First, Second, and Third Order Finite-Volume Schemes for Diffusion

Hiro Nishikawa

National Institute of Aerospace CFD Seminar, December 18, 2012

Supported by ARO (PM: Dr. Frederick Ferguson), NASA, Software Cradle.

All Created Equal

“all men are created equal,”

- Thomas Jefferson, The Declaration of Independence

Are all PDEs created equal?

Don't Seem So

Hyperbolic (inviscid):

- Principle of "upwinding" (dissipation) led to many useful schemes.

- Robust 1st-order schemes - a home to come back.

- A variety of unstructured, high-order schemes.

Parabolic (viscous):

- Lack of universal guiding principles.

- Lack of robust 1st-order schemes (rely on inconsistent scheme).

- Much less variety for unstructured, high-order schemes.

- Degraded accuracy of derivatives on irregular grids.

They don't seem created equal...

Who Created PDEs?

"all men are created equal, that they are endowed by their Creator with certain unalienable Rights,"

- Thomas Jefferson, The Declaration of Independence

We created PDEs.

Then, we can recreate them equal.

Recreate Them Hyperbolic

First-Order Hyperbolic System Method

JCP2007, 2010, 2012, AIAA2009, 2011, 2013

$$\mathbf{U}_t + \mathbf{A}\mathbf{U}_x = \mathbf{B}\mathbf{U}_{xx} + \mathbf{C}\mathbf{U}_{xxx} + \cdots + \mathbf{S}$$

$$\Downarrow$$

$$\tilde{\mathbf{W}}_t + \tilde{\mathbf{A}}\tilde{\mathbf{W}}_x = 0$$

Methods for hyperbolic systems apply to all PDEs.
Dramatic simplification/improvements to numerical methods

NIA Sandwich Seminar, March 2012, Future CFD Conference, August 2012

Turn Every Food into a Burger

Simple, Efficient, Accurate.

Sushi Burger!

It looks eccentric, but the taste is the same, or even better.

Hyperbolic Diffusion System

Nishikawa, JCP2007

Sushi Burger for Diffusion

$$\partial_t u = \nu \left(\partial_{xx} u + \partial_{yy} u\right)$$

steady

$$\begin{array}{rcl} \partial_t u & = & \nu \left(\partial_x p + \partial_y q\right) \\ \partial_t p & = & (\partial_x u - p)/T_r \\ \partial_t q & = & (\partial_y u - q)/T_r \end{array} \quad \rightarrow \quad \begin{array}{rcl} 0 & = & \nu \left(\partial_x p + \partial_y q\right), \\ p & = & \partial_x u, \\ q & = & \partial_y u, \end{array} \quad \rightarrow \quad 0 = \nu \left(\partial_{xx} u + \partial_{yy} u\right),$$

First-order system is equivalent to the diffusion equation in the steady state for *any relaxation time Tr* (not stiff at all):

$$T_r = \frac{L_r^2}{\nu}, \quad L_r = \frac{1}{2\pi}$$

Unsteady computation possible by dual-time stepping (implicit).

Nishikawa, JCP2007

Hyperbolic Diffusion System

Sushi Burger for Diffusion

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} + \partial_y \mathbf{G} = \mathbf{S},$$

$$\mathbf{U} = \begin{bmatrix} u \\ p \\ q \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -\nu p \\ -u/T_r \\ 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} -\nu q \\ 0 \\ -u/T_r \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 \\ -p/T_r \\ -q/T_r \end{bmatrix}$$

Normal Jacobian: $\mathbf{A}_n = \dfrac{\partial \mathbf{H}}{\partial \mathbf{U}} = \dfrac{\partial(\mathbf{F} n_x + \mathbf{G} n_y)}{\partial \mathbf{U}}$

Real eigenvalues: $\lambda_1 = -\sqrt{\dfrac{\nu}{T_r}}, \quad \lambda_2 = \sqrt{\dfrac{\nu}{T_r}}, \quad \lambda_3 = 0$

Methods for hyperbolic systems directly apply to diffusion.

Energy Estimate

Integrate over the domain $\ell^E = (u,\, L_r^2 p,\, L_r^2 q)$ x (hyperbolic system):

steady

$$\frac{d}{dt}\int_\Omega E\,dV = -\nu\int_\Omega (p^2+q^2)\,dV - \oint_{\partial\Omega} \mathbf{f}^E\cdot\mathbf{n}\,dA$$

$E = u^2 + L_r^2(p^2+q^2)$

$\mathbf{f}^E = (-\nu u p, -\nu u q)$

which reduces to the energy estimate for the Laplace equation:

$$0 = \int_\Omega \nabla u\cdot\nabla u\,dV - \oint_{\partial\Omega} u\frac{\partial u}{\partial n}dA$$

Energy estimate consistent with steady diffusion problem

Diffusion is Hyperbolic

If you have a good inviscid scheme,
you have a very good viscous scheme.

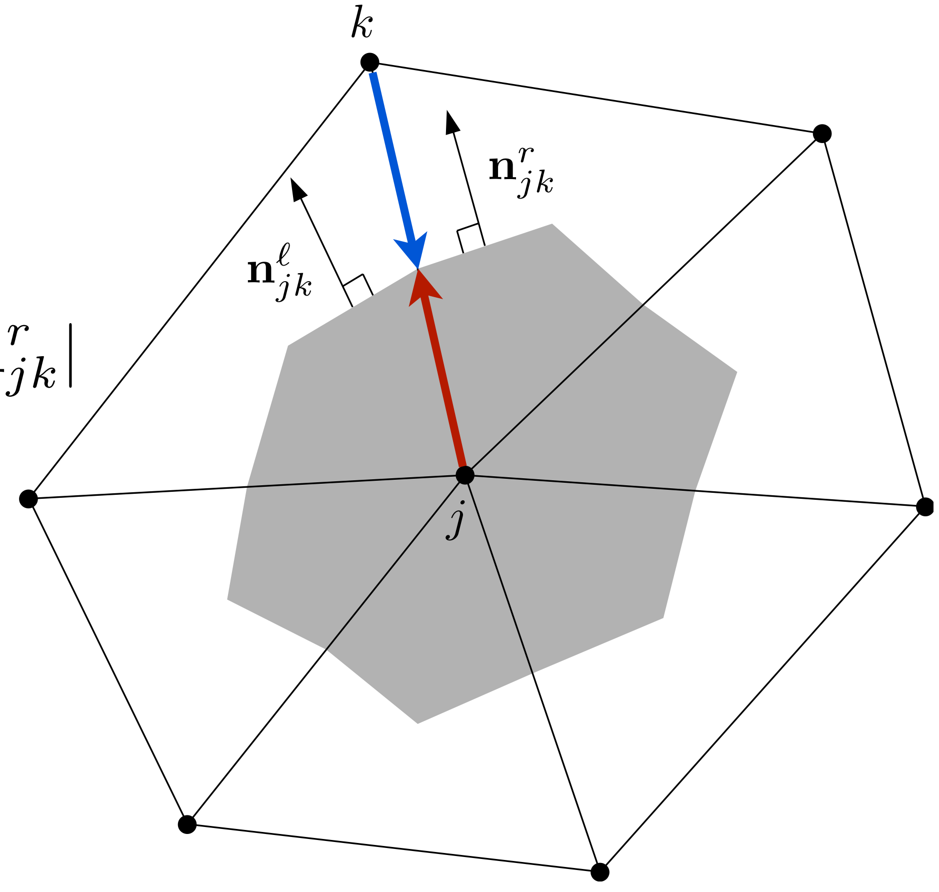
Edge-Based Finite-Volume Method

NASA's FUN3D; Software Cradle's SC/Tetra; DLR Tau code, etc.

Edge-based finite-volume scheme:

$$V_j \frac{d\mathbf{U}_j}{dt} = -\sum_{k \in \{k_j\}} \Phi_{jk} A_{jk} + \mathbf{S}_j V_j$$

$$A_{jk} = |\mathbf{n}_{jk}^{\ell} + \mathbf{n}_{jk}^{r}|$$



with the upwind flux at edge midpoint:

$$\Phi_{jk} = \frac{1}{2}(\mathbf{H}_L + \mathbf{H}_R) - \frac{1}{2}|\mathbf{A}_n|(\mathbf{U}_R - \mathbf{U}_L)$$

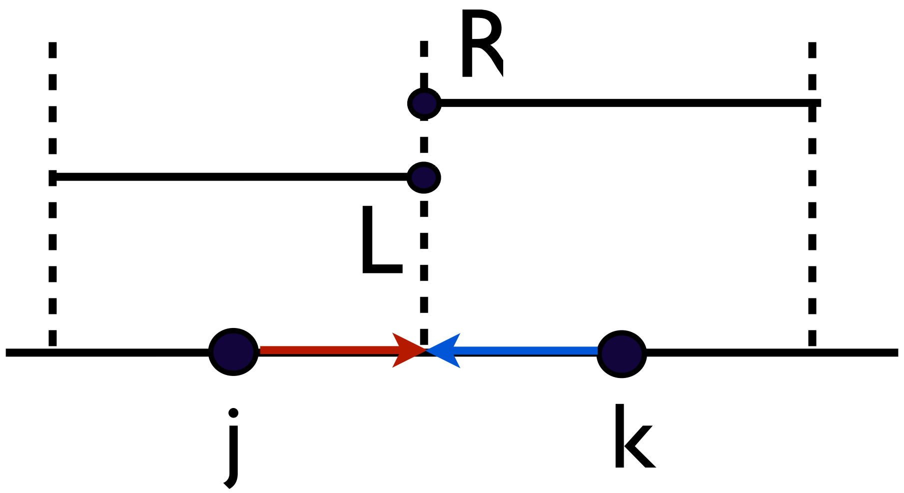
Accuracy is determined by the left and right states.

First-Order Scheme

Home sweet home for diffusion schemes

First-Order Scheme for Diffusion

Left and right states:



$$\mathbf{U}_L = \mathbf{U}_j, \quad \mathbf{U}_R = \mathbf{U}_k$$

Discrete Energy Estimate:

$$\sum_{j\in\{j\}} V_j \frac{dE_j}{dt} = -\sum_{e_b\in\{e_b\}} \frac{\mathbf{f}_1^E + \mathbf{f}_2^E}{2} \hat{\mathbf{n}}_{e_b} A_{e_b} - \sum_{j\in\{j\}} \nu(p_j^2 + q_j^2)V_j - \frac{\nu}{2L_r} \sum_{e\in\{e\}} \epsilon_{jk} A_{jk}$$

Consistent with continuous energy estimate *Dissipation*

$$\epsilon_{jk} = (u_k - u_j)^2 + L_r^2 \left[(p_k - p_j, q_k - q_j) \cdot \hat{\mathbf{n}}_{jk}\right]^2 \geq 0$$

First-order upwind diffusion scheme is energy-stable for general grids.

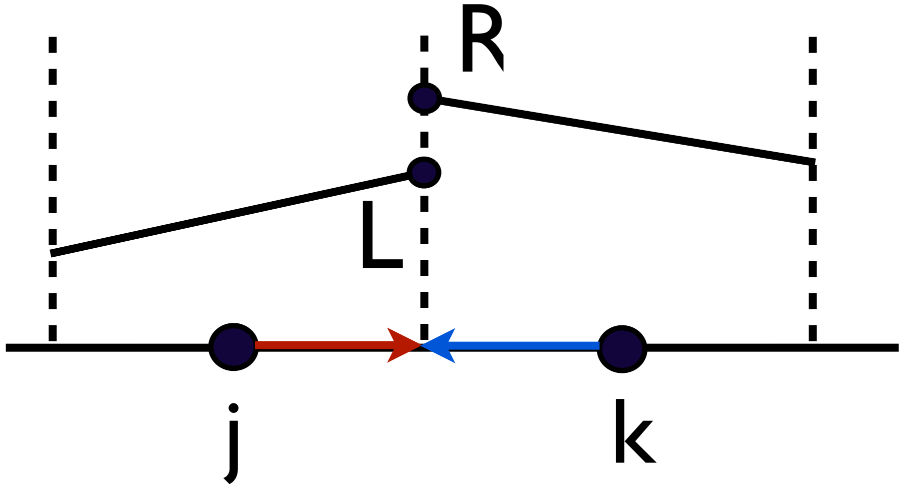
Second-Order Scheme

Even better

Second-Order Scheme

For triangular/tetrahedral and smooth mixed grids.

1. Compute gradients at nodes (e.g., LSQ).
2. Extrapolate the solution to the midpoint.



Left and right states:

$$u_L = u_j + \frac{1}{2}\boxed{(p_j, q_j)} \cdot \Delta \mathbf{l}_{jk}, \quad u_R = u_k - \frac{1}{2}\boxed{(p_k, q_k)} \cdot \Delta \mathbf{l}_{jk}$$

$$p_L = p_j + \frac{1}{2}\nabla p_j \cdot \Delta \mathbf{l}_{jk}, \quad p_R = p_k - \frac{1}{2}\nabla p_k \cdot \Delta \mathbf{l}_{jk}$$

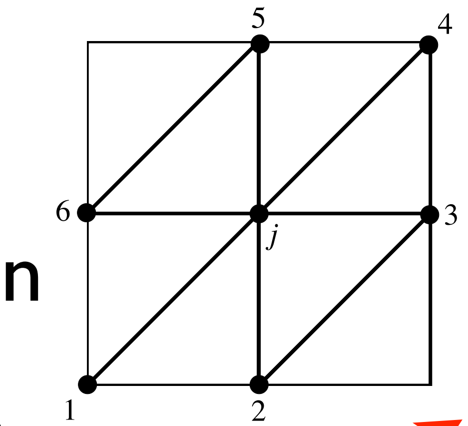
$$q_L = q_j + \frac{1}{2}\nabla q_j \cdot \Delta \mathbf{l}_{jk}, \quad q_R = q_k - \frac{1}{2}\nabla q_k \cdot \Delta \mathbf{l}_{jk}$$

$$\Delta \mathbf{l}_{jk} = (x_k - x_j, y_k - y_j)$$

Gradients are not needed for the solution.

Taylor Expansion

Source discretization



steady

$$\frac{du_j}{dt} = \nu(\partial_x p + \partial_y q)$$

$$- \frac{\nu h}{6L_r}\left[(\sqrt{2}+\sqrt{5})\partial_x(p-\partial_x u) + \sqrt{2}\partial_y(p-\partial_x u) + \sqrt{2}\partial_x(q-\partial_y u) + (\sqrt{2}+\sqrt{5})\partial_y(q-\partial_y u)\right]$$

$$- \frac{\nu h^2}{12}\left[\partial_{xx}(\partial_x p + \partial_y q) + \partial_{xy}(\partial_x p + \partial_y q) + \partial_{yy}(\partial_x p + \partial_y q)\right] + O(h^3),$$

steady

$$\frac{dp_j}{dt} = \frac{1}{T_r}(\partial_x u - p) - \frac{h^2}{6T_r}\left[\partial_{xx}(p-\partial_x u) + \partial_{xy}(p-\partial_x u) + \partial_{xy}(q-\partial_y u) + \frac{1}{2}(\partial_{xx}p + \partial_{yy}p + \partial_{xx}q)\right] + O(h^3),$$

steady

$$\frac{dq_j}{dt} = \frac{1}{T_r}(\partial_y u - q) - \frac{h^2}{6T_r}\left[\partial_{yy}(q-\partial_y u) + \partial_{xy}(q-\partial_y u) + \partial_{xy}(p-\partial_x u) + \frac{1}{2}(\partial_{xx}q + \partial_{yy}q + \partial_{xx}p)\right] + O(h^3)$$

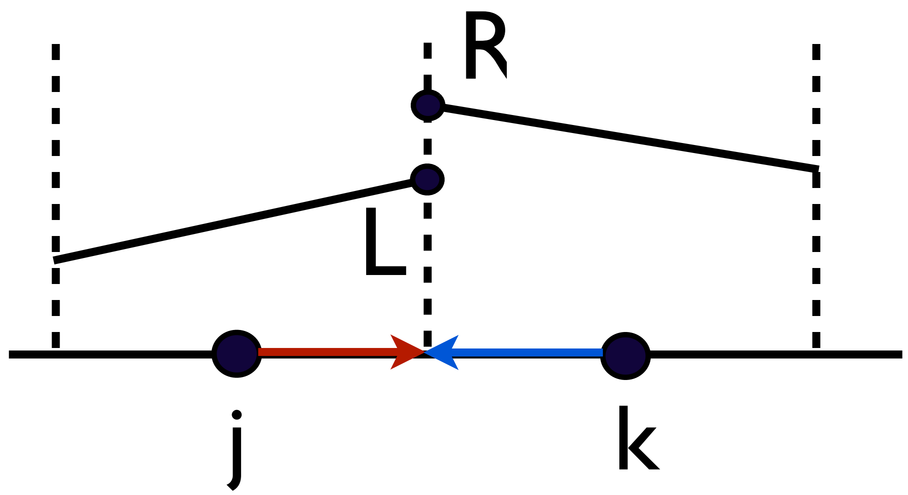
Second-order accurate for solution and gradients.
First-order error makes it stable with forward Euler.

Third-Order Scheme

A new wave

Third-Order Scheme (Katz and Sankaran JCP2011)

For triangular/tetrahedral grids only.



1. 2nd-order gradients at nodes (e.g., LSQ quadratic fit).

2. Extrapolate flux/solution to the midpoint.

$$\mathbf{H}_L = \mathbf{H}_j + \frac{1}{2}\nabla\mathbf{H}_j \cdot \Delta\mathbf{l}_{jk}, \quad \mathbf{H}_R = \mathbf{H}_k - \frac{1}{2}\nabla\mathbf{H}_k \cdot \Delta\mathbf{l}_{jk}$$

Third-order accuracy on a second-order stencil

Source term needs special discretization (NIA CFD Seminar 12-04-12).

Divergence Form of Source

Write the source term at each node j as follows:

$$\mathbf{S} \longrightarrow \partial_x \mathbf{F}^s + \partial_y \mathbf{G}^s$$

$$\mathbf{F}^s = \begin{bmatrix} 0 \\ (y - y_j)\, q/T_r \\ -(x - x_j)\, q/T_r \end{bmatrix}, \quad \mathbf{G}^s = \begin{bmatrix} 0 \\ -(y - y_j)\, p/T_r \\ (x - x_j)\, p/T_r \end{bmatrix}$$

So that

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} + \partial_y \mathbf{G} = \mathbf{S}$$

$$\downarrow$$

$$\partial_t \mathbf{U} + \partial_x (\mathbf{F} - \mathbf{F}^s) + \partial_y (\mathbf{G} - \mathbf{G}^s) = \mathbf{0}$$

Source term discretization is not needed.

Fully Hyperbolic Diffusion System

Ultimate Sushi Burger for Diffusion

The new system

$$\partial_t \mathbf{U} + \partial_x (\mathbf{F} - \mathbf{F}^s) + \partial_y (\mathbf{G} - \mathbf{G}^s) = \mathbf{0}$$

has the following eigenvalues:

$$\lambda_1 = -\sqrt{\frac{\nu}{T_r}}, \quad \lambda_2 = \sqrt{\frac{\nu}{T_r}}, \quad \lambda_3 = \frac{(x - x_j)n_x + (y - y_j)n_y}{T_r}.$$

Third eigenvalue is no longer zero.
A new wave created by the fully hyperbolic formulation.

Equivalence to the Original System

Fully hyperbolic diffusion system is equivalent to the following:

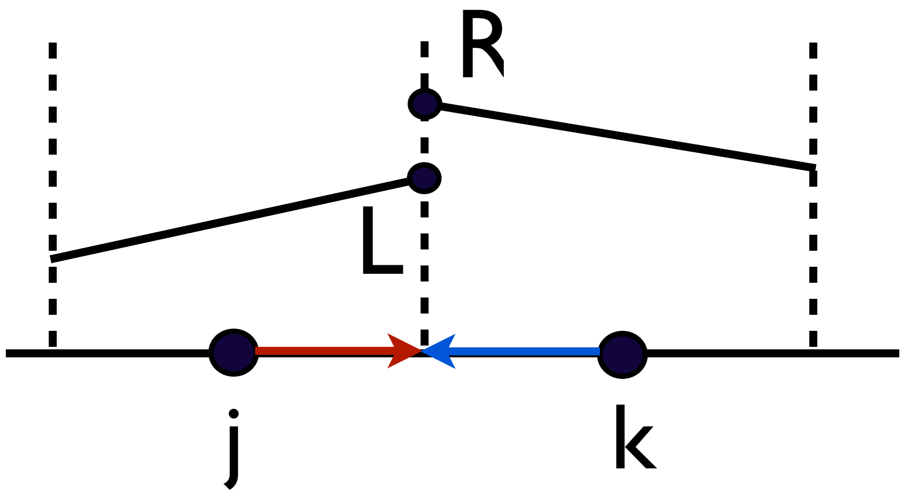
$$\begin{aligned} \partial_t u &= \nu\,(\partial_x p + \partial_y q), \\ T_r \partial_t p &= (\partial_x u - p) + \underline{(y - y_j)\,(\partial_x q - \partial_y p)}, \\ T_r \partial_t q &= (\partial_y u - q) - \underline{(x - x_j)\,(\partial_x q - \partial_y p)}. \end{aligned}$$

Equivalent to the original system precisely at node j.

Consistency constraint $\partial_x q - \partial_y p$ *vanishes in the steady state.*

Third-Order Scheme (Katz and Sankaran JCP2011)

1. 2nd-order gradients at nodes (e.g., LSQ).
2. Extrapolate flux/solution to the midpoint.
3. Upwind flux for fully hyperbolic system.



Left and right states:

$$u_L = u_j + \frac{1}{2}\boxed{(p_j, q_j)} \cdot \Delta\mathbf{l}_{jk}, \quad u_R = u_k - \frac{1}{2}\boxed{(p_k, q_k)} \cdot \Delta\mathbf{l}_{jk}$$

$$p_L = p_j + \frac{1}{2}\nabla p_j \cdot \Delta\mathbf{l}_{jk}, \quad p_R = p_k - \frac{1}{2}\nabla p_k \cdot \Delta\mathbf{l}_{jk}$$

$$q_L = q_j + \frac{1}{2}\nabla q_j \cdot \Delta\mathbf{l}_{jk}, \quad q_R = q_k - \frac{1}{2}\nabla q_k \cdot \Delta\mathbf{l}_{jk}$$

Gradients are not needed for the solution.

Taylor Expansion

steady

$$\frac{du_j}{dt} = \nu(\partial_x p + \partial_y q)$$
$$- \frac{\nu h}{6L_r}\left[(\sqrt{2}+\sqrt{5})\partial_x(p - \partial_x u) + \sqrt{2}\partial_y(p - \partial_x u) + \sqrt{2}\partial_x(q - \partial_y u) + (\sqrt{2}+\sqrt{5})\partial_y(q - \partial_y u)\right]$$
$$- \frac{\nu h^2}{12}\left[\partial_{xx}(\partial_x p + \partial_y q) + \partial_{xy}(\partial_x p + \partial_y q) + \partial_{yy}(\partial_x p + \partial_y q)\right] + O(h^3),$$

steady

$$\frac{dp_j}{dt} = \frac{1}{T_r}(\partial_x u - p) - \frac{h^2}{6T_r}\left[(\partial_{xx} + \partial_{xy})(q - \partial_y u) + \partial_{xx}(p - \partial_x u) + \partial_y(\partial_x q - \partial_y p)\right] + O(h^3),$$

steady

$$\frac{dq_j}{dt} = \frac{1}{T_r}(\partial_y u - q) - \frac{h^2}{6T_r}\left[(\partial_{xy} + \partial_{yy})(p - \partial_x u) + \partial_{yy}(q - \partial_y u) - \partial_x(\partial_x q - \partial_y p)\right] + O(h^3)$$

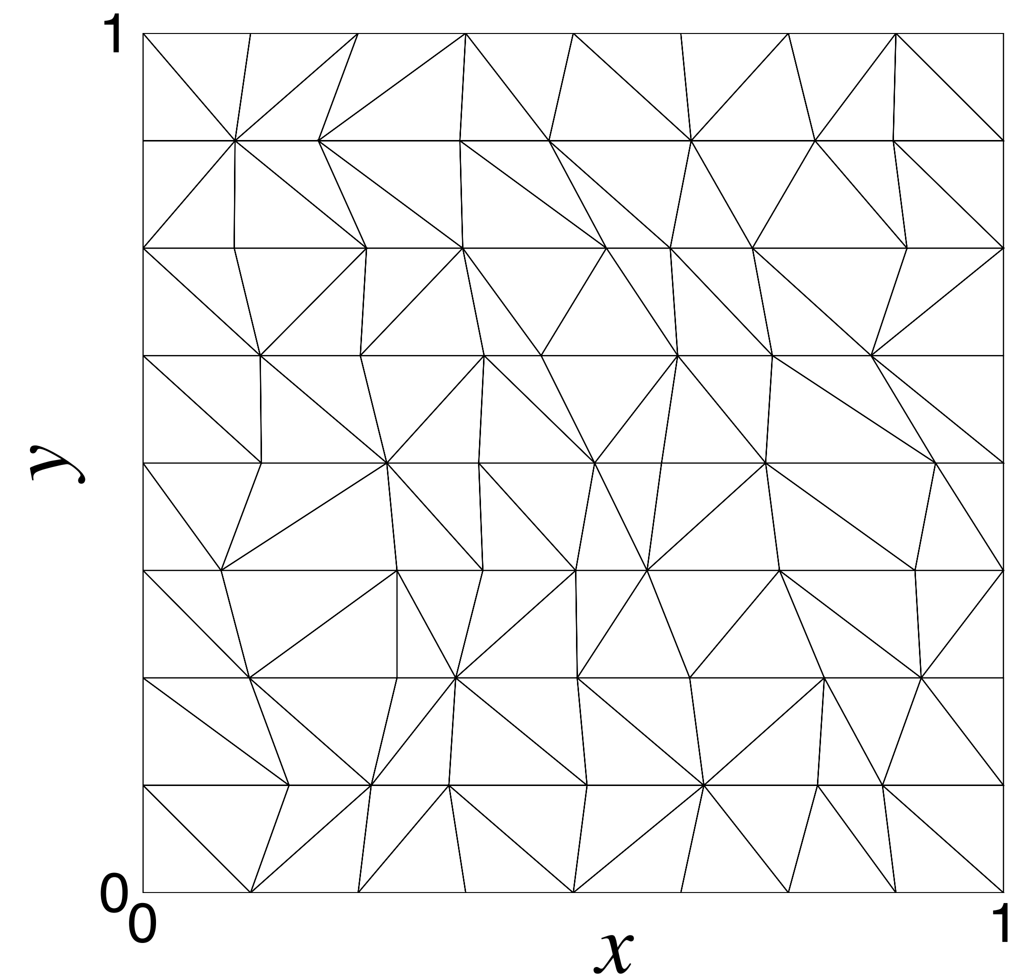
Third-order accurate for solution and gradients.

First-order error makes it stable with forward Euler.

Numerical Experiment

Exact solution:

$$u(x, y) = \frac{\sinh(\pi x)\sin(\pi y) + \sinh(\pi y)\sin(\pi x)}{\sinh(\pi)}$$



- n x n grids: n = 9, 17, 33, 65, 129, 257.

- Dirichlet boundary condition.

- 10 neighbors for quadratic fit.
  (to avoid ill-conditioning of LSQ matrix)

- Forward Euler time stepping

- Steady state reached when residual drops below 1.0E-15

- Comparison with the Galerkin scheme

Max CFL Number

Define the time step by

$$\Delta t_j = \mathrm{CFL} \frac{2V_j}{\sum_{k \in \{k_j\}} (\nu / L_r A_{jk} + V_j / T_r)} = O(h) \gg O(h^2)$$

O(h) time step is typical for hyperbolic systems.

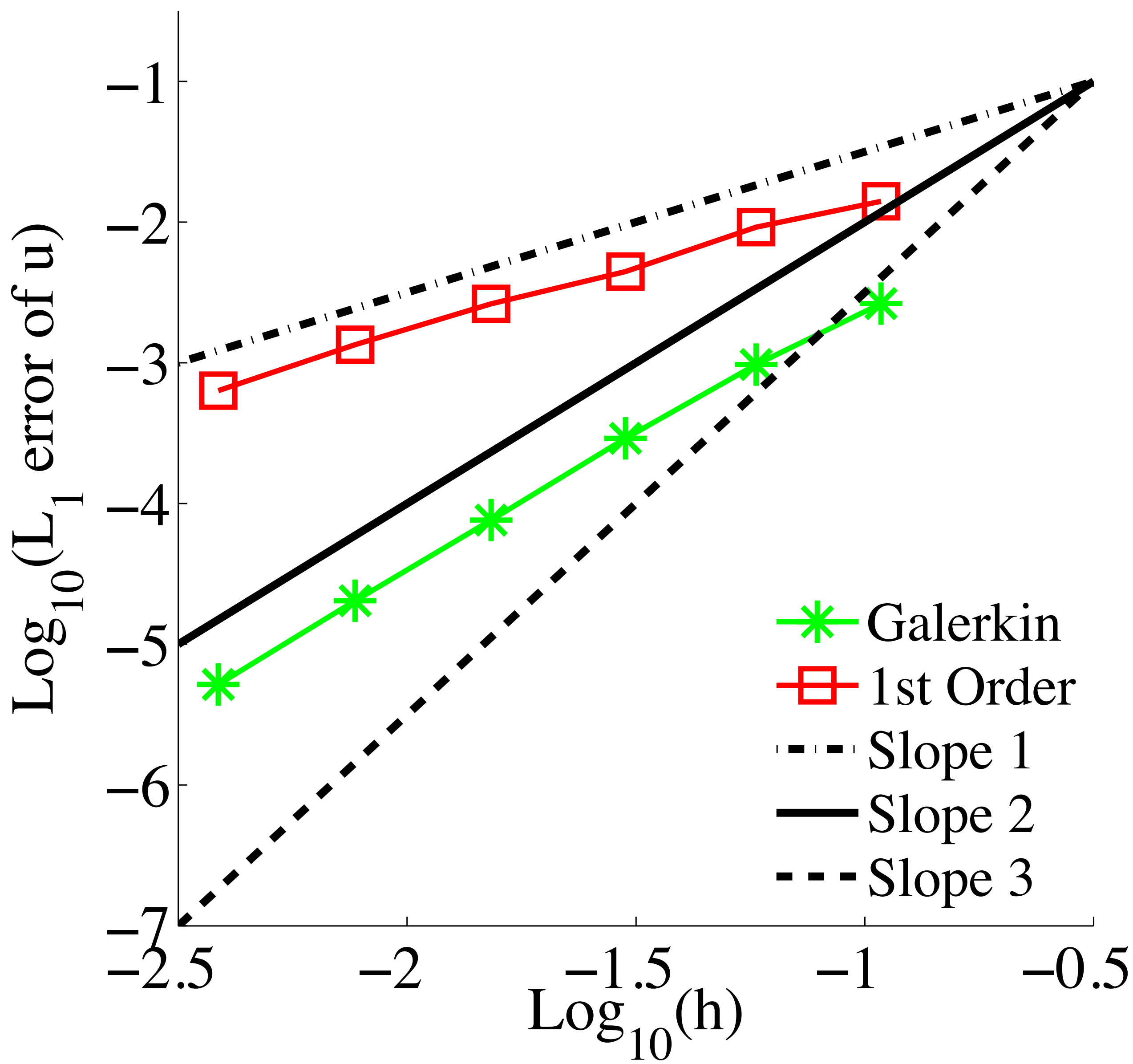
Max CFL numbers determined by Fourier analysis:

	First-Order	Second-Order	Third-Order
Forward Euler	1.3032	0.7313	0.7313

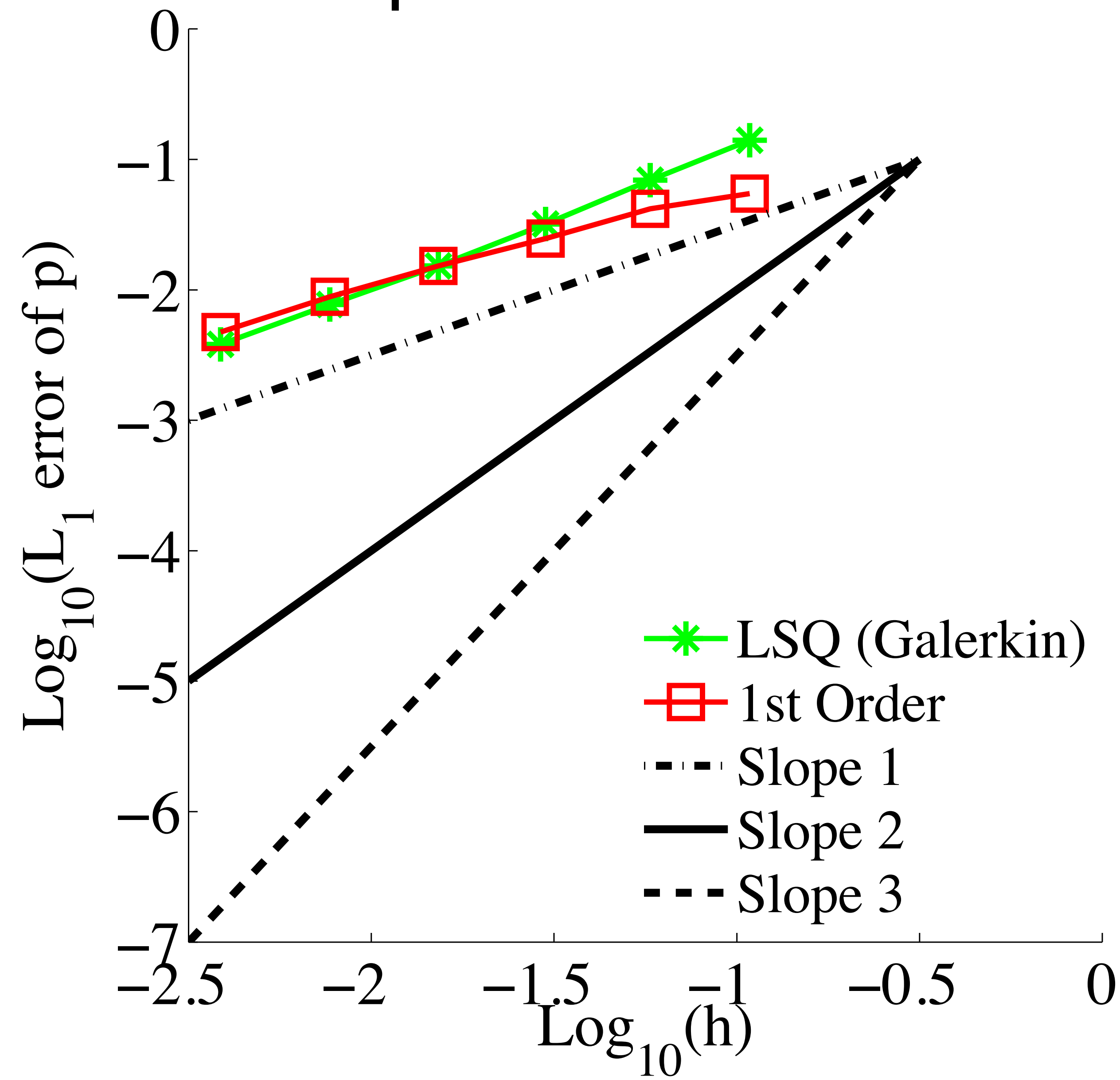
Hyperbolic schemes allow large O(h) time step.

Error Convergence 1

u: solution



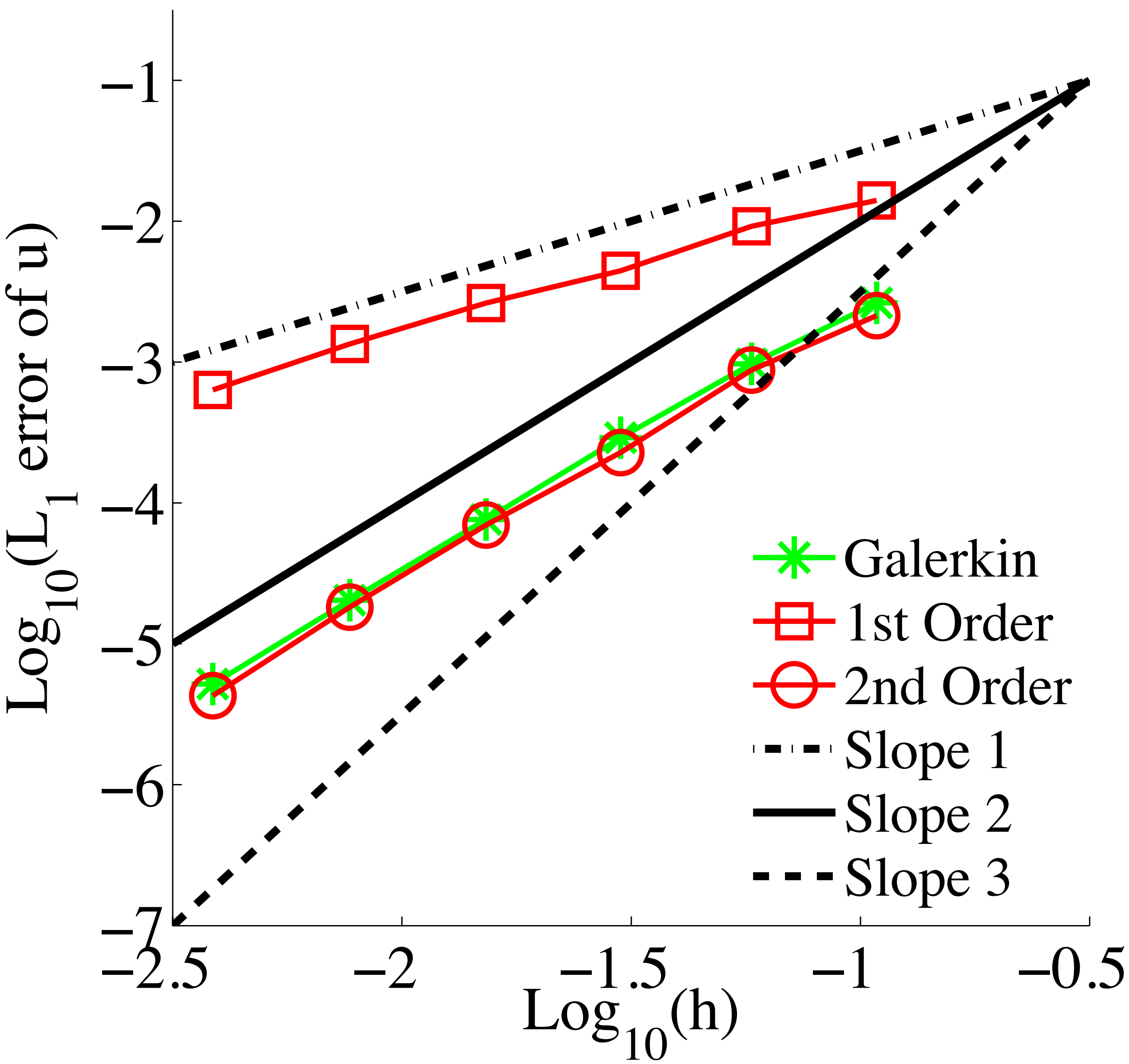
p: x-derivative



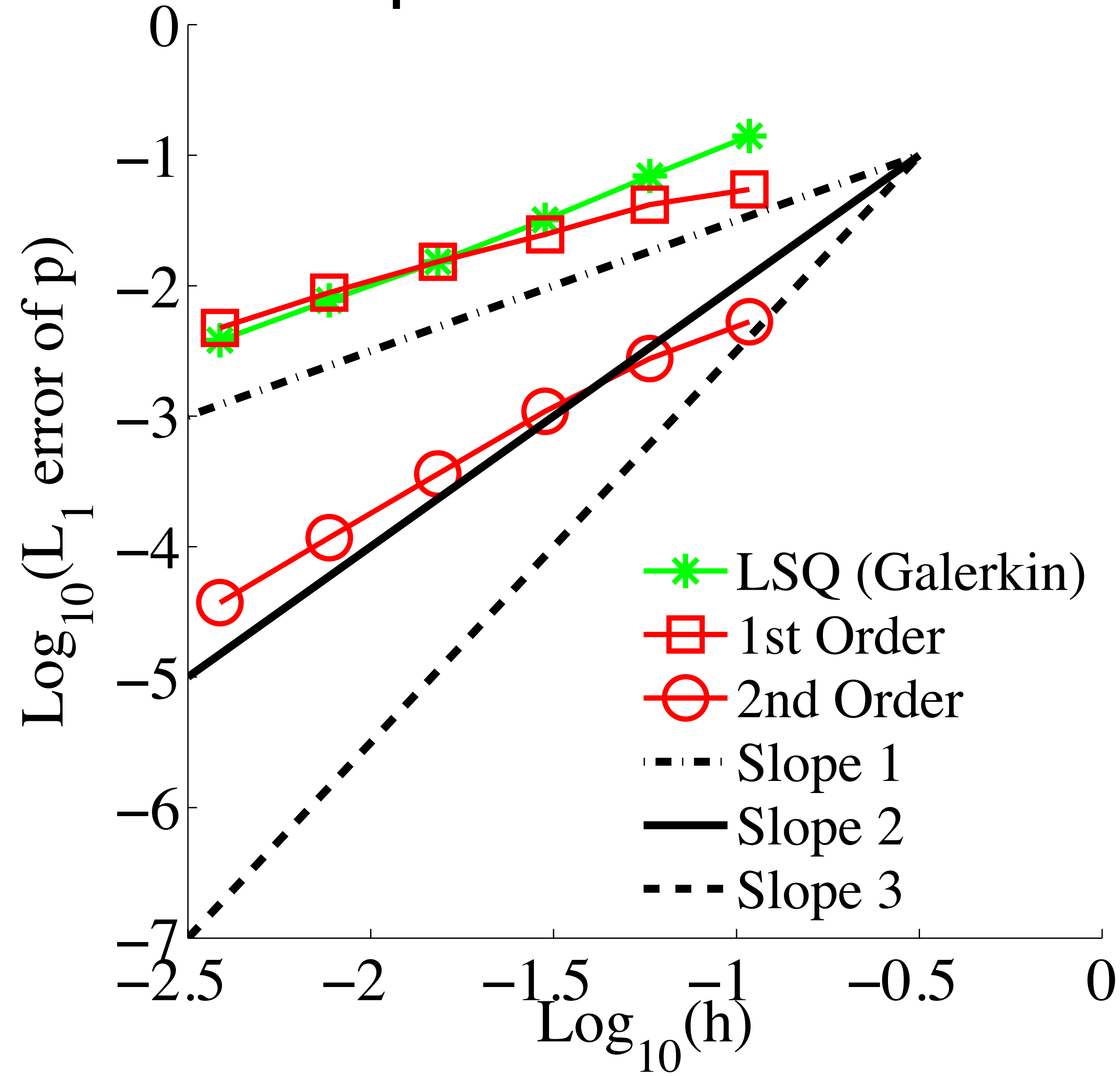
1st-order accurate solution and gradients.

Error Convergence 2

u: solution

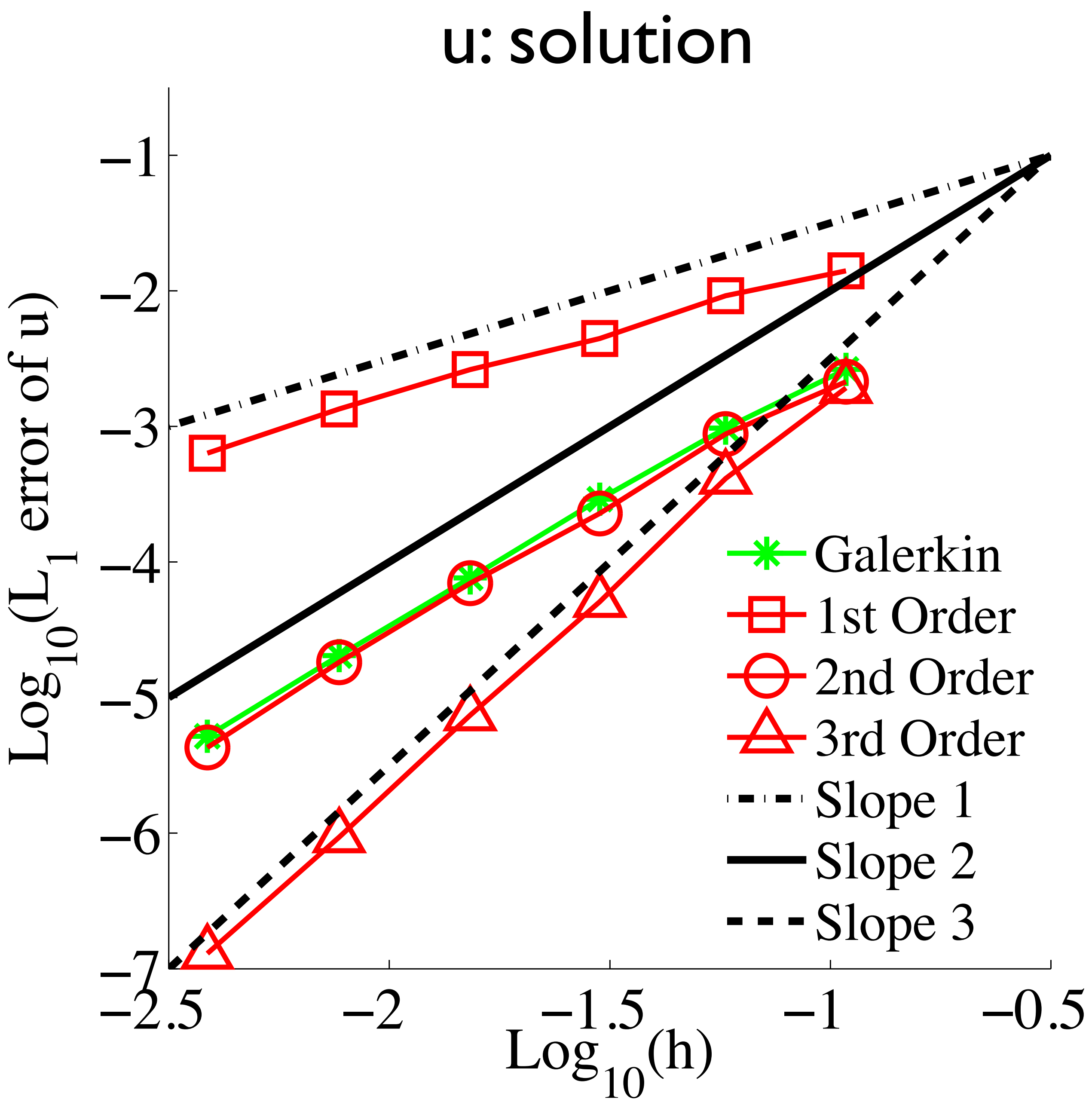


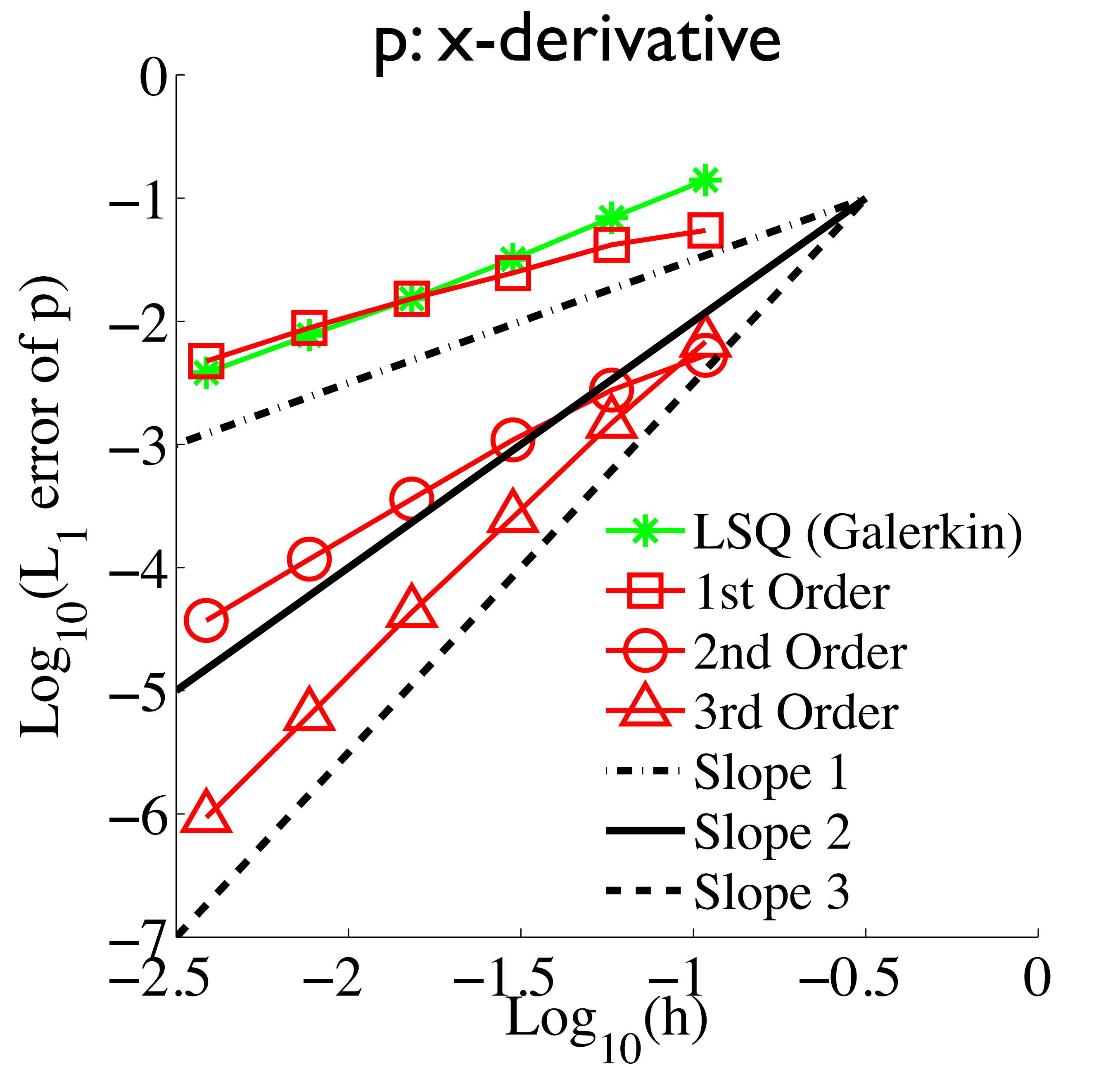
p: x-derivative



2nd-order accurate solution and gradients.

Error Convergence 3





3rd-order accurate solution and gradients.

Cost Comparison

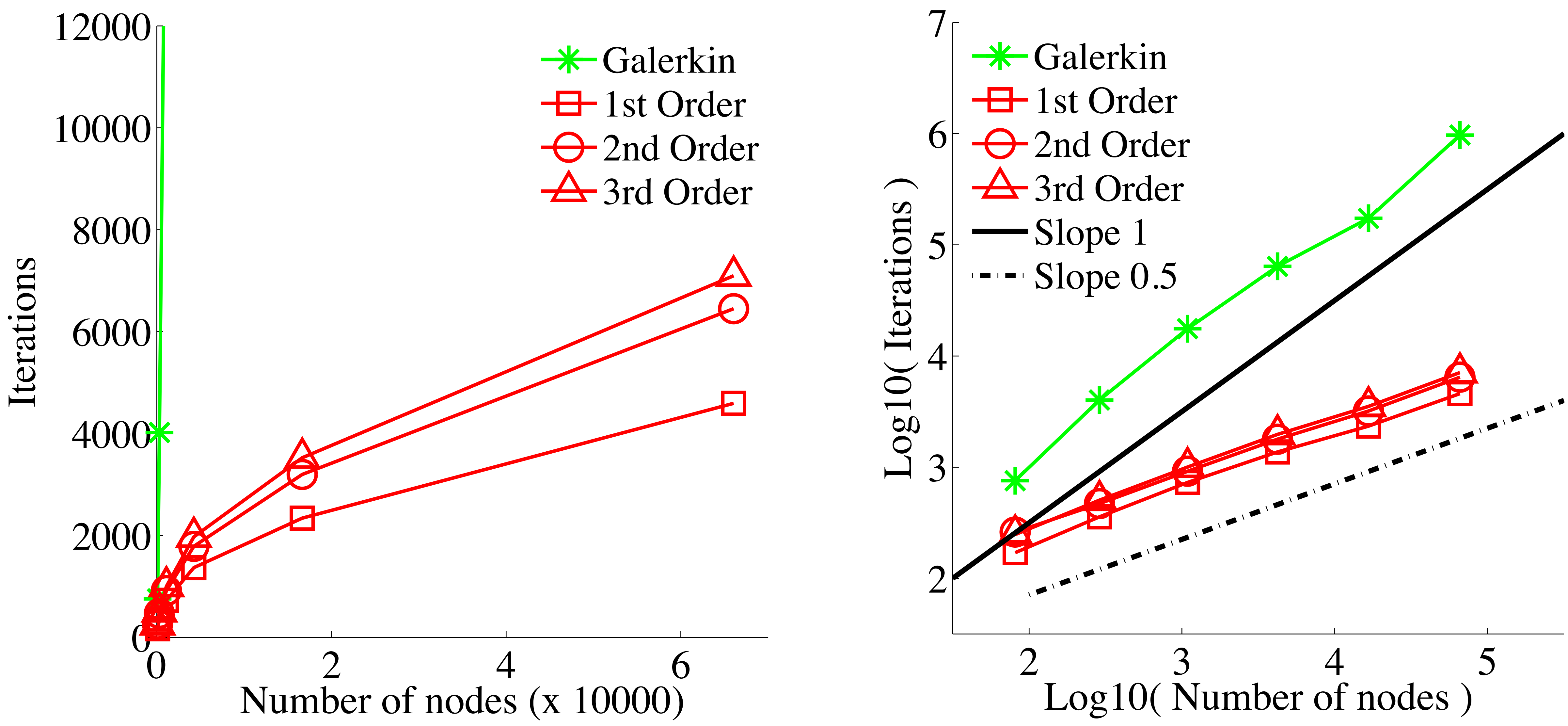
Cost per time step (the irregular grid case).

	Galerkin	First-Order	Second-Order	Third-Order
Forward Euler	0.66	1.00	1.26	1.33

Hyperbolic schemes are cheaper than Galerkin.

Number of Iterations

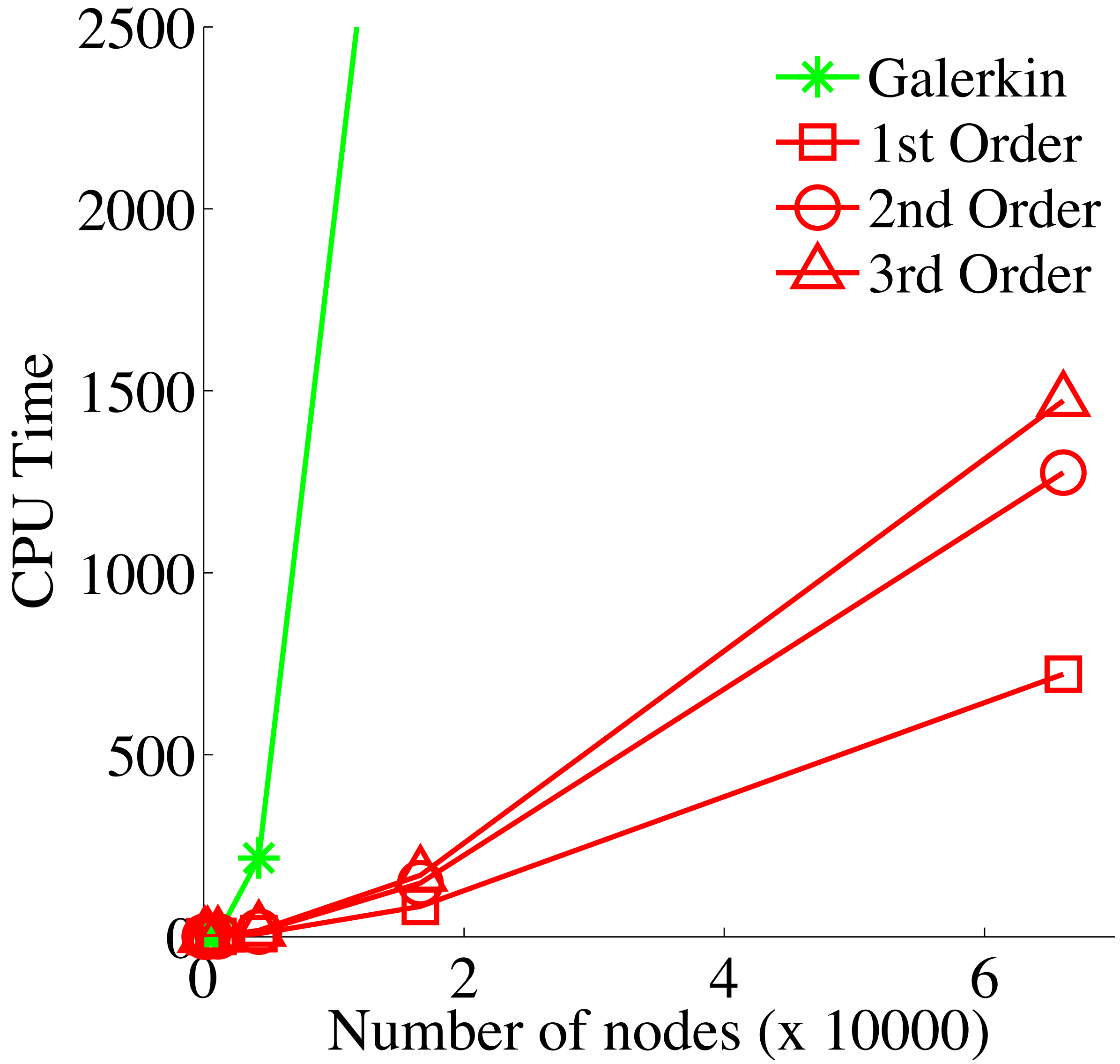
Hyperbolic schemes allow O(h) time step, not O(h^2).

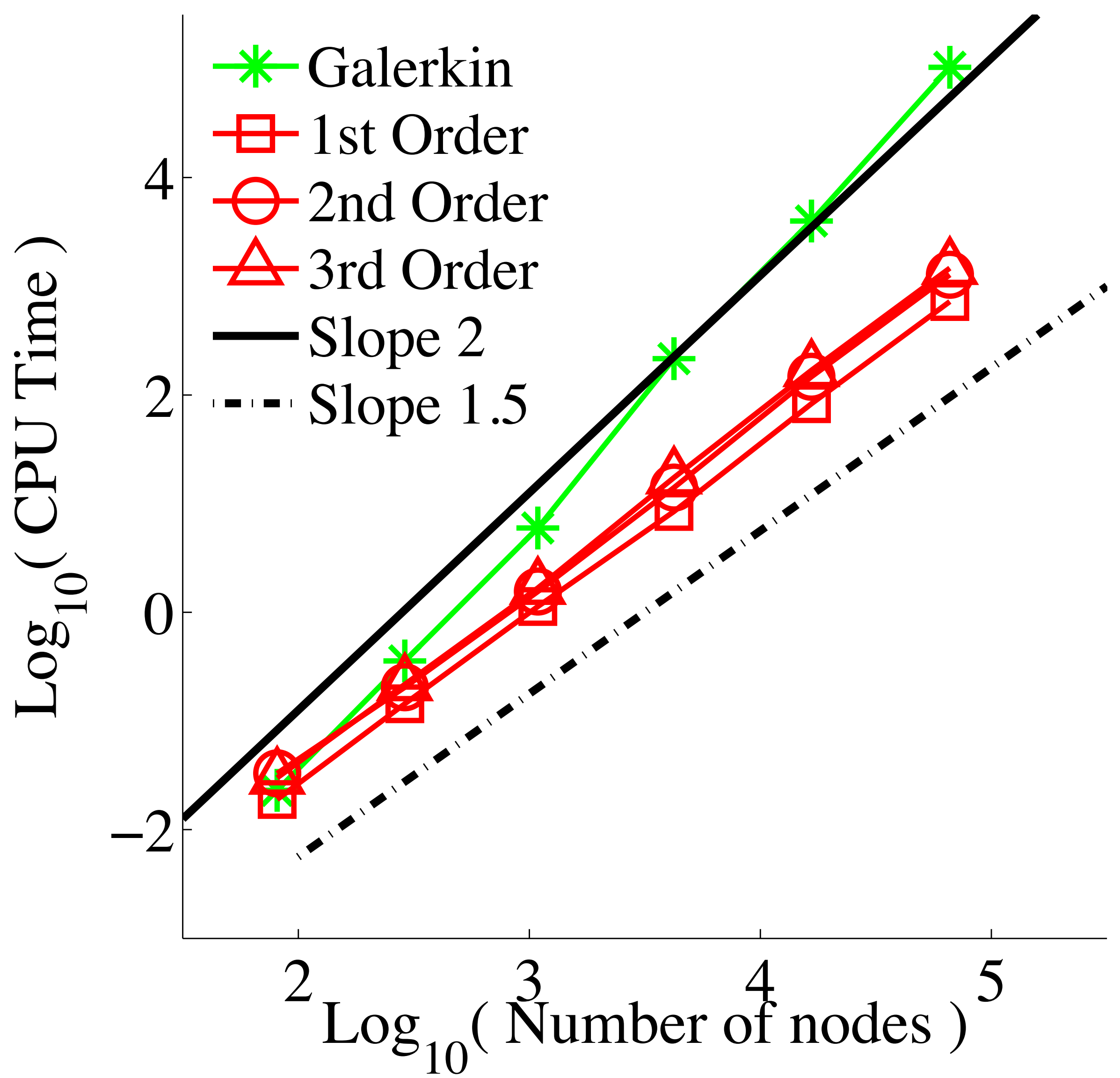


Orders of magnitude acceleration by O(h) time step.

Time to Solution

O(1/h) acceleration overwhelms the increased cost per time step.





Orders of magnitude acceleration in CPU time.

Conclusions

Diffusion and source recreated hyperbolic.

1. *Energy-stable first-order scheme is constructed for diffusion.*
2. *Equal order of accuracy for sol. and gradients on irregular grids.*
3. *Third-order diffusion scheme by fully hyperbolic system*

Third-order scheme is incomparably efficient and accurate.

Future Work

Advection-diffusion problems (uniformly third-order, stretched grids).
Implicit schemes.
Time-dependent problems.

Third-order scheme for Navier-Stokes (2nd-order scheme in AIAA2011).
New system for accurate velocity gradients (vorticity, turb source).
Hyperbolic formulation for turbulence models (robust diffusion).

Other Applications:

- *High-order residual-distribution schemes and dispersion equation at INRIA*
- *3rd-order active flux scheme at University of Michigan*
- *Entropy-consistent scheme at Universiti Sains Malaysia*
- *Many other potential applications: DG, SV, CESE, SUPG, etc.*

Declaration of Hyperbolicity

We hold these truths to be self-evident, that all PDEs are created equal, that they are endowed by us with certain unalienable Rights, that among these are hyperbolicity, consistent and accurate schemes and the pursuit of robustness.