

Rechenleistung bei Vektoroperationen

1. In numerischen Algorithmen oft benutzte Unterprogramme sind die Vektoroperationen:

```
s = scapr(n,X,Y)            $\sum x_i y_i$   
call vsaxpy(n,Z,X,alpha,Y)  $z_i := x_i + \alpha y_i$ 
```

mit einfacher Genauigkeit (REAL) aller Variablen

bzw. als DoublePrecision-Variante `dscapr`, `vdaxpy`.

Programmieren Sie diese 4 Unterprogramme und testen Sie diese zunächst an einfachen Beispielen, indem Sie die Vektoren im Hauptprogramm mit konkreten Werten belegen (nicht alle Komponenten eintippen!) – $x_i = f_x(i, n)$, $y_i = f_y(i, n)$.

Beachten Sie die Unterscheidung zwischen einfacher und doppelter Genauigkeit bei der Vereinbarung der Variablen bzw. Vektoren und der Namen der aufzurufenden Funktionsunterprogramme im Hauptprogramm.

2. Um die Rechenleistung zu messen, ruft man jeweils das Unterprogramm in einer Programmschleife (M -mal) auf. Dabei werden die Anzahl M und die Vektorlänge n „hinreichend“ groß gewählt, damit die Anzahl $2 \cdot n \cdot M$ der Gleitkommaoperationen (flops) wenigstens eine Sekunde Rechenzeit benötigt (Zeitmessung erfolgt außerhalb dieser Schleife, also für alle M Aufrufe des jeweiligen Unterprogramms zusammen).

Man darf etwa 50...2000 Mflops (Mio. Gleitkomma-Operationen pro Sekunde) erwarten, d.h. man sollte M in Abhängigkeit von n so wählen, dass $2 \cdot n \cdot M$ ca. 100 Mio. ist, um etwa 1 sec Rechenzeit zu erreichen (ggf. ist diese Zahl anzupassen).

Erläuterungen zur Zeitmessung in Fortran gibt es unter [Secnds.html](#).

Entsprechende Fortran90-Routinen zur Zeitmessung können ebenfalls genutzt werden, siehe [TimeDate.html](#).

Man stelle fest, ob signifikante Rechenzeit-Unterschiede zwischen einfach- und doppeltgenauer Rechnung bestehen.

Das Programm soll später dazu verwendet werden, eine Tabelle der Rechenleistung in Abhängigkeit von der Vektorlänge auszugeben.

3. Solche Vektoroperationen können eventuell noch beschleunigt werden, und zwar durch *unrolled loops*, d.h. die innere Schleife der Unterprogramme läuft nicht mit Schrittweite 1, sondern mit einer Schrittweite $\Delta i > 1$:

$$m = \text{mod}(n, \Delta i)$$

$$S = \sum_{i=1}^m x_i \cdot y_i$$

$$S = S + \sum_{i=m+1(\Delta i)}^n ((x_i \cdot y_i) + (x_{i+1} \cdot y_{i+1}) + \dots + (x_{i+\Delta i-1} \cdot y_{i+\Delta i-1}))$$

Analog kann man in jedem Schleifendurchlauf bei `vsaxpy` mehrere Komponenten $z_i, z_{i+1}, \dots, z_{i+\Delta i-1}$ berechnen, um die Anzahl der Schleifendurchläufe zu verringern.

Vergleichen Sie die Rechenzeiten für verschiedene Werte von Δi (z.B. $\Delta i = 3, 4, 5, 6, 7$).

4. Berechnung der Euklidischen Norm eines Vektors als Funktionsunterprogramm

$$\text{Xnorm} = \text{ENorm}(n, X) \quad \sqrt{\sum_{i=1}^n x_i^2}$$