High-Order RD Hyperbolic Advection-Diffusion Schemes: 3rd-, 4th-, and 6th-Order

Alireza Mazaheri\textsuperscript{1} and Hiroaki Nishikawa\textsuperscript{2}

\textsuperscript{1}Aerothermodynamics Branch, NASA Langley Research Center
Alireza.Mazaheri@nasa.gov

\textsuperscript{2}National Institute for Aerospace
Hiroaki.Nishikawa@nasa.gov

June 16, 2014
Overview

1. Objectives
2. Basic Formulation
3. Extension to High-Order
4. Some Results
5. Summary
6. Future Work
Objectives

- Develop a Robust, Accurate, and Efficient Viscous Solver
  
  a) Residual Distribution (RD)
     - Compact Second-Order
     - Newton Method
       \( \leq 10 \) sub-iterations for implicit time stepping
  
  b) Hyperbolic System Formulation
     - Compact Viscous Stencil
     - No Second Derivatives
Consider an Advection-Diffusion equation:

$$\partial_t u + a \partial_x u = \nu \partial_{xx} u + \tilde{S}(x)$$

We hyperbolize the equation by setting $p = u_x$ as:

$$\partial_\tau u + a \partial_x u = \nu \partial_x p - \frac{\alpha}{\Delta t} u + S(x)$$

$$\partial_\tau p = (\partial_x u - p)/T_r$$
Consider an Advection-Diffusion equation:

\[ \partial_t u + a \partial_x u = \nu \partial_{xx} u + \tilde{S}(x) \]

We hyperbolize the equation by setting \( p = u_x \) as:

\[ \partial_\tau u + a \partial_x u = \nu \partial_x p - \frac{\alpha}{\Delta t} u + S(x) \]

\[ \partial_\tau p = (\partial_x u - p)/T_r \]
Consider an Advection-Diffusion equation:

\[ \partial_t u + a \partial_x u = \nu \partial_{xx} u + \tilde{S}(x) \]

We hyperbolize the equation by setting \( p = u_x \) as:

- **Advection**

\[
\partial_\tau u + a \partial_x u = \nu \partial_x p - \frac{\alpha}{\Delta t} u + S(x)
\]

\[
\partial_\tau p = \frac{(\partial_x u - p)}{T_r}
\]

- **Hyperbolic Diffusion**
Consider an Advection-Diffusion equation:

\[ \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} + \tilde{S}(x) \]

We hyperbolize the equation by setting \( p = u_x \) as:

- **Advection**

  \[ \frac{\partial}{\partial \tau} u + a \frac{\partial u}{\partial x} = \nu \frac{\partial p}{\partial x} - \frac{\alpha}{\Delta t} u + S(x) \]

- **Hyperbolic Diffusion**

  \[ \frac{\partial}{\partial \tau} p = (\frac{\partial u}{\partial x} - p) / T_r \]

- **Source Terms**
Consider an Advection-Diffusion equation:

$$\partial_t u + a \partial_x u = \nu \partial_{xx} u + \tilde{S}(x)$$

We hyperbolize the equation by setting $p = u_x$ as:

- **Advection**
  $$\partial_\tau u + a \partial_x u = \nu \partial_x p - \frac{\alpha}{\Delta t} u + S(x)$$

- **Hyperbolic Diffusion**
  $$\partial_\tau p = \frac{\partial_x u - p}{T_r}$$

- **Source Terms**

- **Pseudo-Steady-State**: i.e., Solution at the next physical time step
Hyperbolic Advection-Diffusion

With the hyperbolic formulation, we can rewrite our advection-diffusion equation as a first-order system:

\[
\frac{\partial U}{\partial \tau} + A \frac{\partial U}{\partial x} = S
\]

\[
U = \begin{bmatrix} u \\ p \end{bmatrix}, \quad A = \begin{bmatrix} a & -\nu \\ -1/T_r & 0 \end{bmatrix}, \quad S = \begin{bmatrix} -\alpha u/\Delta t + S(x) \\ -p/T_r \end{bmatrix}
\]
Hyperbolic Advection-Diffusion

With the hyperbolic formulation, we can rewrite our advection-diffusion equation as a first-order system:

\[
\frac{\partial U}{\partial \tau} + A \frac{\partial U}{\partial x} = S
\]

\[
U = \begin{bmatrix} u \\ p \end{bmatrix}, \quad A = \begin{bmatrix} a & -\nu \\ -1/T_r & 0 \end{bmatrix}, \quad S = \begin{bmatrix} -\alpha u/\Delta t + S(x) \\ -p/T_r \end{bmatrix}
\]

With two real wave speeds of

\[
\lambda_{1,2} = \frac{a}{2} \left[ 1 \pm \sqrt{1 + \frac{4\nu}{a^2 T_r}} \right]
\]
Hyperbolic Advection-Diffusion

With the hyperbolic formulation, we can rewrite our advection-diffusion equation as a first-order system:

\[
\frac{\partial U}{\partial \tau} + A \frac{\partial U}{\partial x} = S
\]

\[
U = \begin{bmatrix} u \\ p \end{bmatrix}, \quad A = \begin{bmatrix} a & -\nu \\ -1/Tr & 0 \end{bmatrix}, \quad S = \begin{bmatrix} -\alpha u/\Delta t + S(x) \\ -p/Tr \end{bmatrix}
\]

With two real wave speeds of

\[
\lambda_{1,2} = \frac{a}{2} \left[ 1 \pm \sqrt{1 + \frac{4\nu}{a^2 Tr}} \right]
\]

Hyperbolic in Pseudo Time ($Tr$ is a free parameter.)
RD Scheme
RD Scheme: Cell Residuals

We can now evaluate the cell residuals for a general time-dependent hyperbolic advection-diffusion system as

\[
\Phi^C = \int_{x_i}^{x_{i+1}} (-A U_x + S) dx
\]
RD Scheme: Cell Residuals

We can now evaluate the cell residuals for a general time-dependent hyperbolic advection-diffusion system as

\[ \Phi^C = \int_{x_i}^{x_{i+1}} (-A \mathbf{U}_x + \mathbf{S}) \, dx \]

\[ = \begin{cases} 
-a(u_{i+1} - u_i)^{k+1} + \nu(p_{i+1} - p_i)^{k+1} \\
\frac{1}{Tr}(u_{i+1} - u_i)^{k+1} \\
+ \left[ \int_{x_i}^{x_{i+1}} \mathbf{S} \, dx \right]^{k+1,n-1,n}
\end{cases} \]

RD Scheme: Nodal Residuals

Nodal residuals are evaluated by distributing the cell residuals $\Phi^C$ to the nodes:

$$\frac{d\mathbf{U}_i}{d\tau} = \frac{1}{h_i} (\mathbf{B}^+ \Phi^L + \mathbf{B}^- \Phi^R) = \text{Res}_i$$

$$h_i = \frac{h_L + h_R}{2}$$
RD Scheme: Nodal Residuals

Nodal residuals are evaluated by distributing the cell residuals $\Phi^C$ to the nodes:

$$\frac{d\Phi^C_i}{d\tau} = \frac{1}{h_i} \left( B^+ \Phi^L_i + B^- \Phi^R_i \right) = \text{Res}_i$$

$$h_i = \frac{h_L + h_R}{2}$$

Therefore, we solve $\text{Res}_i = 0$. 

$$B^+ \Phi_{i-1} \quad B^- \Phi_i \quad B^+ \Phi_i \quad B^- \Phi_{i+1}$$

$$i-1 \quad i \quad i+1 \quad i+2$$
<table>
<thead>
<tr>
<th>Objectives</th>
<th>Basic Formulation</th>
<th>Extension to High-Order</th>
<th>Some Results</th>
<th>Summary</th>
<th>Future Work</th>
</tr>
</thead>
</table>

**Implicit Solver**
Implicit formulation for $U = (u_1, p_1, u_2, p_2, \ldots, u_N, p_N)$:

$$U^{k+1} = U^k + \Delta U^k$$

The correction, $\Delta U^k = U^{k+1} - U^k$, is determined by:

$$\frac{\partial \text{Res}}{\partial U} \Delta U^k = -\text{Res}^k$$

- Jacobian: Exact for 2nd-order scheme
- Gauss-Seidel Relaxation
Implicit Solver

Implicit formulation for $U = (u_1, p_1, u_2, p_2, \ldots, u_N, p_N)$:

$$U^{k+1} = U^k + \Delta U^k$$

The correction, $\Delta U^k = U^{k+1} - U^k$, is determined by:

$$\frac{\partial \text{Res}}{\partial U} \Delta U^k = -\text{Res}^k$$

- Jacobian: Exact for 2nd-order scheme
- Gauss-Seidel Relaxation

This is Newton’s Method for second-order scheme.
<table>
<thead>
<tr>
<th>Objectives</th>
<th>Basic Formulation</th>
<th>Extension to High-Order</th>
<th>Some Results</th>
<th>Summary</th>
<th>Future Work</th>
</tr>
</thead>
</table>

**Second-Order Discretization**
Second-Order Discretization

With simple trapezoidal rule for the source terms,

$$
\Phi^C = \begin{bmatrix}
-a(u_{i+1} - u_i) + \nu(p_{i+1} - p_i) - h_R \frac{\alpha}{\Delta t} (u_{i+1} + u_i)/2 \\
\frac{1}{T_r} \left[ u_{i+1} - u_i - \frac{h_R}{2} (p_{i+1} + p_i) \right] \\
\frac{h_R}{2} (\tilde{s}_{i+1} + \tilde{s}_i) \\
0
\end{bmatrix}^{k+1}
+ \begin{bmatrix}
\frac{h_R}{2} (\tilde{s}_{i+1} + \tilde{s}_i) \\
0
\end{bmatrix}^{n-1,n}
$$
Second-Order Discretization

With simple trapezoidal rule for the source terms, we get a uniform second-order scheme for all variables:

\[
\Phi^C = \begin{bmatrix}
-a(u_{i+1} - u_i) + \nu(p_{i+1} - p_i) - h_R \frac{\alpha}{\Delta t} (u_{i+1} + u_i) / 2 \\
\frac{1}{Tr} \left[ u_{i+1} - u_i - \frac{h_R}{2} (p_{i+1} + p_i) \right]
\end{bmatrix}^{k+1}
\]

\[
+ \begin{bmatrix}
\frac{h_R}{2} (\tilde{s}_{i+1} + \tilde{s}_i) \\
0
\end{bmatrix}^{n-1,n}
\]

\[
T.E. (\partial_\tau p) = (\partial_x u_i - p_i) \quad + \quad \frac{h}{2} (\partial_{xx} u_i - \partial_x p_i) \\
+ \quad \frac{h^2}{6} (\partial_{xxx} u_i - \frac{6}{4} \partial_{xx} p_i) + O(h^3)
\]
Extension to High-Order

\[ \Phi^C = \int_{x_i}^{x_{i+1}} (-A U_x + S) \, dx \]
Extension to High-Order

\[
\Phi^C = \int_{x_i}^{x_{i+1}} (-A U_x + S) \, dx
\]

Methods

1. Divergence Formulation of Source Terms (RD-D)
Extension to High-Order

\[ \Phi^C = \int_{x_i}^{x_{i+1}} (-A U_x + S) \, dx \]

**Methods**

1. Divergence Formulation of Source Terms (RD-D)

\[ \int_{x_i}^{x_{i+1}} S \, dx = \int_{x_i}^{x_{i+1}} f^S_x \, dx \]
Extension to High-Order

\[ \Phi^C = \int_{x_i}^{x_{i+1}} (-A U_x + S) \, dx \]

Methods

1. Divergence Formulation of Source Terms (RD-D)
   \[ \int_{x_i}^{x_{i+1}} S \, dx = \int_{x_i}^{x_{i+1}} f_x^S \, dx \]

2. General Trapezoidal Rule (RD-GT)
Extension to High-Order

\[
\Phi^C = \int_{x_i}^{x_{i+1}} (-AU_x + S) \, dx
\]

Methods

1. Divergence Formulation of Source Terms (RD-D)

\[
\int_{x_i}^{x_{i+1}} S \, dx = \int_{x_i}^{x_{i+1}} f_x^S \, dx
\]

2. General Trapezoidal Rule (RD-GT)

\[
\int_{x_i}^{x_{i+1}} S \, dx = \frac{h_R}{2} (S_L + S_R)
\]
Third-Order RD-D Scheme

Third-Order Scheme:
Third-Order RD-D Scheme

- Third-Order Scheme:
  See the paper!
Fourth-Order RD-D Scheme

\[ \Phi^C = \int_{x_i}^{x_{i+1}} (-A U_x + f_x^S) dx \]

We introduce a source function, \( f^S \), based on mid-point:

\[ f^S = \sum_{n=1}^{m \geq 3} \frac{(-1)^{n-1}}{n!} (x - \bar{x})^n \partial_x^{n-1} S \]

\[ = (x - \bar{x}) S - \frac{1}{2} (x - \bar{x})^2 \partial_x S + \frac{1}{6} (x - \bar{x})^3 \partial_{xx} S + \ldots \]

where \( \bar{x} = (x_i + x_{i+1})/2 \).
Fourth-Order RD-D Scheme

What happens to our Original equation with the use of divergence formulation?
Fourth-Order RD-D Scheme

What happens to our Original equation with the use of divergence formulation?

Let’s consider $m = 3$; we have:

$$
\partial_x f^S = S + \frac{(x - \bar{x})^3}{6} \partial_{xxx} S = S + O(h^3)
$$
What happens to our Original equation with the use of divergence formulation?

Let’s consider \( m = 3 \); we have:

\[
\partial_x f^S = S + \frac{(x - \bar{x})^3}{6} \partial_{xxx} S = S + O(h^3)
\]

- That is, we recover the original \( S \) up to \( O(h^m) \).
Fourth-Order RD-D Scheme

- Examining the order of accuracy of the discretized system with the divergence form of the source term:

\[
T.E. \ (\partial_T p) = \underbrace{(\partial_x u_i - p_i)}_{0} + \frac{h_R}{2} \underbrace{(\partial_{xx} u_i - \partial_x p_i)}_{0} + \frac{h_R^2}{6} \underbrace{(\partial_{xxx} u_i - \partial_{xx} p_i)}_{0} + \frac{h_R^3}{24} \underbrace{(\partial_{xxxx} u_i - \partial_{xxx} p_i)}_{0} + \frac{h_R^4}{120} \underbrace{(\partial_{xxxxx} u_i - \frac{5}{4} \partial_{xxxx} p_i)}_{0} + O(h^5)
\]

How come we got 4th-order instead of 3rd-order?
Because we used the mid-point in the divergence formulation!
Examining the order of accuracy of the discretized system with the divergence form of the source term:

\[ T.E. (\partial_T p) = (\partial_x u_i - p_i) + \frac{h_R}{2} (\partial_{xx} u_i - \partial_x p_i) + \frac{h_R^2}{6} (\partial_{xxx} u_i - \partial_{xx} p_i) + \frac{h_R^3}{24} (\partial_{xxxx} u_i - \partial_{xxx} p_i) + \frac{h_R^4}{120} (\partial_{xxxxx} u_i - \frac{5}{4} \partial_{xxxx} p_i) + O(h^5) = O(h^4)! \]

How come we got 4th-order instead of 3rd-order?
Fourth-Order RD-D Scheme

Examining the order of accuracy of the discretized system with the divergence form of the source term:

\[
TE. \ (\partial_r p) = \left( \partial_x u_i - p_i \right) + \frac{h_R}{2} \left( \partial_{xx} u_i - \partial_x p_i \right) + \frac{h_R^2}{6} \left( \partial_{xxx} u_i - \partial_{xx} p_i \right) + \frac{h_R^3}{24} \left( \partial_{xxxx} u_i - \partial_{xxx} p_i \right) + \frac{h_R^4}{120} \left( \partial_{xxxxx} u_i - \frac{5}{4} \partial_{xxxx} p_i \right) + O(h^5)
\]

= \ O(h^4)!

How come we got 4th-order instead of 3rd-order?
Because we used the mid-point in the divergence formulation!
Overview of Fourth-Order RD-D Scheme

For uniform 4th-order results, we discretize the system as

\[
\Phi^C = \int_{x_i}^{x_{i+1}} (-A U_x + S) \, dx \\
\simeq \int_{x_i}^{x_{i+1}} (-A U_x + f_x^S) \, dx \\
\simeq -A (U_{i+1} - U_i) + h_R \frac{\partial x S_i}{2} + \frac{h_R^2}{2} \partial_{xx} S_i
\]

The cost relative to 2nd-order?
Overview of Fourth-Order RD-D Scheme

For uniform 4th-order results, we discretize the system as

\[ \Phi^C = \int_{x_i}^{x_{i+1}} (-AU_x + S) \, dx \]

\[ \simeq \int_{x_i}^{x_{i+1}} (-AU_x + f^S_x) \, dx \]

\[ \simeq -A(U_{i+1} - U_i) + h_R \partial_x S_i + \frac{h_R^2}{2} \partial_{xx} S_i \]

The cost relative to 2nd-order?
- evaluation of the first derivative (2nd-order accurate)
Overview of Fourth-Order RD-D Scheme

For uniform 4th-order results, we discretize the system as

$$
\Phi^C = \int_{x_i}^{x_{i+1}} (-AU_x + S) \, dx \\
\simeq \int_{x_i}^{x_{i+1}} (-AU_x + f^S_x) \, dx \\
\simeq -A(U_{i+1} - U_i) + h_R \frac{\partial_x S_i}{2} + \frac{h_R^2}{2} \frac{\partial_{xx} S_i}{2}
$$

- The cost relative to 2nd-order?
  - evaluation of the first derivative (2nd-order accurate)
  - evaluation of the second derivative (1st-order accurate)
Overview of Fourth-Order RD-D Scheme

For uniform 4th-order results, we discretize the system as

$$\Phi^C = \int_{x_i}^{x_{i+1}} (-A U_x + S) dx$$

$$\simeq \int_{x_i}^{x_{i+1}} (-A U_x + f_x^S) dx$$

$$\simeq -A (U_{i+1} - U_i) + h_R \partial_x S_i + \frac{h_R^2}{2} \partial_{xx} S_i$$

- The cost relative to 2nd-order?
  - evaluation of the first derivative (2nd-order accurate)
  - evaluation of the second derivative (1st-order accurate)

- We can use quadratic fit to evaluate the above derivatives.
Generalized Trapezoidal Rule

\[ \Phi^C = \int_{x_i}^{x_{i+1}} (-A U_x + S) \, dx \]

Introducing a new source term integration:

\[ \int S \, dx \approx \frac{h_R}{2} (S_L + S_R) \]
Generalized Trapezoidal Rule

\[ \Phi^C = \int_{x_i}^{x_{i+1}} (-A U_x + S) \, dx \]

Introducing a new source term integration:

\[ \int S \, dx \simeq \frac{h_R}{2} (S_L + S_R) \]

where we define the left and right states of \( S \) as

\[
S_L = S_i + C^L_1 \partial_x S_i + C^L_2 \partial_{xx} S_i \\
S_R = S_{i+1} + C^R_1 \partial_x S_{i+1} + C^R_2 \partial_{xx} S_{i+1}
\]
Third-Order RD-GT Scheme

\[
\int S \, dx \simeq \frac{h_R}{2} [S_i + S_{i+1} \\
+ (C^L_1 \partial_x S_i + C^R_1 \partial_x S_{i+1}) \\
+ (C^L_2 \partial_{xx} S_i + C^R_2 \partial_{xx} S_{i+1})]
\]

Note: The first two terms are what we had in 2nd-order. The rest are the corrections to get to high-order schemes.

- **Third-Order RD-GT:**
Third-Order RD-GT Scheme

\[ \int S \, dx \approx \frac{h_R}{2} [S_i + S_{i+1} + (C_1^L \partial_x S_i + C_1^R \partial_x S_{i+1}) + (C_2^L \partial_{xx} S_i + C_2^R \partial_{xx} S_{i+1})] \]

Note: The first two terms are what we had in 2nd-order. The rest are the corrections to get to high-order schemes.

- **Third-Order RD-GT:**

  \[
  C_1^L + C_1^R = 0, \quad C_1^R h_R + C_2^L + C_2^R = -\frac{h_R^2}{6}, \quad C_1^R h_R + 2C_2^R \neq -\frac{h_R^2}{6}
  \]
Third-Order RD-GT Scheme

\[ \int S \, dx \simeq \frac{h_R}{2} \left[ S_i + S_{i+1} \right] + (C^L_1 \, \partial_x S_i + C^R_1 \, \partial_x S_{i+1}) + (C^L_2 \, \partial_{xx} S_i + C^R_2 \, \partial_{xx} S_{i+1}) \]

Note: The first two terms are what we had in 2nd-order. The rest are the corrections to get to high-order schemes.

- Third-Order RD-GT:

\[ C^L_1 + C^R_1 = 0, \quad C^R_1 h_R + C^L_2 + C^R_2 = -\frac{h_R^2}{6}, \quad C^R_1 h_R + 2C^R_2 \neq -\frac{h_R^2}{6} \]

- Many possibilities, for example:

\[ C^R_1 = -C^L_1 = -h_R/6 \]
\[ C^R_2 = -C^L_2 = -h_R^2/10 \]
Fourth-Order RD-GT Scheme

\[
\int S \, dx \simeq \frac{h_R}{2} \left[ S_i + S_{i+1} \right]
\]

\[
+ (C^L_i \partial_x S_i + C^R_i \partial_x S_{i+1})
\]

\[
+ (C^L_2 \partial_{xx} S_i + C^R_2 \partial_{xx} S_{i+1})
\]

- Fourth-Order RD-GT:
Fourth-Order RD-GT Scheme

\[
\int S \, dx \simeq \frac{h_R}{2} [S_i + S_{i+1} \\
+ (C^L_1 \partial_x S_i + C^R_1 \partial_x S_{i+1}) \\
+ (C^L_2 \partial_{xx} S_i + C^R_2 \partial_{xx} S_{i+1})]
\]

- Fourth-Order RD-GT:

\[
C^L_1 + C^R_1 = 0, \quad C^R_1 h_R + C^L_2 + C^R_2 = -\frac{h^2_R}{6}, \quad \frac{C^R_1}{2} h_R + C^R_2 = -\frac{h^2_R}{12}
\]
Fourth-Order RD-GT Scheme

\[ \int S \, dx \simeq \frac{h_R}{2}[S_i + S_{i+1} + (C_L^1 \partial_x S_i + C_R^1 \partial_x S_{i+1}) + (C_L^2 \partial_{xx} S_i + C_R^2 \partial_{xx} S_{i+1})] \]

- Fourth-Order RD-GT:
  \[ C_L^1 + C_R^1 = 0, \quad C_L^1 h_R + C_R^2 = -\frac{h_R^2}{6}, \quad \frac{C_R^1}{2} h_R + C_R^2 = -\frac{h_R^2}{12} \]

- Again, many possibilities. In particular, we can select the most efficient and attractive coefficients:
Fourth-Order RD-GT Scheme

\[
\int S \, dx \simeq \frac{h_R}{2} [S_i + S_{i+1} + (C_L^1 \partial_x S_i + C_R^1 \partial_x S_{i+1}) + (C_L^2 \partial_{xx} S_i + C_R^2 \partial_{xx} S_{i+1})]
\]

- **Fourth-Order RD-GT:**

  \[C_L^1 + C_R^1 = 0, \quad C_R^1 h_R + C_L^2 + C_R^2 = -\frac{h_R^2}{6}, \quad \frac{C_R^1}{2} h_R + C_R^2 = -\frac{h_R^2}{12}\]

- Again, many possibilities. In particular, we can select the most efficient and attractive coefficients:

  \[C_R^1 = -C_L^1 = -\frac{h_R}{6}, \quad C_R^2 = C_L^2 = 0, \quad \text{no second derivatives!}\]
Fourth-Order RD-GT Scheme

Expanding the cell residual around node $i$:

$$T.E.(\partial_\tau u_i) = (-a\partial_x u_i + \nu \partial_x p_i + S_i)$$

$$+ \frac{h_R}{2} (-a\partial_{xx} u_i + \nu \partial_{xx} p_i + \partial_x S_i)$$

$$+ \frac{h_R^2}{6} (-a\partial_{xxx} u_i + \nu \partial_{xxx} p_i + \partial_{xx} S_i)$$

$$+ \frac{h_R^3}{24} (-a\partial_{xxxx} u_i + \nu \partial_{xxxx} p_i + \partial_{xxx} S_i)$$

$$+ \frac{h_R^4}{120} (-a\partial_{xxxxx} u_i + \nu \partial_{xxxxx} p_i + \frac{5}{6} \partial_{xxx} S_i) + O(h^5)$$

$$= 0 + O(h^4)$$
Sixth-Order RD-GT Schemes

\[ \int S \, dx \simeq \frac{hR}{2} [S_i + S_{i+1}] \]

\[ + (C_1^L \partial_x S_i + C_1^R \partial_x S_{i+1}) \]

\[ + (C_2^L \partial_{xx} S_i + C_2^R \partial_{xx} S_{i+1}) \]

- Fifth-Order RD-GT:
Sixth-Order RD-GT Schemes

\[ \int S \, dx \simeq \frac{h_R}{2} \left[ S_i + S_{i+1} \right. \\
+ \left. (C^L_1 \partial_x S_i + C^R_1 \partial_x S_{i+1}) \right. \\
+ \left. (C^L_2 \partial_{xx} S_i + C^R_2 \partial_{xx} S_{i+1}) \right] \]

- **Fifth-Order RD-GT:**
  - Fourth-Order Constraints & \[ \frac{C^R_1}{3} h_R + C^R_2 = -\frac{h^2_R}{20} \]
Sixth-Order RD-GT Schemes

\[ \int S \, dx \approx \frac{h_R}{2} [S_i + S_{i+1}] \]

\[ + (C^L_1 \partial_x S_i + C^R_1 \partial_x S_{i+1}) \]

\[ + (C^L_2 \partial_{xx} S_i + C^R_2 \partial_{xx} S_{i+1}) \]

- Fifth-Order RD-GT:
  Fourth-Order Constraints \& \( \frac{C^R_1}{3} h_R + C^R_2 = -\frac{h^2_R}{20} \)

- A Unique Solution:
  \( C^R_1 = -C^L_1 = -h_R/5 \)
  \( C^R_2 = C^L_2 = h^2_R/60 \)
Sixth-Order RD-GT Schemes

\[
\int S \, dx \simeq \frac{h_R}{2} [S_i + S_{i+1}]
+ (C^L_1 \partial_x S_i + C^R_1 \partial_x S_{i+1})
+ (C^L_2 \partial_{xx} S_i + C^R_2 \partial_{xx} S_{i+1})
\]

- Fifth-Order RD-GT:
  - Fourth-Order Constraints & \( \frac{C^R_1}{3} h_R + C^R_2 = -\frac{h_R^2}{20} \)
  - A Unique Solution:
    \( C^R_1 = -C^L_1 = -h_R/5 \)
    \( C^R_2 = C^L_2 = h_R^2/60 \)

- Why then 6th-order not fifth-order?
Sixth-Order RD-GT Schemes

\[ \int S \, dx \simeq \frac{h_R}{2} [S_i + S_{i+1} + (C^L_1 \partial_x S_i + C^R_1 \partial_x S_{i+1}) + (C^L_2 \partial_{xx} S_i + C^R_2 \partial_{xx} S_{i+1})] \]

- Fifth-Order RD-GT:
  Fourth-Order Constraints & \( \frac{C^R_1}{3} h_R + C^R_2 = -\frac{h_R^2}{20} \)
  A Unique Solution:
    \( C^R_1 = -C^L_1 = -\frac{h_R}{5} \)
    \( C^R_2 = C^L_2 = \frac{h_R^2}{60} \)

- Why then 6th-order not fifth-order?
  Because the Sixth-Order Constraint \( \frac{C^R_1}{4} h_R + C^R_2 = -\frac{h_R^2}{30} \) is also satisfied!
Some Results

Steady state boundary layer problem:

$$\partial_t u + a \partial_x u = \nu \partial_{xx} u + s(x)$$

where

$$s(x) = \frac{\pi}{Re} \left[ a \cos(\pi x) + \pi \nu \sin(\pi x) \right], \quad Re = a/\nu.$$
$O(N)$ + Newton Convergence

- Fast and Newton + $O(N)$ convergence on irregular grids
- 2nd-order Jacobian acts like exact for 3rd, 4th, and 6th-order schemes

**Table:** Boundary layer problem (Residuals Criteria : $< 10^{-8}$.)

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Scheme Order</th>
<th>GS /Newton</th>
<th>Newton iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>3rd</td>
<td>163</td>
<td>8</td>
</tr>
<tr>
<td>50</td>
<td>4th</td>
<td>163</td>
<td>8</td>
</tr>
<tr>
<td>50</td>
<td>6th</td>
<td>163</td>
<td>8</td>
</tr>
<tr>
<td>100</td>
<td>3rd</td>
<td>324</td>
<td>7</td>
</tr>
<tr>
<td>100</td>
<td>4th</td>
<td>324</td>
<td>7</td>
</tr>
<tr>
<td>100</td>
<td>6th</td>
<td>324</td>
<td>7</td>
</tr>
<tr>
<td>500</td>
<td>3rd</td>
<td>1647</td>
<td>7</td>
</tr>
<tr>
<td>500</td>
<td>4th</td>
<td>1647</td>
<td>7</td>
</tr>
<tr>
<td>500</td>
<td>6th</td>
<td>1647</td>
<td>7</td>
</tr>
</tbody>
</table>
Boundary Layer Problem: 3rd-Order

Solutions and Convergence on Irregular Grids

Third-Order RD Schemes

$u$ (RD–GT)
$p = u_x$ (RD–GT)
$u$ (RD–D)
$p = u_x$ (RD–D)
Boundary Layer Problem: 4th- and 6th-Order

Convergence on Irregular Grids

(a) fourth-order \((Re = 1)\)

(b) sixth-order \((Re = 10)\)
Unsteady Non-Linear Problem

Problem Statement:

\[ \partial_t u + \partial_x f = \partial_x (\nu u_x) + S(x, t), \quad x \in (0, 1) \]

where \( f = u^2/2, \nu = u, \) and

\[ S(x, t) = u_t^e + \frac{1}{2}((u^e)^2)_x - (u_x^e)^2 - u^e u^e_{xx} \]

Exact Solution:

\[ u^e(x, t) = \text{Real} \left( \frac{\sinh(x \sqrt{i\omega/\nu})}{\sinh(\sqrt{i\omega/\nu})} U e^{i\omega t} \right) + C, \quad C > 1 \]
Unsteady Non-Linear Problem

Solution:

Solution gradient:

- GS-Relaxation: 2 orders of magnitude reduction
- Residual tolerance: $\leq 10^{-8}$
Unsteady Non-Linear Problem

$O(N)$ Convergence and Newton!

Table: Average data over 1000 time steps are shown (Irregular Grids.)

<table>
<thead>
<tr>
<th>Nodes</th>
<th>RD-GT Scheme</th>
<th>Order</th>
<th>GS/Newton</th>
<th>Newton</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>3rd</td>
<td>435</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4th</td>
<td>430</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6th</td>
<td>431</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>3rd</td>
<td>879</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4th</td>
<td>868</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6th</td>
<td>864</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>3rd</td>
<td>1772</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4th</td>
<td>1749</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6th</td>
<td>1737</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
Unsteady Non-Linear Problem: 4th- and 6th-Order

Convergence on Irregular Grids

Fourth-Order RD Schemes

- Slope 2
- Slope 3
- Slope 4

Sixth-Order RD Scheme

- Slope 2
- Slope 3
- Slope 4
- Slope 6

(c) fourth-order + BDF4
(d) sixth-order + BDF6
Developed uniform very high-order time-accurate RD schemes for general hyperbolic advection-diffusion on irregular grids
Summary

- Developed uniform very high-order time-accurate RD schemes for general hyperbolic advection-diffusion on irregular grids
- Proposed two new source integration techniques:
Summary

- Developed uniform very high-order time-accurate RD schemes for general hyperbolic advection-diffusion on irregular grids
- Proposed two new source integration techniques:
  1) a new divergence formulation

Showed $O(N)$ convergence rate for the linear system + Newton for all the proposed high-order schemes.
Summary

- Developed uniform very high-order time-accurate RD schemes for general hyperbolic advection-diffusion on irregular grids
- Proposed two new source integration techniques:
  1) a new divergence formulation
  2) corrections to the trapezoidal rule
Summary

- Developed uniform very high-order time-accurate RD schemes for general hyperbolic advection-diffusion on irregular grids
- Proposed two new source integration techniques:
  1) a new divergence formulation
  2) corrections to the trapezoidal rule
- Proposed a fourth-order scheme that compared to the second-order scheme only costs evaluation of first-derivative of the source term
Summary

- Developed uniform very high-order time-accurate RD schemes for general hyperbolic advection-diffusion on irregular grids
- Proposed two new source integration techniques:
  1) a new divergence formulation
  2) corrections to the trapezoidal rule
- Proposed a fourth-order scheme that compared to the second-order scheme only costs evaluation of first-derivative of the source term
- Shown $O(N)$ convergence rate for the linear system + Newton for all the proposed high-order schemes
Works Underway:

- Extension to multi-dimensions (snapshot in the next slide)
Works Underway:

- Extension to multi-dimensions (snapshot in the next slide)
- Inclusion of shocks and discontinuities
Works Underway:

- Extension to multi-dimensions (snapshot in the next slide)
- Inclusion of shocks and discontinuities
- Effects of separating advection and diffusion eigen structures
Works Underway: Extensions to Multi-Dimensions

Problem: Steady 2D Advection-Diffusion

\[ \partial_t u + a \partial_x u + b \partial_y u = \nu (\partial_{xx} u + \partial_{yy} u) \]

Exact Solution:

\[ u(x, y) = C \cos(A \pi \eta) \exp\left(\frac{-2A^2 \pi^2 \nu}{1 + \sqrt{1 + 4A^2 \pi^2 \nu^2}} \xi\right), \]

where \( \xi = ax + by, \eta = bx - ay. \)

- Problem Setup: \( u \) is specified on the boundaries.
  - Solve for \( u, \partial_x u \) and \( \partial_y u \).
- GS-Relaxation: 1000 or 5 orders of magnitude reduction
- Residual tolerance: \( \leq 10^{-11} \)
Works Underway: Extensions to Multi-Dimensions

Newton: 3; GS/Newton : ≤ 300; \( a = 2, b = 1, \nu = 0.01 \)