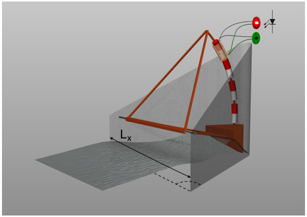
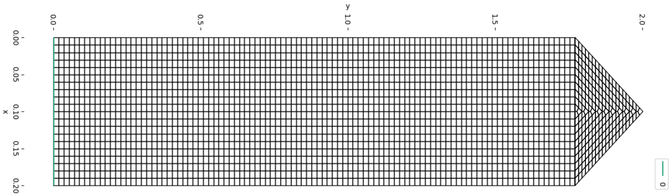
Numerical methods for fluid dynamics

Onno Bokhove,
£: CDT Fluid Dynamics



Outline: assessment

- Attendance at practical sessions.
- Three numerical exercises (on finite difference, finite volume and finite element methods); to be handed in individually but you may work as a team.
- Example programs (for use at your own risk) will be provided in Python. Python use is recommended.



Finite differences: θ-method

- θ–method will be explained for heat equation w. Dirichlet BCs:

$$\partial_t u = \partial_{xx} u \tag{1}$$
$$u(x,0) = u_0(x) \tag{2}$$
$$u(0,t) = u(1,t) = 0. \tag{3}$$

- Mesh points are $x_j = j\Delta x$; constant time step is used $t_n = n\Delta t$ for $j = 0, \dots, N_x$ and $n = 0, 1, \dots$.
- Time step can also be varied, in which case Δt_n varies and t_n is sum of time steps taken.
- Approximate values of $u(x,t)$ on mesh points are denoted by $U_j^n \approx u(x_j, t_n)$.
- Initial values are $U_j^0 = u_0(x_j)$; in general exact (why could there be an issue here?).

Finite differences: approximations

- The next issue is to find a difference approximation of the PDE (1) in terms of the approximations U_j^n.
- Time derivative is approximated in a forward manner, expressed in terms of several difference operators Δ_{+t} and δ_t:

$$
\begin{aligned}
(\partial_t u)(x_j, t_n) &\approx \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{\Delta t} && (4)\\
&\equiv \frac{\Delta_{+t} u(x_j, t_n)}{\Delta t} && (5)\\
&\equiv \frac{\delta_t u(x_j, t_{n+1/2})}{\Delta t} && (6)\\
&\approx (\partial_t u)(x_j, t_{n+1/2}) && (7)
\end{aligned}
$$

- *Exercise:* check approximations by performing suitable Taylor expansions of u around, e.g., $t^n = t_n$ or $t^{n+1/2} = t_{n+1/2}$.

Finite differences: Taylor expansions

- 2^{nd} spatial derivative approximated symmetrically as

$$(\partial_{xx} u)(x_j, t_n) \approx \frac{u(x_{j-1}, t_n) - 2u(x_j, t_n) + u(x_{j+1}, t_n)}{\Delta x^2} \tag{8}$$

$$= \frac{\delta_x^2 u(x_j, t_n)}{\Delta x^2} = \frac{\delta_x(\delta_x u)|_{x_j}^{t_n}}{\Delta x^2} \tag{9}$$

  with $\delta_x u(x,t) \equiv \big(u(x + \Delta x/2, t) - u(x - \Delta x/2, t)\big)/\Delta x$.
- *Exercise:* check this approximation y using Taylor expansions of u around, e.g., t^n and x_j.
- This approximation also holds at t_{n+1}

$$(\partial_{xx} u)(x_j, t_{n+1}) \approx \frac{u(x_{j-1}, t_{n+1}) - 2u(x_j, t_{n+1}) + u(x_{j+1}, t_{n+1})}{\Delta x^2}$$

$$= \frac{\delta_x^2 u(x_j, t_{n+1})}{\Delta x^2}. \tag{10}$$

Finite differences: θ scheme

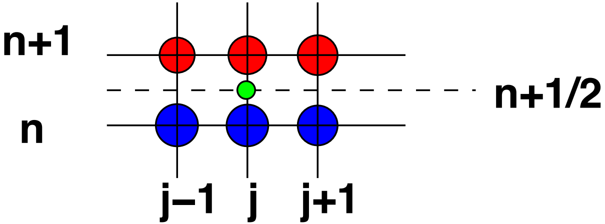
- By combining approximations with $\mu = \Delta t / \Delta x^2$, PDE (1) can be approximated on a 6-point stencil (see Fig.))

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{\theta}{\Delta x^2}\delta_x^2 U_j^{n+1} + \frac{(1-\theta)}{\Delta x^2}\delta_x^2 U_j^n \tag{11a}$$

$$U_j^{n+1} = U_j^n + \mu\theta(U_{j-1}^{n+1} - 2U_j^{n+1} + U_{j+1}^{n+1}) + \mu(1-\theta)(U_{j-1}^n - 2U_j^n + U_{j+1}^n). \tag{11b}$$

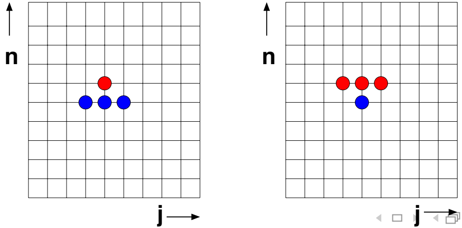
- Rewritten form with unknowns on the LHS and $0 \le \theta \le 1$

$$-\mu\theta U_{j-1}^{n+1} + (1+2\mu\theta)U_j^{n+1} - \mu\theta U_{j+1}^{n+1} = (1 - 2\mu(1-\theta))\,U_j^n + \mu(1-\theta)(U_{j-1}^n + U_{j+1}^n). \tag{12}$$



Finite differences: EF, EB, CN schemes

- When $\theta = 0$ scheme is explicit, solved for U_j^{n+1}: *Euler forward* scheme uses stencil of 4 points with 1 point in future.
- When $\theta = 1$ scheme is fully implicit; *Euler backward* scheme. Uses a stencil of 4 points with 3 points in future, see Fig.
- When $\theta = 1/2$, a stencil of 6 points is used: this is the classical Crank-Nicolson scheme (Crank & Nicolson 1947).
- *Exercise θ–scheme:* suitable space-time grid point for a Taylor expansion?



Finite difference methods: homework

- Study Morton and Mayers (2005), Chapter 2, sections 2.2, 2.4, 2.5.
- Read Morton and Mayers (2005), Chapter 2 (intro), sections 2.1, 2.3.
- Sign up to GitHub and send login name.
- Run/study the two example codes and study the example task.
- Study and start exercise-I.

Finite difference methods: Fourier analysis

- To assess stability of scheme, Fourier analysis will be used. Substitute Fourier Ansatz $U_j^n = \lambda^n e^{ijk\Delta x}$ with imaginary number i satisfying $i^2 = -1$, amplification factor λ, and wavenumber k into discretization (11).
- In general λ is complex with real and imaginary parts such that $\lambda = \Re(\lambda) + i\,\Im(\lambda)$. Scheme is stable when $|\lambda| \leq 1$, which for complex λ implies that we need to take the modulus of λ.
- When $|\lambda| > 1$, approximation U_j^n will blow up over time since $|\lambda|^n$ becomes unbounded and the scheme is unstable.

$$-\theta\mu\lambda e^{-ik\Delta x} + (1+2\theta\mu)\lambda - \theta\mu\lambda e^{ik\Delta x} = 1 - 2(1-\theta)\mu + (1-\theta)\mu(e^{ik\Delta x} + e^{-ik\Delta x}) \tag{13}$$

$$\iff \lambda + \lambda\theta\mu(2 - e^{ik\Delta x} - e^{-ik\Delta x}) = 1 - (1-\theta)\mu(2 - e^{ik\Delta x} - e^{-ik\Delta x}) \tag{14}$$

$$\iff \lambda = \frac{1 - 4(1-\theta)\mu\sin^2(k\Delta x/2)}{1 + 4\theta\mu\sin^2(k\Delta x/2)}. \tag{15}$$

Finite difference methods: Fourier analysis

- Note that this scheme is a special, symmetric case for which λ is real. Since $0 \leq \theta \leq 1$ and $\mu > 0$, we note that $\lambda < 1$.
- Instability can then occur only when $\lambda < -1$, i.e., when

$$1 - 4(1-\theta)\mu \sin^2(k\Delta x/2) < -\left(1 + 4\theta\mu \sin^2 k(\Delta x/2)\right)$$
$$\implies 4\mu(1-2\theta)\sin^2(k\Delta x/2) > 2. \quad (16)$$

- Instability occurs for $\mu(1-2\theta) > 1/2$ for case $k\Delta x/2 = \pi/2$. For $\theta \geq 1/2$ the θ–scheme unconditionally stable, while for $0 \leq \theta < 1/2$ scheme conditionally stable when

$$\mu = \frac{\Delta t}{(\Delta x)^2} < \frac{1}{[2(1-2\theta)]}. \quad (17)$$

- Note that we ignored the boundary conditions in deriving Fourier stability because we imposed the harmonic Ansatz for periodic boundaries. Other BCs, see Morton and Mayers 2005.

Finite difference methods: maximum principle

Theorem

M&M 2005: The θ–method (11) satisfies

$$
\begin{aligned}
U_{min} &\leq U_j^n \leq U_{max} \\
U_{min} &= \min\left(U_0^m, 0 \leq m \leq n; U_j^0, 0 \leq j \leq N_x; U_{N_x}^m, 0 \leq m \leq n\right) \\
U_{max} &= \max\left(U_0^m, 0 \leq m \leq n; U_j^0, 0 \leq j \leq N_x; U_{N_x}^m, 0 \leq m \leq n\right)
\end{aligned}
$$

given the conditions $0 \leq \theta \leq 1$ and $\mu(1-\theta) \leq 1/2$.

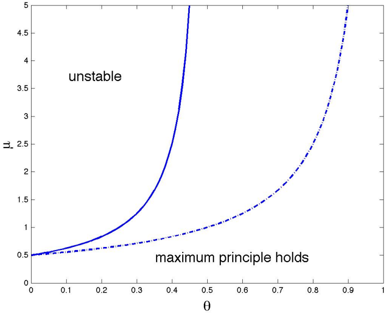
Finite difference methods: maximum principle

- Verification of conditions required for maximum principle to hold forms a practical test for stability of numerical finite difference schemes for diffusion and advection-diffusion equations.
- Maximum principle states that value of variable U_j^n bounded between boundary values and initial values. E.g., when u is seen as a temperature, temperature values can not go above or below temperatures imposed initially or at boundaries.

Finite difference methods: stability criteria

The Fourier stability condition (17) and condition for maximum principle, displayed in Fig.:

$$\mu = \frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{[2(1-2\theta)]} \quad \text{and} \quad \mu(1-\theta) \leq \frac{1}{2}. \qquad (18)$$



Finite difference methods: homework, week 2

- Study Morton and Mayers (2005), Chapter 2, sections 2.7, 2.11.
- Read Morton and Mayers (2005), Chapter 2, sections 2.8, 2.9, 2.10.
- Continue/finish exercise-I.
- Urgent: get *Firedrake* installed on your machines, asap, please!
- . . . hints for Exercise-I . . . .

Finite volume or Godunov method

- Finite volume methods may be most natural for hyperbolic PDEs expressed as conservation laws.
- We only consider the 1D case here:

$$\partial_t \mathsf{u} + \partial_x(\mathsf{f}(\mathsf{u}) = 0 \qquad \text{or} \tag{19}$$

$$\partial_t \mathsf{u}_i + \partial_{x_j} \mathsf{f}_{ij} = 0 \quad \text{with} \quad j = 1 \tag{20}$$

with $\mathsf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathsf{u} = (f_1, f_2, \ldots, f_n)$.

Finite volume: example conservation laws

Examples:

- Euler equations in 2D:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{pmatrix} =0 \qquad (21)$$

  with u and v velocity components in x and y, p pressure and E total energy. EOS: $p=(\gamma-1)(E-\frac{1}{2}\rho(u^2+v^2))$.

- Shallow-water equations in 1D:

$$\partial_t h+\partial_x(hu) =0, \partial_t(hu)+\partial_x(hu^2+\frac{1}{2}gh^2)=0 \qquad (22)$$

  with water depth $h(x,t)$ and depth-averaged velocity $u(x,t)$.

Finite volume: examples

Examples of conservative systems with extra terms:

- Width-averaged shallow-water or St. Venant equations:

$$\partial_t A + \partial_s(Au) = S \tag{23}$$

$$\partial_t(Au) + \partial_s\left(Au^2 + gAh\right) = gh\partial_s A - gA\partial_s b - F, \tag{24}$$

with source and friction terms $S = S(s,t)$,
$F = gC_m Au|u|/R(A,s)^{4/3}$; along-river coordinate s;
cross-section $A(s,t)$; water depth $h = h(A,s)$; depth-averaged velocity $u(s,t)$; river slope $-\partial_s b$, and, accelation of gravity g.

Finite volume: examples

- Limit of kinematic river equation, conservative with extra source term:

$$\partial_t A + \partial_s \left(A R(A,s)^{2/3} \sqrt{-\partial_s b} / C_m \right) = S, \qquad (25)$$

with Manning coefficient C_m, hydraulic radius $R(A, s)$ (wetted area A over wetted perimeter) and "volume" $S(s, t)$.

Finite volume: overview for Burgers-advection system

- Illustrate Godunov method or finite volume discretization for simple system of hyperbolic equations.
- Step by step discretization for system of uncoupled Burgers' and linear advection equations

$$u_t + \left(u^2/2\right)_x = 0 \qquad (26)$$
$$v_t + a\, v_x = 0 \qquad (27)$$

  with $a > 0$ constant, $u = u(x,t)$ and $v = v(x,t)$ on $x \in [0, L]$, $(\cdot)_t = \partial_t$, etc.
- Boundary conditions required, not specified presently.
- Initial conditions $u(x,0)$ and $v(x,0)$ are given at $t = t_0 = 0$.

Finite volume method: studying

From Leveque's book *Finite volume methods*:

- Study §1.1 (conservation law) §3.1 (linear advection Eq.) till §3.1.1 & 3.2 (Burgers' Eq).
- Study §3.1, §3.3 (shock formation).

Godunov method example: Step-1

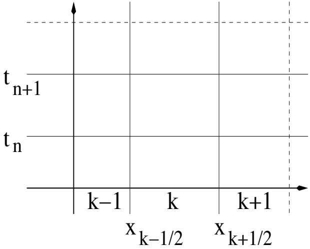
- Exposition of simple uncoupled system plus algorithm facilitates implementation for genuinely coupled fluid system, presented elsewhere.
- *Step 1*:
  System (26) is written concisely in hyperbolic form (20)

$$\mathrm{u}_t + (\mathrm{f}(\mathrm{u}))_x = 0, \tag{28}$$

  after identification $\mathrm{u} = (u, v)^T$ and flux $\mathrm{f}(\mathrm{u}) = (u^2/2, a\,v)^T$ (transpose $(\cdot)^T$).

Godunov method example: Step-2

- *Step 2:* Define space-time mesh with N "finite volumes" on domain $x \in [0, L]$ in time interval $I_n = [t_n, t_{n+1}]$ (Fig. 22).
- Cell k occupies $x_{k-1/2} < x < x_{k+1/2}$ and $k = 1, 2, \ldots, N$.
- $N + 1$ cell boundaries $x_{1/2}, \ldots, x_{N-1/2}, x_{N+1/2}$. Cell lengths $h_k = x_{k+1/2} - x_{k-1/2}$ and time step $\Delta t_n = t_{n+1} - t_n$ may vary.
- There are $n = 0, \ldots, N_t$ time intervals I_n, where $t = t_n$ is the time after n time steps, initial conditions at $t = t_0 = 0$.



Godunov method example: Step-3

- *Step 3:* Integrate (28) in space-time element $x_{k-1/2} < x < x_{k+1/2}$ and $t_n < t < t_{n+1}$, Fig. 22.
- Via coordinate transformation $x' = x - x_{k+1/2}$, $t' = t - t_n$, right-bottom corner becomes origin $(x', t')^T = (0, 0)^T$.
- After integration of (28) over space-time element:

$$\mathrm{U}_k^{n+1} = \mathrm{U}_k^n - \frac{1}{h_k} \int_{t_n}^{t_{n+1}} \mathrm{f}_{k+1/2}(t) - \mathrm{f}_{k-1/2}(t)\,\mathrm{d}t, \tag{29}$$

with mean cell average U_k in cell k

$$\mathrm{U}_k(t) = \frac{1}{h_k} \int_{x_{k-1/2}}^{x_{k+1/2}} \mathrm{u}(x, t)\,\mathrm{d}x. \tag{30}$$

- Flux is at the cell boundaries: $\mathrm{f}_{k+1/2}(t) = \mathrm{f}\left(\mathrm{u}(x = x_{k+1/2}, t)\right)$.
- $\mathrm{U}_k(t)$ in (30) and $\mathrm{f}_{k+1/2}(t)$ still functions of time t, and $\mathrm{U}_k^n = \mathrm{U}_k(t = t_n)$, etc.

Godunov method example: Step-3

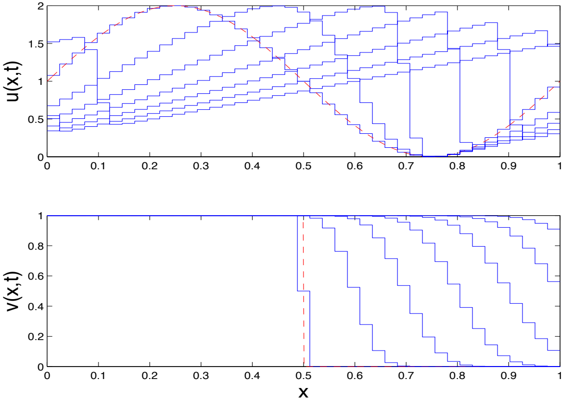
- Integral expression (29) exact provided that $\mathrm{u}(x,t)$ known. Start at $n=0$, calculate $\mathrm{U}_k^0 = \mathrm{U}_k(t=t_0)$ using (30).
- Graphically, U_k^0 is projection of initial data on piecewise constant profiles at time t_0, cf. initial step profiles in Fig. 25.
- Determine $\mathrm{f}_{k+1/2}(t)$ over $t_n < t < t_{n+1}$ in (29) to obtain

$$\mathrm{U}_k^{n+1} = \mathrm{U}_k^n - \frac{\Delta t}{h_k}\left(\mathrm{F}_{k+1/2}(\mathrm{U}_k^n, \mathrm{U}_{k+1}^n) - \mathrm{F}_{k-1/2}(\mathrm{U}_{k-1}^n, \mathrm{U}_k^n)\right) \tag{31}$$

with (approximate) numerical flux

$$\mathrm{F}_{k+1/2}(\mathrm{U}_k^n, \mathrm{U}_{k+1}^n) \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathrm{f}_{k+1/2}(t)\,\mathrm{d}t. \tag{32}$$

Godunov method example: Step-3



Godunov method example: Step-4

- *Step 4:* Calculate numerical fluxes $\mathrm{F}_{k+1/2}(\mathrm{U}_k^n, \mathrm{U}_{k+1}^n)$ in (32) at all nodes $x_{k+1/2}$ for $k = 0, 2, \ldots, N$.
- There are various strategies to find (32), usually involving an approximation.
- Consider classical Godunov strategy based on the Riemann solution.
- Calculate $\mathrm{F}_{k+1/2}(t)$ in (32) exactly over $t_n < t < t_{n+1}$ in (29), only feasible for piecewise constant approximation U_k^n at time t_n and starting with projected initial condition U_k^0, instead of smoother initial condition.
- Solution of Riemann problem in new coordinates, i.e. with $\mathrm{u} = \mathrm{u}(x', t')$ and $\mathrm{f} = \mathrm{f}\left(\mathrm{u}(x', t')\right)$, provides such exact solution.

Godunov method example: Riemann problem

- The Riemann problem is defined as

$$\frac{\partial \mathrm{u}}{\partial t'} + \frac{\partial \mathrm{f}}{\partial x'} = 0 \tag{33}$$

for special initial conditions

$$\mathrm{u}(x', t' = 0) = \begin{cases} \mathrm{u}_{left} = \mathrm{u}_l = \mathrm{U}_k^n & \text{for} \quad x' < 0 \\ \mathrm{u}_{right} = \mathrm{u}_r = \mathrm{U}_{k+1}^n & \text{for} \quad x' \geq 0 \end{cases} . \tag{34}$$

- Riemann solution such that $u(x', t')$ constant along characteristics $x' = x_0' + C\, t'$ for some C depending on $\mathrm{u}_{l,r}$.
- $\mathrm{f}_{k+1/2}(t) = \mathrm{f}_{k+1/2}(\mathrm{u}(x_{k+1/2}, t))$ is constant; (32) can be integrated exactly —note that $\mathrm{u}(x' = 0, t') = \mathrm{u}(x_{k+1/2}, t)$ due to coordinate change.

Godunov method example: Riemann problem

NB:

- While formula (29) exact, numerical approximation occurs because we continue from a piecewise constant profile with value U_k^n in cell k at a certain time t_n.
- At cell boundary $x_{k+1/2}$ between cells k and $k+1$, adjacent values are U_k^n and U_{k+1}^n.
- When other cell boundaries $x_{k-1/2}$ and $x_{k+3/2}$ are "far away" in sense that time step Δt is "small enough", we are locally dealing with a Riemann problem around cell boundary $x_{k+1/2}$.
- Although Riemann solution no longer piecewise constant within each cell at end of time step, we continue only with information stored in the mean values U_k^{n+1}.
- This is the approximation made.

Godunov method example: Riemann problem

- The characteristic form of (26)–(27) is as follows

$$\frac{\mathrm{d}u}{\mathrm{d}t} = 0 \quad \text{on} \quad \frac{\mathrm{d}x}{\mathrm{d}t} = u \tag{35}$$
$$\frac{\mathrm{d}v}{\mathrm{d}t} = 0 \quad \text{on} \quad \frac{\mathrm{d}x}{\mathrm{d}t} = a, \tag{36}$$

which has solution

$$x = x_{01} + ut, u = u_0(x_{01}) \quad \text{s. t.} \quad u(x,t) = u_0\left(x - tu(x,t)\right)$$
$$x = x_{02} + at, v = v_0(x_{02}) \quad \text{s. t.} \quad v(x,t) = v_0(x - at).$$

- At $t = 0$ we see that $x = x_{01}$ or $x = x_{02}$ and these parameterise the domain for each equation.
- Instead of two PDEs, we have four ODEs.
- For constant a, solution linear advection equation is shift of original profile to left or right, depending on sign of a.

Godunov method: studying

From Leveque's book *Finite volume methods*:

- Study §3.5 (Riemann problem), §3.6 (shock speed).
- Study §13.2 (Godunov method), §13.5 (Godunov for scalar conservation laws); read Chap. 9.

Godunov method example: Riemann problem

The Riemann problem for the uncoupled system (26) special: consists of separate Riemann problems for two equations.

- Solution of Riemann problem for linear advection equation:

$$\frac{\partial v}{\partial t'} + \frac{\partial (a\, v)}{\partial x'} = 0 \tag{37}$$

  with constant $a > 0$.

- NB: piecewise constant initial conditions

$$v(x', t' = 0) = \begin{cases} v_{left} = v_l = V_k^n & \text{for} \quad x' < 0 \\ v_{right} = v_r = V_{k+1}^n & \text{for} \quad x' \geq 0 \end{cases} . \tag{38}$$

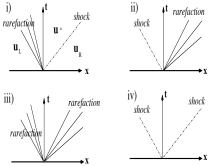
- For $a > 0$ all characteristics are $x' = x_0 + a\, t'$. Solution $v(x' = 0, t') = v(x_{k+1/2}, t) = v_l^n$ with $\mathrm{f}\left(v(x_{k+1/2}, t)\right) = a\, v_l^n$ within a sufficiently small time interval.

Godunov method example: Riemann problem

Linear advection equation: Integral (32) straightforward to evaluate; Godunov scheme (31) becomes

$$V_k^{n+1} = V_k^n - \frac{\Delta t}{h_k} a \left(V_k^n - V_{k-1}^n\right). \tag{39}$$

- Correspondence of (39) with upwind finite difference discretization is clear (Chapter 4 in M& M), but V_k^n is mean value of $v(x, t_n)$ over cell k and not a grid point value.



Godunov method example: CFL condition

- Application of maximum principle to (39) yields, by imposing that all coefficients 1, $(1 - \Delta t a/h_k)$ and $a\Delta t/h_k$ of $V_k^{n+1}, V_k^n, V_{k-1}^n$ are larger than zero (and $a > 0$):

$$V_k^{n+1} = (1 - \frac{\Delta t}{h_k} a) V_k^n + \frac{\Delta t}{h_k} a V_{k-1}^n \tag{40}$$

$$\Longrightarrow 1 - \frac{\Delta t}{h_k} a > 0 \Longleftrightarrow \Delta t < \frac{h_k}{a}. \tag{41}$$

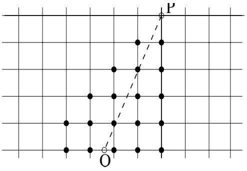
- The well-known Courant-Friedrichs-Lewy or CFL condition.
- When $a < 0$, the upwind discretization should be:

$$V_k^{n+1} = V_k^n + \frac{\Delta t}{h_k} |a| \, (V_{k+1}^n - V_k^n). \tag{42}$$

- For general a, the CFL condition thus reads $\Delta t < h_k/|a|$, which also makes sense dimensionally since a is a "wind" speed.

Godunov method example: CFL condition

Graphically, upwinding for linear advecion equation:

(a) 

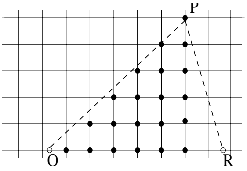
(b) 

Figure: Consider the linear advection equation $u_t + a\, u_x = 0$ with $a > 0$. (a) The solution $u(x,t) = u^0(x - at)$ has a characteristic tracing through point P back to point Q satisfying the CFL condition $\Delta t < \Delta x/|a|$. (b) The CFL condition is violated when $\Delta t > \Delta x/|a|$ as the information transported along the characteristic falls outside the discretization stencil. It shows the dependence of the numerical solution on the initial data.

Godunov method example: Riemann problem

- Burgers' equation allows discontinuous or shock solutions, where $u(x,t)$ obtains different limiting values.
- Discontinuity resides at position $x = x_b(t)$ and moves with shock speed $s \equiv \mathrm{d}x_b/\mathrm{d}t$.
- Integrate Burgers' equation in (26) around $x_b(t)$, and let $\epsilon \to 0$, to obtain

$$\begin{aligned} \lim_{\epsilon\to 0}\int_{x_b-\epsilon}^{x_b+\epsilon}\frac{\partial u(x,t)}{\partial t}\mathrm{d}x + \int_{x_b-\epsilon}^{x_b+\epsilon}\frac{\partial (u^2/2)}{\partial x}\mathrm{d}x &= 0\\ \Leftrightarrow \frac{\partial}{\partial t}\lim_{\epsilon\to 0}\int_{x_b(t)-\epsilon}^{x_b(t)+\epsilon} u(x,t)\mathrm{d}x - \frac{\mathrm{d}x_b(t)}{\mathrm{d}t}\lim_{\epsilon\to 0} u(x,t)\big|_{x_b(t)-\epsilon}^{x_b(t)+\epsilon} + [u^2/2] &= 0\\ \Leftrightarrow -\frac{\mathrm{d}x_b(t)}{\mathrm{d}t}[u] + [u^2/2] = 0 \Leftrightarrow s = \frac{\mathrm{d}x_b(t)}{\mathrm{d}t} = \frac{[u^2/2]}{[u]} &= \frac{1}{2}(u_l + u_r) \end{aligned} \tag{43}$$

  with $[u] = u_r - u_l$, $[u^2/2] = (u_r^2 - u_l^2)/2$, $u_r = u(x,t)|_{x\downarrow x_b}$ and $u_l = u(x,t)|_{x\uparrow x_b}$.
- In literature, approximate numerical fluxes are found such as Roe solvers and kinetic fluxes.

Godunov method example: Riemann problem Burgers

- *Exercise-Burgers:* (i) Derive the Rieman solution for the Riemann problem of the Burgers' equation.
- *Exercise-Burgers:* (ii) Complete the Godunov method for the Burgers' equation using this Riemann solution.
- See also `https://www.youtube.com/watch?v=izMsj639hGI` and `https://www.youtube.com/watch?v=goL8_rET1H0`

Provide both the actual solution in formula form as well as the solution in graphical form in the characteristic t,x-plane (or t',x'-plane).

Godunov method example: Riemann problem Burgers

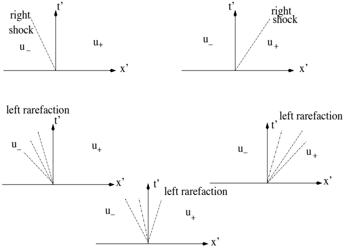


Figure: Graphical solution of Riemann problem for Burgers' equation. $u_l > u_r$: shock wave with shock spreed $s = (u_l + u_r)/2$. $u_l \leq u_r$: rarefaction wave results with solution x'/t' in the interval $u_l\, t' < x' < u_r\, t'$. u_l and u_r: initial condition in definition Riemann problem.

Godunov method example: Riemann problem Burgers

- Solution is constant along each characteristic by differentiating solution along the characteristic:

$$\frac{d}{dt}u(x(t),t) = \frac{\partial}{\partial t}u(x(t),t) + \frac{\partial}{\partial x}u(x(t),t)\frac{\mathrm{d}x}{\mathrm{d}t} = u_t + uu_x = 0.$$

- If solution is smooth then it can be constructed by backtracking characteristics to initial solution as long as they do not cross each other: For each (x,t) we can solve the following equations for ξ:

$$u = u_0(\xi) = u(\xi,0) \qquad \text{and} \qquad x = \xi + u(\xi,0)t$$

after which we can determine the solution at time t:

$$u(x,t) = u(\xi,0).$$

- Hence, the implicit solution of Burgers' equation is $u(x,t) = u_0(x - u(x,t)\,t)$, since

$$\begin{aligned} 0 = \frac{d}{dt}u(x(t),t) =& \frac{\partial u(x,t)}{\partial t} + u\frac{\partial u(x,t)}{\partial x} \qquad \text{and} \\ \partial_x u =& u_0'(1 - t\partial_x u) \Longrightarrow \partial_x u = \frac{u_0'}{1+tu_0'} \end{aligned}$$

with u_0' the derivative of u_0 with respect to its argument.

Godunov method example: Riemann problem Burgers

- The solution to the Riemann problem is either a shock wave or a rarefaction wave.
- It is a shock wave when $u_l > u_r$ and a rarefaction wave when $u_l < u_r$, which follows from considering the characteristics $\mathrm{d}x/\mathrm{d}t = u$ in the x–t-plane.
- The shock wave has shock speed $s = (u_l + u_r)/2$ and its position is given by $x' = s\,t'$; to the left of the shock $u(x', t) = u_l$ and to the right $u(x', t) = u_r$.
- Since the numerical flux is evaluated at $x' = 0$ (i.e. at $x = x_{k+1/2}$), the flux $u^2/2$ is thus either $u_l^2/2 = (U_k^n)^2/2$ when $s > 0$, or $u_r^2/2 = (U_{k+1}^n)^2/2$ when $s < 0$ for the shock wave case.

Godunov method example: Riemann problem Burgers

- The rarefaction wave has characteristics $dx'/dt' = u$ on which u is constant. The tail and the head of the rarefaction wave lie at $x' = u_l\, t'$ and $x' = u_r\, t'$, respectively.
- Hence the rarefaction wave solution is

$$u(x',t)' = \begin{cases} u_l & x' < u_l\, t' \\ x'/t' & u_l\, t' < x' < u_r\, t' \\ u_r & x' > u_r\, t' \end{cases} . \tag{44}$$

- We deduce from this solution that $u_l < u_r$. So at $x' = 0$, or $x = x_{j+1/2}$, we find for the rarefaction wave case that $u(0,t') = u_l$ when $u_l > 0$, $u(0,t') = 0$ when $u_l < 0$ and $u_r > 0$, and $u(0,t') = u_r$ when $u_r < 0$.
- Note that $u(x',t') = x'/t'$ is a similarity solution of Burgers' equation.

Godunov method example: Riemann problem Burgers

- Numerical flux function F at each face $x_{k+1/2}$ is defined as:

$$F(U_k^n, U_{k+1}^n) = \frac{1}{\triangle t} \int_{t_n}^{t_{n+1}} \frac{1}{2} \left(u^\dagger(U_k^n, U_{k+1}^n) \right)^2 dt \qquad (45)$$

  with $u^\dagger(U_k^n, U_{k+1}^n) = u(x_{k+1/2}, t)$ for the special piecewise constant data at time t_n.

- Finally, we need to evaluate the integral (45) by combining these shock wave and rarefaction wave solutions for $u_l > u_r$ and $u_l < u_r$ respectively.
- Note that the solution is constant at $x' = x - x_{k+1/2} = 0$, which simplifies the time integration in (45).

Godunov method example: Riemann problem

Homework Exercise-II.

Godunov method example: Firedrake implementation

- The finite volume or Godunov method can be implemented in Firedrake as a discontinuous Galerkin finite element method ("DGFEM") of order 0, abbreviated as DG0.
- Rather than implementing each finite-volume discretisation, equation by equation for each volume, Firedrake implements the system of equations in one go.
- In either case, note that in 1D there are N_K volumes but $N_K + 1$ numerical fluxes (for inflow/outflow) and that each flux $F_{k+1/2}$ is used twice, once as influx in cell $K + 1$ and once as outflux in cell K.
- Hence, a loop to establish the fluxes before a loop over the cells avoids calculating the fluxes twice.

Godunov method example: Firedrake implementation

- Godunov method for cell k (or K):

$$\Delta s_k(\bar{A}_k^{n+1} - \bar{A}_k^n) + \Delta t \left(F(A_{k+1}^n, A_k^n, s)|_{s=s_{k+1/2}} - F(A_k^n, A_{k-1}^n, s)|_{s=s_{k-1/2}} \right) = 0.$$

- Consider this as a DG0 discretisation with test function $w_k = w_K = 1$ in cell K and zero otherwise; multiply by w_K.
- Integral over cell K & boundary integral ("summation" 1D) over element "faces" Γ (points 1D):

$$\int_K w_k(\bar{A}_k^{n+1} - \bar{A}_k^n)\mathrm{d}x + \int_{\partial K} w_k \Delta t \left(F(A_{k+1}^n, A_k^n, s)|_{s=s_{k+1/2}} - F(A_k^n, A_{k-1}^n, s)|_{s=s_{k-1/2}} \right) \mathrm{d}\Gamma = 0.$$

Godunov method in Firedrake

Finite volume or DG0 in Firedrake:

- Sum over all cells:

$$\sum_K \int_K w_k(\bar{A}_k^{n+1} - \bar{A}_k^n)\mathrm{d}x$$
$$+ \sum_K \int_{\partial K} w_k \Delta t \left(F(A_{k+1}^n, A_k^n, s)|_{s=s_{k+1/2}} - F(A_k^n, A_{k-1}^n, s)|_{s=s_{k-1/2}} \right) \mathrm{d}\Gamma = 0.$$

- Transfer the sum over the elements for the fluxes into a sum over the faces and assign each flux contribution per face to two equations!

Godunov method in Firedrake: faces

- This transfer leads to two contributions (Ambati & B. 2007): one from the inside of that element and from the adjacent element to that face (outward normal used)

$$
\begin{aligned}
\sum_K \int_{\partial K} w \hat{n} F \mathrm{d}\Gamma &= \sum_\Gamma \int_\Gamma \hat{n}_l F^l w^l + \hat{n}_r F^r w^r \,\mathrm{d}\Gamma \\
&= \sum_\Gamma \int_\Gamma (\alpha F^l + \beta F^r)(\hat{n}_l w^l + \hat{n}_r w^r) + (\hat{n}_l F^l + \hat{n}_r F^r)(\beta w^l + \alpha w^r)\,\mathrm{d}\Gamma \\
&= \sum_\Gamma \int_\Gamma (\alpha F^l + \beta F^r)(\hat{n}_l w^l + \hat{n}_r w^r)\,\mathrm{d}\Gamma \\
&\approx \sum_\Gamma \hat{n}^l \hat{F}(U_l, U_r, \hat{n}_l)(w^l - w^r)\,\mathrm{d}\Gamma \qquad (46a)
\end{aligned}
$$

- given that $\hat{n}^l = -\hat{n}^r$ and the flux is continuous $F^l = F^r$ such that $\hat{n}_l F^l = -\hat{n}_r F^r$, wherein, $\alpha + \beta = 1$.
- Notation $(\cdot)^{l,r}$ is arbitrary also in 1D, since each face assigned "left" "right" or "$\pm$" side.
- NB Easiest to derive the above (46) going backwards!

Godunov for river kinematics: wetted $P(A, s)$

Wetted perimemeter $P(A, s)$ as function of cross-sectional river area A and along-river coordinate s:

- rectangular channel of width $w_0(s)$:

$$A = w_0(s)h, \quad P(A, s) = w_0(s) + 2h = w_0(s) + 2A/w_0(s);$$

- narrow rectangular base channel of width w_b and height h_b within wider rectangular flood-plain channel with width $w_0(s)$:

$$A = \begin{cases} w_b h & h < h_b, A < w_b h_b \\ w_b h_b + w_0(s)(h - h_b) & h \geq h_b, A \geq w_b h_b \end{cases},$$

$$P(A, s) = \begin{cases} w_b + 2A/w_b & A < w_b h_b \\ w_b + 2h_b + w_0(s) - w_b + 2(A - w_b h_b)/w_0(s) & A \geq w_b h_b \end{cases}. \tag{47}$$

Godunov for river kinematics: inflow Q_0

Base inflow $Q(s=0,t)=Q_0(t)$ at $s=0$, given depth H_0:

- rectangular channel of width $w_0(s)$:

$$A_0 = w_0(0)H_0, \quad Q_0 = \frac{A_0^{5/3}\sqrt{-\partial_s b}/C_m}{P(A_0,0)^{2/3}}; \tag{48}$$

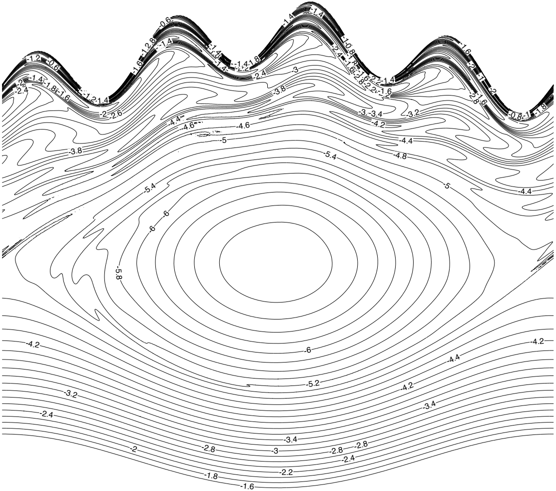
- narrow rectangular base channel of width w_b and height h_b within wider rectangular flood-plain channel with width $w_0(s)$:

$$A_0 = \begin{cases} w_b H_0 & H_0 < h_b, A_0 < w_b h_b \\ w_b h_b + w_0(s)(H_0 - h_b) & H_0 \geq h_b, A_0 \geq w_b h_b \end{cases}, \quad Q(s=0^-) = Q_0 = \frac{A_0^{5/3}\sqrt{-\partial_s b}/C_m}{P(A_0,0)^{2/3}}. \tag{49}$$

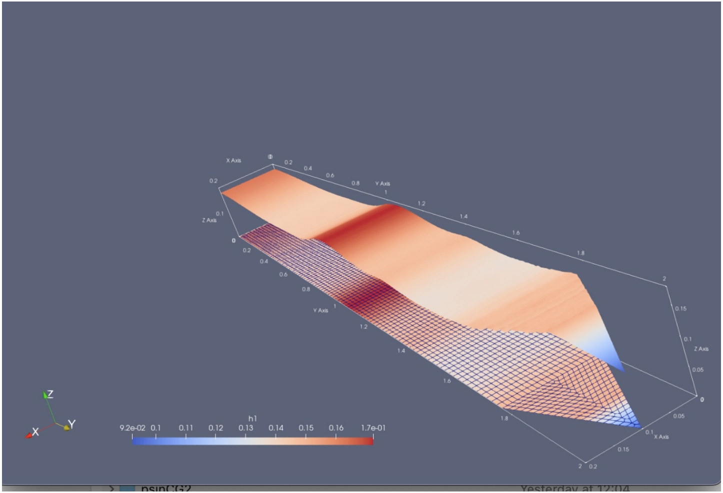
Godunov for river kinematics: code

- Split TC2 in two cases: with constant Q_0 and with a peak $Q_0(t)$. Test.
- Error in code 14-11-2025: use fd.Constant($\cdots$) for constants used in Firedrake-UFL.
- Sign up and use the Firedrake Slack channel to ask about these fd.Constant's and such.
- There is also a Firedrake UoL Teams-page.

Finite Element Method



Finite Element Method



FEMs are commonly used in engineering community, often perceived more to be more difficult. Archetypical examples include:

- 2D Poisson equation $-\nabla^2\phi(x,y) = f(x,y)$ with unknown $\phi = \phi(x,y)$, given function $f = f(x,y)$, coordinates $x, y \in \Omega \in \mathbb{R}^2$ with domain Ω and Dirichlet and/or Neumann BCs at domain boundary $\partial\Omega$;
- 1D hyperbolic linear advection and Burgers' equations, with $u = u(x,t)$,

$$\partial_t u + \partial_x u = 0 \qquad \text{and} \qquad \partial_t u + u\,\partial_x u = 0$$

- 1D advection-diffusion or viscous Burgers' equations, with $u = u(x,t)$ and diffusion coefficient $\kappa > 0$,

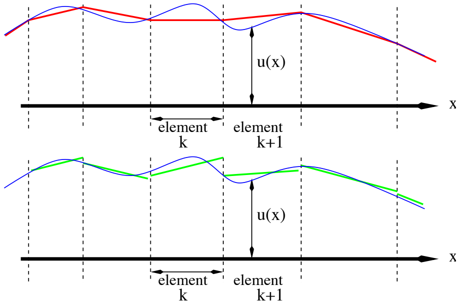
$$\partial_t u + \partial_x u = \kappa\,\partial_{xx} u.$$

Domain is $x \in [0, L]$ with $L > 0$ and IC is $u(x,0) = u_0(x)$. BC for linear advection is $u(0,t) = u_b(t)$, for Burgers' equation BCs depend on characteristics.

Finite Element Method

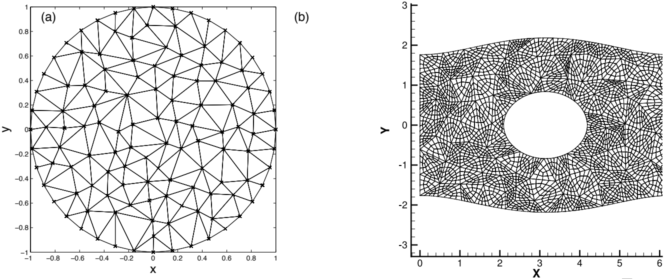
Two finite element methods will be presented (hybrid ones exist):

- (a) a (second-order) continuous Galerkin (CG) finite element method on triangular, quadrilateral or mixed meshes;
- (b) a (space) discontinuous Galerkin (DG) finite element method.



Finite Element Method

Consider the triangular or quadrilateral meshes below. In the CG finite element method considered, the function $\phi(x, y)$ will be approximated by a piecewise linear function per element based on the nodal values of ϕ. Hence, the discretization of ϕ denoted as ϕ_h will be continuous (C^0–continuity).



Finite Element Method: steps

CG and DG FEMs usually contain following steps:

- I. *Derive an integral formulation: either a weak formulation from an equation or from a variational principle:*
  (i) either each equation is multiplied by its own arbitrary test function, integrated over the domain of validity entirely or as a sum of integrals over all elements, and integrated by parts to obtain the weak formulation; or,
  (ii) it follows by variations of a variational principle (VP), by hand or in automated fashion ("ML").
- II. *Form a discretized weak formulation/algebraic system:*
  (iii) Variables are expanded in domain or in each element in a series in terms of a finite number of basis functions. Each basis function has compact support over neighboring elements (for CGFEM) or within each element (for DGFEM). Expansion substituted into weak formulation, test function may be chosen to coincide with a basis function, to obtain discretized weak formulation.

Finite Element Method: steps

CG and DG FEMs usually contain following steps:

- II. . . . Resulting system is a linear or nonlinear algebraic system. (iv) In automated FEM-environments such as Firedrake, these discretisations are done automatically, after choosing mathematicaly appropriate functions spaces, basis- and test functions. For case stemming from a VP and ML, weak forms follow by a "derivative" operation.
- III. *Evaluate integrals in a local coordinate system:* A local or reference coordinate system is used to evaluate integrals. In CGFEM global matrices and vectors are assembled in assembly routine.
- IV. *Solve the algebraic system:* Resulting algebraic system is solved (iteratively) using forward time stepping methods or linear algebra routines (such as PETSc).

Finite Element Method: PDE

- For a given function f, consider the PDE with (in)homogeneous boundary conditions in $\Omega = [0,1] \times [0,1]$:

$$
\begin{aligned}
-\nabla^2 u =& f, \quad \text{e.g.} \quad f = 2\pi^2 \sin(\pi x)\cos(\pi y) && \text{(50a)}\\
u|_{\Gamma_1} =& a \quad \text{e.g.} \quad u(0,y) = u(1,y) = 0 && \text{(50b)}\\
\nabla u \cdot \mathrm{n}|_{\Gamma_2} =& 0, \quad \text{e.g.} \quad (\partial_y u)(x,0) = (\partial_y u)(x,1) = 0, && \text{(50c)}
\end{aligned}
$$

  where the boundary $\partial\Omega = \Gamma_1 \bigcup \Gamma_2$. $\Gamma_1 : x = 0, 1$, $\Gamma_2 : y = 0, 1$.

- Solution $u = u(x,y)$ minimises the *functional* (arguments are functions):

$$I[u] = \iint\limits_{\Omega} \frac{1}{2}\|\nabla u|^2 - uf \,\mathrm{d}\Omega \tag{51}$$

  with smooth space $\Sigma = \{u \quad \text{smooth} | u|_{\Gamma_1} = 0\}$.

- If u minimises (51) then u satisfies (50).

Finite Element Method: step-1

- Minimisation (Riesz' method) with variation $(\delta u)(x, y) = \eta(x, y)$ a function:

$$\delta I \equiv \frac{\mathrm{d}I}{\mathrm{d}\epsilon} = \lim_{\epsilon \to 0} \frac{I[u(x, y) + \epsilon(\delta u)(x, y)] - I[u(x, y)]}{\epsilon} = 0 \quad (52)$$

$$\Longrightarrow \iint_{\omega} \nabla u \cdot \nabla(\delta u) - f \delta u \, \mathrm{d}\Omega = 0, \quad (53)$$

  yielding the *weak formulation*. Further manipulation yields equation of motion, with outward normal $\hat{\mathrm{n}}$ at $\partial\Omega$:

$$\Longrightarrow -\iint_{\omega} (\delta u)(\nabla^2 u + f) \, \mathrm{d}\Omega + \int_{\Gamma_1} \cancelto{0}{\delta u} \nabla u \cdot \hat{\mathrm{n}} \, \mathrm{d}\Gamma + \int_{\Gamma_2} \delta u \cancelto{0}{\nabla u \cdot \hat{\mathrm{n}}} \, \mathrm{d}\Gamma = 0 \quad (54)$$

  using "integration by parts"/Gauss' theorem and the BCs, also $(\delta u)|_{\Gamma_1} = 0$.
- Given the arbitrariness of test function δu, the integrand must be zero pointwise for every (x, y).

Finite Element Method: step-1

- Weak fomulation, multiply by test function $\eta(x, y)$, integrate over Ω and use BCs:

$$\int_\Omega \eta \nabla^2 u + \eta f \, \mathrm{d}\Omega = 0 \tag{55}$$

$$\int_\Omega -\nabla \eta \cdot \nabla u + \eta f \, \mathrm{d}\Omega + \int_{\Gamma_1} \eta \, \nabla u \cdot \hat{\mathsf{n}} \, \mathrm{d}\Gamma + \int_{\Gamma_2} \eta \, \nabla u \cdot \hat{\mathsf{n}} \, \mathrm{d}\Gamma = 0 \tag{56}$$

$$\int_\Omega -\nabla \eta \cdot \nabla u + \eta f \, \mathrm{d}\Omega = 0, \tag{57}$$

similary yielding the *weak formulation*.

Finite Element Method: step-2 CGFEM

- CGFEM: Expand u and η into their compact basis functions $\varphi_j(x,y)$ (using Einstein summation conventoin):

$$u(x,y) \approx u_h(x) = u_j\varphi_j(x,y), \quad \eta(x,y) \approx \eta_h(x,y) = a_j\varphi_j(x,y) \Longrightarrow \eta_h(x,y) = \varphi_i(x,y) \tag{58}$$

  with u_j, a_j known or zero at Dirichlet boundary conditions and taking the simplest case with $a_j = \delta_{ij}$.
- Substitution in the weak formulation yields:

$$A_{ij}u_j = b_i \quad \text{with} \tag{59}$$

$$A_{ij} \equiv \iint_\Omega \nabla\varphi_i(x,y)\cdot\nabla\varphi_j(x,y)\,\mathrm{d}\Omega,\ b_i = \iint_\Omega \varphi_i(x,y)f(x,y)\,\mathrm{d}\Omega \tag{60}$$

$$\Longrightarrow A_{i'j'}u_{j'} = b_{i'} - \sum_{k=N_{nodes}+1}^{N_n} A_{i'k}b_k \tag{61}$$

  with i', j' nodes excluding ones on the Dirichlet boundary Γ_1 and k on the Dirichlet boundary with N_n degrees of freedom (or nodes)and N_{nodes} non-Dirichlet nodes with the Dirichlet nodes placed at the end of the array. Why this exclusion?

Finite Element Method: step-2 CGFEM

Alternatively, derive this algebraic system:

- by substuting the finite element expansion for u_h into the minimisation principle,
- introducing the integrals, and
- taking variations with respect to $u_{j'}$ with test "function" $\delta u_{j'} = \eta_{j'}$.
- *Exercise:* Peform this derivation. Also see the FEM textbook.

Finite Element Method: step-3 CGFEM

Evaluate integrals and assembly:

- rewrite integrals in local reference coordinates.

Finite Element Method: step-3 CGFEM

We consider finite elements in two dimensions spanned by coordinates $\bar{x} = (x, y)^T$ with transpose $(\cdot)^T$. The domain $\Omega \subset \mathrm{R}^2$ is partitioned into a mixture of N_{el} quadrilateral and triangular elements such that we obtain the tesselation

$$\mathcal{T}_h = \{K_{\mathsf{k}} | \bigcup_{\mathsf{k}=1}^{N_{\mathrm{el}}} \bar{K}_{\mathsf{k}} = \bar{\Omega} \text{ and } K_{\mathsf{k}} \cap K_{k'} = \emptyset \text{ if } \mathsf{k} \neq \mathsf{k}', 1 \leq \mathsf{k}, \mathsf{k}' \leq N_{\mathrm{el}}\} \quad (62)$$

with $\bar{K}_{\mathsf{k}}$ the closure of element K_{k}. Each such element K_{k} has $N_{\mathrm{n}}^{\mathsf{k}} = N_{\mathsf{n}} = 3$ or 4 nodes, which are numbered locally in a counterclockwise fashion from 0 to $N_{\mathsf{n}} - 1$.

Finite Element Method: step-3 CGFEM

It is convenient to introduce a reference element $\hat{K}$ and define the mapping $F_K : \mathbb{R}^2 \to \mathbb{R}^2$ between the reference element $\hat{K}$ and element K_k as follows

$$\bar{x} = F_{K_\mathrm{k}}(\bar{\zeta}) = \sum_{\alpha=0}^{N_\mathrm{n}^\mathrm{k}-1} \bar{x}_{\mathrm{k},\alpha}\, \chi_\alpha(\bar{\zeta}) \tag{63}$$

with reference coordinate $\bar{\zeta} = (\zeta_1, \zeta_2)^T$, local node coordinates $\bar{x}_{\mathrm{k},\alpha}$ of element k, and shape functions

$$\chi_\alpha(\bar{x}) = \chi_\alpha(F_{K_\mathrm{k}}(\bar{\zeta})). \tag{64}$$

Finite Element Method: step-3 CGFEM, evaluation

The Galerkin basis functions $w_i(x, y)$ with global node number i are now defined on element K_k in a piecewise linear manner on the local reference element as:

$$w_\alpha(x, y) = \hat{w}_\alpha(F_K^{-1}(x, y)) = \chi_\alpha(\bar{\zeta})$$

with α the local element index on element K_k for which $w_\alpha = 1$ on global node i and zero at the other nodes. This choice of basis functions gives formally second-order accuracy. Other (higher-order) basis functions can be found in the literature (e.g., Brenner and Scott, and Bernsen et al.).

Finite Element Method: step-3 CGFEM, evaluation

The definition of the global matrix and vector components suggests the definition of the following $N_n^k \times N_n^k$ elemental matrix and $N_n^k \times 1$ vector

$$
\begin{aligned}
\hat{A}_{\alpha\beta} &= \int_K \nabla\chi_\alpha \cdot \nabla\chi_\beta \, \mathrm{d}\Omega \\
&= \int_{\hat{K}} \left((J^T)^{-1} \begin{pmatrix} \frac{\partial\chi_\alpha}{\partial\zeta_1} \\ \frac{\partial\chi_\alpha}{\partial\zeta_2} \end{pmatrix} \right) \cdot \left((J^T)^{-1} \begin{pmatrix} \frac{\partial\chi_\beta}{\partial\zeta_1} \\ \frac{\partial\chi_\beta}{\partial\zeta_2} \end{pmatrix} \right) |\det(J)| \, \mathrm{d}\overline{\zeta} \qquad (65) \\
\hat{b}_\alpha &= \int_K f \, w_\alpha \, \mathrm{d}\Omega \\
&= \int_{\hat{K}} f\big(x(\zeta_1,\zeta_2), y(\zeta_1,\zeta_2)\big)\chi_\alpha(\zeta_1,\zeta_2) |\det(J(\overline{\zeta})| \, \mathrm{d}\overline{\zeta} \qquad (66)
\end{aligned}
$$

for $\alpha, \beta = 0, \dots, N_n^k - 1$ on each reference element $\hat{K}$.

Finite Element Method: step-3 CGFEM, tris

Triangles have $N_{\mathrm{n}} = 3$ nodes $\bar{x}_{\mathrm{k},\alpha} = \bar{x}_\alpha$, corresponding to nodes $\bar{\zeta} = (0,0)^T, (1,0)^T, (0,1)^T$ in the reference element, and the shape functions are

$$\chi_0(\bar{\zeta}) = \zeta_0 = 1 - \zeta_1 - \zeta_2, \quad \chi_1(\bar{\zeta}) = \zeta_1, \quad \chi_2(\bar{\zeta}) = \zeta_2. \tag{67}$$

The three normal vectors for triangular elements are

$$\hat{n}_0 = \frac{1}{|\bar{x}_1 - \bar{x}_0|}(y_1 - y_0, x_0 - x_1)^T, \qquad \hat{n}_1 = \frac{1}{|\bar{x}_2 - \bar{x}_1|}(y_2 - y_1, x_1 - x_2)^T,$$
$$\hat{n}_2 = \frac{1}{|\bar{x}_0 - \bar{x}_2|}(y_0 - y_2, x_2 - x_0)^T. \tag{68}$$

NB, the vector: $\bar{x} = \bar{x}_0\chi_0(\bar{\zeta}) + \bar{x}_1\chi_1(\bar{\zeta}) + \bar{x}_2\chi_2(\bar{\zeta})$.

Finite Element Method: step-3 CGFEM, tris

The Jacobian $J_3 = J$ of the transformation between $\bar{x}$ and $\bar{\zeta}$ is defined via

$$J_3^T = \begin{pmatrix} \partial_{\zeta_1} x & \partial_{\zeta_1} y \\ \partial_{\zeta_2} x & \partial_{\zeta_2} y \end{pmatrix} = \begin{pmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{pmatrix}. \tag{69}$$

We can compute the inverse of the Jacobian using the following relations

$$\frac{\partial x}{\partial x} = \frac{\partial x}{\partial \zeta_1}\frac{\partial \zeta_1}{\partial x} + \frac{\partial x}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial x} = 1, \qquad \frac{\partial x}{\partial y} = \frac{\partial x}{\partial \zeta_1}\frac{\partial \zeta_1}{\partial y} + \frac{\partial x}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial y} = 0,$$

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial \xi_1}\frac{\partial \zeta_1}{\partial x} + \frac{\partial y}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial x} = 0 \qquad \frac{\partial y}{\partial y} = \frac{\partial y}{\partial \xi_1}\frac{\partial \zeta_1}{\partial y} + \frac{\partial y}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial y} = 1,$$

Finite Element Method: step-3 CGFEM, tris

or, in matrix notation

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \zeta_1} & \frac{\partial x}{\partial \zeta_2} \\ \frac{\partial y}{\partial \zeta_1} & \frac{\partial y}{\partial \zeta_2} \end{pmatrix} \begin{pmatrix} \frac{\partial \zeta_1}{\partial x} & \frac{\partial \zeta_1}{\partial y} \\ \frac{\partial \zeta_2}{\partial x} & \frac{\partial \zeta_2}{\partial y} \end{pmatrix} = JJ^{-1}.$$

Hence, gradients in $\bar{x}$ transform as follows

$$\begin{pmatrix} \partial_x V \\ \partial_y V \end{pmatrix} = (J^T)^{-1} \begin{pmatrix} \partial_{\zeta_1} V \\ \partial_{\zeta_2} V \end{pmatrix} = \frac{1}{\det J_3} \begin{pmatrix} y_2 - y_0 & y_0 - y_1 \\ x_0 - x_2 & x_1 - x_0 \end{pmatrix} \begin{pmatrix} \partial_{\zeta_1} V \\ \partial_{\zeta_2} V \end{pmatrix} \tag{70}$$

with determinant $\det J_3$ and $|J_3| = |\det J_3|$. Triangles have three faces or sides $S_0, \ldots, S_2$ spanned by node pairs $(\bar{x}_0, \bar{x}_1)$, $\ldots$, $(\bar{x}_2, \bar{x}_0)$. The faces $S_0, \ldots, S_2$ correspond to $\zeta_2 = 0, \zeta_1 + \zeta_2 = 1$ and $\zeta_1 = 0$ in the reference element, respectively.

Finite Element Method: step-3 CGFEM, quads

For quadrilaterals there are $N_{\mathrm{n}} = 4$ nodes $\bar{x}_{\mathsf{k},\alpha} = \bar{x}_\alpha$. In the reference element $\bar{\zeta} \in (-1,1)^2$, and we start counterclockwise with $\alpha = 0$ at node $(-1,-1)^T$ in the reference element. Furthermore, the shape functions are

$$\begin{aligned} \chi_0(\bar{\zeta}) &= (1-\zeta_1)(1-\zeta_2)/4, & \chi_1(\bar{\zeta}) &= (1+\zeta_1)(1-\zeta_2)/4, \\ \chi_2(\bar{\zeta}) &= (1+\zeta_1)(1+\zeta_2)/4, & \chi_3(\bar{\zeta}) &= (1-\zeta_1)(1+\zeta_2)/4. \end{aligned} \tag{71}$$

The four normal vectors for quadrilateral elements are

$$\begin{aligned} \hat{n}_0 &= \frac{1}{|\bar{x}_1 - \bar{x}_0|}\left(y_1 - y_0, x_0 - x_1\right)^T, & \hat{n}_1 &= \frac{1}{|\bar{x}_2 - \bar{x}_1|}\left(y_2 - y_1, x_1 - x_2\right)^T, \\ \hat{n}_2 &= \frac{1}{|\bar{x}_3 - \bar{x}_2|}\left(y_3 - y_2, x_2 - x_3\right)^T, & \hat{n}_3 &= \frac{1}{|\bar{x}_0 - \bar{x}_3|}\left(y_0 - y_3, x_3 - x_0\right)^T, \\ \bar{x} &= \bar{x}_0\chi_0(\bar{\zeta}) + \bar{x}_1\chi_1(\bar{\zeta}) + \bar{x}_2\chi_2(\bar{\zeta}) + \bar{x}_3\chi_3(\bar{\zeta}) \end{aligned} \tag{72}$$

Finite Element Method: step-3 CGFEM, quads

The Jacobian $J_4 = J$ of the transformation between $\bar{x}$ and $\bar{\zeta}$ is defined via

$$
\begin{aligned}
J_4^T &= J_4^T(\bar{\zeta}) = \begin{pmatrix} \partial_{\zeta_1} x & \partial_{\zeta_1} y \\ \partial_{\zeta_2} x & \partial_{\zeta_2} y \end{pmatrix} \\
&= \frac{1}{4} \begin{pmatrix} (1-\zeta_2)(x_1 - x_0) + (1+\zeta_2)(x_2 - x_3) & (1-\zeta_2)(y_1 - y_0) + (1+\zeta_2)(y_2 - y_3) \\ (1-\zeta_1)(x_3 - x_0) + (1+\zeta_1)(x_2 - x_1) & (1-\zeta_1)(y_3 - y_0) + (1+\zeta_1)(y_2 - y_1) \end{pmatrix}. \qquad (73)
\end{aligned}
$$

Finite Element Method: step-3 CGFEM, quads

Hence, gradients in $\bar{x}$ transform as follows

$$\begin{pmatrix} \partial_x V \\ \partial_y V \end{pmatrix} = (J^T)^{-1} \begin{pmatrix} \partial_{\zeta_1} V \\ \partial_{\zeta_2} V \end{pmatrix} = \frac{1}{\det J_4} \begin{pmatrix} \partial_{\zeta_2} y & -\partial_{\zeta_1} y \\ -\partial_{\zeta_2} x & \partial_{\zeta_1} x \end{pmatrix} \begin{pmatrix} \partial_{\zeta_1} V \\ \partial_{\zeta_2} V \end{pmatrix} \tag{74}$$

with the determinant $\det J_4$ and $|J_4| = |\det J_4|$. Quadrilaterals have four faces $S_0, \ldots, S_3$ spanned by node pairs $(\bar{x}_0, \bar{x}_1)$, ..., $(\bar{x}_3, \bar{x}_0)$. The sides $S_0, \ldots, S_3$ correspond to $\zeta_2 = -1, \zeta_1 = 1, \zeta_2 = 1$, and $\zeta_1 = -1$ in the reference element, respectively.

Finite Element Method: step-3 meshing

- Mesh: element, local and global node numbering.
- Mesh file.

Finite Element Method: step-3 CGFEM

Perform matrix and vector assembly:

- Given these elemental matrices and vectors we assemble the global matrix, A, and global vector, b, in the following *assembly algorithm*:
  set all components of A = 0 and b = 0 to zero: $A_{ij} = b_i = 0$
  for all elements K_k, $\mathrm{k} = 1, N_{\mathrm{el}}$, do
  . for $\alpha = 1, N_n^\mathrm{k}$ do
  . $i = Index(\mathrm{k}, \alpha)$
  . for $\beta = 1, N_n^\mathrm{k}$ do
  . $j = Index(\mathrm{k}, \beta)$
  . $A_{ij} = A_{ij} + \hat{A}_{\alpha\beta}$
  . $b_i = b_i + \hat{b}_\alpha$

Finite Element Method: step-4 CGFEM

Solve algebraic system:

- linear algebra solvers,
- time-stepping; and/or,
- iterative solvers for nonlinear algebraic systems.