste radici : {-1.16109 + 1.75438 i, -1.16109 - 1.75438 i, -0.677815, 0.}

N::meprec : Internal precision limit $MaxExtraPrecision = 50.` reached while evaluating

$$\left\{\frac{1}{2}\left(-\frac{3}{2}-\frac{-6+3\,(2+\text{Power}[\ll 2\gg]+\text{Power}[\ll 2\gg])}{2\left(2+\frac{1}{\text{Plus}[\ll 2\gg]^{1/3}}+\text{Plus}[\ll 2\gg]^{1/3}\right)}+\frac{i}{\sqrt{\frac{2}{\text{Plus}[\ll 2\gg]}}}\right),\ \frac{2\left(2+\frac{1}{\left(\frac{7}{2}+\frac{3\sqrt{5}}{2}\right)^{1/3}}+\left(\frac{7}{2}+\frac{3\sqrt{5}}{2}\right)^{1/3}\right)}{-\frac{3}{2}-\frac{-6+3\,\text{Plus}[\ll 3\gg]}{2\,(2+\text{Power}[\ll 2\gg]+\text{Power}[\ll 2\gg])}+\frac{i}{\sqrt{2\,\text{Power}[\ll 2\gg]}}},\right.$$

$$\frac{1}{2}\left(-\frac{3}{2}+\frac{-6+3\,(2+\text{Power}[\ll 2\gg]+\text{Power}[\ll 2\gg])}{2\left(2+\frac{1}{\text{Plus}[\ll 2\gg]^{1/3}}+\text{Plus}[\ll 2\gg]^{1/3}\right)}-\frac{1}{\sqrt{\frac{2}{\text{Plus}[\ll 2\gg]}}}\right),$$

$$\left.\frac{2\left(\frac{1}{2}\left(-2-\frac{1}{\text{Plus}[\ll 2\gg]^{1/3}}-\text{Plus}[\ll 2\gg]^{1/3}\right)+\frac{1}{2}\left(2+\frac{1}{\left(\frac{7}{2}+\text{Times}[\ll 2\gg]\right)^{1/3}}+\left(\frac{7}{2}+\text{Times}[\ll 2\gg]\right)^{1/3}\right)\right)}{-\frac{3}{2}+\frac{-6+3\,\text{Plus}[\ll 3\gg]}{2\,(2+\text{Power}[\ll 2\gg]+\text{Power}[\ll 2\gg])}-\frac{1}{\sqrt{2\,\text{Power}[\ll 2\gg]}}}\right\}.\gg$$

Controllo che le radici diano tutte 0

Dovrebbe essere un vettore di zeri :
$\{0.\times 10^{-98}+0.\times 10^{-98}\,i,\ 0.\times 10^{-97}+0.\times 10^{-97}\,i,\ 0.\times 10^{-99},\ 0.\times 10^{-99}\}$

Ora faccio una prova dei tre possibili casi: o quattro radici reali o due reali e due complesse o quattro complesse...

In[14]:=

```
Print["Trovo un polinomio con quattro  radici reali \n",
  Expand[(x - 7) (x + 9) (x - 5) (x - 1)]];
unpol = {-315, 388, -70, -4, 1};
Print["Polinomio : ", unpol, "\nRadici : ",
  N[solve4[unpol], 16]];
Print["Trovo un polinomio con due radici reali \n",
  Expand[(x - 7) (x + 9) (x^2 + x + 1)]];
unpol = {-63, -61, -60, +3, 1};
Print["Polinomio : ", unpol, "\nRadici : ",
  N[solve4[unpol], 16]];
Print["Trovo un polinomio con quattro radici complesse \n",
  Expand[(x^2 - 13 x + 49) (x^2 + 5 x + 9)]];
unpol = {441, 128, -7, -8, 1};
radici = solve4[unpol];
Print["Polinomio : ", unpol, "\nRadici : ", N[radici, 16]];
forsezero = N[unpol. Table[x^(j - 1), {j, 5}] /.
   x -> N[radici, 50], 50];
Print["Controllo che le radici diano tutte 0 "];
Print["Dovrebbe essere un vettore di zeri : ",
  forsezero]
```

Trovo un polinomio con quattro radici reali
$-315 + 388\,x - 70\,x^2 - 4\,x^3 + x^4$

Polinomio : {-315, 388, -70, -4, 1}
Radici :
{7.000000000000000, 1.000000000000000, -9.000000000000000, 5.000000000000000}

Trovo un polinomio con due radici reali
$-63 - 61\,x - 60\,x^2 + 3\,x^3 + x^4$

Polinomio : {-63, -61, -60, 3, 1}
Radici : {-0.5000000000000000 + 0.8660254037844386 i,
-0.5000000000000000 - 0.8660254037844386 i, -9.000000000000000, 7.000000000000000}

Trovo un polinomio con quattro radici complesse
$441 + 128\,x - 7\,x^2 - 8\,x^3 + x^4$

Polinomio : {441, 128, -7, -8, 1}
Radici :
{6.500000000000000 + 2.598076211353316 i, -2.500000000000000 - 1.658312395177700 i,
6.500000000000000 - 2.598076211353316 i, -2.500000000000000 + 1.658312395177700 i}

Controllo che le radici diano tutte 0

Dovrebbe essere un vettore di zeri :
$\{0.\times 10^{-46} + 0.\times 10^{-46}\,i,\ 0.\times 10^{-47} + 0.\times 10^{-47}\,i,\ 0.\times 10^{-46} + 0.\times 10^{-46}\,i,\ 0.\times 10^{-47} + 0.\times 10^{-47}\,i\}$

Ed infine faccio un bel po' di prove casuali ma escludendo il caso che il termine noto risulti zero.

In[27]:=

```
For[mieitest = 3, mieitest > 0, mieitest--,
 Print["=====\nConto alla rovescia: ", mieitest];
  unpo = {RandomInteger[{-7, 7}] / 5,
     RandomInteger[{-999, 999}] / 500, RandomInteger[{-999, 999}] / 500,
   RandomInteger[{-999, 999}] / 500, 1};
  If[unpo[[1]] == 0, unpo[[1]] = -1];
  Print["Un polinomio : ", unpo];
  radici = solve4[unpo];
  Print[N[radici]];
  Print[N[Abs[radici]]];
  forsezero = N[unpo . Table[x^(j - 1), {j, 5}] /.
   x -> N[radici, 50], 50];
  Print["Dovrebbe essere un vettore di zeri : ",
  forsezero]];
```

=====
Conto alla rovescia: 3

Un polinomio : $\left\{-\frac{1}{5}, \frac{909}{500}, \frac{87}{125}, \frac{37}{500}, 1\right\}$

{-1.09832, 0.105624, 0.459351 + 1.23004 i, 0.459351 - 1.23004 i}

{1.09832, 0.105624, 1.31301, 1.31301}

Dovrebbe essere un vettore di zeri :
$\left\{0.\times 10^{-50}, 0.\times 10^{-51}, 0.\times 10^{-49} + 0.\times 10^{-49}\, i, 0.\times 10^{-49} + 0.\times 10^{-49}\, i\right\}$

=====
Conto alla rovescia: 2

Un polinomio : $\left\{\frac{7}{5}, \frac{449}{250}, \frac{999}{500}, -\frac{112}{125}, 1\right\}$

{0.876359 + 1.50695 i, -0.428359 - 0.526501 i, 0.876359 - 1.50695 i, -0.428359 + 0.526501 i}

{1.74324, 0.678745, 1.74324, 0.678745}

Dovrebbe essere un vettore di zeri :
$\left\{0.\times 10^{-48} + 0.\times 10^{-49}\, i, 0.\times 10^{-50} + 0.\times 10^{-50}\, i, 0.\times 10^{-48} + 0.\times 10^{-49}\, i, 0.\times 10^{-50} + 0.\times 10^{-50}\, i\right\}$

=====
Conto alla rovescia: 1

Un polinomio : $\left\{\frac{6}{5}, -\frac{57}{50}, \frac{23}{500}, \frac{91}{100}, 1\right\}$

{-1.05815 - 0.937428 i, 0.603155 + 0.486484 i, -1.05815 + 0.937428 i, 0.603155 - 0.486484 i}

{1.41367, 0.774895, 1.41367, 0.774895}

Dovrebbe essere un vettore di zeri :
$\left\{0.\times 10^{-49} + 0.\times 10^{-49}\, i, 0.\times 10^{-50} + 0.\times 10^{-50}\, i, 0.\times 10^{-49} + 0.\times 10^{-49}\, i, 0.\times 10^{-50} + 0.\times 10^{-50}\, i\right\}$

Ora provo a vedere che problemi nascono a risolvere un caso che so che è mal risolto col metodo che sfrutta l'algoritmo di Tartaglia...

In[28]:=

```
nonfacile = {923 521 / 10 000, 119 164 / 1000, 5766 / 100, 124 / 10, 1};
Print[N[nonfacile]];
radici = solve4[nonfacile];
Print["Le radici trovate (eventualmente numeri complessi:\n ", N[radici]];
Print["I loro valori assoluti ", N[Abs[radici]]];
forsezero = N[nonfacile . Table[x^(j - 1), {j, 5}] /.
   x -> N[radici, 50], 50];
Print["Tutto bene se ottengo  tutti zeri :\n", forsezero]
```

```
{92.3521, 119.164, 57.66, 12.4, 1.}

Le radici trovate (eventualmente numeri complessi:
 {-3.1, -3.1, -3.1, -3.1}

I loro valori assoluti {3.1, 3.1, 3.1, 3.1}

Tutto bene se ottengo  tutti zeri :
```

$\{0.\times 10^{-47}, 0.\times 10^{-47}, 0.\times 10^{-47}, 0.\times 10^{-47}\}$

La soluzione è vistosamente instabile. Basta modificare di un valore piccolissimo i dati e si ottengono radici molto diverse... Tuttavia il calcolo delle radici risulta preciso... La modifica dei coefficienti però è fatta in modo rigoroso e non usando numeri in virgola mobile... Per esempio:

In[35]:=

```
circanonfacile = nonfacile;
circanonfacile[[1]] = circanonfacile[[1]] + 1 / 1 000 000 000;
Print[N[circanonfacile, 20]];
circaradici = solve4[circanonfacile];
Print["Le radici trovate (eventualmente numeri complessi:\n ",
  N[circaradici]];
Print["I loro valori assoluti ", N[Abs[circaradici]]];
circaforsezero = N[circanonfacile . Table[x^(j - 1), {j, 5}] /.
   x -> N[circaradici, 50], 50];
Print["Tutto bene se ottengo  tutti zeri :\n", circaforsezero]
```

```
{92.352100001000000000, 119.16400000000000000,
 57.660000000000000000, 12.400000000000000000, 1.0000000000000000000}

Le radici trovate (eventualmente numeri complessi:
 {-3.10398 - 0.00397635 i, -3.09602 + 0.00397635 i,
  -3.10398 + 0.00397635 i, -3.09602 - 0.00397635 i}

I loro valori assoluti {3.10398, 3.09603, 3.10398, 3.09603}

Tutto bene se ottengo  tutti zeri :
```

$\{0.\times 10^{-47} + 0.\times 10^{-47}\, i, 0.\times 10^{-47} + 0.\times 10^{-47}\, i, 0.\times 10^{-47} + 0.\times 10^{-47}\, i, 0.\times 10^{-47} + 0.\times 10^{-47}\, i\}$

Oppure dando una piccola modifica in questo modo...

In[43]:=

```
circanonfacile = nonfacile + 1 / 1 000 000 000;
circanonfacile[[5]] = 1;
Print[N[circanonfacile, 20]];
circaradici = solve4[circanonfacile];
Print["Le radici trovate (eventualmente numeri complessi:\n ",
  N[circaradici]];
Print["I loro valori assoluti ", N[Abs[circaradici]]];
circaforsezero = N[circanonfacile . Table[x^(j - 1), {j, 5}] /.
   x -> N[circaradici, 50], 50];
Print["Tutto bene se ottengo  tutti zeri :\n", circaforsezero]
```

```
{92.352100001000000000, 119.16400000100000000,
 57.660000001000000000, 12.400000001000000000, 1.0000000000000000000}

Le radici trovate (eventualmente numeri complessi:
 {-3.11226, -3.08782, -3.09996 + 0.0122174 i, -3.09996 - 0.0122174 i}

I loro valori assoluti {3.11226, 3.08782, 3.09998, 3.09998}

Tutto bene se ottengo  tutti zeri :
```

$\{0. \times 10^{-47}, 0. \times 10^{-47}, 0. \times 10^{-47} + 0. \times 10^{-47}\, i, 0. \times 10^{-47} + 0. \times 10^{-47}\, i\}$

Questo però non vale quando il calcolo NON E' fatto a livello simbolico... Uso valori numerici... e viene uno schifo...

In[51]:=

```
circanonfacile = N[nonfacile + 1 / 1 000 000 000];
circanonfacile[[5]] = 1;
Print[N[circanonfacile, 20]];
circaradici = solve4[circanonfacile];
Print["Le radici trovate (eventualmente numeri complessi:\n ",
  N[circaradici]];
Print["I loro valori assoluti ", N[Abs[circaradici]]];
circaforsezero = N[circanonfacile . Table[x^(j - 1), {j, 5}] /.
   x -> N[circaradici, 50], 50];
Print["Tutto bene se ottengo  tutti zeri :\n", circaforsezero]
```

```
{92.3521, 119.164, 57.66, 12.4, 1.0000000000000000000}

Le radici trovate (eventualmente numeri complessi:
 {-3.10166 + 0.00115913 i,
  -3.09834 + 0.002033 i, -3.10166 - 0.00115913 i, -3.09834 - 0.002033 i}

I loro valori assoluti {3.10166, 3.09834, 3.10166, 3.09834}

Tutto bene se ottengo  tutti zeri :
```

$\{-2.23329 \times 10^{-8} + 1.66923 \times 10^{-11}\, i, -2.22852 \times 10^{-8} + 2.92452 \times 10^{-11}\, i,$
$-2.23329 \times 10^{-8} - 1.66923 \times 10^{-11}\, i, -2.22852 \times 10^{-8} - 2.92452 \times 10^{-11}\, i\}$

Insomma... il trovare le radici dell'equazione di quarto grado è intrinsecamente una cosa delicata se il calcolo non viene fatto a livello simbolico..

Conclusione

Mi sembra che la traduzione da Javascript a *Mathematica* del solutore dell'equazione di quarto grado funzioni...
Ora dovrei fare il contrario ossia tradurre il solutore robustissimo di terzo grado da Mathematichese a Javascript e vedere come il tutto funziona...

In[59]:=
```
DateString[]
```

Out[59]=
```
Wed 15 May 2013 13:40:54
```