

C++ für Ingenieure

Einführung in die objektorientierte Programmierung

Kapitel 1 Grundlagen der Programmierung

1.2 Grundstrukturen

Erstellt am

Beschreibung Übungsbeispiel: Vektoren – Kräfte im Raum

Vektoren – Kräfte im Raum

Außer der Möglichkeit der Komponentenangabe, gibt es zur Richtungsangabe eines Vektors zum Ursprung und den Koordinaten, die Angabe der Richtungswinkel (**Bild 1**).

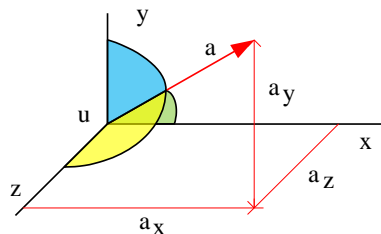


Bild 1 Richtungswinkel

Nehmen Sie deren Bestimmung in Ihre Berechnung mit auf.

$$\varphi_i = \arccos \frac{a_i}{|\alpha|}, \quad |\alpha| = \sqrt{\sum_{i=x,y,z} a_i^2} \quad (1)$$

Zur Bestimmung der Richtungswinkel ist die obige Formel in das vorhandene Programm zu integrieren. Ein paar zusätzliche Vereinfachungen machen den Code lesbarer. Da die Winkel üblicherweise in Grad anzugeben sind, muss noch eine Umrechnung von Bogenmaß in Grad durchgeführt werden. Dies erfolgt mit der Formel.

$$\text{Winkelmaß} = \text{Bogenmaß} \cdot \frac{180^\circ}{\pi} \quad (2)$$

Damit ändert sich das Programm wie nachfolgend dargestellt.

Code 1 Anwendungsbeispiel – Kräfte im Raum

```
/* Kraefte.cpp
   Das Programm bestimmt
   mit Parametern von Kräften im freien Raum
   resultierende Kraft, Moment und Dyname
   so wie der Richtungswinkel
*/

#include <iostream.h>
#include <math.h>

int main ()
{
    char weiter;
    double x, y, z;
    double Fx, Fy, Fz;
```

```
double Sux, Suy, Suz;
double SuFx, SuFy, SuFz;
double SuMx, SuMy, SuMz;
double Fr, Mr, p;
double Mx, My, Mz;
double ax, ay, az;
// neu
double alpha;
double phi_x, phi_y, phi_z;
double w_x, w_y, w_z;
// ---
do
{
    cout << "weiter (E=Ende/N=Neu/W=Weiter/A=Ausgabe) ";
    cin >> weiter;
    switch (weiter)
    {
        case 'E': /* Ende */
            break;
        case 'N': /* Neuanfang */
            Sux = 0;
            Suy = 0;
            Suz = 0;
            SuFx = 0;
            SuFy = 0;
            SuFz = 0;
            SuMx = 0;
            SuMy = 0;
            SuMz = 0;
        case 'W': /* weiter */
            // Eingaben
            cout << "Angriffspunkt (x) = ";
            cin >> x;
            cout << "Angriffspunkt (y) = ";
            cin >> y;
            cout << "Angriffspunkt (z) = ";
            cin >> z;

            cout << "Kraftkomponente (x) = ";
            cin >> Fx;
            cout << "Kraftkomponente (y) = ";
            cin >> Fy;
            cout << "Kraftkomponente (z) = ";
            cin >> Fz;

            // Auswertung
            Sux = Sux+x;
            Suy = Suy+y;
            Suz = Suz+z;

            SuFx = SuFx+Fx;
            SuFy = SuFy+Fy;
            SuFz = SuFz+Fz;

            SuMx = SuMx+y*Fz-z*Fy;
```

```

        SuMy = SuMy+z*Fx-x*Fz;
        SuMz = SuMz+x*Fy-y*Fx;
        break;
    case 'A': /* Ausgabe */
        Fr = sqrt(SuFx*SuFx+SuFy*SuFy+SuFz*SuFz);
        Mr = sqrt(SuMx*SuMx+SuMy*SuMy+SuMz*SuMz);
        ax = (SuFy*SuMz-SuFz*SuMy)/(Fr*Fr);
        ay = (SuFz*SuMx-SuFx*SuMz)/(Fr*Fr);
        az = (SuFx*SuMy-SuFy*SuMx)/(Fr*Fr);
        p = SuFx*SuMx+SuFy*SuMy+SuFz*SuMz;
        Mx = p*SuFx/(Fr*Fr);
        My = p*SuFy/(Fr*Fr);
        Mz = p*SuFz/(Fr*Fr);
        alpha = sqrt(ax*ax+ay*ay+az*az);
        phi_x = acos (ax/alpha);
        phi_y = acos (ay/alpha);
        phi_z = acos (az/alpha);

        // Umrechnung vom Bogenmaß in Winkelgrade
        w_x = phi_x/4/atan(1)*180;
        w_y = phi_y/4/atan(1)*180;
        w_z = phi_z/4/atan(1)*180;

        cout << "Resultierende Kraft    = " << Fr << endl;
        cout << "Resultierendes Moment = " << Mr << endl;
        cout << "Dynamie Ortsvektor (x) = " << ax << endl;
        cout << "Dynamie Ortsvektor (y) = " << ay << endl;
        cout << "Dynamie Ortsvektor (z) = " << az << endl;
        cout << "Momentenvektor      (x) = " << Mx << endl;
        cout << "Momentenvektor      (y) = " << My << endl;
        cout << "Momentenvektor      (z) = " << Mz << endl;
        cout << "Dynamie              (alpha) = " << alpha << endl;
        cout << "Richtungswinkel (w_x) = " << w_x << endl;
        cout << "Richtungswinkel (w_y) = " << w_y << endl;
        cout << "Richtungswinkel (w_z) = " << w_z << endl;
        break;
    default:
        cout << "Falsche Eingabe!" << endl;
    }
}
while (weiter!='E');
}

```

Mit dem gleichen Beispiel wie im Buch erfolgt die Ausgabe nach Bild 2.

```
weiter <E=Ende/N=Neu/W=Weiter/A=Ausgabe> N
Angriffspunkt <x> = 3
Angriffspunkt <y> = 0
Angriffspunkt <z> = 0
Kraftkomponente <x> = 0
Kraftkomponente <y> = 3
Kraftkomponente <z> = 0
weiter <E=Ende/N=Neu/W=Weiter/A=Ausgabe> W
Angriffspunkt <x> = 0
Angriffspunkt <y> = 1
Angriffspunkt <z> = 0
Kraftkomponente <x> = 0
Kraftkomponente <y> = 0
Kraftkomponente <z> = 4
weiter <E=Ende/N=Neu/W=Weiter/A=Ausgabe> W
Angriffspunkt <x> = 0
Angriffspunkt <y> = 0
Angriffspunkt <z> = 2
Kraftkomponente <x> = 5
Kraftkomponente <y> = 0
Kraftkomponente <z> = 0
weiter <E=Ende/N=Neu/W=Weiter/A=Ausgabe> A
Resultierende Kraft = 7.07107
Resultierendes Moment = 14.0357
Dynamische Ortsvektor <x> = -0.26
Dynamische Ortsvektor <y> = -0.58
Dynamische Ortsvektor <z> = 0.76
Momentenvektor <x> = 8.6
Momentenvektor <y> = 5.16
Momentenvektor <z> = 6.88
Dynamische <alpha> = 0.990757
Richtungswinkel <w_x> = 105.214
Richtungswinkel <w_y> = 125.832
Richtungswinkel <w_z> = 39.9067
weiter <E=Ende/N=Neu/W=Weiter/A=Ausgabe> _
```

Bild 2 Die Ausgabe des Beispiels