Lax-Wendroff Flux Reconstruction method for conservation laws

Praveen Chandrashekar
praveen@math.tifrbng.res.in
http://cpraveen.github.io

Center for Applicable Mathematics
Tata Institute of Fundamental Research
Bangalore-560065, India
http://math.tifrbng.res.in

IGP/IWR School on Hardware-aware Scientific Computing
Bangalore & Heidelberg
4-15 October, 2021

Joint work with

Arpit Babbar, TIFR-CAM, Bangalore
Sudarshan Kumar, Dept. of Math., IISER, Trivandrum

Outline

Theme: High order, single step methods for hyperbolic conservation laws, compressible Euler and compressible Navier-Stokes

- Method-of-lines approach + high order in space (FV, DG)
- Lax-Wendroff idea
- Flux reconstruction idea[1]
- Lax-Wendroff + Flux-Reconstruction
- Issues concerning numerical flux
- Some results

[1] `http://www.pyfr.org`

Method-of-lines approach

Hyperbolic conservation law

$$u_t + f(u)_x = 0$$

1) Spatial discretization:
Finite volume method: semi-discrete

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} + \frac{f_{j+\frac{1}{2}} - f_{j-\frac{1}{2}}}{\Delta x} = 0, \qquad j = 0, 1, \ldots, M-1$$

Numerical flux

$$f_{j+\frac{1}{2}} = f(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}})$$

2) System of ODE: apply a Runge-Kutta scheme in time

Space and time discretization are decoupled

Runge-Kutta approach: $\frac{\mathrm{d}U}{\mathrm{d}t} = R(U,t)$

3'rd order SSPRK: $U^n \to U^{n+1}$

(Needs three vectors)

$$
\begin{aligned}
U^{(1)} &= U^n + \Delta t\ R(U^n, t_n) \\
U^{(2)} &= \frac{3}{4}U^n + \frac{1}{4}[U^{(1)} + \Delta t\ R(U^{(1)}, t_n + \Delta t)] \\
U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}[U^{(2)} + \Delta t\ R(U^{(2)}, t_n + \tfrac{1}{2}\Delta t)]
\end{aligned}
$$

Classical 4'th order RK: $U^n \to U^{n+1}$

(Needs four vectors)

$$
\begin{aligned}
k_1 &= R(U^n, t_n) \\
k_2 &= R(U^n + \tfrac{1}{2}\Delta t\ k_1, t_n + \tfrac{1}{2}\Delta t) \\
k_3 &= R(U^n + \tfrac{1}{2}\Delta t\ k_2, t_n + \tfrac{1}{2}\Delta t) \\
k_4 &= R(U^n + \Delta t\ k_3, t_n + \Delta t) \\
U^{n+1} &= U^n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

Higher order $\Longrightarrow$ more stages, more memory

Discontinuous Galerkin method

Find $u_h(t) \in V_h^N$ for degree $N \geq 0$

$$V_h^N = \{v \in L^2(\Omega) : v|_{\Omega_e} \in \mathbb{P}_N \ \forall e\}, \qquad \Omega = \bigcup_e \Omega_e$$

such that, for all $v_h \in V_h^N$

$$\boxed{\int_{\Omega_e} v_h \frac{\partial u_h}{\partial t} \mathrm{d}x - \int_{\Omega_e} f(u_h) \frac{\partial v_h}{\partial x} \mathrm{d}x + f_{e+\frac{1}{2}} v_h(x^-_{e+\frac{1}{2}}) - f_{e-\frac{1}{2}} v_h(x^+_{e-\frac{1}{2}}) = 0}$$

is satisfied.

- $f_{e+\frac{1}{2}}$ comes from FVM
- Requires some quadrature for integrals
- Efficient, quadrature-free implementation
  - Use nodal Lagrange basis functions supported at Gauss points
  - Use same Gauss points for quadrature
  - Equivalent to some versions of FR method
- System of ODE: apply an RK method in time

Discontinuous FE space in 1-D

Approximate u_h by piecewise degree $N \geq 0$ polynomials in each element

$$\Omega_e = [x_{e-\frac{1}{2}}, x_{e+\frac{1}{2}}], \qquad \Delta x_e = x_{e+\frac{1}{2}} - x_{e-\frac{1}{2}}$$

Map each element to a reference element

$$\Omega_e \to [0,1], \qquad x \to \xi = \frac{x - x_{e-\frac{1}{2}}}{\Delta x_e}$$

Choose $N+1$ distinct nodes: *solution points* (GL or GLL)

$$0 \leq \xi_0 < \xi_1 < \cdots < \xi_N \leq 1$$

Associated quadrature weights $\{w_j\}$
ℓ_j is a Lagrange polynomial of degree N

$$\ell_j(\xi) = \prod_{i=0, i\neq j}^{N} \frac{\xi - \xi_i}{\xi_j - \xi_i} \in \mathbb{P}_N$$

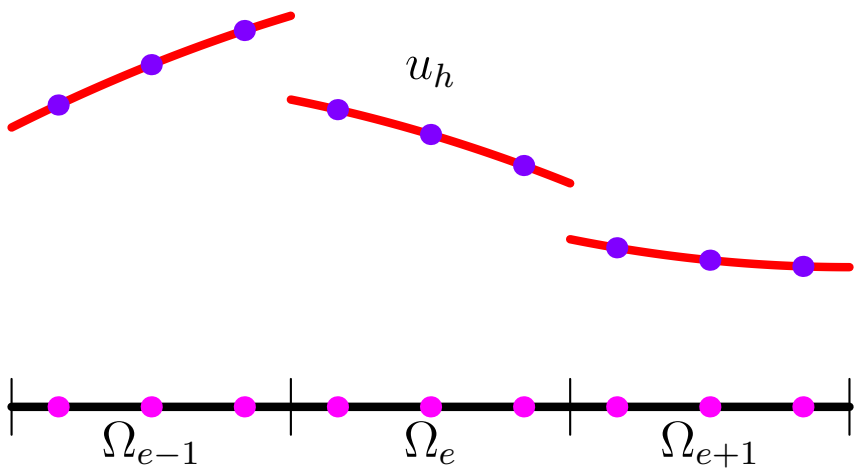
Discontinuous FE space in 1-D

Solution inside an element

$$x \in \Omega_e : \qquad u_h(\xi, t) = \sum_{j=0}^{N} u_j^e(t) \ell_j(\xi)$$

The dofs $\{u_j^e\}$ are solution values at the solution points

$$u_j^e(t) = u_h(\xi_j, t)$$



Differentiation and interpolation

Differentiation matrix: $\mathsf{D} = [D_{ij}]$

$$D_{ij} = \ell_j'(\xi_i), \qquad 0 \le i, j \le N$$

Spatial derivatives of the solution u_h at all the solution points

$$\begin{bmatrix} \partial_x u_h(\xi_0, t) \\ \vdots \\ \partial_x u_h(\xi_N, t) \end{bmatrix} = \frac{1}{\Delta x_e} \mathsf{D}\mathsf{u}(t), \qquad \mathsf{u} = \begin{bmatrix} u_0^e \\ \vdots \\ u_N^e \end{bmatrix}$$

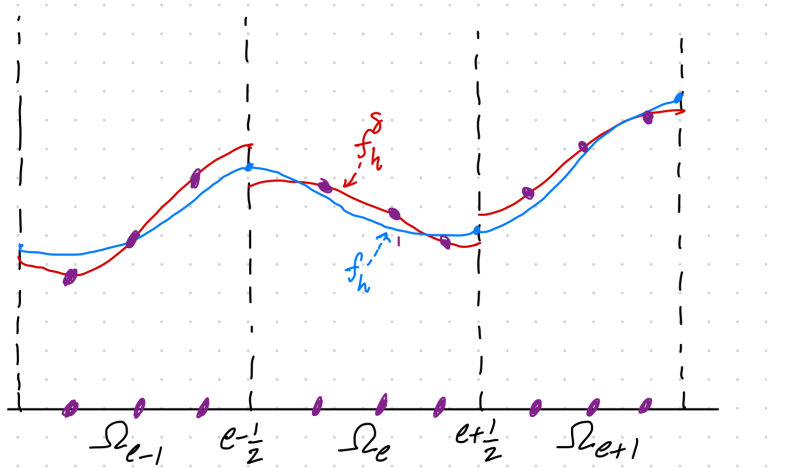
Vandermonde matrix wrt to left and right boundaries of reference cell

$$\mathsf{V}_L = [\ell_0(0), \ell_1(0), \ldots, \ell_N(0)]^\top, \qquad \mathsf{V}_R = [\ell_0(1), \ell_1(1), \ldots, \ell_N(1)]^\top$$

RKFR method [1]

Step 1. Build discontinuous flux in each element: $f_h^\delta \in \mathbb{P}_N$

$$f_h^\delta(\xi,t) = \sum_{j=0}^{N} f(u_j^e(t))\ell_j(\xi)$$



Step 2. Correct f_h^δ to make it continuous across elements: $f_h \in \mathbb{P}_{N+1}$

$$f_h(\xi,t) = \left[\boldsymbol{f}_{e-\frac{1}{2}}(t) - f_h^\delta(0,t)\right] g_L(\xi) + f_h^\delta(\xi,t) + \left[\boldsymbol{f}_{e+\frac{1}{2}}(t) - f_h^\delta(1,t)\right] g_R(\xi)$$

Numerical flux: $\boldsymbol{f}_{e+\frac{1}{2}}(t) = f(u_h(x^-_{e+\frac{1}{2}},t), u_h(x^+_{e+\frac{1}{2}},t))$

Step 3. Satisfy PDE at solution points

$$\frac{\mathrm{d}u_j^e}{\mathrm{d}t}(t) = -\frac{1}{\Delta x_e}\frac{\partial f_h}{\partial \xi}(\xi_j,t), \qquad 0 \le j \le N \qquad \text{(Quad-free)}$$

Integrate ODE with an RK method

FR correction functions

Huynh [1], Vincent et al. [2]

Conditions needed for flux continuity: $f_h(x^-_{e+\frac{1}{2}}, t) = f_h(x^+_{e+\frac{1}{2}}, t) = f_{e+\frac{1}{2}}(t)$

$$g_L(0) = 1, \quad g_R(0) = 0$$
$$g_L(1) = 0, \quad g_R(1) = 1$$
$$\text{and} \qquad g_{L/R}(\xi) \approx 0, \quad \xi \in (0,1)$$

Radau correction function[2] ($c = 0$ in [2], $g_{L/R} \perp \mathbb{P}_{N-1}$)

$$g_L(\xi) = \frac{(-1)^N}{2}[L_N(2\xi-1) - L_{N+1}(2\xi-1)]$$
$$g_R(\xi) = \frac{1}{2}[L_N(2\xi-1) + L_{N+1}(2\xi-1)]$$

g2 correction function ($c = c_{HU}(N)$ in [2], $g_{L/R} \perp \mathbb{P}_{N-2}$)

$$g_L(\xi) = \frac{(-1)^N}{2}\left[L_N(2\xi-1) - \frac{(N+1)L_{N-1}(2\xi-1) + NL_{N+1}(2\xi-1)}{2N+1}\right]$$
$$g_R(\xi) = \frac{1}{2}\left[L_N(2\xi-1) + \frac{(N+1)L_{N-1}(2\xi-1) + NL_{N+1}(2\xi-1)}{2N+1}\right]$$

[2] $L_N \in \mathbb{P}_N$ are Legendre polynomials

FR correction functions: From Vincent et al. [2]

$$\tilde{g}_L(s) = \frac{(-1)^N}{2}\left[L_N(s) - \frac{\eta_N L_{N-1}(s) + L_{N+1}(s)}{1+\eta_N}\right], \qquad \eta_N = \frac{c}{2}(2N+1)(a_N N!)^2$$

Semi-discrete stability

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\|u_h\|^2 + \frac{c}{2}\|\partial_x^N u_h\|^2\right) \leq 0$$

$$g_L(\xi) = \tilde{g}_L(2\xi - 1)$$

$c = 0$: Radau
$\implies$ Nodal DG + GL

$c = c_{HU}(N)$: g2
$\implies$ Nodal DG + GLL

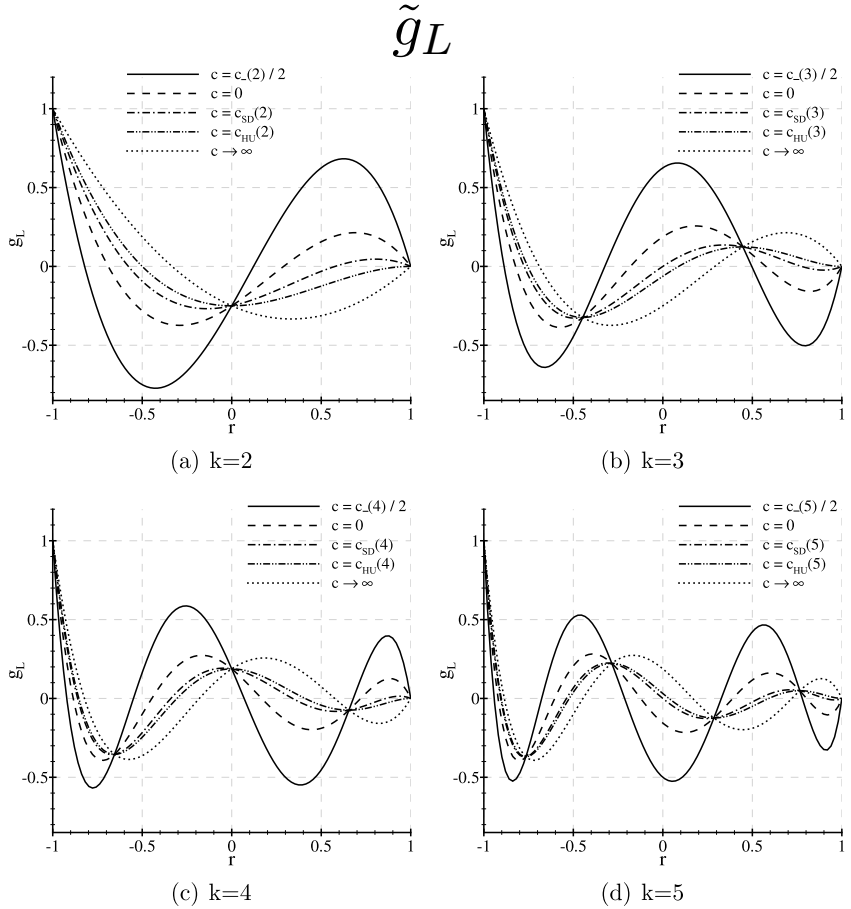


Fig. 1 Plots of the correction function g_L when $k = 2$ (**a**), $k = 3$ (**b**), $k = 4$ (**c**), and $k = 5$ (**d**), for various values of c

Algorithm 1: RKFR scheme

$t = 0$;
while $t < T$ **do**
    for *each RK stage* **do**
        Loop over cells and assemble rhs;
        Loop over faces and assemble rhs;
        Update solution to next RK stage;
        Perform MPI data exchange;
        Apply a posteriori limiter;
        Apply positivity limiter;
        Perform MPI data exchange;
    $t = t + \Delta t$;

Classical Lax-Wendroff method

Taylor expand in time

$$u(x_j, t_n + \Delta t) = u(x_j, t_n) + u_t(x_j, t_n)\Delta t + u_{tt}(x, t)\frac{\Delta t^2}{2} + O(\Delta t^3)$$

Use PDE to replace time derivative in terms of space derivatives

$$\begin{aligned} u_t &= -f_x \\ u_{tt} &= -(f_t)_x = -(f'(u)u_t)_x = (A(u)f_x)_x \end{aligned}$$

Use central differencing in space

$$\begin{aligned} u_t &= -f_x \approx \frac{f_{j+1} - f_{j-1}}{2\Delta x} \\ u_{tt} &= (A(u)f_x)_x \approx \frac{A(u_{j+\frac{1}{2}})\frac{f_{j+1}-f_j}{\Delta x} - A(u_{j-\frac{1}{2}})\frac{f_j - f_{j-1}}{\Delta x}}{\Delta x} \end{aligned}$$

Classical Lax-Wendroff method

Single step LW update

$$u_j^{n+1} = u_j^n + u_t(x_j, t_n)\Delta t + u_{tt}(x_j, t_n)\frac{\Delta t^2}{2}$$

- Space and time discretization are coupled
- Requires the flux jacobian matrix A
- Second order in space and time
- Can be written as a finite volume scheme

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}})$$

$$F_{j+\frac{1}{2}} = \frac{1}{2}(f_j + f_{j+1}) - \frac{\Delta t}{2\Delta x}A(u_{j+\frac{1}{2}})(f_{j+1} - f_j)$$

Some broad class of methods

- Taylor-Galerkin method [3]
- Lax-Wendroff-FV [4] and Lax-Wendroff-DG [5, 6]
- ADER-FV [7] and ADER-DG [8, 9]
- Two derivative RK methods [10, 11]
- LWFR method [12]

Local solution or predictor

- Cauchy-Kowalevsky (CK) procedure
- Continuous RK
- Local Galerkin method

CK procedure

- Compute jacobians
- Use finite difference
- Approximate Lax-Wendroff [13, 6]

LWFR method

Taylor expansion in time around $t = t_n$

$$u^{n+1} = u^n + \sum_{m=1}^{N+1} \frac{\Delta t^m}{m!} \partial_t^m u^n + O(\Delta t^{N+2})$$

Use the PDE

$$\partial_t u = -\partial_x f \qquad \Longrightarrow \qquad \partial_t^m u = -(\partial_t^{m-1} f)_x$$

so that

$$\begin{aligned} u^{n+1} &= u^n - \Delta t \left[\sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!} \partial_t^m f \right]_x + O(\Delta t^{N+2}) \\ &= u^n - \Delta t \frac{\partial F}{\partial x}(u^n) + O(\Delta t^{N+2}) \end{aligned}$$

LWFR method

where

$$F(u) = f(u) + \frac{\Delta t}{2}\partial_t f(u) + \ldots + \frac{\Delta t^N}{(N+1)!}\partial_t^N f(u)$$

is the time average flux, since

$$F = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \left[f + (t-t_n)\partial_t f + \ldots + \frac{(t-t_n)^N}{N!}\partial_t^N f \right] \mathrm{d}t$$

Following RKFR

- approximate time average flux F inside each element by a continuous polynomial $F_h(\xi)$
- Update solution at all the solution nodes

$$\boxed{(u_j^e)^{n+1} = (u_j^e)^n - \frac{\Delta t}{\Delta x_e}\frac{\mathrm{d}F_h}{\mathrm{d}\xi}(\xi_j), \qquad 0 \le j \le N}$$

Single step method for all orders

Needs two vectors

LWFR conservation property

$\{w_j\}$ quadrature weights associated with solution points

Integrate over one element using quadrature induced by solution points

$$\Delta x_e \sum_{j=0}^{N} w_j (u_j^e)^{n+1} = \Delta x_e \sum_{j=0}^{N} w_j (u_j^e)^n - \Delta t \sum_{j=0}^{N} w_j \frac{\partial F_h}{\partial \xi}(\xi_j)$$

F_h is polynomial of degree $\leq N+1$, can be exactly integrated with GLL/GL points

$$\int_{\Omega_e} u_h^{n+1} \mathrm{d}x = \int_{\Omega_e} u_h^n \mathrm{d}x - \Delta t [F_{e+\frac{1}{2}} - F_{e-\frac{1}{2}}]$$

total mass inside the cell changes only due to the element boundary fluxes $\Longrightarrow$ the scheme is conservative.

LW flux reconstruction: F_h and $\partial_\xi F_h$

Step 1. Use the *approximate Lax-Wendroff procedure* to compute the time average flux F at all the solution points

$$F_j^e \approx F(\xi_j), \qquad 0 \leq j \leq N$$

Step 2. Build a local approximation of the time average flux inside each element by interpolating at the solution points

$$F_h^\delta(\xi) = \sum_{j=0}^{N} F_j^e \ell_j(\xi)$$

which however may not be continuous across the elements.

Step 3. Modify the flux approximation $F_h^\delta(\xi)$ so that it becomes continuous across the elements

$$F_h(\xi) = \left[\boldsymbol{F}_{e-\frac{1}{2}} - F_h^\delta(0)\right] g_L(\xi) + F_h^\delta(\xi) + \left[\boldsymbol{F}_{e+\frac{1}{2}} - F_h^\delta(1)\right] g_R(\xi)$$

LW flux reconstruction: F_h and $\partial_\xi F_h$

$\boldsymbol{F}_{e+\frac{1}{2}}$: numerical flux function that approximates the flux F at $x = x_{e+\frac{1}{2}}$.
g_L, g_R: Correction functions chosen from the FR literature.

Step 4. Derivatives of continuous flux approx at solution points

$$\partial_\xi \mathsf{F}_h = \left[F_{e-\frac{1}{2}} - \mathsf{V}_L^\top \mathsf{F}\right] \mathsf{b}_L + \mathsf{D}\mathsf{F} + \left[F_{e+\frac{1}{2}} - \mathsf{V}_L^\top \mathsf{F}\right] \mathsf{b}_R,$$

$$\mathsf{b}_L = \begin{bmatrix} g_L'(\xi_0) \\ \vdots \\ g_L'(\xi_N) \end{bmatrix}, \qquad \mathsf{b}_R = \begin{bmatrix} g_R'(\xi_0) \\ \vdots \\ g_R'(\xi_N) \end{bmatrix}$$

Re-write as

$$\partial_\xi \mathsf{F}_h = F_{e-\frac{1}{2}} \mathsf{b}_L + \mathsf{D}_1 \mathsf{F} + F_{e+\frac{1}{2}} \mathsf{b}_R, \qquad \mathsf{D}_1 = \mathsf{D} - \mathsf{b}_L \mathsf{V}_L^\top - \mathsf{b}_R \mathsf{V}_R^\top$$

Compute $\partial_\xi \mathsf{F}_h$ using cell loop and face loop
How to compute F and $F_{e+\frac{1}{2}}$?

LW procedure for F

$$F(u) = f(u) + \frac{\Delta t}{2}\partial_t f(u) + \ldots + \frac{\Delta t^N}{(N+1)!}\partial_t^N f(u)$$

Classical procedure

$$f_t = f'(u)u_t = -A(u)f_x$$

Approximate f_x directly or via

$$f_x = A(u)u_x$$

At next order

$$f_{tt} = -A(u)(f_t)_x - \frac{\partial A}{\partial t} f_x, \qquad \frac{\partial A_{ij}}{\partial t} = A'_{ij}(u) \cdot u_t = -A'_{ij}(u) \cdot f_x$$

At higher order accuracy, algebra becomes complicated, need to compute Jacobian matrices and higher order tensors, computational cost increases.

Approximate LW procedure [13, 6]

$$F(u) = f(u) + \frac{\Delta t}{2}\partial_t f(u) + \ldots + \frac{\Delta t^N}{(N+1)!}\partial_t^N f(u)$$

Idea: use finite differencing in time to approximate time derivatives of f

Example: $N = 1$

$$\begin{aligned}
f_t(\xi,t) &\approx \frac{f(u(\xi,t+\Delta t)) - f(u(\xi,t-\Delta t))}{2\Delta t} \\
u(\xi,t\pm\Delta t) &\approx u(\xi,t) \pm u_t(\xi,t)\Delta t \\
u_t(\xi,t) &\approx -f_x(\xi,t) \qquad \leftarrow \quad \text{use differentiation matrix} \\
F(\xi) &\approx f(u(\xi,t)) + \frac{\Delta t}{2} f_t(\xi,t)
\end{aligned}$$

Note: Neglected term in Taylor series is $O(\Delta t^{N+1})$

Estimate $\partial_t^m f$ to atleast $O(\Delta t^{N+1-m})$ accuracy

Approximate LW procedure [13, 6]

Define the notation:

$$u^{(0)} = u, \qquad f^{(0)} = f(u)$$

and

$$u^{(m)} = \Delta t^m \partial_t^m u, \qquad f^{(m)} = \Delta t^m \partial_t^m f, \qquad m = 1, 2, \ldots$$

Time derivatives of the solution are computed using the PDE

$$u^{(m)} = -\partial_x f^{(m-1)}, \qquad m = 1, 2, \ldots$$

ALW procedure: $N = 1$

Given the solution in Ω_e

$$\mathsf{u} = [u_0^n,\ u_1^n,\ , \ldots, u_N^n]^\top$$

compute

$$\begin{aligned}
\mathsf{f} &= f(\mathsf{u}) \\
\mathsf{u}^{(1)} &= -\frac{\Delta t}{\Delta x_e}\mathsf{D}\mathsf{f} \\
\mathsf{f}^{(1)} &= \frac{1}{2}\left[f\left(\mathsf{u}+\mathsf{u}^{(1)}\right) - f\left(\mathsf{u}-\mathsf{u}^{(1)}\right)\right] \\
\mathsf{F} &= \mathsf{f} + \frac{1}{2}\mathsf{f}^{(1)}
\end{aligned}$$

and

$$\partial_\xi \mathsf{F} = \mathsf{D}_1 \mathsf{F}$$

ALW procedure: $N = 3$

$$
\begin{aligned}
\mathsf{f} &= f(\mathsf{u}) \\
\mathsf{u}^{(1)} &= -\frac{\Delta t}{\Delta x_e}\mathsf{Df} \\
\mathsf{f}^{(1)} &= \frac{1}{12}\left[-f\left(\mathsf{u}+2\mathsf{u}^{(1)}\right)+8f\left(\mathsf{u}+\mathsf{u}^{(1)}\right)-8f\left(\mathsf{u}-\mathsf{u}^{(1)}\right)+f\left(\mathsf{u}-2\mathsf{u}^{(1)}\right)\right] \\
\mathsf{u}^{(2)} &= -\frac{\Delta t}{\Delta x_e}\mathsf{Df}^{(1)} \\
\mathsf{f}^{(2)} &= f\left(\mathsf{u}+\mathsf{u}^{(1)}+\frac{1}{2}\mathsf{u}^{(2)}\right)-2f(\mathsf{u})+f\left(\mathsf{u}-\mathsf{u}^{(1)}+\frac{1}{2}\mathsf{u}^{(2)}\right) \\
\mathsf{u}^{(3)} &= -\frac{\Delta t}{\Delta x_e}\mathsf{Df}^{(2)} \\
\mathsf{f}^{(3)} &= \frac{1}{2}\left[f\left(\mathsf{u}+2\mathsf{u}^{(1)}+\frac{2^2}{2!}\mathsf{u}^{(2)}+\frac{2^3}{3!}\mathsf{u}^{(3)}\right)-2f\left(\mathsf{u}+\mathsf{u}^{(1)}+\frac{1}{2!}\mathsf{u}^{(2)}+\frac{1}{3!}\mathsf{u}^{(3)}\right)\right. \\
&\qquad \left.+2f\left(\mathsf{u}-\mathsf{u}^{(1)}+\frac{1}{2!}\mathsf{u}^{(2)}-\frac{1}{3!}\mathsf{u}^{(3)}\right)-f\left(\mathsf{u}-2\mathsf{u}^{(1)}+\frac{2^2}{2!}\mathsf{u}^{(2)}-\frac{2^3}{3!}\mathsf{u}^{(3)}\right)\right] \\
\mathsf{F} &= \mathsf{f}+\frac{1}{2}\mathsf{f}^{(1)}+\frac{1}{6}\mathsf{f}^{(2)}+\frac{1}{24}\mathsf{f}^{(3)}
\end{aligned}
$$

Numerical flux

RK scheme: Use the trace values of solution

$$f_{e+\frac{1}{2}}(t) = f(u^-_{e+\frac{1}{2}}(t), u^+_{e+\frac{1}{2}}(t)) \qquad \text{(Approximate/Riemann solver)}$$

LW scheme:

- need numerical flux for time average flux F
- we build F at solution points
- use this F for the numerical flux

Numerical flux [14]

We need a numerical flux for the time average flux F

$$F(u) = f(u) + \frac{\Delta t}{2}\partial_t f(u) + \ldots + \frac{\Delta t^N}{(N+1)!}\partial_t^N f(u) = f(u) + \tilde{F}(u)$$

Numerical flux

$$F_{e+\frac{1}{2}} = f(u^-_{e+\frac{1}{2}}, u^+_{e+\frac{1}{2}}) + \frac{1}{2}(\tilde{F}^-_{e+\frac{1}{2}} + \tilde{F}^+_{e+\frac{1}{2}})$$

$f(\cdot,\cdot)$ is any classical two-point numerical flux

Not an upwind flux even if $f(\cdot,\cdot)$ is upwind, due to averaging of higher order terms $\tilde{F}$

Numerical flux: D1 [5]

Rusanov/local-Lax-Friedrich type flux

$$F_{e+\frac{1}{2}} = \frac{1}{2}[F^{-}_{e+\frac{1}{2}} + F^{+}_{e+\frac{1}{2}}] - \frac{1}{2}\lambda_{e+\frac{1}{2}}[u_h(x^{+}_{e+\frac{1}{2}}, t_n) - u_h(x^{-}_{e+\frac{1}{2}}, t_n)] \quad \text{(D1)}$$

Linear advection equation: $u_t + a u_x = 0$

$$\lambda_{e+\frac{1}{2}} = |a|$$

Non-linear PDE like Burger's equation: $u_t + f(u)_x = 0$

$$\lambda_{e+\frac{1}{2}} = \max_i |f'(\overline{u}^n_i)|\} \quad \text{(Lax-Friedrich)}$$

$$\lambda_{e+\frac{1}{2}} = \max\{|f'(\overline{u}^n_e)|, |f'(\overline{u}^n_{e+1})|\} \quad \text{(Rusanov/LLF)}$$

$\overline{u}^n_e$ is the cell average solution in element Ω_e at time t_n.

Dissipation term evaluated at t_n, central part uses time average flux
Not an upwind flux; reduced CFL numbers
Depends on the following quantities:

$$\overline{u}^n_e, \overline{u}^n_{e+1}, u_h(x^{-}_{e+\frac{1}{2}}, t_n), u_h(x^{+}_{e+\frac{1}{2}}, t_n), F^{-}_{e+\frac{1}{2}}, F^{+}_{e+\frac{1}{2}}$$

Numerical flux: D2

Rusanov/Lax-Friedrich type flux

$$F_{e+\frac{1}{2}} = \frac{1}{2}[F^-_{e+\frac{1}{2}} + F^+_{e+\frac{1}{2}}] - \frac{1}{2}\lambda_{e+\frac{1}{2}}[U^+_{e+\frac{1}{2}} - U^-_{e+\frac{1}{2}}] \tag{D2}$$

where

$$U = u + \frac{\Delta t}{2}\partial_t u + \ldots + \frac{\Delta t^N}{(N+1)!}\partial_t^N u$$

is time average solution.

Upwind flux for linear advection: $u_t + a u_x = 0$

$$F_{e+\frac{1}{2}} = \begin{cases} F^-_{e+\frac{1}{2}} & a > 0 \\ F^+_{e+\frac{1}{2}} & a \leq 0 \end{cases}$$

Numerical flux depends on the following quantities:

$$\overline{u}^n_e, \overline{u}^n_{e+1}, U^-_{e+\frac{1}{2}}, U^+_{e+\frac{1}{2}}, F^-_{e+\frac{1}{2}}, F^+_{e+\frac{1}{2}}$$

Numerical flux: central part

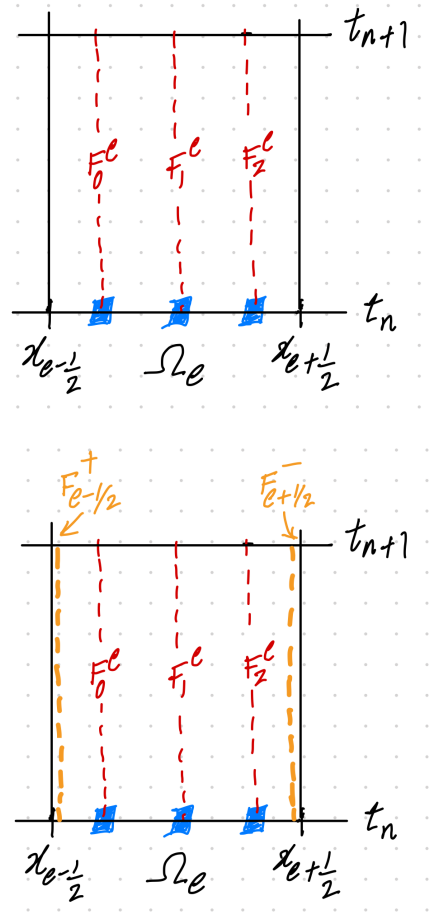
$$F_{e+\frac{1}{2}} = \frac{1}{2}[F^{-}_{e+\frac{1}{2}} + F^{+}_{e+\frac{1}{2}}] - \dots$$

1. **Average-Extrapolate (AE)** to the faces

$$F^{\pm}_{e+\frac{1}{2}} = F^{\delta}_{h}(x^{\pm}_{e+\frac{1}{2}})$$

Sub-optimal accuracy for some non-linear problems, robustness issues

2. **Extrapolate-Average (EA)** flux at the element faces similar to solution points



- Even for linear problems $f(x,u) = a(x)u(x)$

$$I_h(au_h) \neq I_h(a)I_h(u_h) \quad \text{at faces}$$

- For GLL points, the two methods are identical

Algorithm 2: LWFR scheme

$t = 0$;
while $t < T$ **do**
    Loop over cells and assemble rhs;
    Loop over faces and assemble rhs;
    Update solution to next time level;
    Perform MPI data exchange;
    Apply a posteriori limiter;
    Apply positivity limiter;
    Perform MPI data exchange;
    $t = t + \Delta t$;

Fewer MPI communication
Fewer applications of limiter
Cell computations more than in RK
Potential to hide MPI communication behind computations

ADER-DG scheme [9]

- Current solution

$$x \in \Omega_e : \qquad u_h^n(\xi) = \sum_{j=0}^{N} u_j^e(t_n)\ell_j(\xi)$$

- Cell local space-time solution and flux: $\tau = (t - t_n)/\Delta t$

$$\tilde{u}_h(\xi,\tau) = \sum_{k=0}^{N}\sum_{j=0}^{N} \tilde{u}_{jk}^e \ell_j(\xi)\ell_k(\tau), \quad \tilde{f}_h(\xi,\tau) = \sum_{k=0}^{N}\sum_{j=0}^{N} f(\tilde{u}_{jk}^e)\ell_j(\xi)\ell_k(\tau)$$

ADER-DG scheme [9]

- Find $\tilde{u}_h$, cell local Galerkin method: For $0 \leq r, s \leq N$

$$\int_{t_n}^{t_{n+1}} \int_{\Omega_e} \left[\partial_t \tilde{u}_h + \partial_x \tilde{f}_h \right] \ell_r(\xi) \ell_s(\tau) \mathrm{d}x \mathrm{d}t = 0$$

Integrate by parts wrt time in first term

$$\int_{\Omega_e} \tilde{u}_h(\xi, 1) \ell_r(\xi) \ell_s(1) \mathrm{d}x + \int_{t_n}^{t_{n+1}} \int_{\Omega_e} \left[-\tilde{u}_h \ell_r(\xi) \partial_t \ell_s(\tau) + (\partial_x \tilde{f}_h) \ell_r(\xi) \ell_s(\tau) \right] \mathrm{d}x \mathrm{d}t$$
$$= \int_{\Omega_e} u_h^n \ell_r(\xi) \ell_s(0) \mathrm{d}x$$

Non-linear system with $(N+1)^2$ unknowns[3]: picard iterations

ADER-DG scheme [9]

- Update solution, $u_h^n \to u_h^{n+1}$: For $0 \le i \le N$

$$\int_{\Omega_e} u_h^{n+1} \ell_i(\xi) \mathrm{d}x = \int_{\Omega_e} u_h^n \ell_i(\xi) \mathrm{d}x - \int_{t_n}^{t_{n+1}} \int_{\Omega_e} \tilde{f}_h \partial_x \ell_i(\xi) \mathrm{d}x \mathrm{d}t$$
$$+ \int_{t_n}^{t_{n+1}} f(\tilde{u}^-_{e-\frac{1}{2}}(t), \tilde{u}^+_{e-\frac{1}{2}}(t)) \ell_i(0) \mathrm{d}t$$
$$- \int_{t_n}^{t_{n+1}} f(\tilde{u}^-_{e+\frac{1}{2}}(t), \tilde{u}^+_{e+\frac{1}{2}}(t)) \ell_i(1) \mathrm{d}t$$

[3] In 3-d, we have $(N+1)^4$ unknowns.

Fourier stability analysis in 1-D

Linear advection equation

$$u_t + a u_x = 0$$

CFL number

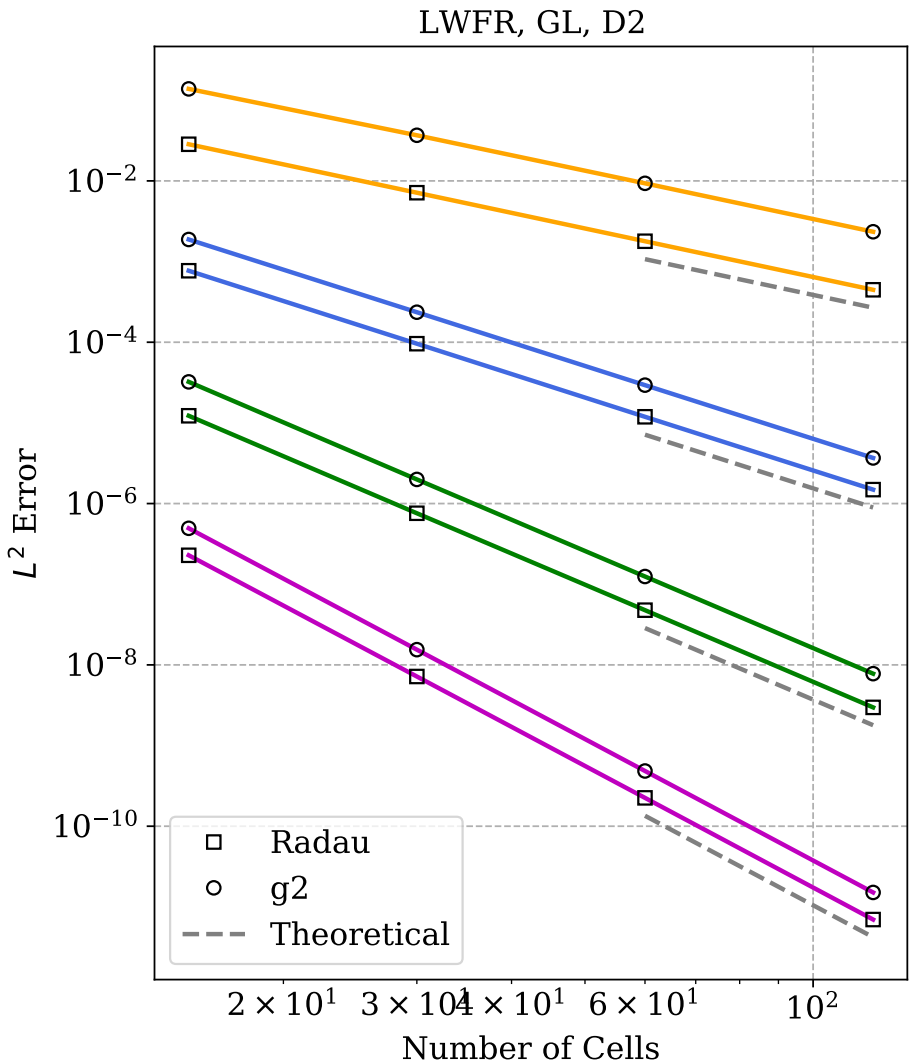
$$\mathrm{CFL} = \frac{|a|\Delta t}{\Delta x}$$

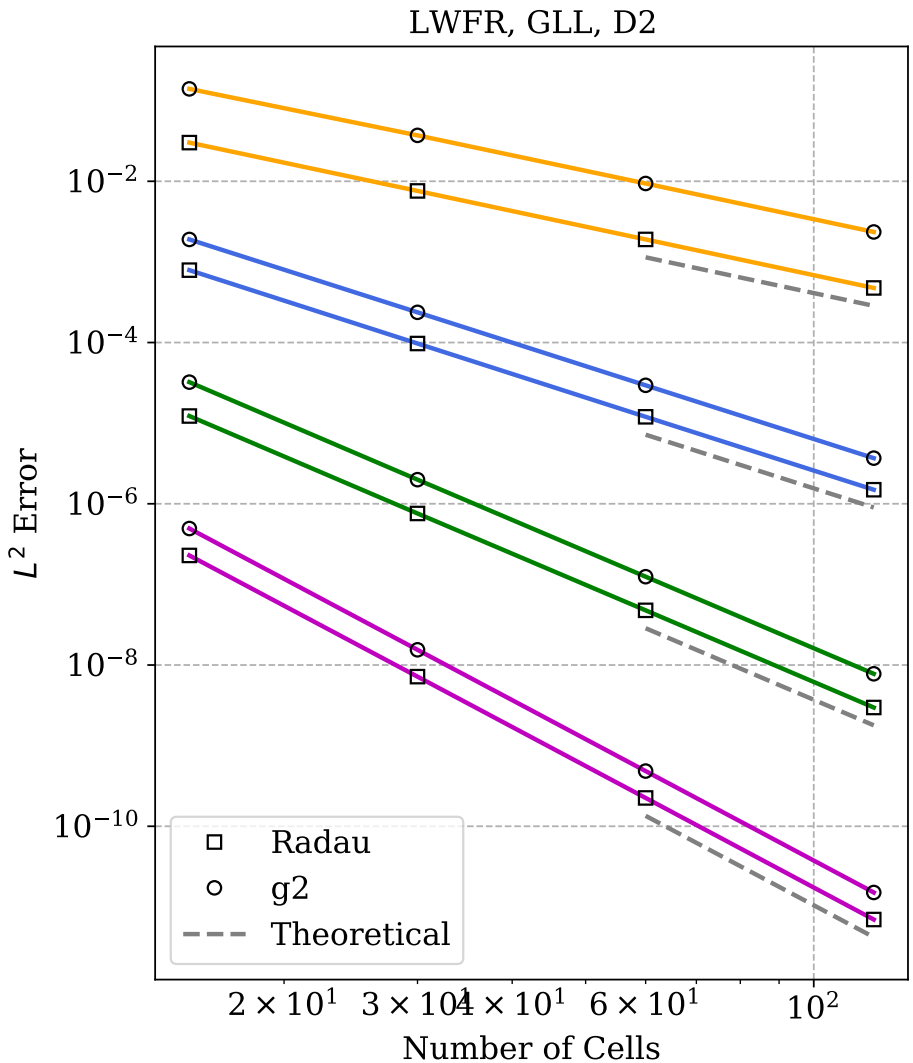
N	Radau				g_2			
	D1	D2	Ratio	RKFR	D1	D2	Ratio	RKFR
1	0.226	0.333	1.47	0.333	0.465	1.000	2.15	1.000
2	0.117	0.170	1.45	0.210	0.204	0.333	1.63	0.450
3	0.072	0.103	1.43	0.145	0.116	0.170	1.47	0.287
4	0.049	0.069	1.40	0.116	0.060	0.103	1.72	0.212

- D1 is not upwind flux
- D2 is upwind flux
- D2 cfls are same as those of ADER-DG [9]

Numerical results: 1-D, scalar

Linear advection: convergence $\|u - u_h\|_{L^2} = O(h^{N+1})$

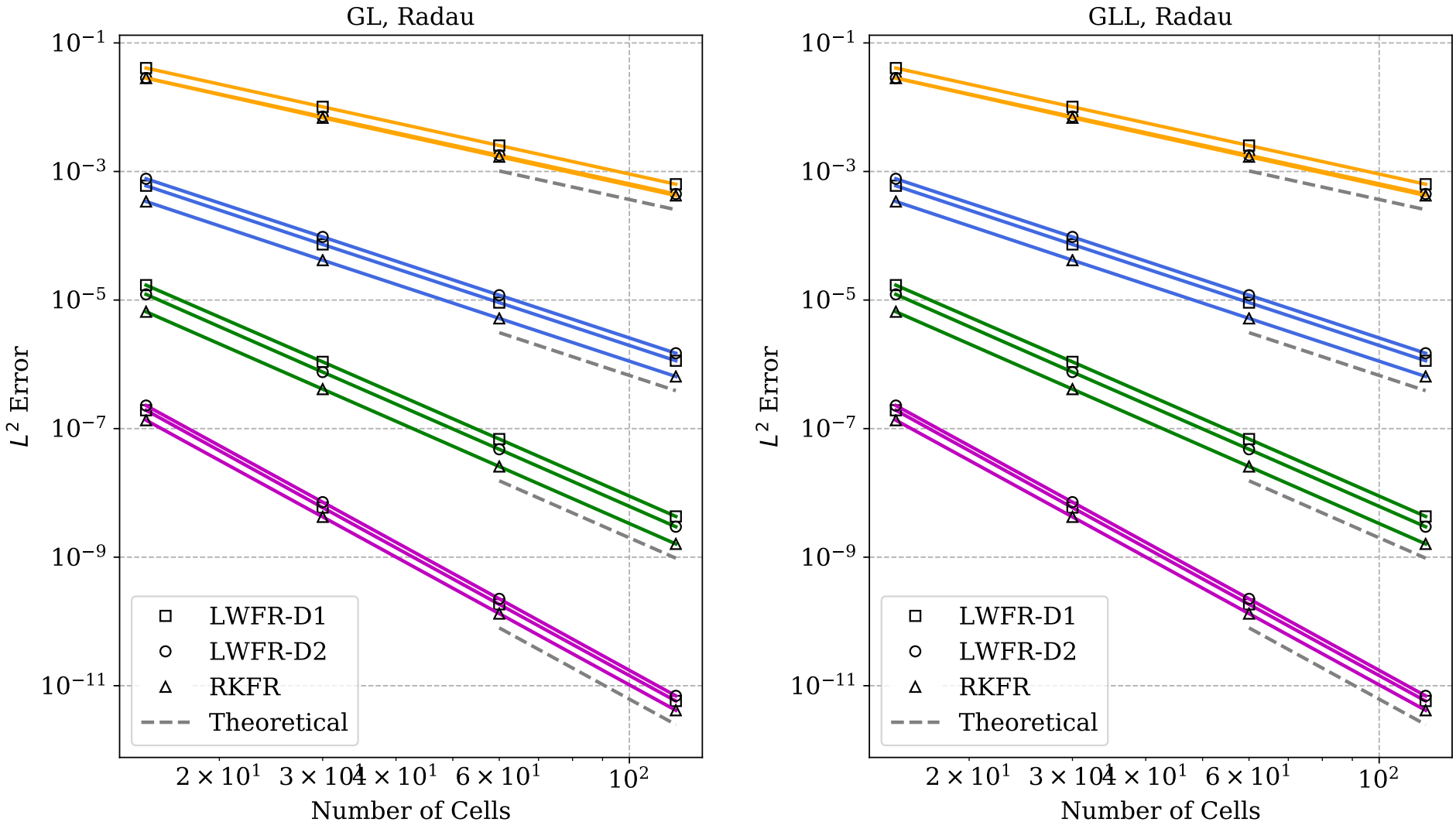




Radau vs g2: g2 is less accurate

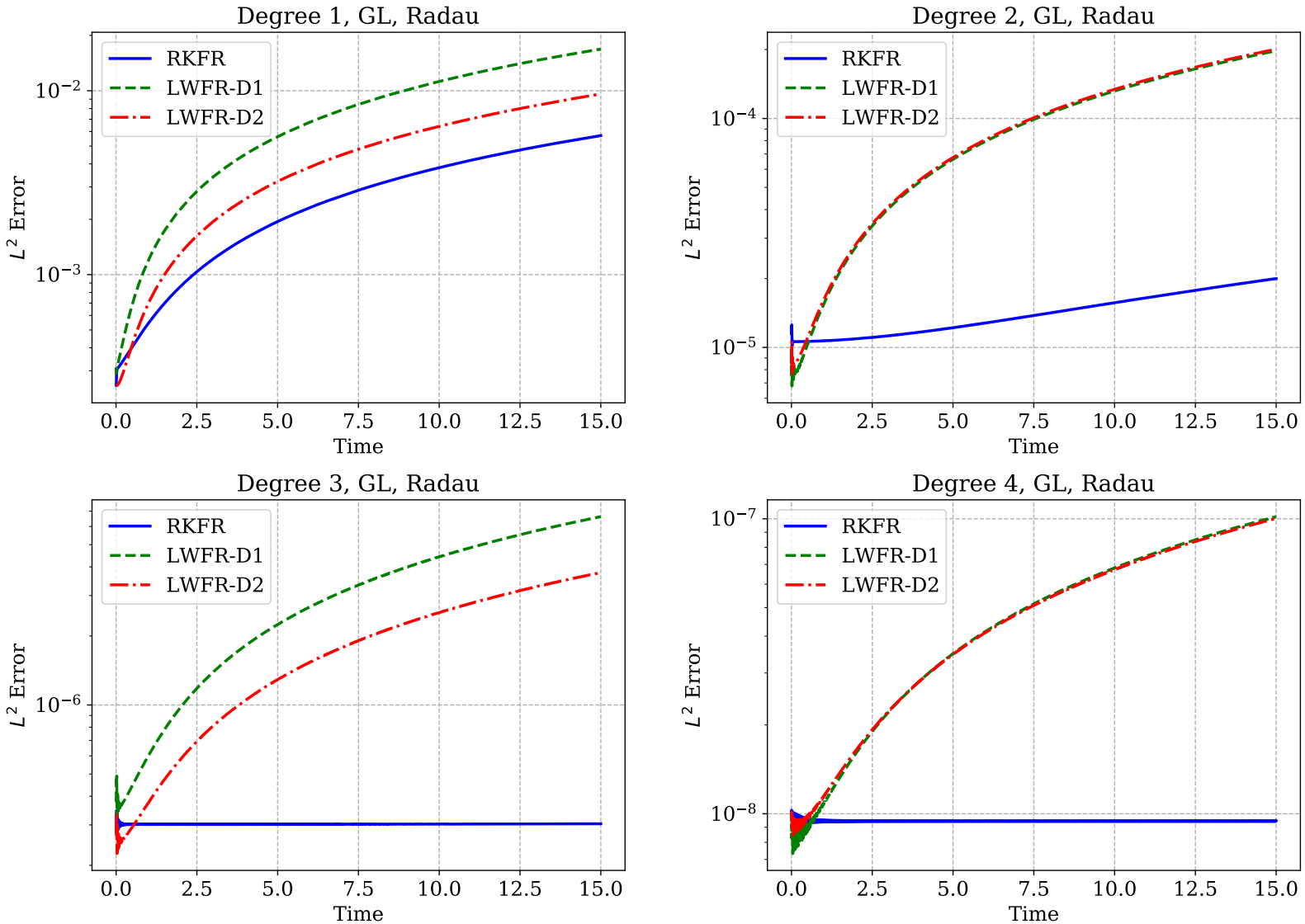
GL vs GLL: no significant difference

Linear advection: convergence

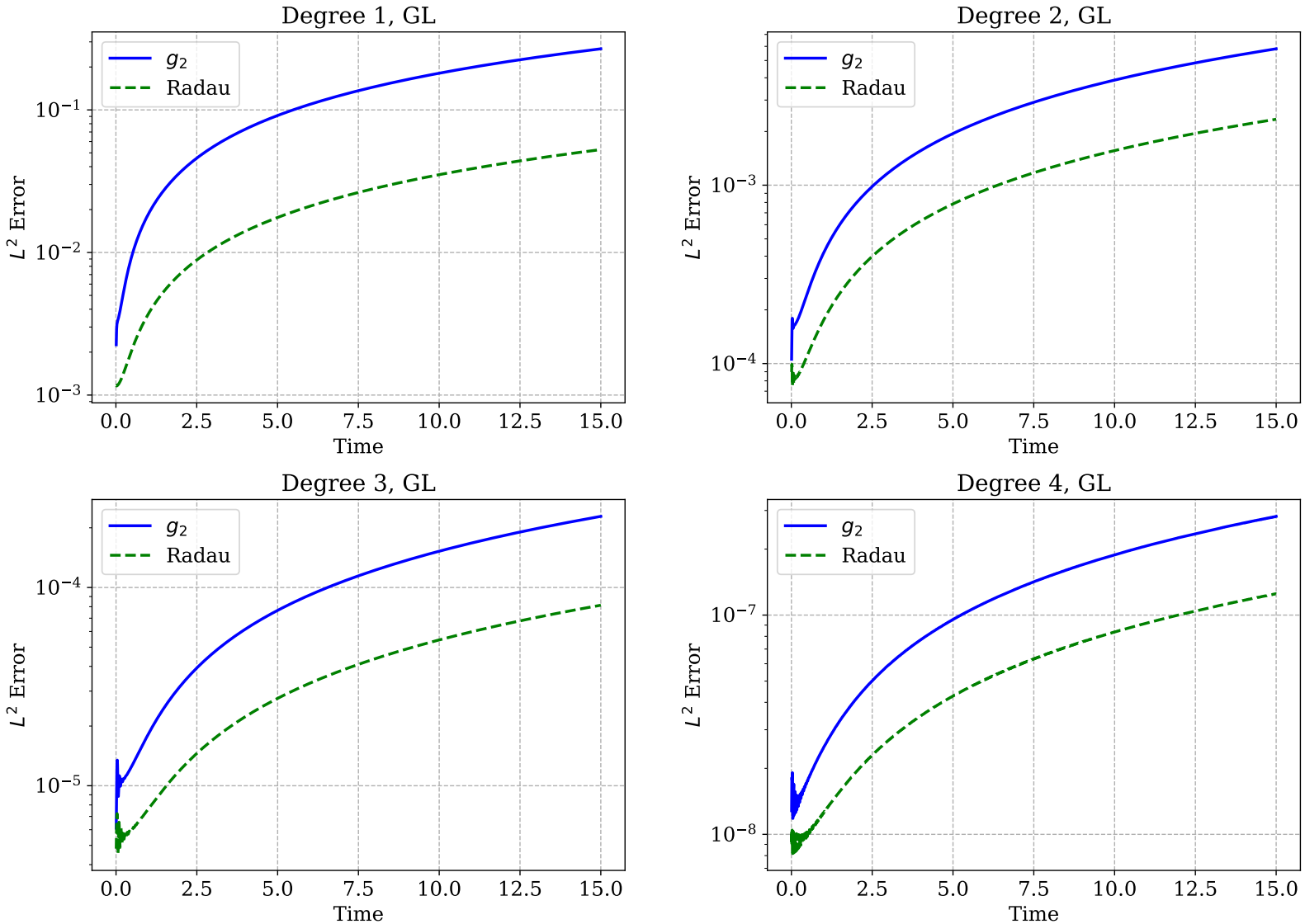


RKFR bit more accurate than LWFR

Linear advection: $u(x,0) = \sin(2\pi x)$, 120 dofs



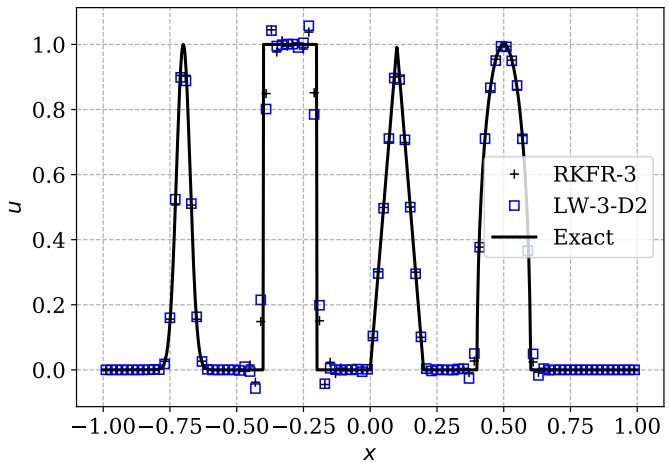
Linear advection: $u(x,0) = \sin(2\pi x)$, 120 dofs

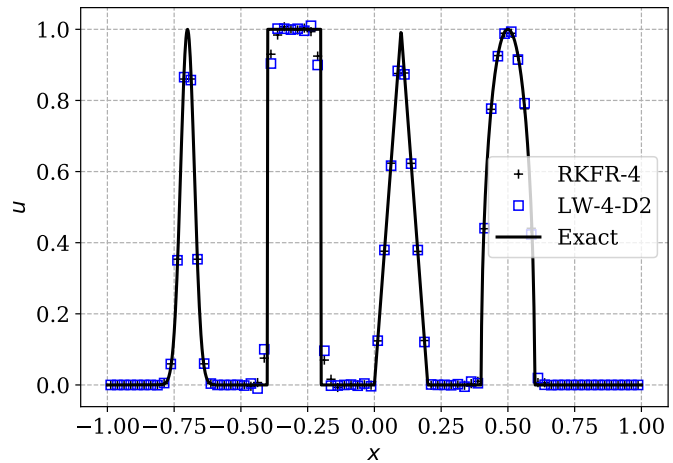


Linear advection: composite signal

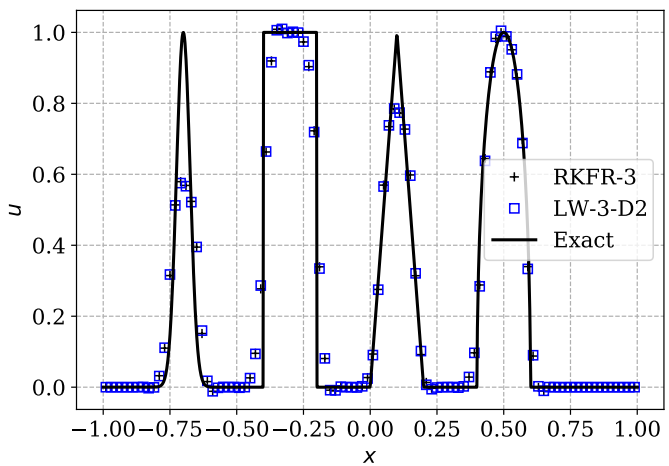
$t = 8$ (4 periods), 400 dofs

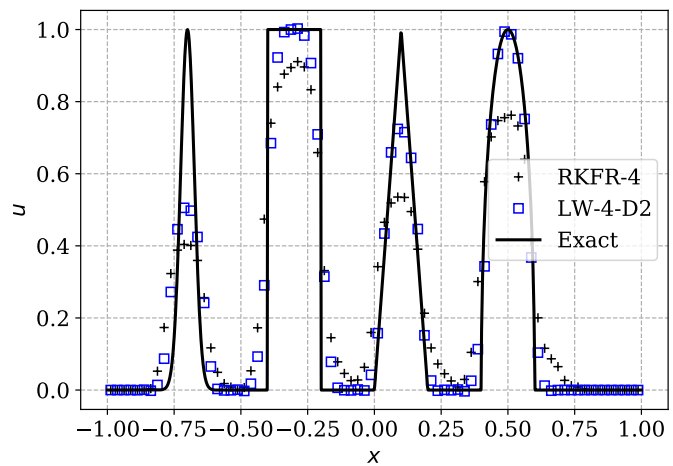
Un-limited



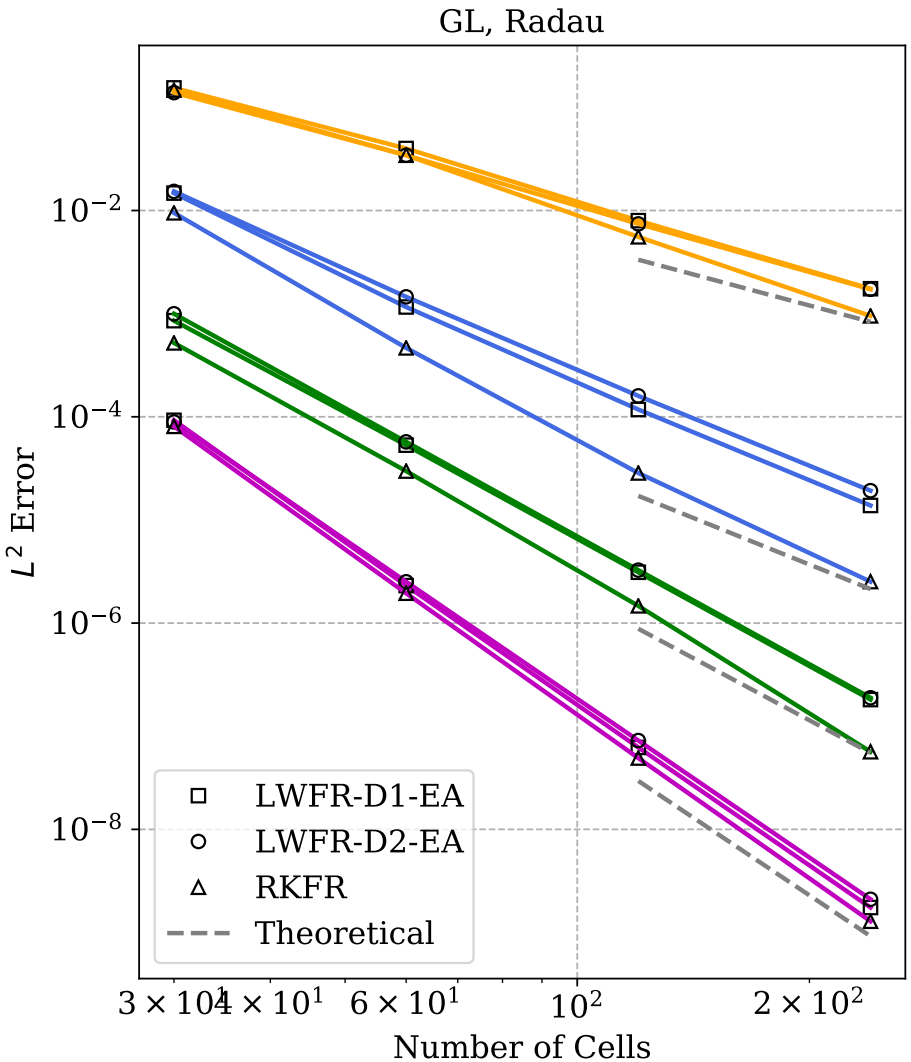


Limited

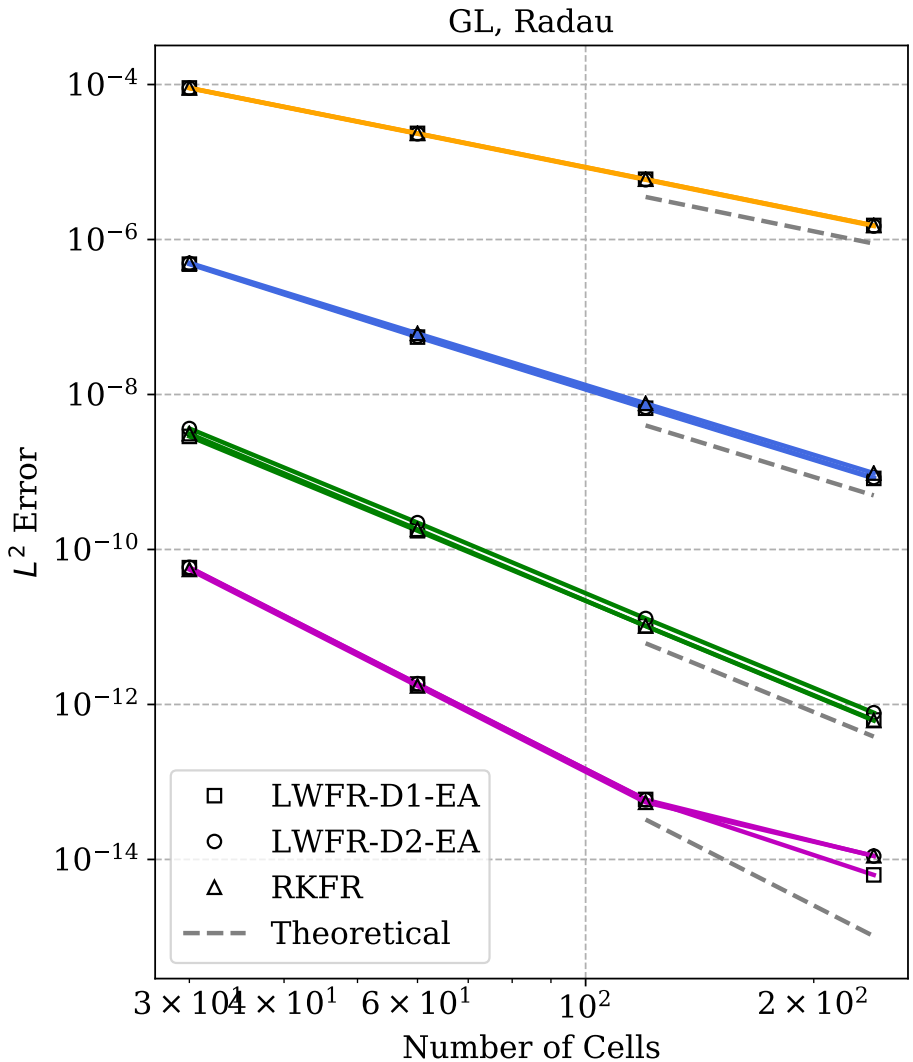




Variable linear advection: $u_t + (a(x)u)_x = 0$

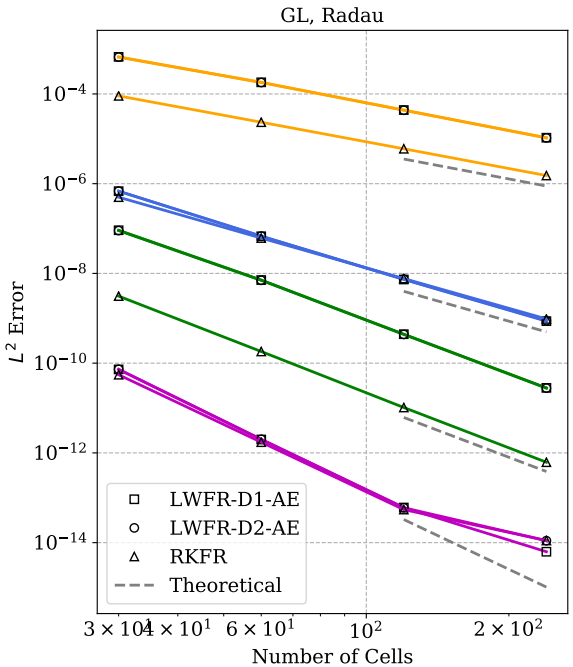


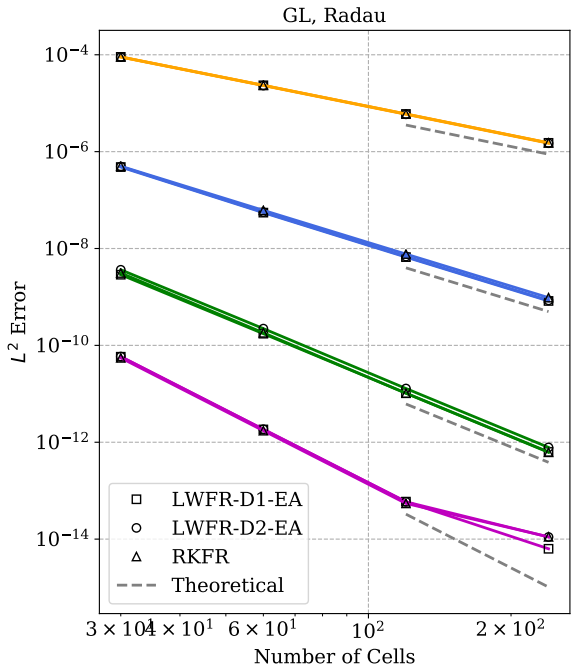
$a(x) = x$

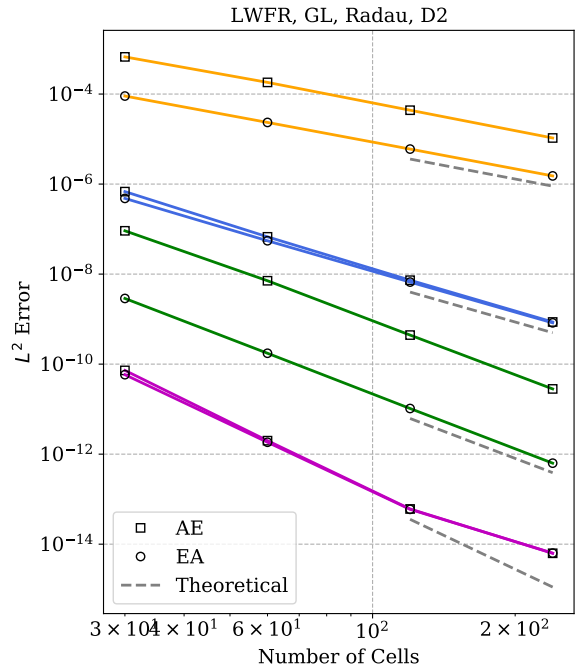


$a(x) = x^2$

Variable linear advection: $a(x) = x^2$







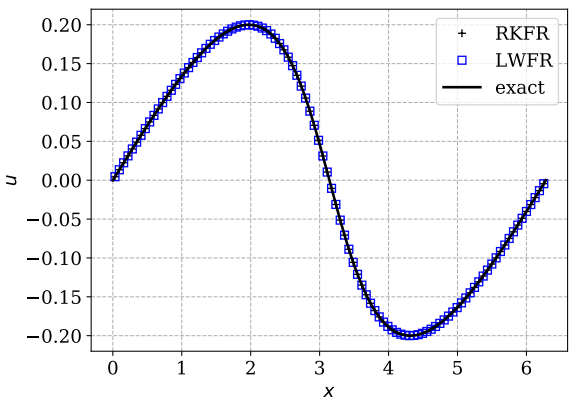
LWFR-AE less accurate than RKFR at odd degrees

LWFR-EA and RKFR are similar

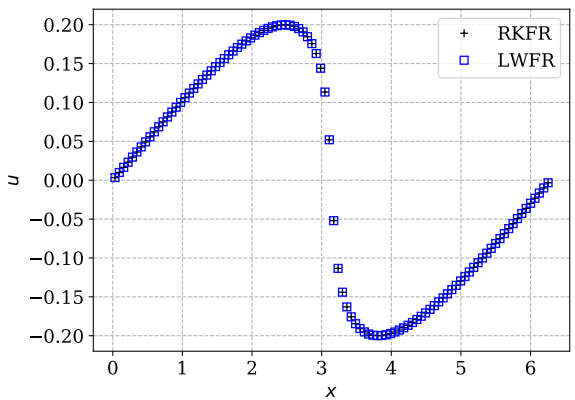
LWFR-AE is less accurate than LWFR-EA at odd degrees

Burger's equation: $u(x,0) = 0.2\sin(x)$

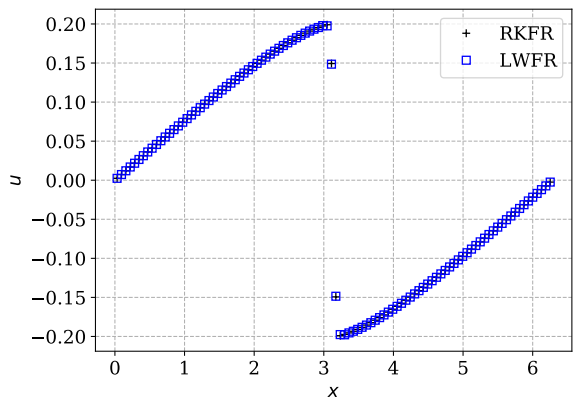
$N = 3$, GL + Radau + D2 + TVB, 200 dofs



$t = 2$

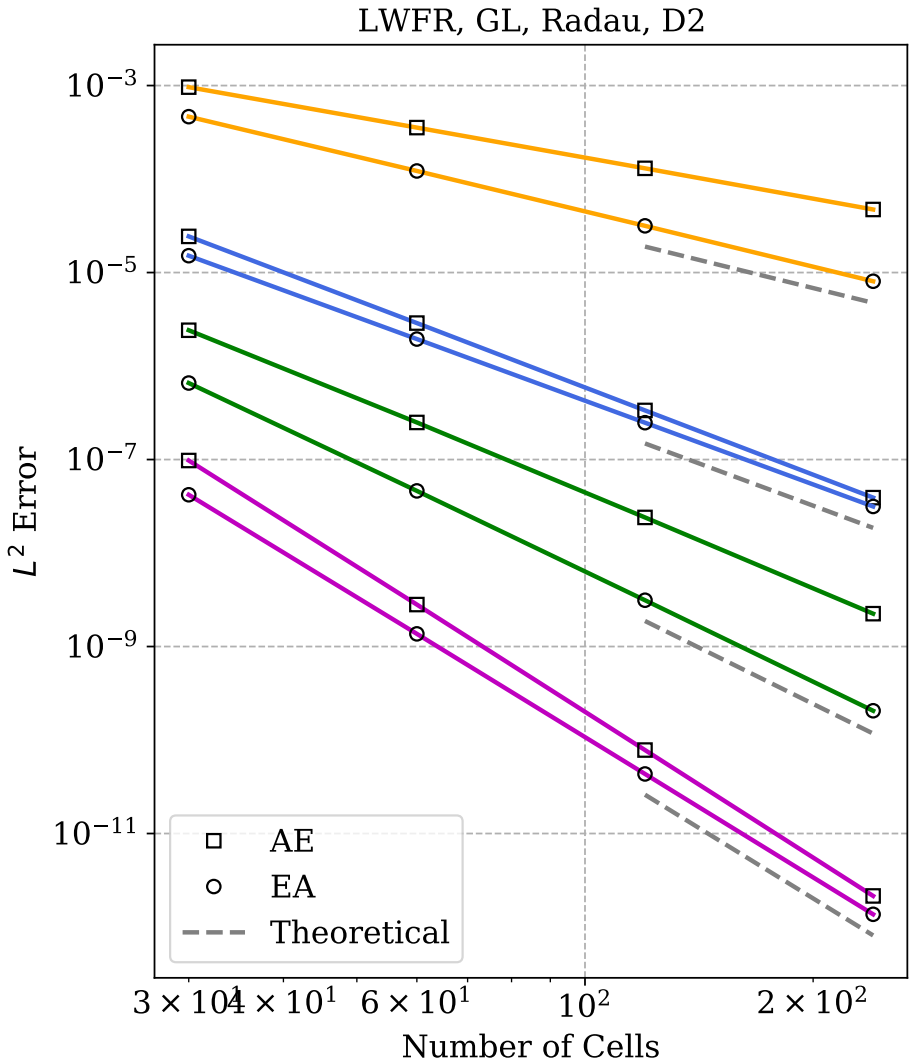


$t = 4.5$



$t = 8$

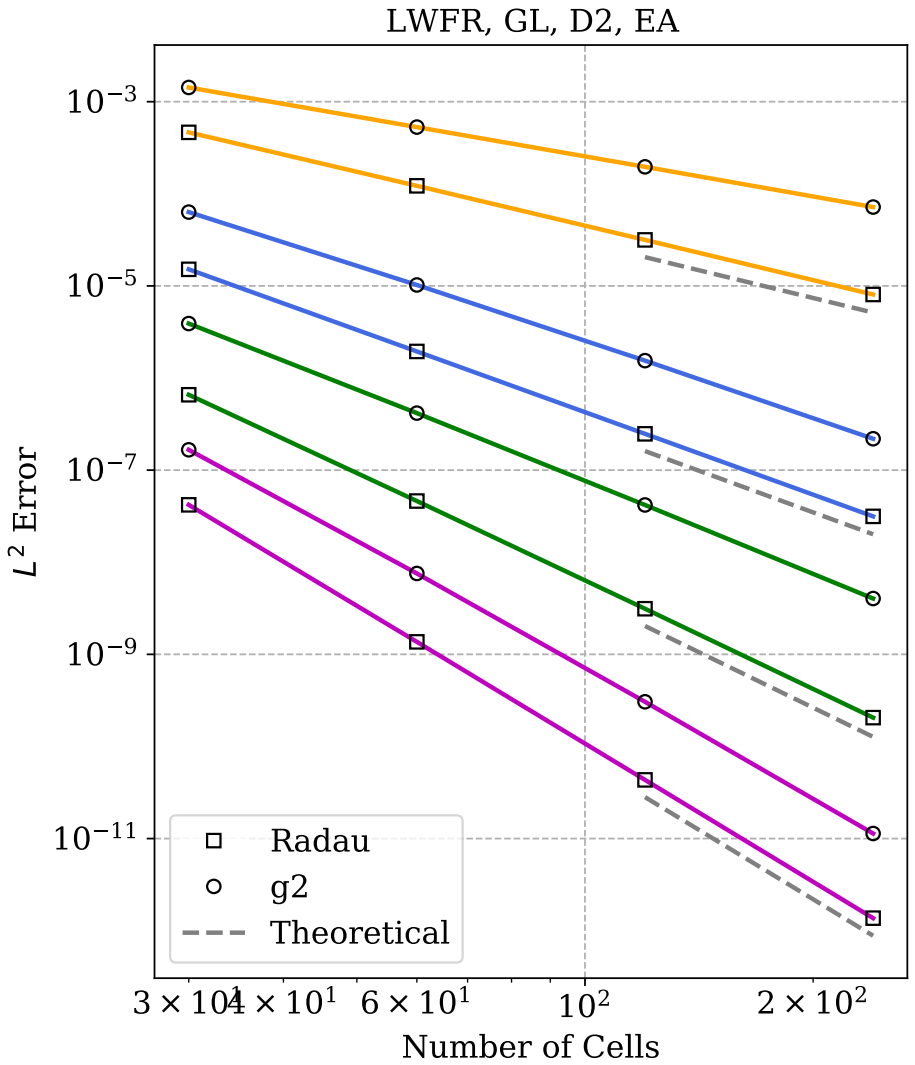
Burger's equation: conv, $u(x,0) = 0.2\sin(x)$, $t = 2$



AE: $N = 1, 3$ lose about half order convergence rate

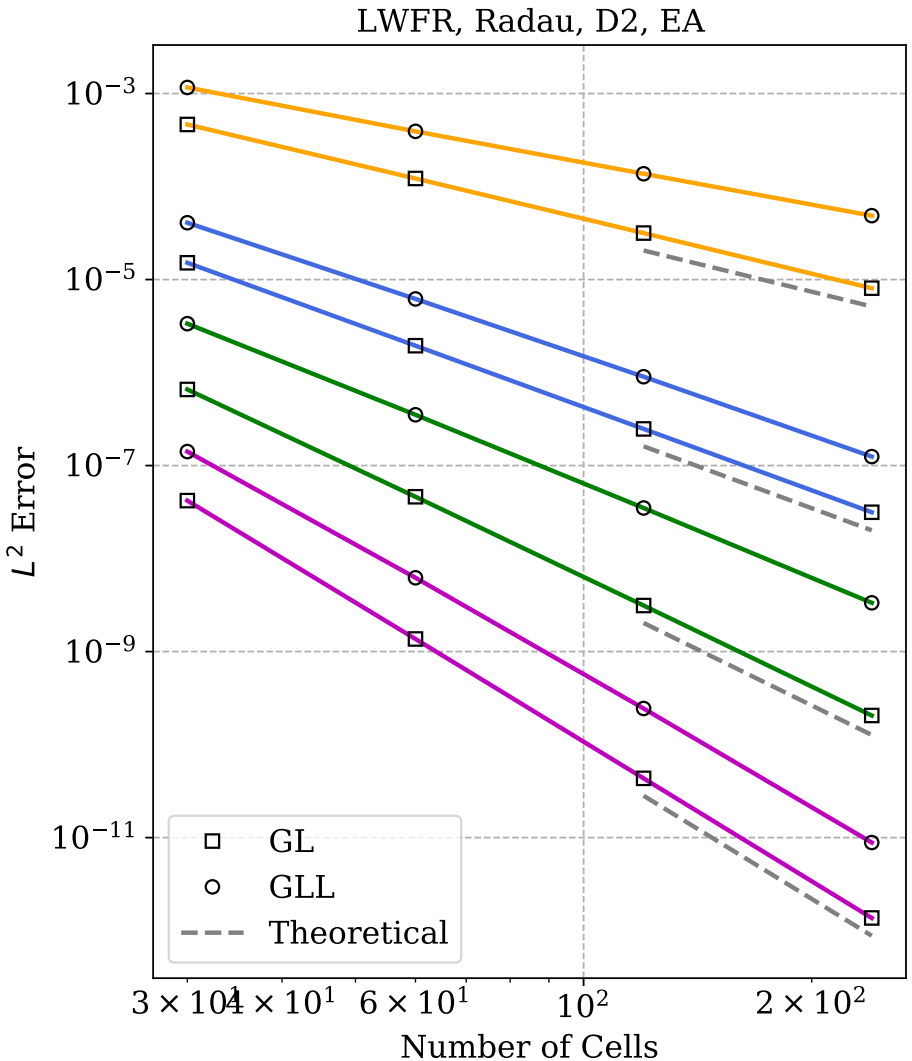
Problem seems related to sonic-point $f'(u) = 0$ and/or non-linearity

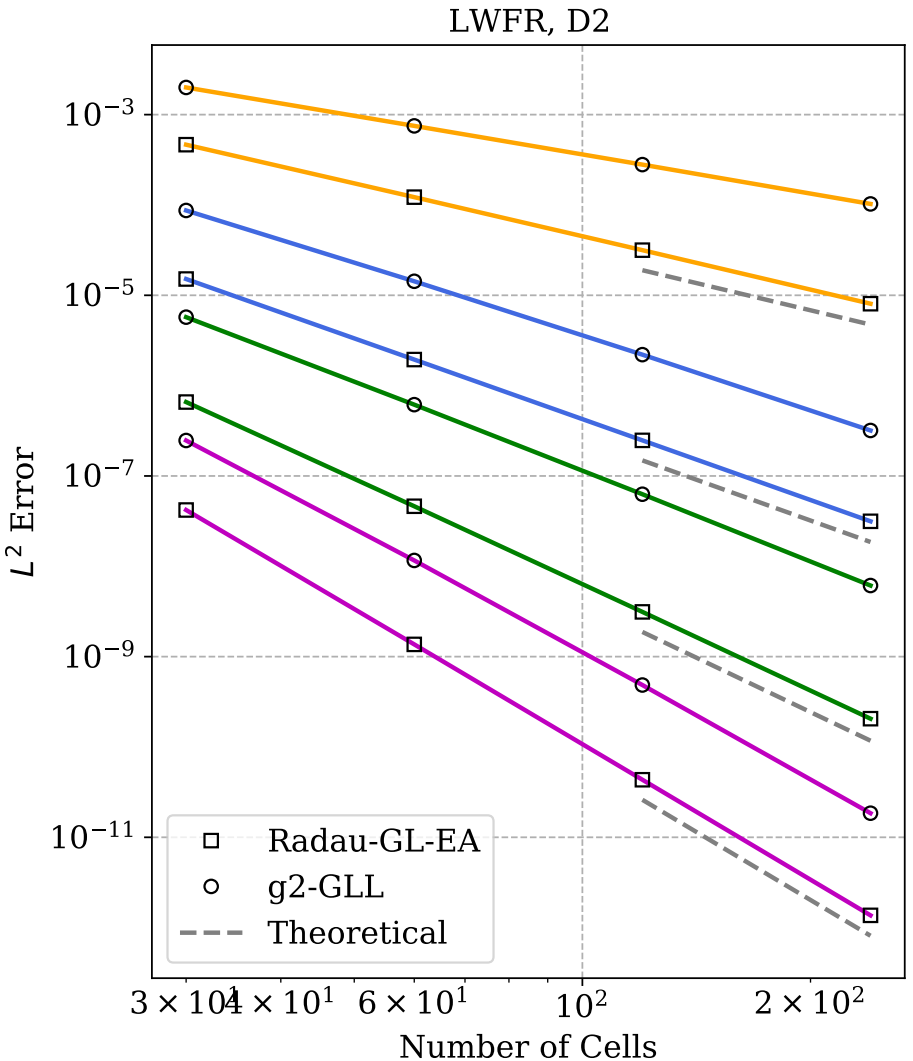
Burger's convergence: EA, Radau vs g2



Even with EA, g2 loses convergence rate at odd degrees

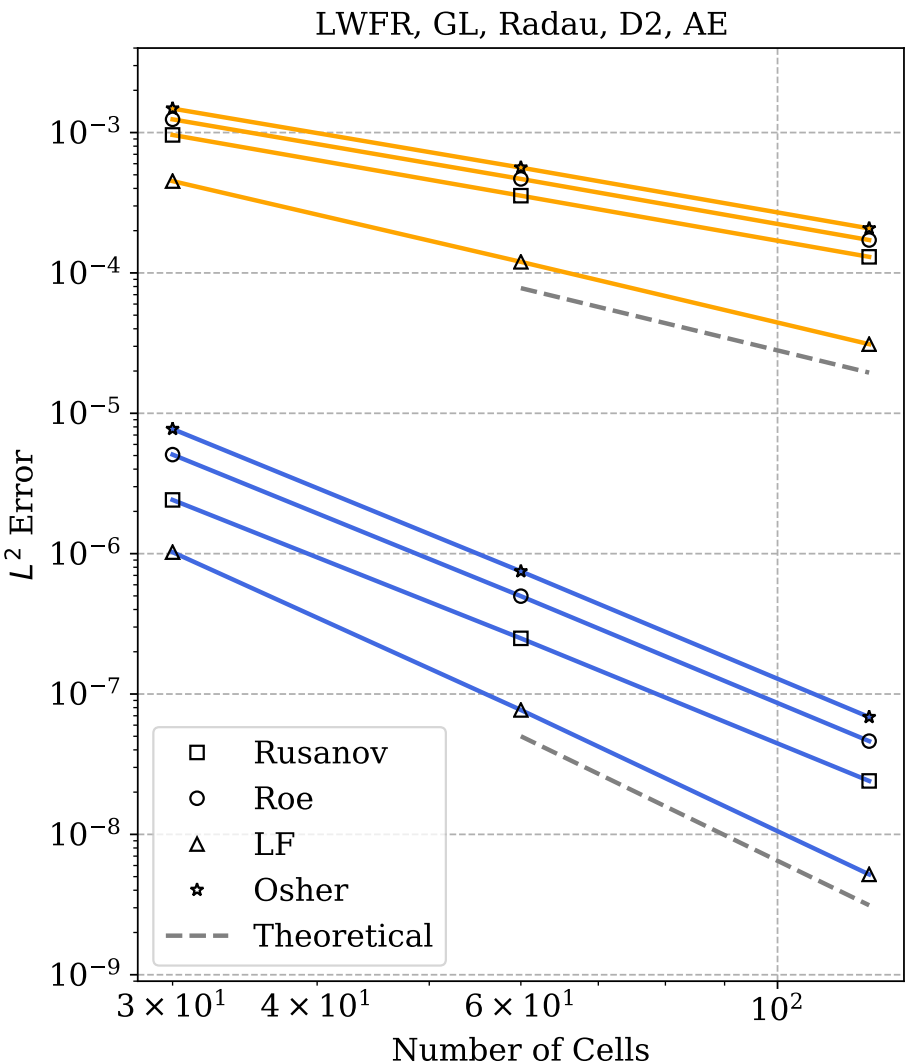
Burger's convergence: GL vs GLL

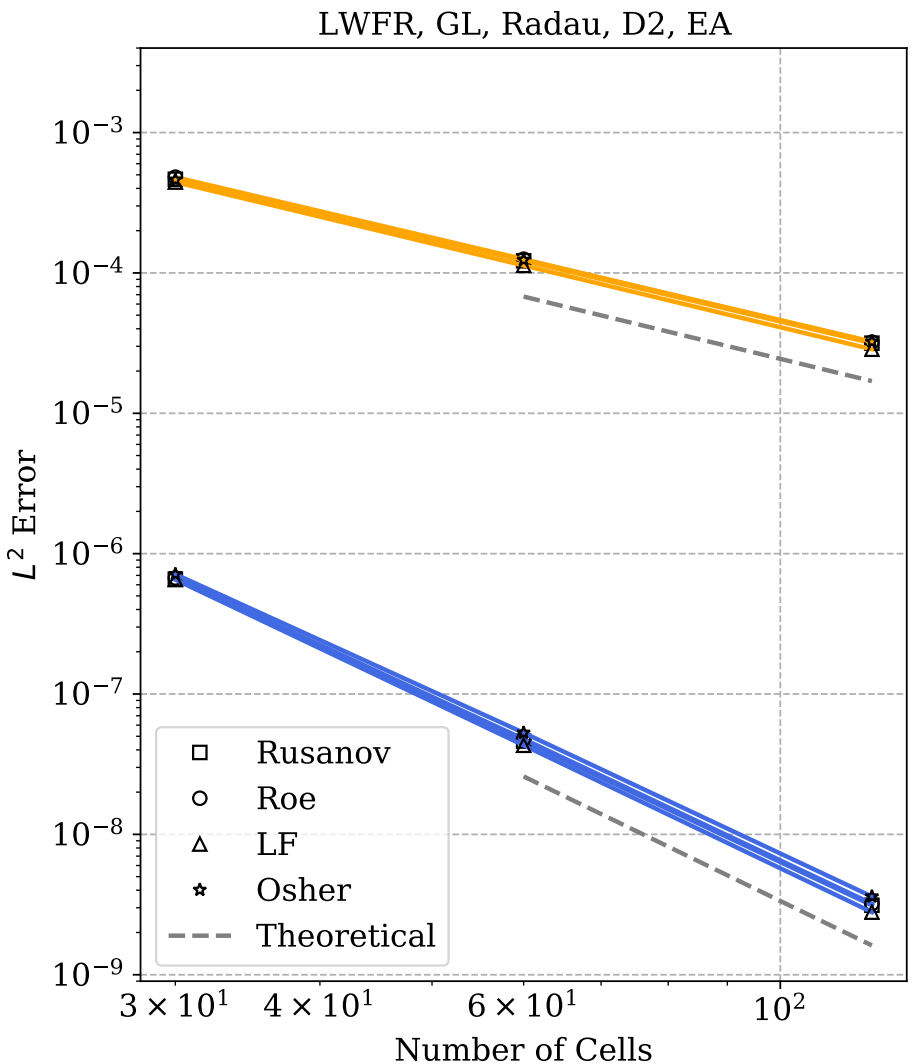




GLL: we cannot fix sub-optimal convergence rates at odd degrees

Burger's convergence: EA vs AE, different fluxes



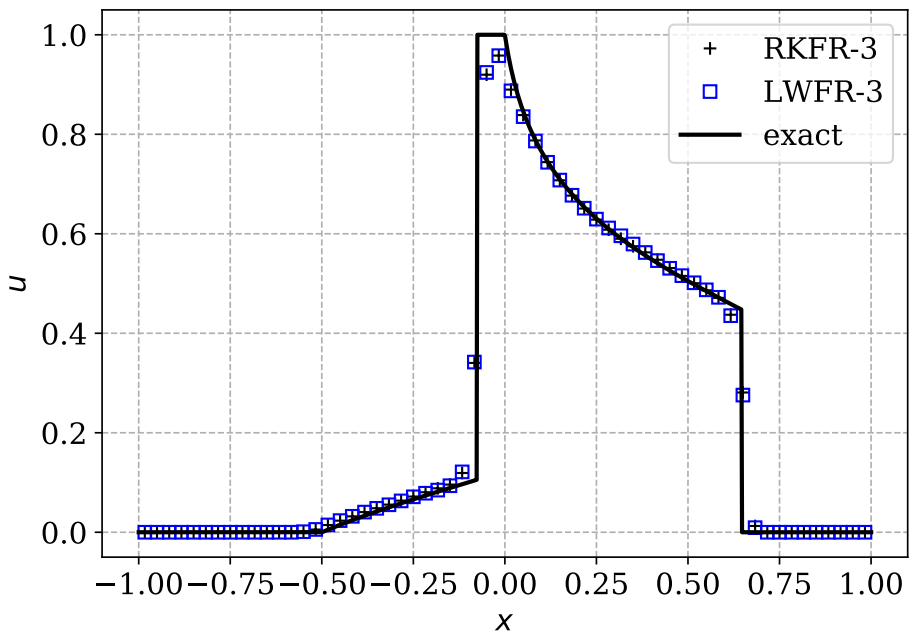


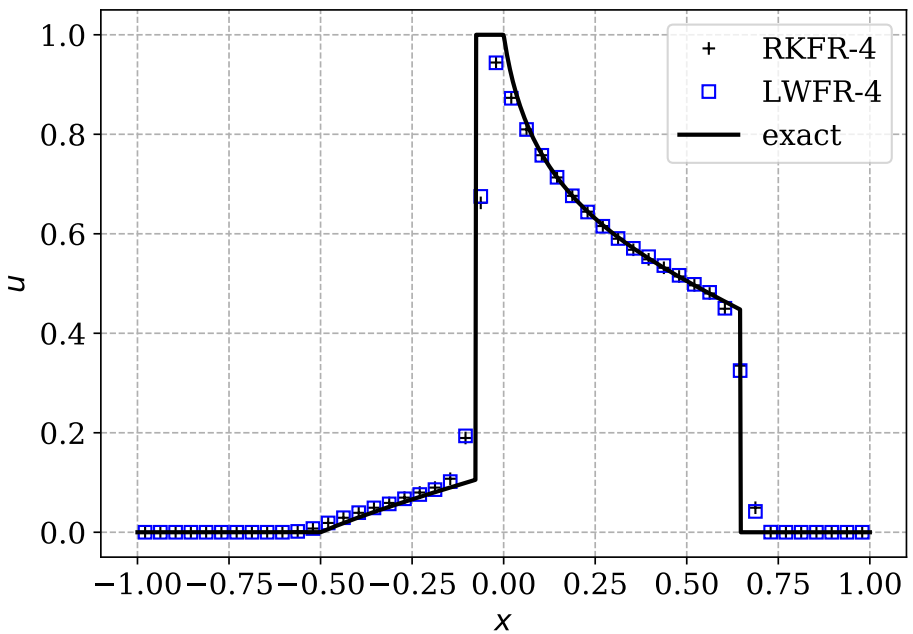
$N = 1, 3$

Buckley-Leverett equation

$$u_t + f(u)_x = 0, \qquad f(u) = \frac{4u^2}{4u^2 + (1-u)^2}$$

GL + Radau, 200 dofs, $t = 0.4$





Euler equations in 1-D

Conservation of: mass, momentum and energy

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho v\\ E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho v\\ p+\rho v^2\\ (E+p)v\end{bmatrix}=0, \qquad E=\frac{p}{\gamma-1}+\frac{1}{2}\rho v^2$$

$\rho =$ density, $\quad v =$ velocity, $\quad p =$ pressure, $\quad E =$ total energy

Rusanov-type flux

$$F_{e+\frac{1}{2}}=\frac{1}{2}[F^-_{e+\frac{1}{2}}+F^+_{e+\frac{1}{2}}]-\frac{1}{2}\lambda_{e+\frac{1}{2}}[U^+_{e+\frac{1}{2}}-U^-_{e+\frac{1}{2}}]$$

$$\lambda_{e+\frac{1}{2}}=\max\{|\bar{v}^n_e|+\bar{c}^n_e,|\bar{v}^n_{e+1}|+\bar{c}^n_{e+1}\}$$

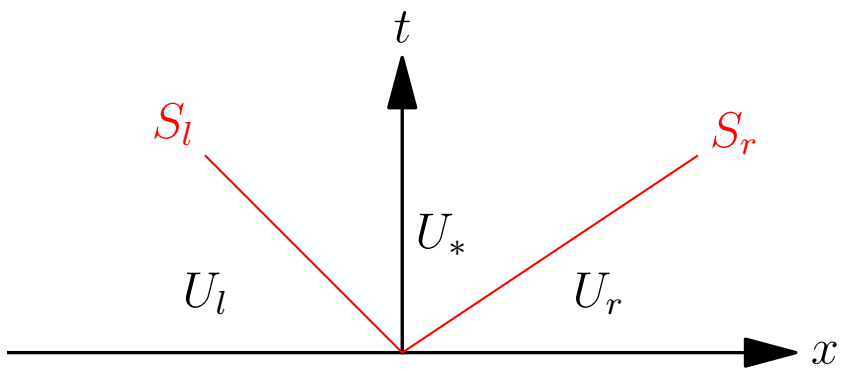
Roe-type flux

$$F_{e+\frac{1}{2}}=\frac{1}{2}[F^-_{e+\frac{1}{2}}+F^+_{e+\frac{1}{2}}]-\frac{1}{2}|A(\overline{u}^n_e,\overline{u}^n_{e+1})|[U^+_{e+\frac{1}{2}}-U^-_{e+\frac{1}{2}}]$$

HLL flux: 2 wave model

Estimate smallest and largest wave speeds

$$S_l = \max\{\lambda_{\min}(\bar{u}_l), \lambda_{\min}(\bar{u})\}, \qquad S_r = \max\{\lambda_{\max}(\bar{u}_r), \lambda_{\max}(\bar{u})\}$$



Jump conditions

$$\begin{aligned} F_* - F_l &= S_l(U_* - U_l) \\ F_r - F_* &= S_r(U_r - U_*) \end{aligned}$$

Solve for the state and flux

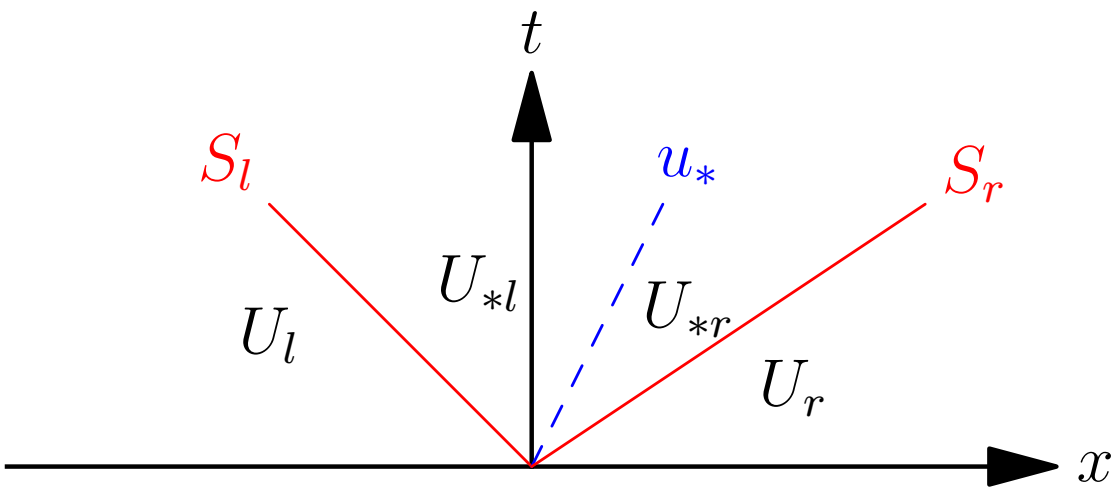
$$U_* = \frac{S_r U_r - S_l U_l - (F_r - F_l)}{S_r - S_l}, \qquad F_* = \frac{S_r F_l - S_l F_r + S_l S_r (U_r - U_l)}{S_r - S_l}$$

HLL flux

$$F(U_l, U_r, F_l, F_r; \bar{u}_l, \bar{u}_r) = \begin{cases} F_l & \text{if } S_l > 0 \\ F_r & \text{if } S_r < 0 \\ F_* & \text{otherwise} \end{cases} \qquad \text{(Upwind flux)}$$

HLLC flux: 3 wave model

Slowest, fastest and contact wave



u_*, U_{*l}, U_{*r} can be determined by satisfying jump conditions across all three waves

$$F(U_l, U_r, F_l, F_r; \bar{u}_l, \bar{u}_r) = \begin{cases} F_l & \text{if } S_l > 0 \\ F_r & \text{if } S_r < 0 \\ F_{*l} = F_l + S_l(U_{*l} - U_l) & \text{if } S_l < 0 < u_* \\ F_{*r} = F_r + S_r(U_{*r} - U_r) & \text{if } u_* < 0 < S_r \end{cases}$$

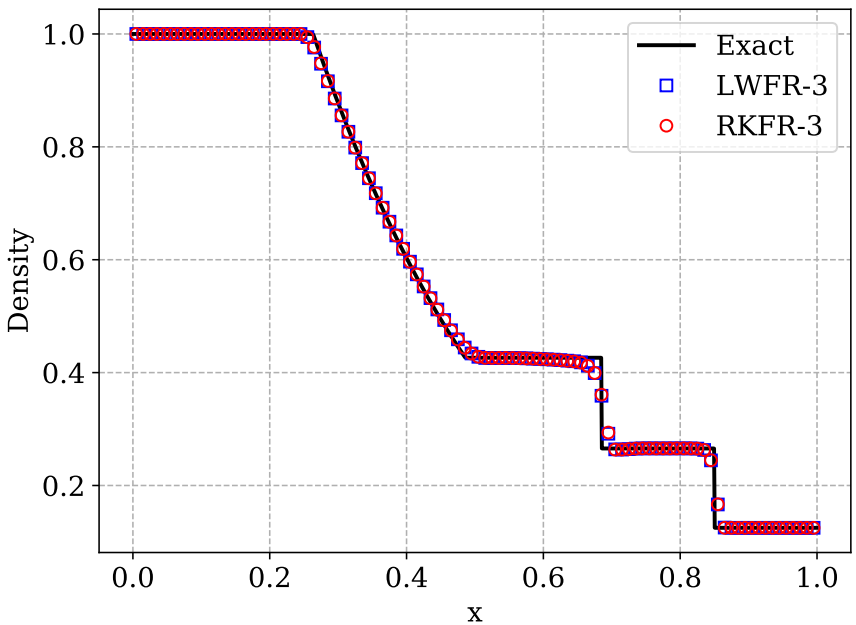
This is an upwind flux.

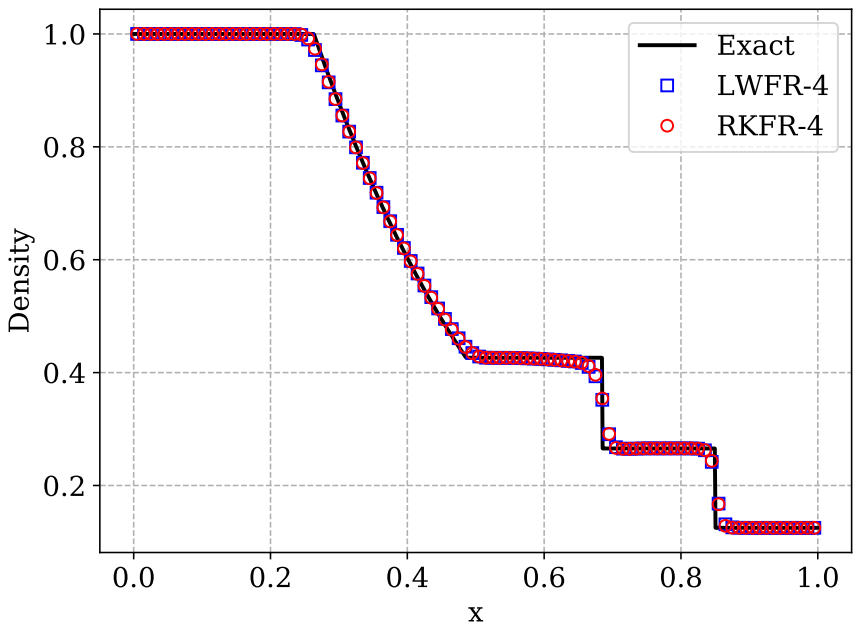
Numerical results: 1-D, systems

1-D Euler: Sod test

$$(\rho, v, p) = \begin{cases} (1.0,\ 0,\ 1) & x \leq 0.5 \\ (0.125,\ 0,\ 0.1) & x > 0.5 \end{cases}$$

GL + Radau + D2 + EA, 100 cells, $t = 0.2$

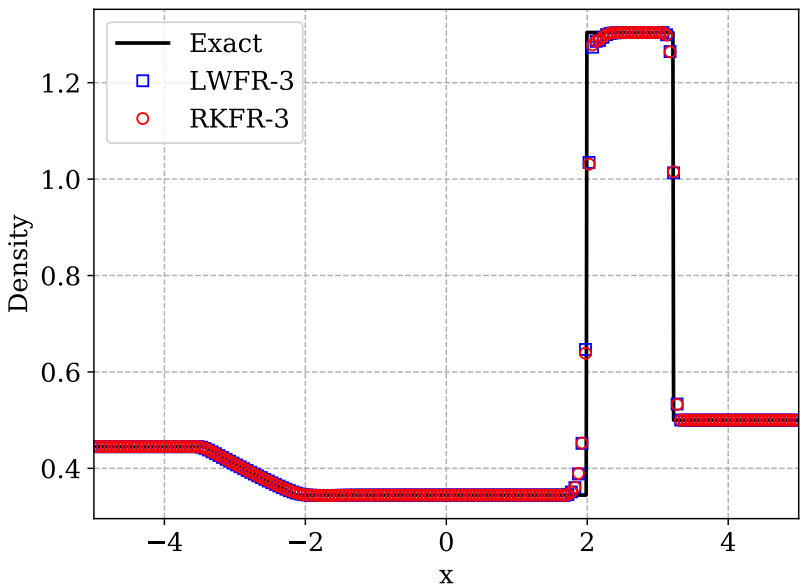


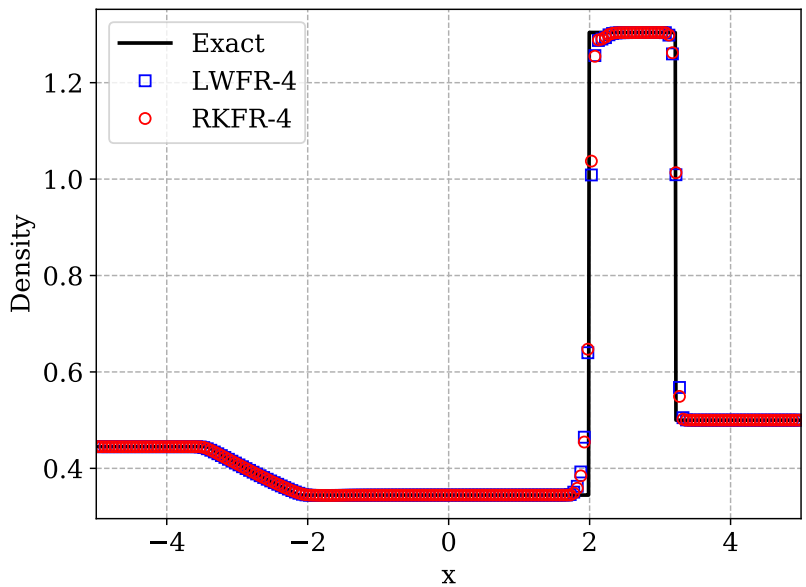


1-D Euler: Lax test

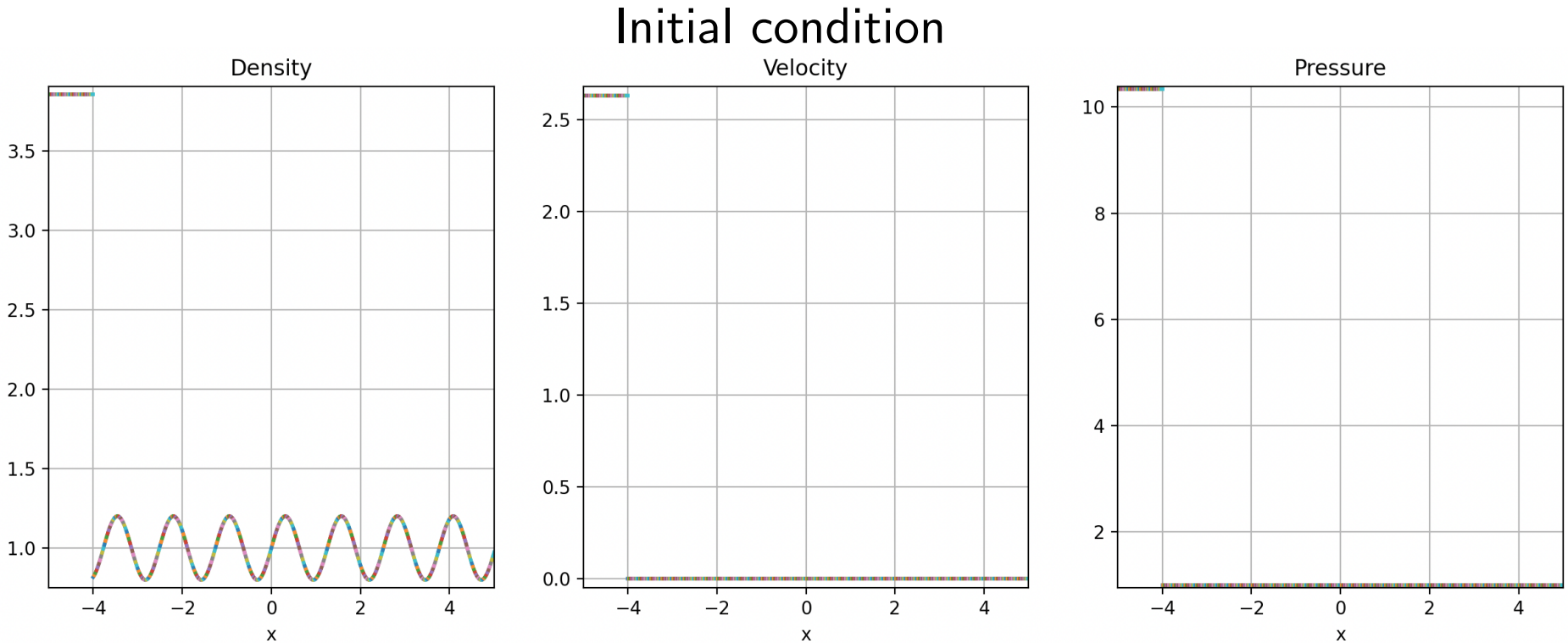
$$(\rho, v, p) = \begin{cases} (0.455,\ 0.689,\ 3.528) & x \leq 0 \\ (0.5,\ 0,\ 0.571) & x > 0 \end{cases}$$

GL + Radau + D2 + EA, 200 cells, $t = 1.3$



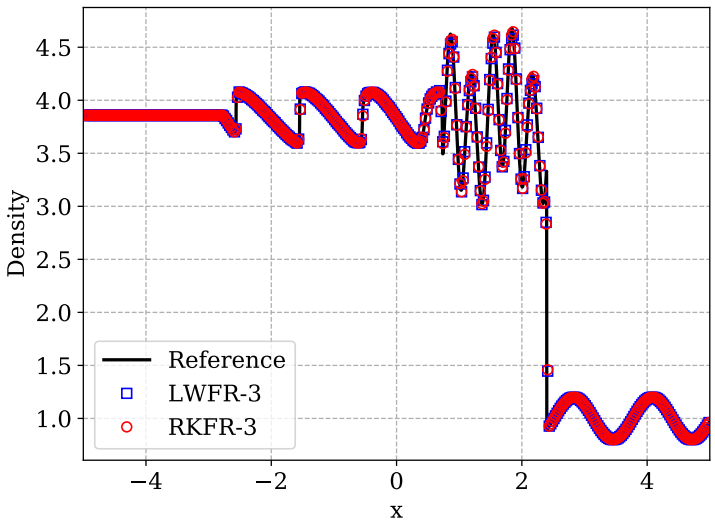


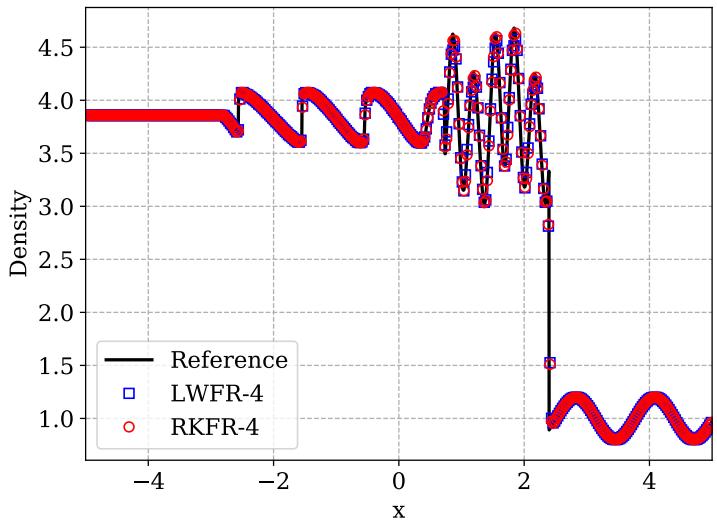
1-D Euler: Shu-Osher test

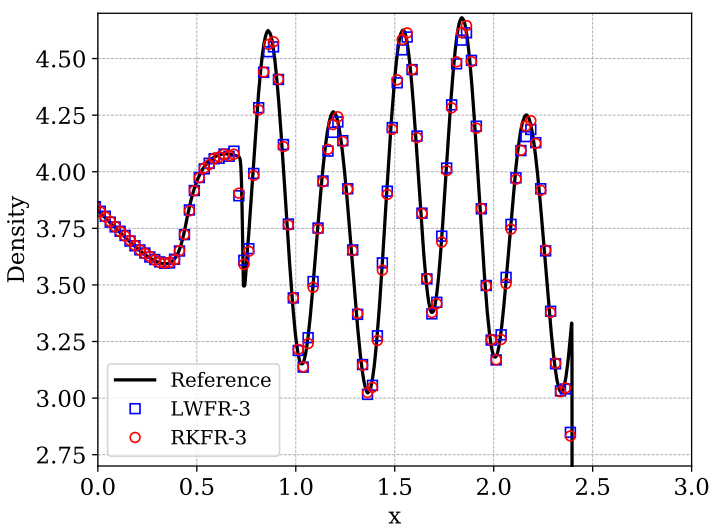


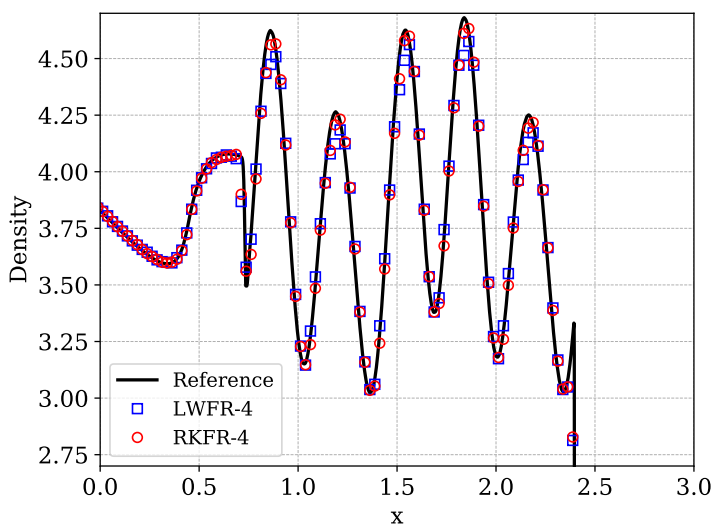
1-D Euler: Shu-Osher test

GL + Radau + D2 + EA, 400 cells, $t = 1.8$



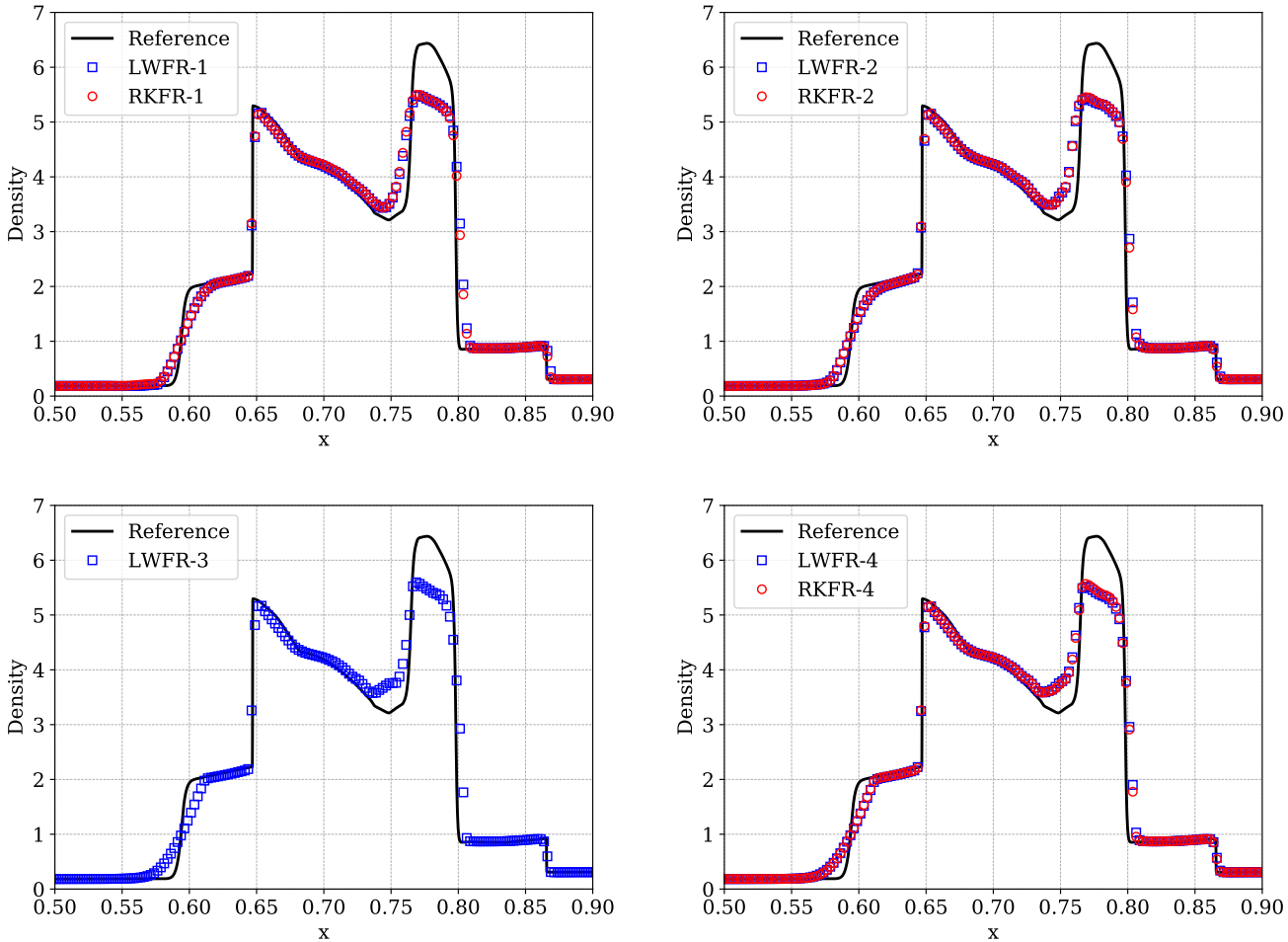






1-D Euler: Blast test

GL + Radau + D2 + EA, 400 cells, $t = 0.038$



AE (not shown) fails for odd degrees

2-D problems

Conservation law

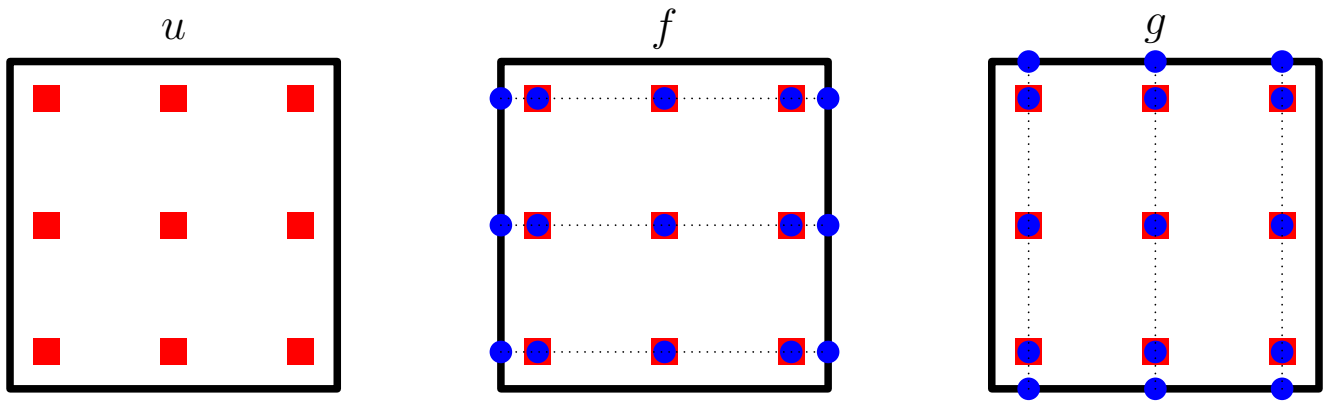
$$u_t + f(u)_x + g(u)_y = 0$$

Map each element $(\Delta x_e, \Delta y_e)$ to reference element

$$\Omega_e \to \hat{K} = [0,1] \times [0,1], \qquad (x,y) \to (\xi, \eta)$$

Solution u_h: piecewise tensor product polynomial

$$(x,y) \in \Omega_e : \qquad u_h = \sum_{i=0}^{N} \sum_{j=0}^{N} u_{ij}^e \ell_i(\xi) \ell_j(\eta)$$



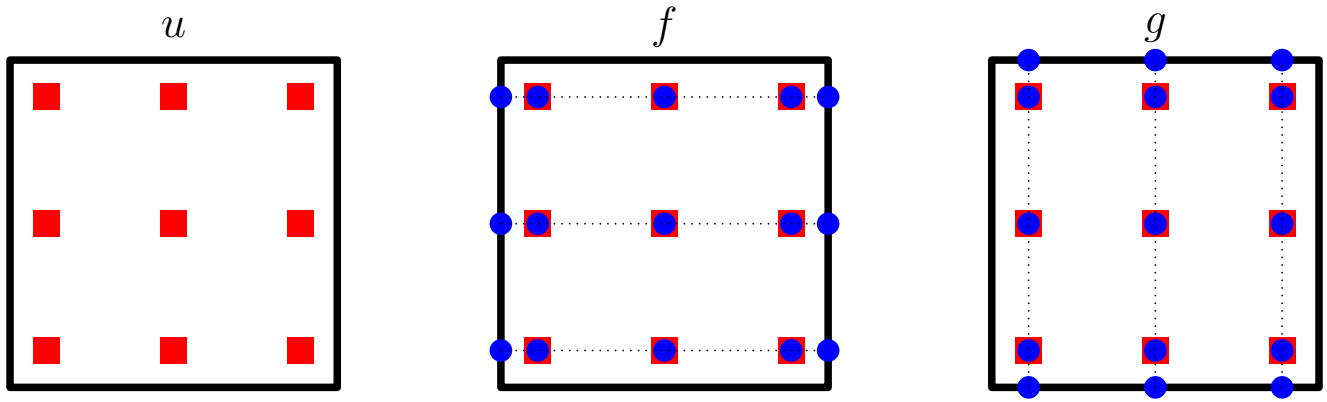
2-D scheme

Step 1. Compute time average flux using ALW at all solution points

$$F^e_{ij}, \qquad G^e_{ij}, \qquad 0 \le i, j \le N$$

Step 2. Build discontinuous flux approximation

$$F^\delta_h(\xi,\eta) = \sum_{i=0}^{N}\sum_{j=0}^{N} F^e_{ij}\ell_i(\xi)\ell_j(\eta), \qquad G^\delta_h(\xi,\eta) = \sum_{i=0}^{N}\sum_{j=0}^{N} G^e_{ij}\ell_i(\xi)\ell_j(\eta)$$



Step 3. Apply FR along $\eta = \xi_j$ and $\xi = \xi_i$ lines

$$F_h(\xi,\xi_j) = [F_{e-\frac{1}{2},j} - F^\delta_h(0,\xi_j)]g_L(\xi) + F^\delta_h(\xi,\xi_j) + [F_{e+\frac{1}{2},j} - F^\delta_h(1,\xi_j)]g_R(\xi), \quad 0 \le j \le N$$

$$G_h(\xi_i,\eta) = [G_{e-\frac{1}{2},i} - G^\delta_h(\xi_i,0)]g_L(\eta) + G^\delta_h(\xi_i,\eta) + [G_{e+\frac{1}{2},i} - G^\delta_h(\xi_i,1)]g_R(\eta), \quad 0 \le i \le N$$

2-D scheme

Step 4. Update solution: For $0 \le i, j \le N$

$$(u_{ij}^e)^{n+1} = (u_{ij}^e)^n - \Delta t \left[\frac{1}{\Delta x_e} \frac{\partial F_h}{\partial \xi}(\xi_i, \xi_j) + \frac{1}{\Delta y_e} \frac{\partial G_h}{\partial \eta}(\xi_i, \xi_j) \right]$$

Matrix form. Define flux matrices

$$\mathsf{F}_e(i,j) = F_{ij}^e, \qquad \mathsf{G}_e(i,j) = G_{ij}^e$$

compute derivatives of discontinuous flux by a matrix-matrix product

$$\partial_\xi F_h^\delta(:,:) = \mathsf{D}\mathsf{F}_e, \qquad \partial_\eta G_h^\delta(:,:) = \mathsf{G}_e \mathsf{D}^\top$$

Update in matrix form

$$u_e^{n+1} = u_e^n - \left[\frac{\Delta t}{\Delta x_e} \mathsf{D}_1 \mathsf{F}_e + \frac{\Delta t}{\Delta y_e} \mathsf{G}_e \mathsf{D}_1^\top \right] - \frac{\Delta t}{\Delta x_e} \left[\mathsf{b}_L \mathsf{F}_{e-\frac{1}{2}}^\top + \mathsf{b}_R \mathsf{F}_{e+\frac{1}{2}}^\top \right] - \frac{\Delta t}{\Delta y_e} \left[\mathsf{G}_{e-\frac{1}{2}} \mathsf{b}_L^\top + \mathsf{G}_{e+\frac{1}{2}} \mathsf{b}_R^\top \right]$$

Fourier stability analysis in 2-D

$$u_t + a_1 u_x + a_2 u_y = 0$$

CFL numbers

$$\sigma_1 = \frac{a_1 \Delta t}{\Delta x_e}, \qquad \sigma_2 = \frac{a_2 \Delta t}{\Delta y_e}$$

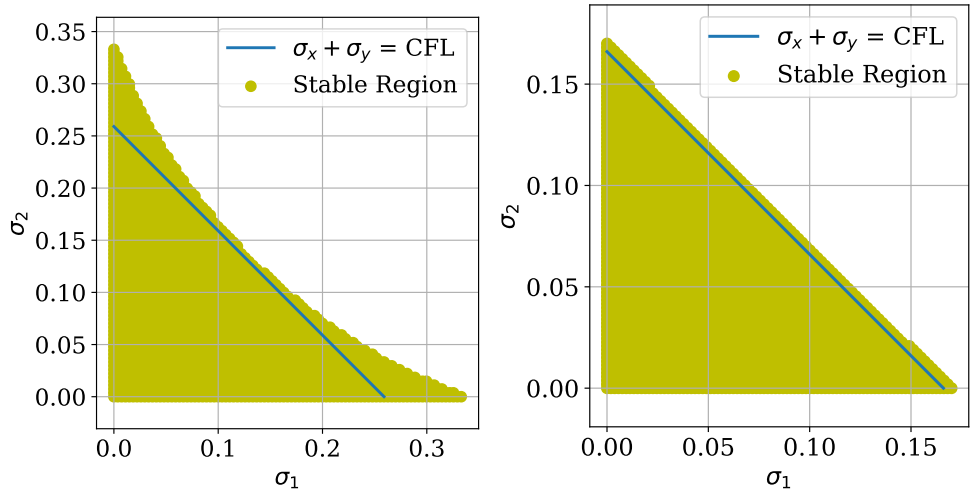
Stable region

$$|\sigma_1| + |\sigma_2| \leq \mathrm{CFL}(N)$$

N	1	2	3	4
Radau	0.259	0.166	0.101	0.067
g2	0.511	0.348	0.178	0.108

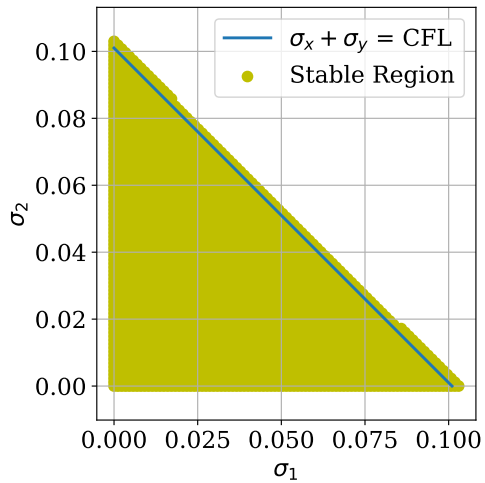
Using D2 dissipation (upwind flux)

Fourier stability analysis in 2-D: LWFR with Radau

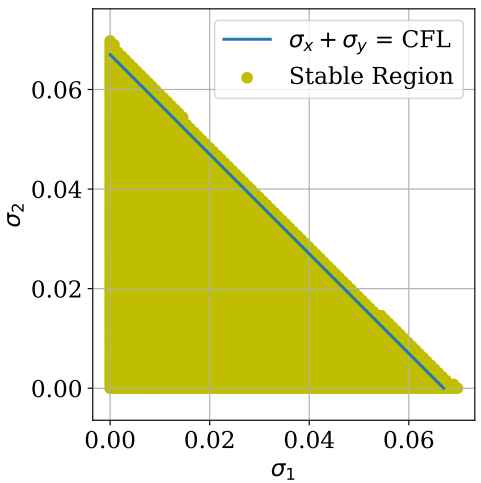


(a) Degree 1

(b) Degree 2



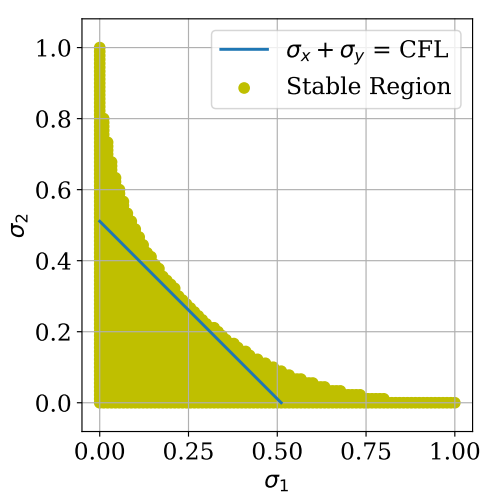
(c) Degree 3



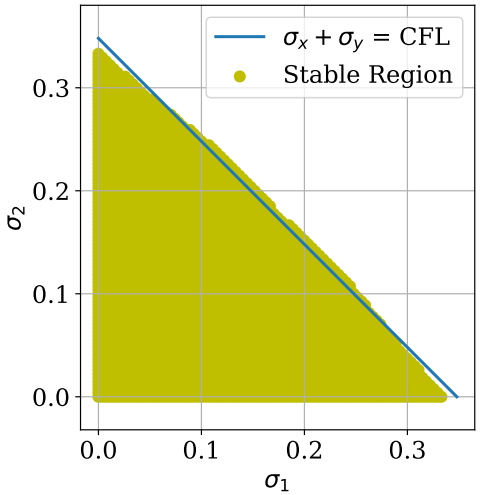
(d) Degree 4

Allowed Δt is reduced for waves moving at an angle to the mesh

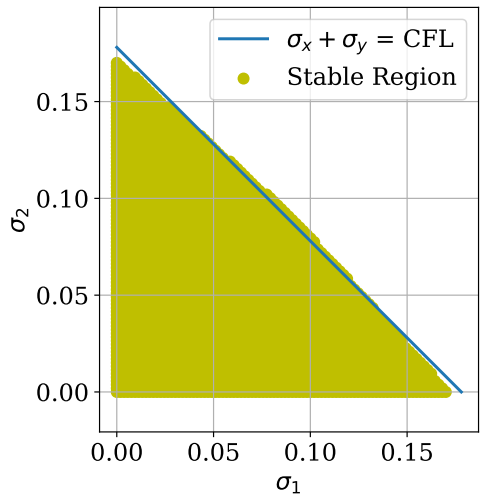
Fourier stability analysis in 2-D: LWFR with g2



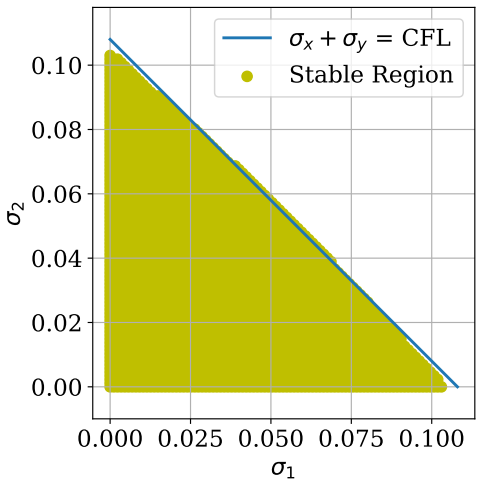
(a) Degree 1



(b) Degree 2



(c) Degree 3

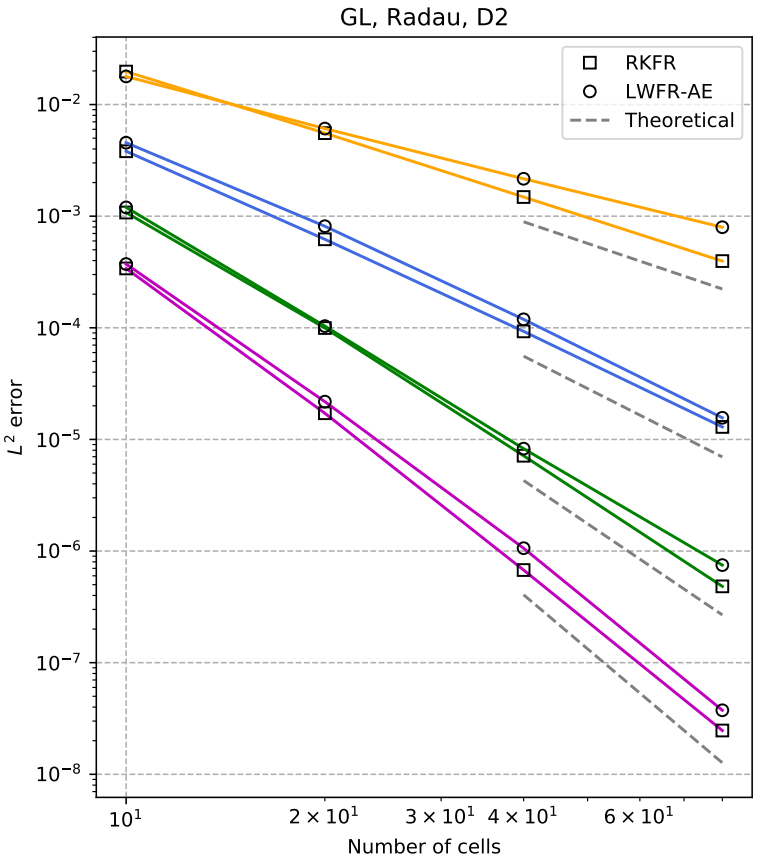


(d) Degree 4

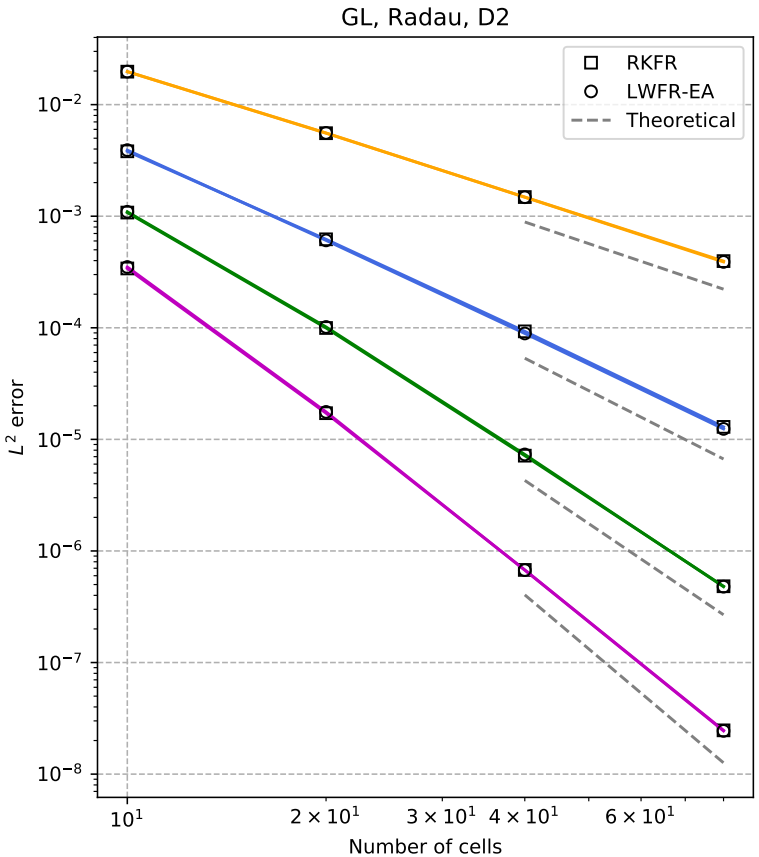
Allowed Δt is reduced for waves moving at an angle to the mesh

Numerical results: 2-D

2-D Burger's equation: convergence test



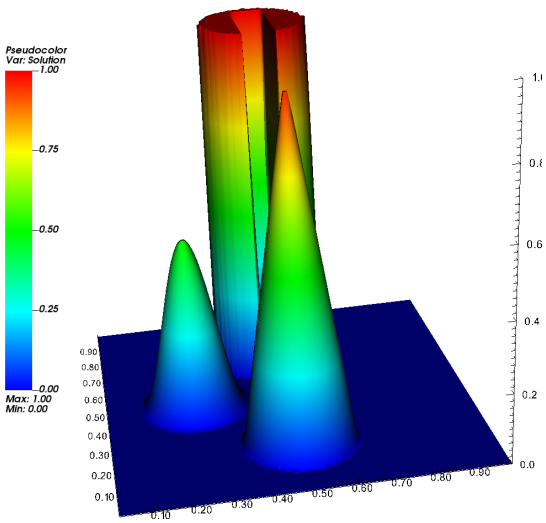
LWFR: $O(h^{N+\frac{1}{2}})$ for N odd



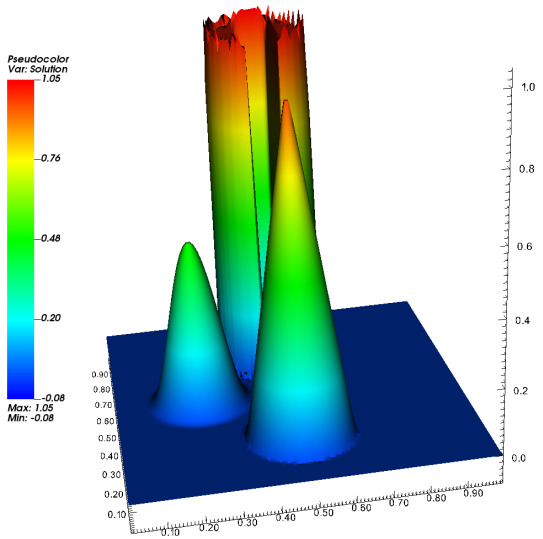
$O(h^{N+1})$ for all N

2-D linear advection

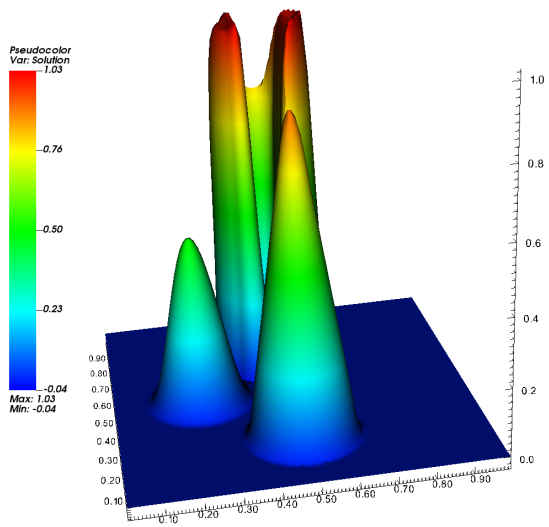
LWFR-D2, $N = 3$, 100×100 cells



Exact



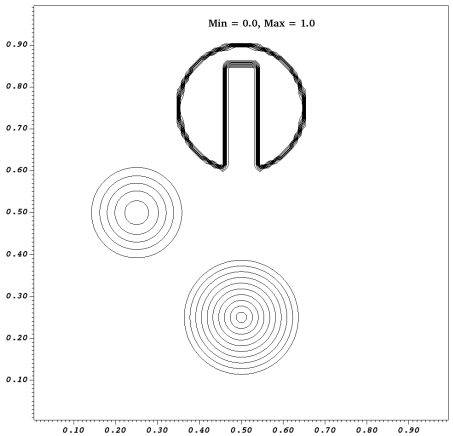
no limiter



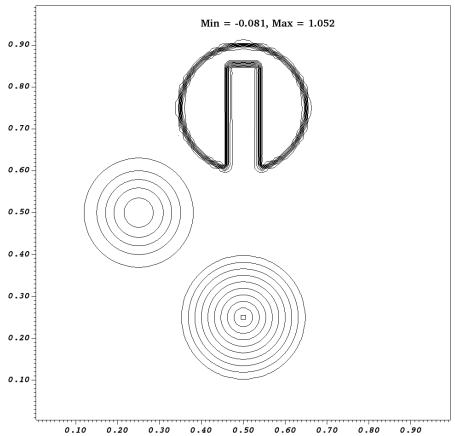
TVB limiter

2-D linear advection: $N = 3$, 100×100 cells

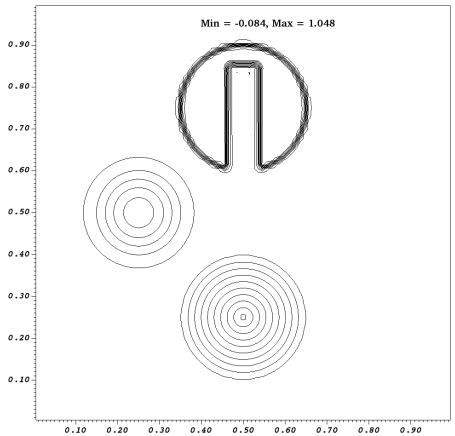
No limiter



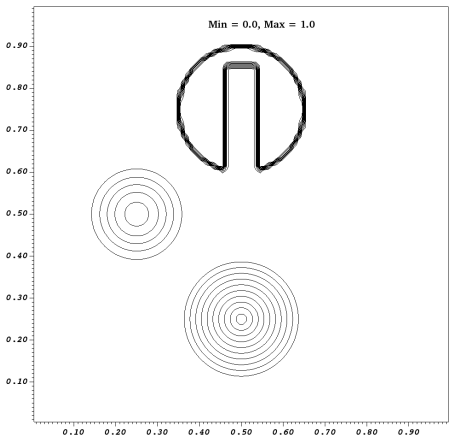
Exact

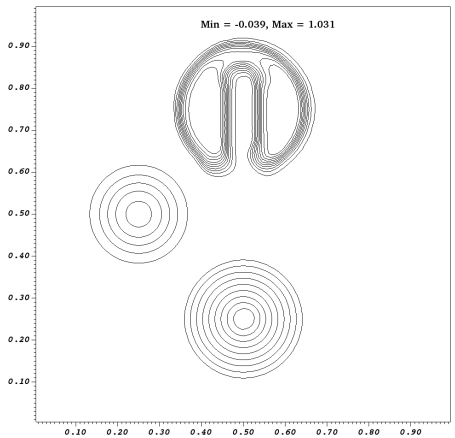


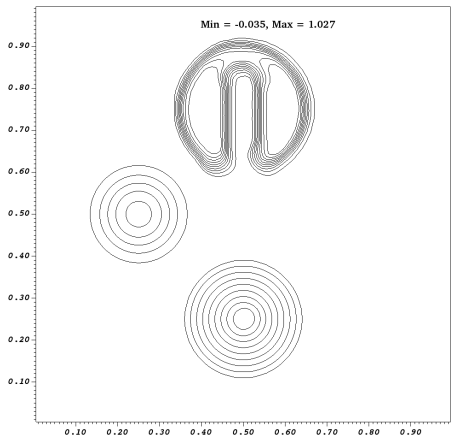
RKFR



LWFR







TVD limiter

Summary and conclusions

- Combine LW with FR idea: single step method for all orders
- Only requires flux evaluations
- Nodal DG/FR type approach: quadrature free, matrix operations
- Classical FR corrections, Radau and g2, lead to Fourier stable schemes
- Numerical fluxes from traditional FV can be used
  - Use time average solution in addition to time average flux
  - Enhanced Fourier CFL numbers, same as ADER-DG
- Numerical flux: Average-Extrapolate (AE)
  Loss of accuracy/rate for non-linear at odd degrees
  GL + Radau + D2 + AE good at even degrees
- Numerical flux: Extrapolate-Average (EA), GL points
  Optimal rates observed at all degrees
  GL + Radau + D2 + EA good at all degrees
- g2 and/or GLL points do not lead to uniformly optimal performance
- LWFR CFL numbers are smaller than DG; but same as ADER-DG
- Constant linear advection: LWFR less accurate than RKFR

Ongoing and future work

- Better limiters
- 2-D/3-D systems
- Unstructured and curved grids
- Convection-diffusion problems
- Parallel implementation

Thank You

References

[1] H. T. Huynh, A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods, in: 18th AIAA Computational Fluid Dynamics Conference, AIAA, Miami, FL, 2007.

[2] P. E. Vincent, P. Castonguay, A. Jameson, A New Class of High-Order Energy Stable Flux Reconstruction Schemes, Journal of Scientific Computing 47 (1) (2011) 50–72.
doi:10.1007/s10915-010-9420-z.

[3] J. Donea, A Taylor–Galerkin method for convective transport problems, International Journal for Numerical Methods in Engineering 20 (1) (1984) 101–119.
doi:10.1002/nme.1620200108.

References

[4] J. Qiu, C.-W. Shu, Finite Difference WENO Schemes with Lax–Wendroff-Type Time Discretizations, SIAM Journal on Scientific Computing 24 (6) (2003) 2185–2198.
doi:10.1137/S1064827502412504.

[5] J. Qiu, M. Dumbser, C.-W. Shu, The discontinuous Galerkin method with Lax–Wendroff type time discretizations, Computer Methods in Applied Mechanics and Engineering 194 (42-44) (2005) 4528–4543.
doi:10.1016/j.cma.2004.11.007.

[6] R. Bürger, S. K. Kenettinkara, D. Zorío, Approximate Lax–Wendroff discontinuous Galerkin methods for hyperbolic conservation laws, Computers & Mathematics with Applications 74 (6) (2017) 1288–1310.
doi:10.1016/j.camwa.2017.06.019.

References

[7] V. A. Titarev, E. F. Toro, ADER: Arbitrary High Order Godunov Approach, Journal of Scientific Computing 17 (1/4) (2002) 609–618. doi:10.1023/A:1015126814947.

[8] F. Lörcher, G. Gassner, C.-D. Munz, A Discontinuous Galerkin Scheme Based on a Space–Time Expansion. I. Inviscid Compressible Flow in One Space Dimension, Journal of Scientific Computing 32 (2) (2007) 175–199. doi:10.1007/s10915-007-9128-x.

[9] M. Dumbser, D. S. Balsara, E. F. Toro, C.-D. Munz, A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes, Journal of Computational Physics 227 (18) (2008) 8209–8253. doi:10.1016/j.jcp.2008.05.025.

References

[10] R. P. K. Chan, A. Y. J. Tsai, On explicit two-derivative Runge-Kutta methods, Numerical Algorithms 53 (2-3) (2010) 171–194. doi:10.1007/s11075-009-9349-1.

[11] J. Li, Z. Du, A Two-Stage Fourth Order Time-Accurate Discretization for Lax–Wendroff Type Flow Solvers I. Hyperbolic Conservation Laws, SIAM Journal on Scientific Computing 38 (5) (2016) A3046–A3069. doi:10.1137/15M1052512.

[12] S. Lou, C. Yan, L.-B. Ma, Z.-H. Jiang, The Flux Reconstruction Method with Lax–Wendroff Type Temporal Discretization for Hyperbolic Conservation Laws, Journal of Scientific Computing 82 (2) (2020) 42. doi:10.1007/s10915-020-01146-8.

References

[13] D. Zorío, A. Baeza, P. Mulet, An Approximate Lax–Wendroff-Type Procedure for High Order Accurate Schemes for Hyperbolic Conservation Laws, Journal of Scientific Computing 71 (1) (2017) 246–273.
doi:10.1007/s10915-016-0298-2.

[14] J. Qiu, A Numerical Comparison of the Lax–Wendroff Discontinuous Galerkin Method Based on Different Numerical Fluxes, Journal of Scientific Computing 30 (3) (2007) 345–367.
doi:10.1007/s10915-006-9109-5.