Other topics

Numerical Flow Simulation

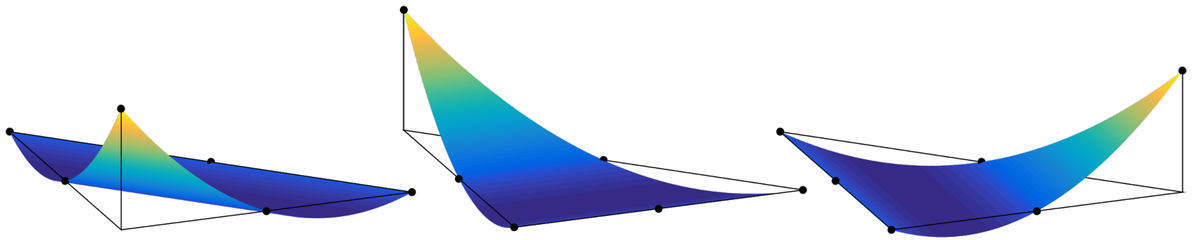
Edouard Boujo

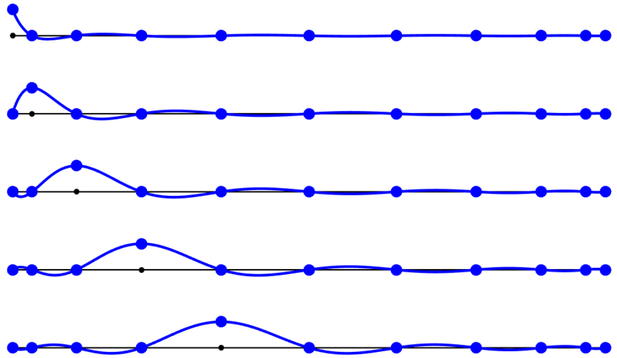
Fall 2024

École
polytechnique
fédérale
de Lausanne

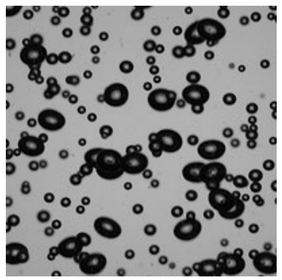
This week's lecture

- Brief overview of 2 topics:
  1. Discretization/computation methods (other than FVM)





  2. Techniques for handling fluid interfaces



Discretization/computation methods (reminder)

	Some software (Commercial – Free/open)
▪ Finite difference method (FDM)	(Mostly academic)
▪ **Finite volume method (FVM)**	Fluent, Star-CCM+, OpenFOAM…
▪ Finite element method (FEM)	Comsol, FreeFEM++, AVBP…
▪ Spectral element method (SEM)	Nek5000…
▪ Spectral methods	(Mostly academic)
▪ Particle-based methods	SPH-flow, Fluidix…
▪ Lattice Boltzmann methods (LBM)	PowerFLOW…

Discretization/computation methods (reminder)

- Finite difference method (FDM)
- **Finite volume method (FVM)**
- Finite element method (FEM)
- Spectral element method (SEM)

Applicable to **any PDE** (electrodynamics, thermodynamics, pattern formation, quantum mechanics, statistical mechanics, economics…)

- Spectral methods

- Particle-based methods

Specific to continuum media (**solid / fluid**)

- Lattice Boltzmann methods (LBM)

Specific to **fluid**

Discretization/computation methods

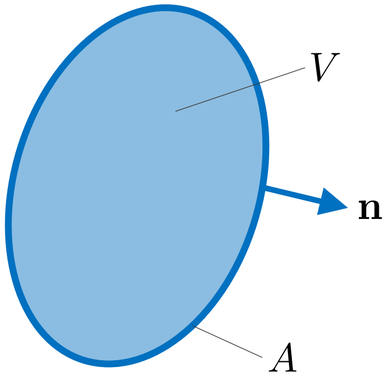
- Finite difference method (FDM)
- **Finite volume method (FVM)**
- Finite element method (FEM)
- Spectral element method (SEM)

- Spectral methods

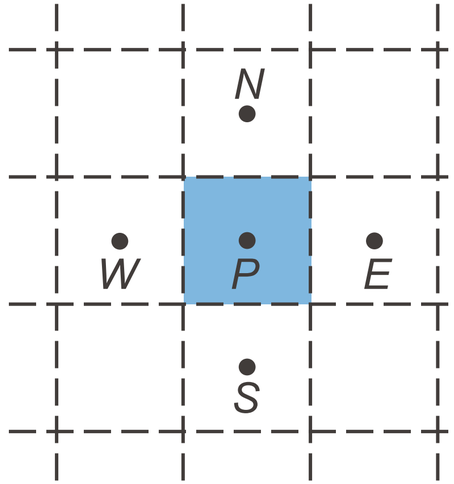
- Particle-based methods

- Lattice Boltzmann methods (LBM)

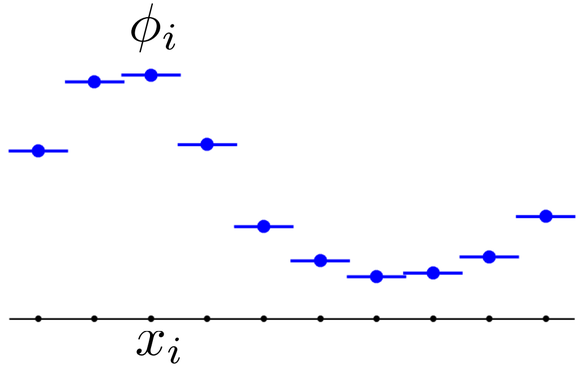
Finite volume method (reminder)



- Conservation equation $\dfrac{\partial(\rho\phi)}{\partial t} + div(\rho\phi\mathbf{u}) = div(\Gamma\, grad(\phi)) + S$
- Integral form: $\dfrac{\partial}{\partial t}\int_V \rho\phi\, dV + \oint_A \rho\phi\mathbf{u}\cdot\mathbf{n}\, dA = \oint_A \Gamma\, grad(\phi)\cdot\mathbf{n}\, dA + \int_V S\, dV$



- Divide the domain into a partition of control volumes
- Discretize the solution as nodal values: $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_i, \ldots \phi_n)$
- Approximate volume integrals as functions of nodal values, and surface integrals as functions of face values
- Algebraic system of equations $\mathbf{A}(\boldsymbol{\phi}) = \mathbf{0}$

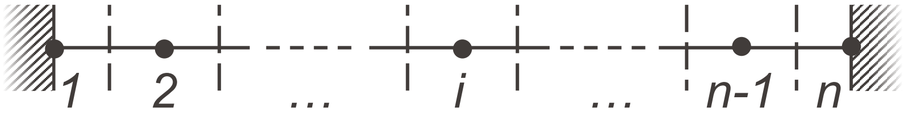


- Note: solution defined at the nodes. No information between nodes, only information about averaged value in each CV (integral formulation).

Finite volume method (reminder)

- Example: 1D steady diffusion (e.g. heat conduction)

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+S=0$$

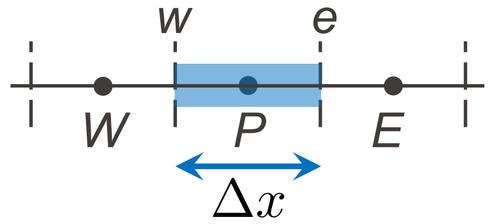


- Discretize into *n* CVs, *n* nodes: $T(\mathbf{x}) \rightarrow \boldsymbol{T}=(T_1, T_2 \ldots, T_i \ldots, T_n)$
- Integration over CV + divergence theorem:

$$0=\left(k\frac{\partial T}{\partial x}\right)_e-\left(k\frac{\partial T}{\partial x}\right)_w+\int_{x_w}^{x_e} S\,dx$$

- Approximation as function of nodal values (CD):

$$0 \approx k_e\frac{T_E-T_P}{\delta x_{PE}}-k_w\frac{T_P-T_W}{\delta x_{WP}}+S_P\Delta x$$

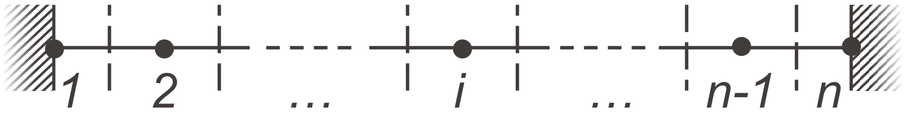


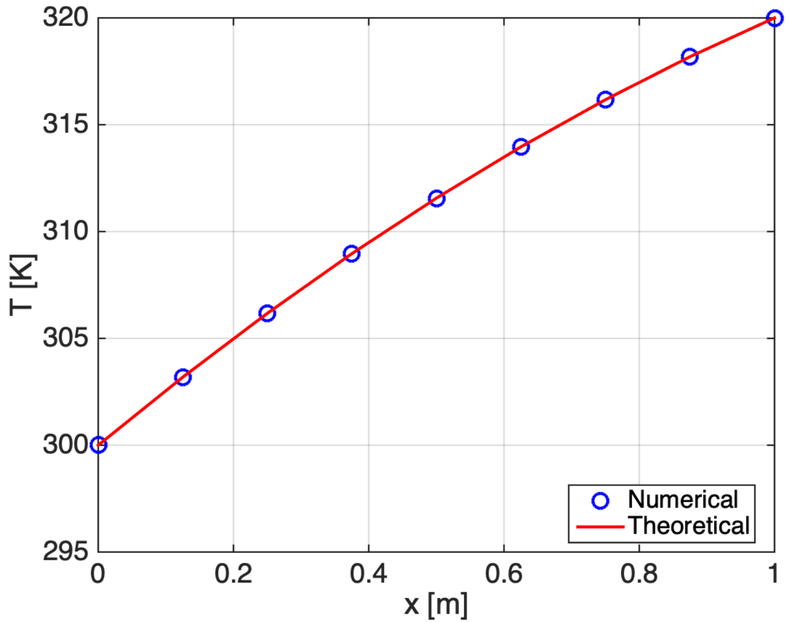
- Linear algebraic equation: $a_P T_P = a_W T_W + a_E T_E + b = \sum_{nb} a_{nb} T_{nb} + b$
- Linear system (sparse matrix): $\mathbf{AT}=\mathbf{b}$

Finite volume method (reminder)

- Example: 1D steady diffusion (e.g. heat conduction)

```
L  = 1;
Ta = 300;
Tb = 320;
k  = 400;
Sc = 5000;
n  = 9;   x = linspace(0,L,n);   dx = L/(n-1);

%--- Matrix and boundary conditions
A = zeros(n,n);   b = zeros(n,1);
for i=2:n-1
    A(i,i-1:i+1) = k*[-1 2 -1]/dx;
    b(i)         = Sc*dx;
end
A(1,1) = 1;   b(1) = Ta;
A(n,n) = 1;   b(n) = Tb;

%--- Numerical solution
T = A\b;

%--- Plot
figure('color','w'), hold on, grid on, box on,
plot(x,T,'bo')

%--- Theoretical solution
T_theo = -Sc/(2*k)*x.^2 + ((Tb-Ta)/L+Sc*L/(2*k))*x + Ta;
plot(x,T_theo,'r-')
```





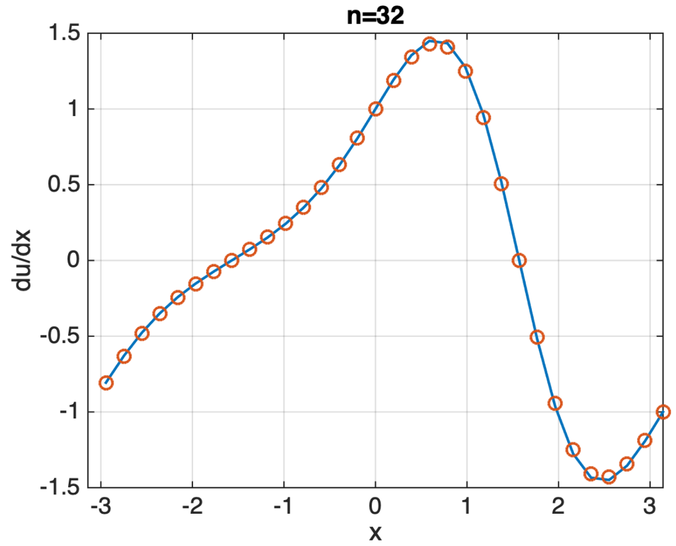
Finite volume method

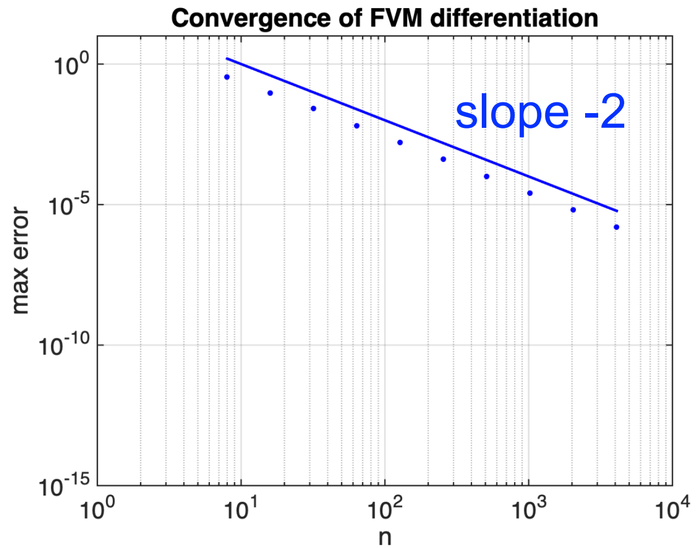
- Focus on differentiation: 1st derivative

```
nvec = 2.^(3:12);    % Number of grid points
figure, hold on, box on, grid on
for n = nvec
    dx = 2*pi/n;    x = -pi + dx*(1:n)';    % Grid generation

    %--- Differentiation matrix
    D = zeros(n,n);
    for i=2:n-1
        D(i, i-1:i+1) = [-1/2 0 1/2]/dx;
    end
    D(1,1:2)          =  [0 1/2]/dx;    D(1,end) = -1/2/dx;
    D(end,end-1:end) = [-1/2 0]/dx;    D(end,1) =  1/2/dx;
    D = sparse(D);

    u = exp(sin(x));      % Function
    uprime = cos(x).*u; % Analytical derivative
    uprime_num = D*u;    % Numerical derivative

    %--- Error
    err = norm(uprime-uprime_num, inf);
    plot(n,err,'b.','markersize',15),
end
set(gca,'xscale','log'), set(gca,'yscale','log')
xlabel n, ylabel('max error'),
plot(nvec, 100./nvec.^2, 'b')
```

Example on a periodic domain:

$$u(x) = e^{\sin(x)} \quad \longrightarrow \quad u'(x) = \cos(x)u(x)$$





Discretization/computation methods

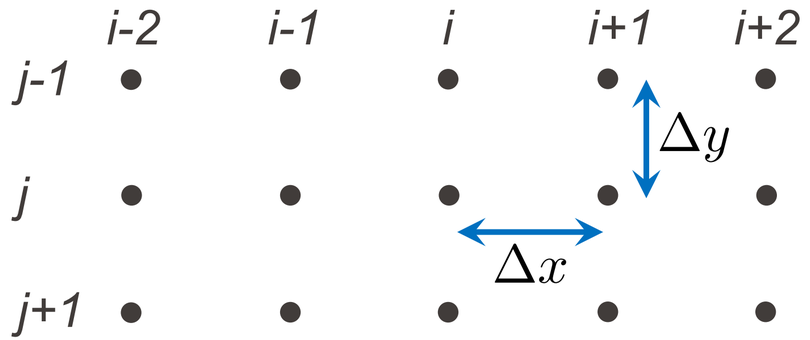
- **Finite difference method (FDM)**
- Finite volume method (FVM)
- Finite element method (FEM)
- Spectral element method (SEM)

- Spectral methods

- Particle-based methods

- Lattice Boltzmann methods (LBM)

Finite difference method



- Solution discretized at grid points
- Derivatives expressed by combining Taylor expansions:

$$f_{i\pm1} = f_i \pm \Delta x \left.\frac{\partial f}{\partial x}\right|_i + \frac{\Delta x^2}{2!}\left.\frac{\partial^2 f}{\partial x^2}\right|_i \pm \frac{\Delta x^3}{3!}\left.\frac{\partial^3 f}{\partial x^3}\right|_i + O(\Delta x^4)$$

$$f_{i\pm2} = f_i \pm \ldots$$

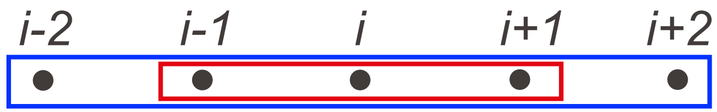
$$f_{i\pm3} = f_i \pm \ldots$$

- For example, 2nd-order approximations of 1st and 2nd derivatives:

$$f_{i+1} - f_{i-1} = 2\Delta x \left.\frac{\partial f}{\partial x}\right|_i + O(\Delta x^3) \quad \Rightarrow \quad \left.\frac{\partial f}{\partial x}\right|_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x} + O(\Delta x^2)$$

$$f_{i+1} + f_{i-1} = 2f_i + \Delta x^2 \left.\frac{\partial^2 f}{\partial x^2}\right|_i + O(\Delta x^4) \quad \Rightarrow \quad \left.\frac{\partial^2 f}{\partial x^2}\right|_i = \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + O(\Delta x^2)$$

Finite difference method



- Can increase the order of accuracy by widening the stencil (i.e. using more neighbors). For example (on a uniform grid):

Derivative	Accuracy	−5	−4	−3	−2	−1	0	1	2	3	4	5
1	2					−1/2	0	1/2				
	4				1/12	−2/3	0	2/3	−1/12			
	6			−1/60	3/20	−3/4	0	3/4	−3/20	1/60		
	8		1/280	−4/105	1/5	−4/5	0	4/5	−1/5	4/105	−1/280	
2	2					1	−2	1				
	4				−1/12	4/3	−5/2	4/3	−1/12			
	6			1/90	−3/20	3/2	−49/18	3/2	−3/20	1/90		
	8		−1/560	8/315	−1/5	8/5	−205/72	8/5	−1/5	8/315	−1/560	
3	2				−1/2	1	0	−1	1/2			
	4			1/8	−1	13/8	0	−13/8	1	−1/8		
	6		−7/240	3/10	−169/120	61/30	0	−61/30	169/120	−3/10	7/240	
4	2				1	−4	6	−4	1			
	4			−1/6	2	−13/2	28/3	−13/2	2	−1/6		
	6		7/240	−2/5	169/60	−122/15	91/8	−122/15	169/60	−2/5	7/240	

Wikipedia

Finite difference method

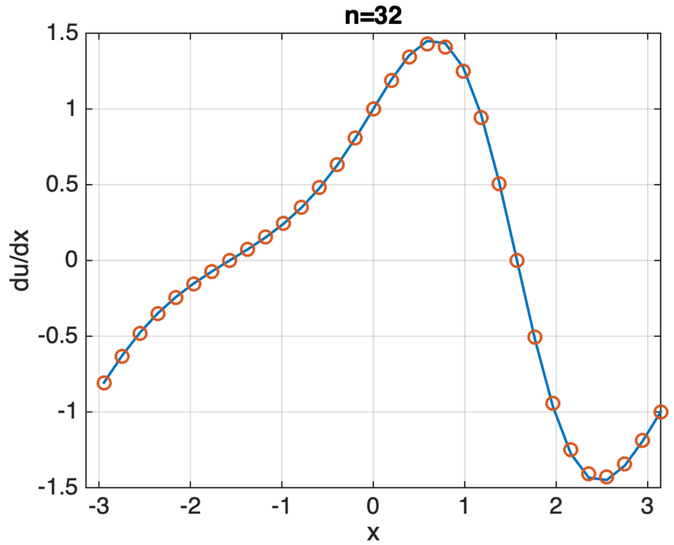
- Focus on differentiation: 1st derivative

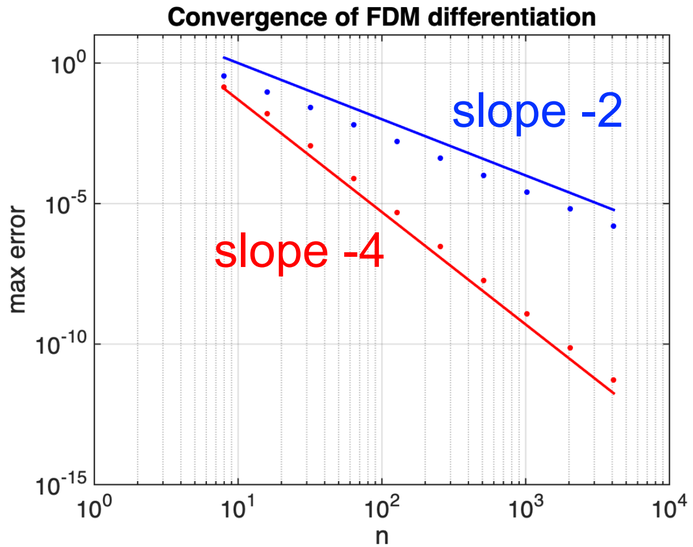
```
nvec = 2.^(3:12);    % Number of grid points
figure, hold on, box on, grid on
for n = nvec
    dx = 2*pi/n;    x = -pi + dx*(1:n)';    % Grid generation

    %--- Differentiation matrix
    e = ones(n,1);
    D = sparse(1:n,[2:n 1],2*e/3,n,n)...
       -sparse(1:n,[3:n 1 2],e/12,n,n);
    D = (D-D')/dx;

    u = exp(sin(x));     % Function
    uprime = cos(x).*u; % Analytical derivative
    uprime_num = D*u;    % Numerical derivative

    %--- Error
    err = norm(uprime-uprime_num, inf);
    plot(n,err,'r.','markersize',15),
end
set(gca,'xscale','log'), set(gca,'yscale','log')
xlabel n, ylabel('max error'),
plot(nvec, 500./nvec.^4, 'r')
```

↑ Example code: 4th-order FD.
(In 1D, 2nd-order FD similar to 2nd-order FV.)

Example on a periodic domain:

$$u(x) = e^{\sin(x)} \quad \longrightarrow \quad u'(x) = \cos(x)u(x)$$





Finite difference method

- Wide stencils are not ideal. In practice, accuracy not better if solution not smooth enough at the scale of the grid.
- Alternatively, can increase the order of accuracy with "**compact FD**": introduce additional degrees of freedom by combining Taylor expansions that involve the unknown derivative at other grid points.
- For example, for a 4th-order approximation of the 1st derivative, instead of looking for the standard expression

$$f'_i = \frac{af_{i-2} + bf_{i-1} + cf_i + df_{i+1} + ef_{i+2}}{\Delta x}$$

look for a compact expression of the form

$$\alpha f'_{i-1} + \beta f'_i + \gamma f'_{i+1} = \frac{af_{i-1} + bf_i + cf_{i+1}}{\Delta x}$$

and find in this case:

$$f'_{i-1} + 4f'_i + f'_{i+1} = \frac{3f_{i-1} - 3f_{i+1}}{\Delta x}$$

Finite difference method

- Example: $$f'_{i-1} + 4f'_i + f'_{i+1} = \frac{3f_{i-1} - 3f_{i+1}}{\Delta x}$$

- Denoting the discretized solution $\mathbf{f} = (f_1, f_2, \ldots, f_i, \ldots f_n)$, this can be written as a linear system, to be solved for the derivative:

$$\mathbf{A}\mathbf{f}' = \mathbf{B}\mathbf{f}$$

- When solving a PDE, e.g. $\frac{\partial f}{\partial t} + c\frac{\partial f}{\partial x} = 0$,

  standard FD yields $\frac{\partial \mathbf{f}}{\partial t} + c\mathbf{D}\mathbf{f} = 0$

  while compact FD yields $\frac{\partial \mathbf{f}}{\partial t} + c\mathbf{f}' = 0 \Rightarrow \mathbf{A}\frac{\partial \mathbf{f}}{\partial t} + c\mathbf{B}\mathbf{f} = 0$

  to be solved for **f**, and where **A**, **B** have a narrower stencil than **D**.

Discretization/computation methods

- Finite difference method (FDM)
- Finite volume method (FVM)
- **Finite element method (FEM)**
- Spectral element method (SEM)

- Spectral methods

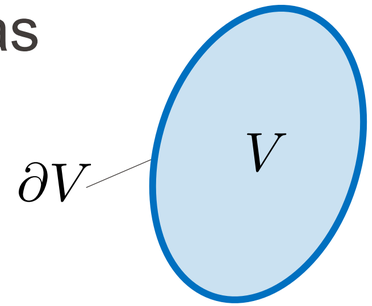
- Particle-based methods

- Lattice Boltzmann methods (LBM)

Finite element method

- Transform the problem into its variational formulation (“weak form”).
  For example, to the steady diffusion problem written in ”strong form” as

Find *u* such that $\nabla^2 u + s = 0 \quad \text{in } V \quad \text{and} \quad u = 0 \quad \text{on } \partial V$



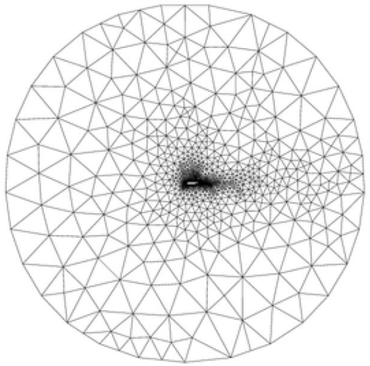
corresponds the “weak form”

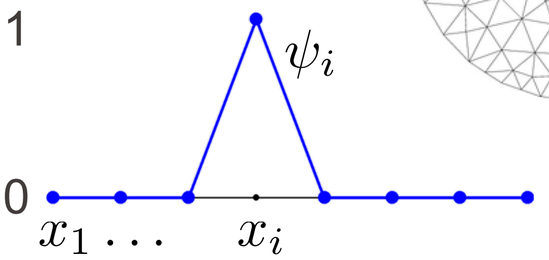
Find *u* such that $\int_\Omega \nabla^2 u \, w \, dV + \int_\Omega s \, w \, dV = 0 \qquad \forall w$

or $-\int_\Omega \nabla u \cdot \nabla w \, dV + \int_\Omega s \, w \, dV = 0 \qquad \forall w$

Idea: instead of solving the PDE directly, ask for its projection on any function to be satisfied.

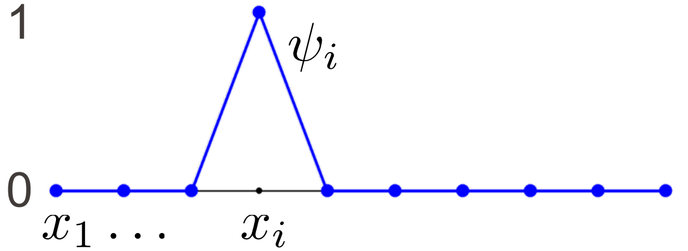
- Divide the domain into a partition of elements and nodes.
- Introduce basis functions (usually low-order polynomials) such that
  $\psi_i(x_i) = 1$ at node *i*,
  $\psi_i(x_j) = 0$ at other nodes *j* of the element.





(Piecewise def.: $\psi_i = 0$ on elements not containing node *i*)

Finite element method



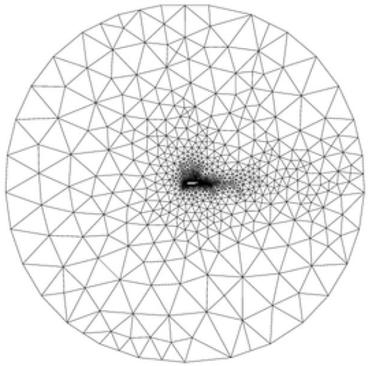
- Discretize the solution *u* and test function *w* as combinations of basis functions:

$$u(x) = \sum_{i=1}^{n} u_i \psi_i(x) \qquad w(x) = \sum_{i=1}^{n} w_i \psi_i(x)$$

- Note: because of the properties of the ψ_i, the unknowns $\mathbf{u} = (u_1, u_2, \ldots, u_i, \ldots u_n)$ are nodal values: $u_i = u(x_i)$
- Insert into the weak form to obtain the discretized equation:

$$-\int_\Omega \nabla u \cdot \nabla w \, dV + \int_\Omega s \, w \, dV = 0 \qquad \forall w$$

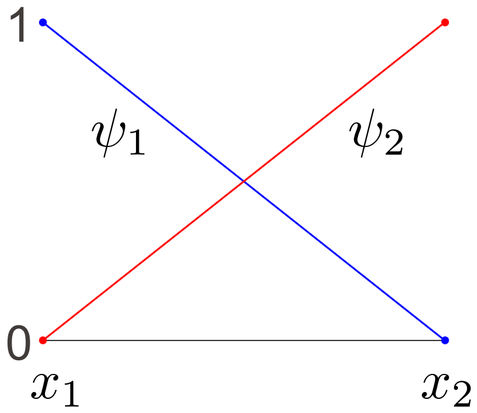
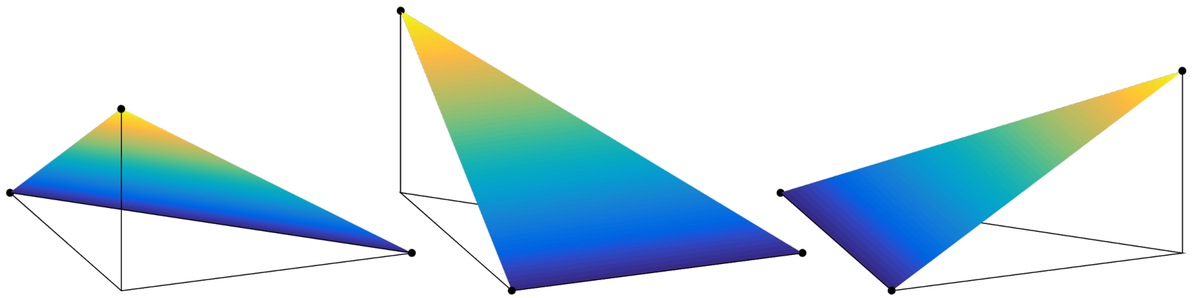
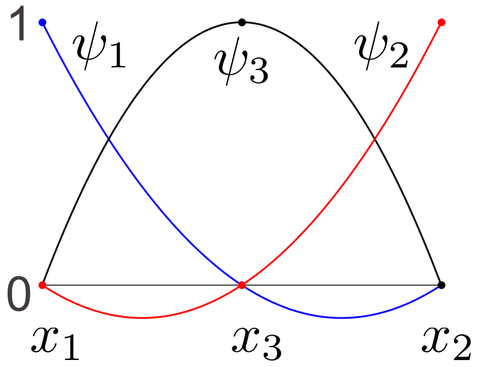
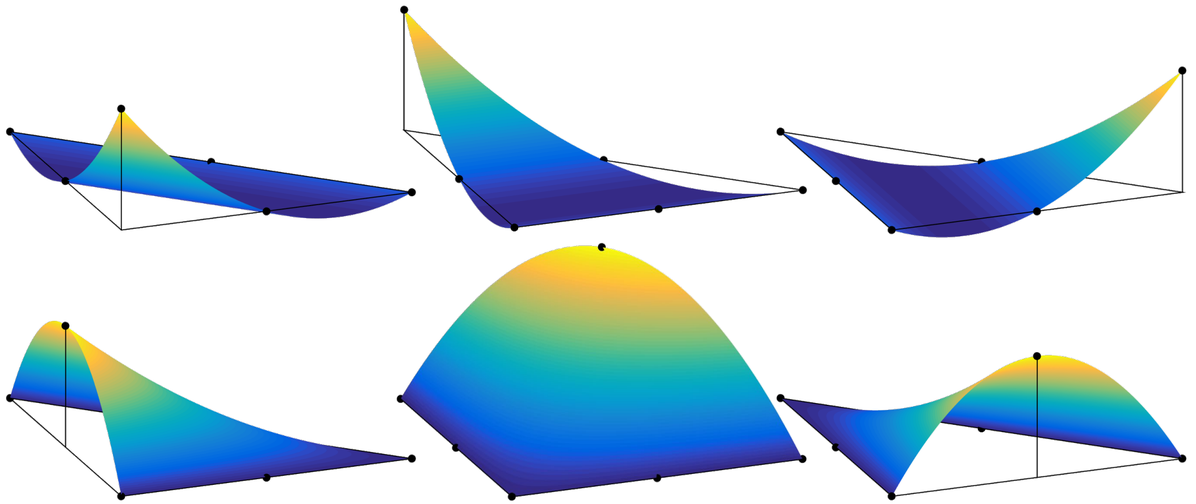
$$\sum_i u_i \int_\Omega \nabla \psi_i \cdot \nabla \psi_j \, dV = \sum_i s_i \int_\Omega \psi_i \psi_j \, dV \quad \forall j$$



- Algebraic system, of matrix form: $\mathbf{Au} = \mathbf{Mb}$

  with (sparse) matrices: $\mathbf{A}_{i,j} = \int_\Omega \nabla \psi_i \cdot \nabla \psi_j \, dV \qquad \mathbf{M}_{i,j} = \int_\Omega \psi_i \psi_j \, dV$

Finite element method

Common basis functions	1D	2D
Linear P1 elements (1 node at each vertex)		
Quadratic P2 elements (1 node at each vertex and each edge midpoint)		

Discretization/computation methods

- Finite difference method (FDM)
- Finite volume method (FVM)
- Finite element method (FEM)
- Spectral element method (SEM)

- **Spectral methods**

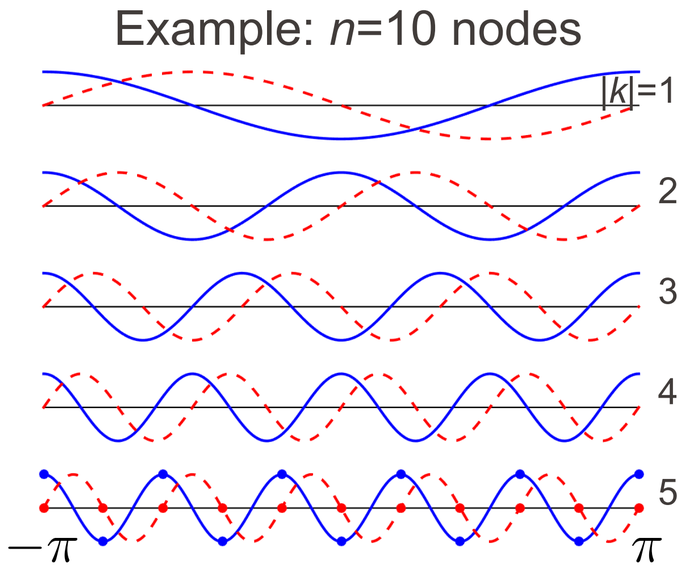
- Particle-based methods

- Lattice Boltzmann methods (LBM)

Spectral methods

- Expansion on basis functions that are defined globally (over the whole domain)
- On **periodic** domains:
  - Basis functions: $\psi_k(x) = e^{ikx}$ (trigonometric polynomials)
    Expansion: Fourier series

$$u(x) = \sum_{k=-n/2}^{n/2-1} \hat{u}_k \psi_k(x) = \sum_{k=-n/2}^{n/2-1} \hat{u}_k e^{ikx}$$



  - The unknowns $\hat{\mathbf{u}} = (\hat{u}_{-n/2}, \ldots, \hat{u}_k, \ldots)$ are the expansion coefficients (and are **not** nodal values).
  - Differentiation becomes very easy: $\psi' = ik\psi$, and thus

$$u'(x) = \sum ik\hat{u}_k\psi_k(x)$$

  - Can solve any PDE in Fourier space, and then come back to physical space.
  - Can also derive (dense) differentiation matrices (e.g. $\mathbf{u}' = \mathbf{D}\mathbf{u}$) and solve directly in physical space.

Spectral methods

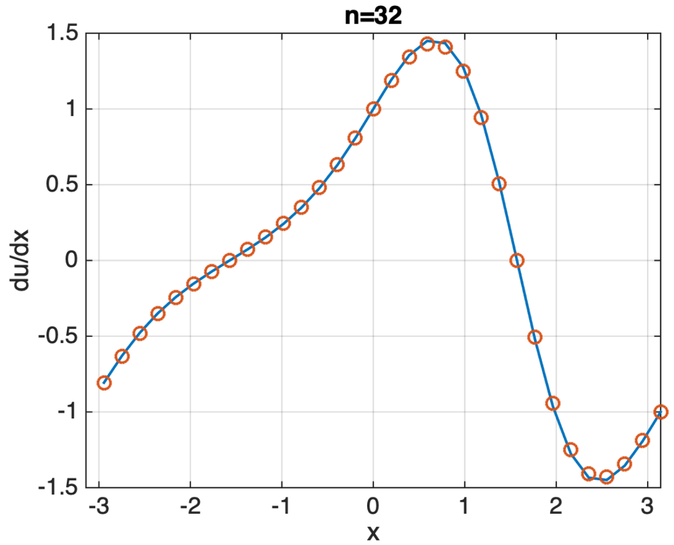
Example on a periodic domain:

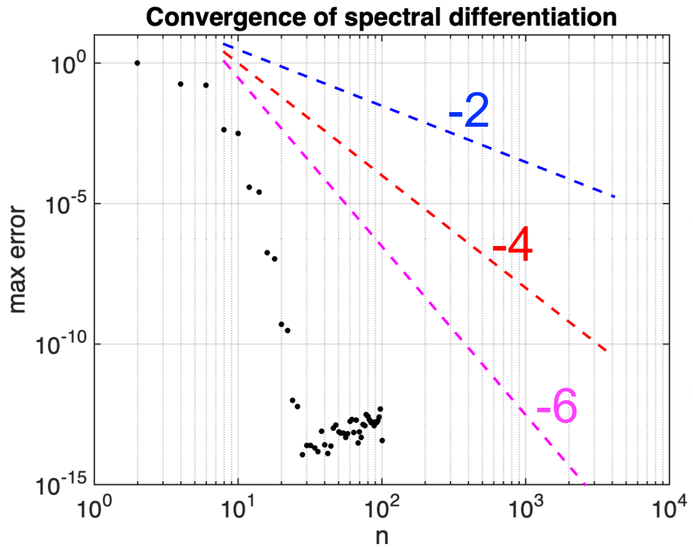
$$u(x) = e^{\sin(x)} \quad \longrightarrow \quad u'(x) = \cos(x)u(x)$$

- Focus on differentiation: 1st derivative

```
nvec = 2:2:100;    % Number of grid points
figure, hold on, box on, grid on
for n = nvec;
    h = 2*pi/n;    x = -pi + h*(1:n)';    % Grid generation

    %--- Differentiation matrix
    column = [0 .5*(-1).^(1:n-1).*cot((1:n-1)*h/2)];
    D = toeplitz(column,column([1 n:-1:2]));

    u = exp(sin(x));      % Function
    uprime = cos(x).*u; % Analytical derivative
    uprime_num = D*u;    % Numerical derivative

    %--- Error
    err = norm(uprime_num-uprime, inf);
    plot(n,err,'k.','markersize',15),
end
set(gca,'xscale','log'),
set(gca,'yscale','log')
xlabel n, ylabel('max error'),
```

$$\mathbf{D} = \begin{pmatrix} 0 & & & -\frac{1}{2}\cot\frac{1h}{2} \\ -\frac{1}{2}\cot\frac{1h}{2} & \ddots & \ddots & \frac{1}{2}\cot\frac{2h}{2} \\ \frac{1}{2}\cot\frac{2h}{2} & \ddots & & -\frac{1}{2}\cot\frac{3h}{2} \\ -\frac{1}{2}\cot\frac{3h}{2} & & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{1}{2}\cot\frac{1h}{2} \\ \frac{1}{2}\cot\frac{1h}{2} & & & 0 \end{pmatrix}$$





For smooth functions: reach **spectral accuracy** (more accurate than any polynomial-order method)

Numerical Flow Simulation

From *"Spectral Methods in Matlab"*, L. Trefethen

Spectral methods

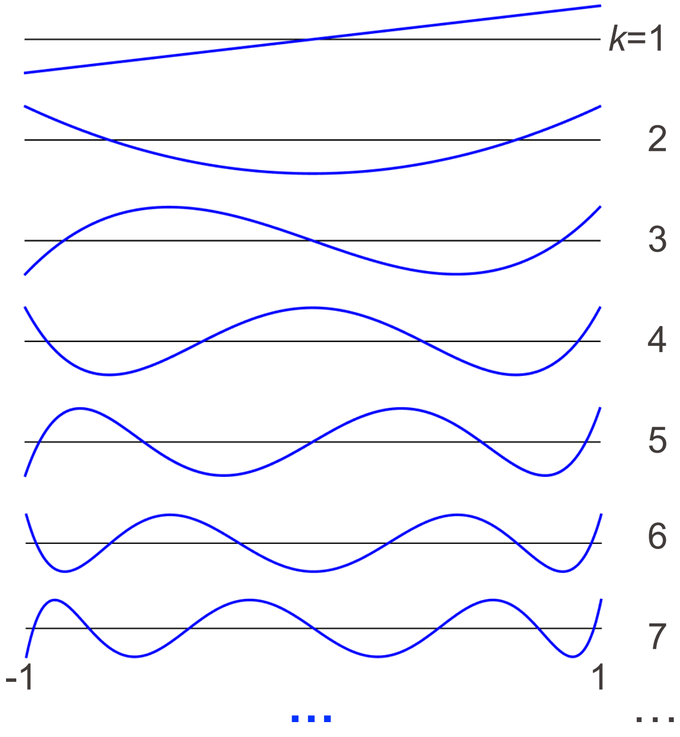
- Expansion on basis functions that are defined globally (over the whole domain)
- On **non-periodic** domains:
  - Basis functions: for ex. $T_k(x) = \cos\left(k \cos^{-1}(x)\right)$ (Chebyshev polynomials). Expansion:

$$u(x) = \sum_{k=0}^{n} a_k T_k(x)$$

  - The unknowns $\mathbf{a} = (a_0, a_1 \ldots, a_k, \ldots, a_n)$ are the expansion coefficients (and are **not** nodal values).
  - Collocation method: enforce the governing equations on *n*+1 points. For ex.: Gauss-Chebyshev-Lobatto points:

$$x_j = \cos\left(\frac{j\pi}{n}\right)$$

→ obtain an algebraic system for the a_k (dense matrix).



Discretization/computation methods

- Finite difference method (FDM)
- Finite volume method (FVM)
- Finite element method (FEM)
- **Spectral element method (SEM)**

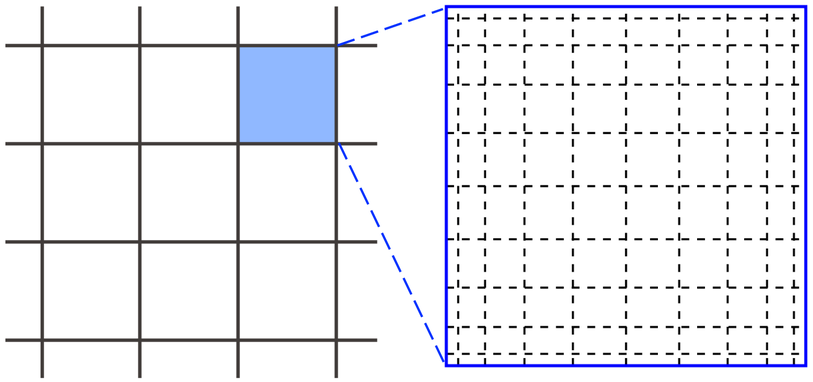
- Spectral methods

- Particle-based methods

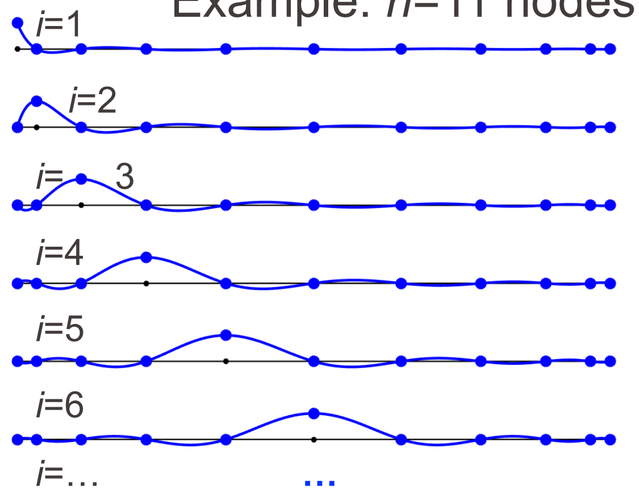
- Lattice Boltzmann methods (LBM)

Spectral element methods

- One specific type of FEM, with high-order polynomials on each element:
  - Domain divided into a partition of elements and nodes.
  - Within each element, nodes are clustered near the boundaries (e.g. Gauss-Lobatto-Legendre points).
  - Piecewise basis functions: Lagrange polynomials such that $\psi_i(x_i) = 1$ at node *i*,
    $\psi_i(x_j) = 0$ at other nodes *j* of the element.



Example: *n*=11 nodes



  - Discretize the solution *u* as combination of basis functions:

$$u(x) = \sum_{i=1}^{n} u_i \psi_i(x)$$

  - In 2D/3D, basis functions are often a tensor product (separation of variables, unlike FEM):

$$u(x,y) = \sum_{i,j} u_{ij} \psi_i(x) \psi_j(y)$$

  - Combines flexibility of FEM (complex geometries) and spectral accuracy of SM.

Discretization/computation methods

- Finite difference method (FDM)
- Finite volume method (FVM)
- Finite element method (FEM)
- Spectral element method (SEM)

- Spectral methods

- **Particle-based methods**

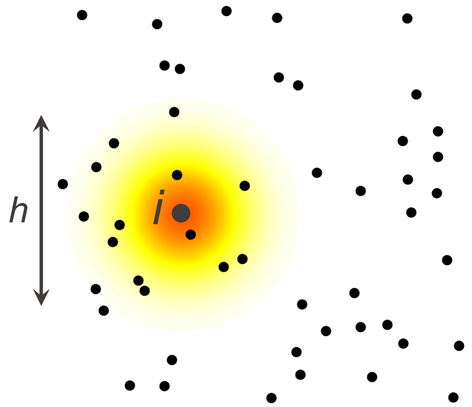
- Lattice Boltzmann methods (LBM)

Particle methods (here: smoothed-particle hydrodynamics, SPH)

- Mesh-free methods
- Lagrangian description: track a discrete set of fluid “particles”
- Mass conservation is automatically satisfied
- Any field *f(x)* is represented as a spatial average:

$$f(\mathbf{x}) = \int_\Omega f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}' \approx \sum_j f(\mathbf{x}_j)W(\mathbf{x}-\mathbf{x}_j,h)V_j$$

V_j: volume of particle j



W: weighting function (“kernel”) of characteristic size *h* (“smoothing length”)

- Solve NS eqs (in Lagrangian form) and track particles by solving

$$\frac{\mathrm{d}\mathbf{r}_i}{\mathrm{d}t} = \mathbf{v}_i$$

- No special treatment required for interfaces (free-surface flow, multiphase flows)
- High computational cost, but well suited to parallel computing
- Some boundary conditions difficult to define

Discretization/computation methods

- Finite difference method (FDM)
- Finite volume method (FVM)
- Finite element method (FEM)
- Spectral element method (SEM)

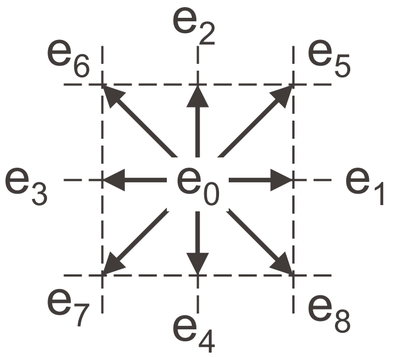
- Spectral methods

- Particle-based methods

- **Lattice Boltzmann methods (LBM)**

Lattice Boltzmann methods

- Do not solve the Navier-Stokes equations. Model the fluid as discrete fictive particles, that perform propagation and collision/relaxation steps.
- Fictive particles move in discrete directions (lattice). They transport density over space and time.
- LBM can also be understood as a discrete version of the Boltzmann equation (statistical/molecular description of a probability density function for the velocity and momentum).
- Under some assumptions, can re-derive the NS equations from the discrete LB equations.



Example of lattice:
D2Q9 (2D, 9 directions)

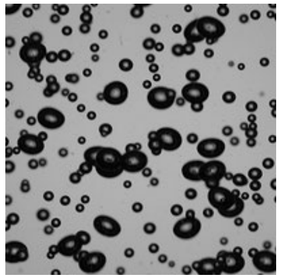
- Pros:
  - Runs efficiently in parallel
  - Suitable for multi-phase flows
  - Handles well complex geometries
- Cons:
  - High-Mach number flows
  - Multi-phase limited to small density ratios

Summary: overview of some characteristics

	Main advantage	Meshing capability
▪ Finite difference (FDM)	Very easy to implement	Simple geometries
▪ Finite volume (FVM)	Conservation is enforced	Complex geometries
▪ Finite element (FEM)	Strong mathematical foundation	Complex geometries
▪ Spectral element (SEM)	Combines flexibility of FEM with accuracy of spectral methods	Rather complex geometries
▪ Spectral methods	High accuracy (for smooth solutions)	Simple geometries
▪ Particle-based methods	Suitable for multi-phase. Easily parallelizable	Complex geometries
▪ Lattice Boltzmann (LBM)	Suitable for multi-phase. Easily parallelizable	Complex geometries

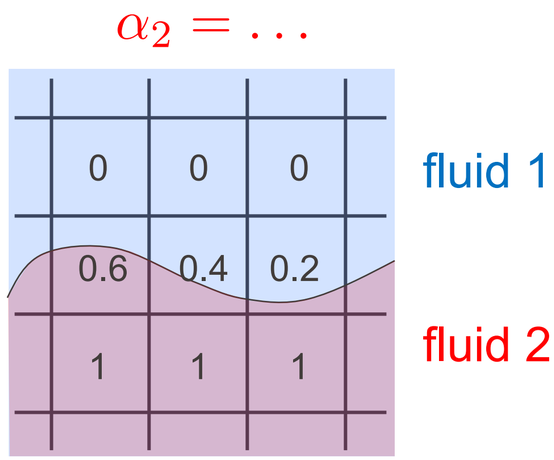
Techniques for fluid interfaces

- Relevant to two-phase flows: immiscible liquids, liquid-gas free surface, bubbles, etc.


- In particle methods (mesh-free, Lagrangian): no additional treatment needed
- In other methods (mesh-based, Eulerian), several options:
  1. Volume of fluid (VOF)
  2. Level-set
  3. Deforming mesh

Techniques for fluid interfaces



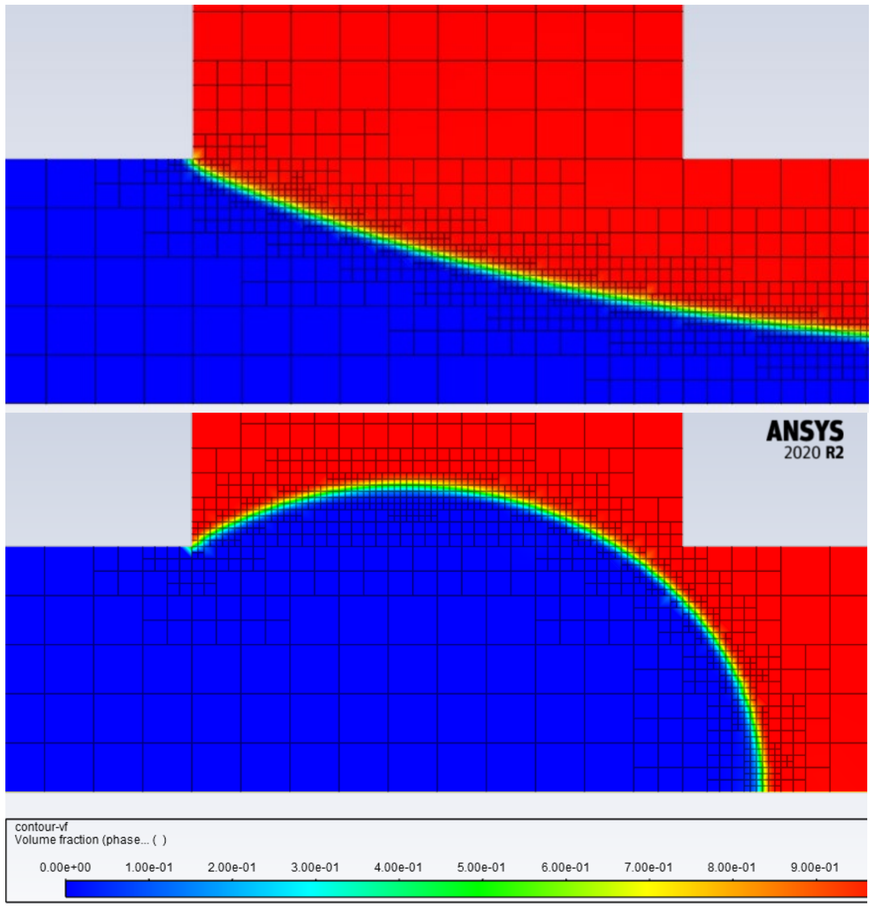
1. Volume of fluid (VOF)

- Each phase *i* characterized by its **volume fraction** α_i
  - $\alpha_i = 0$: the mesh cell is empty of fluid *i*,
  - $\alpha_i = 1$: the mesh cell is full of fluid *i*,
  - $0 < \alpha_i < 1$: the mesh cell contains the interface between fluid *i* and one or more fluids.
  - In each cell: $\sum_i \alpha_i = 1$ (the volume is constant)
- Appropriate properties and variables (e.g. density) are assigned to each cell (local weighted average): $$f = \sum_i \alpha_i f_i$$
- Additional transport equations for volume fractions: $$\frac{\partial \alpha_i}{\partial t} + (\mathbf{u} \cdot \nabla)\alpha_i = 0$$
- Interface not solved for, but **approximately reconstructed**.
- Pros: conserves volume; easy to implement.
- Cons: smears the surface; spatial derivatives (e.g. needed to compute the curvature for surface tension effects) difficult to compute (α_i discontinuous across the interface)

Techniques for fluid interfaces

1. Volume of fluid (VOF)

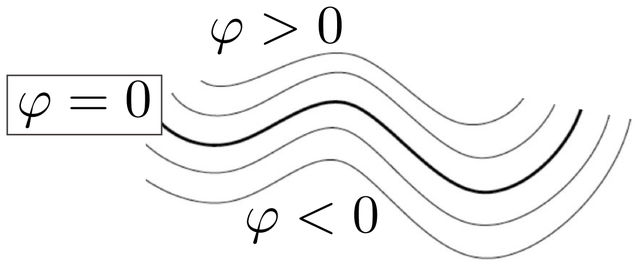
- Volume fraction is a convenient quantity for dynamic mesh adaption: refine near the interface, coarsen far from it, update over time → accurate solution, efficient usage of resources. (In Fluent: pre-defined adaption setting for VOF.)
- Example: “flow-focusing” microfluidic device for droplet production. *Re* << 1.



Techniques for fluid interfaces

2. Level-set

- Interface = a certain iso-contour of a globally defined level-set function φ.
- For ex., φ = signed **distance** to the interface; interface: iso-contour $\varphi = 0$.



- Interface location explicitly **solved for** with a transport equation:

$$\frac{\partial \varphi}{\partial t} + (\mathbf{u} \cdot \nabla)\varphi = 0$$

(In practice, additional source term for numerical stability.)

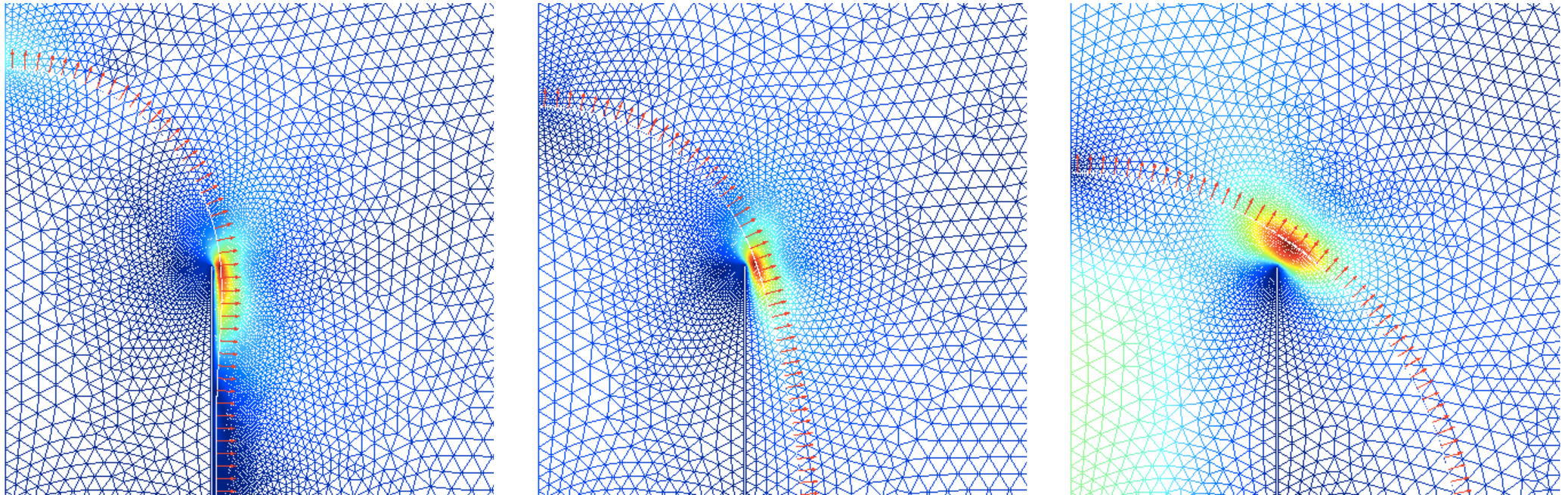
- Pros: smooth and continuous, spatial derivatives can be calculated accurately.
- Cons: does not conserve volume; numerically challenging (e.g. highly distorted interfaces).

Techniques for fluid interfaces

3. Deforming mesh

- Solve flow in each phase with appropriate BC (involving stresses + surface tension).
- Move the interface with the fluid velocity.
- Deform the mesh: given the interface position, move interior mesh points according to a smoothing method (e.g. Poisson equation, or “material” behavior inspired by solid mechanics).
- Example: liquid jet in air, pre-wetted nozzle walls, small Weber number (surface tension >> inertia)

Numerical Flow Simulation



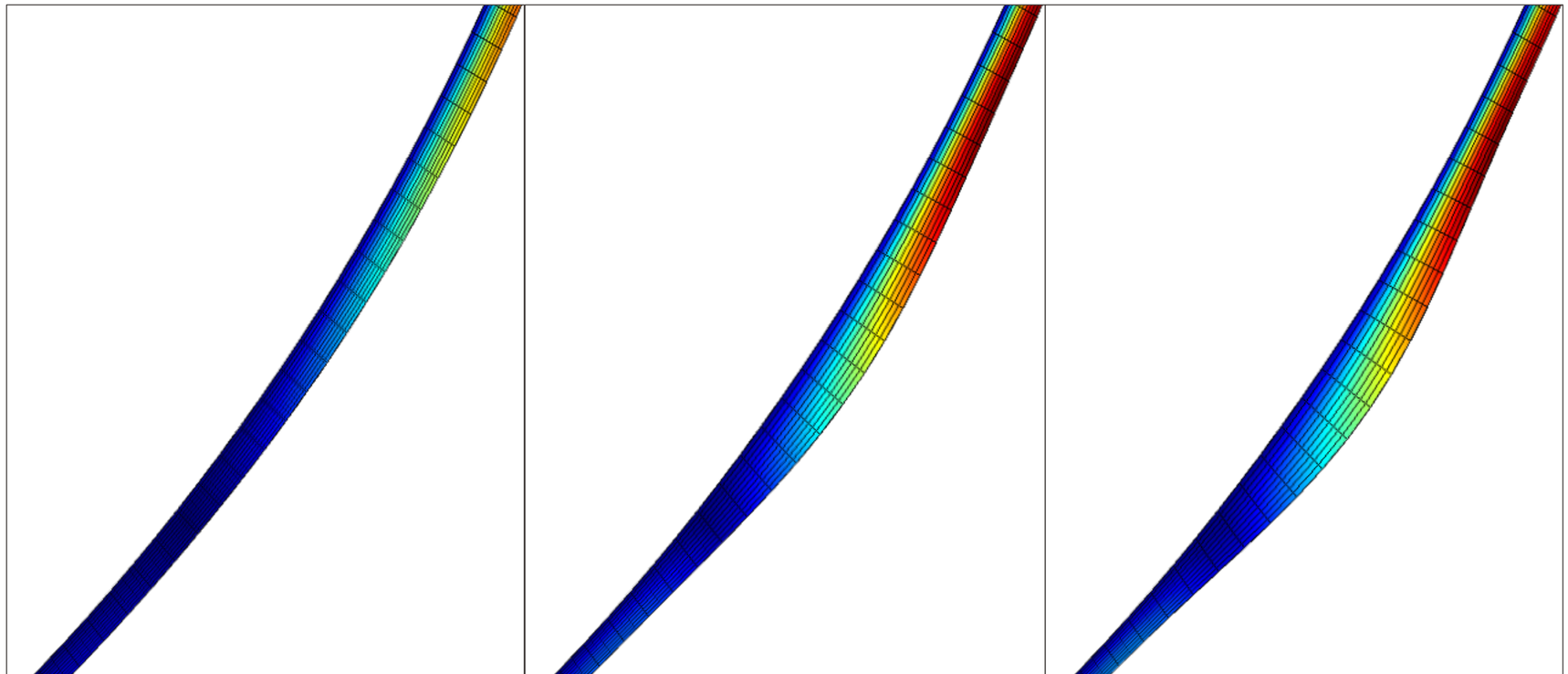
(Comsol, courtesy of S. Eghbali)

Techniques for fluid interfaces

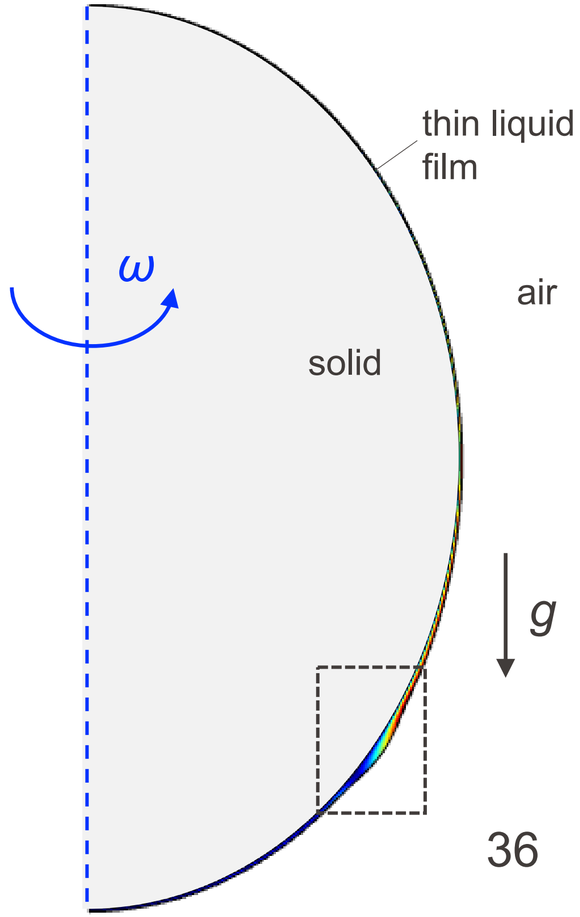
3. Deforming mesh

- Useful to reduce the computational cost when the flow is of interest in one phase only (if can neglect the flow in other phases, and its effect on the interface): mesh and solve for the flow in that phase only.
- Example: gravity-driven coating of the outer surface of a solid axisymmetric ellipsoid rotating around its vertical axis. *Re* << 1.

Numerical Flow Simulation



(Comsol)



Thank you for your attention!