

Programmieren für Physiker

Interfakultatives Institut für Anwendungen der Informatik
Institut für Theoretische Teilchenphysik

Prof. Dr. M. Steinhauser, Dr. A. Mildenerger
<http://comp.physik.kit.edu>

SS 2011 – Blatt 06
Bearbeitungszeitraum: bis 25. Mai 2011

Aufgabe 15: Polynom-Interpolation

Pflichtaufgabe

Bei numerischen Berechnungen tritt häufig der Wunsch nach Funktionswerten von Funktionen auf, von denen nur einige Punkte bekannt sind. Dann müssen, mittels einer Modellannahme über die Funktion, die neu zu berechnenden Werte inter- oder extrapoliert werden. Ein Schema, das Sie in der Vorlesung kennengelernt haben, besteht darin, für $n + 1$ gegebene Punkte $(x_i, f(x_i))$, $i = 1, \dots, n + 1$ ein Polynom vom Grad n zu bestimmen, und dieses Polynom für Vorhersagen zu verwenden.

Schreiben Sie ein C++-Programm, das mittels des Algorithmus von Neville eine Polynom-Interpolation durchführt. In der Datei `/home/ck11/daten/a15-interpol.dat` oder auf der WWW-Seite finden Sie einen Datensatz. Verwenden Sie eine `while (datei >> ...)` Schleife, um aus der Datei Wertepaare $(x_i, f(x_i))$ in zwei Felder einzulesen, bis das Ende der Datei erreicht ist. Der Benutzer soll nun die Möglichkeit haben, x -Werte einzugeben, für die der interpolierte Funktionswert berechnet wird. Dies soll so lange geschehen, bis der Benutzer den Wert $x = 0$ eingibt. Geben Sie dabei jeden interpolierten Wert auf dem Bildschirm aus und schreiben Sie die Werte x und $f(x)$ zeilenweise in eine Ergebnisdatei `a15-interpol-res.dat`.

Die ursprüngliche Punktedatei und die von Ihnen erzeugte Ergebnisdatei schauen Sie sich bitte in einem geeigneten Plot-Programm (z.B. `xmgrace` oder `gnuplot`) an. Stellen Sie bitte die Ausgabe so ein, dass die gegebenen und die neu berechneten Funktionswerte durch Punkte und verschiedene Farben erkennbar sind.

Aufgabe 16: magische Quadrate

Pflichtaufgabe

Eine quadratische Anordnung der Zahlen $1, \dots, n^2$ heißt *magisches Quadrat* der Größe n , wenn folgende Bedingung erfüllt ist: Sämtliche Zeilensummen und sämtliche Spaltensummen und auch beide Summen über die Diagonalen sind identisch. Beispiel für $n = 3$:

8	1	6
3	5	7
4	9	2

Mit folgendem Verfahren läßt sich für ungerade n ein magisches Quadrat erzeugen. Es werden sukzessive die Zahlen $1, \dots, n^2$ eingetragen:

- Starten Sie den Auffüllvorgang beim mittleren Feld der obersten Zeile. Tragen Sie hier die 1 ein.
- Wiederholen Sie den folgenden Schritt, bis nacheinander alle Zahlen eingetragen sind: Ist das Feld diagonal rechts oben vom aktuellen Feld aus noch frei?
 - Ja: Tragen Sie hier die nächste Zahl ein. Gehen Sie nun von diesem Feld aus weiter.
 - Nein: Wählen als neues Feld den Platz unterhalb des aktuellen Feldes. Tragen Sie hier die nächste Zahl ein. (Dieser Platz sollte noch frei sein – ansonsten ist ein Fehler im Laufe des Verfahrens passiert.)

Bei den Schritten nach rechts oben kann es vorkommen, dass Sie sich über den rechten oder den oberen Rand des Quadrats hinaus bewegen. In so einem Falle springen Sie in das Kästchen am entsprechenden gegenüberliegenden Rand des Quadrats. (Eventuell müssen Sie über beide Ränder gleichzeitig hüpfen.)

Schreiben Sie ein Programm, das ein magisches Quadrat in einem zweidimensionalen Feld erstellt und ausgibt, dabei ist die ungerade Größe n vorher einzugeben.

Zusatz (freiwillig): Welchen Summenwert erwarten Sie in jeder Zeile, Spalte und Diagonalen? Überlegen Sie sich einen analytischen Ausdruck. Implementieren Sie die beiden Diagonalsummen in Ihrem Programm und testen Sie, ob sie den erwarteten Wert liefern.

Zum Knobeln: Nim-Spiel

freiwillig

Beim Nim-Spiel in der hier betrachteten Variante gibt es zwei Spieler und zwei Häufen mit Streichhölzern auf dem Tisch. Die Spieler sind abwechselnd am Zug und haben dabei jeweils folgende zwei Möglichkeiten: Entweder von genau einem Haufen ihrer Wahl eine beliebige Zahl (≥ 1) an Hölzchen zu entfernen, oder aber von beiden Häufen genau die gleiche Zahl an Hölzchen zu entfernen. In jedem Zug muss also immer mindestens ein Streichholz weggenommen werden. Wer mit seinem Zug das letzte Streichholz (oder die letzten Streichhölzer) wegnehmen kann, hat gewonnen.

Schreiben Sie zunächst ein C++-Programm, welches Sie dieses Spiel mit zwei Spielern spielen lässt. In diesem Fall soll das Programm also die Streichholzhäufen verwalten und dabei von beiden Spielern abwechselnd gültige Züge erfragen, bis ein Spieler gewonnen hat.

Interessant ist es, sich zu überlegen, welche Strategie in diesem Spiel zum Gewinn führt. Gibt es Haufengrößen, von denen aus der Sieg erzwungen werden kann?

Ersetzen Sie in Ihrem Programm einen Mitspieler durch den Computer und bringen Sie ihm Ihre Gewinnstrategie bei.

Hinweis: Es ist zweckmäßig, mit zwei zufälligen Stapeln von je etwa 5 bis 25 Streichhölzern zu beginnen. Um dies zu erreichen, können Sie vom Rechner erzeugte Zufallszahlen verwenden. Nach der Anweisung `#include <cstdlib>` im Kopf des Programms können Sie mit `rand()%a` ganzzahlige Zufallszahlen im Intervall $[0, a[$ erzeugen. Initialisieren Sie den Generator durch einmaliges `srand((unsigned int)time(0))` am Anfang des Programms.
