Divergence Formulation of Source Term

Hiro Nishikawa

National Institute of Aerospace CFD Seminar, December 4, 2012

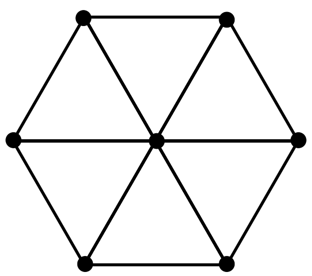
Give Up or Never Give Up

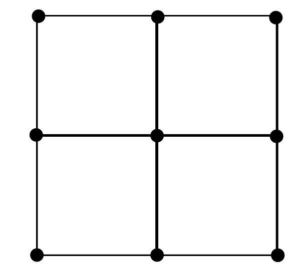
"There're times when it is good to give up."
"I agree."

I give up to get creative.

Interesting Schemes

Economical high-order schemes





Residual-distribution schemes (Roe, VKI, INRIA, etc.)

Residual-based compact schemes (Corre and Lerat, JCP2001)

Third-order edge-based finite-volume scheme (Katz and Sankaran JCP2011)

These schemes contain the target equation (or residual) in the truncation error (TE):

E.g., for linear advection, an RD scheme has the following TE,

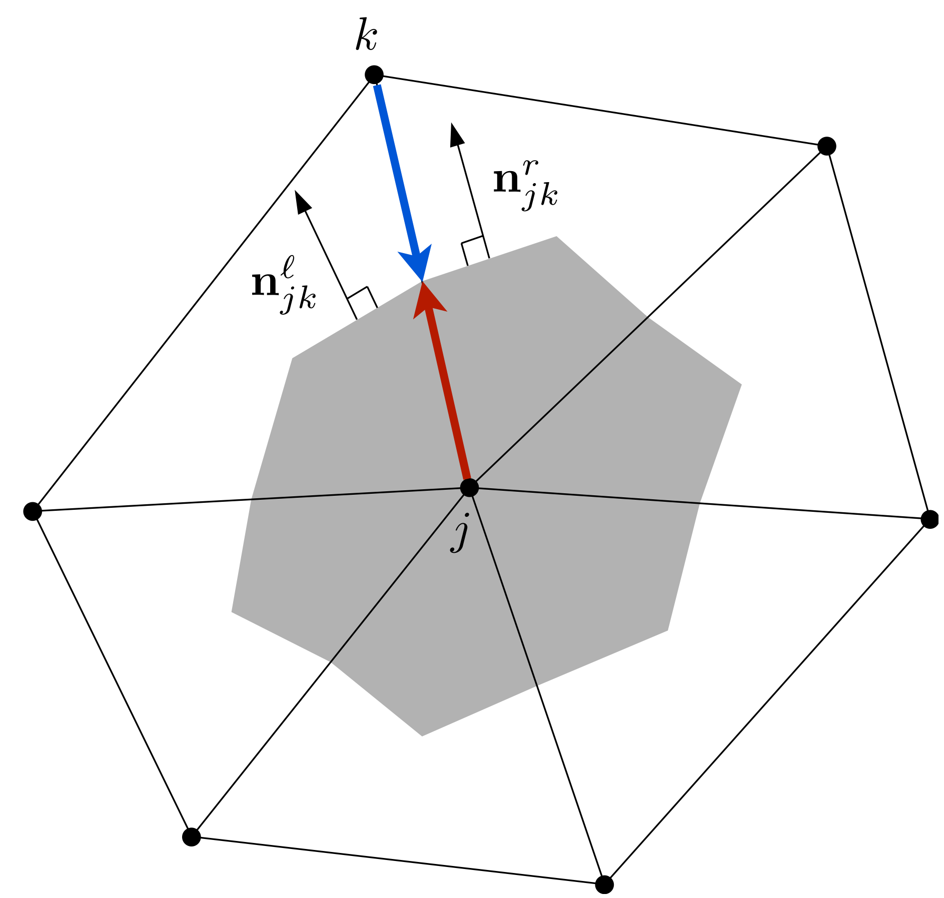
$$\mathcal{TE} = \frac{h}{2a}(a\partial_x + b\partial_y)(a\partial_x u + b\partial_y u) + O(h^2)$$

Leading term vanishes in steady state, and accuracy upgraded to second-order (Residual property).

This talk will focus on the third-order FV scheme for conservation laws with a source term.

Second-Order FV Scheme

Conservation law: $\partial_x f + \partial_y g = 0$



Edge-based finite-volume scheme:

$$0 = -\sum_{k \in \{k_j\}} \phi_{jk} A_{jk}$$

with the upwind flux at edge midpoint:

$$\phi_{jk} = \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R) \cdot \hat{\mathbf{n}}_{jk} - \frac{1}{2}|\lambda|(u_R - u_L)$$

with the left and right solution values:

$$u_L = u_j + \frac{1}{2}(\nabla u)_j \cdot \Delta \mathbf{l}_{jk}, \quad u_R = u_k - \frac{1}{2}(\nabla u)_k \cdot \Delta \mathbf{l}_{jk}$$

$$A_{jk} = |\mathbf{n}^\ell_{jk} + \mathbf{n}^r_{jk}|$$

$$\hat{\mathbf{n}}_{jk} = (\mathbf{n}^\ell_{jk} + \mathbf{n}^r_{jk})/A_{jk}$$

$$\mathbf{F} = (f, g)$$

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

$$\lambda = (\partial_u f, \partial_u g) \cdot \hat{\mathbf{n}}_{jk}$$

Second-order accurate with first-order accurate gradients.

NASA's FUN3D, Software Cradle's SC/Tetra, etc.

Third-Order FV Scheme

(Katz and Sankaran JCP2011)

1. Extrapolate the fluxes: $\phi_{jk} = \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R)\cdot\hat{\mathbf{n}}_{jk} - \frac{1}{2}|\lambda|(u_R - u_L)$

Left and right fluxes are computed by

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

2. Second-order gradients (e.g., LSQ quadratic fit)

$$(\nabla u)_j = \sum_{k\in\{k_j\}} (u_k - u_j) \begin{bmatrix} c^x_{jk} \\ c^y_{jk} \end{bmatrix} \qquad (\nabla\mathbf{F})_j = \begin{bmatrix} \frac{\partial f}{\partial u}u_x & \frac{\partial f}{\partial u}u_y \\ \frac{\partial g}{\partial u}u_x & \frac{\partial g}{\partial u}u_y \end{bmatrix}_j$$

LSQ coefficients

The resulting scheme has the truncation error on triangular(tetrahedral) grids:

$$\mathcal{TE} = (C_1\partial_{xx} + C_2\partial_{xy} + C_3\partial_{yy})(\partial_x f + \partial_y g)h^2 + O(h^3)$$

Third-order scheme on second-order stencil

Conservation Law with Source

For a conservation law with a source term:

$$\partial_x f + \partial_y g = s$$

Source term includes a time-derivative term.

We add a source term discretization to the third-order scheme:

$$0 = -\sum_{k\in\{k_j\}} \phi_{jk} A_{jk} + \int_{V_j} s\, dV$$

$$\int_{V_j} s\, dV = s_j V_j$$

Source term must be discretized to yield

$$\mathcal{TE} = (C_1 \partial_{xx} + C_2 \partial_{xy} + C_3 \partial_{yy})(\partial_x f + \partial_y g - s)h^2 + O(h^3)$$

This is critical for extending the third-order scheme to time-dependent problems.

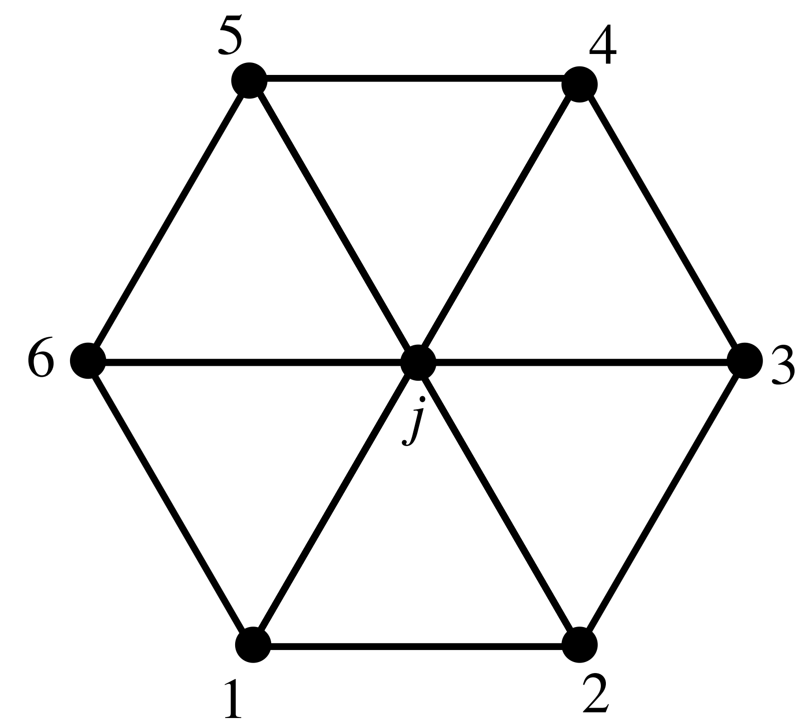
Special formulas exist for regular grids.

Formulas for Regular Grids

Equilateral-triangular stencil:

$$\int_{V_j} s\, dV = \frac{V_j}{12}(6s_j + s_1 + s_2 + s_3 + s_4 + s_5 + s_6)$$

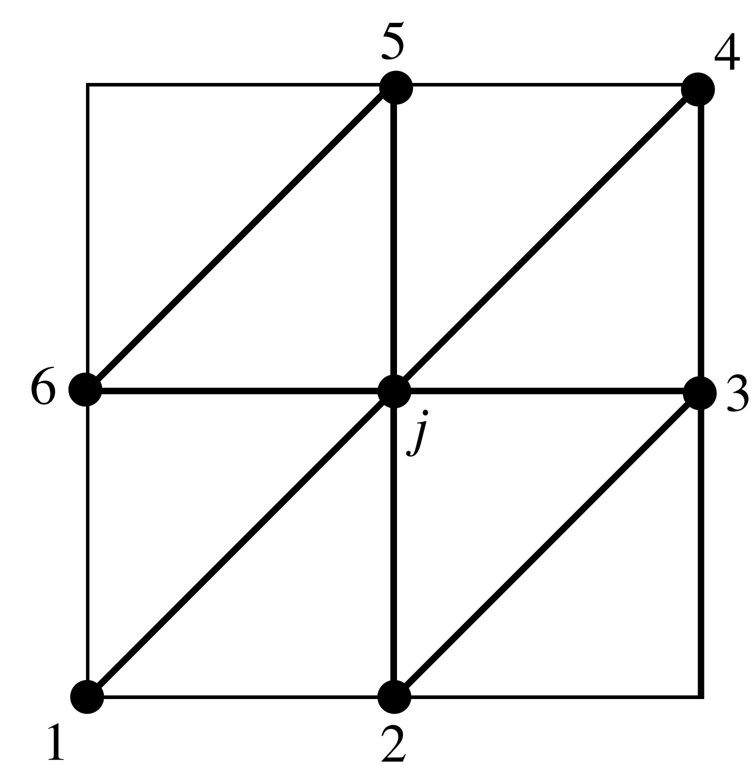
Galerkin Discretization (Katz and Sankaran JCP2011)



Right-triangular stencil:

$$\int_{V_j} s\, dV = \frac{V_j}{24}(30s_j - s_1 - s_2 - s_3 - s_4 - s_5 - s_6)$$

(Nishikawa, JCP2012)



How can I come up with such a formula for irregular grids?

I can't.

New Problem

Can we write the source term in the divergence form?

$$s \longrightarrow \partial_x f^s + \partial_y g^s$$

If possible, we can write

$$\partial_x f + \partial_y g = s$$

$$\longrightarrow \quad \partial_x f + \partial_y g = \partial_x f^s + \partial_y g^s$$

$$\longrightarrow \quad \partial_x (f - f^s) + \partial_y (g - g^s) = 0$$

Then, source term discretization will not be needed.

Nishikawa, JCP2012

Divergence Formulation of Source Term

$$s \longrightarrow \partial_x f^s + \partial_y g^s$$

where

$$f^s = \frac{1}{2}(x - x_j)s + \frac{1}{4}(x - x_j)^2 \partial_x s + \frac{1}{12}(x - x_j)^3 \partial_{xx} s$$

$$g^s = \frac{1}{2}(y - y_j)s + \frac{1}{4}(y - y_j)^2 \partial_y s + \frac{1}{12}(y - y_j)^3 \partial_{yy} s$$

(xj, yj) is a point in a computational grid.

Then, $\partial_x f + \partial_y g = s$ can be written as a single divergence form:

$$\partial_x (f - f^s) + \partial_y (g - g^s) = 0$$

Source term discretization is no longer needed.

Gradient and Hessian of the source are needed,
which can be computed by the quadratic fit.

Equivalent up to Third-Order

The divergence form,

$$\partial_x (f - f^s) + \partial_y (g - g^s) = 0$$

can be expanded as

$$\underline{\partial_x f + \partial_y g =} \; s + \frac{1}{12}(x - x_j)^3 \partial_{xxx} s + \frac{1}{12}(y - y_j)^3 \partial_{yyy} s$$

At node j, it is equivalent to the original equation.
In the neighborhood, equivalent up to third-order,

which is sufficient for third-order scheme.

One-Component Forms

The divergence form is equivalent to the following:

$$f^s = (x - x_j)s + \frac{1}{2}(x - x_j)^2 \partial_x s + \frac{1}{6}(x - x_j)^3 \partial_{xx} s$$

$$g^s = 0$$

or

$$f^s = 0$$

$$g^s = (y - y_j)s + \frac{1}{2}(y - y_j)^2 \partial_y s + \frac{1}{6}(y - y_j)^3 \partial_{yy} s$$

All are equivalent to one another up to third-order.

Nishikawa, JCP2012

Third-Order Scheme

Conservation law: $\partial_x(f - f^s) + \partial_y(g - g^s) = 0$

Edge-based finite-volume scheme:

$$0 = -\sum_{k \in \{k_j\}} (\phi_{jk} + \psi_{jk}) A_{jk}$$

with the central flux for the source flux:

$$\psi_{jk} = \frac{1}{2}(\mathbf{F}^s_L + \mathbf{F}^s_R) \cdot \hat{\mathbf{n}}_{jk} \qquad \mathbf{F}^s = (f^s, g^s)$$

with the left and right flux values:

$$\mathbf{F}^s_L = \mathbf{F}^s_j + \frac{1}{2}(\nabla \mathbf{F}^s)_j \cdot \Delta \mathbf{l}_{jk}, \quad \mathbf{F}^s_R = \mathbf{F}^s_k - \frac{1}{2}(\nabla \mathbf{F}^s)_k \cdot \Delta \mathbf{l}_{jk}$$

The resulting scheme has the truncation error:

$$\mathcal{TE} = (C_1 \partial_{xx} + C_2 \partial_{xy} + C_3 \partial_{yy})(\partial_x f + \partial_y g - s)h^2 + O(h^3)$$

Third-order achieved without source term discretization.

One-Component Case

$$f^s = (x - x_j)s + \frac{1}{2}(x - x_j)^2 \partial_x s + \frac{1}{6}(x - x_j)^3 \partial_{xx} s$$

$$g^s = 0$$

Edge-based finite-volume scheme:

$$0 = - \sum_{k \in \{k_j\}} (\phi_{jk} + \psi_{jk}) A_{jk} \qquad 0 = - \sum_{k \in \{k_j\}} \phi_{jk} A_{jk} + \int_{V_j} s \, dV$$

with the central flux for the source flux:

$$\psi_{jk} = \frac{1}{2}(f_L^s + f_R^s, 0) \cdot \hat{\mathbf{n}}_{jk}$$

with the left and right flux values:

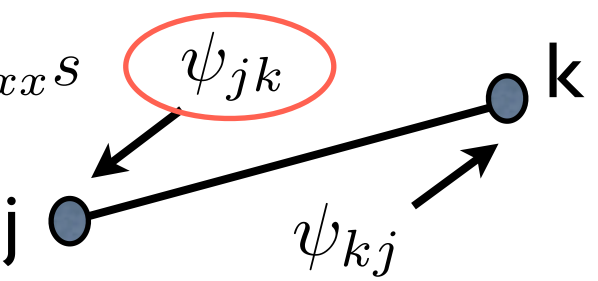
$$f_L^s = f_j^s + \frac{1}{2}(\nabla f^s)_j \cdot \Delta \mathbf{l}_{jk}, \quad f_R^s = f_k^s - \frac{1}{2}(\nabla f^s)_k \cdot \Delta \mathbf{l}_{jk}$$

Source discretization replaced by a scalar central scheme.

Some details not given in Nishikawa, JCP2012

Source Flux and Flux Gradients

Source flux: $f^s = (x - x_j)s + \frac{1}{2}(x - x_j)^2 \partial_x s + \frac{1}{6}(x - x_j)^3 \partial_{xx} s$



1. Compute the source flux at nodes:

$$f_j^s = 0, \quad f_k^s = (x_k - x_j)s_k + \frac{1}{2}(x_k - x_j)^2 \partial_x s_k + \frac{1}{6}(x_k - x_j)^3 \partial_{xx} s_k$$

2. Compute the gradient of the source flux:

Ignored for third-order

$$(\nabla f)_j = \begin{bmatrix} s_j \\ 0 \end{bmatrix} \quad (\nabla f)_k = \begin{bmatrix} s_k + \frac{1}{6}(x_k - x_j)^3 \partial_{xxx} s_k \\ (x_k - x_j)\partial_y s_k + \frac{1}{2}(x_k - x_j)^2 \partial_{xy} s_k + \frac{1}{6}(x_k - x_j)^3 \partial_{xxy} s_k \end{bmatrix}$$

Ignored for third-order

3. Compute the left and right fluxes:

$$f_L^s = f_j^s + \frac{1}{2}(\nabla f^s)_j \cdot \Delta \mathbf{l}_{jk}, \quad f_R^s = f_k^s - \frac{1}{2}(\nabla f^s)_k \cdot \Delta \mathbf{l}_{jk}$$

4. Compute the central flux for node j: $\psi_{jk} = \frac{1}{2}(f_L^s + f_R^s, 0) \cdot \hat{\mathbf{n}}_{jk}$

The other flux needs to be computed separately because $\psi_{jk} \neq -\psi_{kj}$.

Source flux discretization is not conservative (of course).

Exact Divergence Form

If the source term is simple enough, e.g.,

$$\partial_x f + \partial_y g = \cos(x - y)$$

it can be written exactly as

$$\partial_x f + \partial_y g = \partial_x f^s + \partial_y g^s$$

$$f^s = \sin(x - y), \quad g^s = 0$$ (The choice is not unique.)

Therefore, again, source term discretization is not needed.

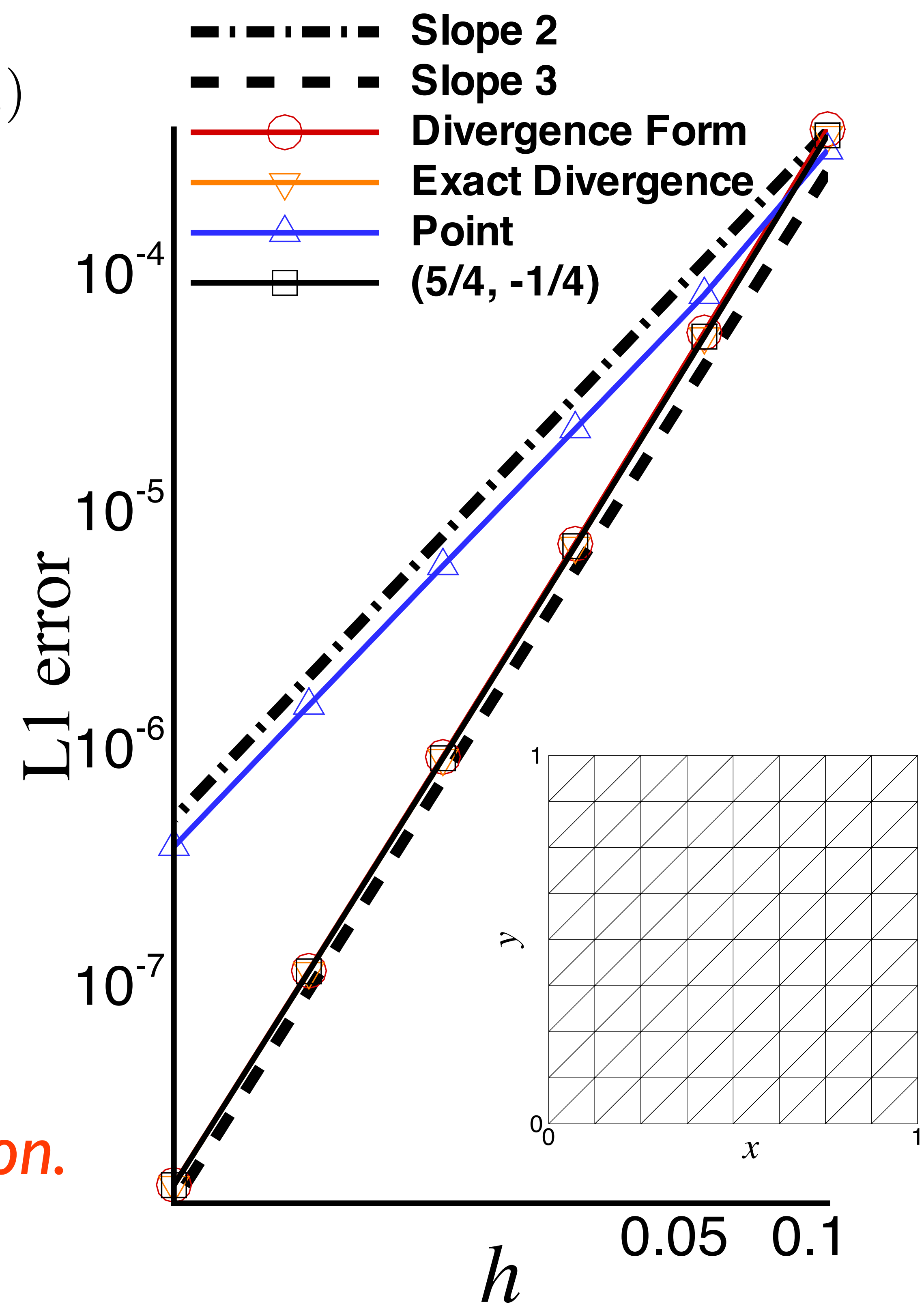
1. *Second derivatives are not needed.*
2. *Gradient of the source term is needed.*
   (It can be computed analytically or by a quadratic fit.)
3. *This is not possible for time-derivative terms (only discrete values).*

Burgers Equation: *Regular Grids*

$$\partial_x(u^2/2) + \partial_y u = \cos(x-y)(\sin(x-y)-1)$$

Exact solution: $u(x,y) = \sin(x-y)$

- n x n grids: n = 9, 17, 33, 65, 129, 257.
- Dirichlet boundary condition.
- 6 neighbors for quadratic fit.
- Time-stepping by RK2 to steady state.
- (5/4, -1/4) indicates the special formula.



Second-order with the point discretization.

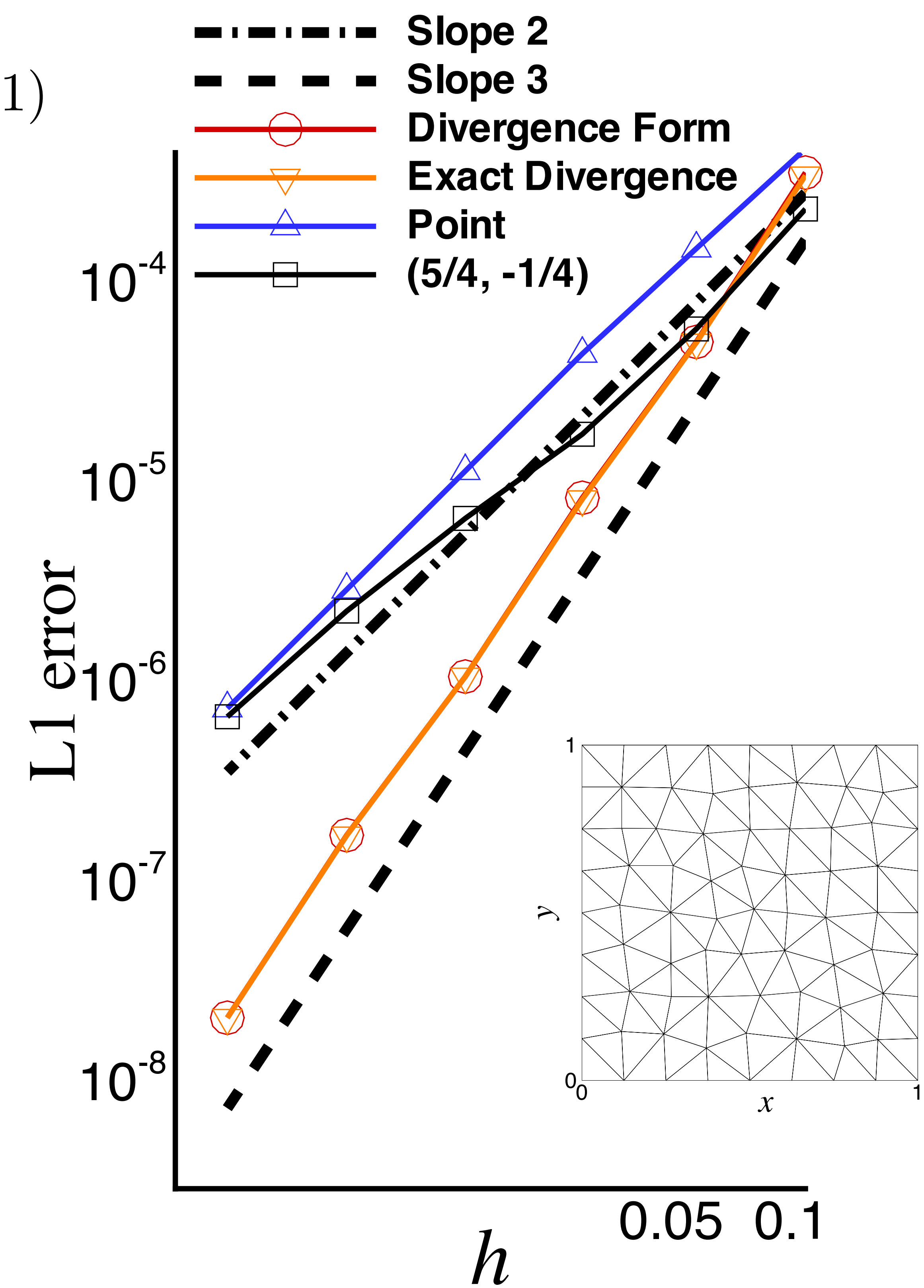
Burgers Equation: *Irregular Grids*

$$\partial_x(u^2/2) + \partial_y u = \cos(x-y)(\sin(x-y)-1)$$

Exact solution: $u(x,y) = \sin(x-y)$

- 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)
- Time-stepping by RK2 to steady state.
- (5/4, -1/4) indicates the special formula.

Only the divergence formulation achieved third-order accuracy.



Conclusion

Third-order finite-volume scheme made simple
for source term by the divergence formulation.

Future work:

Relation with the formula of Katz (Katz 2012, unpublished); looks similar.
Application to unsteady computation (time derivative as a source).
Application to other discretization methods.
Application to other types of source terms (involving the solution). → *Hyperbolic*

which I might have never been able to even think about
if I had not given up.