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

Hiroaki Nishikawa

National Institute of Aerospace

21st AIAA Computational Fluid Dynamics Conference, San Diego, CA

June 24-27, 2013

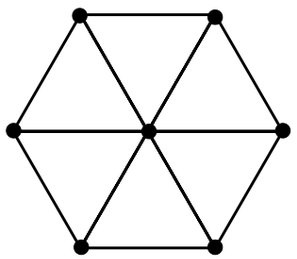
Supported by ARO (PM: Dr. Frederick Ferguson), Software Cradle, NASA LaRC.

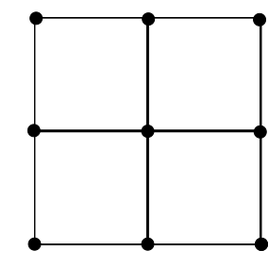
Do you read newspaper?

Stay informed.

Someone somewhere may have already solved your problem.

Interesting High-Order Schemes



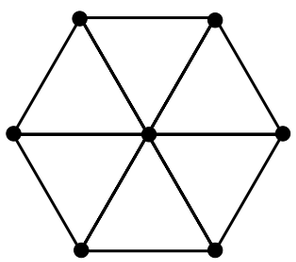


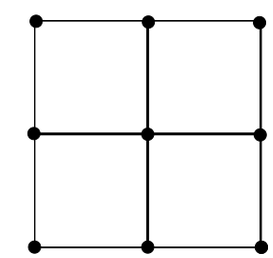
Residual-distribution schemes (Roe, VKI, INRIA, etc.)

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

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

Interesting 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)

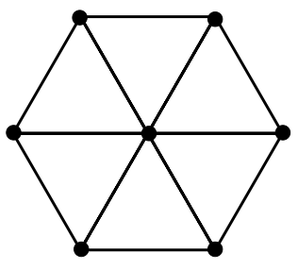
These schemes contain the target equation (or residual) in the truncation error (TE):

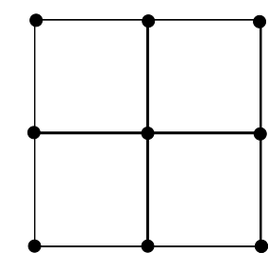
E.g., for linear advection, an RD scheme has the following TE,

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

Leading term vanishes in steady state, and accuracy upgraded to second-order.

Interesting 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.

This talk will focus on the third-order FV scheme for *advection-diffusion* equation.

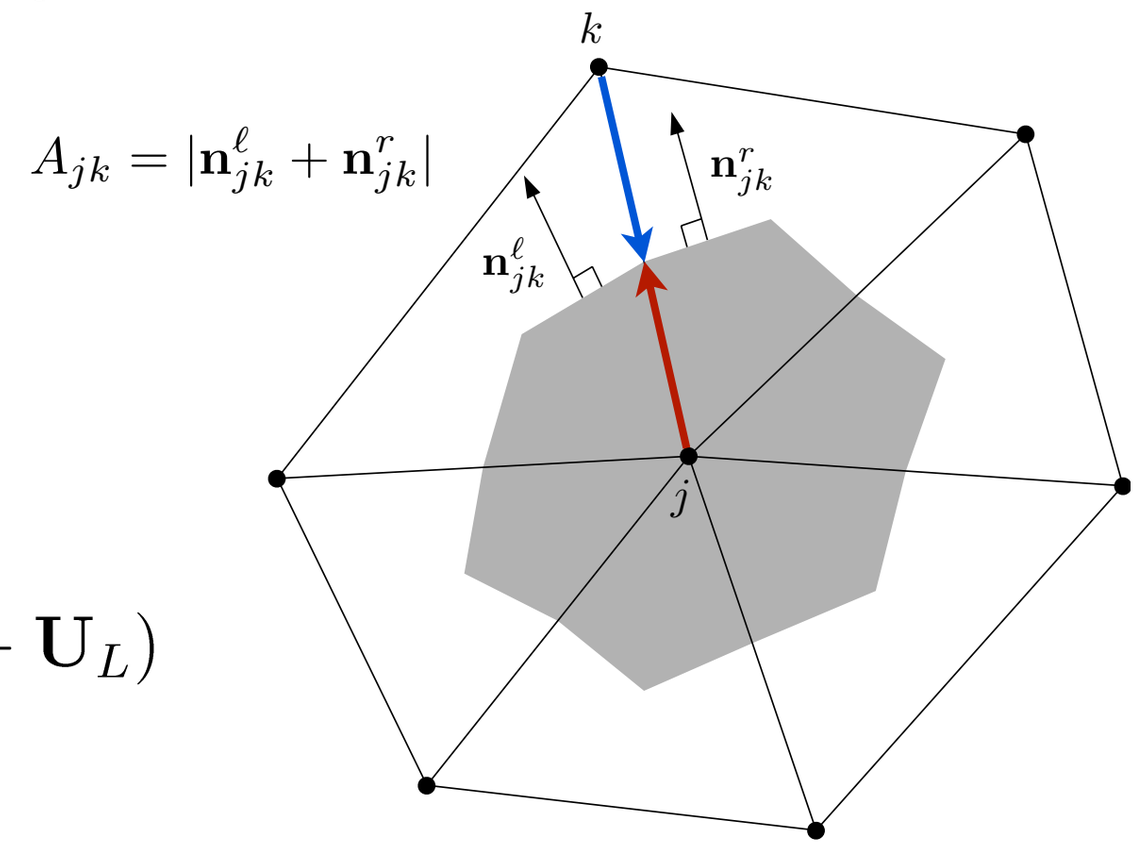
Edge-Based Finite-Volume Method

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

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

Edge-based finite-volume scheme:

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



with the upwind flux at edge midpoint:

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

Accuracy with left/right states:

- 1st-order with nodal values
- 2nd-order with linear extrapolation, linear LSQ
- 3rd-order with linear extrapolation, quadratic LSQ

Katz&Sankaran(JCP2011)

Implicit Solver

Residual at node j : $$0 = -\sum_{k \in \{k_j\}} \Phi_{jk} A_{jk} + \mathbf{S}_j V_j$$

System of residual equations : $$0 = \mathbf{Res}(\mathbf{U}_h)$$

Update : $$\mathbf{U}_h^{n+1} = \mathbf{U}_h^n + \Delta \mathbf{U}_h$$

$$\frac{\partial \mathbf{Res}}{\partial \mathbf{U}_h} \Delta \mathbf{U}_h = -\mathbf{Res}(\mathbf{U}_h^n)$$

- Jacobian based on 1st-order scheme
- GS relaxation with tolerance 0.01

Third-Order Advection

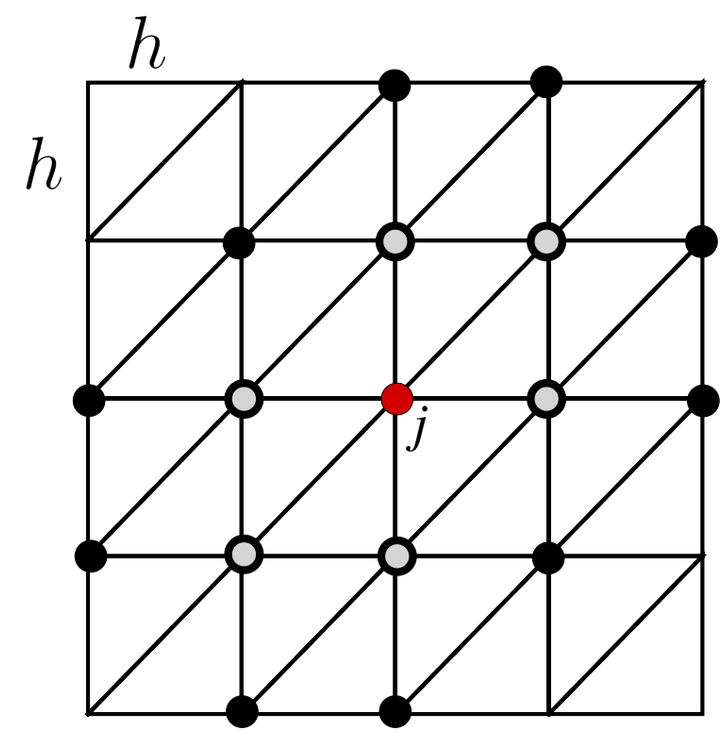
For triangular/tetrahedral grids.

Scalar advection equation:

$$\partial_x f + \partial_y g = 0$$

where

$$(f, g) = (au, bu)$$



Third-order FV scheme has the truncation error:

$$\mathcal{T}_j^{\mathrm{adv}} = \frac{h^2}{12}\left[\partial_{xx}(\partial_x f + \partial_y g) + \partial_{xy}(\partial_x f + \partial_y g) + \partial_{yy}(\partial_x f + \partial_y g)\right] + O(h^3)$$

Third-Order Advection

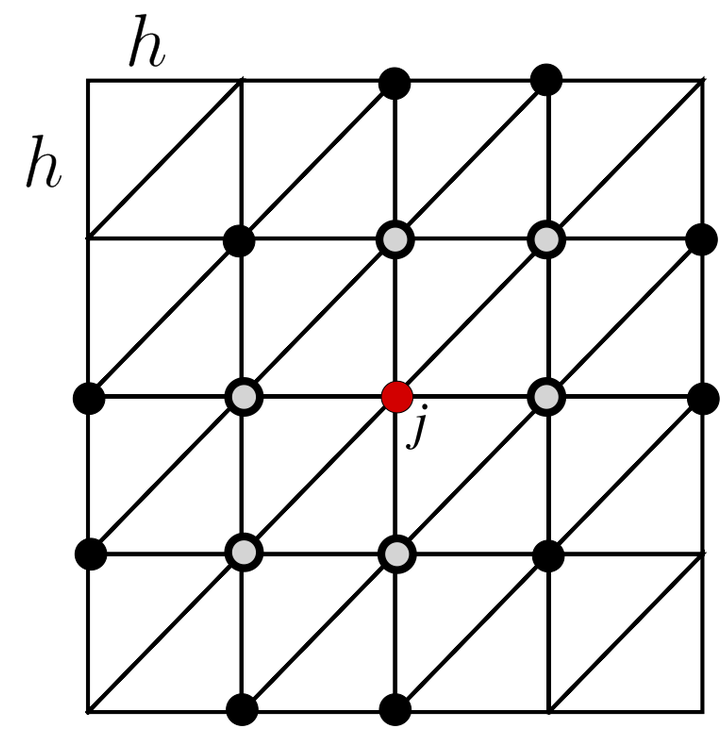
For triangular/tetrahedral grids.

Scalar advection equation:

$$\partial_x f + \partial_y g = 0$$

where

$$(f, g) = (au, bu)$$



Third-order FV scheme has the truncation error:

$$\mathcal{T}_j^{\mathrm{adv}} = \frac{h^2}{12}\left[\partial_{xx}(\partial_x f + \partial_y g) + \partial_{xy}(\partial_x f + \partial_y g) + \partial_{yy}(\partial_x f + \partial_y g)\right] + O(h^3)$$

Third-Order Advection

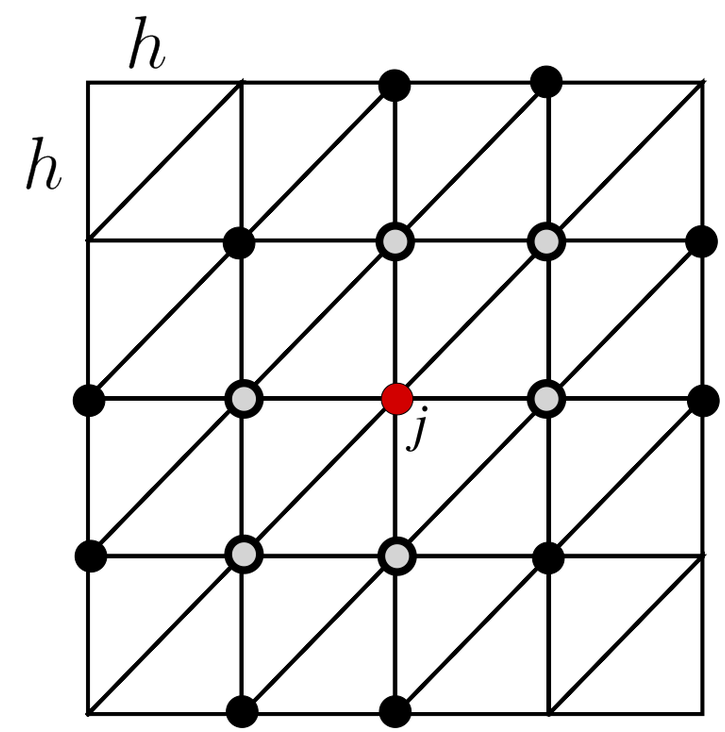
For triangular/tetrahedral grids.

Scalar advection equation:

$$\partial_x f + \partial_y g = 0$$

where

$$(f, g) = (au, bu)$$



Third-order FV scheme has the truncation error:

$$\mathcal{T}_j^{\mathrm{adv}} = \frac{h^2}{12}\left[\partial_{xx}(\partial_x f + \partial_y g) + \partial_{xy}(\partial_x f + \partial_y g) + \partial_{yy}(\partial_x f + \partial_y g)\right] + O(h^3)$$

3rd-order accurate on 2nd-order stencil *Low-Cost High-Order*

Third-Order Advection-Diffusion

$$\partial_x f + \partial_y g - \nu \left(\partial_{xx} u + \partial_{yy} u\right) = 0$$

Diffusion scheme MUST have a 2nd-order truncation error in the form:

$$\boxed{\mathcal{T}_j^{\text{diff}} = -\frac{h^2}{12}\left[\partial_{xx}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{xy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{yy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right))\right] + O(h^3)}$$

Third-Order Advection-Diffusion

$$\partial_x f + \partial_y g - \nu \left(\partial_{xx} u + \partial_{yy} u\right) = 0$$

Diffusion scheme MUST have a 2nd-order truncation error in the form:

$$\boxed{\mathcal{T}_j^{\text{diff}} = -\frac{h^2}{12}\left[\partial_{xx}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{xy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{yy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right))\right] + O(h^3)}$$

so that the advection-diffusion scheme has TE in the form:

$$\mathcal{T}_j^{\text{adv-diff}} = \mathcal{T}_j^{\text{adv}} + \mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx} r + \partial_{xy} r + \partial_{yy} r\right] + O(h^3)$$

$$r = \partial_x f + \partial_y g - \nu\left(\partial_{xx} u + \partial_{yy} u\right)$$

Third-Order Advection-Diffusion

$$\partial_x f + \partial_y g - \nu \left(\partial_{xx} u + \partial_{yy} u\right) = 0$$

Diffusion scheme MUST have a 2nd-order truncation error in the form:

$$\boxed{\mathcal{T}_j^{\text{diff}} = -\frac{h^2}{12}\left[\partial_{xx}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{xy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{yy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right))\right] + O(h^3)}$$

so that the advection-diffusion scheme has TE in the form:

$$\mathcal{T}_j^{\text{adv-diff}} = \mathcal{T}_j^{\text{adv}} + \mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\cancel{\partial_{xx} r} + \cancel{\partial_{xy} r} + \cancel{\partial_{yy} r}\right] + O(h^3)$$

$$r = \partial_x f + \partial_y g - \nu\left(\partial_{xx} u + \partial_{yy} u\right)$$

Third-Order Advection-Diffusion

$$\partial_x f + \partial_y g - \nu \left(\partial_{xx} u + \partial_{yy} u\right) = 0$$

Diffusion scheme MUST have a 2nd-order truncation error in the form:

$$\boxed{\mathcal{T}_j^{\text{diff}} = -\frac{h^2}{12}\left[\partial_{xx}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{xy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{yy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right))\right] + O(h^3)}$$

so that the advection-diffusion scheme has TE in the form:

$$\mathcal{T}_j^{\text{adv-diff}} = \mathcal{T}_j^{\text{adv}} + \mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\cancel{\partial_{xx} r} + \cancel{\partial_{xy} r} + \cancel{\partial_{yy} r}\right] + O(h^3)$$

$$r = \partial_x f + \partial_y g - \nu\left(\partial_{xx} u + \partial_{yy} u\right)$$

A compatible diffusion scheme is required for uniform third-order accuracy.

Linear Galerkin for Diffusion

$$\partial_x f + \partial_y g - \nu \left(\partial_{xx} u + \partial_{yy} u\right) = 0$$

Linear Galerkin scheme has the TE in the form:

$$\mathcal{T}_j^{\text{diff}} = \frac{\nu h^2}{12}(\partial_{xxxx} u + \partial_{yyyy} u) + O(h^3)$$

[3rd-order advection] + [Linear Galerkin] gives

$$\mathcal{T}_j^{\text{adv-diff}} = \mathcal{T}_j^{\text{adv}} + \mathcal{T}_j^{\text{diff}}$$

$$= \frac{h^2}{12}\left[\partial_{xx}(\underline{\partial_x f + \partial_y g + \nu \partial_{xx} u}) + \partial_{xy}(\underline{\partial_x f + \partial_y g}) + \partial_{yy}(\underline{\partial_x f + \partial_y g + \nu \partial_{yy} u})\right] + O(h^3)$$

None of these terms vanish for advection-diffusion in the steady state.

This is a second-order scheme...

3rd-order only in the advection limit.

Third-Order Galerkin for Diffusion

$$\partial_x f + \partial_y g - \nu \left(\partial_{xx} u + \partial_{yy} u\right) = 0$$

Third-order Galerkin (Nishikawa, VKI Lecture, 2006) has the TE in the form:

$$\mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{xy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{yy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right))\right] + O(h^3)$$

Third-Order Galerkin for Diffusion

$$\partial_x f + \partial_y g - \nu \left(\partial_{xx} u + \partial_{yy} u\right) = 0$$

Third-order Galerkin (Nishikawa, VKI Lecture, 2006) has the TE in the form:

$$\mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{xy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{yy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right))\right] + O(h^3)$$

[3rd-order advection] + [3rd-order Galerkin] gives

$$\mathcal{T}_j^{\text{adv-diff}} = \mathcal{T}_j^{\text{adv}} + \mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx}r' + \partial_{xy}r' + \partial_{yy}r'\right] + O(h^3)$$

$$r' = \partial_x f + \partial_y g + \nu\left(\partial_{xx}u + \partial_{yy}u\right)$$

Third-Order Galerkin for Diffusion

$$\partial_x f + \partial_y g - \nu \left(\partial_{xx} u + \partial_{yy} u\right) = 0$$

Third-order Galerkin (Nishikawa, VKI Lecture, 2006) has the TE in the form:

$$\mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{xy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{yy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right))\right] + O(h^3)$$

[3rd-order advection] + [3rd-order Galerkin] gives

$$\mathcal{T}_j^{\text{adv-diff}} = \mathcal{T}_j^{\text{adv}} + \mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx}r' + \partial_{xy}r' + \partial_{yy}r'\right] + O(h^3)$$

$$r' = \partial_x f + \partial_y g + \nu\left(\partial_{xx}u + \partial_{yy}u\right)$$

Third-Order Galerkin for Diffusion

$$\partial_x f + \partial_y g - \nu \left(\partial_{xx} u + \partial_{yy} u\right) = 0$$

Third-order Galerkin (Nishikawa, VKI Lecture, 2006) has the TE in the form:

$$\mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{xy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right)) + \partial_{yy}(\nu\left(\partial_{xx}u + \partial_{yy}u\right))\right] + O(h^3)$$

[3rd-order advection] + [3rd-order Galerkin] gives

$$\mathcal{T}_j^{\text{adv-diff}} = \mathcal{T}_j^{\text{adv}} + \mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx}r' + \partial_{xy}r' + \partial_{yy}r'\right] + O(h^3)$$

$$r' = \partial_x f + \partial_y g + \nu\left(\partial_{xx}u + \partial_{yy}u\right)$$

This is NOT uniformly third-order accurate.

3rd-order only in the advection limit or in the diffusion limit.

Edge-Based Diffusion Scheme

The edge-based diffusion scheme (e.g., Nishikawa, AIAA2010, C&F2011) has a potential for achieving uniform third-order accuracy with a cubic fit, requiring at least, 9 neighbors in 2D and 18 neighbors in 3D.

1. Very large stencil.
2. Robustness (cubic gradient reconstruction).
3. The h-ellipticity.
4. Inconsistent Jacobian (lack of compact 1st-order scheme)

Edge-Based Diffusion Scheme

The edge-based diffusion scheme (e.g., Nishikawa, AIAA2010, C&F2011) has a potential for achieving uniform third-order accuracy with a cubic fit, requiring at least, 9 neighbors in 2D and 18 neighbors in 3D.

1. Very large stencil.
2. Robustness (cubic gradient reconstruction).
3. The h-ellipticity.
4. Inconsistent Jacobian (lack of compact 1st-order scheme)

Not pursued in this work. Why?

Edge-Based Diffusion Scheme

The edge-based diffusion scheme (e.g., Nishikawa, AIAA2010, C&F2011) has a potential for achieving uniform third-order accuracy with a cubic fit, requiring at least, 9 neighbors in 2D and 18 neighbors in 3D.

1. Very large stencil.
2. Robustness (cubic gradient reconstruction).
3. The h-ellipticity.
4. Inconsistent Jacobian (lack of compact 1st-order scheme)

Not pursued in this work. Why?
Because I read the newspaper.

THE CFD NEWS

- FRIDAY, JANUARY 11 2013 -

www.cfdbooks.com THE WORLD'S FAVORITE CFD NEWSPAPER - Since 1910

Hyperbolicity Declared for PDEs

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.

Declaration of hyperbolicity presented by Dr. Hiroaki Nishikawa (NIA) at the 51st AIAA Aerospace Sciences Meeting, January 10, 2013, Grapevine, TX, USA

Hyperbolicity was declared, for the first time in the entire CFD history, for all partial differential equations (PDEs) at 51st AIAA Aerospace Sciences Meeting held in Grapevine, Texas, on January 10, 2013. It was declared totally unexpectedly at the technical talk by Dr. Hiroaki Nishikawa, Senior Research Scientist at National Institute of Aerospace (USA). "I was preparing for this moment since 2007. I thought I had to do it now in order to rescue people suffering from various difficulties with non-hyperbolic PDEs like parabolic PDEs," says Nishikawa.

It seems like the declaration came to rescue, but it has generated heated controversy among researchers.

"That's totally crazy. I'm not going to let anyone make me hyperbolic," says Dr. Parabolic at the Viscous Institute of Technology. He claims, "It is simply wrong because each type of PDEs is designed to model specific physical phenomena and one may not change it no matter what. It is a completely wrong idea." Nishikawa argues that the hyperbolicity is just for the sake of numerically solving the PDEs and at the end of the day the numerical solution satisfies the original equations, parabolic or whatever. Parabolic counters,

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

$$\Downarrow$$

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

PDE is made hyperbolic by turning non-hyperbolic terms on the right hand side, including a source term, into a hyperbolic system such that it reduces to the original in the steady state.

"Totally insane. Successful numerical schemes should reflect the nature of the PDE they are solving. Upwind scheme for isotropic diffusion has no chance to work." Although it sounds right, the numerical results shown by Nishikawa indicate that the claim is not true. In fact, unusually good results have been obtained by the upwind scheme for diffusion and viscous flow problems. They are unusually good because high-order accurate gradients have been obtained at a dramatically 'reduced' cost.

These interesting results have attracted a number of researchers around the world. Professor Elliptic at the University of Smooth says, "It is quite nice and welcome. As I see it, the hyperbolized parabolic-PDE is hyperbolic in time but remains elliptic in space. It's just like the acoustic sub-system of the Euler equations, which is hyperbolic in time but elliptic in space in subsonic flows." Dr. Muscl at Monotone National Laboratory (currently under reconstruction) is another researcher who welcomed the declaration. He says, "It's a wonderful news. I feel like I've got a lot more places to work at than I thought."

On the other hand, Always Nolimiter, a graduate student of aerospace engineering, says he's been scared to death since he heard the news. He says, "I'm so scared because the hyperbolic Navier-Stokes equations may generate additional shockwaves due to the hyperbolic viscous term. I just don't know what to do. They're gonna blow me up!" According to Professor Elliptic, however, the student is worrying for nothing. He says, "No shockwaves will be generated by the hyperbolic viscous term. Like I said, they are elliptic in space. There will be no shockwaves running across the domain."

Taro Sushiyama, one of the best sushi chefs in town, commented on the analogy of Sushi Burger repeatedly used by Nishikawa to illustrate the concept. He says, "It looks eccentric. It's against tradition and not acceptable in our world. But it's an interesting idea. In another world, maybe, only the taste matters. If Nishikawa-san succeeds, I'll be happy to make a fine sushi burger for him."

While the heated debate continues, progress is being made towards the birth of practical all-hyperbolic CFD codes. The key to success seems to lie in the taste, not in looks, as Sushiyama implied.

Sushi burger, a radical approach. "Looks eccentric, but it's simple to make and tastes the same or even better."

PDEs Recreated Hyperbolic

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

$$\Downarrow$$

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

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

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

$$\Downarrow$$

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

PDE is made hyperbolic by turning non-hyperbolic terms on the right hand side, including a source term, into a hyperbolic system such that it reduces to the original in the steady state.

"Totally insane. Successful numerical schemes should reflect the nature of the PDE they are solving. Upwind scheme for isotropic diffusion has no chance to work." Although it sounds right, the numerical results shown by Nishikawa indicate that the claim is not true. In fact, unusually good results have been obtained by the upwind scheme for diffusion and viscous flow problems. They are unusually good because high-order accurate gradients have been obtained at a dramatically 'reduced' cost.

These interesting results have attracted a number of researchers around the world. Professor Elliptic at the University of Smooth says, "It is quite nice and welcome. As I see it, the hyperbolized parabolic-PDE is hyperbolic in time but remains elliptic in space. It's just like the acoustic sub-system of the Euler equations, which is hyperbolic in time but elliptic in space in subsonic flows." Dr. Muscl at Monotone National Laboratory (currently under reconstruction) is another researcher who welcomed the declaration. He says, "It's a wonderful news. I feel like I've got a lot more places to work at than I thought."

On the other hand, Always Nolimiter, a graduate student of aerospace engineering, says he's been scared to death since he heard the news. He says, "I'm so scared because the hyperbolic Navier-Stokes equations may generate additional shockwaves due to the hyperbolic viscous term. I just don't know what to do. They're gonna blow me up!" According to Professor Elliptic, however, the student is worrying for nothing. He says, "No shockwaves will be generated by the hyperbolic viscous term. Like I said, they are elliptic in space. There will be no shockwaves running across the domain."

Taro Sushiyama, one of the best sushi chefs in town, commented on the analogy of Sushi Burger repeatedly used by Nishikawa to illustrate the concept. He says, "It looks eccentric. It's against tradition and not acceptable in our world. But it's an interesting idea. In another world, maybe, only the taste matters. If Nishikawa-san succeeds, I'll be happy to make a fine sushi burger for him."

While the heated debate continues, progress is being made towards the birth of practical all-hyperbolic CFD codes. The key to success seems to lie in the taste, not in looks, as Sushiyama implied.

Sushi burger, a radical approach. "Looks eccentric, but it's simple to make and tastes the same or even better."

Foods Recreated as Burgers

Simple, Efficient, Accurate.

Looks eccentric? But the taste is the same, or even better.

Foods Recreated as Burgers

Simple, Efficient, Accurate.

Looks eccentric? But the taste is the same, or even better.

Hyperbolized Diffusion

- Discretization made simple (e.g., upwind for diffusion)
- 1st-order diffusion scheme (e.g., P0 DG)
- Consistent Jacobian for implicit diffusion solver
- Higher-order accurate gradients on irregular grids
- O(1/h) acceleration in convergence

Hyperbolized Diffusion

- Discretization made simple (e.g., upwind for diffusion)
- 1st-order diffusion scheme (e.g., P0 DG)
- Consistent Jacobian for implicit diffusion solver
- Higher-order accurate gradients on irregular grids
- O(1/h) acceleration in convergence

See AIAA2013-1125 for details.

Hyperbolic Advection-Diffusion

Sushi Burger for Advection-Diffusion

$$
\begin{aligned}
\partial_t u + a\,\partial_x u + b\,\partial_y u &= \nu\,(\partial_x p + \partial_y q) \\
\partial_t p &= (\partial_x u - p)/T_r \\
\partial_t q &= (\partial_y u - q)/T_r
\end{aligned}
$$

1. Hyperbolic in time (JCP2010)
2. Equivalent to the advection-diffusion equation in steady state.

$$a\,\partial_x u + b\,\partial_y u = \nu\,(\partial_{xx} u + \partial_{yy} u)$$

3. Relaxation time Tr is a free parameter (no stiff source).
4. Stiffness due to second derivatives eliminated: O(1/h) Jacobian
5. Equal order of accuracy for solution and gradients

Hyperbolic Advection-Diffusion

Sushi Burger for Advection-Diffusion

Advection — Hyperbolic Diffusion

$$
\begin{aligned}
\partial_t u + \boxed{a\,\partial_x u + b\,\partial_y u} &= \boxed{\nu\left(\partial_x p + \partial_y q\right)} \\
\partial_t p &= \boxed{\left(\partial_x u - p\right)/T_r} \\
\partial_t q &= \boxed{\left(\partial_y u - q\right)/T_r}
\end{aligned}
$$

1. Hyperbolic in time (JCP2010)
2. Equivalent to the advection-diffusion equation in steady state.

$$a\,\partial_x u + b\,\partial_y u = \nu\left(\partial_{xx} u + \partial_{yy} u\right)$$

3. Relaxation time Tr is a free parameter (no stiff source).
4. Stiffness due to second derivatives eliminated: O(1/h) Jacobian
5. Equal order of accuracy for solution and gradients

Fully Hyperbolic Advection-Diffusion

Advection + Fully Hyperbolic Diffusion (AIAA2013-1125)

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} + \partial_y \mathbf{G} = \mathbf{0}$$

Normal Flux:

$$\mathbf{H}_n = \mathbf{F} n_x + \mathbf{G} n_y = \mathbf{H}_n^a + \mathbf{H}_n^d + \mathbf{H}_n^s$$

$$= \begin{bmatrix} a_n u \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\nu(p n_x + q n_y) \\ -u n_x / T_r \\ -u n_y / T_r \end{bmatrix} + \begin{bmatrix} 0 \\ (y - y_j)(q n_x - p n_y)/T_r \\ -(x - x_j)(q n_x - p n_y)/T_r \end{bmatrix}$$

where $a_n = a n_x + b n_y \qquad T_r = \dfrac{L_r^2}{\nu}, \quad L_r = \dfrac{1}{2\pi}$ (AIAA2013-1125)

NOTE: It reduces to the scalar advection as $\nu \to 0$.

Fully Hyperbolic Advection-Diffusion

Advection + Fully Hyperbolic Diffusion (AIAA2013-1125)

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} + \partial_y \mathbf{G} = \mathbf{0}$$

Normal Flux:

$$\mathbf{H}_n = \mathbf{F} n_x + \mathbf{G} n_y = \mathbf{H}_n^a + \mathbf{H}_n^d + \mathbf{H}_n^s$$

$$= \begin{bmatrix} a_n u \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\nu(p n_x + q n_y) \\ -u n_x / T_r \\ -u n_y / T_r \end{bmatrix} + \begin{bmatrix} 0 \\ (y - y_j)(q n_x - p n_y)/T_r \\ -(x - x_j)(q n_x - p n_y)/T_r \end{bmatrix}$$

where $a_n = a n_x + b n_y \qquad T_r = \dfrac{L_r^2}{\nu}, \quad L_r = \dfrac{1}{2\pi}$ (AIAA2013-1125)

NOTE: It reduces to the scalar advection as $\nu \to 0$.

Upwind flux constructed for each independently.

Upwind Flux for All

Define the numerical flux as a sum of upwind fluxes:

$$\Phi_{jk} = \Phi^a_{jk} + \Phi^d_{jk} + \Phi^s_{jk}$$

Upwind Advection $$\Phi^a_{jk} = \frac{1}{2}(\mathbf{H}^a_{nL} + \mathbf{H}^a_{nR}) - \frac{1}{2}|\mathbf{A}^a_n|(\mathbf{U}_R - \mathbf{U}_L)$$

Upwind Diffusion $$\Phi^d_{jk} = \frac{1}{2}(\mathbf{H}^d_{nL} + \mathbf{H}^d_{nR}) - \frac{1}{2}|\mathbf{A}^d_n|(\mathbf{U}_R - \mathbf{U}_L)$$

Upwind Source $$\Phi^s_{jk} = \frac{1}{2}(\mathbf{H}^s_{nL} + \mathbf{H}^s_{nR}) - \frac{1}{2}|\mathbf{A}^s_n|(\mathbf{U}_R - \mathbf{U}_L)$$

Upwind Flux for All

Define the numerical flux as a sum of upwind fluxes:

$$\Phi_{jk} = \Phi^a_{jk} + \Phi^d_{jk} + \Phi^s_{jk}$$

Upwind Advection $$\Phi^a_{jk} = \frac{1}{2}(\mathbf{H}^a_{nL} + \mathbf{H}^a_{nR}) - \frac{1}{2}|\mathbf{A}^a_n|(\mathbf{U}_R - \mathbf{U}_L)$$

Upwind Diffusion $$\Phi^d_{jk} = \frac{1}{2}(\mathbf{H}^d_{nL} + \mathbf{H}^d_{nR}) - \frac{1}{2}|\mathbf{A}^d_n|(\mathbf{U}_R - \mathbf{U}_L)$$

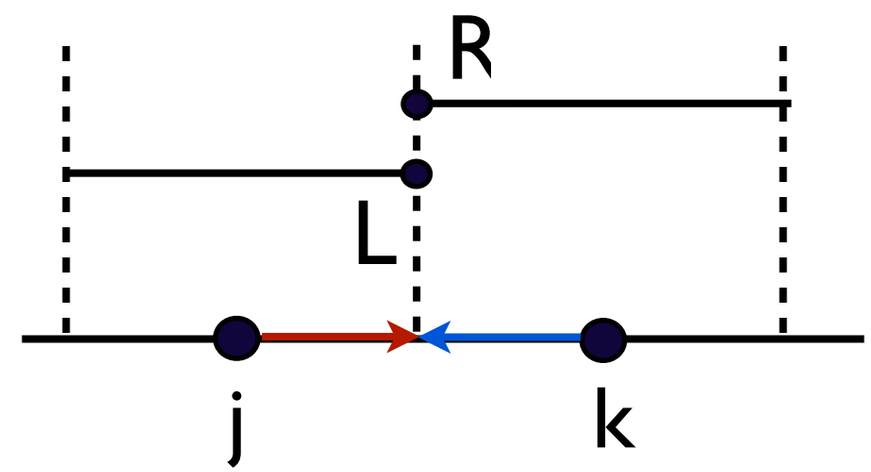
Upwind Source $$\Phi^s_{jk} = \frac{1}{2}(\mathbf{H}^s_{nL} + \mathbf{H}^s_{nR}) - \frac{1}{2}|\mathbf{A}^s_n|(\mathbf{U}_R - \mathbf{U}_L)$$

Same scheme, same truncation error.
The compatibility problem doesn't exist.

First-Order Scheme

Left and right states:

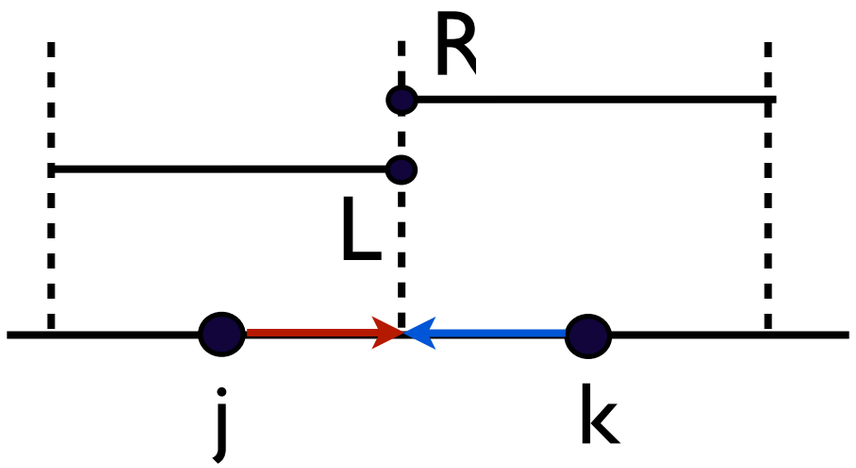
$$\mathbf{U}_L = \mathbf{U}_j, \quad \mathbf{U}_R = \mathbf{U}_k$$



First-Order Scheme

Left and right states:

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



Schemell: Reconstruct the solution by using p and q.

(AIAA 2013-1125)

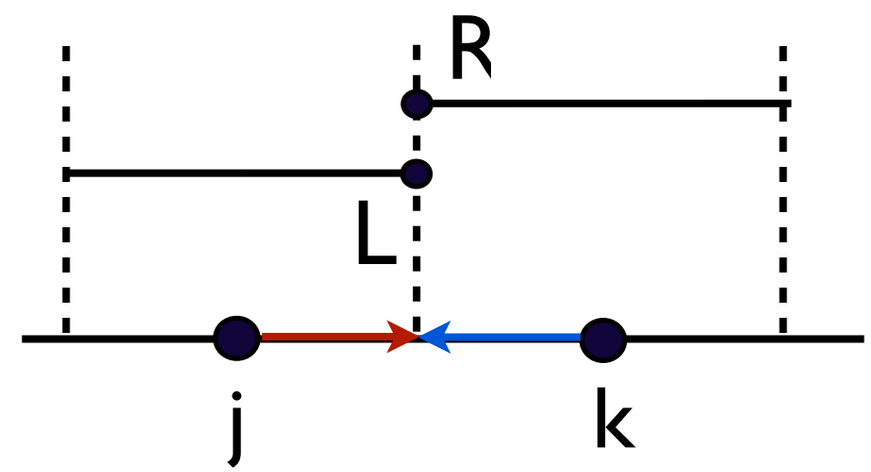
$$u_L = u_j + \frac{1}{2}\left(p_j, q_j\right) \cdot \Delta \mathbf{l}_{jk}, \qquad u_R = u_k - \frac{1}{2}\left(p_k, q_k\right) \cdot \Delta \mathbf{l}_{jk},$$

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

First-Order Scheme

Left and right states:

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



Schemell: Reconstruct the solution by using p and q.

(AIAA 2013-1125)

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

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

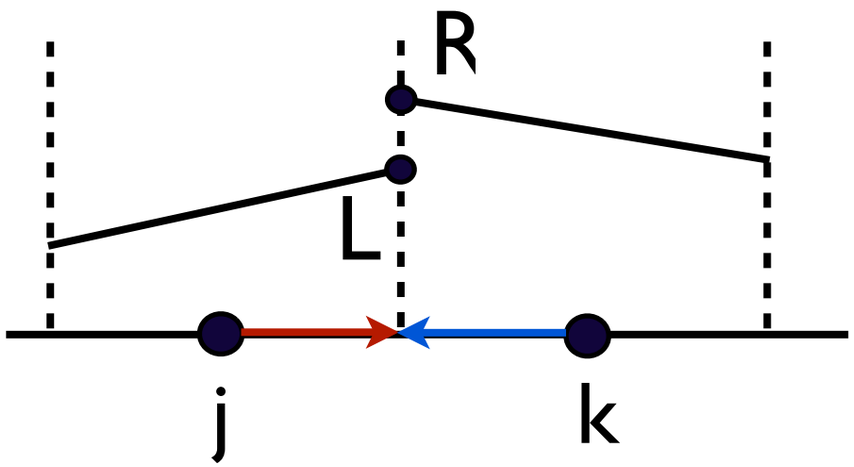
Stencil remains compact.

First-order, but second-order in the advection limit.

Second-Order Scheme

For triangular/tetrahedral and smooth mixed grids.

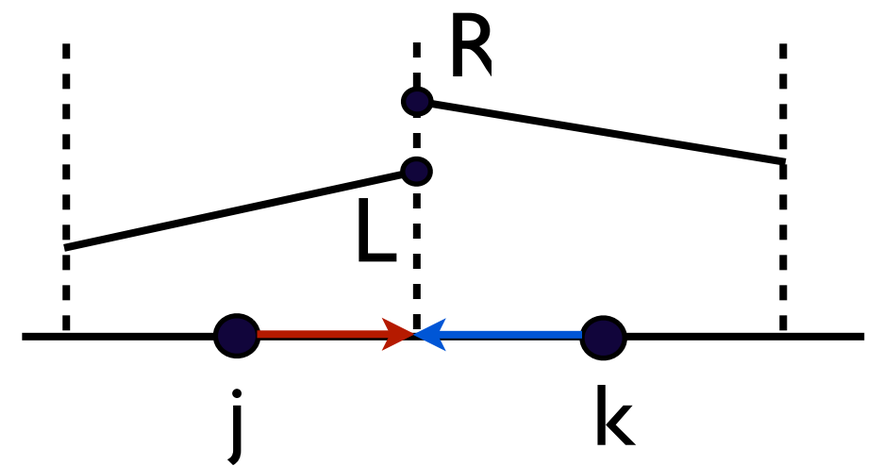
1. Compute gradients at nodes (e.g., LSQ).
2. Extrapolate the solution to the midpoint.



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:

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

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

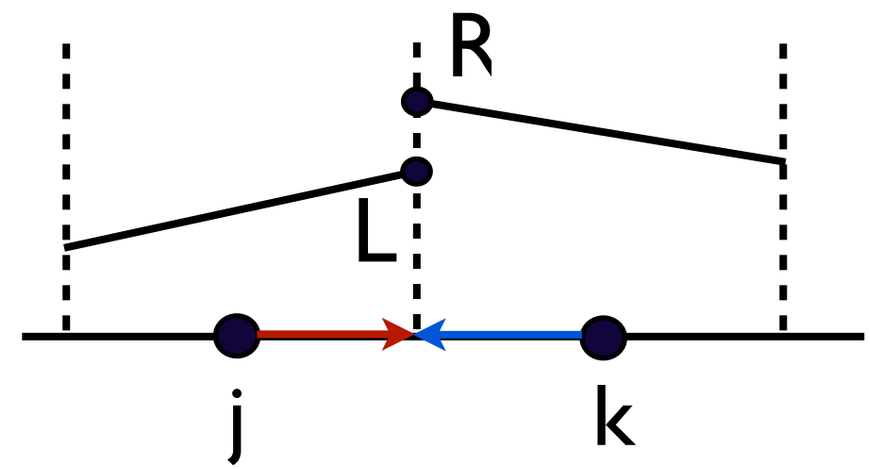
$$q_L = q_j + \frac{1}{2}\nabla q_j \cdot \Delta \mathbf{l}_{jk}, \quad q_R = q_k - \frac{1}{2}\nabla q_k \cdot \Delta \mathbf{l}_{jk}$$

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

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:

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

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

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

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

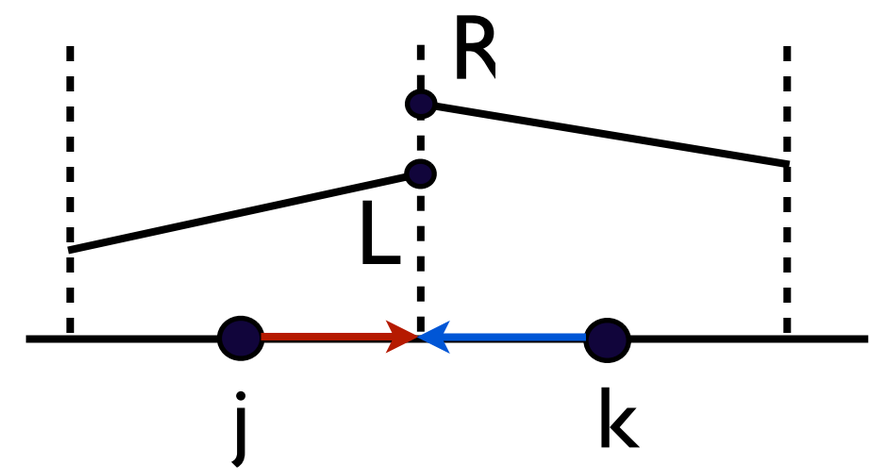
Second-order, but third-order in the advection limit.

(Katz and Sankaran JCP2011)

Third-Order Scheme

For triangular/tetrahedral grids.

1. Quadratic LSQ gradients at nodes.
2. Extrapolate the solution to the midpoint.

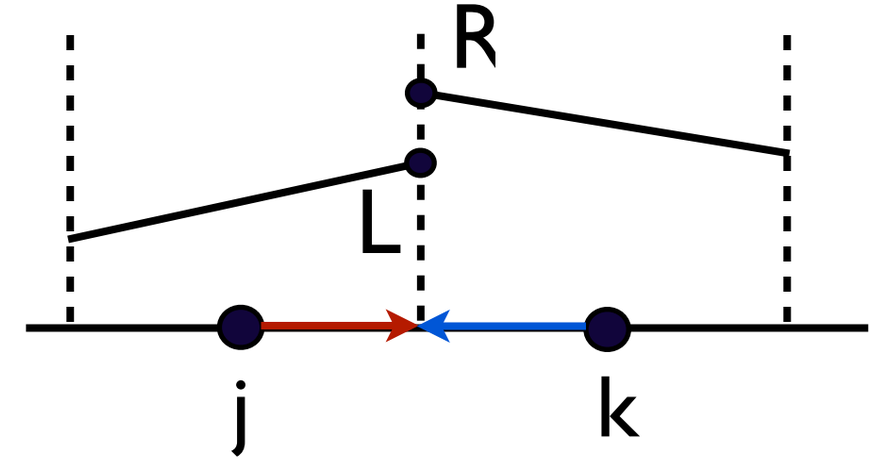


(Katz and Sankaran JCP2011)

Third-Order Scheme

For triangular/tetrahedral grids.

1. Quadratic LSQ gradients at nodes.
2. Extrapolate the solution to the midpoint.



Left and right states:

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

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

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

Uniformly third-order accurate for u, p, and q.

Truncation Error

3rd-order scheme

$$a\,\partial_x u + b\,\partial_y u - \nu\,(\partial_x p + \partial_y q) = 0,\;\; \partial_x u - p = 0,\;\; \partial_y u - q = 0$$

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

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

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

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

Truncation Error

3rd-order scheme

$$a\,\partial_x u + b\,\partial_y u - \nu\,(\partial_x p + \partial_y q) = 0,\;\; \partial_x u - p = 0,\;\; \partial_y u - q = 0$$

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

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

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

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

Truncation Error

3rd-order scheme

$$a\,\partial_x u + b\,\partial_y u - \nu\,(\partial_x p + \partial_y q) = 0,\;\; \partial_x u - p = 0,\;\; \partial_y u - q = 0$$

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

$$+\boxed{\frac{h^2}{12}\left[\partial_{xx}r + \partial_{xy}r + \partial_{yy}r\right]} + O(h^3)$$

$$r = a\,\partial_x u + b\,\partial_y u - \nu(\partial_x p + \partial_y q)$$

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

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

Truncation Error

3rd-order scheme

$$a\,\partial_x u + b\,\partial_y u - \nu\,(\partial_x p + \partial_y q) = 0,\;\; \partial_x u - p = 0,\;\; \partial_y u - q = 0$$

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

$$+\boxed{\frac{h^2}{12}\left[\partial_{xx}r + \partial_{xy}r + \partial_{yy}r\right]} + O(h^3)$$

$$r = a\,\partial_x u + b\,\partial_y u - \nu(\partial_x p + \partial_y q)$$

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

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

Truncation Error

3rd-order scheme

$$a\,\partial_x u + b\,\partial_y u - \nu\,(\partial_x p + \partial_y q) = 0,\;\; \partial_x u - p = 0,\;\; \partial_y u - q = 0$$

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

$$+\frac{h^2}{12}\left[\partial_{xx}r + \partial_{xy}r + \partial_{yy}r\right] + O(h^3)$$

$$r = a\,\partial_x u + b\,\partial_y u - \nu(\partial_x p + \partial_y q)$$

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

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

Truncation Error

3rd-order scheme

$$a\,\partial_x u + b\,\partial_y u - \nu\,(\partial_x p + \partial_y q) = 0,\;\; \partial_x u - p = 0,\;\; \partial_y u - q = 0$$

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

$$+\frac{h^2}{12}\left[\partial_{xx}r + \partial_{xy}r + \partial_{yy}r\right] + O(h^3)$$

$$r = a\,\partial_x u + b\,\partial_y u - \nu(\partial_x p + \partial_y q)$$

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

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

Of course, uniformly 3rd-order accurate.

Numerical Results

Exact solution (See "I Do Like CFD, VOL.1"):

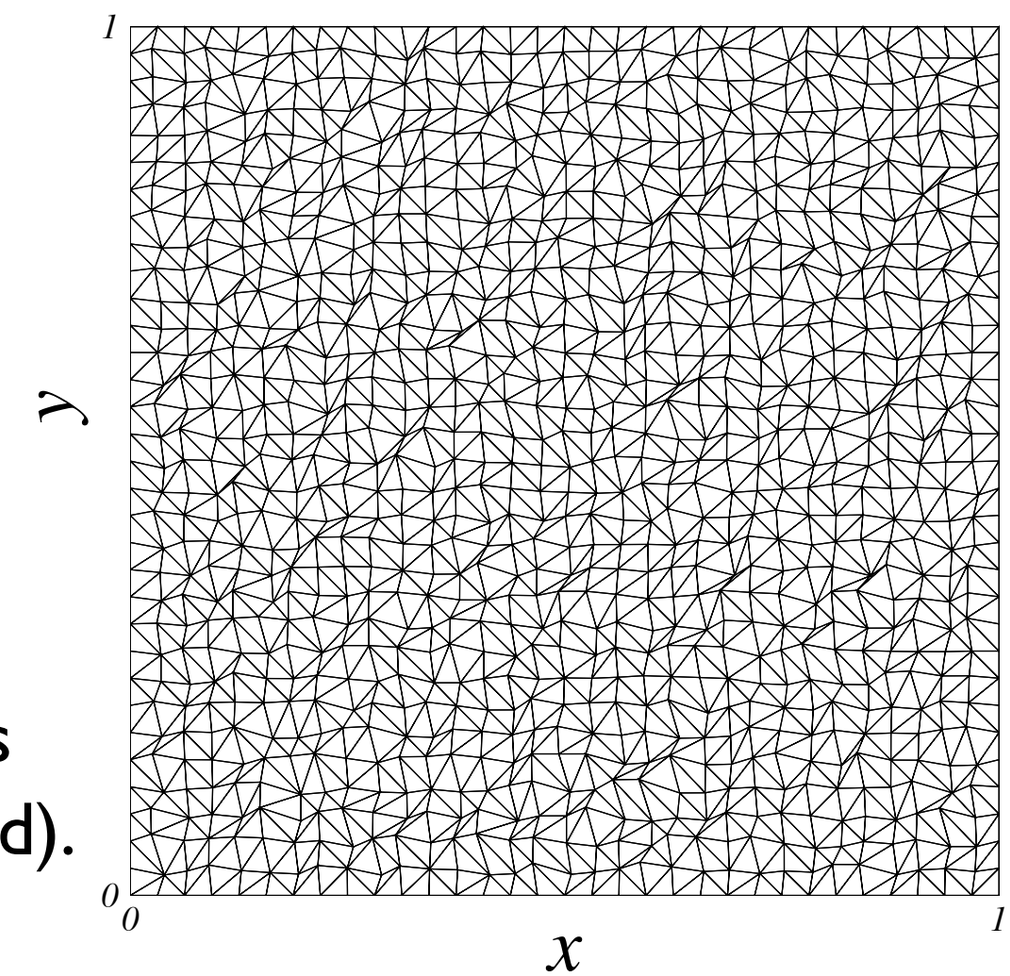
$$u(x,y) = \cos(2\pi\eta)\exp\left(\frac{-2\pi^2\nu}{1+\sqrt{1+4\pi^2\nu^2}}\xi\right) \qquad \xi = ax+by, \quad \eta = bx-ay \quad (a,b)=(1.23,0.12)$$

$$\nu = \frac{\sqrt{a^2+b^2}}{Re}$$

$$Re = 10^{-6}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^6$$

- 8 irregular grids:

  N = n x n n=33, 65, 97, 129, 161, 193, 225, 257
- Dirichlet boundary condition.
- Quadratic fit (full augmentation, two-steps)
- Defect Correction (Implicit solver)
- Converged when residual drops by10 orders
- Compare with the Galerkin and Galerkin(3rd).

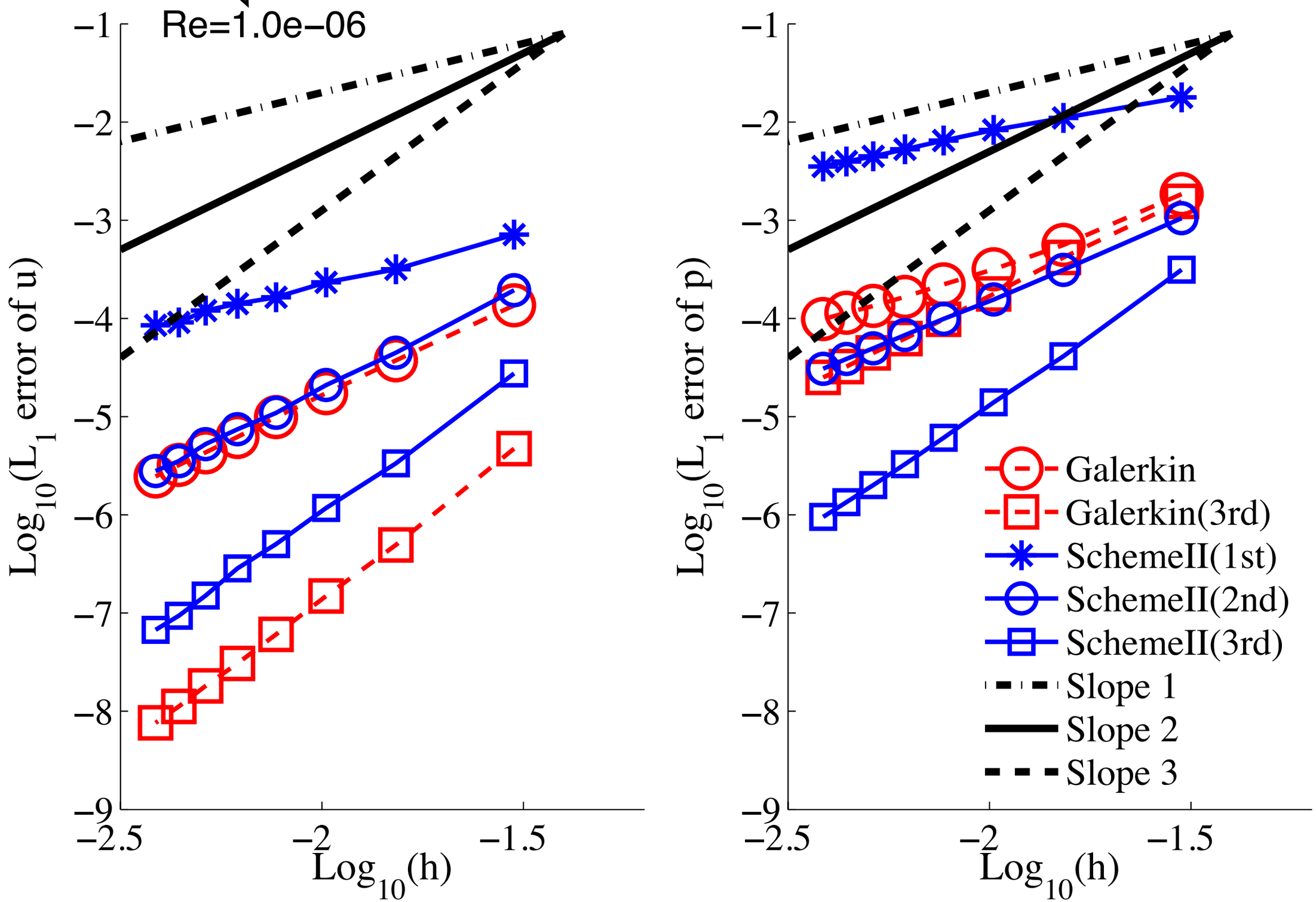


Discretization and Jacobians

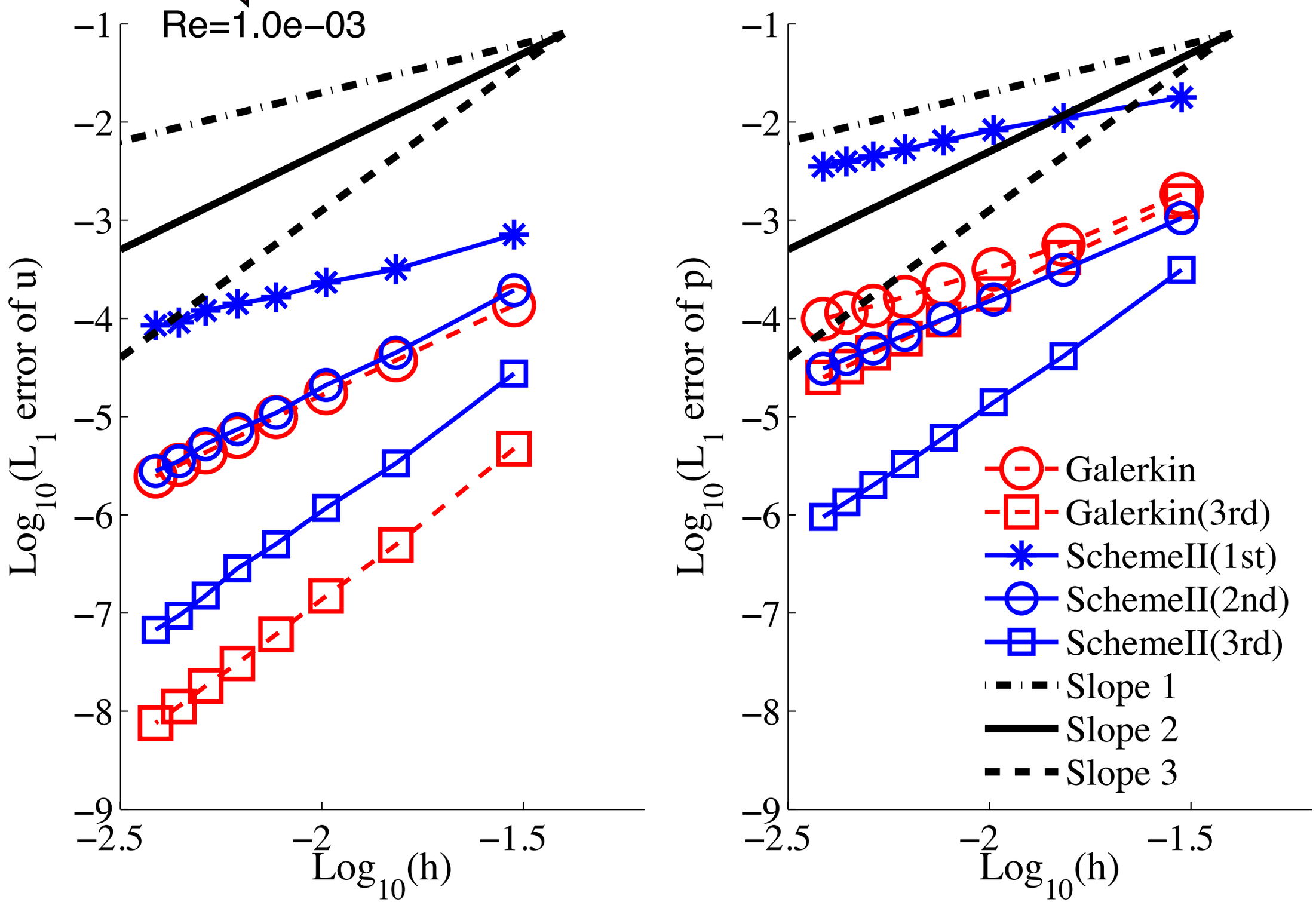
Designation	Discretizaiton		Jacobian	
	Advection	Diffusion	Advection	Diffusion
Galerkin	3rd-order	2nd-order Galerkin	1st-order	Exact
Galerkin(3rd)	3rd-order	3rd-order Galerkin	1st-order	2nd-order
Schemell(1st)	1st-order upwind		Exact	
Schemell(2nd)	2nd-order upwind		1st-order upwind	
Schemell(3rd)	3rd-order upwind		1st-order upwind	

Simple like pure advection schemes.

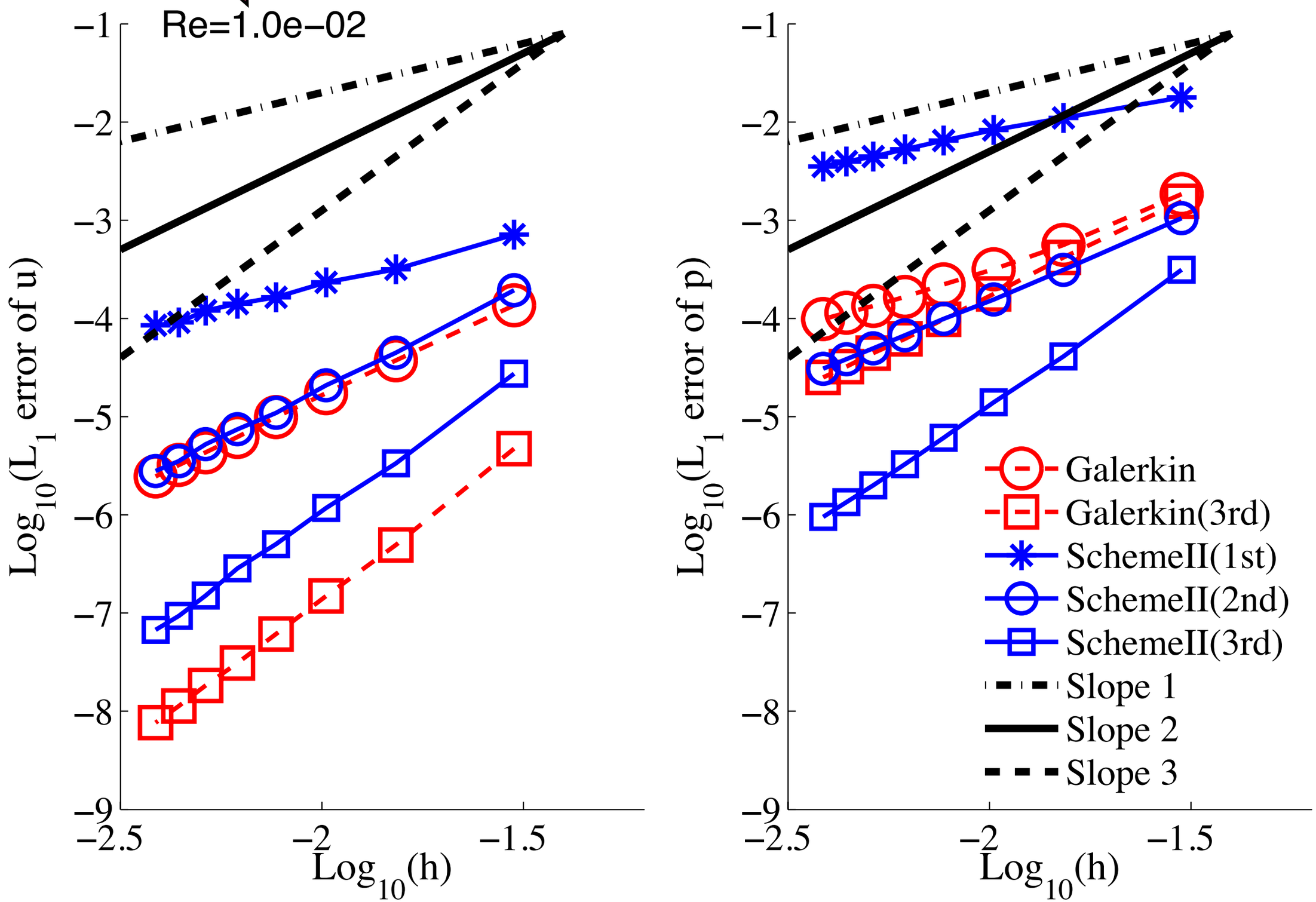
Error Convergence



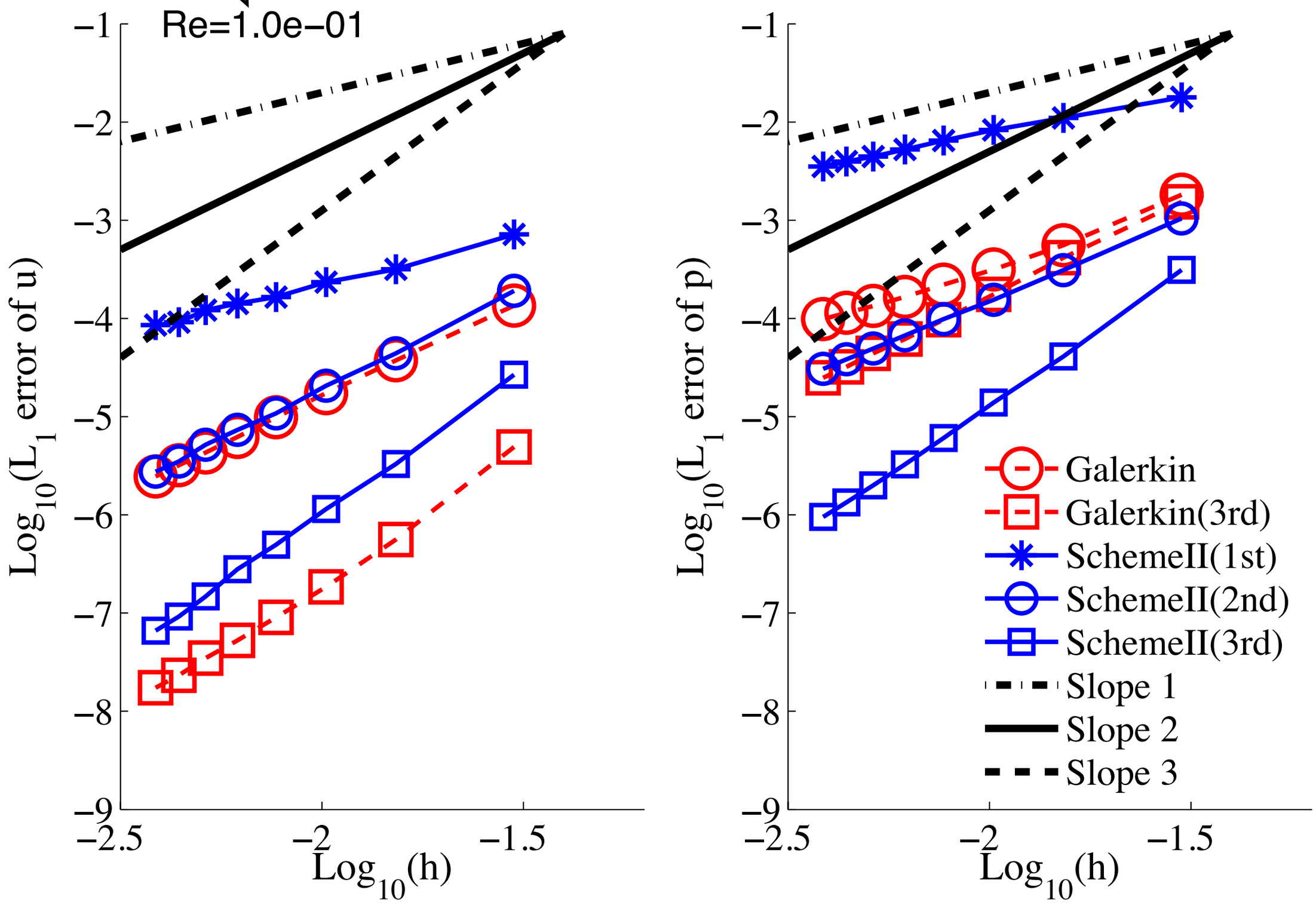
Error Convergence



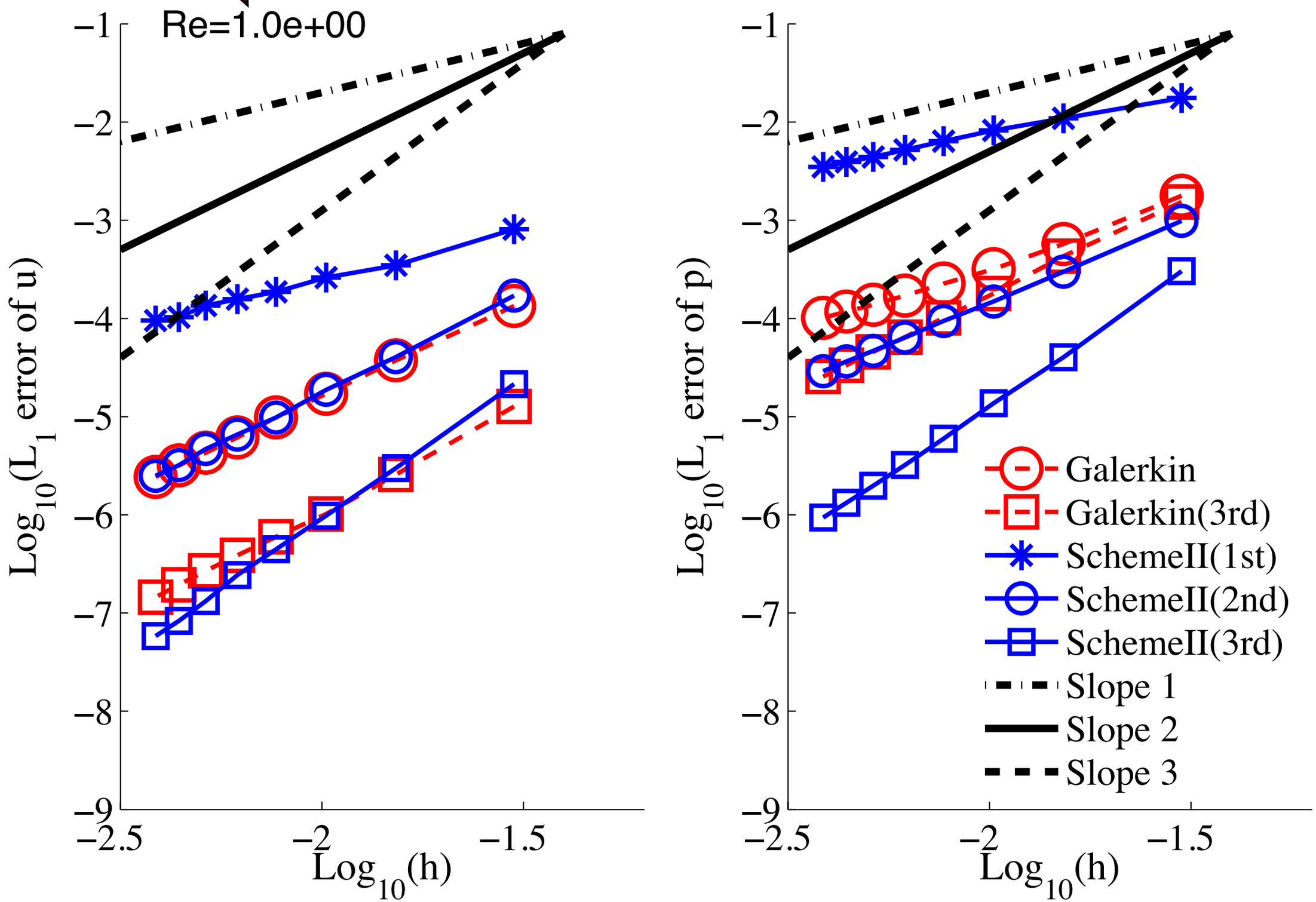
Error Convergence



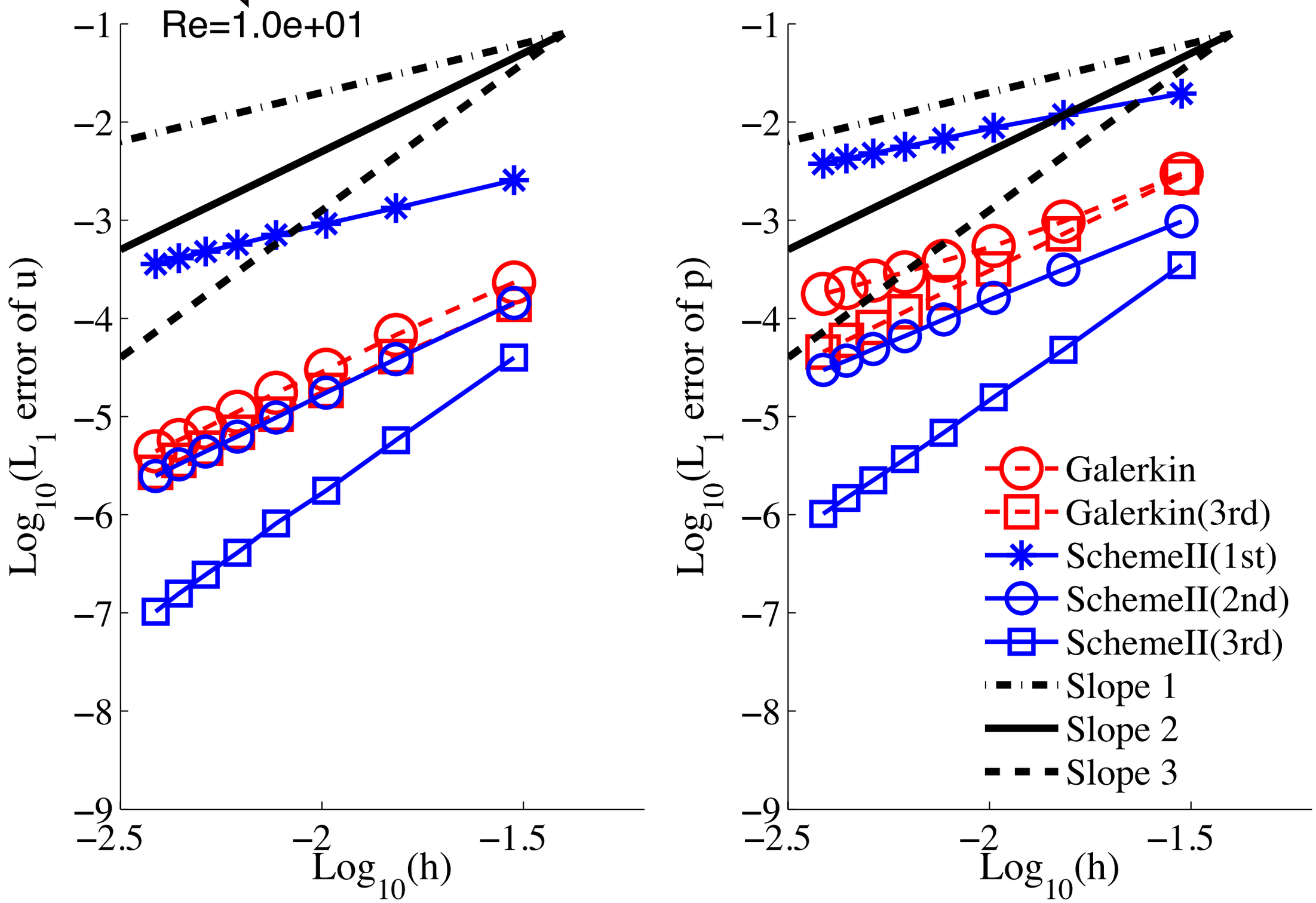
Error Convergence



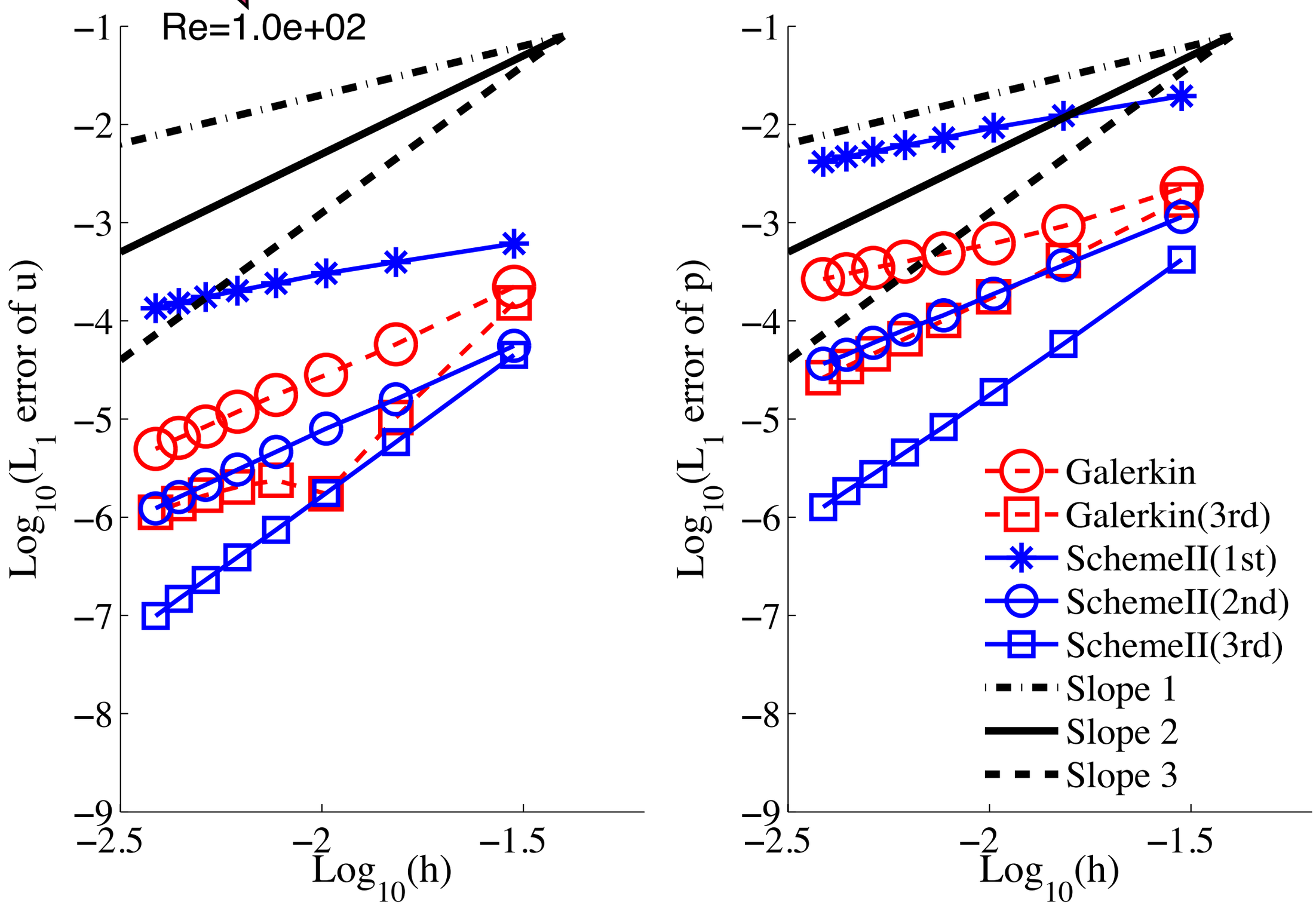
Error Convergence



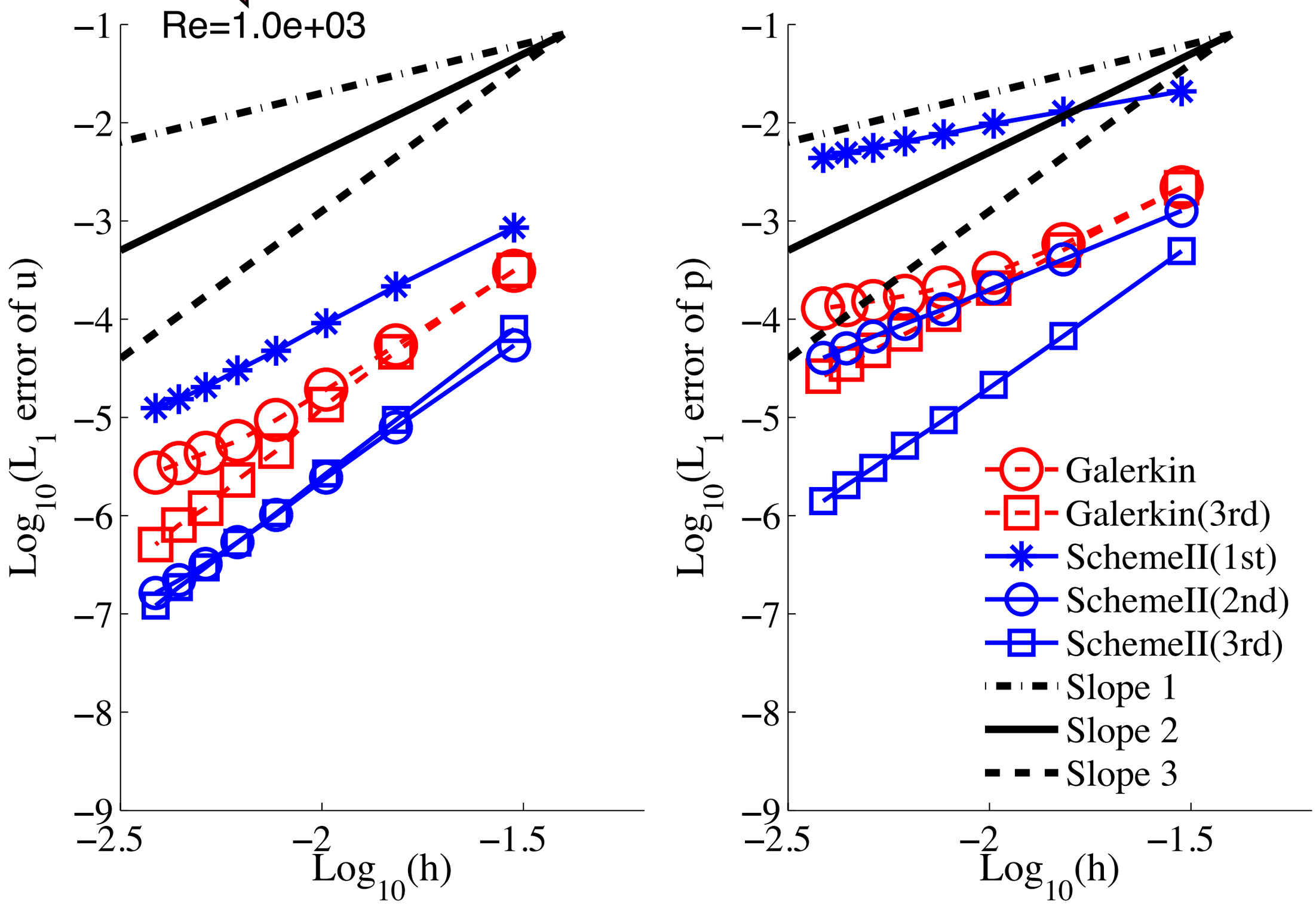
Error Convergence



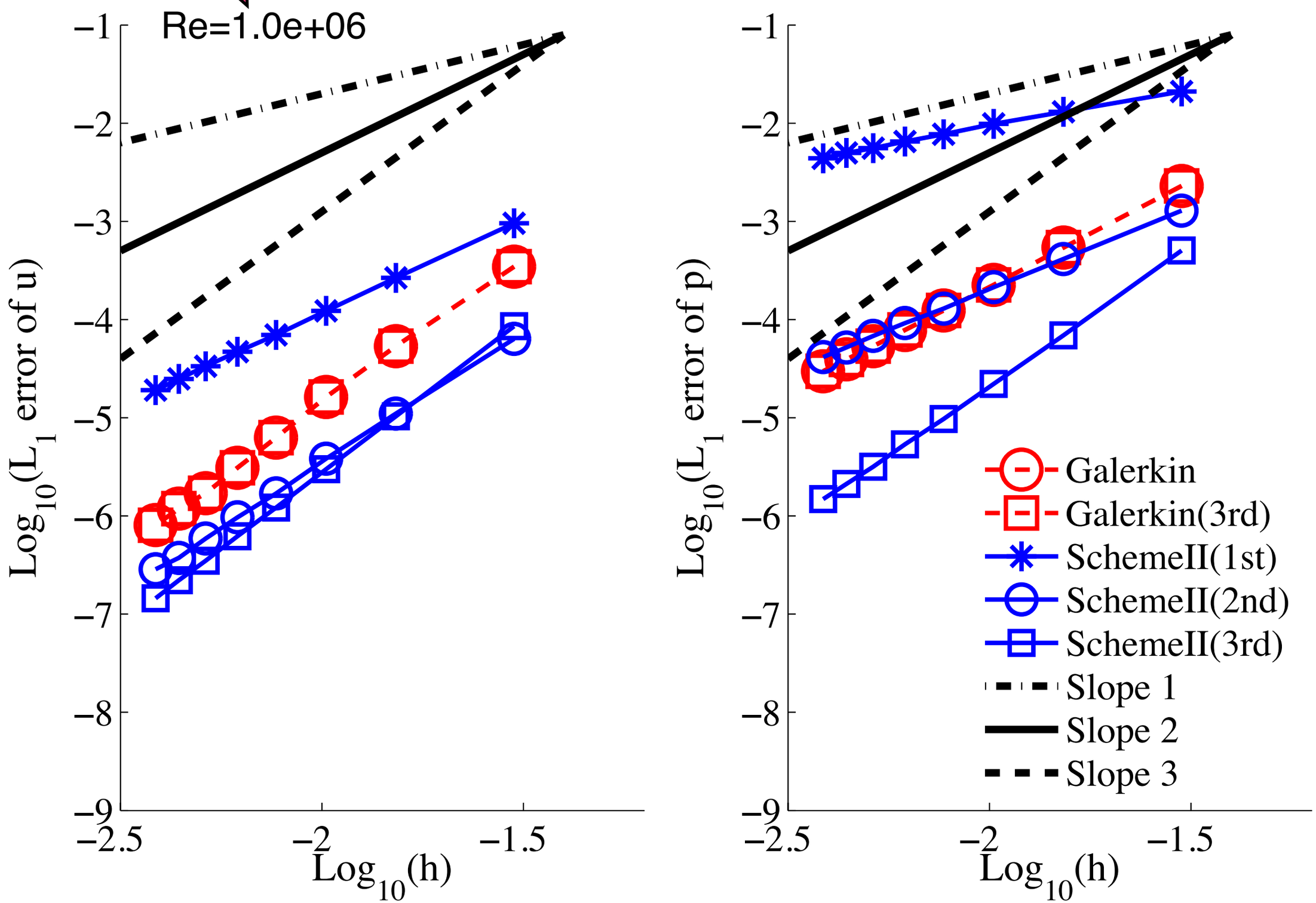
Error Convergence



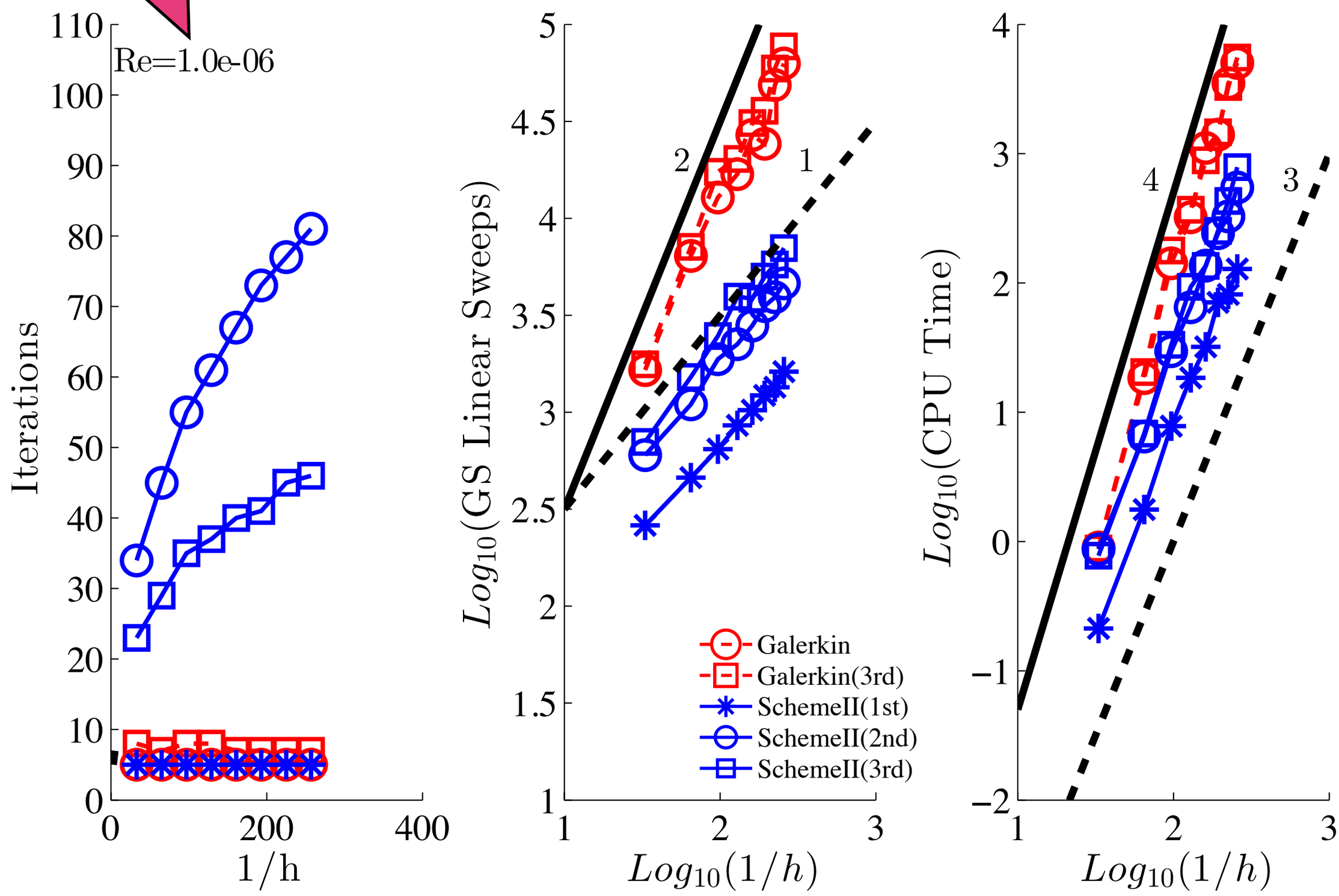
Error Convergence



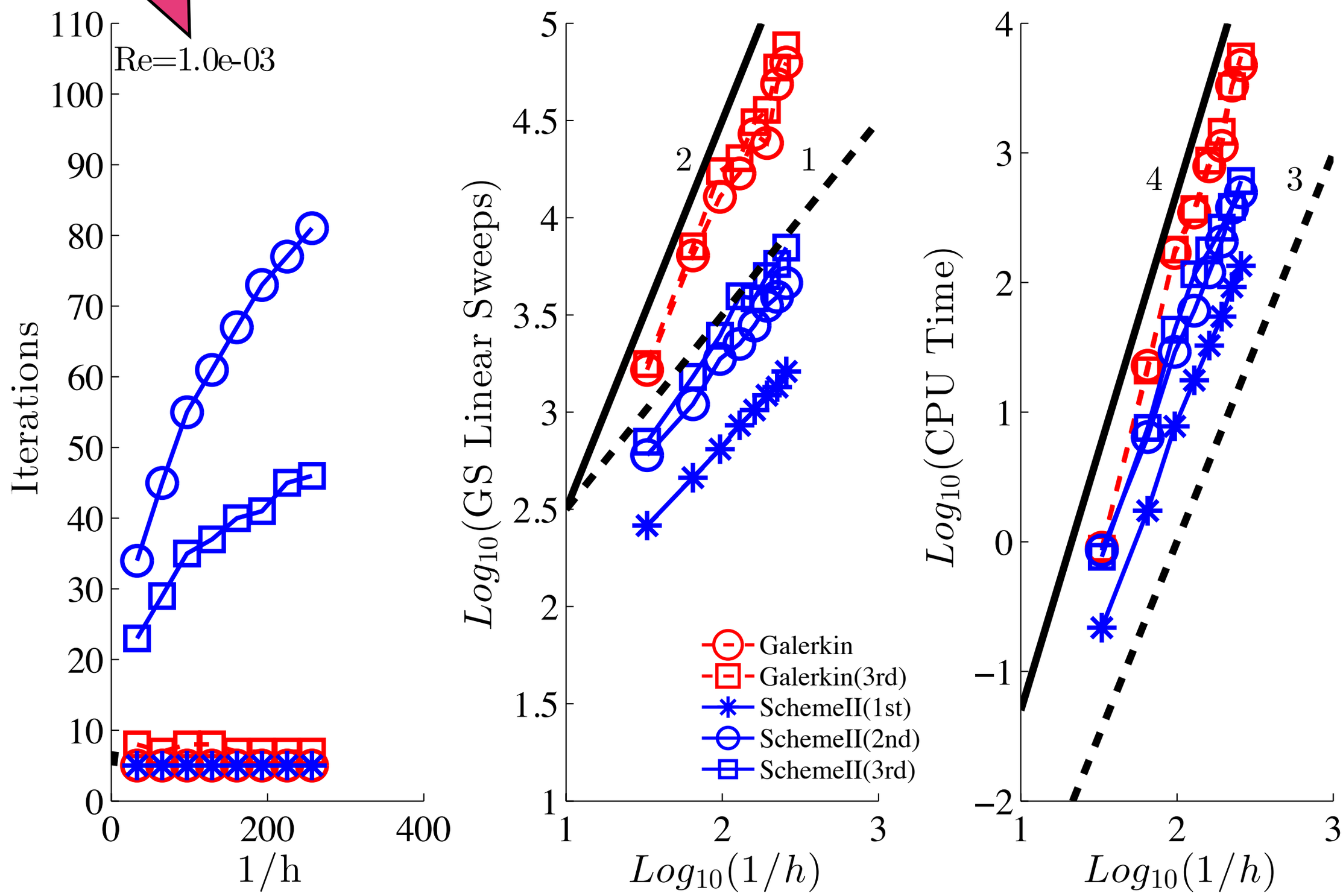
Error Convergence



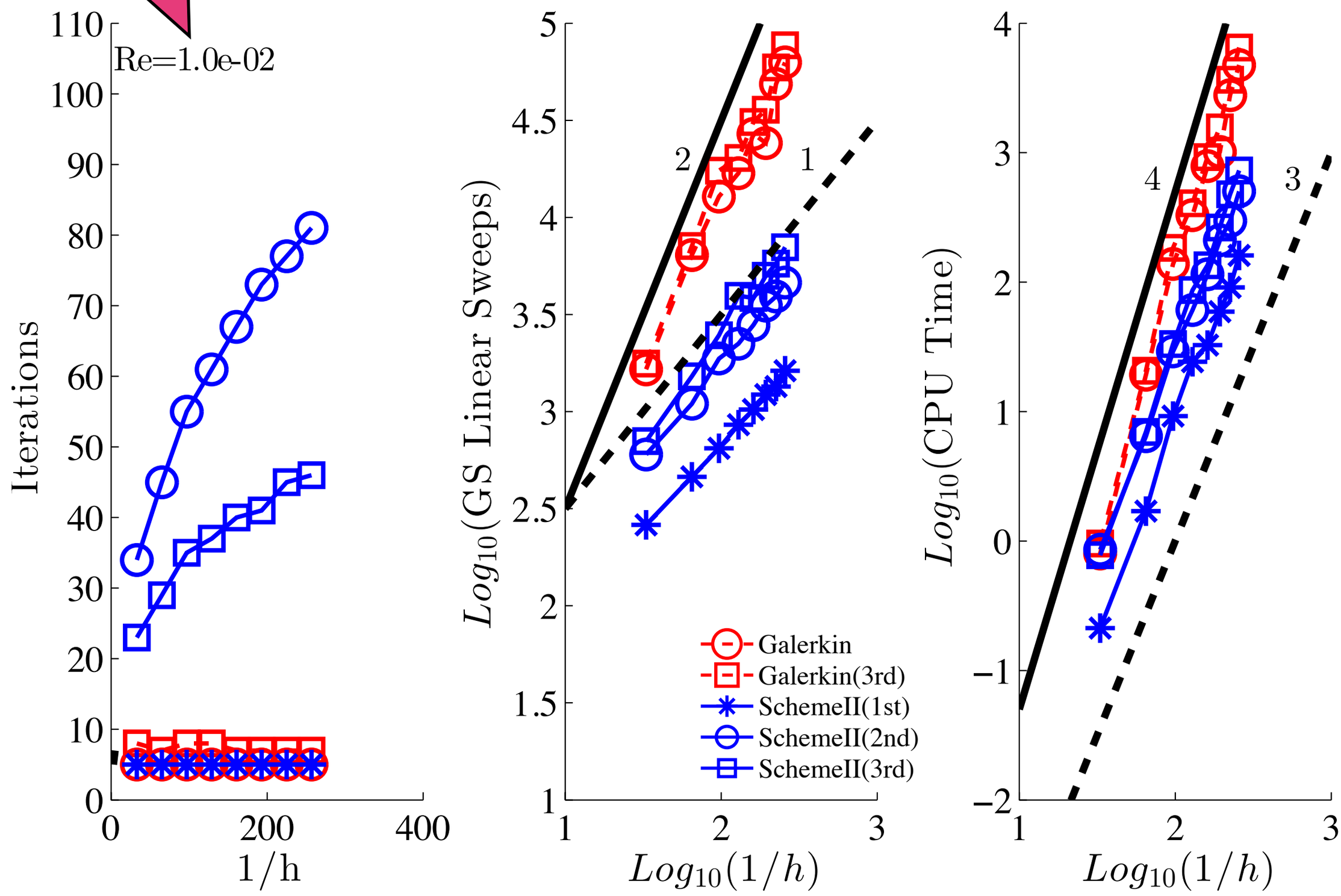
Iterative Convergence



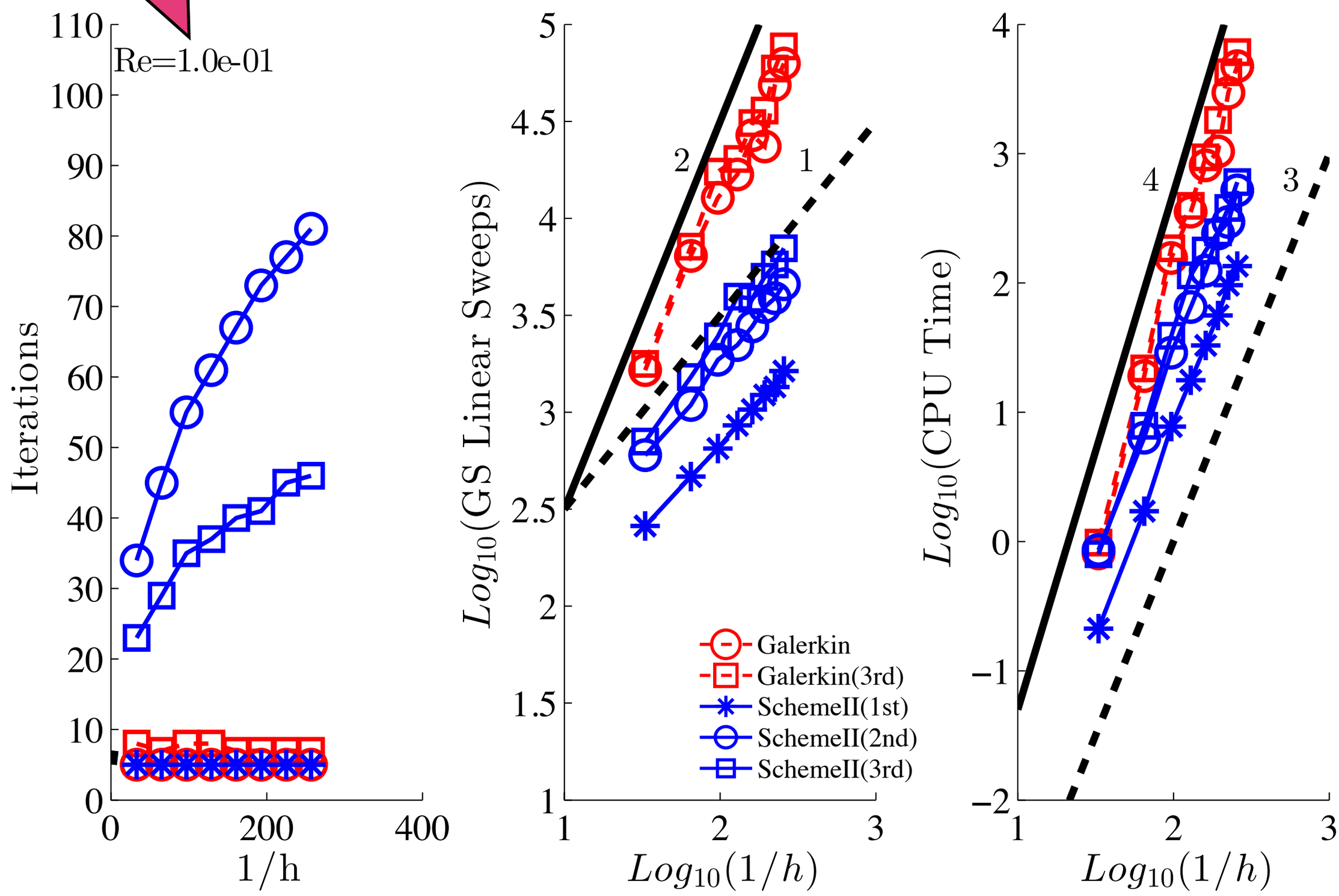
Iterative Convergence



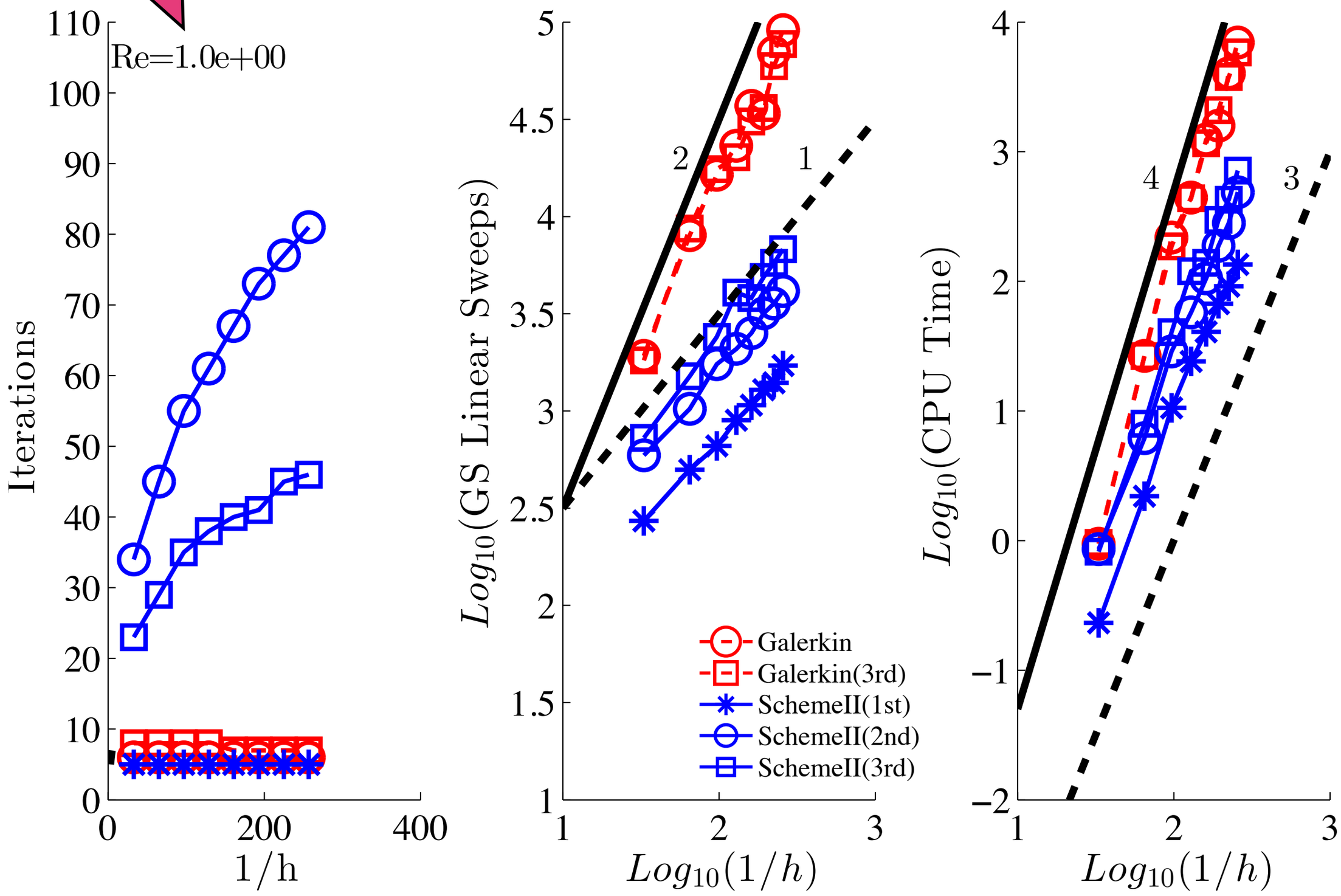
Iterative Convergence



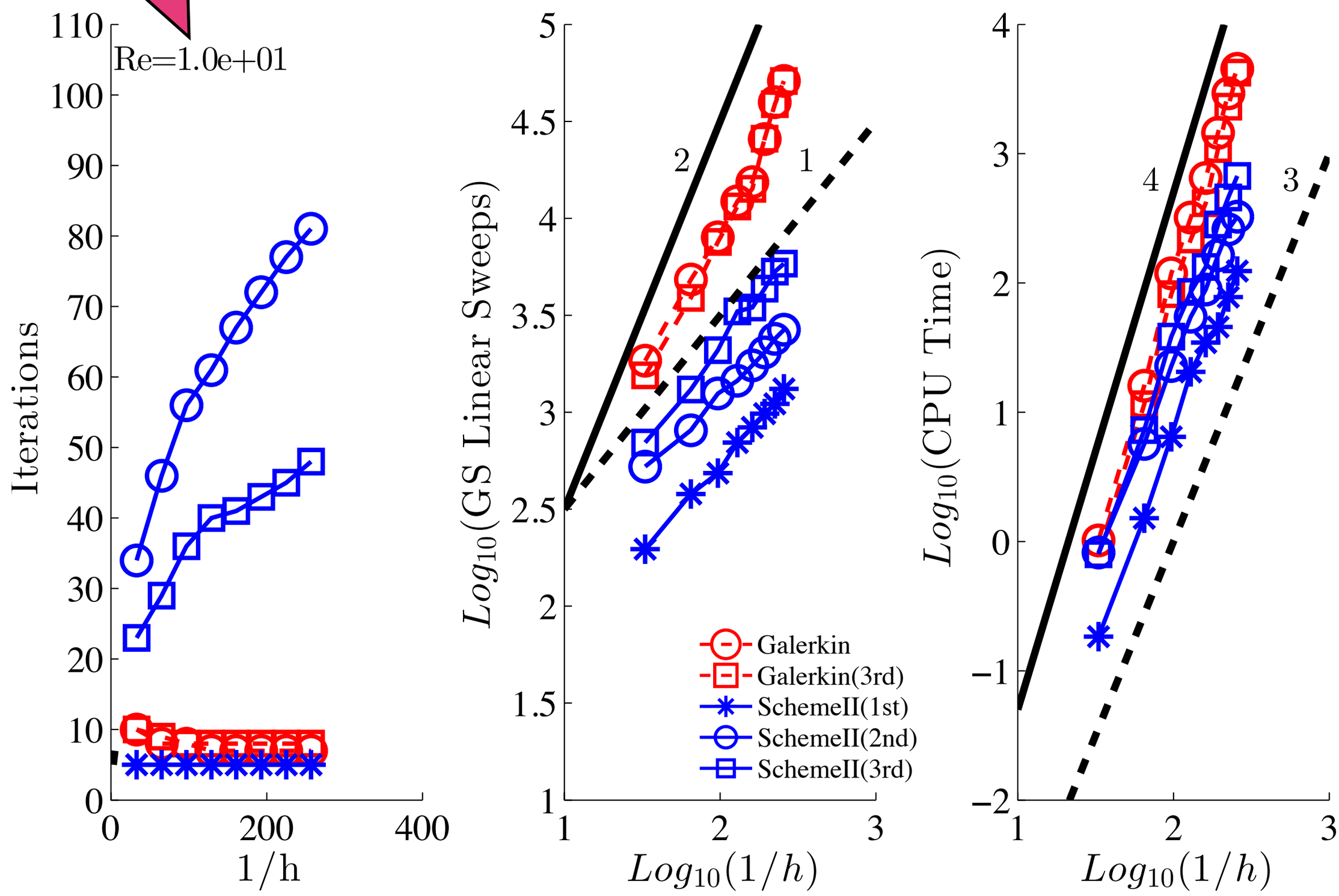
Iterative Convergence



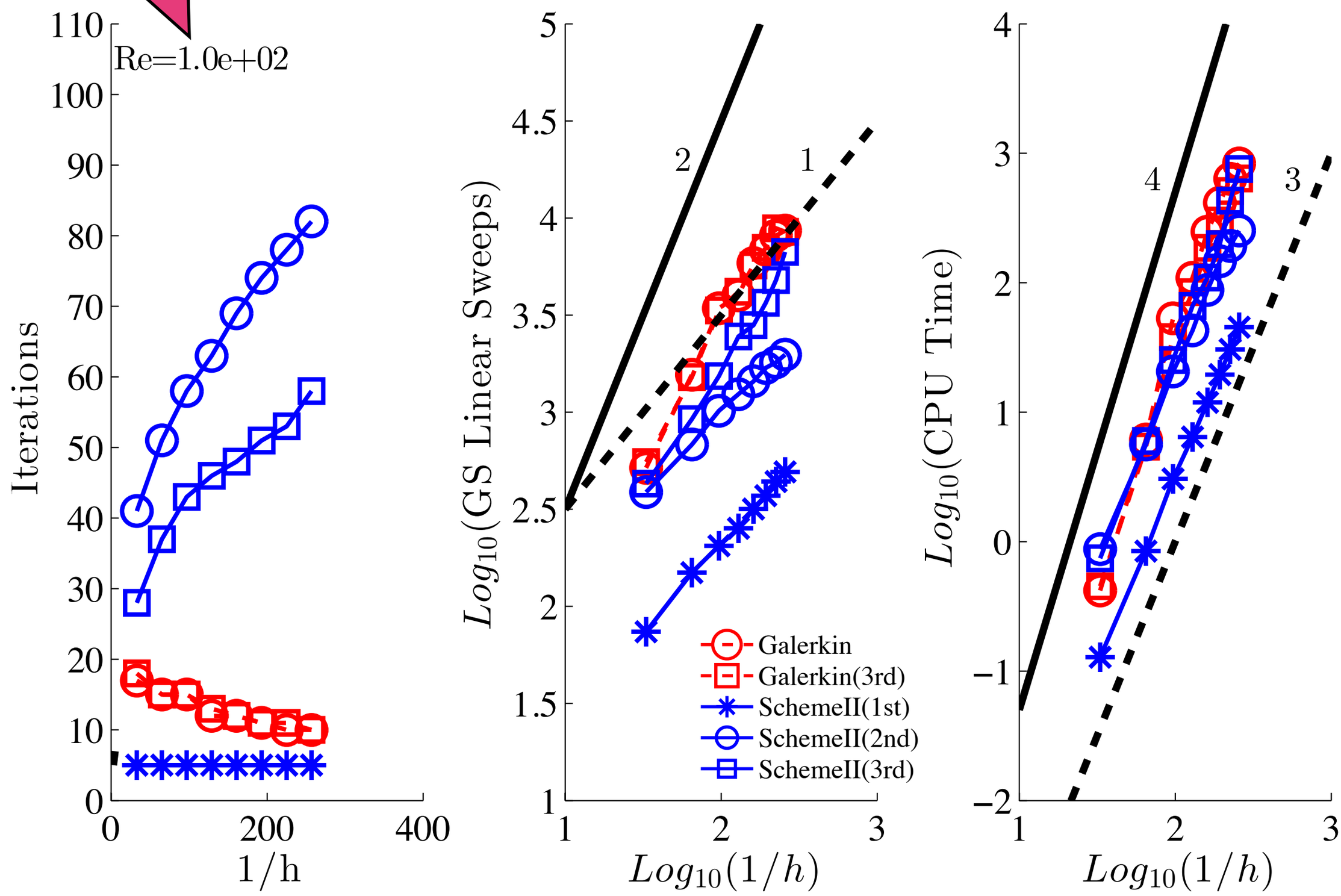
Iterative Convergence



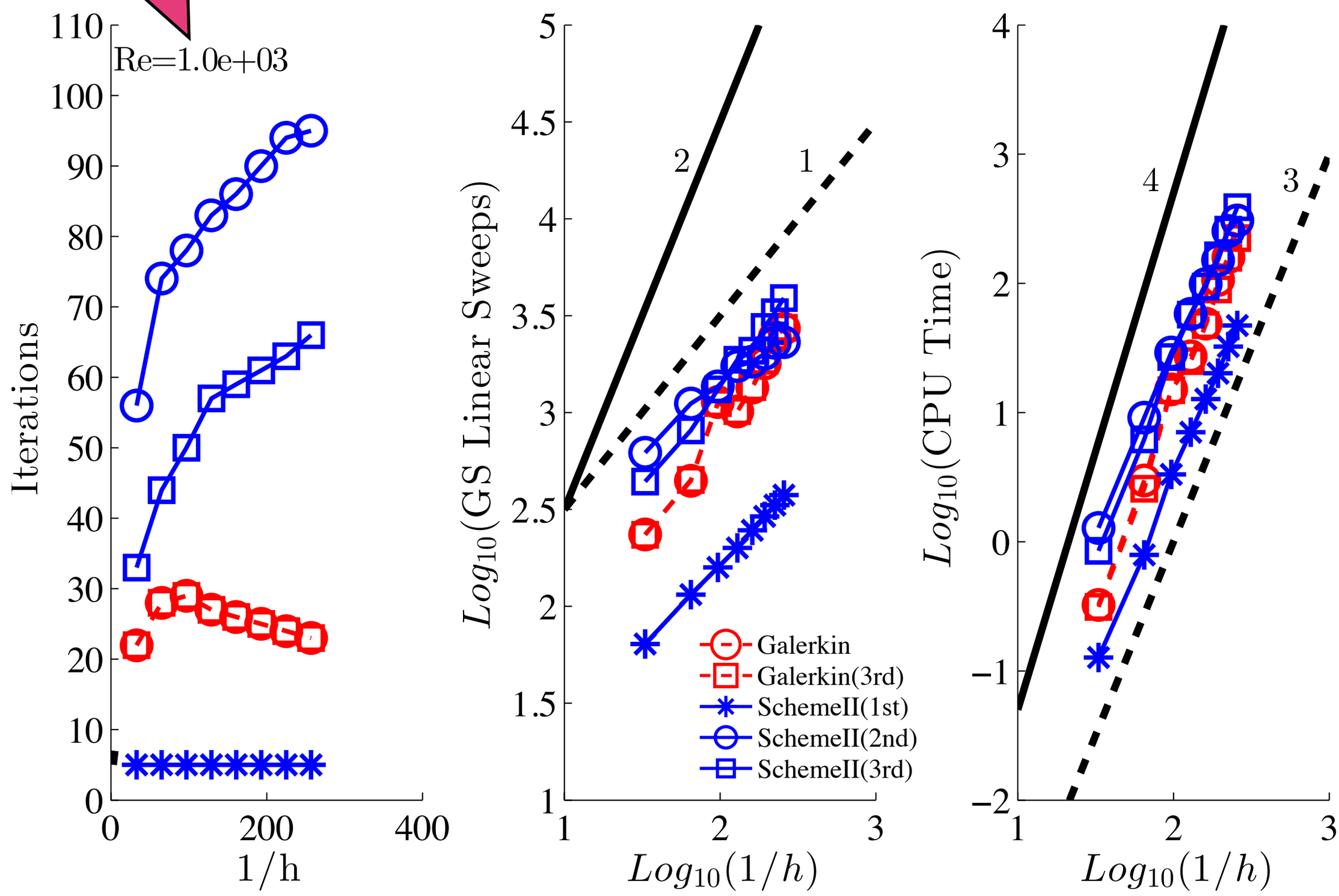
Iterative Convergence



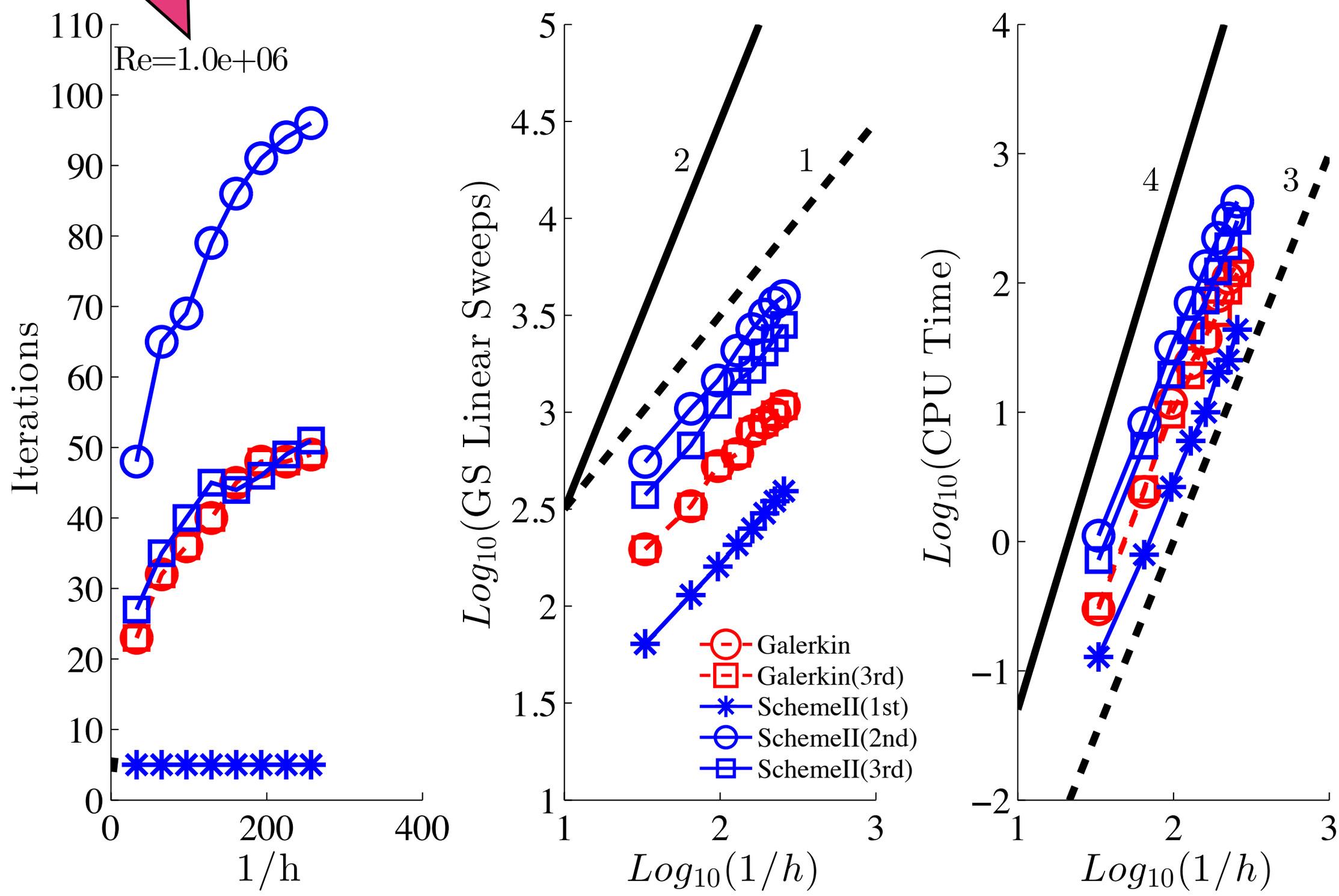
Iterative Convergence



Iterative Convergence



Iterative Convergence



Conclusions

Uniformly accurate implicit upwind FV schemes constructed for the advection-diffusion equation on irregular grids, and verified for a wide range of Reynolds numbers

1st-order scheme: *1st-order accurate solution and gradients*

2nd-order accurate solution in advection limit

compact, no gradient reconstruction, Newton convergence

2nd-order scheme: *2nd-order accurate solution and gradients*

3rd-order accurate solution in advection limit

Linear LSQ gradients

3rd-order scheme: *3rd-order accurate solution and gradients*

nearly at the cost of 2nd-order scheme

Simple, Efficient, Accurate.

Future Work

- *High aspect ratio grids*

- *Time-accurate computations by implicit time-stepping*
(Alireza Mazaheri NASA LaRC)

- *3rd-order hyperbolic Navier-Stokes solver*

Many many many others...

THE CFD NEWS

- FRIDAY, JANUARY 11 2013 -

www.cfdbooks.com THE WORLD'S FAVORITE CFD NEWSPAPER - Since 1910

Hyperbolicity Declared for PDEs

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.

Declaration of hyperbolicity presented by Dr. Hiroaki Nishikawa (NIA) at the 51st AIAA Aerospace Sciences Meeting, January 10, 2013, Grapevine, TX, USA

Hyperbolicity was declared, for the first time in the entire CFD history, for all partial differential equations (PDEs) at 51st AIAA Aerospace Sciences Meeting held in Grapevine, Texas, on January 10, 2013. It was declared totally unexpectedly at the technical talk by Dr. Hiroaki Nishikawa, Senior Research Scientist at National Institute of Aerospace (USA). "I was preparing for this moment since 2007. I thought I had to do it now in order to rescue people suffering from various difficulties with non-hyperbolic PDEs like parabolic PDEs," says Nishikawa.

It seems like the declaration came to rescue, but it has generated heated controversy among researchers.

"That's totally crazy. I'm not going to let anyone make me hyperbolic," says Dr. Parabolic at the Viscous Institute of Technology. He claims, "It is simply wrong because each type of PDEs is designed to model specific physical phenomena and one may not change it no matter what. It is a completely wrong idea." Nishikawa argues that the hyperbolicity is just for the sake of numerically solving the PDEs and at the end of the day the numerical solution satisfies the original equations, parabolic or whatever. Parabolic counters,

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

$$\Downarrow$$

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

PDE is made hyperbolic by turning non-hyperbolic terms on the right hand side, including a source term, into a hyperbolic system such that it reduces to the original in the steady state.

"Totally insane. Successful numerical schemes should reflect the nature of the PDE they are solving. Upwind scheme for isotropic diffusion has no chance to work." Although it sounds right, the numerical results shown by Nishikawa indicate that the claim is not true. In fact, unusually good results have been obtained by the upwind scheme for diffusion and viscous flow problems. They are unusually good because high-order accurate gradients have been obtained at a dramatically 'reduced' cost.

These interesting results have attracted a number of researchers around the world. Professor Elliptic at the University of Smooth says, "It is quite nice and welcome. As I see it, the hyperbolized parabolic-PDE is hyperbolic in time but remains elliptic in space. It's just like the acoustic sub-system of the Euler equations, which is hyperbolic in time but elliptic in space in subsonic flows." Dr. Muscl at Monotone National Laboratory (currently under reconstruction) is another researcher who welcomed the declaration. He says, "It's a wonderful news. I feel like I've got a lot more places to work at than I thought."

On the other hand, Always Nolimiter, a graduate student of aerospace engineering, says he's been scared to death since he heard the news. He says, "I'm so scared because the hyperbolic Navier-Stokes equations may generate additional shockwaves due to the hyperbolic viscous term. I just don't know what to do. They're gonna blow me up!" According to Professor Elliptic, however, the student is worrying for nothing. He says, "No shockwaves will be generated by the hyperbolic viscous term. Like I said, they are elliptic in space. There will be no shockwaves running across the domain."

Taro Sushiyama, one of the best sushi chefs in town, commented on the analogy of Sushi Burger repeatedly used by Nishikawa to illustrate the concept. He says, "It looks eccentric. It's against tradition and not acceptable in our world. But it's an interesting idea. In another world, maybe, only the taste matters. If Nishikawa-san succeeds, I'll be happy to make a fine sushi burger for him."

While the heated debate continues, progress is being made towards the birth of practical all-hyperbolic CFD codes. The key to success seems to lie in the taste, not in looks, as Sushiyama implied.

Sushi burger, a radical approach. "Looks eccentric, but it's simple to make and tastes the same or even better."