

Universität Osnabrück Fachbereich Physik	Numerische Physik WS 2004/2005	Apl. Prof. Dr. Jürgen Schnack Dipl.-Phys. Matthias Exler
---	-----------------------------------	---

Aufgabenblatt 1

1.1 Fortran unter UNIX

- Loggen Sie sich auf luna.rz.uni-osnabrueck.de mit Ihrem Rechenzentrumsaccount (wie bei der Email) ein.
- Legen Sie ein Verzeichnis für Ihre Fortranprogramme an, z.B. NumPhys.
- Schreiben Sie ein dreizeiliges Fortran-programm, das die Zeile „Hello, World!“ ausgibt. Der Fortran-Compiler heißt xlf.
- Schreiben Sie ein Programm, das die Primzahlen bis 100 berechnet. Beschreiben Sie die von Ihnen verwendete Methode in Worten.
- Editieren Sie ein File mit dem Namen `Makefile`, z.B. wie folgt

```
1  FC = xlf
2  FFLAGS = -O
3
4  OBJSO1 = HelloWorld.o
5  OBJSO2 = prim.o
6
7
8  BINARIES = HelloWorld prim
9
10 all : HelloWorld prim
11
12 clean:
13     rm $(BINARIES) *.o
14
15
16
17 HelloWorld : $(OBJSO1)
18     $(FC) -o HelloWorld $(OBJSO1) $(LIBS)
19
20 prim : $(OBJSO2)
21     $(FC) -o prim $(OBJSO2) $(LIBS)
```

Dieses Makefile können Sie für das Compilieren und Linken Ihrer Programme verwenden, der Aufruf lautet einfach `make`.

1.2 Mathematica und matlab

Rufen Sie die beiden Programme Mathematica und matlab auf und informieren Sie sich in der Hilfe

- wie man die Sinusfunktion im Intervall $[0, 2\pi]$ graphisch darstellt;
- wie man eine Liste von Wertepaaren eingibt und graphisch darstellt.

1.3 Numerische Genauigkeit

Bearbeiten Sie das folgende Problem sowohl mit Mathematica als auch mit matlab. Kommentieren Sie Ihre Notebooks!!!

Die Zahl e kann durch die Folge

$$e_k = \left(1 + \frac{1}{k}\right)^k \quad (1)$$

approximiert werden. Wie genau kann man bei diesem Verfahren realistischerweise werden?

- Legen Sie ein Feld **Powers** mit natürlichen Zahlen von 1 bis n_{max} an.
- Bilden Sie ein zweites Feld **EApprox** mit Werten e_k , wobei $k = 10^n$ und geben Sie die erhaltenen Werte aus.
- Bilden Sie ein drittes Feld **RelativeFehler** mit den relativen Abweichungen von e und geben Sie auch dieses aus.
- Stellen Sie die Beträge der relativen Fehler in einem halblogarithmischen Plot als Funktion von n , d.h. **Powers** dar.

Hinweise:

Arbeiten Sie in matlab mit reellen Zahlen und in mathematica mit natürlichen Zahlen. In matlab können Sie problemlos bis $n_{max} = 12$ gehen. Wie weit kommen Sie mit mathematica?

Zusatzaufgabe:

- Die Standardplots sind einfach unbrauchbar. Machen Sie was Vernünftiges daraus, also: Achsenbeschriftung, größere Schrift, richtige Datenpunkte statt „Fliegenschiff“.
- Programmieren Sie alternativ ein mathematica-notebook mit reellen Zahlen.
- Wenn Sie noch mehr tun wollen, können Sie das Problem noch in FORTRAN lösen, allerdings sollte sich das Ergebnis nicht von dem mit matlab erhaltenen unterscheiden. Sie können allerdings ausprobieren, was passiert, wenn Sie statt **real*8** nur **real*4** nehmen.

1.4 Funktionen und Ihre Darstellung

Portieren Sie die in der Vorlesung mit Mathematica vorgestellten Lösungen nach Matlab. Wenn nötig, können Sie die Mathematica-Notebooks von meiner Webseite herunterladen. Lösen Sie insbesondere die folgenden Aufgaben:

- a. Stellen Sie die Funktionen $f(x) = 2 \sin(4x)$ und $g(x, y) = \sin(x) \cos(y)$ in ein- bzw. zweidimensionalen Plots dar. Der Definitionsbereich der einzelnen Variablen soll dabei stets von Null bis 2π gehen. Gestalten Sie die Kurven farbig und fügen Sie Achsenbeschriftungen und Titel an.
- b. Finden Sie heraus, wie sie mit Matlab mehrer Funktionen in einen Plot zeichnen können und erstellen Sie eine Legende.
- c. Kann man mit Matlab einen Zeitverlauf zeigen und evtl. sogar ein animiertes Gif davon produzieren?
- d. Finden Sie heraus, wie man mit Matlab die Nullstelle von $\cos(x) - x$ bestimmen kann. Welches Verfahren wird verwendet?