

Aufgabe 5 Informatik 1 WiSe 2024

Leistung Wasserkraftwerk - Wasserkraft-Rechner

Lernziel Informatik: Erstellen von Funktionen

In dieser Aufgabe geht es um die Berechnung der maximalen Leistung eines Wasserkraftwerks an einem Stausee (Talsperre) in Abhängigkeit von der Höhe des Wasserstands.



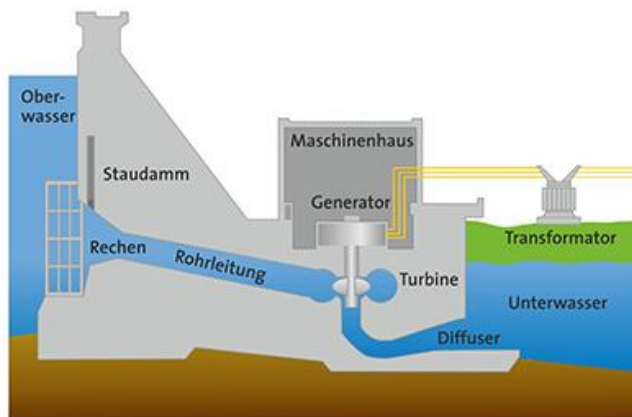
Das größte Wasserkraftwerk der Welt liegt in Itaipú, an der Grenze zwischen Brasilien und Paraguay.

Aus einem Stausee, der mehr als doppelt so groß ist wie der Bodensee zwischen Deutschland und der Schweiz, stürzt das Wasser aus 118 Metern Höhe durch zwanzig Rohre nach unten.

Dabei wird so viel Strom erzeugt, wie ihn in derselben Zeit zwölf Atomkraftwerke bereitstellen könnten.

Für die Berechnung der Leistung eines solchen Kraftwerks ist es unerheblich, ob das Wasser durch Rohre nach unten ‚stürzt‘ oder ob es unten am Becken entnommen wird. Relevant ist vor allem die Anzahl der Turbinen, der Volumenstrom und die Höhe des Wasserpegels über der Turbinenhöhe (Pegel Oberwasser).

Für diese Aufgabe betrachten wir vereinfacht ein Wasserkraftwerk mit nur einer Turbine:



Zur Berechnung der Leistung einer Turbine in einem Wasserkraftwerk aus Höhe und Volumenstrom:

- Der Volumenstrom gibt an, welche Durchflussmenge pro Sekunde die Turbine antreibt.
- Der Volumenstrom Q kann aus der Querschnittsfläche A des jeweiligen Wasserzulaufs (Rohrleitung) und der Fließgeschwindigkeit v des Wassers berechnet werden:
- $Q = A \cdot v$

Aber wie kommt man nun auf v ?

Ein primitiver Ansatz: Wenn man einen Behälter mit Flüssigkeit darin und einem kleinen Loch unten hat, dann entspricht die Geschwindigkeit des austretenden Strahls in etwa derjenigen, die ein Körper hätte, der im freien Fall den Höhenunterschied h von der Flüssigkeitsoberfläche bis zur Austrittsöffnung zurückgelegt hat. Es ergibt sich somit:

$$v = \sqrt{2 \cdot g \cdot h}$$

mit der Erdanziehungskraft $g = 9,81 \text{ m/s}^2$.

Zum Nachvollziehen der Formel lohnt sich ein Blick auf:

<https://studyflix.de/ingenieurwissenschaften/freier-fall-1354>

Diese Formel können wir nutzen, wobei h hier die Höhe des Wasserpegels im Staubecken ist. Für den **Volumenstrom Q** ergibt sich dann:

$$Q = A \cdot \sqrt{2 \cdot g \cdot h}$$

Die Formel für die Berechnung der **Leistung P** ist:

$$P = Q \cdot g \cdot h \cdot w$$

mit dem Wirkungsgrad $w = 0,85$ (85% Wirkungsgrad gewählt für diese Aufgabe).

Aufgabe 5.1 a)

Schreiben Sie eine Funktion für die Berechnung des Volumenstroms Q .

Übergabewerte (Parameter) sollen sein:

- Höhe des Wasserpegels h
- Querschnittsfläche des Wasserzulaufs A

Rückgabewert soll sein:

- Volumenstrom Q

Wählen Sie selbst sinnvolle Bezeichner (Variablennamen) und Datentypen.

Aufgabe 5.1 b)

Schreiben Sie eine Funktion für die Berechnung der Leistung P.

Übergabewerte (Parameter) sollen sein:

- Höhe des Wasserpegels h
- Querschnittsfläche des Wasserzulaufs A

Rückgabewert soll sein:

- Leistung P

Wählen Sie selbst sinnvolle Bezeichner (Variablennamen) und Datentypen.

Hinweis: Diese Funktion soll die Funktion aus a) aufrufen, um Q zu berechnen.

Aufgabe 5.1 c)

Sie sollen nun die Werte für verschiedene Pegelstände zwischen 0 und 100 Metern berechnen und ausgeben.

In Ihrem Hauptprogramm (main) sollte folgendes passieren:

- Deklaration von Variablen für Volumenstrom und Leistung etc.
- in 10-Meter-Schritten in einer Schleife jeweils Volumenstrom und Leistung mit Hilfe der Funktionen aus a) und b) berechnen, der entsprechenden Variablen zuweisen und die beiden Werte mit printf ausgeben (mit vorangestellter Meterzahl s.u.).

Hierbei nehmen wir für die Querschnittsfläche **A einen Wert von 4** (m²) an.

Das Ergebnis müsste in etwa so aussehen:

| | | |
|-----|-----------|--------------|
| 0 | 0.000000 | 0.000000 |
| 10 | 56.028564 | 4671.941895 |
| 20 | 79.236359 | 13214.248047 |
| 30 | 97.044319 | 24276.123047 |
| ... | | |
| ... | | |
| ... | | |

Aufgabe 5.2 (Achtung, dieser Aufgabenteil wird noch ergänzt...):

Die Ausgabe der Werte für q und p soll nun nicht mehr in Tabellenform erfolgen. Statt dessen soll eine grafische Visualisierung umgesetzt werden. Das sollte wieder mit Codeblocks und einem Grafikfenster mit z.B. 500x500 Pixeln (wie in Aufgabe 4.4) geschehen.

Wie in CodeBlocks ein **grafikfähiges Projekt** erstellt wird, entnehmen Sie bitte der aktuellen Version des Dokuments **codeblocks-windows.pdf** (LEA-Kurs, Ordner ‚Praktikum‘) im Abschnitt „Erstellung eines grafikfähigen Projekts in CodeBlocks“. Diese Anleitung gilt auch für Mac OS mit PlayForMac.

Als Grundlage nutzen Sie Ihre Lösung aus Aufgabe 5.1 c), die in das leere main.c in dem neuen CodeBlocks-Projekt kopiert werden muss (Copy-Paste).

Unter folgendem Link finden Sie eine Auflistung der in der **Borland Graphics Library** verfügbaren **Funktionen**:

http://winbgim.codecutter.org/V6_0/doc/index.html

Sie können gerne kreativ sein – für diese Aufgabe sind aber wahrscheinlich am einfachsten zu nutzen:

- `void moveto (int x, int y);`
- `void lineto (int x, int y);`

Die einfachste Herangehensweise besteht dann darin (siehe Abbildung unten),

- den Wert von **h (Höhe)** für die **X-Achse** zu nehmen.
- Die Werte von **q und p** jeweils auf der **Y-Achse** abzubilden.

Dafür:

- im Programm **vor der Schleife**, welche die verschiedenen Höhen durchläuft, zunächst mit ***moveto(x, y);*** einen Startpunkt im Grafikfenster zu setzen. Hierbei wird noch nichts gezeichnet.
- **in der Schleife** jeweils mit ***lineto(x, y);*** eine Linie zum nächsten Punkt des Graphen zu zeichnen.

Die Herausforderung besteht hierbei vor allem darin, die Werte von h, q und p jeweils so umzurechnen, dass sie in das Grafikfenster mit 500x500 Pixeln passen.

Außerdem sollte bedacht werden, dass in einem normalen Liniendiagramm der Y-Wert 0 unten ist – im Grafikfenster befindet er sich jedoch oben.

Tipps:

- Für die Darstellung von zwei Kurven (für q und p) in einem (Linien-)Diagramm werden wahrscheinlich zwei Schleifen benötigt.
- Schön wäre natürlich die Darstellung in zwei verschiedenen Farben – schauen Sie hierzu in die Dokumentation der BGI (Link s.o.) bei der Funktion ***setcolor(int color);***

Denken Sie daran (wie in Aufgabe 4.4), dass:

- oben in main.c mit ***#include <graphics.h>*** die Grafik-Bibliothek eingebunden werden muss.
- im main-Anweisungsblock nach der Deklaration der Variablen der einmalige Aufruf von ***initwindow(500, 500);*** durchgeführt werden muss.

Eine korrekte Visualisierung könnte z.B. so aussehen:

