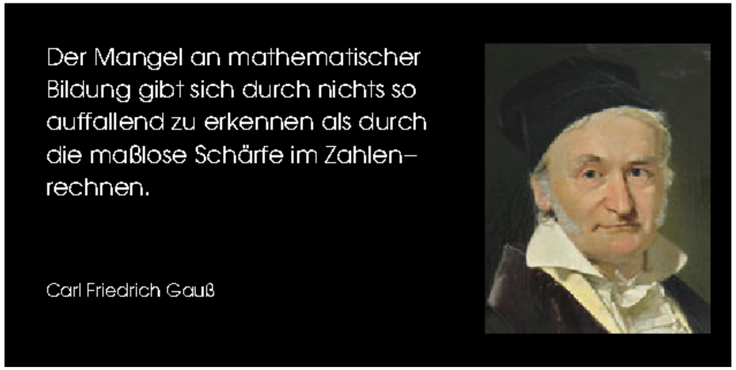
# 7. Arithmetik

Hochschule RheinMain




https://beruhmte-zitate.de/zitate/2001405-carl-friedrich-gauss-der-mangel-an-mathematischer-bildung-gibt-sich-dur/



Notizen



## Arithmetik

Hochschule RheinMain

Die **Arithmetik** ist ein Teilgebiet der Mathematik. Sie umfasst das Rechnen mit den Zahlen, vor allem den natürlichen Zahlen. Sie beschäftigt sich mit den Grundrechenarten, also mit der Addition, Subtraktion, Multiplikation, Division sowie den zugehörigen Rechengesetzen. Zur Arithmetik gehört auch die Teilbarkeitslehre mit den Gesetzen der Teilbarkeit ganzer Zahlen sowie der Division mit Rest.

Quelle: Wikipedia

Beispiel: Berechne $ax^5 + bx^4 + cx^3 + dx^2 + ex + f$ in einem C-Programm

a) `a*x*x*x*x*x + b*x*x*x*x + c*x*x*x + d*x*x +e*x + f`
b) `((((a*x + b)*x + c)*x + d)*x +e)*x + f`

Was ist der Unterschied zwischen den beiden Varianten?
Variante a) ist die „naheliegende" mathematische Umsetzung der Formel.
Variante b) ist wegen der vielen Klammern schlechter „mit einem Blick" zu erfassen.
Aber
Die Variante a) enthält 15 Multiplikationen, b) hingegen nur 5.
Die Variante b) lässt sich also viel effizienter berechnen.
In der zweiten Formel spielt die Musik der Informatik:

a mal x + b mal x + c mal x + d mal x + e mal x + f

→ Die zweite Formel (sog. *Horner Schema*) lässt sich viel besser „algorithmisieren".



Notizen

# Algorithmisierung von Formeln

Hochschule RheinMain

Ausgangsformel: $((((ax + b)x + c)x + d)x + e) + f$

Indizierung der Variablen: $((((a_0x + a_1)x + a_2)x + a_3)x + a_4) + a_5$

Iterative Berechnung:
$$\begin{aligned} z_0 &= a_0 \\ z_1 &= z_0x + a_1 \\ z_2 &= z_1x + a_2 \\ z_3 &= z_2x + a_3 \\ z_4 &= z_3x + a_4 \\ z_5 &= z_4x + a_5 \end{aligned}$$

Notation:
$$z_n = \begin{cases} a_0 & \text{für } n = 0 \\ z_{n-1}x + a_n & \text{für } n = 2, 3, 4, 5 \end{cases}$$

Viele Aufgaben, die in der Mathematik durch geschlossene Ausdrücke formuliert werden, lassen sich algorithmisieren. Zum Beispiel die Entwicklung von Funktionen in Potenzreihen oder Lösung von Gleichungssystemen durch Iterationsverfahren. Computerprogramme sind für iterative Berechnungen besonders geeignet.



Notizen



# Zahlenfolgen

Hochschule RheinMain

Anschauliche Definition einer (unendlichen) Zahlenfolge:

$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots$

Explizite Definition der selben Zahlenfolge:

$a_k = \frac{1}{2^k}, k = 0, 1, 2, \ldots$

Die explizite Definition ermöglicht eine direkte Berechnung
Induktive Definition:

- Die Folge startet mit $a_0 = 1$.
- Jedes weitere Folgenglied erhält man durch Halbierung des vorherigen: $a_k = \frac{a_{k-1}}{2}$

Als Formel:
$$a_k = \begin{cases} 1 & \text{falls } k = 0 \\ \frac{a_{k-1}}{2} & \text{falls } k = 1, 2, 3, \ldots \end{cases}$$

Die induktive Definition ermöglicht eine iterative Berechnung



Notizen

# Beispiel

Hochschule RheinMain

**Aufgabenstellung**

Eine Studentin möchte bei seiner Bank ein Darlehen in einer bestimmten Höhe aufnehmen.
Sie vereinbart eine feste monatliche Ratenzahlung.
Diese Rate dient dazu, die monatlich anfallenden Zinsen zu bezahlen und enthält darüber hinaus einen Tilgungsbetrag, mit dem das Darlehen abbezahlt wird.
In dem Maße, in dem die Darlehensschuld abgetragen wird, sinkt der Anteil der Zinsen an der monatlichen Ratenzahlung und der Tilgungsbetrag wächst entsprechend.
Daraus ergibt sich ein ganz bestimmter Tilgungsplan, den wir aufstellen wollen.
Darüber hinaus wollen wir noch einige, durchaus bankenübliche, Zusatzregelungen wie zum Beispiel Zinsbindung und Sondertilgungen in die Berechnung einfließen lassen.



Notizen



# Präzisierung der Aufgabenstellung

Hochschule RheinMain

Ausgangspunkt für den Tilgungsplan ist die anfängliche Darlehenssumme bzw. die **Restschuld**, die jeweils noch zu Buche steht.
Mit der Bank wird ein sogenannter **Nominalzins** vereinbart.
Die Restschuld wird monatlich mit $\frac{1}{12}$ dieses Nominalzinses verzinst.
Die monatlich zu zahlende **Rate** wird ebenfalls festgelegt und muss natürlich größer als die anfallenden Zinsen sein, damit noch ein Tilgungsbetrag übrigbleibt.
Der **Tilgungsbetrag** ergibt sich dann aus der Monatsrate nach Abzug der monatlichen Zinsen.
Wegen des Risikos von Zinsschwankungen garantiert die Bank den obigen Nominalzins nur über einen gewissen Zeitraum. In diesem Zeitraum besteht dann eine **Zinsbindung**.
Nach Ablauf der Zinsbindung gelten die dann marktüblichen Zinsen, die im Vorhinein natürlich nur geschätzt werden können und ein gewisses Risiko in dem Tilgungsplan darstellen.
Letztlich wird mit der Bank noch vereinbart, dass jährliche **Sondertilgungen** in einer bestimmten Höhe getätigt werden können.



Notizen

# Formalisierung und Modellierung

Hochschule RheinMain

Mit $rest_n$ bezeichnen wir die Restschuld nach Ablauf von $n$ Monaten.
Damit ist $rest_0$ der volle Darlehensbetrag.
Es gibt zwei Zinssätze:

- $zins_1$ für den Zeitraum innerhalb,
- $zins_2$ für den Zeitraum außerhalb der Zinsbindung

Die Zinsbindung ($bindung$) wird dabei in Jahren angegeben.
Damit gilt für den Zinssatz ($zins_n$) im $n$-ten Monat:

$$zins_n = \begin{cases} zins_1 & \text{falls } n \leq bindung \cdot 12 \\ zins_2 & \text{falls } n > bindung \cdot 12 \end{cases}$$

Daraus ergibt sich die monatliche Zinslast ($zinsen_n$):

$$zinsen_n = \frac{rest_n \cdot zins_n}{12 \cdot 100}$$



Notizen



# Formalisierung und Modellierung (2)

Hochschule RheinMain

Was von der monatlichen Rate nach Abzug der Zinsen übrig bleibt, dient zur Tilgung, wobei maximal in Höhe der Restschuld getilgt wird:

$$tilgung_n = \begin{cases} rate - zinsen_n & \text{falls } rate - zinsen_n \leq rest_n \\ rest_n & \text{falls } rate - zinsen_n > rest_n \end{cases}$$

Wir haben noch die jährlich vereinbarten Sonderzahlungen zu berücksichtigen, die maximal in Höhe der nach der Tilgung noch bestehenden Restschuld erfolgen:

$$sonderz_n = \begin{cases} sondertilgung & \text{falls } n \text{ durch 12 teilbar und} \\ & sondertilgung < rest_n - tilgung_n \\ rest_n - tilgung_n & \text{falls } n \text{ durch 12 teilbar und} \\ & sondertilgung \geq rest_n - tilgung_n \\ 0 & \text{falls } n \text{ nicht durch 12 teilbar} \end{cases}$$

Insgesamt ergibt sich nach Abzug aller Zahlungen der neue Darlehensrest:

$$rest_{n+1} = rest_n - tilgung_n - sonderz_n$$



Notizen

# Berechnung des Tilgungsplans (1)

Hochschule RheinMain

Hauptprogramm mit Variablendefinitionen, Einlesen der Daten und Hauptverarbeitungsschleife

```
void main()
{
    float rest, rate, zins1, zins2, sondertilgung;
    int bindung;
    int monat;
    float zins, zinsen, tilgung, sonderz;
    printf( "Darlehen: ");
    scanf( "%f", &rest);
    printf( "Nominalzins: ");
    scanf( "%f", &zins1);
    printf( "Monatsrate: ");
    scanf( "%f", &rate);
    printf( "Zinsbindung (Jahre): ");
    scanf( "%d", &bindung);
    printf( "Zinssatz nach Bindung: ");
    scanf( "%f", &zins2);
    printf( "Jaehrliche Sondertilgung: ");
    scanf( "%f", &sondertilgung);
    printf( "\nTilgungsplan:\n\n");
    printf( "Monat Zinssatz Zinsen Tilgung Sondertilg Rest\n");
    for( monat = 1; rest > 0; monat = monat + 1)
    {
        // Hier wird eine Zeile des Tilgungsplans (siehe nächste Seite) berechnet
    }
}
```

Die Schleife wird ausgeführt, solange noch eine Restschuld besteht.



Notizen



# Berechnung des Tilgungsplans (2)

Hochschule RheinMain

Inhalt der Hauptverarbeitungsschleife, Berechnung und Ausgabe der Daten für einen Monat

```
printf( "%5d", monat);

if( monat <= bindung * 12)
    zins = zins1;
else
    zins = zins2;
printf( " %10.2f", zins);

zinsen = rest * zins / 1200;
printf( " %10.2f", zinsen);

tilgung = rate - zinsen;
if( tilgung > rest)
    tilgung = rest;
printf( " %10.2f", tilgung);

rest = rest - tilgung;
sonderz = 0;
if( (monat % 12) == 0)
{
    sonderz = sondertilgung;
    if( sonderz > rest)
        sonderz = rest;
}
printf( " %10.2f", sonderz);

rest = rest - sonderz;
printf( " %10.2f", rest);
printf( "\n");
```

$$zins_n = \begin{cases} zins_1 & \text{falls } n \leq bindung \cdot 12 \\ zins_2 & \text{falls } n > bindung \cdot 12 \end{cases}$$

$$zinsen_n = \frac{rest_n \cdot zins_n}{12 \cdot 100}$$

$$tilgung_n = \begin{cases} rate - zinsen_n & \text{falls } rate - zinsen_n \leq rest_n \\ rest_n & \text{falls } rate - zinsen_n > rest_n \end{cases}$$

$$sonderz_n = \begin{cases} sondertilgung & \text{falls } n \text{ durch 12 teilbar und} \\ & sondertilgung < rest_n - tilgung_n \\ rest_n - tilgung_n & \text{falls } n \text{ durch 12 teilbar und} \\ & sondertilgung \geq rest_n - tilgung_n \\ 0 & \text{falls } n \text{ nicht durch 12 teilbar} \end{cases}$$

$$rest_{n+1} = rest_n - tilgung_n - sonderz_n$$



Notizen

# Der fertige Tilgungsplan

Hochschule RheinMain

**Eingabe:**

```
Darlehen: 100000
Nominalzins: 6.5
Monatsrate: 3000
Zinsbindung (Jahre): 1
Zinssatz nach Bindung: 8.0
Jaehrliche Sondertilgung: 10000
```

**Ausgabe:**

Tilgungsplan:

| Monat | Zinssatz | Zinsen | Tilgung | Sondertilg | Rest |
|---|---|---|---|---|---|
| 1 | 6.50 | 541.67 | 2458.33 | 0.00 | 97541.66 |
| 2 | 6.50 | 528.35 | 2471.65 | 0.00 | 95070.02 |
| 3 | 6.50 | 514.96 | 2485.04 | 0.00 | 92584.98 |
| 4 | 6.50 | 501.50 | 2498.50 | 0.00 | 90086.48 |
| 5 | 6.50 | 487.97 | 2512.03 | 0.00 | 87574.45 |
| 6 | 6.50 | 474.36 | 2525.64 | 0.00 | 85048.80 |
| 7 | 6.50 | 460.68 | 2539.32 | 0.00 | 82509.48 |
| 8 | 6.50 | 446.93 | 2553.07 | 0.00 | 79956.41 |
| 9 | 6.50 | 433.10 | 2566.90 | 0.00 | 77389.51 |
| 10 | 6.50 | 419.19 | 2580.81 | 0.00 | 74808.70 |
| 11 | 6.50 | 405.21 | 2594.79 | 0.00 | 72213.91 |
| 12 | 6.50 | 391.16 | 2608.84 | 10000.00 | 59605.07 |
| 13 | 8.00 | 397.37 | 2602.63 | 0.00 | 57002.44 |
| 14 | 8.00 | 380.02 | 2619.98 | 0.00 | 54382.45 |
| 15 | 8.00 | 362.55 | 2637.45 | 0.00 | 51745.00 |
| 16 | 8.00 | 344.97 | 2655.03 | 0.00 | 49089.97 |
| 17 | 8.00 | 327.27 | 2672.73 | 0.00 | 46417.23 |
| 18 | 8.00 | 309.45 | 2690.55 | 0.00 | 43726.68 |
| 19 | 8.00 | 291.51 | 2708.49 | 0.00 | 41018.20 |
| 20 | 8.00 | 273.45 | 2726.55 | 0.00 | 38291.65 |
| 21 | 8.00 | 255.28 | 2744.72 | 0.00 | 35546.93 |
| 22 | 8.00 | 236.98 | 2763.02 | 0.00 | 32783.91 |
| 23 | 8.00 | 218.56 | 2781.44 | 0.00 | 30002.46 |
| 24 | 8.00 | 200.02 | 2799.98 | 10000.00 | 17202.48 |
| 25 | 8.00 | 114.68 | 2885.32 | 0.00 | 14317.16 |
| 26 | 8.00 | 95.45 | 2904.55 | 0.00 | 11412.61 |
| 27 | 8.00 | 76.08 | 2923.92 | 0.00 | 8488.70 |
| 28 | 8.00 | 56.59 | 2943.41 | 0.00 | 5545.29 |
| 29 | 8.00 | 36.97 | 2963.03 | 0.00 | 2582.26 |
| 30 | 8.00 | 17.22 | 2582.26 | 0.00 | 0.00 |



Notizen

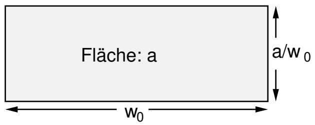


# Algorithmisierung der Wurzelberechnung

Hochschule RheinMain

Aufgabe: Lösen der Gleichung $w^2 = a$ für $a > 0$ (Berechung von $w = \sqrt{a}$)
Geometrische Interpretation: Gesucht ist ein Quadrat der Kantenlänge $w$, dessen Fläche $a$ ist.

Erste (willkürliche) Näherung: $w_0 = a$
Wenn wir $w_0$ als eine Seitenlänge eines Rechtecks auffassen, das die Fläche $a$ haben soll, so müssen wir $\frac{a}{w_0}$ als Länge der anderen Seite wählen.




Das ist, außer für $a = 1$, eine ungenügende Annäherung an ein Quadrat, aber wenn wir im nächsten Schritt den Mittelwert aus den beiden Kantenlängen wählen, wird das Rechteck schon deutlich quadratischer:

$$w_1 = \frac{1}{2}\left(w_0 + \frac{a}{w_0}\right)$$

Wir bestimmen zu $w_1$ wieder die die Länge der zweiten Seite ($\frac{a}{w_1}$) und fahren mit der Mittelwertbildung fort:

$$w_2 = \frac{1}{2}\left(w_1 + \frac{a}{w_1}\right)$$

Allgemein:

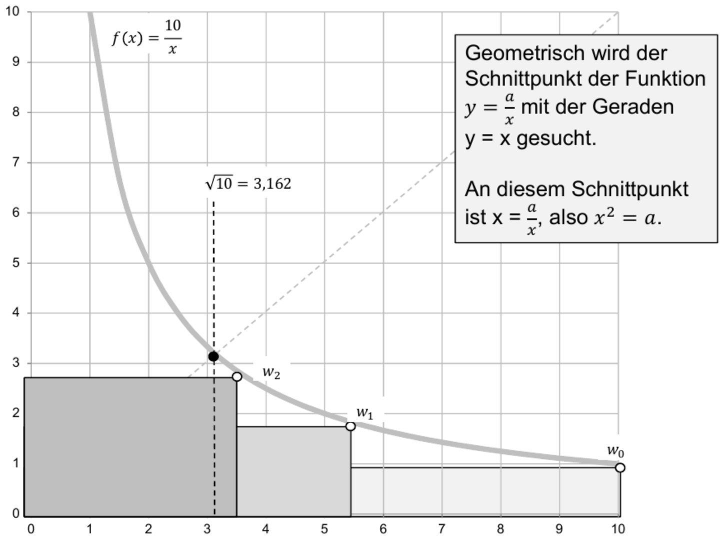
$$w_n = \frac{1}{2}\left(w_{n-1} + \frac{a}{w_{n-1}}\right)$$



Notizen

# Geometrie des Verfahrens

Hochschule RheinMain






Notizen



# Programmierung des Iterationsverfahrens

Hochschule RheinMain

```
void main()
{
    float a, w;
    int i;
    printf( "Bitte Zahl eingeben: ");

    scanf( "%f", &a);
    w = a;
    for( i = 0; i < 10; i++)
    {
        w = (w + a/w)/2;
        printf( "%f\n", w);
    }
}
```

$$w_n = \begin{cases} a & \text{falls } n = 0 \\ \frac{1}{2}\left(w_{n-1} + \frac{a}{w_{n-1}}\right) & \text{falls } n = 1, 2, 3, \ldots \end{cases}$$

**Ausgabe:**

```
Bitte Zahl eingeben: 10
5.500000
3.659091
3.196005
3.162456
3.162278
3.162278
3.162278
3.162278
3.162278
3.162278
```

**Fragen:**

Strebt die Folge immer gegen den gesuchten Wert?
Wann ist der berechnete Wert genau genug, um aufhören zu können?



Notizen

# Konvergenz des Verfahrens

Hochschule RheinMain

Die Mathematik sagt, dass die Folge $w_n$ für $n \geq 1$ monoton fallend ist und gegen $\sqrt{a}$ konvergiert. Insbesondere ist $(w_n)^2 \geq a$ für $n \geq 1$.
Damit kann man das Verfahren abbrechen, sobald eine vorgegebene Genauigkeit (z.B. 0.001) erstmals erreicht ist:

```
void main()
{
    float a, w;
    int i;
    printf( "Bitte Zahl eingeben: ");

    scanf( "%f", &a);
    w = a;
    for( i = 0; i < 10; i++)
    {
        w = (w + a/w)/2;
        printf( "%f\n", w);
        if(w*w - a < 0.001)
            break;
    }
}
```

Abbruchkriterium

**Ausgabe:**

```
Bitte Zahl eingeben: 10
5.500000
3.659091
3.196005
3.162456
3.162278
```

Ohne flankierende mathematische Überlegungen können Sie nicht sicher sein, dass dieses Programm korrekt arbeitet.
Dieses Verfahren wurde übrigens nicht von Informatikern erfunden. Das Verfahren war in Mesopotamien bereits 1750 v. Chr. bekannt. Um 100 n. Chr. wurde es von dem griechischen Mathematiker Heron von Alexandria beschrieben und daher auch *Heron-Verfahren* genannt.



Notizen



# Zusammenfassung

Hochschule RheinMain

In diesem Kapitel wurden Vorgehensweisen zur Überführung Arithmetischer Ausdrücke in Computerprogramme gezeigt
**Mathematik ist die wichtigste Grundlage der Informatik.**



Notizen