

Fundamentals of Simulation Methods

Notes



8. Collisionless particle systems

8.1. N-particle ensembles

▷ start. mech.: description of evolution in
6N-dim. phase space

point in 6N-dim phase space

$$\vec{w} = (\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N, \vec{v}_1, \vec{v}_2, \dots, \vec{v}_N)$$

▷ state of N-particle ensemble @ t:
exact one-point particle distribution fct.

$$f(\vec{x}, \vec{v}, t) = \sum_{i=1}^N \delta[\vec{x} - \vec{x}_i(t)] \cdot \delta[\vec{v} - \vec{v}_i(t)]$$

→ # of particles $\Omega(\vec{x}, \vec{v})$ @ t

▷ probability of system $\Omega(t)$ to be at given state

$$p(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N, \vec{v}_1, \vec{v}_2, \dots, \vec{v}_N) d\vec{x}_1 d\vec{x}_2 \cdots d\vec{x}_N d\vec{v}_1 \cdots d\vec{v}_N$$

$$= p(\vec{w}) d\vec{w}$$

▷ reduced statistical description from ensemble averaging:

$$\langle \vec{F}(\vec{x}, \vec{v}, t) \rangle = \int \vec{F} \cdot \rho d\vec{\omega}$$

$$= \int \rho \cdot \left\{ \delta[\vec{x} - \vec{x}_1(t)] \delta[\vec{v} - \vec{v}_1(t)] + \delta[\vec{x} - \vec{x}_2(t)] \delta[\vec{v} - \vec{v}_2(t)] \right. \\ \left. + \dots + \delta[\vec{x} - \vec{x}_N(t)] \delta[\vec{v} - \vec{v}_N(t)] \right\} d\vec{\omega}$$

$$= \int \rho(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N, \vec{v}_1, \vec{v}_2, \dots, \vec{v}_N) d\vec{x}_2 \dots d\vec{x}_N d\vec{v}_2 \dots d\vec{v}_N$$

$$+ \int \rho(\vec{x}_1, \vec{x}_1, \dots, \vec{v}_1, \vec{v}_1, \dots, \vec{v}_N) d\vec{x}_1 \dots d\vec{x}_N d\vec{v}_1 \dots d\vec{v}_N$$

$$+ \dots + \int \rho(\vec{x}_1, \dots, \vec{x}_{N-1}, \vec{x}_N, \vec{v}_1, \dots, \vec{v}_{N-1}, \vec{v}_N) d\vec{x}_1 \dots d\vec{x}_{N-1} \\ d\vec{v}_1 \dots d\vec{v}_{N-1}$$

all N arguments of ρ in \vec{x} and \vec{v} can be permuted

→ describes mean number of particles in

$$(\vec{x}_1, \vec{x}_1 + d\vec{x}) (\vec{v}_1, \vec{v}_1 + d\vec{v}) \sim$$

"one-point distribution fact."

$$f_1(\vec{x}, \vec{v}, t) = N \int p(\vec{x}, \vec{x}_2, \dots, \vec{x}_N, \vec{v}, \vec{v}_2, \dots, \vec{v}_N) d\vec{x}_2 \dots d\vec{x}_N d\vec{v}_2 \dots d\vec{v}_N$$

▷ two particle distribution

$$f_2(\vec{x}, \vec{v}, \vec{x}', \vec{v}', t) = \langle f(\vec{x}, \vec{v}, t) f(\vec{x}', \vec{v}', t) \rangle$$

$$= N(N-1) \int p(\vec{x}, \vec{x}', \vec{x}_3, \dots, \vec{x}_N, \vec{v}, \vec{v}', \vec{v}_3, \dots, \vec{v}_N) d\vec{x}_3 \dots d\vec{x}_N$$

$$d\vec{v}_3 \dots d\vec{v}_N$$

▷ closure $f_3, f_4, \dots \rightarrow \text{BBGKY hierarchy}$

8.2. Uncorrelated systems

▷ reduction of dimensionality by termiology

for BBGKY hierarchy @ certain level

→ closure

▷ simplest closure: regard particles as
uncorrelated

$$f_2(\vec{x}, \vec{v}, \vec{x}', \vec{v}', t) = f_1(\vec{x}, \vec{v}, t) f_1(\vec{x}', \vec{v}', t)$$

► Liouville's theorem: conservation of probability in phase space $\sim p(\vec{w})$
 fulfills continuity eq:

$$\frac{\partial p}{\partial t} + \vec{v}_w \cdot (\vec{p} \dot{\vec{w}}) = 0$$

$$= \frac{\partial p}{\partial t} + \sum_i \left(p \frac{d \dot{x}_i}{d \vec{x}_i} + \frac{\partial p}{\partial \vec{x}_i} \cdot \dot{x}_i + p \frac{d \dot{v}_i}{d \vec{v}_i} + \frac{\partial p}{\partial \vec{v}_i} \cdot \dot{v}_i \right)$$

cancel because:

► Hamiltonian mechanics \rightarrow

$$\dot{\vec{x}} = \frac{\partial H}{\partial \vec{p}}, \quad \dot{\vec{p}} = - \frac{\partial H}{\partial \vec{x}}$$

$$\leadsto \frac{\partial \dot{\vec{x}}}{\partial \vec{x}} = \frac{\partial^2 H}{\partial \vec{x} \partial \vec{p}}, \quad \frac{\partial \dot{\vec{p}}}{\partial \vec{p}} = - \frac{\partial H}{\partial \vec{x} \partial \vec{p}}$$

$$\leadsto \frac{\partial \dot{\vec{x}}}{\partial \vec{x}} = - \frac{\partial \dot{\vec{v}}}{\partial \vec{v}}$$

$$\frac{\partial p}{\partial t} + \sum_i \left(\vec{v}_i \cdot \frac{\partial p}{\partial \vec{x}_i} + \vec{a}_i \cdot \frac{\partial p}{\partial \vec{v}_i} \right) = 0$$

Liouville's
equation

$$\text{particle acceleration } \vec{a}_i = \dot{\vec{v}}_i = \frac{\vec{F}_i}{m_i}$$

⇒ collisionless (uncorrelated) limit → evolutionary eq. for one-point distrib. fct.

$$f_1(\vec{x}, \vec{v}, t) := f(\vec{x}, \vec{v}, t)$$

→ multiply Liouville eq. by $F(\vec{x}, \vec{v}, t)$,

integrate over $d\vec{w}$

$$\rightarrow 1^{\text{st}} \text{ term} \rightarrow \frac{\partial f}{\partial t}$$

→ 2nd term:

$$\sum_i \int F \cdot (\vec{v}_i \cdot \frac{\partial p}{\partial \vec{x}_i}) d\vec{w}$$

$$= \int \underbrace{[\vec{v} \cdot \frac{\partial p}{\partial \vec{x}} d\vec{x}_2 \cdots d\vec{x}_N d\vec{v}_2 \cdots d\vec{v}_N + \dots]}_{N \text{ terms}}$$

$$= \vec{v} \cdot \frac{\partial p}{\partial \vec{x}} \int \underbrace{[p \cdot d\vec{x}_2 \cdots d\vec{x}_N d\vec{v}_2 \cdots d\vec{v}_N + \dots]}_{N \text{ terms}}$$

$$= v \cdot \frac{\partial f}{\partial \vec{x}}$$

→ 3rd term: analogous

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} + \vec{a} \cdot \frac{\partial f}{\partial \vec{v}} = 0$$

Vlasov equation
("collisionless
Boltzmann eq.")

▷ collisionless system → acceleration not due to interaction with single other particles but collective effect

→ example self-gravity

source: mass density

$$g(\vec{x}, t) = m \int f(\vec{x}, \vec{v}, t) d\vec{v}$$

Poisson's equation

$$\nabla^2 \phi = 4\pi G g \quad \phi: \text{grav. potential}$$

$$\nabla \cdot \vec{a} = - \frac{\partial \phi}{\partial \vec{x}}$$

→ Poisson-Vlasov system

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} - \frac{\partial \phi}{\partial \vec{x}} \cdot \frac{\partial f}{\partial \vec{v}} = 0$$

$$\nabla^2 \phi = 4\pi G m \int f(\vec{x}, \vec{v}, t) d\vec{v}$$

8.3. Collisionless systems

▷ consider N-particle system, size R

$$t_{\text{cross}} = \frac{R}{v} \quad v: \text{typical particle velocity}$$

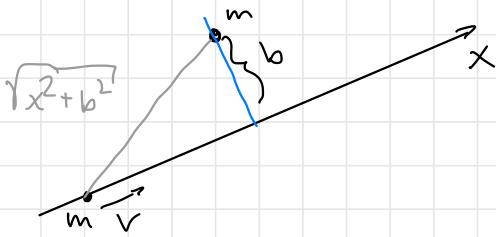
▷ self-gravitating system → virial theorem →

$$v^2 \approx \frac{G N m}{R} = \frac{GM}{R} \quad M = N \cdot m$$

▷ rare of particle to experience weak deflections by other particles

▷ "Born approximation": deflection angle

$$\vartheta_d \ll 1$$



b : "impact parameter"

$$\begin{aligned}\Delta p_{\perp} &= m \Delta v_{\perp} = \int_{-\infty}^{\infty} F_{\perp} dt = -m \int_{-\infty}^{\infty} \nabla_{\perp} \Phi dt \\ &= - \int_{-\infty}^{\infty} \frac{2}{\partial b} \left(\frac{G m^2}{\sqrt{b^2 + v^2 t^2}} \right) dt \\ &= \frac{2 G m^2}{b v}\end{aligned}$$

▷ deflection angle : $\vartheta_d \approx \tan \vartheta_d = \frac{\Delta p_{\perp}}{p} = \frac{2 G m}{b v^2}$

def : $b_0 = \frac{2 G m}{v^2} \quad \approx b < b_0$: large-angle deflection

$$\approx \vartheta_d = \frac{b_0}{b}$$

▷ random walk : $\langle \vartheta_d \rangle = 0$ but:

$$\langle \vartheta_d^2 \rangle = \sum_{\text{all encounters}} \vartheta_d^2 = \sum \left(\frac{b_0}{b} \right)^2$$

encounters with $[b, b + db]$ during t:

$$dN = n v t 2\pi b db \sim$$

$$\langle v_d^2 \rangle = \int_{b_{\min}}^{b_{\max}} \left(\frac{b_0}{b} \right)^2 dN = 2\pi n v t b_0^2 \ln \Lambda$$

"Coulomb logarithm" $\ln \Lambda = \ln \left(\frac{b_{\max}}{b_{\min}} \right)$

▷ cylinder of radius R \cap

$$N = v t \pi R^2 n$$

$$\begin{aligned} \langle (\Delta v_{\perp})^2 \rangle &\approx v^2 \langle v_d^2 \rangle = 2N \left(\frac{v b_0}{R} \right)^2 \ln \Lambda \\ &= 8N \left(\frac{G m}{R v} \right)^2 \ln \Lambda \end{aligned}$$

▷ def relaxation time

$$t_{\text{relax}} := \frac{v^2}{\langle (\Delta v_{\perp})^2 \rangle} t_{\text{cross}}$$

$$= \frac{N}{8 \ln \Lambda} t_{\text{cross}}$$

$$\triangleright b_{\max} \approx R, \quad b_{\min} = b_0$$

$$b_{\min} = \frac{2R}{N}$$

$$\sim \ln \Lambda = \ln \left(\frac{N}{2} \right) \sim \ln(N)$$

$$\sim t_{\text{relax}} = \frac{N}{8 \ln N} t_{\text{cross}}$$

\triangleright system collisionless if $t_{\text{relax}} \gg t_{\text{age}}$

$\sim N$ large!

8.4. N-body models of collisionless systems

\triangleright Vlasov - Poisson system

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} - \frac{\partial \phi}{\partial \vec{x}} \cdot \frac{\partial f}{\partial \vec{v}} = 0$$

$$\nabla^2 \phi = 4\pi G m \int f(\vec{x}, \vec{v}, t) d\vec{v}$$

\triangleright re-introduce particles to discrete collisionless fluid but: far fewer than in real system

→ give them higher mass → fiducial
"macro-particles"

▷ exs. of motion for macro-particles

$$\ddot{\vec{x}}_i = -\nabla \Phi(\vec{x}_i) \quad (*)$$

$$\Phi(\vec{x}) = -G \sum_{j=1}^N \frac{m_j}{[(\vec{x} - \vec{x}_j)^2 + \varepsilon^2]^{1/2}} \quad (**)$$

▷ if $t_{\text{relax}} \gg t_{\text{sim}}$ despite smaller $N \rightarrow$
numerical model behaves as collisionless system
over t_{sim} , collecting grav. potential sufficiently
smooth

▷ mass m_j of macro-particles does not enter (*),
but only (**) \sim if N sufficient to describe
grav. potential accurately \rightarrow orbits of macro-
particles as valid as those of real particles

▷ N-body model \longrightarrow one realization of
one-point distr. fct. f but not ensemble
average!

▷ numerical problems

.) large-angle scatterings

.) orbits in singular potentials \rightarrow sufficiently accurate num. integration expensive

.) possible formation of bound particle pairs
 \rightarrow violation of collisionless behavior

\rightarrow prevented by gravitational softening length ε
in the potential (**)

~ bound pairs not formed if $\langle v^2 \rangle \gg \frac{Gm}{\varepsilon}$

\rightarrow but: implies small resolution length scale

~ softening choice: compromise between spatial resolution, discrusses noise in orbits and
grav. potential, computational cost, relaxation effects

8.5. Time evolution of N-body models

> 2 tasks :

- 1) integrate eq. of motion (*) in time
- 2) compute r.h.s. of (*) \rightarrow solve (**)

1) use ODE integration schemes \rightarrow preferably
symplectic scheme

2) direct summation :

$$\ddot{\vec{x}}_i = -G \sum_{j=1}^N \frac{m_j}{[(\vec{x}_i - \vec{x}_j)^2 + \epsilon^2]^{3/2}} (\vec{x}_i - \vec{x}_j)$$

\rightarrow exact but **expensive!** \rightarrow sum of N particle
forces has to be computed for each of the N

e.g. \rightarrow cost: $\Theta(N^2)$

\rightarrow large N needed

\rightarrow need fast approximation

or: direct summation w/ **hardware acceleration**

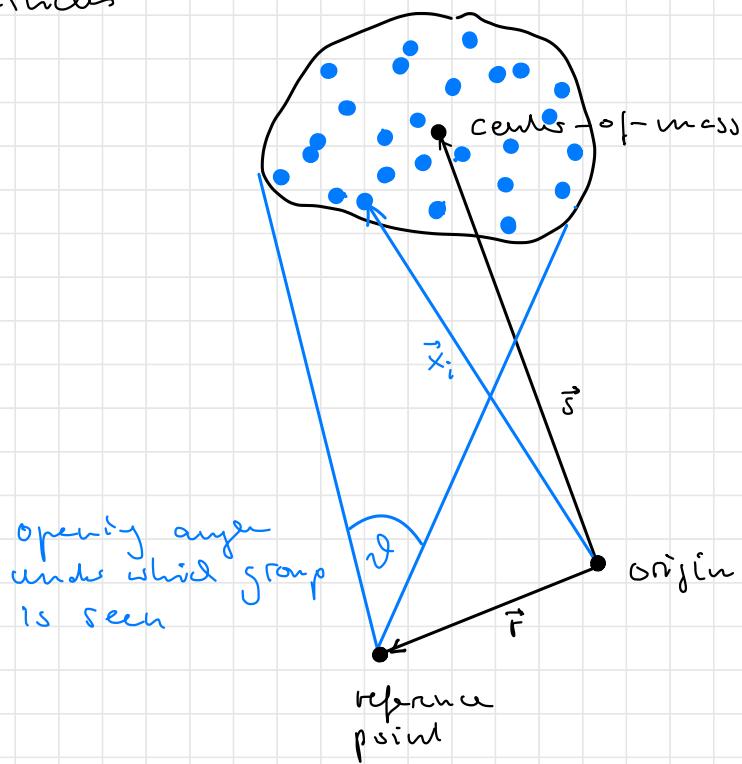
GRAPE boards, GPUs,

9. Tree methods

- Central idea: instead of summing up forces from all N particles → use multipole expansion of distant groups

9.1. Multipole expansion

- group of particles



▷ potential of group:

$$\phi(\vec{r}) = -G \sum_i \frac{m_i}{|\vec{r} - \vec{x}_i|} = -G \sum_i \frac{m_i}{|\vec{r} - \vec{s} + \vec{s} - \vec{x}_i|}$$

▷ if δ small $\sim |\vec{x}_i - \vec{s}| \ll |\vec{r} - \vec{s}|$

▷ Taylor expansion ($\vec{y} := \vec{r} - \vec{s}$)

$$\frac{1}{|\vec{y} + \vec{s} - \vec{x}_i|} = \frac{1}{|\vec{y}|} - \frac{\vec{y}(\vec{s} - \vec{x}_i)}{|\vec{y}|^3}$$

$$+ \frac{1}{2} \frac{\vec{y}^T [3(\vec{s} - \vec{x}_i)(\vec{s} - \vec{x}_i)^T - (\vec{s} - \vec{x}_i)^2] \vec{y}}{|\vec{y}|^5}$$

+ ...

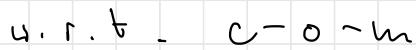
▷ multipole moments

monopole: $M = \sum_i m_i$

dipole: $\vec{D} = \sum_i m_i (\vec{s} - \vec{x}_i)$

quadrupole: $Q_{ij} = \sum_k m_k [3(\vec{s} - \vec{x}_k)_i (\vec{s} - \vec{x}_k)_j - \delta_{ij} (\vec{s} - \vec{x}_k)^2]$

▷ dipole : how much mass distributed asymmetrically



$$\vec{s} = \frac{1}{M} \sum_i m_i \vec{x}_i$$

→ dipole vanishes by construction

▷ up to quadrupole order

$$\phi(\vec{r}) \approx -G \left(\frac{M}{|\vec{y}|} + \frac{1}{2} \frac{\vec{y}^T Q \vec{y}}{|\vec{y}|^5} \right)$$

acceptable if

$$\mathcal{J} \approx \frac{\langle |\vec{x}_i - \vec{s}| \rangle}{|\vec{y}|} \simeq \frac{l}{y} \ll 1$$

l : radius of group

▷ force from differentiation

Q.2. Hierarchical grouping

- ▷ tree algorithms \rightarrow tree-like data structure to
 -) hierarchically group particles
 -) pre-compute multipole moments
- ▷ different strategies \rightarrow popular: Barnes & Hut (1986)
 - \rightarrow Oct-tree
- ▷ sequentially insert particles into tree \rightarrow compute multipole moments recursively

Q.3. Tree walk

- ▷ force calculation by "walking the tree"
 -) start out @ root
 -) check for each subsequent node whether $\Delta \theta < \Delta\theta_c$ (prescribed tolerance angle)
 -) if so: accept multipole expansion of node
 - \rightarrow calculate corresponding particle force

→ add to total force, stop walk
along branch

•) if not : open node → check for
SLW - nodes

▷ resulting force : approximation → accuracy
controlled by θ_c

▷ cost of tree-based force calculation :

consider sphere w/ radius R , N particles

~ homogeneously distributed \rightarrow mean particle
spacing :

$$d = \left[\frac{4\pi}{3} \frac{R^3}{N} \right]^{1/3}$$

→ comp. cost for calculating force on central
particle? \rightarrow # of interaction terms

$$N_{\text{nodes}} \approx \int_d^R \frac{4\pi r^2 dr}{l^3(r)}$$

$l(r)$: node size @ distance r

$$l(r) \simeq \mathcal{O}_c r$$

$$\sim N_{\text{nodes}} = \frac{4\pi}{V_c} \ln \frac{R}{d} \propto \frac{\ln N}{\mathcal{O}_c^3} \quad (*)$$

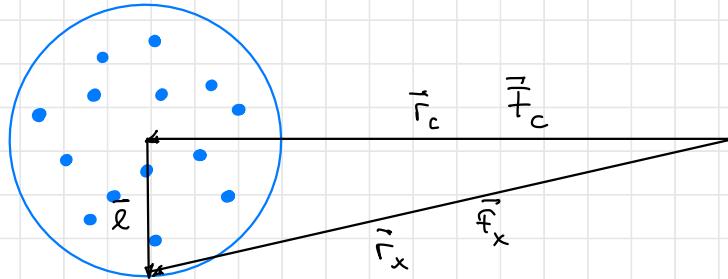
\sim total comp. cost : $\mathcal{O}(N \ln N) \rightarrow$ strong

improvement over $\mathcal{O}(N^2)$ of direct summation!

▷ typical force error : keep only monopoles

\rightarrow error in force per unit mass from one node

\sim truncation error



$$\Delta F_{\text{node}} \sim |F_c - F_x|$$

$$= \left| \frac{G M_{\text{node}}}{r_c^2} - \frac{G M_{\text{node}}}{r_x^2} \right|$$

$$= \left| \frac{G M_{\text{node}}}{r_c^2} - \frac{G M_{\text{node}}}{r_c^2 + l^2} \right|$$

$$= \frac{G M_{node}}{r_c^2} \frac{l^2}{r_c^2} = \frac{G M_{node}}{r_c^2} \vartheta_c^2$$

$$(\Delta F_{tot})^2 \sim N_{node} (\Delta F_{node})^2 \approx \frac{M_{tot}}{N_{node}}$$

$$= N_{node} \left(\frac{G M_{node}}{r_c^2} \vartheta_c^2 \right)^2$$

$$\propto \frac{\vartheta_c^4}{N_{node}} \stackrel{(*)}{\propto} \vartheta_c^7$$

~ force error of scheme (v, monopoles only):

$$(\Delta F_{tot}) \propto \vartheta_c^{3.5} \rightarrow \text{roughly inverse to}$$

involved comp. cost for fixed N

10. Basic fluid dynamics

- ▷ evolution of matter in which collisions between particles are frequent
 - efficient description @ large characteristic scale l of system
- ▷ fluid characterized by density ρ , velocity \vec{v} , temperature T
- ▷ concept of fluid mechanics applicable if
 - (i) $l \gg l_{\text{mfp}}$
 - ~ microscopic behavior of particles negligible
 - ~ concept of fluid element with scale l_f
 - $l \gg l_f \gg l_{\text{mfp}}$
 - ~ # of particles in f.e. large
 - ~ average quantities ($\rho, \vec{v}, T \dots$) can be defined:
 - velocity of . f.e. \vec{v} from particle velocities \vec{u}

$$\vec{u} = \vec{v} + \vec{w} \quad \vec{v} = \langle \vec{u} \rangle$$

$\lambda_{\text{mfp}} \ll l_f \rightarrow$ random velocity component is small \rightarrow

$$\text{define } T : \langle |\vec{w}|^2 \rangle = \frac{3k_B T}{m}$$

\rightarrow fluid element remains well-defined in evolution: particles exchange over boundary:
slow diffusion process

(ii) interactions between particles must saturate, i.e. forces must be short-ranged

$$\lim_{N \rightarrow \infty} \left(\frac{E}{N} \right) = \text{const.}$$

▷ Derivation of basic equations in physics of continua: general form of balance law

▷ consider extensive quantity A in volume V
(fixed in space)

- ▷ "amount" of A can change only due to production / depletion d_s inside V or due to flux of A d_f over surface ∂V

$$\boxed{\frac{dA}{dt} = \frac{df A}{dt} + \frac{ds A}{dt}}$$

"integral form of balance law"

- ▷ define:

-) density of A: $a \rightarrow A = \int_a dV$

-) flux density $\vec{j}_a \rightarrow \frac{df A}{dt} = - \int_{\partial V} \vec{j}_a d\vec{s}$

-) production/depletion of A in $V \rightarrow$
production density $s(a)$

$$\frac{ds A}{dt} = \int s(a) dV$$

$$\begin{aligned} \textcircled{n} \quad \frac{d}{dt} \int_a dV &= \int \frac{\partial}{\partial t} a dV \\ &= - \int_{\partial V} \vec{j}_a d\vec{s} + \int s(a) dV \end{aligned}$$

$$= \int_V [-\nabla \cdot \vec{j}_a + s(a)] dV$$

V arbitrary \curvearrowright

$$\frac{\partial a}{\partial t} = -\nabla \cdot \vec{j}_a + s(a)$$

"differential form of balance law"

16.1 The equations of fluid dynamics

▷ eqs. of fluid dyn. based on 3 principles:
conservation of mass, momentum, energy

•) mass conservation

mass density: ρ

mass flux density: $\rho \vec{v}$

$$\curvearrowright \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0$$

"continuity equation"

"Eulerian form" \rightarrow describes flow in fixed

lab frame of reference

→ def. "Lagrangian derivative":

$$\frac{D}{Dt} := \frac{\partial}{\partial t} + \vec{v} \cdot \nabla$$

use $\nabla \cdot \vec{A} = \vec{A} \cdot \nabla + \vec{v} \cdot \vec{\nabla}$

~
$$\boxed{\frac{D\vec{A}}{Dt} + g \nabla \cdot \vec{v} = 0}$$
 "Lagrangian form"

→ describes temporal change along with motion of fluid element

1) momentum balance:

Sources: external forces \vec{f} , e.g. gravity

$$\vec{f} = -\nabla \phi$$

Momentum density: $\rho \vec{v}$

Momentum flux density: $g(\vec{v} \otimes \vec{v}) + \underline{\Pi}$

$$(\vec{v} \otimes \vec{v})_{ik}^{\swarrow} = v_i v_k$$

outer product

pressure tensor $\underline{\Pi} := P \mathbb{1} - \tau$

isotropic pressure P

viscous stress tensor: $\underline{\underline{\tau}}$

$$\rho \frac{\partial(\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v} \otimes \vec{v}) + \nabla \cdot \underline{\underline{\tau}} = \rho \vec{f}$$

→ combine w/ continuity eq.

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \nabla) \cdot \vec{v} + \frac{1}{\rho} \nabla \cdot \underline{\underline{\tau}} = \vec{f}$$

"Navier - Stokes equation"

ideal fluid: no viscosity

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \nabla) \cdot \vec{v} + \frac{\nabla P}{\rho} = \vec{f}$$

"Euler equation"

→ Lagrangian form:

$$\frac{D \vec{v}}{Dt} + \frac{\nabla P}{\rho} = \vec{f}$$

•) energy balance

sources : external forces

specific total energy : ε_{tot}

total energy density : $\rho \varepsilon_{\text{tot}}$

total energy flux density : $\rho \varepsilon_{\text{tot}} \vec{v} + \vec{v}$ \square

ideal fluid $\rightarrow \rho \varepsilon_{\text{tot}} \vec{v} + P \vec{v}$

$$\text{A} \quad \frac{\partial \varepsilon_{\text{tot}}}{\partial t} + \frac{1}{\rho} \nabla \cdot [(\rho \varepsilon_{\text{tot}} + P) \vec{v}] = \vec{v} \cdot \vec{f}$$

Lagrangian form

$$\frac{D \varepsilon_{\text{tot}}}{Dt} + \frac{1}{\rho} \nabla \cdot P \vec{v} = \vec{v} \cdot \vec{f}$$

10.2. Scaling properties of viscous flows

▷ assume τ linear in velocity derivatives

→ "Newtonian fluid"

$$\tau = \eta [\nabla \vec{v} + (\nabla \vec{v})^T - \frac{2}{3} (\nabla \cdot \vec{v}) \mathbb{1}] + \xi (\nabla \cdot \vec{v}) \mathbb{1}$$

$\eta(s, T, \dots)$ "shear viscosity"

$\zeta(s, T, \dots)$ "bulk viscosity"

▷ $\frac{D\vec{v}}{Dt} + g \vec{v} \cdot \nabla \vec{v} = 0 \rightarrow$ incompressible fluid

$$\frac{D\vec{v}}{Dt} = 0 \quad \wedge$$

$$\nabla \cdot \vec{v} = 0$$

$$\sim \tau = \eta [\nabla \vec{v} + (\nabla \vec{v})^T] J$$

def. "kinematic viscosity" $\nu := \frac{\eta}{g}$

▷ Navier - Stokes (no ext. forces)

$$\frac{D\vec{v}}{Dt} + \frac{\nabla P}{g} - \nu \nabla \cdot [\nabla \vec{v} + (\nabla \vec{v})^T] = 0$$

▷ consider flow problem characterized by

$$L_0, V_0, g_0$$

→ def. dimensionless fluid variables:

$$\hat{\vec{v}} = \frac{\vec{v}}{V_0}, \quad \hat{\vec{x}} = \frac{\vec{x}}{L_0}, \quad \hat{g} = \frac{g}{g_0}$$

$$\hat{P} = \frac{\bar{P}}{g_0 V_0 L}, \quad \text{and}$$

$$\hat{t} = \frac{t}{L_0 / V_0}, \quad \hat{\nabla} = L_0 \nabla$$

→ NS in "dimensionless form"

$$\frac{D \hat{v}}{Dt} = - \frac{\hat{\nabla} \hat{P}}{\hat{\rho}} + \frac{\nu}{L_0 V_0} \hat{\nabla} \cdot [\hat{\nabla} \hat{v} + (\hat{\nabla} \hat{v})^T]$$

▷ structure of possible solutions completely determined by Reynolds number

$$Re := \frac{L_0 V_0}{\nu}$$

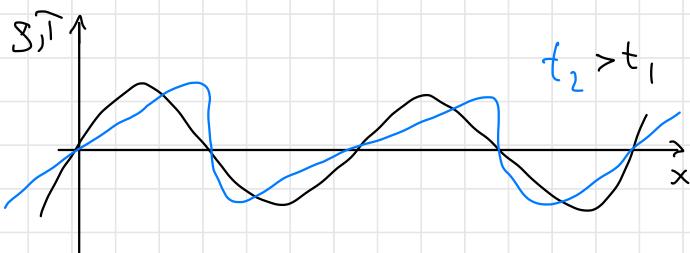
→ Reynolds number similarity

▷ interpretation: Re measures importance of inertial forces $\frac{D \hat{v}}{Dt}$ relative to viscous forces

$Re \rightarrow \infty$ recovers Euler eqs.

10.3. Hydrodynamical shocks

- ▷ Shock waves: ρ, \vec{v}, T, s jump by finite amounts
- ▷ Can result from collision flows or discontinuities in initial conditions (i.c.)
- ▷ can develop from perfectly smooth i.c. by "wave steepening"



$c_s T$ for g^T and $T^T \approx$ crabs of sound waves overtake troughs \rightarrow wave "breaks"

\rightarrow shock forms

- ▷ Euler \rightarrow shocks $\stackrel{!}{=}$ discontinuities
 - ↗ differential form of eqs. does not provide "weak solutions" \rightarrow but integral form

remains valid \Rightarrow fluxes of mass,
 momentum, energy continuous @ shock
 front

\triangleright shock connecting 2 piecewise constant state

\rightarrow "Rankine - Hugoniot jump conditions"

- frame of reference in which shock
 stationary ($v_s = 0$):

$$\gamma_1 v_1 = \gamma_2 v_2$$

$$\gamma_1 v_1^2 + P_1 = \gamma_2 v_2^2 + P_2$$

$$(\gamma_1 \varepsilon_{\text{tot},1} + P_1) v_1 = (\gamma_2 \varepsilon_{\text{tot},2} + P_2) v_2$$

} (*)

\triangleright ideal gas: pre-shock gas streams supersonically
 into discontinuity ($v_1 > c_1$, $c_1^2 = \gamma \frac{P_1}{\gamma_1}$)

\sim Mach number

$$M_1 = \frac{v_1}{c_1} > 1$$

measures shock strength

- $v_2 < v_1$, $\gamma_2 > \gamma_1$, $T_2 > T_1$
- post-shock flow subsonic $v_2 < c_2$
- (*) \sim

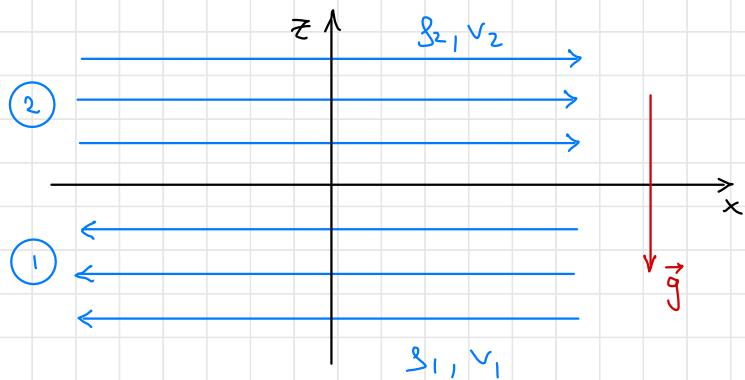
$$\frac{\gamma_2}{\gamma_1} = \frac{(\gamma + 1) M_1^2}{(\gamma - 1) M_1^2 + 2}$$

10.4. Flow instabilities

- small perturbation grows rapidly, tapping some source of free energy

- considers shear

flow in
homogeneous
grav. field



▷ analyze stability with perturbation theory

→ eq. for fct. $\xi(x, t) = z$ that describes

z -location of interface between the 2 flow phases ① and ②

▷ for single perturbative Fourier mode

$$\xi = \begin{cases} 1 & e^{i(kx - \omega t)} \\ 0 & \end{cases}$$

→ dispersion relation

$$\omega^2(g_1 + g_2) - 2\omega k(g_1 v_1 + g_2 v_2) + k^2(g_1 v_1^2 + g_2 v_2^2)$$

$$+ (g_2 - g_1) k g = 0$$

↗ exp. growing mode if solutions for ω with positive imaginary part

▷ special cases

(i) Rayleigh-Taylor instability

▷ fluid at rest $v_1 = v_2 = 0$ ↗

$$\omega^2 = \frac{(g_1 - g_2) k g}{g_1 + g_2}$$

▷ $g_2 > g_1$, ↗ unstable : $\omega^2 < 0$

▷ buoyancy-driven instability

→ pot. energy of grav. field trapped

▷ arbitrarily small wavelengths unstable,

mode w, smallest $\lambda = \frac{2\pi}{k}$ grows fastest

▷ $g_1 > g_2$: stable stratification

(ii) Kelvin-Helmholtz instability

▷ $g = 0$ ↗

$$\omega_{1/2} = \frac{k(g_1 v_1 + g_2 v_2)}{g_1 + g_2} \pm ik \sqrt{\frac{g_1 g_2}{g_1 + g_2}} |v_1 - v_2|$$

▷ $|v_1 - v_2| > 0$ ↗ always unstable

▷ grows fastest for small scales (high k)

▷ stabilization of shear flows by gravity

$$v_1 \neq v_2, g > 0 \quad \text{?}$$

$$\omega_{1/2} = \frac{k(\rho_1 v_1 + \rho_2 v_2)}{\gamma_1 + \gamma_2} \pm \sqrt{\frac{-k\gamma_1\gamma_2(v_1 - v_2)^2 - (\gamma_1 + \gamma_2)(\gamma_2 - \gamma_1)kg}{\gamma_1 + \gamma_2}}$$

? stable for $\gamma_1 > \gamma_2$ and

$$(v_1 - v_2)^2 < \frac{(\gamma_1 + \gamma_2)(\gamma_1 - \gamma_2)g}{k\gamma_1\gamma_2}$$

? stable for sufficiently small k

10.5 Turbulence

▷ unsteady, irregular, chaotic fluid flow

→ v varies significantly and irregularly in time and position

▷ "statistical phenomena"

▷ consider subsonic flow → approx. treated as incompressible & solenoidal (divergence-free)

Motions \sim only shear \sim subject to fluid instability (K_{Ht}) \rightarrow "swirling" or "vortex-like" structures on many scales
 \rightarrow "eddies" as conceptual boundary blocks

\triangleright origin: non-linear part of momentum balance
eq. \rightarrow incompressible Navier-Stokes

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} + \underbrace{\frac{\nabla p}{\rho}}_{\text{non-lin. term}} - \nu \nabla \cdot [\nabla \vec{v} + (\nabla \vec{v})^T] = 0$$

$\underbrace{\hspace{10em}}$ $\underbrace{\hspace{10em}}$

non-lin. term \rightarrow counteracted by viscous term

\curvearrowright turbulence for $Re = \frac{L_0 V_0}{\nu} \gg 1$

\triangleright model incompressible turbulent flow as compound of eddies of various sizes $l \rightarrow$ velocities $v(l)$

$$\rightarrow \text{"eddy turnover timescale"} \quad \tau(l) = \frac{l}{v(l)}$$

\triangleright largest eddies: $l \sim L_0, v(l) \sim V_0$

$$Re(L_0) = \frac{L_0 V_0}{\nu} \gg 1 \quad \curvearrowright \text{viscosity unimportant}$$

\rightarrow unstable \rightarrow break up \rightarrow transfer

energy to smaller eddies \rightarrow "turbulent cascade"

$\triangleright \nu$ unimportant down to smallest scales

$$Re(l) = \frac{l v(l)}{\nu} \approx 1$$

$\sim \nu$ starts to dominate \sim kin. energy
dissipated away

$\triangleright Q$: size of smallest eddies?

Q : How does $v(l)$ scale with l ?

\triangleright Kolmogorov's "theory" \rightarrow based on
3 hypotheses:

1. for $Re(l_0) \gg 1$ turbulent motions @ $l \ll l_0$

become statistically isotropic

2. for $Re(l_0) \gg 1$ — statistics of small-scale
turbulent motions has universal form
 \rightarrow determined by ν and energy injection

rate (per unit mass) ε

→ implication of 1. and 2. → unique characteristic "Kolmogorov length" η ,
velocity v_η , time scale T_η :

$$\eta = \left(\frac{v^3}{\varepsilon} \right)^{1/4}$$

$$v_\eta \equiv (\varepsilon v)^{1/4}, \quad T_\eta \equiv \left(\frac{v}{\varepsilon} \right)^{1/2}$$

$$\sim \text{Re}(\eta) = \frac{\eta v_m}{\nu} = 1 \rightarrow \eta \rightarrow \text{range of dissipation}$$

3. for $\text{Re}(L_0) \gg 1 \rightarrow$ range of scales

$L_0 \gg l \gg \eta$ when statistics of motions

universal and only dependent on ε

→ \rightarrow unimportant Θ & \sim flow driven by

non-linear advection term → "inertial

range" \sim given l and $\varepsilon \rightarrow$ constant)

characteristic

$$v(\lambda) = (\varepsilon \lambda)^{1/3} \quad \tau(\lambda) = \left(\frac{\lambda^2}{\varepsilon}\right)^{1/3}$$

→ energy transfer rate ε in inertial range

$$\varepsilon \sim \frac{v^2(\lambda)}{\tau(\lambda)}$$

$$\rightarrow \varepsilon = \frac{V_0^3}{L_0}$$

▷ eddies @ $\lambda \hat{=} k = \frac{L_u}{\lambda}$ (wavenumber)

→ limit energy between k_1 and k_2

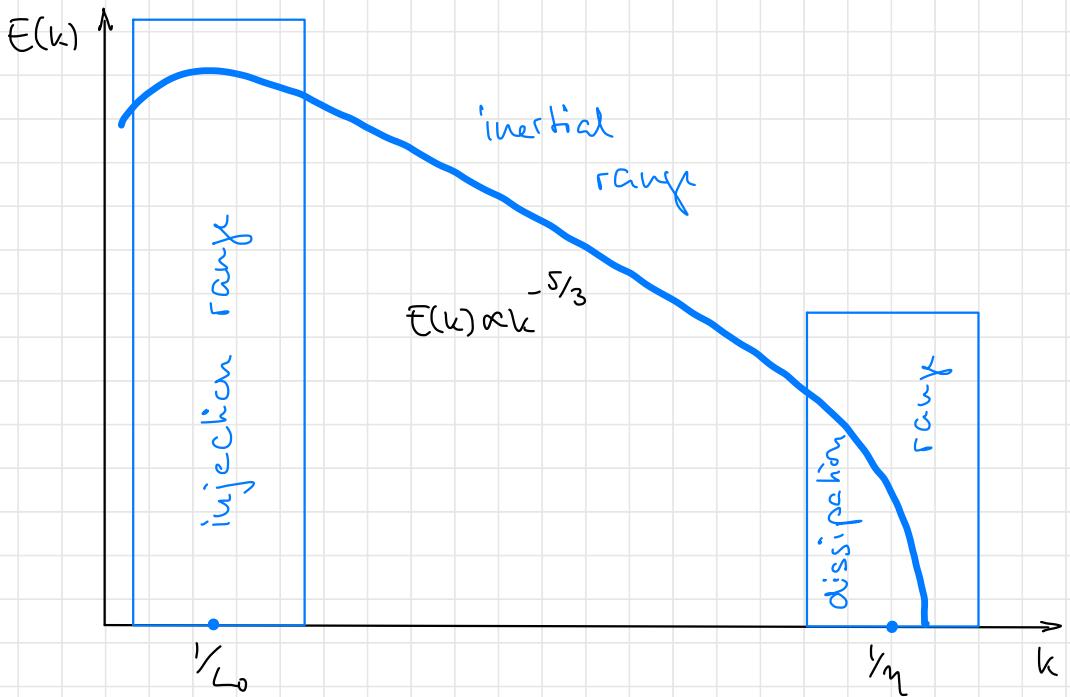
$$\Delta E = \int_{k_1}^{k_2} E(k) dk$$

$$E(k) = C \varepsilon^a k^b$$

→ dimensional analysis: $a = 2/3$, $b = -5/3$

$$E(k) = C \varepsilon^{2/3} k^{-5/3}$$

Kolmogorov power spectrum



II. Eulerian hydrodynamics

II.1. Partial differential equations

- ▷ no general approach, depends on type of PDE
- ▷ PDEs classified by
 -) order
 -) linearity
 -) homogeneity
 -) type

example: linear 2nd order PDE:

dependent variable: u

2 independent variables: x, y

$$a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + d \frac{\partial u}{\partial x} + e \frac{\partial u}{\partial y} + f u = g$$

→ classified by discriminant:

$$D = b^2 - 4ac$$

→ distinguish

$$D \left\{ \begin{array}{ll} < 0 & \text{"elliptic"} \\ = 0 & \text{"parabolic"} \\ > 0 & \text{"hyperbolic"} \end{array} \right\} \text{PDE}$$

prototypes:

.) wave eq.

$$\frac{\partial^2 u}{\partial t^2} - c_s^2 \frac{\partial^2 u}{\partial x^2} = 0$$

$$a=1, b=0, c=-c_s^2 \cap D=4c_s^2 > 0$$

↪ hyperbolic

.) heat conduction eq.

$$\frac{\partial u}{\partial t} - \lambda^2 \frac{\partial^2 u}{\partial x^2} = 0$$

$$a=0, b=0, c=-\lambda^2 \cap D=0$$

↪ parabolic

.) Laplace eq.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

$$a=1, \quad b=0, \quad c=1 \quad \text{and} \quad D=-4$$

\rightarrow elliptic

- general lin. 2nd order PDE w/ more than 2 independent variables:

$$\sum_{i,j=1}^n a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^n b_i \frac{\partial u}{\partial x_i} + cu + d = 0$$

\rightarrow eigenvalues of coefficient matrix a_{ij}

determine type:

-) elliptic: all eigenvalues positive or
— a — negative
-) parabolic: one zero eigenvalue, others
all >0 or all <0
-) hyperbolic: one ev. positive, all others
 <0 , or vice versa

▷ linear systems of 1st order homogeneous PDEs:

$$\frac{\partial u_i}{\partial t} + \sum_j A_{ij} \cdot \frac{\partial u_i}{\partial x_j} = 0$$

A = (A_{ij}) : coefficient matrix

→ hyperbolic if A diagonalizable, only real eigenvalues

▷ extension to non-linear

$$\frac{\partial \vec{u}}{\partial t} + \frac{\partial}{\partial \vec{x}} [\vec{f}(\vec{u})] = 0$$

↑ some non-lin. fct.

\vec{f} : flux

→ rewrite into quasi-linear form:

$$\frac{\partial \vec{u}}{\partial t} + \underbrace{\frac{\partial \vec{f}}{\partial \vec{u}} \cdot \frac{\partial \vec{u}}{\partial \vec{x}}}_{\text{Jacobian of flux}} = 0$$

Jacobian of flux

hyperbolic if Jacobian diagonalizable with real eigenvalues

▷ character of PDE types:

•) elliptic: often connected to static problems

or equilibrium states

→ important example: Poisson eq.

→ solution completely determined by
boundary conditions

•) parabolic: often 2nd order → describe

slowly changing processes (e.g. diffusion)

→ require initial data $\bar{u}(\bar{x}, t_0)$ and
b.c.

→ solutions become smoother with time

•) hyperbolic: initial value problem

→ solutions can develop steep regions
and discontinuities

→ require i.c. $\bar{u}(\bar{x}, t_0)$ and some
b.c.

- ▷ boundary conditions :
 - .) Dirichlet : specify u @ boundary
 - .) von Neumann : specify derivative @ boundary

11.2. Discretization approaches

- ▷ finite difference methods : approximate differential operators by finite differences on structured mesh
- ▷ finite volume method : particularly useful for hyperbolic convection laws
- ▷ spectral methods : solution represented by linear combination of func. \rightarrow transform PDE into algebraic eqs. or ODEs
example: solving Poisson eq w/ Fourier
- ▷ method of lines : semi-discrete :
all derivatives except for one approximated by finite diff. \rightarrow set of ODEs

often: discrete in space, but not in time

\rightarrow ODE describing time evolution

\triangleright finite elements: subdivide domain into cells ("elements") of arbitrary shape, represent solution in terms of simple functions per element

PDE \rightarrow algebraic problem for coefficients of func. \rightarrow similar to spectral, but local

II. 3. Simple advection

\triangleright 1st order hyperbolic PDE: 1D lin. advection

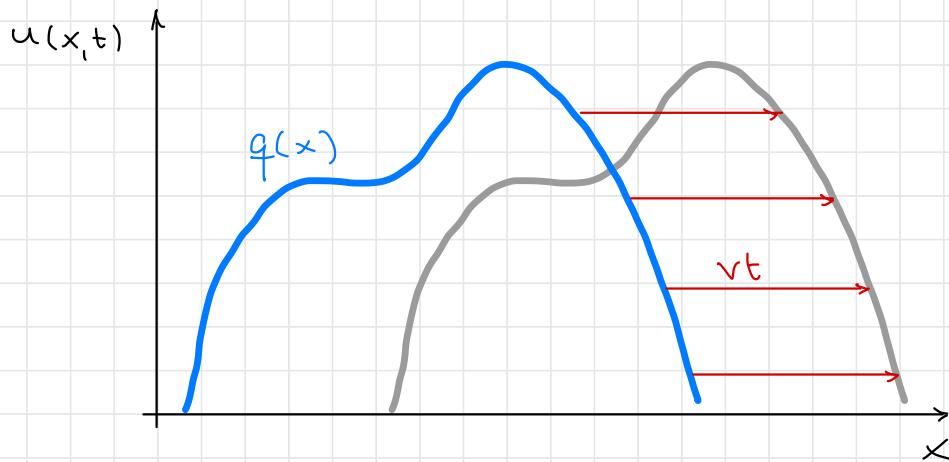
ay. w , constant coefficient

$$\frac{\partial u}{\partial t} + v \cdot \frac{\partial u}{\partial x} = 0 \quad u = u(x, t), \quad v = \text{const}$$

\triangleright i.e.: $u(x, t=0) = g(x) \rightarrow$

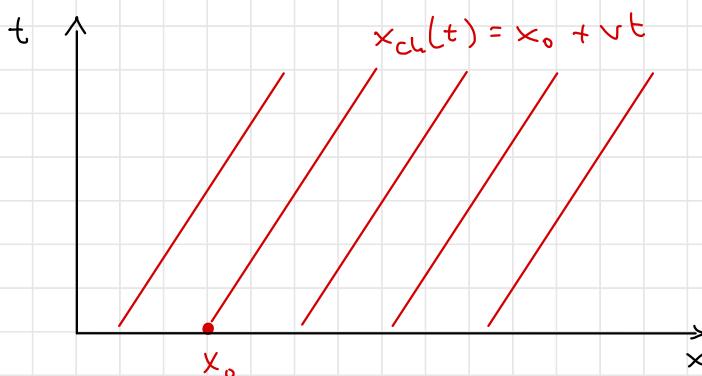
Solution:

$$u(x,t) = q(x - vt)$$



▷ points start at x_0 $\longrightarrow x_{ch}(t) = vt + x_0$

$x_{ch}(t)$: "characteristics" along which
"information" propagates \rightarrow in our case:
in "downstream" direction (i.e. w.r.t. flow)



▷ num. solution: try method of lines

$$\frac{du_i}{dt} + v \frac{u_{i+1} - u_{i-1}}{2h} = 0$$

time discretization ψ , Euler scheme

$$u_i^{(n+1)} = u_i^{(n)} - v \frac{u_{i+1}^{(n)} - u_{i-1}^{(n)}}{2h} \Delta t$$

→ violently unstable!

→ reason:

-) characteristics propagate downstream
-) information to update u_i taken from upstream (u_{i-1}) and downstream (u_{i+1}) side

→ cur:

use one-sided approximations for
spatial derivatives

$$\frac{du_i}{dt} + v \underbrace{\frac{u_i - u_{i-1}}{h}}_{(*)} = 0$$

"upwind" differencing \rightarrow stable!

\rightarrow caveat: depends on sign of v
 v negative \nearrow

\triangleright problem: solution smoothed out \rightarrow "numerical diffusion"

recall (*):

$$\frac{u_i - u_{i-1}}{h} = \frac{u_{i+1} - u_{i-1}}{2h} - \frac{u_{i+1} - 2u_i + u_{i-1}}{2h}$$

$$\therefore \frac{du_i}{dt} + v \frac{u_{i+1} - u_{i-1}}{2h} = \frac{vh}{2} \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

$$\text{recall: } \left(\frac{\partial^2 u}{\partial x^2} \right)_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

$$\text{def: } D := \frac{v h}{2} \quad \wedge$$

$$\frac{\partial u}{\partial t} + v \cdot \frac{\partial u}{\partial x} = D \frac{\partial^2 u}{\partial x^2}$$

\rightarrow num. diffusion decreases for $h \rightarrow 0$,
increases with v

\triangleright additional stability requirement: small enough Δt

if $\Delta t > \frac{h}{v}$, update of u_i would have
to include information from u_{i-2}

\nwarrow Courant - Friedrichs - Lewi (CFL)

condition:

$$\Delta t \leq \frac{h}{v}$$

\triangleright hyperbolic conservation law, e.g.

$$\frac{\partial g}{\partial t} + \frac{\partial (Fv)}{\partial x} = 0 \quad v = v(x), \quad F = F(v)$$

→ possible discretization

$$f_i^{(n+1)} = f_i^{(n)} + \frac{\Delta t}{2\Delta x} (\hat{f}_{i-1}^{(n)} - \hat{f}_{i+1}^{(n)})$$

→ highly unstable

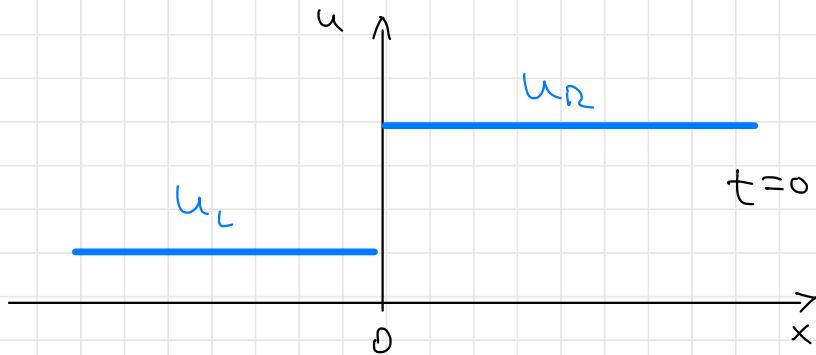
→ cur - use upwind scheme: simple direction of characteristics given by sign of mass flux

▷ problem: less obvious for systems of non-linear PDEs

→ cur: "Riemann solver" to identify local characteristic structure

II. 4 Riemann problem

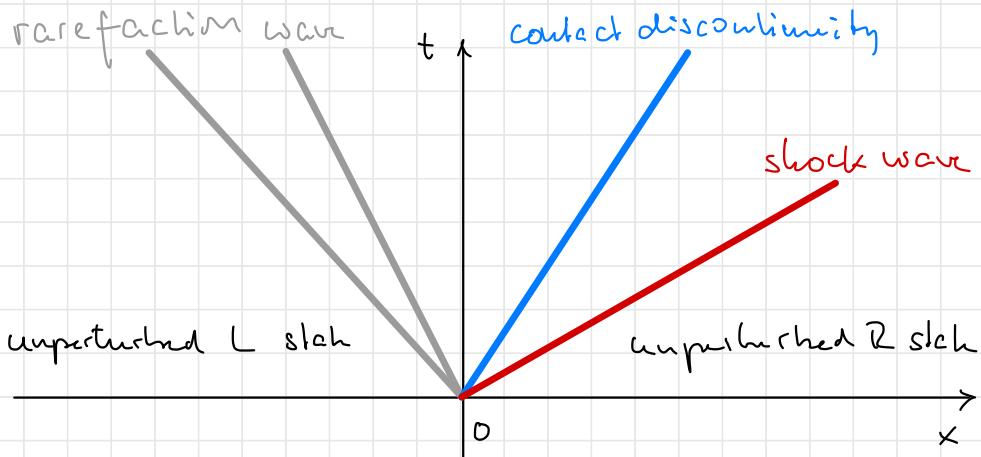
▷ initial value problem for hyperbolic system
"piecewise constant initial state
(left L and right R)



→ task: determine evolution for $t > 0$

- ▷ important special case: Riemann problem for Euler eqs.
- specify i.e. in "primitive variables"

$$\vec{u}_{L/R} = \begin{pmatrix} \rho_{L/R} \\ P_{L/R} \\ \vec{v}_{L/R} \end{pmatrix}$$
- ▷ can be solved analytically for ideal gcs
- ▷ solution always contains characteristics for 3 self-similar waves:



.) shock \rightarrow compression of fluid, irreversible conversion kin. energy \rightarrow heat,
discontinuities in ρ , v_{normal} , P , S

.) contact discontinuity \rightarrow traces original boundary between L and R fluid phases; P , v_{normal} constant,
jumps in S , S , T

.) rarefaction wave/fan \rightarrow expansion
of fluid \rightarrow smooth connection

between the 2 states are finite

spatial region \rightarrow not a discontinuity

► properties of solution:

-) Contact discontinuity: always present in middle
-) "Sandwiched" between shocks and/or rarefaction waves
-) waves propagate w/ constant speed
-) Self-similarity: if solution known
 - ① at any $t > 0 \rightarrow$ can be scaled to $t' \neq t \wedge$ fluid quantities (g^*, p^* , v^*)
 - ② $x=0$ are constant in time for $t > 0$
-) "Sod shock tube problem" for i.e.
 $\vec{v}_R = \vec{v}_L = 0$

11.5. Finite volume discretization

▷ "Riemann solvers"

▷ hyperbolic conservation laws:

$$\frac{\partial \vec{u}}{\partial t} + \nabla \cdot \vec{f} = 0$$

→ Euler eqs.

state vector $\vec{u} = \begin{pmatrix} s \\ g\vec{v} \\ g^2 e \end{pmatrix}$

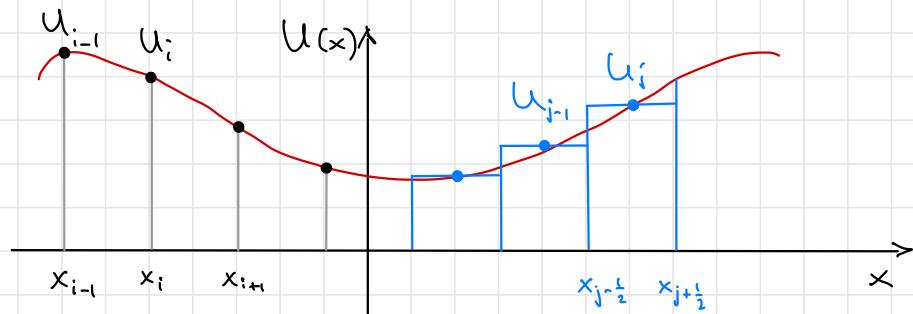
w/ specific energy $e = e_{th} + \frac{\vec{v}^2}{2}$

e_{th} : thermal energy per unit mass

flux vector $\vec{f} = \begin{pmatrix} g\vec{v} \\ g\vec{v}\vec{v}^T + P\vec{I} \\ (\rho e + P)\vec{v} \end{pmatrix}$

EoS: $P = P(s, e_{th})$

▷ finite difference vs. finite volume discretiz.



→ divide domain into "cells" → form
"cell averages"

$$\bar{u}_i = \frac{1}{V_i} \int_{\text{Cell:}} \bar{u}(\vec{x}) dV$$

▷ update cell averages:

$$1. \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx \int_{t_n}^{t_{n+1}} dt \left(\frac{\partial \bar{u}}{\partial t} + \frac{\partial \bar{F}}{\partial x} \right) = 0$$

$$\approx \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx [\bar{u}(x, t_{n+1}) - \bar{u}(x, t_n)]$$

$$+ \int_{t_n}^{t_{n+1}} dt \left[\bar{F}(x_{i+\frac{1}{2}}, t) - \bar{F}(x_{i-\frac{1}{2}}, t) \right] = 0$$

$$\approx \Delta x \left[u_i^{(n+1)} - u_i^{(n)} \right]$$

$$+ \int_{t_n}^{t_{n+1}} dt \left[\bar{F}(x_{i+\frac{1}{2}}, t) - \bar{F}(x_{i-\frac{1}{2}}, t) \right] = 0$$

2. $\vec{F}(x_{i+\frac{1}{2}}, t)$ for $t > t_n$

\rightarrow given by solution of Riemann problem

with $\vec{U}_L = u_i^{(n)}$ and $\vec{U}_R = u_{i+1}^{(n)}$

\rightarrow @ interface: solution independent of time

$$\bar{F}(x_{i+\frac{1}{2}}, t) = \vec{F}_{i+\frac{1}{2}}^* = \bar{F}_{\text{Riemann}}(\vec{U}_i^{(n)}, u_{i+1}^{(n)})$$

3. explicit update formula

$$\vec{U}_i^{(n+1)} = \vec{U}_i^{(n)} + \frac{\Delta t}{\Delta x} \left[\vec{F}_{i-\frac{1}{2}}^* - \vec{F}_{i+\frac{1}{2}}^* \right]$$



Godunov 1959 → "Godunov scheme"

▷ no approximation → exact solution for arbitrary long Δt ? → NO!

1. Riemann problems not independent because of finite Δx → but ok for small enough Δt :

$$\Delta t \leq \frac{\Delta x}{c_{\max}} < \begin{array}{l} \text{maximum wave speed} \\ \text{in Riemann solution} \end{array}$$

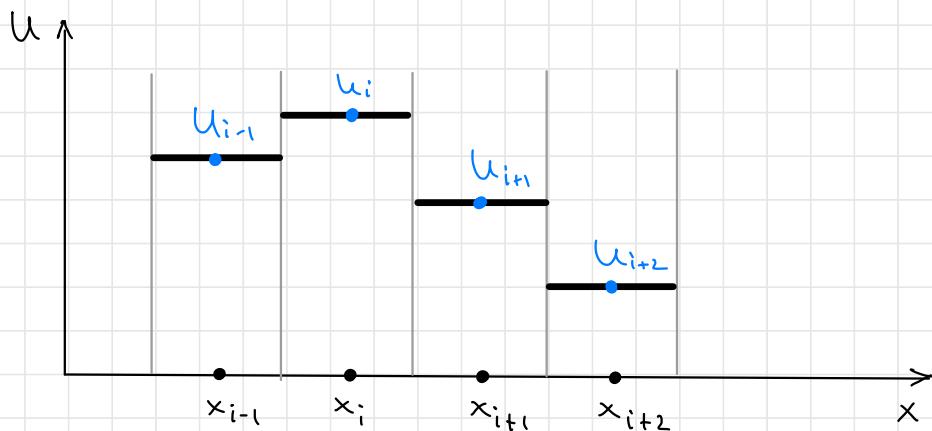
→ CFL criterion

2. after finding Δt → solution mapped back to cells → error

11. 6. REA schemes

⇒ "Reconstruction - Evolve - Average" (REA):

1. **Reconstruct:** extrapolate from cell averages
to cell edges → simplest version:
piecewise constant



2. **Evolve:** by Δt with Godunov approach
→ choose Δt smaller than CFL condition
3. **Average:** resulting state back to cell
averages → in practice given by fluxes
→ repeat cycle

$$\triangleright \text{for } 2: \bar{F}^* = \bar{F}_{\text{Riemann}}(\bar{u}_1, \bar{u}_n)$$

→ exact or approximate Riemann solution

11.7. Extension to multiple dimensions

▷ Euler in 3D

$$\partial_t \begin{pmatrix} g \\ gu \\ gv \\ gw \\ ge \end{pmatrix} + \partial_x \begin{pmatrix} gu \\ gu^2 + p \\ guv \\ gvw \\ gu(ge + p) \end{pmatrix} + \partial_y \begin{pmatrix} gv \\ guv \\ gv^2 + p \\ gvw \\ gv(ge + p) \end{pmatrix}$$

$$+ \partial_z \begin{pmatrix} gw \\ gvw \\ gvw \\ gw^2 + p \\ gw(ge + p) \end{pmatrix} = 0$$

$$e = e_{\text{th}} + \frac{1}{2}(u^2 + v^2 + w^2), \text{ ideal gas:}$$

$$p = (\gamma - 1) g e_{\text{th}}$$

$$\triangleright \partial_t \bar{u} + \partial_x \bar{F} + \partial_y \bar{G} + \partial_z \bar{H} = 0$$

11. 7.1 Dimensional splitting

▷ consider:

$$\partial_t \vec{U} + \partial_x \vec{F} = 0$$

$$\partial_t \vec{U} + \partial_y \vec{G} = 0$$

$$\partial_t \vec{U} + \partial_z \vec{H} = 0$$

▷ "augmented" 1D problems: transverse velocities still appear, but spatial differentiation in only 1 direction \rightarrow as simple as 1D problem

▷ common method to solve 1D problem

\rightarrow formally: solution operators $X(\Delta t)$, $Y(\Delta t)$, $Z(\Delta t)$

\sim full time advance

$$\vec{u}^{(n+1)} \approx Z(\Delta t) Y(\Delta t) X(\Delta t) \vec{u}^n$$

→ exact for linear advection problem,

but only 1st order approximation in general

▷ higher-order split schemes, e.g.

2nd order in 2D:

$$\vec{u}^{(n+1)} \approx \frac{1}{2} [X(\Delta t) Y(\Delta t) + Y(\Delta t) X(\Delta t)] \vec{u}^n$$

or:

$$\vec{u}^{(n+1)} \approx X\left(\frac{\Delta t}{2}\right) Y(\Delta t) X\left(\frac{\Delta t}{2}\right) \vec{u}^n$$

2nd order in 3D

$$\vec{u}^{(n+1)} \approx X\left(\frac{\Delta t}{2}\right) Y\left(\frac{\Delta t}{2}\right) Z(\Delta t) Y\left(\frac{\Delta t}{2}\right) X\left(\frac{\Delta t}{2}\right) \vec{u}^n$$

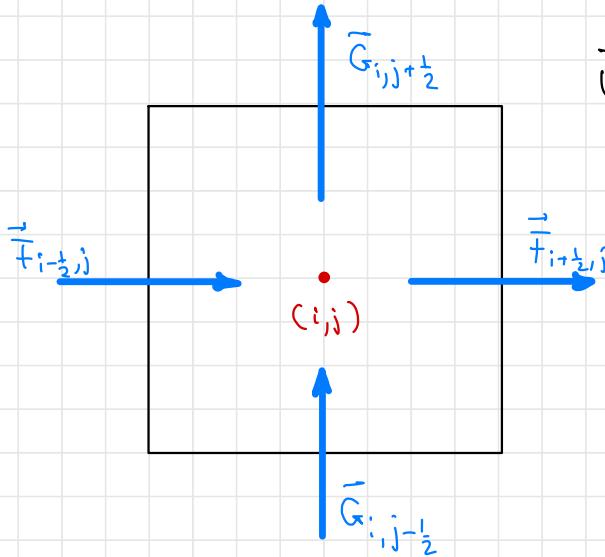
▷ note: each of few 1D steps updates \vec{u}

in respective direction over full Δt

→ "sweeps" → updates applied sequentially

11. 7. 2. Unsplit schemes

- ▷ Simultaneous flux update to cells



$$\bar{u}_{i,j}^{(n+1)} = \bar{u}_{i,j}^{(n)}$$

$$+ \frac{\Delta t}{\Delta x} (\bar{F}_{i-\frac{1}{2},j} - \bar{F}_{i+\frac{1}{2},j})$$

$$+ \frac{\Delta t}{\Delta x} (\bar{G}_{i,j-\frac{1}{2}} - \bar{G}_{i,j+\frac{1}{2}})$$

- ▷ Advantage : can be applied to irregular grids

- ▷ General expression

$$\bar{u}_{i,j}^{(n+1)} = \bar{u}_{i,j}^{(n)} - \frac{\Delta t}{\sqrt{\sigma}} \int_{\partial V} \bar{F} \cdot d\bar{S}$$

- ▷ Disadvantage : more memory needed, higher-order accuracy more complicated

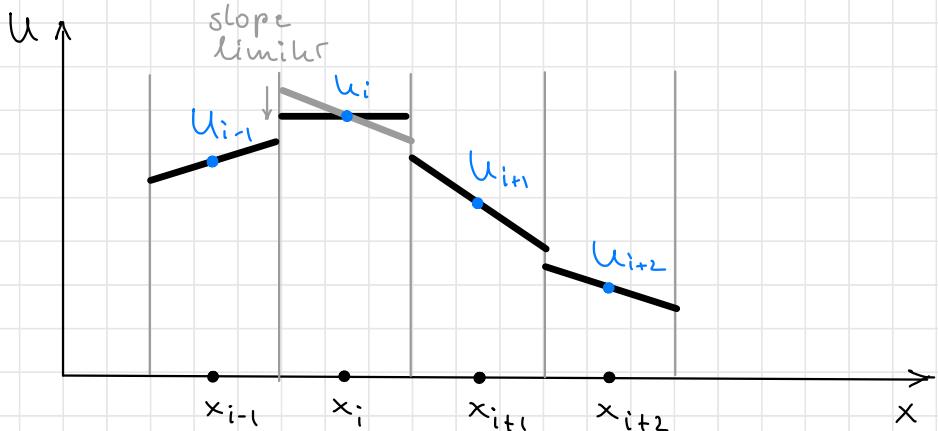
11.8. Higher-order accuracy

- ▷ faster convergence to exact solution
- ▷ numerical solution with N points:
$$g(x_i), i=1, \dots, N \rightarrow \text{error?} \rightarrow \text{e.g. } L_1 - \text{norm}$$
$$L_1 = \frac{1}{N} \sum_{i=1}^N |g_i - g(x_i)|$$

→ Should decrease with increasing N

→ rate of convergence different for schemes:

 - 1st order accurate: $L_1 \propto N^{-1}$
 - 2nd order accurate: $L_1 \propto N^{-2}$- ▷ basic Godunov scheme: only 1st order
- 2nd order: replace piecewise constant reconstruction by piecewise linear



- 1) estimate gradients in each cell (simple finite difference scheme)
- 2) apply slope limiting to avoid introduction of new extrema
- 3) estimate L and R states from gradients ∇g by extrapolation

$$g_{i+\frac{1}{2}}^L = g_i + (\nabla g)_i \frac{\Delta x}{2}$$

$$g_{i+\frac{1}{2}}^R = g_{i+1} - (\nabla g)_{i+1} \frac{\Delta x}{2}$$

- 4) for reaching 2nd order in time and for stability \sim flux estimate in

middle of step Δt

$$\underline{g}_{i+\frac{1}{2}} = g_i + (\nabla g)_i \frac{\Delta x}{2} + \left(\frac{\partial g}{\partial t} \right)_i \frac{\Delta t}{2}$$

$$\underline{g}_{i+\frac{1}{2}}^2 = g_{i+1} + (\nabla g)_{i+1} \frac{\Delta x}{2} + \left(\frac{\partial g}{\partial t} \right)_{i+1} \frac{\Delta t}{2}$$

→ generally: $\vec{u}_i, (\partial_x \vec{u})_i, (\partial_t \vec{u})_i$

→ how to determine $\partial_t \vec{u}$?

→ exploit Jacobian matrix

$$\partial_t \vec{u} = - \partial_x \vec{F}(\vec{u}) = - \frac{\partial \vec{F}}{\partial \vec{u}} \cdot \partial_x \vec{u}$$

$$= - \underbrace{A(\vec{u})}_{\text{Jacobian}} \cdot \partial_x \vec{u}$$

$$(\partial_t \vec{u})_i = - A(\vec{u}_i) \cdot (\partial_x \vec{u})_i$$

$$\sim \vec{u}_{i+\frac{1}{2}} = \vec{u}_i + \left[\frac{\Delta x}{2} - \frac{\Delta t}{2} A(\vec{u}_i) \right] \cdot (\partial_x \vec{u})_i$$

R analogies

"MUSCL - Hancock scheme"

▷ higher-order \rightarrow "piecewise parabolic method"
(PPM) \rightarrow parabolic shapes in cells

ENO/WEENO schemes: higher-order reconstr.
with polynomials \rightarrow more neighboring cells
involved in determining coefficients

12. Smoothed particle hydrodynamics

- ▷ Key idea: approximate continuum mechanics
 - ↳ particle-based discretization
- ▷ SPH: Lagrangian method → follows fluid elements → local resolution automatically follows mesh
- ▷ SPH: excellent conservation properties: energy, linear & angular momentum, mass, entropy
 - conserve simultaneously
- ▷ potential problem of Lagrangian methods:
 - grid-based approaches difficult in multi-D
 - frequent re-gridding required to avoid grid-tangling
 - would obscure Lagrangian nature of scheme
- ▷ SPH: mesh-free method

12.1. The SPH concept

- ▷ Lucy (1977), Gingold & Monaghan (1977)
- ▷ recall Euler eqs. in Lagrangian form

$$\frac{D\vec{s}}{Dt} + g \nabla \cdot \vec{v} = 0 \quad \frac{D}{Dt} = \frac{\partial}{\partial t} + \vec{v} \cdot \nabla$$

$$\frac{D\vec{v}}{Dt} + \frac{\nabla p}{\rho} = \vec{f}$$

\vec{f} : specific external force,
e.g. gravity $\vec{f} = -\nabla \phi$

$$\frac{D\varepsilon_{\text{tot}}}{Dt} + \frac{1}{\rho} \nabla \cdot \vec{P} \vec{v} = \vec{v} \cdot \vec{f} \quad \varepsilon_{\text{tot}} = \varepsilon_{\text{th}} + \frac{(\vec{v})^2}{2}$$

$$\frac{D\varepsilon_{\text{th}}}{Dt} + \frac{P}{\rho} \nabla \cdot \vec{v} = 0$$

- ▷ eqs. of motion for fluid package i

$$\frac{d\vec{x}_i}{dt} = \vec{v}_i \quad (*)$$

$$\frac{d\vec{v}_i}{dt} + \frac{1}{\rho_i} \nabla P_i = \vec{f} \quad (**)$$

→ particle-based analog of momentum eq.

- ▷ mass conservation, total energy balance automatically accounted for: particle masses and specific entropies stay constant
- but shocks produce entropy → need to introduce "artificial viscosity" η
additional term in momentum eq. +
need to follow energy balance eq.,
also required for determining η from EoS
- ▷ fluid properties only via $\frac{1}{\rho_i} P_i$ -term;
 P_i : hydrodynamical pressure assigned to particle
- ▷ how to estimate $\nabla \rho$ at single point particle?
- ▷ SPH: smoothly smear out hydrodyn. quantities from particle location

12. 2. Kernel interpolants

- ▷ kernel summation interpolant for estimating density
- ▷ for field $\vec{F}(\vec{r}) \rightarrow$ def. smooth interpolated $\vec{F}_s(\vec{r})$ through convolution with kernel $w(\vec{r}, h)$

$$\vec{F}_s(\vec{r}) = \int \vec{F}(\vec{r}') w(\vec{r} - \vec{r}', h) d\vec{r}'$$

h : characteristic width

- ▷ normalization:

$$\int_0^\infty w(\vec{r} - \vec{r}', h) d\vec{r}' = 1$$

$$\lim_{h \rightarrow 0} w(\vec{r} - \vec{r}', h) = \delta_D(\vec{r} - \vec{r}')$$

- ▷ w symmetric, sufficiently smooth (at least 2x diff'able)
 - example: Gaussian \rightarrow not optimal

▷ better choice kernels with finite support,

e.g. Cubic Spline with $W(r, h) = w(q := \frac{r}{2h})$

$$w_{3D}(q) = \frac{8}{\pi} \begin{cases} 1 - 6q^2 + 6q^3 & 0 \leq q < \frac{1}{2} \\ 2(1-q)^3 & \frac{1}{2} \leq q \leq 1 \\ 0 & q > 1 \end{cases}$$

- ▷ suppose $F_i = f(\vec{r}_i)$, points @ \vec{r}_i have mass m_i , during \vec{s}_i \sim associated volume element $\Delta r_i^3 \sim \frac{m_i}{\vec{s}_i}$
- ▷ if points sample kernel volume densely enough:

$$F_s(\vec{r}) \approx \sum_j \frac{m_j}{\vec{s}_j} F_j W(\vec{r} - \vec{r}_j, h)$$

→ order of accuracy in 1D, equal spacing d
of points $\rightarrow h = d \sim 2^{\text{nd}}$ order accuracy

→ more complicated for irregular multi-D
particle distributions, but need $h \geq d$

→ typically ~ 33 neighbors within smoothing length in 3D

▷ $f_s(\vec{r})$ defined everywhere, and differentiable thanks to kernel

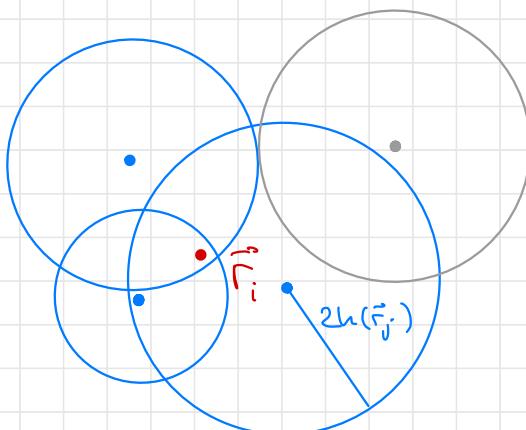
▷ e.g. $f(\vec{r}) = g(\vec{r}) \approx$

$$g_s(\vec{r}) \approx \sum_j m_j w(\vec{r} - \vec{r}_j, h)$$

▷ possible: Spatially variable $h = h(\vec{r}, t)$

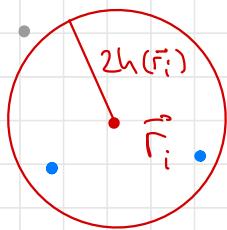
→ 2 options: consider g @ particle i

1) "scatter approach": $w(\vec{r}_i - \vec{r}_j, h(\vec{r}_j)) := w_{ij}(h_j)$



advantage: $\int_{\text{V}} g_s(\vec{r}) d\vec{r} = \sum_i m_i$

2) "gather approach": $w(\vec{r}_i - \vec{r}_j, h(\vec{r}_i)) := w_{ij}(h_i)$



advantage: simple

- choice introduces asymmetry → violates Newton's 3rd law
- symmetrization

$$h_{ij} = \frac{1}{2} (h(\vec{r}_i) + h(\vec{r}_j))$$

$$\sim \bar{w}_{ij} = w_{ij} \left[\frac{1}{2} (h_i + h_j) \right]$$

- split estimate of density of particle i

$$g_i = \sum_{j=1}^N m_j w_{ij}(h_i)$$

→ advantage of kernels w/ finite support:

summation restricted to N nbr neighbours

within $2h \cap$ cost of summation $\Theta(N_{nbr}N)$

instead of $\Theta(N^2)$ for Gaussian kernel

- ▷ for Monte-Carlo sampling: accuracy problematic → but SPH does not sample in Poissonian fashion → distances between particles equilibrium due to $P \approx$ interpolation errors ↓, but still significant source of noise

- ▷ other SPH quantities: A

$$A(\vec{r}) = \sum_{j=1}^N \frac{A_j(\vec{r}_j)}{S_j(\vec{r}_j)} g_j(r_j)$$

$$\approx A(\vec{r}) = \sum_{j=1}^N \frac{A_j(\vec{r}_j)}{S_j(\vec{r}_j)} m_j w(\vec{r} - \vec{r}_j, h)$$

- ▷ differential operators in SPH:

-) gradients, e.g.

$$\nabla P(\vec{r}_i) = \sum_{j=1}^N \frac{\rho_j}{S_i} m_j \nabla_i W(\vec{r}_i - \vec{r}_j, h)$$

$$(\nabla \cdot \vec{v})_i = \sum_{j=1}^N \frac{m_j}{S_i} \vec{v}_j \cdot \nabla_i W(\vec{r}_i - \vec{r}_j, h)$$

→ alternative form → $\vec{g} \nabla \cdot \vec{v} = \nabla \cdot (\vec{g} \vec{v}) - \vec{v} \cdot \nabla \vec{g}$

$$\begin{aligned} (\nabla \cdot \vec{v}) &= \frac{1}{S_i} [\nabla_i \cdot (\vec{g} \vec{v})_i - \vec{v}_i \cdot (\nabla \vec{g})_i] \\ &= \frac{1}{S_i} \sum_{j=1}^N m_j (\vec{v}_j - \vec{v}_i) \cdot \nabla_i W(\vec{r}_j - \vec{r}_i, h) \end{aligned}$$

→ advantage: guarantees zero divergence
if all particle velocities equal

12.3. Variational derivation of SPH

▷ Euler eqs. can be derived from Lagrangian via variational principles (Eckart 1960)

$$L = \int g \left(\frac{\dot{\vec{v}}^2}{2} - e_{th} \right) dV$$

▷ discretized eq. of motion (Springer & Haugrid 2002)

→ discretize Lagrangian → derive SPH eqs.
of motion

▷ discretize L in terms of fluid particles
of mass m_i

$$L_{\text{SPH}} = \sum_i \left(\frac{1}{2} m_i \vec{v}_i^2 - m_i e_{\text{th},i} \right)$$

↳ implicit assumption: e_{th} can be
expressed through specific thermodynamic
entropy A_i :

$$P_i = A_i g_i^\gamma = (\gamma - 1) g_i e_{\text{th},i}$$

γ : adiabatic index

isentropic flows: $A_i = \text{const}$

⇒ define:

$$e_{\text{th},i} = A_i \frac{g_i^{\gamma-1}}{\gamma-1}$$

→ h_i needed to estimate g_i

- ~ allows for adaptive kernel widths
- to keep # particles in kernel ~ const.
- ~ require mass inside kernel $g h^3 = \text{const.}$
- $\Rightarrow g_i = g_i(\bar{r}_1, \bar{r}_2, \dots, \bar{r}_N, h_i)$
- $\Rightarrow h_i = h_i(\bar{r}_1, \bar{r}_2, \dots, \bar{r}_N)$

> Euler-Lagrange eq. \rightarrow eqs. of motion

$$\frac{d}{dt} \frac{\partial L_{\text{SPH}}}{\partial \dot{r}_i} - \frac{\partial L_{\text{SPH}}}{\partial r_i} = 0$$

$$m_i \frac{d\vec{v}_i}{dt} = - \sum_{j=1}^N m_j \frac{\vec{r}_{ij}}{g_j^2} \frac{\partial g_j}{\partial \vec{r}_i} \quad (*)$$

$$\frac{\partial g_j}{\partial \vec{r}_i} = \nabla_i g_j + \boxed{\frac{\vec{r}_{ij}}{\partial h_j} \frac{\partial h_j}{\partial \vec{r}_i}} \quad (**)$$

from $g_j h_j^3 = \text{const}$ | differentiable w.r.t.
 \vec{r}_i

$$\rightarrow \boxed{\frac{\partial g_i}{\partial h_j} \quad \frac{\partial h_j}{\partial r_i}} \left[1 + \frac{3}{h_j} g_i \left(\frac{\partial g_j}{\partial h_j} \right)^{-1} \right] = -\nabla_i g_j$$

with $(**)$ \rightarrow

$$\frac{\partial g_j}{\partial r_i} = \left(1 + \frac{h_j}{3g_j} \frac{\partial g_j}{\partial h_j} \right)^{-1} \nabla_i g_j \quad (***)$$

$\underbrace{\qquad\qquad\qquad}_{:= f_j}$

$$g_j = \sum_{k=1}^N m_k w_{jk}(h_j)$$

$$\nabla_i g_j = m_i \nabla_i w_{ji}(h_j) + \delta_{ij} \sum_{k=1}^N m_k \nabla_i w_{ik}(h_i)$$

(4*)

(4*) \rightarrow $(****)$ \rightarrow (*)

$$\boxed{\frac{d\bar{v}_i}{dt} = - \sum_{j=1}^N m_j \left[f_i \frac{\bar{r}_i}{\bar{g}_i^2} \nabla_i w_{ij}(h_i) + f_j \frac{\bar{r}_j}{\bar{g}_j^2} \nabla_i w_{ij}(h_j) \right]}$$

\triangleright converts system of PDEs \rightarrow set of ODEs

- ▷ only momentum eq. has to be solved explicitly
- ▷ later: introduce viscosity (needed for shocks)
 - requires to evolve 1 thermodynamic variable per particle (thermal energy or entropy) → $e_{th,i}$ instead A

$$e_{th,i} = A_i \frac{\gamma^{5/4} - 1}{\gamma - 1} \quad | \quad \frac{d}{dt}$$

$$\boxed{\frac{de_{th,i}}{dt} = \frac{p_i}{\rho_i^2} \sum_j \vec{v}_j \cdot \frac{\partial g_i}{\partial \vec{r}_i}}$$

$$\begin{aligned} & \stackrel{(***)}{=} f_i \frac{p_i}{\rho_i^2} \sum_j m_j (\vec{v}_i - \vec{v}_j) \cdot \nabla w_{ij} (u_i) \\ & \stackrel{(4*)}{=} \end{aligned}$$

energy equation of SPH

- ▷ readily fulfills conservation laws of energy, momentum, angular momentum

12.4 Artificial viscosity

- ▷ Specific entropy always increases @ shock front
 - inviscid description no longer valid
- ▷ introduce dissipation @ shocks
 - artificial viscosity
 - friction force to damp motions of particles
 - broadens shock → resolvable & describable in terms of differential eqs.
- ▷ how to ensure right amount of dissipation?
 - introduce viscosity into dynamics in "conservative" fashion
- ▷ add viscous force to eq. of motion:

$$\frac{d\vec{v}_i}{dt} \Big|_{visc} = - \sum_{j=1}^N m_i \nabla_{ij} J_i \vec{w}_{ij}$$

(scalar) viscosity factor ∇_{ij} \rightarrow symmetric

$\text{u.r.t } i, j \Rightarrow$ viscous force between 2 interacting particles axisymmetric

\rightarrow lin. & angular momentum conserved

\triangleright total energy conservation requires

\rightarrow in terms of entropy:

$$\frac{d A_i}{dt} \Big|_{\text{visc}} = \frac{1}{2} \sum_{j=1}^{N-1} \underbrace{m_j \nabla_{ij} \vec{v}_{ij} \cdot \nabla_i \bar{w}_{ij}}_{\vec{v}_{ij} := \vec{v}_i - \vec{v}_j}$$

\rightarrow in terms of specific thermal energy

$$\frac{d e_m}{dt} \Big|_{\text{visc}} = \frac{1}{2} \sum_{j=1}^N m_j \nabla_{ij} \vec{v}_{ij} \cdot \nabla_i \bar{w}_{ij}$$

\triangleright how to ensure viscosity active only @ shocks?

\rightarrow viscosity factor: common choice \rightarrow combination of bulk & von Neumann-Richtmyer viscosity

$$\nabla_{ij} = \begin{cases} \frac{1}{\delta_{ij}} [-\alpha c_{ij} \mu_{ij} + \beta \mu_{ij}^2] & \text{if } \vec{v}_{ij} \cdot \vec{r}_{ij} < 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{ij} := \frac{h_{ij} v_{ij} \cdot \vec{r}_{ij}}{|r_{ij}|^2 + \varepsilon h_{ij}}$$

$(\dots)_{ij}$: arithmetic mean
for particles i, j

c → speed of sound

$$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$$

parameters α, β set strength of viscosity

→ common choice $\alpha \approx 0.5 - 1.0$

$$\beta = 2\alpha$$

$\varepsilon \approx 0.01$ protects against singularities
if particles get very close

▷ viscosity vanishes for solid-body rotation,
but not for shear flow → cured by
"Balsara switch"

$$\frac{1}{2} (f_i^{AV} + f_j^{AV}) \nabla \cdot \vec{j}$$

$$f_i^{AV} := \frac{|\nabla \cdot \vec{v}|_i}{|\nabla \cdot \vec{v}|_i + |\nabla \times \vec{v}|_i}$$

12.5. Advantages / disadvantages of SPT

13. Finite element methods

- ▷ complex geometries \rightarrow FEM: spatial discretization on "irregular" unstructured grid, e.g. constructed from triangles (2D), tetrahedra (3D)
 - \rightarrow shapes arbitrary, but must fit entire volume V
 - \rightarrow "finite elements"
- ▷ no obvious choice for higher-order finite differences

13.1 The Galerkin method

- ▷ example: linear elliptic PDE:

$$L[y(\vec{x})] = f(\vec{x})$$

\rightarrow to be solved for $y(\vec{x})$ in volume V with surface $\partial V \rightarrow$ set Neumann or Dirichlet b.c. on ∂V , e.g. $y=0 \quad \partial V$

→ Simplest example: Poisson eq.

$$\nabla^2 y(\vec{x}) = f(\vec{x})$$

▷ Galerkin: devise basis fcts.

$$\varphi_i(\vec{x}) \quad i=1, \dots, n$$

→ find linear combination of $\varphi_i(\vec{x})$ that best approximates solution

$$y(\vec{x}) \approx \sum_{i=1}^n x_i \varphi_i(\vec{x})$$

x_i : numerical constants

→ task: determine x_i

$\varphi_i(\vec{x})$: arbitrary, but linearly independent,
zero @ boundary if b.c. chosen

▷ define residual

$$R[y(\vec{x})] = L[y(\vec{x})] - f(\vec{x})$$

▷ Galerkin: determine α_i such that $R[y(\vec{x})]$ linearly orthogonal to every basis fct.

$\varphi_i(\vec{x})$ u.r.t L^2 inner product

$$\langle f(\vec{x}), g(\vec{x}) \rangle \equiv \int \limits_{\Delta} f(\vec{x}) g(\vec{x}) d^3x$$

$$\text{~} \nabla \int \limits_{\Delta} \varphi_i(\vec{x}) R[y(\vec{x})] d^3x = 0 \quad \forall i \in [1, n]$$

▷ determine α_i

$$\int \limits_{\Delta} \varphi_i(\vec{x}) \left\{ L[y(\vec{x})] - f(\vec{x}) \right\} d^3x = 0 \quad \forall i$$

$$\int \limits_{\Delta} \varphi_i(\vec{x}) \left\{ \sum_{k=1}^n \alpha_k L[\varphi_k(\vec{x})] - f(\vec{x}) \right\} d^3x = 0 \quad \forall i$$

$$\text{~} \nabla \sum_{k=1}^n \alpha_k \int \limits_{\Delta} \varphi_i(\vec{x}) L[\varphi_k(\vec{x})] d^3x \quad (*)$$

$$- \int \limits_{\Delta} \varphi_i(\vec{x}) f(\vec{x}) d^3x = 0 \quad \forall i$$

▷ choose $\varphi_i(\vec{x})$ so that integrals can be computed analytically
e.g. polynomials, sine / cosine waves...

$$\rightarrow M_{ik} := \int \varphi_i(\vec{x}) \cup [\varphi_k(\vec{x})] d^3x$$

$$b_i := \int \varphi_i(\vec{x}) f(\vec{x}) d^3x$$

$$\sim \sum_{k=1}^n M_{ik} x_k = b_i \quad \forall i \in [1, n]$$

→ matrix equation

$$\underline{M} \cdot \vec{x} = \vec{b}$$

→ simplified if \underline{M} sparse \rightarrow use localized basis fcts.

▷ allow for piecewise basis fcts.

trick: integrate by parts $\therefore L = \nabla^2$

$$M_{ik} = \int_V \varphi_i(\vec{x}) \nabla^2 \varphi_k(\vec{x}) d^3x$$

$$= \int_{\partial V} \varphi_i(\vec{x}) \nabla \varphi_k(\vec{x}) \cdot \vec{n} dS - \int_V \nabla \varphi_i(\vec{x}) \cdot \nabla \varphi_k(\vec{x}) d^3x$$

\vec{n} : normal vector on surface ∂V

b.c. $\varphi_i(\vec{x}) = 0 \quad @ \partial V \quad \sim$ surface integral
variables \rightarrow

$$M_{ik} = - \int_V \nabla \varphi_i(\vec{x}) \cdot \nabla \varphi_k(\vec{x}) d^3x$$

\rightarrow compute M_{ik} from singly diff'able

basis fcts. \rightarrow reason: M_{ik} does not depend

on $\nabla^2 \varphi_k(\vec{x})$, but on its integral!

\rightarrow Galerkin method casts PDE into "weak" form

\rightarrow advantage: simpler piecewise polynomial basis fcts. possible, such as triangles in 1D

side remark:

$$M_{ik} = \int_V q_i(\vec{x}) \nabla^2 q_k(\vec{x}) d^3x \quad \rightarrow \text{integrate by parts}$$

use: $\nabla \cdot (u \vec{V}) = u \nabla \cdot \vec{V} + \vec{\nabla} u \cdot \vec{V}$

$$\int_V \nabla \cdot (u \vec{V}) d^3x = \int_V u \vec{V} \cdot \vec{n} dS = \int_V u \nabla \cdot \vec{V} d^3x$$

divergence theorem

$$+ \int_V \vec{\nabla} u \cdot \vec{V} d^3x$$

$$q_i(\vec{x}) = u \quad \nabla q_k(\vec{x}) = \vec{V}$$

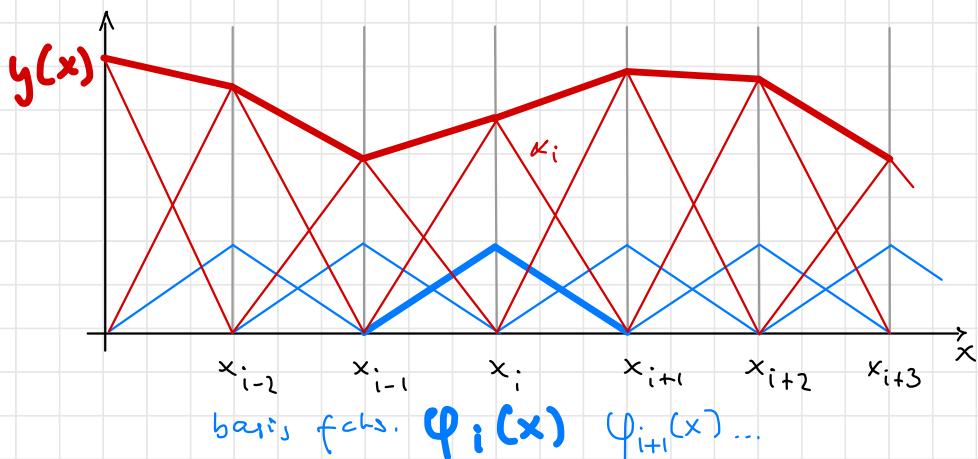
▷ example: piecewise linear approximation to Poisson eq. in 1D

$$\frac{\partial^2 y(x)}{\partial x^2} = f(x)$$

→ discretization: $x_i := x_0 + i \Delta x$

→ choose triangular basis func.

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{\Delta x} & \text{for } x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{\Delta x} & \text{for } x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

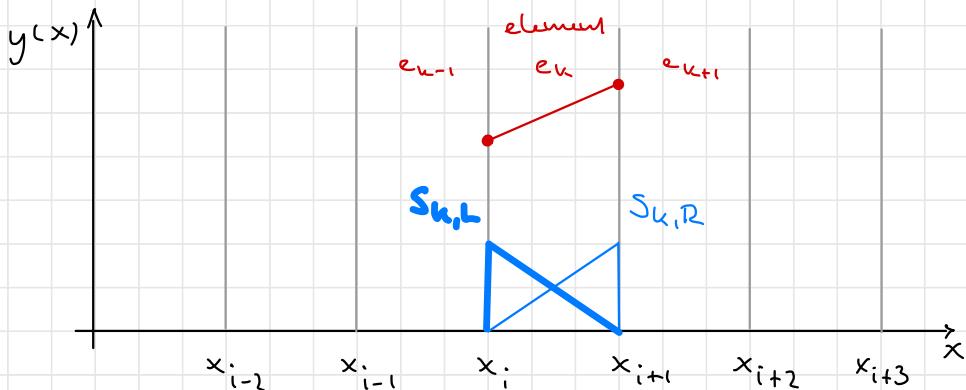


- generalize to 2D, 3D
- polynomial degree of approximation can be increased
- flexibility: irregular triangulation possible

13.2. Finite element method

- ▷ divide volume V into cells → "elements"
- ▷ express PDE on elements, e.g. with Galerkin method
 - choose basis fcts to describe interaction between cells
- ▷ 1D example: regular grid $x_i = x_0 + i \Delta x$
 - cell walls

cell k : between x_i and x_{i+1} \triangleq element



→ approximate lin. fct. between x_i and x_{i+1}
 constructed from superposition of 2 shape
 functions

$$S_{k,L}(x) = \begin{cases} \frac{x_{i+1} - x}{\Delta x} & \text{for } x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

$$S_{k,R}(x) = \begin{cases} \frac{x - x_i}{\Delta x} & \text{for } x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

→ $S_{k,L}$ and $S_{k,R}$ determine form of element

→ "glue together" elements of neighbouring cell
 to enable interaction: combine $s_{k,L}$ and
 $s_{k,R}$ into single basis fct., $\varphi_i(x)$ has to
 be continuous: $s_{k,L}(x=x_i) = s_{k-1,R}(x=x_i)$

$$\curvearrowleft \varphi_i(x) = s_{i,L}(x) + s_{i-1,R}(x)$$

→ triangular basis fcts.

→ application to 1D Poisson eq.

$$\frac{\partial^2 y(x)}{\partial x^2} = f(x)$$

→ triangular $\varphi_i(x)$

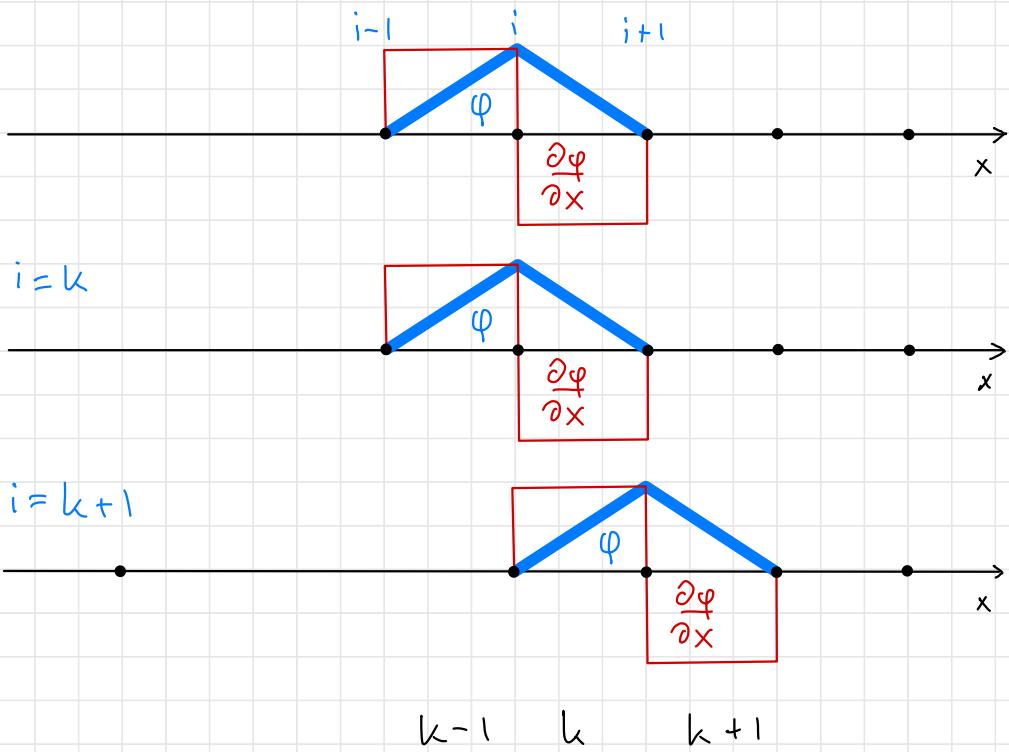
$$y(x) \approx \sum_{i=1}^n \alpha_i \varphi_i(x)$$

→ determine α_i via

$$\sum_{k=1}^n M_{ik} \alpha_k = b_i \quad \text{with} \quad b_i = \int f(x) \varphi_i(x) dx$$

$$M_{inh} = - \int_v \frac{\partial \varphi_i(x)}{\partial x} \frac{\partial \varphi_k(x)}{\partial x} dx$$

$$\frac{\partial \varphi_i(x)}{\partial x} = \begin{cases} \frac{1}{\Delta x} & \text{for } x \in [x_{i-1}, x_i] \\ -\frac{1}{\Delta x} & \text{for } x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$



$$\nabla \mu_{ik} = \begin{cases} -\frac{2}{\Delta x} & \text{for } i=k \\ \frac{1}{\Delta x} & \text{for } i=k \pm 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{ik} = \left(\frac{1}{\Delta x}\right)^2 \begin{pmatrix} -1 & 1 & & & & & 0 \\ 1 & -2 & 1 & & & & \\ - & \ddots & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & 1 & -2 & 1 \\ 0 & & & 1 & -2 & 1 & & \\ & & & & 1 & -1 & & \text{boundaries!} \end{pmatrix}$$

\Rightarrow equivalent to simple finite-difference approx
to 1D Poisson eq.

\triangleright 2D example: cover arbitrary (irregular) area

by triangles \rightarrow "triangulation" \rightarrow complete

\rightarrow each triangle k described by 3 nodes

$\bar{x}_{kl}, l=1,2,3 \rightarrow$ point inside Δ : 2 coordinates

$\xi, \eta :$

$$\vec{x} = \vec{x}_1 + \xi (\vec{x}_2 - \vec{x}_1) + \eta (\vec{x}_3 - \vec{x}_1)$$

lin. fct. on Δ :

$$g(\xi, \eta) = g_1 + g_2 \xi + g_3 \eta$$

\rightarrow def. shape fct. : (Q one node
 0 @ other nodes

$$S_{k,1}(\xi, \eta) = 1 - \xi - \eta$$

$$S_{k,2}(\xi, \eta) = \xi$$

$$S_{k,3}(\xi, \eta) = \eta$$

$\rightarrow q_i(\vec{x})$: sum of shape fcts., co-joined
 @ each vertex x_i

> 3D example: similar, but volume filled

4, tetrahedra \rightarrow 3 variables: ξ, η, λ

\rightarrow 4 shape fcts.

13.3. Structural mechanics

- ▷ engineering: deformation of solid object filling V, surface ∂V under stress
→ more complex elliptic eqs.

- ▷ **deformation** @ point \vec{x} $\hat{\vec{x}} \approx$ small displacement
 $\delta\vec{x}$ to new position

$$\hat{\vec{x}}(\vec{x}) = \vec{x} + \delta\vec{x}(\vec{x})$$

- ▷ not always deformation: constant $\delta x \neq \hat{x}(\vec{x})$
→ translation

$$\begin{aligned}\delta x(x, y, z) &= x \cos \alpha + y \sin \alpha - x \\ \delta y(x, y, z) &= -x \sin \alpha + y \cos \alpha - y \\ \delta z(x, y, z) &= 0\end{aligned}\quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{rotation}$$

- ▷ define **strain tensor**

$$\epsilon_{kl} = \frac{1}{2} (\nabla_k \delta x_l + \nabla_l \delta x_k)$$

▷ deformation if any $\epsilon_{kl} \neq 0$

→ decomposition

diagonal part: stretching / compression

off-diagonal part: shear

▷ Strain causes stress

rank-2 tensor $\underline{\sigma}$

▷ in linear "elastic" regime forces reversible -

linear stress-strain relation:

$$\sigma_{ij} = \sum_{kl} C_{ijkl} \epsilon_{kl} \quad C_{ijkl}: \text{"stiffness tensor"}$$

▷ strong strain → sub-linear stress increase →

"plastic" regime

▷ design goal: stay in elastic regime inside V

→ tested with FEM

▷ material properties contained in stiffness tensor

→ assume isotropic material

→ symmetric stiffness tensor

⇒ force equilibrium: "static field equation"

$$\nabla_i \sigma_{ij}(\vec{x}) = -f_j(\vec{x}) \rightarrow \text{Einstein's summation}$$

$$C_{ijkl} \nabla_i \varepsilon_{kl}(\vec{x}) = -f_j(\vec{x})$$

$$\frac{1}{2} C_{ijkl} \nabla_i [\nabla_k \delta x_l(\vec{x}) + \nabla_l \delta x_k(\vec{x})] = -f_i(\vec{x})$$

symmetry $C_{ijll} = C_{ijlk} \wedge$

$$C_{ijlk} \nabla_i \nabla_k \delta x_l(\vec{x}) = -f_j(\vec{x})$$

→ coupled set of 3 ($j=1, 2, 3$) 2nd-order

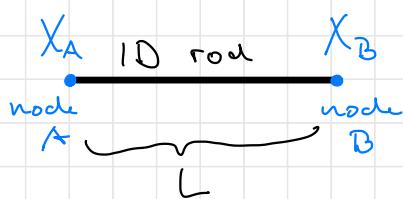
PDEs for 3 unknowns $\delta x_1(\vec{x}), \delta x_2(\vec{x}), \delta x_3(\vec{x})$

⇒ set b.c. on $\partial V \rightarrow$ solve with FEM

13.3.1 The truss model

⇒ trusses → 1D-pieces of material, length L , stiffness \sim 1D finite element → connect into 3D structure by "vertices", "nodes"

→ direct stiffness method (DSM)



before deformation: $X_A^{(0)} = 0$

$X_B^{(0)} = L$

→ 1D deformation: $X_A = X_A^{(0)} + \delta x_A$

$X_B = X_B^{(0)} + \delta x_B$

→ fix B: $\delta x_B = 0$

→ force on A: $f_A = K \delta x_A$

K : stiffness coefficient

$f_B = -K \delta x_A$

$$\begin{pmatrix} f_A \\ f_B \end{pmatrix} = \underbrace{\begin{pmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{pmatrix}}_{\text{stiffness matrix}} \begin{pmatrix} \delta x_A \\ \delta x_B \end{pmatrix} = K \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} \delta x_A \\ \delta x_B \end{pmatrix}$$

task: ext. force given → solve matrix eq.
for displacements

→ placement of 1D rod in 2D Space →

node positions: $X_A, Y_A; X_B, Y_B$

displacements: $\delta x_A, \delta y_A; \delta x_B, \delta y_B$

forces: $f_A^x, f_A^y; f_B^x, f_B^y$

$$\text{rod length: } L = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}$$

angle of rod ϑ

$$\cos \vartheta = \frac{X_B - X_A}{L} \quad \sin \vartheta = \frac{Y_B - Y_A}{L}$$

→ known initial positions $X_A^{(0)}, Y_A^{(0)} \dots$

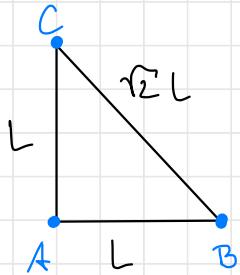
→ estimate L, ϑ

A matrix eq.

$$\begin{pmatrix} f_A^x \\ f_A^y \\ f_B^x \\ f_B^y \end{pmatrix} = K \begin{pmatrix} A & -A \\ -A & A \end{pmatrix} \begin{pmatrix} \delta x_A \\ \delta y_A \\ \delta x_B \\ \delta y_B \end{pmatrix}$$

with $A = \begin{pmatrix} \cos^2 \vartheta & \cos \vartheta \sin \vartheta \\ \cos \vartheta \sin \vartheta & \sin^2 \vartheta \end{pmatrix}$

▷ construct triangle from 3 rods



→ displacement, force vector:

6 elements

→ stiffness matrix 6×6

→ compute stiffness matrix numerically:

.) create 6×6 matrix → all elements zeroed

.) subroutine to connect nodes i and j →

place submatrix A with proper sign into
matrix

.) call subroutine for $(i=1, j=2)$

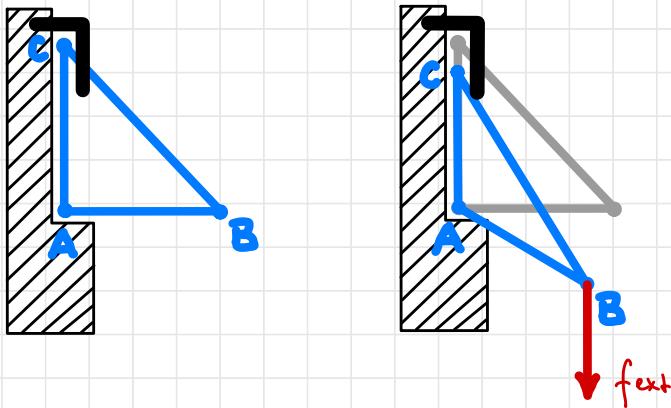
$(i=1, j=3)$

$(i=2, j=3)$

▷ all 3 rods involved

- ▷ triangle is: 1) small object
2) 2D triangular finite element

12. 3.2. Simple example



A: fixed

B: free

C: vertical motion possible

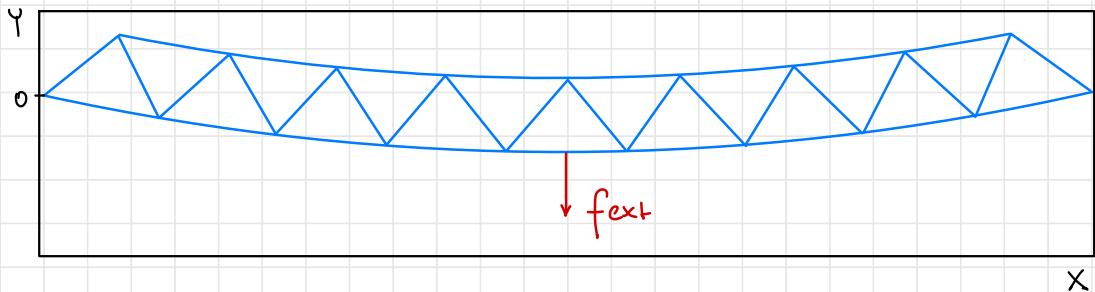
- ▷ stiffness eq. w. b.c. $\delta x_A = 0, \delta y_A = 0, \delta x_C = 0$

$$K_0 \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 + \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ 0 & 0 & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \end{pmatrix} \begin{pmatrix} \delta x_A \\ \delta y_A \\ \delta x_B \\ \delta y_B \\ \delta x_C \\ \delta y_C \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -f_{ext} \\ 0 \\ 0 \end{pmatrix}$$

▷ solve numerically
and hightly solvable

13.3.3. Bridge model

- ▷ use subroutine for connecting 2 nodes i and j
- ▷ bridge of N horizontal segments of length L and with support structure of height H $\rightarrow 2N+1$ nodes



$$x_i^{(0)} = \begin{cases} (i-1) \frac{L}{2} & \text{for } i=1, 3, 5, \dots, 2N+1 \\ (i - \frac{1}{2}) \frac{L}{2} & \text{for } i=2, 4, 6, \dots, 2N \end{cases}$$

$$y_i^{(0)} = \begin{cases} 0 & \text{for } i \text{ odd} \\ H & \text{for } i \text{ even} \end{cases}$$

start / end of rods

$$i_{\text{start}} = 1, 1, 2, 2, 3, 3 \dots$$

$$i_{\text{end}} = 2, 3, 3, 4, 4, 5, \dots$$

- ▷ force - f on all but 1st and last nodes
- ▷ b.c.: fix locations of 1st and last nodes
(→ set matrix elements accordingly)
- ▷ solve matrix eq numerically for displacements
@ given ext. force

13.4. Discontinuous Galerkin methods

- ▷ FEM approach to solve PDE systems in weak formulation → abandon continuity of solution across elements
- ▷ advantage: straight-forward extension of FVM to higher orders
more accurate & same computational cost

▷ Euler eqs. in 3D (system of hyperbolic PDEs)

$$\frac{\partial \vec{u}}{\partial t} + \sum_{\alpha=1}^3 \frac{\partial \vec{f}_\alpha}{\partial x_\alpha} = 0$$

\vec{u} : state vector

\vec{f}_α : flux vectors in
 x -direction

+ EoS

▷ Solution representation inside discrete cell K

with orthogonal and normalized basis fch. φ_i^K

$$\vec{u}^K(\vec{x}, t) = \sum_{i=1}^{N(K)} \vec{w}_i^K(t) \varphi_i^K(\vec{x}) \quad (*)$$

→ splits dependence on t and \vec{x} :

time-dependent "weights" $w_i^K(t)$ and
constant in time basis fch.

▷ solve (*) for weights

$$\vec{w}_j^K = \frac{1}{|V_K|} \int_K \vec{u}^K \varphi_j^K dV \quad j=1, \dots, N(K) \quad (**) \quad$$

- ▷ choose $\varphi_1 = 1 \wedge w_i^k \triangleq$ cell average of \bar{u}^k
- ▷ \bar{w}_j^k for $j \geq 2$: higher order moments of \bar{u}^k
- ▷ scaled variable: $\vec{\xi}$ (coords. in cell frame of reference)

$$\varphi_i(\vec{\xi}) : [-1, 1]^3 \rightarrow \mathbb{R} \quad (***)$$

- ▷ construction of possible 3D basis fcts. of maximum degree k from products of 3 1D Legendre polynomials $P_u(\xi_1) \cdot P_v(\xi_2) \cdot P_w(\xi_3)$
- ▷ u, v, w : orders of Legendre polynomials
- ▷ $u + v + w \leq k$
- ▷ # of basis fcts.

$$N(k) = \sum_{u=0}^k \sum_{v=0}^{k-u} \sum_{w=0}^{k-u-v} 1 = \frac{1}{6} (k+1)(k+2)(k+3)$$

→ spatial order of scheme: $p = k + 1$

▷ initial conditions:

$$\vec{u}^k(\vec{x}, t=0) = \sum_{i=1}^{N(k)} \vec{w}_i(0) \varphi_i^k(\vec{x})$$

→ exact if i.c. polynomials with degm $\leq k$
otherwise approximation

▷ evolution of weights

$$\int_K \left[\frac{\partial \vec{u}^k}{\partial t} + \sum_{\alpha=1}^3 \frac{\partial \vec{F}_\alpha}{\partial x_\alpha} \right] \varphi_j^k dV = 0$$

flux divergence term → integral by parts,

apply divergence theorem

$$R \frac{d}{dt} \int_K \vec{u}^k \varphi_j^k dV - \sum_{\alpha=1}^3 \int_K \vec{F}_\alpha \frac{\partial \varphi_j^k}{\partial x_\alpha}$$

(i)

(ii)

(4*)

$$+ \sum_{\alpha=1}^3 \int_{\partial K} \vec{F}_\alpha n_\alpha \varphi_j^k dS = 0$$

(iii)

normal vector

▷ 1st term

$$(i) \stackrel{(**)}{=} |V_K| \frac{d\vec{w}_j^K}{dt} \quad \text{time variation of weights}$$

- ▷ discretization of (ii) and (iii): transform to cell frame of reference (**), numerical quadrature of integrals
- ▷ flux in (iii) is over surface of cell K → cannot be evaluated analytically because solution \vec{u} is discontinuous @ cell interface
~ numerical flux function $\vec{F}_{num}(\vec{u}^K_-, \vec{u}^K_+, \hat{n})$
method → calculate with 1D Riemann

Solve

- ▷ discretization of each term in (4*)
→ spatial discretization of Euler eqs.

$$\frac{d\vec{w}_j^K}{dt} + \vec{R}_K = 0 \quad j=1, \dots, N(K)$$

→ set of coupled ODEs

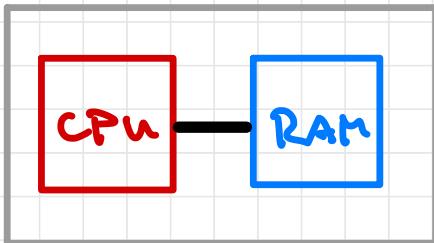
▷ time discretization w/ (higher-order) Runge-Kutta scheme

14. High-performance computing

- ▷ problem: speed of individual cores does not grow much
- ▷ use "Supernode" → increase # cores → parallelization of tasks
- ▷ problem: most algorithms and programming languages constructed for serial computers → structures executed sequentially
- ▷ serial computer? -

14.1 The von Neumann bottleneck

- ▷ serial computer → von Neumann architecture



- ▷ any data manipulation → pass through CPU

→ needs 1 clock cycle per data piece ↗

1 operation per clock cycle (at best)

- ▷ example: add 2 fp numbers:

$$0.92 \times 10^4 + 0.93 \times 10^3 \quad (\text{assume decimal representation})$$

- ▷ steps needed on von Neumann machine:

1) load data from memory to CPU

2) compare exponents: 4 and 3

3) line up exponents: $0.92 \times 10^4, 0.093 \times 10^4$

4) add mantissas : 1.013×10^4

5) shift back to standard representation :

$$0.1013 \times 10^5$$

6) write back to memory

→ 4 clock cycles needed for addition computation

▷ Speed of serial computer limited by its
clock speed →

"von Neumann bottleneck"

▷ HPC works around it :

paradigms of supercomputing :

.) vectorization

.) parallelization

.) hyperthreading

.) device computing

14.2. Vectorization

- ▷ idea: process large amount of data with same operations in pipeline
- ▷ vector instructions: single instruction applied to multiple data elements of a "vector" simultaneously
- ▷ example: add 2 fp vectors of length n :
for $i=1$ to n do
 $c[i] = a[i] + b[i]$
end for
- von Neumann: no difference $\rightarrow 4 \times n$ cycles needed
- vector processor: perform several manipulations in same clock cycle:
 -) compare exponents of $a[i+3]$ and $b[i+3]$
 -) line up $- \cdot -$ $a[i+2]$ and $b[i+2]$

-) add multiples of $a[i+1]$ and $b[i+1]$
-) reformulate $c[i]$

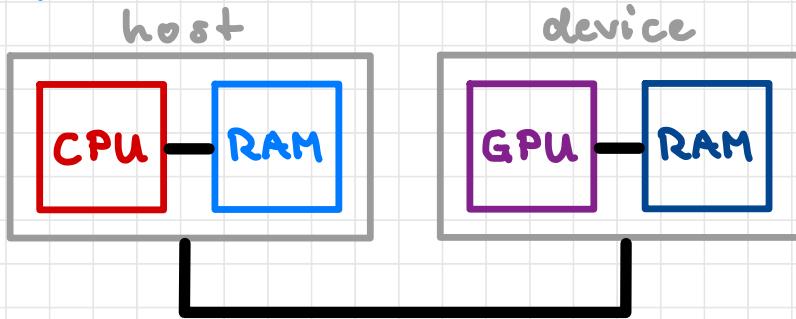
⇒ inside vector: 1 full addition per clock cycle → $4 \times$ faster

- ▷ efficient for long vectors
- ▷ done automatically by modern compilers, implemented to some level on hardware (SSE / AVX) instructions on Intel / AMD
- ▷ enable uninterrupted pipelines → avoid stalls
- ▷ drawback: Specialized, expensive hardware needed if aiming to extreme performance

14.3. Device computing

- ▷ augment compute nodes with specialized accelerator cards tuned for fp operation performance

- ▷ simpler, less flexible, but more efficient for special purpose, e.g. GPUs
- ▷ hybrid mode



→ can be much faster than conventional modes, harder to program

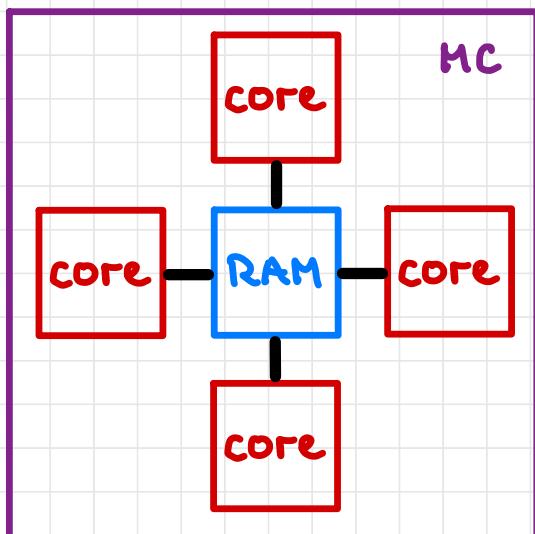
14.4 Hyperthreading

- ▷ problem of modern hardware: memory access much slower than fp operations → cores spend much time waiting for data
- ▷ idea: use idle time for other operations
→ "hyperthreading"

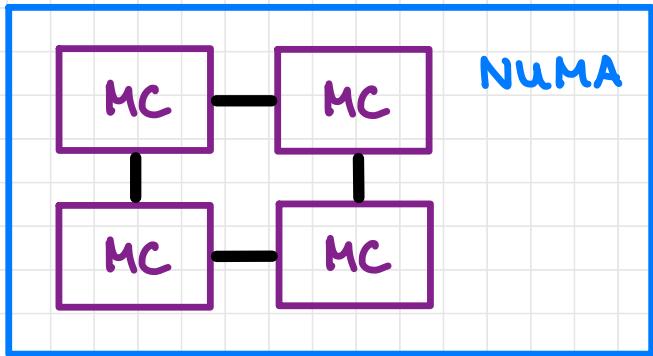
- ▷ overlock core ↳ several execution streams
 - hardware directly toggles between them
 - can be efficient despite of overlock
- ▷ looks like multiple "virtual" cores
- ▷ example: IBM Blugem / Q:
 - Cpus ↳ 16 physical core
 - 4-fold hyperthreading → can start 64 threads
 - not 4x faster, but perhaps $\sim 2\times$

14.5. Parallelization

- ▷ multi-core nodes (MC)



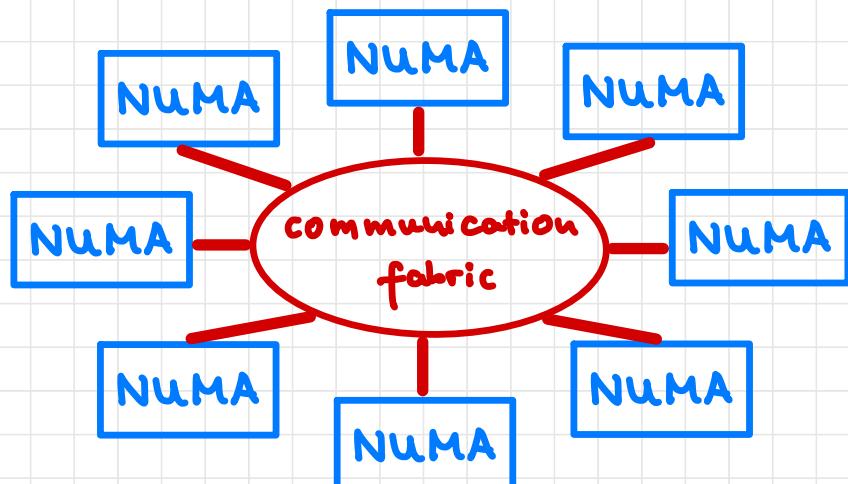
▷ Multi-socket nodes → "non-uniform memory access nodes" (NUMA)



→ full memory accessible to all cores but
not at same speed

→ most efficient when data stored @ "right"
memory bank → usually taken care of by OS

▷ compute clusters



- fast communication required
- currently some $10^5 - 10^6$ cores

14.5.1 Amdahl's Law

- define speedup for P parallel cores :

$$s(P) = \frac{t(1)}{t(P)}$$

— execution time on single core
— α — on P cores

- define intrinsically sequential part of program
 f_{seq} and parallel part $f_{par} = 1 - f_{seq}$

$$\triangleright s(P) = \frac{t(1)}{t(P)} = \frac{f_{seq} + f_{par}}{f_{seq} + \frac{1}{P} f_{par}} = \frac{1}{f_{seq} + \frac{1-f_{seq}}{P}}$$

$$\lim_{P \rightarrow \infty} s(P) = \frac{1}{f_{seq}}$$

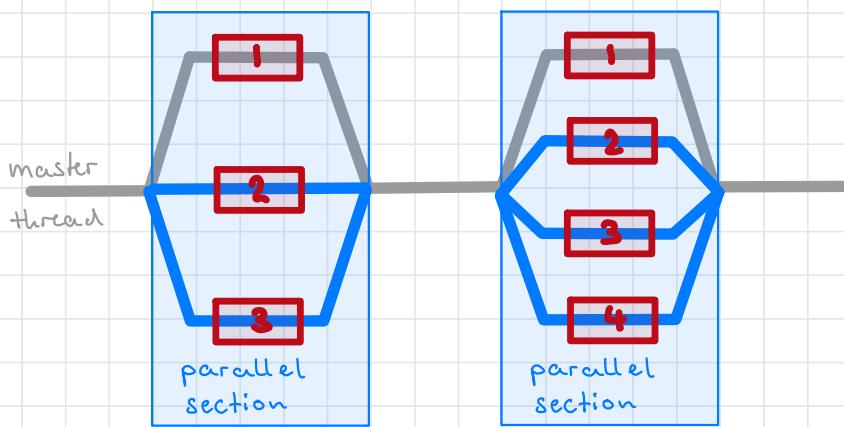
~ high parallel speedup requires small f_{seq}
 → hard!

14.5.2. Shared memory parallelization with OpenMP

- ▷ distribute computational workload to multiple cores w/ access to same memory
- ▷ OpenMP standard → language/compiler extension for C/C++, Fortran
 - .) in program code: add hints to compiler where to execute in parallel (||)
 - .) advantage: minimal code modifications

required

- ▷ OpenMP's "fork-join model"



- programmer identifies || section in code
- program execution is split up into number of threads @ beginning of || section
- threads join into single thread & end

- ▷ example

```
# pragma omp parallel for
```

```
for (i=0; i < 200; i++)
```

```
    some_expensive_computation(i);
```

loop-level parallelism w/ OpenMP

OpenMP compiler will automatically:

-) wake up all available threads @ beginning of loop
-) distribute loop iterations evenly onto them
-) execute them concurrently
-) put threads back to sleep when done

→ master thread continues sequential work

→ works only if no dependencies between loop iterations

→ needed in addition:

-) use OpenMP - capable compiler, e.g.
gcc - fopenmp
-) some more advanced OpenMP features
accessible via library
`#include <omp.h>`
-) define number of threads with

environment variable before starting

program:

```
export OMP_NUM_THREADS = 8
```

▷ "race condition"

threads want to modify same variable @

same time \rightsquigarrow "stochastic linear" due to

"timing noise" \rightarrow subsequent runs of same
program produce non-deterministic results

example: double loop:

```
# pragma omp parallel for private(j)
```

```
for(i=0; i<N; i++)
```

```
{
```

```
    for(j=0; j<N; j++)
```

```
{
```

```
        do_something();
```

```
}
```

```
{
```

- outer loop i will be split up, but all threads share same variable j & cannot carry out inner loop j independently
- cur: each thread has own copy of j

14.5.3. Distributed memory parallelization with MPI

- ▷ "Message Passing Interface" (MPI) → de facto standard → library of functions for exchanging messages between processes
- ▷ mostly same program executed on all cores
 - act on different local data
(SPMD "single program multiple data" paradigm)
- ▷ all data exchanges need to be programmed explicitly → substantial program modification, algorithmic changes necessary
- ▷ can be combined w/ OpenMP → "hybrid parallelization"

▷ general structure of MPI program:

```
#include <mpi.h>
```

```
int main [ int argc, char **argv ]
```

```
{
```

```
    MPI_Init (&argc, &argv);
```

```
:
```

```
    MPI_Finalize ();
```

```
}
```

▷ compiled w/ compiler wrapper; e.g. mpicc

▷ execute w/ mpirun or mpiexec, e.g.

mpirun -np 8 ./myprogram

▷ somewhat different execution path in each

worker → dependent on "rank" of MPI task

0, 1, ..., N-1 within "communicator"

→ send / receive messages depending on rank

- ▷ MPI standard → pre-defined functions for often-used communication patterns
- ▷ MPI initialization
- ▷ simple point-to-point communication
 - example with MPI_Send, MPI_Recv
- ▷ collective communication:
 - broadcasts
 - gather
 - reduction