Everybody’s Recipe for Making Good Diffusion Schemes

Hiroaki Nishikawa
National Institute of Aerospace
Common Approach

Integral form for diffusion:

\[ \int_{\Omega} u_t = \int_{\Omega} \nabla^2 u = \oint_{\partial \Omega} \nabla u \cdot n \, dA \]

Compute the interface gradient.

Bad diffusion schemes lack high-frequency damping:
Large errors, poor convergence, lose consistency…….

Good diffusion schemes have a damping term:
Finite-Volume: edge-term.
High-Order Schemes: penalty term.
Residual-Distribution: ?

Do we really know how to make a diffusion scheme?
Hyperbolic Model for Diffusion

\[ u_t = \nu u_{xx} \]

Diffusion Scheme

\[ u_t = \nu p_x \]
\[ p_t = \frac{(u_x - p)}{T_r} \]

\[ \lambda = \pm \sqrt{\frac{\nu}{T_r}} \]

Advection Scheme

Discretization
difficult

difficult

easy

Hyperbolic Model for Diffusion (Parabolic)
From Advection to Diffusion

Diffusion Equation (Parabolic)

\[ u_t = \nu u_{xx} \]

Hyperbolic Model for Diffusion

\[ u_t = \nu p_x \]
\[ p_t = (u_x - p)/T_r \]

Two models are equivalent if \( p = u_x \)

Derive a **diffusion** scheme from an **advection** scheme.

1. Discretize the hyperbolic system by an **advection** scheme:

\[ \frac{u_j^{n+1} - u_j^n}{\Delta t} = -Res_{j,1}^n, \quad \frac{p_j^{n+1} - p_j^n}{\Delta t} = -Res_{j,2}^n - \frac{1}{T_r} p_j^n. \]

2. Replace the second equation by a direct approximation of \( p = u_x \):

\[ \frac{u_j^{n+1} - u_j^n}{\Delta t} = -Res_{j,1}^n, \quad p_j^n = \text{least-squares gradient, for example} \]

The result is a time-accurate **diffusion** scheme. (no need to store extra variables)
Relaxation Time

The CFL condition:

$$\Delta t \leq \frac{\Delta x}{\sqrt{\nu/T_r}}$$

Keep the hyperbolic behavior over every time step:

$$\Delta t_{\text{max}} \equiv \frac{\Delta x}{\sqrt{\nu/T_r}} = \alpha T_r$$

Solve for $T_r$:

$$T_r = \frac{\Delta x^2}{\alpha^2 \nu}$$

The CFL condition becomes

$$\Delta t \leq \frac{1}{\alpha} \frac{\Delta x^2}{\nu}$$

General Form:

$$T_r = \frac{L_r^2}{\alpha^2 \nu}, \quad L_r = \frac{\Delta x}{D}$$

$$D = \begin{cases} 1 & \text{in 1D} \\ 2 & \text{in 2D} \\ 3 & \text{in 3D} \end{cases}$$
Recipe for Making Good Diffusion Schemes

\[ u_t = \nu u_{xx} \]

1. Discretize the hyperbolic system by an advection scheme.

\[ \mathbf{U}_t + \mathbf{F}_x = \mathbf{Q} \]

\[ \mathbf{U} = \begin{bmatrix} u \\ p \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -\nu p \\ -u/T_r \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 0 \\ -p/T_r \end{bmatrix}. \]

2. Ignore the discrete equation for \( p \), and approximate \( p = u_x \) directly.

3. The result is a time-accurate diffusion scheme, a good one!

Derived diffusion scheme:

1. Automatically equipped with a damping term (edge/penalty term)
2. Implemented in the same way as a corresponding advection scheme.

This recipe is applicable to various discretization methods.
1D Finite-Volume Scheme

1. Discretize the hyperbolic system:

Finite-Volume Method:
\[
\frac{dU_j}{dt} = -\frac{1}{\Delta x} \left[ F_{j+1/2} - F_{j-1/2} \right] + \frac{1}{\Delta x} \int_{L_j} Q \, dx
\]

Upwind Flux:
\[
F_{j+1/2} = \frac{1}{2} \left[ F_R + F_L \right] - \frac{\nu \alpha}{2\Delta x} (U_R - U_L)
\]

2. Discard the second component to get a diffusion scheme:

\[
\frac{du_j}{dt} = -\frac{1}{\Delta x} \left[ f_{j+1/2} - f_{j-1/2} \right]
\]

\[
f_{j+1/2} = -\frac{\nu}{2} \left[ p_R + p_L \right] - \frac{\nu \alpha}{2\Delta x} (u_R - u_L)
\]

The damping term comes from the dissipation of the advection scheme.
Effect of Damping Term

Fourier transformed: \( u_0 \exp(i \beta x / \Delta x) \)

\[
\frac{du_0}{dt} = -\frac{\nu}{\Delta x^2} \left( \sin^2 \beta + 2 \alpha \sin^4 \frac{\beta}{2} \right) u_0
\]

Consistent part Damping

Truncation Error:

\[
\frac{du_j}{dt} = \nu u_{xx} + \nu u_{xxxx} \left( \frac{1}{3} - \frac{\alpha}{8} \right) \Delta x^2 + O(\Delta x^4)
\]

The parameter \( \alpha \) controls damping: 4th-order accurate for \( \alpha = 8/3 \).
Two Dimensions

\[ u_t = \nu (u_{xx} + u_{yy}) \]

Hyperbolic Model:

\[ U_t + F_x + G_y = Q \]

\[
\begin{bmatrix}
    u \\
    p \\
    q
\end{bmatrix},
\begin{bmatrix}
    -\nu p \\
    -u/T_r \\
    0
\end{bmatrix},
\begin{bmatrix}
    -\nu q \\
    0 \\
    -u/T_r
\end{bmatrix},
\begin{bmatrix}
    0 \\
    -p/T_r \\
    -q/T_r
\end{bmatrix}.
\]

Equivalent to the diffusion equation when \( p = u_x, q = u_y \)

Absolute Jacobian:

\[
|A_n| = R_n \Lambda_n R_n^{-1} = \sqrt{\frac{\nu}{T_r}}
\begin{bmatrix}
    1 & 0 & 0 \\
    0 & n_x^2 & n_x n_y \\
    0 & n_x n_y & n_y^2
\end{bmatrix}.
\]

The first component is all we need.
Node-Centered FV Schemes

1. Discretize hyperbolic system: Edge-based advection scheme:

\[
\frac{d U_j}{dt} = -\frac{1}{V_j} \sum_{k \in \{K_j\}} \Phi_{jk} A_{jk} + \frac{1}{V_j} \int_{\Omega_j} Q \, dx \, dy
\]

\[
\Phi_{jk} = \frac{1}{2} \left[ H_{jk}(U_R) + H_{jk}(U_L) \right] - \frac{1}{2} |A_n| (U_R - U_L)
\]

2. Discard the second and third components to get a diffusion scheme:

\[
\frac{du_j}{dt} = -\frac{1}{V_j} \sum_{k \in \{K_j\}} \phi_{jk} A_{jk}
\]

Diffusive flux: \( \phi_{jk} = -\frac{\nu}{2} \left[ (p, q)_R + (p, q)_L \right] \cdot \hat{n}_{jk} - \frac{\nu \alpha}{2L_r} (u_R - u_L) \)

\[
= -\frac{\nu}{2} \left[ (\nabla u)_j + (\nabla u)_k \right] \cdot \hat{n}_{jk} - \frac{\nu \alpha}{2L_r} (u_R - u_L)
\]

Widely-used Avg-LSQ scheme can be reproduced by special choice of alpha.

The Green-Gauss scheme corresponds to using the Green-Gauss gradient for \((p,q)\).
Length Scale and Skewness

Derived diffusion scheme:

\[ \phi_{jk} = -\frac{\nu}{2} \left[ (\nabla u)_j + (\nabla u)_k \right] \cdot \hat{n}_{jk} - \frac{\nu \alpha}{2L_r} (u_R - u_L) \]

Length scale is defined as

\[ L_r = \frac{1}{2} |\Delta l_{jk} \cdot \hat{n}_{jk}| = \frac{1}{2} \Delta l_{jk} |\hat{e}_{jk} \cdot \hat{n}_{jk}| \]

Skewness measure

Damping is amplified for highly-skewed grids.

Widely-used Avg-LSQ scheme:

\[ \phi_{jk} = -\frac{\nu}{2} \left[ (\nabla u)_j + (\nabla u)_k \right] \cdot \hat{n}_{jk} - \left( \hat{e}_{jk} \cdot \hat{n}_{jk} \right) (u_R - u_L) \]

It loses damping for highly-skewed grids.

The derived diffusion scheme is very accurate and robust for highly-skewed grids.
Residual-Distribution Schemes

- **Lax-Wendroff Scheme:**

1. Discretize the hyperbolic system:

   \[
   \frac{dU_j}{dt} = \frac{1}{V_j} \sum_{T \in \{T_j\}} B_j^T \Phi^T,
   \]

   \[
   \Phi^T = \int_T (-A U_x - B U_y + Q) \, dxdy
   \]

   \[
   B_i^T = \frac{1}{3} I + \frac{h}{\sqrt{\nu / Tr}} (A, B) \cdot n_i
   \]

2. Discard the second and third components to get a diffusion scheme:

   \[
   \frac{du_j}{dt} = \frac{1}{V_j} \sum_{T \in \{T_j\}} \left[ \frac{1}{3} \phi^T - \frac{\nu \alpha}{2} \left\{ \nabla u^T - (\bar{p}_T, \bar{q}_T) \right\} \cdot n_j^T \right]
   \]

   **Damping term**

   This becomes the Galerkin scheme for \( \alpha = 2 \) (See Nishikawa JCP2007)

- **LDA Scheme (Upwind):** Very accurate scheme. Details in the paper.

We now have **good** diffusion schemes for residual-distribution method!
Discontinuous Galerkin Schemes

\[ u_j(x, y) = \bar{u}_j + \psi_1 \partial_x u_j + \psi_2 \partial_y u_j \]

Derived diffusion scheme:

\[
V_j \frac{d\bar{u}_j}{dt} = -\int_{\partial T_j} (f, g) \cdot \hat{n} dA
\]

\[
\begin{bmatrix}
\frac{d (\partial_x u_j)}{dt} \\
\frac{d (\partial_y u_j)}{dt}
\end{bmatrix} = M_j^{-1}
\begin{bmatrix}
-\int_{\partial T_j} \psi_1 (f, g) \cdot \hat{n} dA + \int_{T_j} \nabla \psi_1 \cdot (f, g) \\
-\int_{\partial T_j} \psi_2 (f, g) \cdot \hat{n} dA + \int_{T_j} \nabla \psi_2 \cdot (f, g)
\end{bmatrix}
\]

Volume Integral: Integration by parts with \( (f, g) = -\nu \nabla u \)

\[
\int_{T_j} \nabla \psi_1 \cdot (f, g) = -\nu \int_{T_j} u \nabla \psi_1 \cdot \hat{n}_k dA_k - \int_{T_j} u \nabla^2 \psi_1
\]

Diffusive flux derived from the upwind flux:

\[
\phi_k(x_{k_n}) = -\frac{\nu}{2} \left[ \nabla u_j + \nabla u_k \right] \cdot \hat{n}_k - \frac{\nu \alpha}{2 L_r} (u_R - u_L)
\]

Consistent part Damping term (skewness measure)

The DG diffusion scheme is nice and compact: involves only neighbors.
Spectral-volume (SV) scheme:

Polynomial reconstruction over a spectral volume:

\[ u_T(x, y) = \sum_{i \in \{C_T\}} u_i L_i(x, y) \]

\[ \frac{1}{V_i} \int_{C_i} u_T \, dV = u_i \]

Derived diffusion scheme:

\[ \frac{du_i}{dt} = -\frac{1}{V_i} \sum_{k \in \{K_i\}} \phi_{ik} A_k \]

\[ \phi_{jk} = -\frac{\nu}{2} \left[ \nabla u_T + \nabla u_{T_k} \right] \cdot \hat{n}_k - \frac{\nu \alpha}{2L_r} (u_R - u_L) \]

Consistent part \hspace{1cm} \text{Damping term (skewness measure)}

SV scheme is nice and simple; no volume integrals required.
Test Problem – Highly-Skewed Grids

Problem:

\[ u_t = \nu (u_{xx} + u_{yy}) \]
\[ u(x, y, 0) = 5 \sin(\pi x) \sin(4000\pi y) \]

Compute the solution at \( t = 1.0 \times 10^{-8} \)

Irregular Grids: \( 25 \times 25, 33 \times 33, \ldots, 129 \times 129, 137 \times 137 \) (15 grids)

Global time step (the forward-Euler explicit):

\[ \Delta t = 0.003 h^2 \]

Total time steps = 77, 137, 214, 308, 419, 547, 692, 854, 1033, 1229, 1443, 1673, 1921, 2185, 2467.

Compute the error at the data point at the final time.
Results for EBFV Schemes

Lack of damping $\rightarrow$ Large error, Oscillations
Results for RD Schemes

Lack of damping -> Large error and oscillations
Results for DG Schemes

Lack of damping -> Inconsistent, unstable, highly oscillatory
Results for SV Schemes

Lack of damping -> Inconsistent, unstable, highly oscillatory
Future Work

Three dimensions – *straightforward*

Advection-diffusion – *remarkably simple*

Higher-order schemes – *just try them*

Nonlinear systems – *2 strategies*

1. Interface gradient: \( \phi_k = -\nu \nabla u|_k \cdot \hat{n}_k \)

\[
\nabla u|_k = \frac{1}{2} \left[ \nabla u_j + \nabla u_k \right] + \frac{\alpha}{2L_r} (u_R - u_L) \hat{n}_k
\]

2. Follow the principle

Download papers at http://www.hiroakinishikawa.com