

## Programmierung von CAx-Systemen, Modularbeit, Aufgabenteil C++

Sehr geehrte Studierende,

laden Sie den folgenden Artikel von arXiv.org herunter und versuchen Sie, den C++-Quelltext zur Simulation einer Wellenmaschine zu verstehen. Hinweise: Die Kugel am linken Rand der Wellenmaschine ( $n = 1$ ) wird im abgebildeten C++-Programm sinusförmig angeregt; auf die Kugel am rechten Rand ( $n = N$ ) wirkt geschwindigkeitsproportionale Reibung (Reibungskoeffizient  $\alpha$ ). Alle Parameter der Wellenbahn sind fest im abgebildeten C++-Quelltext hinterlegt, das Programm verfügt über keine grafische Benutzeroberfläche:

<https://arxiv.org/pdf/1711.00717.pdf>

Bitte erstellen Sie im Rahmen dieser Modularbeit eine Dialog-Applikation auf Basis von Qt-Widgets zur Simulation einer solchen Wellenmaschine. Über die grafische Benutzeroberfläche sollen die folgenden Parameter einstellbar sein:

1. Die Art der Anregung am linken Rand der Wellenmaschine: Der Anwender soll eine von mindestens drei Alternativen auswählen können (sinus- oder rechteckförmige Anregung, Einheitssprung...).
2. Die Lagerung der Kugel am rechten Rand der Wellenmaschine: Der Anwender soll eine von mindestens zwei Alternativen auswählen können (mit „angepasster“ Reibung – so wie im Artikel gezeigt – oder mit dem Reibungskoeffizienten  $\alpha = 0$ ...).
3. Ein weiterer Simulationsparameter, den Sie sich selbst aussuchen können, zusätzlich zu den bereits genannten Punkten (1) und (2).

Wenn der Anwender in Ihrer Dialog-Applikation den „Start-Knopf“ drückt, soll die animierte Wellenmaschine in einem Chart-Fenster erscheinen – genauso wie bei dem Programm aus dem arXiv-Artikel. Sie dürfen den C++-Quelltext aus dem oben genannten Artikel verwenden. Die Softwarebibliotheken unter <https://kuepper.userweb.mwn.de/software.htm> stehen Ihnen ebenfalls zur Verfügung.

Geben Sie spätestens am 27.07.2020 die folgenden zwei Dateien ab:

1. Das von Ihnen erstellte Qt-Creator-Projekt: Bitte packen Sie dazu den kompletten Projektordner in eine einzelne ZIP-Datei.
2. Eine kurze Ausarbeitung (max. 5 Seiten). Geben Sie die Ausarbeitung als einzelne PDF-Datei ab, verwenden Sie kein anderes Dateiformat. Ihre Ausarbeitung soll die folgenden Inhalte umfassen:
  - Die folgende Bestätigung mit Datum und (eingescannter) Unterschrift: „Hiermit wird erklärt, dass die vorliegende Ausarbeitung selbständig verfasst wurde. Es sind keine anderen als die angegebenen Quellen oder Hilfsmittel verwendet und wörtliche sowie sinngemäße Zitate als solche gekennzeichnet worden.“
  - Eine kurze Bedienungsanleitung zu der von Ihnen erstellten Benutzeroberfläche.
  - Einige Bildschirmfotos von Ihrem laufenden Programm. Es soll die Benutzeroberfläche mit einigen verschiedenen Einstellungen und den dazugehörigen grafischen Ausgaben der Wellenbahn zu erkennen sein.

München, 20.07.2020



### Tipps:

- Legen Sie ein neues Qt-Widgets Projekt für eine Dialog-Applikation an.
- Fügen Sie alle benötigten Bibliotheken (HMChart, HMMatrix, HMDGL) zum Projekt hinzu.
- Nicht vergessen: Für HMChart muss eine Zeile („LIBS += ...“) zur Projektdatei hinzugefügt werden.
- Fügen Sie eine neue Schaltfläche (QPushButton) zum Dialogfenster hinzu und überprüfen Sie, dass der Qt Creator einen passenden Slot in der Datei Dialog.cpp angelegt hat.
- Kopieren Sie den kompletten C++-Quelltext aus dem oben angegebenen Artikel – so wie er ist – nach Dialog.cpp: und zwar unmittelbar vor (!) den gerade angelegten QPushButton-Slot.
- Im arXiv-Artikel gibt es ein Hauptprogramm main(). Die Befehle im Hauptprogramm main() verschieben Sie alle in den QPushButton-Slot. Die – jetzt leere – Funktion main() wird entfernt.
- Nun können Sie das Programm ohne Fehler kompilieren und ausführen. Über die Schaltfläche lässt sich die Simulation der Wellenmaschine starten.
- Aus den Konstanten zu Beginn des Quelltextes machen Sie Variablen: „const auto“ durch „double“ ersetzen (bzw. durch „int“ bei der Kugelanzahl N). Jetzt lassen sich die Werte dieser Parameter während des laufenden Programms verändern – was ja in der Aufgabenstellung gefordert ist...
- Der C++-Quelltext aus dem arXiv-Artikel ist auf der folgenden Seite abgedruckt, damit Sie ihn einfacher kopieren können.
- Viel Erfolg!

```

// Simulation einer Wellenmaschine in C++
#define _USE_MATH_DEFINES

#include <math.h>
#include "chart.h"
#include "hmdgl.hpp"
#include "matrix.hpp"

using namespace std;
using namespace HMDGL;
using namespace HMMatrix;

const auto N      = 20;    // Kugelanzahl
const auto MASSE  = 0.01; // Kugelmasse
const auto FEDER  = 1.0;  // Federkonstante
const auto FREQ   = 0.5;  // Anregungsfrequenz
const auto AMPL   = 1.0;  // Anregungsamplitude
const auto REIB   = sqrt(FEDER * MASSE); // Reibung

Matrix M(2 * N);
Vector b(2 * N);

// DGL-System der Wellenmaschine
Vector sys(double t, Vector u)
{
    double omega = 2 * M_PI * FREQ;
    return M * u + b * AMPL * sin(omega * t);
}

// Nach jedem Simulationsschritt: Kugelpositionen darstellen
void step(double t, Vector u)
{
    chart_line_series (u.Slice(0, N), CHART_BLACK);
    chart_point_series(u.Slice(0, N), CHART_RED );
    chart_end_of_frame();
}

int main()
{
    // Matrix M belegen
    for(int i = 0; i < N; ++i)
    {
        M(i, N + i) = 1;
        M(N + i, i) = -2 * FEDER / MASSE;
        if(i > 0)    M(N + i, i - 1) = FEDER / MASSE;
        if(i < N - 1) M(N + i, i + 1) = FEDER / MASSE;
    }
    M(2 * N - 1, N - 1) = -FEDER / MASSE;
    M(2 * N - 1, 2 * N - 1) = -REIB / MASSE;

    // Vektor b belegen
    b(N) = FEDER / MASSE;

    // DGL-System integrieren
    Vector init(2 * N);
    dgl_rk4(0, 100, 2000, sys, step, init);

    // Animation der Simulationsergebnisse
    chart_frame_timer(100);
    chart_show(CHART_NOBACKGROUND, "Wellenmaschine");
}

```