Discontinuous Galerkin Schemes, Explicit/Implicit Time-stepping

Ammar H. Hakim (ammar@princeton.edu) [1]

[1] Princeton Plasma Physics Laboratory, Princeton, NJ

Princeton University, Course AST560, Spring 2021

What are discontinuous Galerkin schemes?

Discontinuous Galerkin schemes are a class of *Galerkin* schemes in which the solution is represented using *piecewise discontinuous* functions.

- *Galerkin* minimization
- Piecewise *discontinuous* representation

Weak-equality and recovery

- It is important to remember that the discontinuous Galerkin solution is a *representation* of the solution and not the solution itself.
- Notice that even a continuous function will, in general, have a discontinuous *representation* in DG.

We can formalize this idea using the concept of *weak-equality* by stating that DG only determines the solution to an *equivalence class of weakly-equal functions*.

Weak-equality and recovery

- Notice that weak-equality depends on the function space as well as the inner-product we selected.
- The Galerkin L_2 minimization is equivalent to, for example, restating that

$$f'(x,t) \doteq G[f]$$

  This implies

$$(\psi_k, f'(x,t) - G[f]) = 0$$

  which is exactly what we obtained by minimizing the error defined using the L_2 norm.
- Hence, we can say that the *DG scheme only determines the solution in the weak-sense*, that is, all functions that are weakly equal to DG representation can be potentially interpreted as the actual solution.
- This allows a powerful way to construct schemes with desirable properties by *recovering* weakly-equal functions using the DG representations.

Example of recovery: Exponential recovery in a cell

- Consider we have a linear representation of the particle distribution function $f_h(x) = f_0 + x f_1$ in a cell.
- We can use this to *reconstruct* an exponential function that has the desirable property that it is *positive* everywhere in the cell. That is, we want to find

$$\exp(g_0 + g_1 x) \doteq f_0 + x f_1$$

- This will lead to a coupled set of nonlinear equations to determine g_0 and g_1
- Note that this process is not always possible: we need $f_0 > 0$ as well as the condition $|f_1| \leq 3 f_0$. Otherwise, the f_h is not realizable (i.e. there is no positive distribution function with the same moments as f_h).

Example of recovery: Exponential recovery in a cell

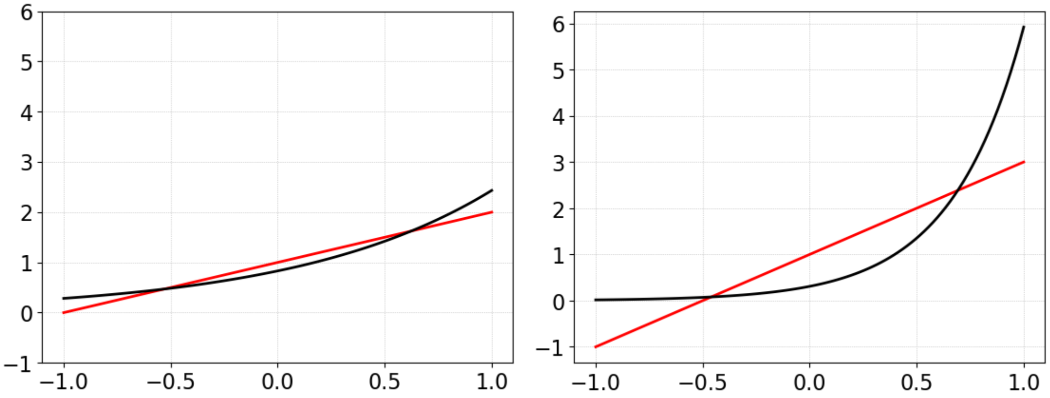


Figure: Recovery of exponential function (black) from linear function (red). Left plot is for $f_0 = 1$, $f_1 = 1$ and right for $f_0 = 1$ and $f_1 = 2$.

Discontinuous Galerkin scheme for linear advection

Consider the 1D passive advection equation on $I \in [L, R]$

$$\frac{\partial f}{\partial t} + \lambda \frac{\partial f}{\partial x} = 0$$

with λ the constant advection speed. $f(x, t) = f_0(x - \lambda t)$ is the exact solution, where $f_0(x)$ is the initial condition. Designing a good scheme is much harder than it looks.

- Discretize the domain into elements $I_j \in [x_{j-1/2}, x_{j+1/2}]$
- Pick a finite-dimensional function space to represent the solution. For DG we usually pick polynomials in each cell but allow discontinuities across cell boundaries
- Expand $f(x, t) \approx f_h(x, t) = \sum_k f_k(t) w_k(x)$.

Find the coefficients that minimize the L_2 norm of the residual

The discrete problem in DG is stated as: find f_h in the function space such that for each basis function φ we have

$$\int_{I_j} \varphi \left(\frac{\partial f_h}{\partial t} + \lambda \frac{\partial f_h}{\partial x} \right) \, dx = 0.$$

Integrating by parts leads to the discrete *weak-form*

$$\int_{I_j} \varphi \frac{\partial f_h}{\partial t} \, dx + \lambda \varphi_{j+1/2} \hat{F}_{j+1/2} - \lambda \varphi_{j-1/2} \hat{F}_{j-1/2} - \int_{I_j} \frac{d\varphi}{dx} \lambda f_h \, dx = 0.$$

Here $\hat{F} = \hat{F}(f_h^+, f_h^-)$ is the consistent *numerical flux* on the cell boundary. Integrals are performed using high-order quadrature schemes.

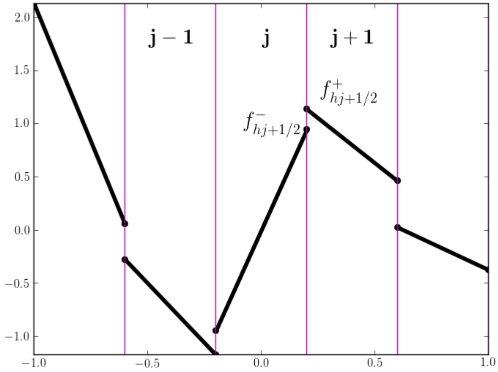
Need to select numerical flux

- Take averages (central fluxes)

$$\hat{F}(f_h^+, f_h^-) = \frac{1}{2}(f_h^+ + f_h^-)$$

- Use upwinding (upwind fluxes)

$$\begin{aligned}\hat{F}(f_h^+, f_h^-) &= f_h^- \quad \lambda > 0 \\ &= f_h^+ \quad \lambda < 0\end{aligned}$$



Example: Piecewise constant basis functions

- A central flux with piecewise constant basis functions leads to the familiar central difference scheme

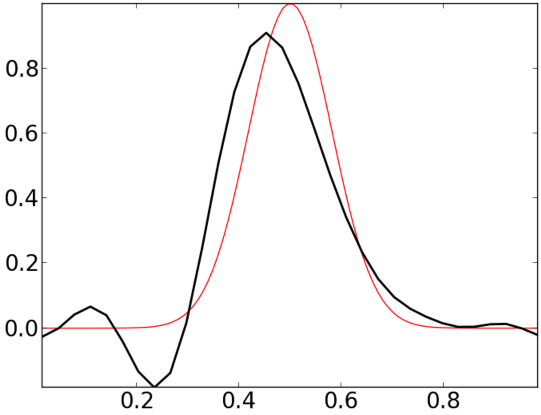
$$\frac{\partial f_j}{\partial t} + \lambda \frac{f_{j+1} - f_{j-1}}{2\Delta x} = 0$$

- An upwind flux with piecewise constant basis functions leads to the familiar upwind difference scheme (for $\lambda > 0$)

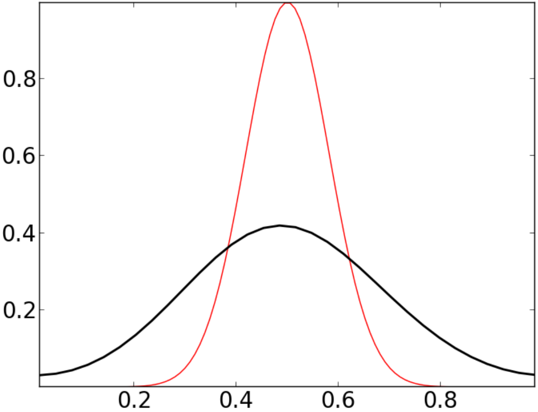
$$\frac{\partial f_j}{\partial t} + \lambda \frac{f_j - f_{j-1}}{\Delta x} = 0$$

Solution is advanced in time using a suitable ODE solver, usually strong-stability preserving Runge-Kutta methods. (See G2 website)

Example: Piecewise constant basis functions with central flux



Example: Piecewise constant basis functions with upwind flux



Passive advection with piecewise linear basis functions

To get better results, we can use piecewise linear polynomials instead. That is, select the basis functions

$$\varphi \in \{1, 2(x - x_j)/\Delta x\}$$

In terms of which the solution in each cell is expanded as $f_j(x,t) = f_{j,0} + 2f_{j,1}(x - x_j)/\Delta x$. With this, some algebra shows that we have the update formulas for *each stage* of a Runge-Kutta method

$$\begin{aligned} f_{j,0}^{n+1} &= f_{j,0}^{n} - \sigma\left(\hat{F}_{j+1/2} - \hat{F}_{j-1/2}\right) \\ f_{j,1}^{n+1} &= f_{j,1}^{n} - 3\sigma\left(\hat{F}_{j+1/2} + \hat{F}_{j-1/2}\right) + 6\sigma f_{j,0} \end{aligned}$$

where $\sigma \equiv \lambda\Delta t/\Delta x$. As these are explicit schemes we need to ensure time-step is sufficiently small. Usually, we need to ensure $\sigma = \lambda\Delta t/\Delta x \leq 1/(2p+1)$.

Passive advection with piecewise linear basis functions

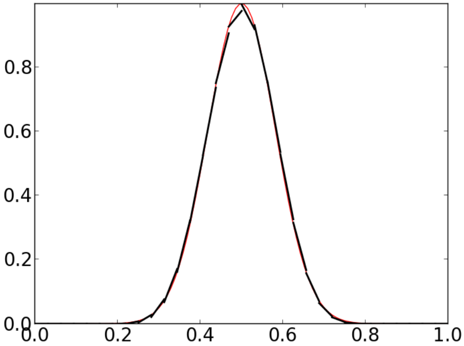


Figure: Advection equation solution (black) compared to exact solution (red) with upwind fluxes and piecewise linear basis functions.

Properties of the discrete equations

From the continuous passive advection equation we can show that, on a periodic domain the total particles are conserved

$$\frac{d}{dt}\int_I f\, dx = 0$$

Also, the L_2 norm of the solution is also conserved

$$\frac{d}{dt}\int_I \frac{1}{2} f^2\, dx = 0$$

We would like to know if our discrete scheme *inherits or mimics these properties*. Sometimes, methods in which the discrete scheme inhert important properties from the continuous equations are called *mimetic* methods. However, note that in general it is impossible to inhert *all* properties and often it is not desirable to do so.

To prove properties start from discrete weak-form

To understand properties of the scheme we must (obviously) use the *discrete weak-form* as the starting point.

$$\int_{I_j} \varphi \frac{\partial f_h}{\partial t}\, dx + \lambda \varphi_{j+1/2} \hat{F}_{j+1/2} - \lambda \varphi_{j-1/2} \hat{F}_{j-1/2} - \int_{I_j} \frac{d\varphi}{dx} \lambda f_h\, dx = 0.$$

A general technique is to use a function belonging to the *finite-dimensional function space* as the test function φ in the discrete weak-form.

Example: consider we set $\varphi = 1$. Then we get

$$\sum_j \int_{I_j} \frac{\partial f_h}{\partial t}\, dx + \lambda \sum_j \left(\hat{F}_{j+1/2} - \hat{F}_{j-1/2} \right) = 0.$$

The second term sums to zero and so we have shown that

$$\frac{d}{dt} \sum_j \int_{I_j} f_h\, dx = 0.$$

To prove properties start from discrete weak-form

Now, consider we use the *solution itself* as the test function. We can do this as the solution, by definition, belongs to the finite-dimensional function space. We get

$$\sum_j \int_{I_j} f_h \frac{\partial f_h}{\partial t}\, dx + \sum_j \left(f^-_{hj+1/2} \hat{F}_{j+1/2} - f^+_{hj-1/2} \hat{F}_{j-1/2} \right) - \sum_j \int_{I_j} \frac{df_h}{dx} f_h \, dx = 0$$

We can write the last term as

$$\sum_j \int_{I_j} \frac{1}{2} \frac{d}{dx} f_h^2 \, dx = \frac{1}{2} \sum_j \left[\left(f^-_{hj+1/2} \right)^2 - \left(f^+_{hj-1/2} \right)^2 \right]$$

If we use *upwind fluxes* we can show that we get

$$\frac{d}{dt} \sum_j \int_{I_j} f_h^2 \, dx = - \sum_j \left(f^-_{hj+1/2} - f^+_{hj-1/2} \right)^2 \leq 0.$$

Hence, the L_2 norm of the solution *will decay and not remain constant*. However, this is the desirable behavior as it ensures L_2 stability of the discrete system. With central fluxes the L_2 norm is conserved. (Prove this)

Summary of DG schemes for passive advection equation

- Pick basis functions. These are usually piecewise polynomials, but could be other suitable functions.
- Construct discrete weak-form using integration by parts.
- Pick suitable numerical fluxes for the surface integrals.
- Use Runge-Kutta (or other suitable) schemes for evolving the equations in time.
- To prove properties of the scheme, start from the discrete weak-form and use appropriate test-functions and simplify.

How to discretize parabolic equations with DG?

- DG is traditionally used to solve hyperbolic PDEs. However, DG is also very good for the solution of parabolic PDEs.
- One challenge here is that parabolic PDEs have *second* derivatives and it is not clear at first how a discontinuous representation can allow solving such systems.

Consider the diffusion equation (subscripts represent derivatives)

$$f_t = f_{xx}$$

Choose function space and multiply by test function in this space to get weak form

$$\int_{I_j} \varphi f_t \, dx = \varphi f_x \Big|_{x_{j-1/2}}^{x_{j+1/2}} - \int_{I_j} \varphi_x f_x \, dx.$$

In DG, as f is discontinuous, it is not clear how to compute the derivative across the discontinuity at the cell interface in the first term. (See SimJ JE16).

Use weak-equality to *recover* continuous function

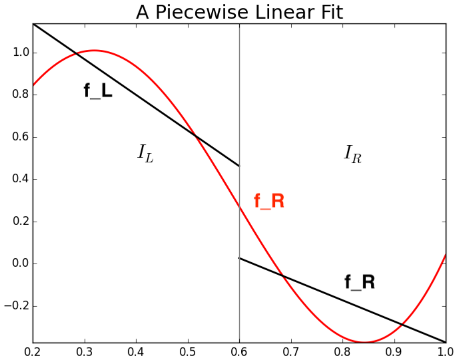


Figure: Given piecewise linear representation (black) we want to recover the continuous function (red) such that **moments of recovered and linear representation are the same in the respective cells.**

Use weak-equality to *recover* continuous function

- Consider recovering $\hat{f}$ on the interval $I = [-1, 1]$, from a function, f, which has a single discontinuity at $x = 0$.
- Choose some function spaces $\mathcal{P}_L$ and $\mathcal{P}_R$ on the interval $I_L = [-1, 0]$ and $I_R = [0, 1]$ respectively.
- Reconstruct a continuous function $\hat{f}$ such that

$$
\begin{aligned}
\hat{f} &\doteq f_L \quad x \in I_L \quad \text{on } \mathcal{P}_L \\
\hat{f} &\doteq f_R \quad x \in I_R \quad \text{on } \mathcal{P}_R.
\end{aligned}
$$

  where $f = f_L$ for $x \in I_L$ and $f = f_R$ for $x \in I_R$.
- To determine $\hat{f}$, use the fact that given $2N$ pieces of information, where N is the number of basis functions in $\mathcal{P}_{L,R}$, we can construct a polynomial of maximum order $2N - 1$. We can hence write

$$
\hat{f}(x) = \sum_{m=0}^{2N-1} \hat{f}_m x^m.
$$

  Plugging this into the weak-equality relations gives a *linear* system for $\hat{f}_m$.

Use recovered function in weak-form

Once we have determined $\hat{f}$ we can use this in the discrete weak-form of the diffusion equation:

$$\int_{I_j} \varphi f_t \, dx = \varphi \hat{f}_x \Big|_{x_{j-1/2}}^{x_{j+1/2}} - \int_{I_j} \varphi_x f_x \, dx.$$

Note that now as $\hat{f}$ is continuous at the cell interface there is no issue in computing its derivative. We can, in fact, do a second integration by parts to get another discrete weak-form

$$\int_{I_j} \varphi f_t \, dx = (\varphi \hat{f}_x - \varphi_x \hat{f}) \Big|_{x_{j-1/2}}^{x_{j+1/2}} + \int_{I_j} \varphi_{xx} f \, dx.$$

This weak-form has certain advantages as the second term does not contain derivatives (which may be discontinuous at cell boundary).

Putting everything together: the Vlasov-Maxwell equation

We would like to solve the Vlasov-Maxwell system, treating it as a partial-differential equation (PDE) in 6D:

$$\frac{\partial f_s}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{v} f_s) + \nabla_{\mathbf{v}} \cdot (\mathbf{F}_s f_s) = C[f_s]$$

where $\mathbf{F}_s = q_s/m_s(\mathbf{E} + \mathbf{v} \times \mathbf{B})$. The EM fields are determined from Maxwell equations

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0$$
$$\epsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{B} = -\mu_0 \mathbf{J}$$

Can we solve VM system *efficiently*, conserve invariants?

We know that the Vlasov-Maxwell system conserves, total number of particles; total (field + particle) momentum; total (field + particle) energy; other invariants. Can a numerical scheme be designed that retains (some or all) of these properties?

For understanding solar-wind turbulence and other problems, we would like a noise-free algorithm that allows studying phase-space cascades correctly, in a noise-free manner.

See Juno et. al JCP **353**, 110-147 (2018); Hakim et. al. JPP **86**, 905860403 (2020) for details.

Time-stepping schemes

In the past few lectures we only discussed how to discretize the spatial terms (FV, DG). How about time? Typically for *hyperbolic problems* we use explicit time-stepping schemes:

- Use a "one step" method in which a Taylor series in time is used to derive a *fully discrete* scheme.
- More common: use a special Runge-Kutta time-stepper specially designed for hyperbolic PDEs, called "Strong Stability Preseving Runge-Kutta" (SSP-RK). If single forward Euler step preserves monotonicity then so will the SSP-RK scheme.

Write the semi-discrete equation as the system of ODEs

$$\frac{df}{dt} = \mathcal{L}(f, t).$$

Note we can write any equation with first-order time-derivatives in this form. (Not just hyperbolic).

Strong Stability Preseving Runge-Kutta Schemes

Basic idea is to combine a series of *first-order forward Euler steps* to march the solution in time. Write forward Euler as

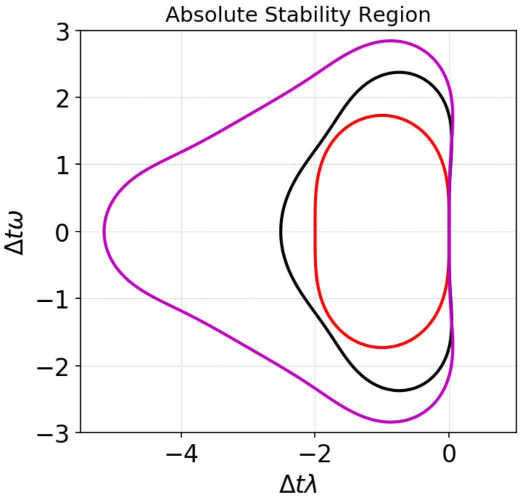
$$\mathcal{F}[f,t] = f + \Delta t \mathcal{L}[f,t]$$

Most common example is SSP-RK3 (third-order in time RK scheme).

$$\begin{aligned}
f^{(1)} &= \mathcal{F}\left[f^n, t^n\right] \\
f^{(2)} &= \frac{3}{4}f^n + \frac{1}{4}\mathcal{F}\left[f^{(1)}, t^n + \Delta t\right] \\
f^{n+1} &= \frac{1}{3}f^n + \frac{2}{3}\mathcal{F}\left[f^{(2)}, t^n + \Delta t/2\right]
\end{aligned}$$

Stability Regions of SSP-RK schemes

- Absolute stability regions for a equation $\dot{f} = (\lambda + i\omega) f$ for SSP-RK2 (red), SSP-RK3 (black) and four stage SSP-RK3 (magenta).
- Without diffusion ($\lambda = 0$) the SSP-RK2 scheme is mildly unstable as it has no intercept on the imaginary axis: the third order schemes should be preferred.
- Notice: intercept on negative real axis increases rapidly with number of stages; intercept on imaginary axis also increases: more stages can lead to schemes with bigger stability region. See David Ketcheson thesis.



Time-scales in a physical system

In typical plasmas the space and time-scales are enormous: plasma- and electron cyclotron-frequencies; light waves; sound waves, Alfven waves; (all MHD waves); resistive relaxation; transport scales. It's an orgy of scales! How to handle all these scales?

- One option: order out scales you do not care about by deriving asymptotic equations. Great example: extended MHD; gyrokinetics.
- However, these equations are still multi-scale! Worse, often there is no clean scale-separation in many interesting problems.

Time-scales in a model problem

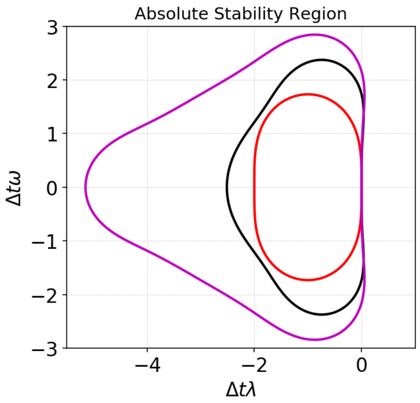
Consider advection-diffusion-reaction-oscillation equation

$$\frac{\partial f}{\partial t} + a\frac{\partial f}{\partial x} = \nu\frac{\partial^2 f}{\partial x^2} + i\Omega f - \gamma f$$

Here $\gamma \geq 0$ and Ω are real. Consider a single mode in space-time $e^{-i\omega t}e^{ikx}$ and get dispersion relation

$$\omega = \underbrace{(ak - \Omega)}_{\omega} - i\underbrace{(\nu k^2 + \gamma)}_{-\lambda}$$

For stability of explicit scheme we must choose $\omega\Delta t$ to lie inside the stability region of the time-stepping scheme.



Time-scales in a model problem

For finite-difference schemes $k_{\mathrm{max}} = 2/\Delta x$. Hence we have

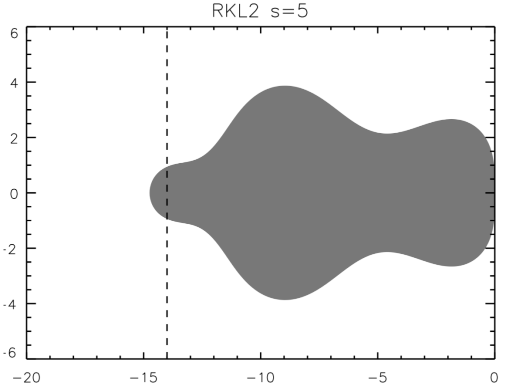
$$\omega = \left(\frac{2a}{\Delta x} - \Omega\right) - i\left(\frac{4\nu}{\Delta x^2} + \gamma\right)$$

Depending on the regime one or the other term may dominate. For example, Ω may be very large. Also, in particular, note that the damping from diffusion goes as $1/\Delta x^2$. This can be a serious limitation for explicit schemes.

- To overcome time-step limitation from Ω (oscillations) we need to use some sort of *time-centered implicit method*; For stiff $\gamma >> 1$ we need an damped implicit scheme.
- For diffusion dominated problems we can use implicit methods, or, preferably *super time-stepping schemes* (STS schemes).
- For advection dominated problems explicit schemes are best. Implicit schemes for *hyperbolic* equations are hard and do not always work well.

"Super-Time Stepping" Schemes

- "Super-Time Stepping" or Runge-Kutta-Legendre (or Runge-Kutta-Chebyshev or ROCK2) schemes work by taking large (10-100s) of RK stages to increase region of stability along negative real axis.
- For s stages the stability increases as s^2: hence, for large s we can get an approximate $s\times$ speed up compared to explicit scheme.
- Note that STS schemes *look* like explicit schemes! No need for complicated linear/nonlinear solvers.



Time-stepping a complex system of equations

To summarize: to update a complex system of nonlinear equations with hyperbolic, parabolic, oscillating and reaction terms:

- For advection terms typically use explicit schemes: implicit schemes are hard. Limited by fastest eigenvalue in the system.
- For oscillating terms use a time-centered implicit scheme (or backward implicit for fastest oscillations); for reactions use a backward implicit scheme;
- For diffusion (even nonlinear diffusion) use a STS scheme.

In a real problem all these need to be combined using *operator splitting* approaches.

How to solve elliptic equations?

Consider the Poisson equation

$$\nabla^2 f = -s$$

- This is an *elliptic equation*. The spatial derivatives can be computed in the same way we did for diffusion equation: integrate over a cell and use *symmetric recovery* to compute edge gradients. Ditto if using DG.
- Will lead to a (large) *linear system*. How to invert this system efficiently?
- Not an easy problem! In real applications matrices are sparse, and can be huge (millions or billions of unknowns). Often coupled to hyperbolic PDEs like collisionless Boltzmann equations: Poisson equation needs to be inverted at each step or even RK stage!

Consider a *direct* inversion (LU decomposition)

- For small problems, say O(100) unknowns, one can use standard LU decomposition: compute L and U once, store them and reuse. Can be very fast.
- However, LU decomposition scales like $O(N^3)$, where N is the number of unknowns.
- Consider a 3D problem on a cube with N_1 cells per direction. $N = N_1^3$. Hence, doubling the number of cells in each direction will increase cost by $(2^3)^3 = 512$!
- For large problems this cost is unacceptably high.

Many, many methods invented to solve this issue

- Instead of directly solving this system we can guess a solution and iteratively improve it: large class of iterative methods have been invented.
- Best methods are the class of *multi-grid* method. Huge literature on these. Not trivial to implement, best to use a library if possible.
- Sometimes simpler iterative methods also work well. Second order Richarson iteration is a good method to use. Belongs to the class of "Chebyshev iteration" schemes. Physical way to think about these schemes is to convert Poisson equation to a pseudo-time-dependent problem by adding "time" derivative terms. Then marrch to steady state.