Fortran (90):

implicit none - sonst werden alle Variablen, welche mit i,j,k,l,m oder n beginnen als **integer** und der Rest als **real** behandelt.

Variablen - **integer**(4) i,j , **real**(8) x,y , **complex**(16) c , **character**(12) **name** .

Arrays - **real**(8) A(4,4) Fortran ist Coloumn Major (im Speicher intern spaltenweise abgespeichert).

Zuweisung - i=2 , x=1.D2 (wäre x **real**(4), dann x=1.e2) , c=CMPLX(x,y) , **name**='...' .

Subroutinenaufruf - **call** subroutine1(parameters) , dabei Call By Reference (Variablenadressen werden übergeben).

Sequenzen - i=i+1 , x=**real**(c) (Re(c)) , y=aimag(c) (Im(c)) , abs(c) ($|c|$) , conjg(c) (c^*) , sqrt(c) ($\sqrt{c}$) , log(x) (ln(x)) , exp(x) (e^x) , x**y (x^y) .

If-Anweisung -
```
if (Bedingung) then
        Anweisungen
else
        Anweisungen
end
```

Bedingungen:

.lt.	<
.le.	≤
.gt.	>
.ge.	≥
.eq.	=
.ne.	≠
.and.	und
.or.	oder

Schleife -
```
do j= start , ende [,schrittweite]
        a(j) = ...
enddo
```
, die Schrittweitenangabe ist optional.

[ANSI-] C:

Präprozessor implementiert.

Variablen - **int** i,j; , **double** z; , **int*** k; [Pointer] , **char*** name; .

Arrays - Row Major (im Speicher intern zeilenweise abgespeichert).

Zuweisung - i = 1; , x=1.0e4 , k = &i , j = *k , name = "file.dat" .

Funktionen - Call By Value (es werden Kopien der übergebenen Variablen erstellt und die originalen unverändert gelassen).

Sequenzen - i=i+1 , ++i , z = (**float**) i (casten) , fabs(x) ($|x|$) , sqrt(x) ($\sqrt{x}$) , log(x) (ln(x)) , exp(x) (e^x) , power(x,y) (x^y).

Schleifen -
```
if (Bedingung) {
        Anweisungen;
}
else {
        Anweisungen;
}

while (Bedingung) {
        Anweisungen;
}
```

```
for (j=0; j<5; j++) {
        Anweisungen;
}
```

Bedingungen:

<	<
<=	≤
>	>
>=	≥
==	=
!=	≠
&&	und
||	oder

Programmierung: (1) Unstrukturierte Programmierung

(2) Strukturierte Programmierung

(3) Prozedurale Programmierung

(4) Modulare Programmierung

(5) Objektorientierte Programmierung

Grundregeln: 1. Korrektheit , 2. Übersichtlichkeit , 3. Effizienz .

Fourier:

$$F(\omega) = \mathrm{FT}\left(f(t)\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)\, \mathrm{e}^{-\mathrm{i}\,\omega t}\, \mathrm{d}t \qquad f(t) = \mathrm{FT}^{-1}\left(F(\omega)\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\omega)\, \mathrm{e}^{\mathrm{i}\,\omega t}\, \mathrm{d}\omega$$

$f \in \mathbb{R} \Rightarrow \quad F(\omega) = F^*(-\omega)$ [$\mathrm{Re}\left(F(\omega)\right)$ gerade, $\mathrm{Im}\left(F(\omega)\right)$ ungerade]

Parseval: $\int_{-\infty}^{\infty} f_1^*(t) f_2(t)\, \mathrm{d}t = \int_{-\infty}^{\infty} F_1^*(\omega) F_2(\omega)\, \mathrm{d}\omega$

Faltung: $[f * g](t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t-t') g(t')\, \mathrm{d}t' = \mathrm{FT}^{-1}\left(FG\right)$

Korrelation ($f, g \in \mathbb{R}$): $\mathrm{corr}(f,g)(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f^*(t'-t)\, g(t')\, \mathrm{d}t' = \mathrm{FT}^{-1}\left(F^* G\right)$

Auto-Korrelation ($f \in \mathbb{R}$): $\mathrm{corr}(f,f)(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f^*(t'-t)\, f(t')\, \mathrm{d}t' = \mathrm{FT}^{-1}\left(|F|^2\right)$

$\int_{-\infty}^{\infty} \mathrm{e}^{-\mathrm{i}\,\omega t}\, \mathrm{d}\omega = 2\pi\, \delta(t)$

Numerisch (DFT):
$\mathcal{O}(N^2)$

$$F_n = \sum_{k=0}^{N-1} \mathrm{e}^{-\frac{2\pi\,\mathrm{i}\,kn}{N}} f_k \qquad f_k = \frac{1}{N} \sum_{n=0}^{N-1} \mathrm{e}^{\frac{2\pi\,\mathrm{i}\,kn}{N}} F_n$$

$F(\omega_n) = \frac{\Delta t}{\sqrt{2\pi}} F_n$, Frequenzauflösung: $\Delta\omega = \frac{2\pi}{N\Delta t}$, Frequenzbereich: $\Omega = \frac{2\pi}{\Delta t}$.

In den meisten Bibliotheken intern „*wrap around order*“: $[f_0, f_1, \ldots, f_{N-1}]$ bzw. $[F_0, F_1, \ldots, F_{N-1}]$.

Numerisch (FFT):
$\mathcal{O}(\log_2(N)N)$

$$F_n = \sum_{k=0}^{\frac{N}{2}-1} \mathrm{e}^{-\frac{2\pi\,\mathrm{i}\,kn}{\frac{N}{2}}} f_{2k} + \mathrm{e}^{-\frac{2\pi\,\mathrm{i}\,n}{N}} \sum_{k=0}^{\frac{N}{2}-1} \mathrm{e}^{-\frac{2\pi\,\mathrm{i}\,kn}{\frac{N}{2}}} f_{2k+1} \quad \text{rekursiv } \log_2(N)\text{-mal.}$$

$\frac{\mathrm{d}}{\mathrm{d}f} \int_{-\infty}^{\infty} F(f(x))\, \mathrm{d}x\big|_h = \int_{-\infty}^{\infty} F'(f(x)) h(x)\, \mathrm{d}x =$ „$F'(f(x))$“

Optimaler-Wiener-Filter $\Phi(\omega)$:
$u(t)$ - eigentl. Signal, $r(t)$ - Response des Messgeräts, $s(t)$ - Messsignal (unverrauscht), $n(t)$ - Rauschen, $c(t) = [r * u](t) + n(t)$ - gemessenes Signal [$U(\omega)$,$R(\omega)$,$S(\omega)$,$N(\omega)$,$C(\omega)$ sind die entsprechenden Größen im Fourierraum].

$$\tilde{\Phi}(\omega) = 1 - \frac{|N|^2}{|S|^2 + |N|^2} \approx 1 - \frac{|N|^2}{|C|^2} \qquad \Rightarrow \qquad U \approx \frac{C}{R} \Phi$$

		analytisch		numerisch	
	Eigenvektor		Eigenwert		Eigenwert
1. Ableitung:	$\mathrm{e}^{\mathrm{i}\,kx}$	$\frac{\mathrm{d}}{\mathrm{d}\,x}$	$\mathrm{i}\,k$	$\frac{1}{2\Delta x}[1\ 0\ -1]$	$\frac{\mathrm{i}\sin(k\,\Delta x)}{\Delta x}$
2. Ableitung:	$\mathrm{e}^{\mathrm{i}\,kx}$	$\frac{\mathrm{d}^2}{\mathrm{d}\,x^2}$	$-k^2$	$\frac{1}{[\Delta x]^2}[1\ -2\ 1]$	$-\frac{4\sin^2(\frac{k\Delta x}{2})}{[\Delta x]^2}$

Anfangswertprobleme:

Für die 1-dimensionale, einfache Kontinuitätsgleichung $\frac{\partial \rho}{\partial t} = -v \frac{\partial \rho}{\partial x}$ gibt es numerisch verschiedene Lösungsansätze:

1. Vorwärts-Euler: $\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}$; mit $\zeta(k) = 1 - v \frac{\mathrm{i}\sin(k\Delta x)}{\Delta x} \Delta t$. [instabil $\forall \Delta t$]
2. Lax: $\frac{u_j^{n+1} - \frac{1}{2}[u_{j+1}^n + u_{j-1}^n]}{\Delta t} = -v \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}$; mit $\zeta(k) = \cos(k\Delta x) - \mathrm{i}\, v \frac{\Delta t \sin(k\Delta x)}{\Delta x}$. [stabil $\forall \Delta t \leq \frac{\Delta x}{v}$]
3. Leap-Frog: $\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = -v \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}$; mit $\zeta(k) = -\mathrm{i}\, \frac{v \sin(k\Delta x)}{\Delta x} \Delta t \pm \sqrt{1 - [\frac{v \sin(k\Delta x) \Delta t}{\Delta x}]^2}$. [stabil $\forall \Delta t \leq \frac{\Delta x}{v}$]

$\zeta(k)$ ist dabei die Änderung des k-ten Eigenvektors mit einem numerischen Integrationsschritt für $\frac{\partial \rho}{\partial t} = -v \frac{\partial \rho}{\partial x}$ mit dem jeweiligen Verfahren. Die **Stabilitätsbedingung nach von Neumann** lautet damit:

$$|\zeta(k)|^2 \leq 1$$

Für die 1-dimensionale, einfache Diffusionsgleichung $\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$ ergeben folgende Verfahren:

1. Vorwärts-Euler: $\frac{u_j^{n+1}-u_j^n}{\Delta t} = D\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}$; mit $\zeta(k) = 1 - 4D\frac{\Delta t}{\Delta x^2}\sin^2(\frac{k\Delta x}{2})$ [stabil $\forall \Delta t \leq \frac{\Delta x^2}{2D}$]
2. Rückwärts-Euler: $\frac{u_j^{n+1}-u_j^n}{\Delta t} = D\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2}$; mit $\zeta(k) = \frac{1}{1+4D\frac{\Delta t}{\Delta x^2}\sin^2(\frac{k\Delta x}{2})}$ [stabil $\forall \Delta t$]
3. Crank-Nicolson: $\frac{u_j^{n+1}-u_j^n}{\Delta t} = \frac{D}{2}\frac{[u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}] + [u_{j+1}^n - 2u_j^n + u_{j-1}^n]}{\Delta x^2}$; mit $\zeta(k) = \frac{1-2D\frac{\Delta t}{\Delta x^2}\sin^2(\frac{k\Delta x}{2})}{1+2D\frac{\Delta t}{\Delta x^2}\sin^2(\frac{k\Delta x}{2})}$ [stabil $\forall \Delta t$]

Für die 1-dimensionale, einfache Schrödingergleichung $\mathrm{i}\frac{\partial u}{\partial t} = [-\frac{\partial^2}{\partial x^2} + V(x)]u = \hat{H}u$ ergeben folgende Verfahren:
$\hat{H}\varphi_k = E(k)\varphi_k$: φ_k Eigenfunktionen von $\hat{H}$, $E(k)$ Eigenwerte

1. Vorwärts-Euler: $\mathrm{i}\frac{u_j^{n+1}-u_j^n}{\Delta t} = [\hat{H}u]_j^n$; mit $\zeta(k) = 1 - \mathrm{i}\,E(k)\Delta t$ [instabil $\forall \Delta t$]
2. Leap-Frog: $\mathrm{i}\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t} = [\hat{H}u]_j^n$; mit $\zeta(k) = -\mathrm{i}\,\Delta t\,E(k) \pm \sqrt{1 - E(k)^2\,\Delta t^2}$ [unitär $\forall \Delta t < \frac{1}{\max_k(|E(k)|)}$]
3. Crank-Nicolson: $\mathrm{i}\frac{u_j^{n+1}-u_j^n}{\Delta t} = \frac{1}{2}\left[[\hat{H}u]_j^n + [\hat{H}u]_j^{n+1}\right]$; mit $\zeta(k) = \frac{1-\mathrm{i}\frac{\Delta t}{2}E(k)}{1+\mathrm{i}\frac{\Delta t}{2}E(k)}$ [unitär $\forall \Delta t$]

Randwertprobleme:

Für die 1-dimensionale, einfache Poissongleichung $\Delta u(x) + f(x) = 0$ ist $u(x)$ auf einem abgeschlossenen Intervall G die Lösung $u(x)$ gesucht, wobei $\begin{cases} u(x) : \forall x \in \partial G & [\text{Dirichlet}] \\ \frac{\partial u}{\partial n}\big|_x : \forall x \in \partial G & [\text{von Neumann}] \end{cases}$ gegeben ist.

Über die Methode der finiten Differenzen Überführung zu $\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + f_i = 0$, $u_i = u(i\Delta x)$, $i \in \{0, 1, \ldots, N\}$.

$\begin{cases} u_0,\ u_N \\ \frac{u_1-u_0}{\Delta x},\ \frac{u_N-u_{N-1}}{\Delta x} \end{cases}$ sind gegeben, weitere u_i somit bestimmbar.

Dies lässt sich auch in Matrix-Schreibweise formulieren: $\overleftrightarrow{A}\vec{u} = \vec{\tilde{f}}$, $\overleftrightarrow{A} \in \mathrm{Mat}(N-1, N-1, \mathbb{C})$, $\vec{u}, \vec{f} \in \mathbb{C}^{N-1}$.

$$A = D + L + U = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_{N-1} \end{pmatrix} + \begin{pmatrix} 0 & & & 0 \\ l_{2\,1} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ l_{N-1\,1} & \cdots & l_{N-1\,N-2} & 0 \end{pmatrix} + \begin{pmatrix} 0 & u_{1\,2} & \cdots & u_{1\,N-1} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & u_{N-2\,N-1} \\ 0 & \cdots & 0 & 0 \end{pmatrix} \quad , \quad e^{\text{numerisch}} = u^{\text{analytisch}} - u^{\text{numerisch}}$$

Lösung:

1. direkte Methoden: Inversion der Matrix, Gauß-Jordan für LGS
2. Fourier-Methoden
3. Relaxationsmethoden: $\tilde{f} := -\frac{\Delta x^2}{2} f$

a) Jacobi-Verfahren: $u_i^{\text{neu}} = \frac{1}{2}[u_{i+1}^{\text{alt}} + u_{i-1}^{alt}] + \frac{\Delta x^2}{2} f_i$ $\qquad Du^{\text{neu}} = u^{\text{alt}} - D^{-1}[Au^{\text{alt}} - \tilde{f}]$

$$D = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix},\ L = -\begin{pmatrix} 0 & & \cdots & 0 \\ \frac{1}{2} & & & \\ 0 & \frac{1}{2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \\ 0 & \cdots & 0 & \frac{1}{2}\ 0 \end{pmatrix},\ U = -\begin{pmatrix} 0 & \frac{1}{2} & 0 & \cdots & 0 \\ & & \frac{1}{2} & \ddots & \vdots \\ \vdots & & \ddots & & 0 \\ & & & & \frac{1}{2} \\ 0 & & \cdots & & 0 \end{pmatrix} \quad , e^{\text{neu}} = \underbrace{-D^{-1}[L+U]}_{R} e^{\text{alt}} \text{ - Konvergenz EW } |\lambda_R| < 1.$$

b) Gauß-Seidel-Verfahren: $u_i^{\text{neu}} = \frac{1}{2}[u_{i+1}^{\text{alt}} + u_{i-1}^{neu}] + \frac{\Delta x^2}{2} f_i$ $\qquad [D+L]u^{\text{neu}} = -U\,u^{\text{alt}} + \tilde{f}]$

$$D = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix},\ L = -\begin{pmatrix} 0 & & \cdots & 0 \\ \frac{1}{2} & & & \\ 0 & \frac{1}{2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \\ 0 & \cdots & 0 & \frac{1}{2}\ 0 \end{pmatrix},\ U = -\begin{pmatrix} 0 & \frac{1}{2} & 0 & \cdots & 0 \\ & & \frac{1}{2} & \ddots & \vdots \\ \vdots & & \ddots & & 0 \\ & & & & \frac{1}{2} \\ 0 & & \cdots & & 0 \end{pmatrix} \quad , e^{\text{neu}} = \underbrace{-[L+D]^{-1}U}_{R} e^{\text{alt}} \text{ - Konvergenz EW } |\lambda_R| < 1.$$

c) Schachbrett-Algorithmus: $u_i^{\text{neu}} = \frac{1}{2}[u_{i+1}^{\text{alt}} + u_{i-1}^{alt}] + \frac{\Delta x^2}{2} f_i$: $\forall i$ ungerade und
$u_i^{\text{neu}} = \frac{1}{2}[u_{i+1}^{\text{neu}} + u_{i-1}^{\text{neu}}] + \frac{\Delta x^2}{2} f_i$: $\forall i$ gerade

Multi-Grid-Verfahren:
Da unterschiedliche Eigenfunktionen einer Dgl [Moden] stark unterschiedliche Eigenwerte bei verschieden feiner

Diskretisierung aufweisen [somit auch der Fehler!], diskretisiert man ein Problem auf einem Gebiet verschieden fein und löst teilweise [und abwechselnd] in beiden Gebieten; den Übergang dazwischen beschreibt man möglichst problemangepasst.

Ist die zu lösende DGL $Au = f$ auf einem Gitter mit Gitterabstand h, so heißt Residuum: $r^h = f^h - A^h u^h$.

Coarse-Grid-Correction: da $Au = f$ und $e^h = u - u^h$ ist, gilt $A^h e^h = \overbrace{A^h u}^{=f^h} - A^h u^h = r^h$ für alle Diskretisierungen h; mit dem Ansatz $e^h = 0$ lässt sich somit der Fehler von u^h näherungsweise berechnen und die Lösung verbessern.

Natürlich kann man auch mehr als zwei Grids verwenden; dabei gibt es dann den V-Zyklus [z.B.: fein → gröber → noch gröber → ... → ganz grob → feiner → ... → fein] und den W-Zyklus [z.B.: fein → gröber → noch gröber → feiner → gröber → feiner → fein].

Finite Elemente Methode:

Statt Entwicklung der Differentialoperatoren nun Entwicklung der Lösung in stetige, stückweise glatte, endliche Basisfunktionen mit kompaktem Träger.

Für die 1-dimensionale, einfache Poissongleichung $\Delta u(x) + f(x) = 0$ auf $\Omega = [0,1]$ mit $u(x) = 0 : u \in \partial\Omega$ gilt:

$$-f = \Delta u \quad \overset{\text{auch } v(x)=0:x\in\partial\Omega \text{ und } v \text{ gutmütig}}{\Longleftrightarrow} \quad -\int_0^1 f(x)\,v(x)\,\mathrm{d}x = \int_0^1 [\Delta u(x)]\,v(x)\,\mathrm{d}x = -\int_0^1 [\partial_x u(x)][\partial_x v(x)]\,\mathrm{d}x$$

[„schwache Integralform“]

Jetzt Entwicklung in $v_j\,, j \in \{1,\dots,N\}$: $u(x) := \sum_{j=1}^{N} u_j\,v_j(x)$, $f(x) := \sum_{j=1}^{N} f_j\,v_j(x)$.

$$\Rightarrow \sum_{i=1}^{N}\int_0^1 [\partial_x[u_i\,v_i(x)]]\,[\partial_x v_j(x)]\,\mathrm{d}x = \sum_{i=1}^{N} u_i \underbrace{\int_0^1 [\partial_x v_i(x)]\,[\partial_x v_j(x)]\,\mathrm{d}x}_{:=L_{ij}} = \sum_{i=1}^{N} f_i \underbrace{\int_0^1 v_i(x)\,v_j(x)\,\mathrm{d}x}_{:=M_{ij}} = \sum_{i=1}^{N}\int_0^1 [f_i\,v_i(x)]\,v_j(x)\,\mathrm{d}x.$$

Die Steifheitsmatrix (L_{ij}) und die Massenmatrix (M_{ij}) und die Kraftentwicklung f_i sind für die Basis jeweils zu berechnen [analytisch]; dann ergibt sich eine „einfach“ zu lösende Aufgabe: $Lu = Mf$.

Molekulardynamik:

Nur statistische, makroskopisch messbare Größen interessant. Ensemble-Mittelwert: $\langle A\rangle = \int \mathrm{d}x\,A(x)\,\rho(x)$

$\rho(x)$ - Phasenraumdichte / Zustandswahrscheinlichkeit.

Alle Möglichkeiten auszuwerten aber ist unmöglich; mit dem zeitlichen Mittelwert $\overline{A} = \lim_{T\to\infty}\frac{1}{T}\int_0^T \mathrm{d}t\,A(t)$ Näherung:

Ergoden-Hypothese: $\langle A\rangle = \overline{A}$ [System entsprechend ... [„ergodisch“]]

Für einfachste, eindimensionale Bewegungsgleichung $m_i\ddot{\vec{r}}_i = \vec{F}_i = -\operatorname{grad}(V(\{\vec{r}_i\}))$:
Initialisierung für $t = 0$, Simulation von n_0 Zeitschritten [→ Anfangswahl verfälscht Ergebnis möglichst wenig], danach in Fortführung Charakteristikum bestimmen $[\overline{A} = \frac{1}{n-n_0}\sum_{j=n_0+1}^{n} A_j]$.

⇒ Verlet-Algorithmus:
$$\vec{r}_i(t+\Delta t) = 2\vec{r}_i(t) - \vec{r}_i(t-\Delta t) + \tfrac{1}{m_i}\vec{F}_i(t)\,[\Delta t]^2$$
$$\vec{v}_i(t+\Delta t) = \tfrac{1}{2\Delta t}[\vec{r}_i(t+\Delta t) - \vec{r}_i(t-\Delta t)$$

⇒ Velocity-Verlet-Algorithmus:
$$\vec{r}_i(t+\Delta t) = \vec{r}_i(t) - \vec{v}_i(t)\,\Delta t + \tfrac{1}{2m_i}\vec{F}_i(t)\,[\Delta t]^2$$
$$\vec{v}_i(t+\Delta t) = \vec{v}_i(t) + \tfrac{1}{2m_i}[\vec{F}_i(t+\Delta t) + \vec{F}_i(t)]\Delta t$$