# Programmimplementierung ab formalem Ansatz

## Umfeld

Liegt zur Lösung ein formaler Weg vor, kann daraus in der Regel einfach ein Entwurf abgeleitet werden. Dies trifft vor allem für mathematische Verfahren zu.

Der Entwurf wird auf einer hohen Ebene durchgeführt, so dass dabei die Hauptfunktionalitäten dargestellt werden.

Nach der Implementierung werden sämtliche Funktionalitäten systematisch geprüft. Dies geschieht mit aussagekräftigen Testbeispielen. Dabei gilt die Regel, dass jeder Programmflusszweig mindestens einmal durchlaufen werden muss. Besondere Beachtung ist Werten zu schenken, die während des Programmlaufes ein Ausnahmeverhalten auslösen können, z.B. eine Division durch Null.

## Aufgabe

Entwerfen Sie aufgrund der folgenden Zusammenstellung ein Programm zur Lösung der kubischen Gleichung und implementieren Sie Ihren Entwurf mit Visual C++.

66 *2. Gleichungen, Funktionen, Vektorrechnung*

**2.1.2.3. Kubische Gleichungen**

*Allgemeine Form* $Ax^3 + Bx^2 + Cx + D = 0;\ A \neq 0.$
Division durch $A$ führt auf die

Normalform $\{x \mid x^3 + ax^2 + bx + c = 0\}$

worin $a = \dfrac{B}{A}, \quad b = \dfrac{C}{A}, \quad c = \dfrac{D}{A}$ ist.

Durch Substitution $x = y - \dfrac{a}{3}$ erhält man die

*reduzierte Form* $y^3 + py + q = 0$

**Cardanische Lösungsformel** für die reduzierte Gleichung

$$y_1 = u + v$$

$$y_2 = -\frac{u+v}{2} + \frac{u-v}{2}\,\mathrm{j}\sqrt{3}$$

$$y_3 = -\frac{u+v}{2} - \frac{u-v}{2}\,\mathrm{j}\sqrt{3}$$

wobei

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

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

*Diskriminante* $D = \left(\dfrac{q}{2}\right)^2 + \left(\dfrac{p}{3}\right)^3$

$D > 0$ ergibt eine reelle und zwei konjugiert komplexe Lösungen.

$D = 0$ ergibt drei reelle Lösungen, darunter eine Doppelwurzel.

**Casus irreducibilis**

$D < 0$ ergibt drei reelle Lösungen, die sich auf goniometrischem Wege errechnen lassen (irreduzibler Fall, casus irreducibilis)

$$y_1 = 2\sqrt{\frac{|p|}{3}}\cos\frac{\varphi}{3}$$

$$y_2 = -2\sqrt{\frac{|p|}{3}}\cos\left(\frac{\varphi}{3} - 60°\right)$$

$$y_3 = -2\sqrt{\frac{|p|}{3}}\cos\left(\frac{\varphi}{3} + 60°\right)$$

*2.1. Gleichungen* 67

$\varphi$ läßt sich aus der Gleichung $\cos\varphi = \dfrac{-\dfrac{q}{2}}{\sqrt{\left(\dfrac{|p|}{3}\right)^3}}$ berechnen.

Die entsprechenden $x$-Werte folgen aus obiger Substitution $x = y - \dfrac{a}{3}$.

*1. Beispiel:* $D > 0)$

$\{x \mid x^3 - 3x^2 + 4x - 4 = 0\}$

Substitution $x = y - \dfrac{-3}{3} = y + 1$

$(y+1)^3 - 3(y+1)^2 + 4(y+1) - 4 = 0$

$y^3 + 3y^2 + 3y + 1 - 3y^2 - 6y - 3 + 4y + 4 - 4 = 0$

Reduzierte Form $y^3 + y - 2 = 0 \qquad p = 1;\ q = -2$

$$u = \sqrt[3]{-\frac{-2}{2} + \sqrt{\left(\frac{-2}{2}\right)^2 + \left(\frac{1}{3}\right)^3}} = \sqrt[3]{1 + \sqrt{\frac{28}{27}}} \approx$$
$$\approx 1{,}264$$

$$v = \sqrt[3]{1 - \sqrt{\frac{28}{27}}} \approx -0{,}264$$

$y_1 = 1{,}264 - 0{,}264 = 1 \qquad \underline{\underline{x_1 = 2}}$

$y_2 = -\dfrac{1}{2} + \dfrac{1{,}528}{2}\,\mathrm{j}\sqrt{3} \qquad \underline{\underline{x_2 = \dfrac{1}{2} + 0{,}764\,\mathrm{j}\sqrt{3}}}$

$y_3 = -\dfrac{1}{2} - \dfrac{1{,}528}{2}\,\mathrm{j}\sqrt{3} \qquad \underline{\underline{x_3 = \dfrac{1}{2} - 0{,}764\,\mathrm{j}\sqrt{3}}}$

bzw.

$$E = \left\{2;\quad \frac{1}{2} + 0{,}764\,\mathrm{j}\sqrt{3};\quad \frac{1}{2} - 0{,}764\,\mathrm{j}\sqrt{3}\right\}$$

*2. Beispiel:* $(D < 0)$

$\{x \mid x^3 - 21x - 20 = 0\}$ (bereits reduzierte Form)

Diskriminante $D = \left(\dfrac{q}{2}\right)^2 + \left(\dfrac{p}{3}\right)^3 = (-10)^2 + (-7)^3 < 0$

5*

68 *2. Gleichungen, Funktionen, Vektorrechnung*

Goniometrischer Lösungsweg:

$\cos\varphi = \dfrac{10}{\sqrt{343}};\ \varphi = 57°\,19'$

$x_1 = 2\sqrt{7}\cos 19°\,6'; \qquad \underline{\underline{x_1 = 5}}$

$x_2 = -2\sqrt{7}\cos(-40°\,54') \qquad \underline{\underline{x_2 = -4}}$

$x_3 = -2\sqrt{7}\cos 79°\,6' \qquad \underline{\underline{x_3 = -1}}$

bzw.

$E = \{5, -4, -1\}$

Sonderfälle der kubischen Gleichung in reduzierter Form

a) $p = 0 \quad \{y \mid y^3 + q = 0\}$ (*binomische Gleichung*)
Lösung siehe Abschnitt „Komplexe Zahlen", S. 13.

b) $q = 0 \quad \{y \mid y^3 + py = 0\}$

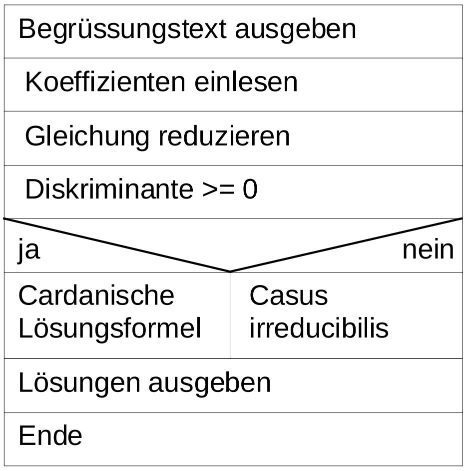
Lösung:

$$y(y^2 + p) = 0 \quad y_1 = 0; \quad y_{2;3} = \begin{cases} \pm\sqrt{-p} & \text{für } p \leqq 0 \\ \pm\,\mathrm{j}\sqrt{p} & \text{für } p > 0 \end{cases}$$

Quelle: Bartsch, Taschenbuch mathematischer Formeln, 1982.

Ein Entwurf der Lösung sieht folgenden Ablauf gemäss formaler Definition vor:

## Kubische Gleichung




Eine mögliche Realisation:

```c
/* IAM Uebung: Loesung der kubischen Gleichung mit Hilfe der Cardanischen Loesungsformel.

   Arbeitsweise:
   Nach Eingabe der Polynomkoeffizienten a,b,c,d wird das reduzierte Polynom mit den
   p, q gebildet und die Diskriminante diskr berechnet:

   Ax^3 + Bx^2 +Cx + D == 0
   a=B/A, b = C/A, c = D/A            -> y^3 + py + q == 0   (mit x == y + 1/3 a)

   Ist die Diskriminante positiv, werden die Loesungen mit der cardanischen Loesungsformel
   berechnet. Ist die Diskriminante negativ (casus irreducibilis), werden die Loesungen mit
   einem trigonometrischen Ansatz bestimmt.

   Referenz: Bartsch, Taschenbuch mathematischer Formeln, S.66-68,(1982)

   Autor: Gerhard Krucker
   Datum: 4.5.2000, 10.5.2005
   Sprache: MS Visual C++ V5.0 (WindowsNT Console Application)
            MS Visual C++ .NET 2003 (Win32 Console Application)
 */

 #include <stdio.h>          // Ein- Ausgabeoperationen
 #include <math.h>           // Fuer sqrt, acos, cos, pow
 #include <conio.h>          // Fuer _kbhit


 /* double sqrt3(double);
    Berechnen des Hauptwertes der kubischen Wurzel aus x, wobei auch negative x zugelassen sind.
 */

 double sqrt3 (double x)
 { if (x == 0)
         return 0.0;
     else if ( x > 0.0)
             return pow(x, 1.0 / 3.0);
         else
              return - pow(fabs(x), 1.0 / 3.0);
 }
```

```
main()
 { double A, B, C, D;          // Polynomkoeffizienten des Originalpolynoms
   double a,b,c;               // Koeffizienten des normalsierten Polynoms
   double p,q;                 // Polynomkoeffizienten des reduzierten Polynoms
   double diskr;               // Diskriminante
   double u,v;                 // Werte fuer cardanische Loesungsformel

   double y1re;                 // Loesung y1 (immer reeel)
   double y2re, y2im;           // Loesung y2 (Real und Imaginaerteil)
   double y3re, y3im;           // Loesung y3     "

   printf("Loesung der kubischen Gleichung ax^3 + bx^2 + cx + d == 0\n");
   printf("Eingabe der Koeffizienten a b c d: ");

   scanf("%lf %lf %lf %lf",&A,&B, &C, &D);     // Koeffizienten einlesen

   // Normalform bilden
   a = B / A; b = C / A; c = D / A;

  // Reduziertes Polynom und Diskriminante bilden
   p = b - 1.0 / 3.0 * a * a;                 // p = c - 1/3 a^2
   q = 2.0 / 27.0 * a * a * a- 1.0 / 3.0 * a * b + c;
   diskr = p * p * p / 27.0 + q * q / 4.0;

   if (diskr > 0)    // Loesungen nach Cardano
   { u = sqrt3(-q / 2.0 + sqrt(diskr));
     v = sqrt3(-q / 2.0 - sqrt(diskr));

       y1re = u + v;
       y2re = y3re = - (u + v) / 2.0;
       y2im = (u - v) / 2.0 * sqrt(3.0);
       y3im = - y2im;
   }
   else   // casus irreducibilis
   { double spabs3;
     double phi;
     const double pi=3.14159265359;

     spabs3 = sqrt(fabs(p)/ 3.0);
     phi = acos ((-q / 2.0) / sqrt(fabs(p) * fabs(p) * fabs(p) / 27.0));

       y1re =  2.0 * spabs3 * cos(phi / 3.0);
       y2re = -2.0 * spabs3 * cos(phi / 3.0 - pi / 3.0);
       y3re = -2.0 * spabs3 * cos(phi / 3.0 + pi / 3.0);
    y2im = y3im = 0.0;
   }

   // Resulate ausgeben
   printf("Loesungen\n");
   printf(" x1: %4.3g \n",y1re-a/3.0);
   printf(" x2: %4.3g %+4.3gj\n",y2re-a/3.0,y2im);
   printf(" x3: %4.3g %+4.3gj\n",y3re-a/3.0,y3im);

   while (!_kbhit());

   return 0;
 }
```