1 Linear Advection in 1D

$$u_t - u_x = 0, \text{ on } \Omega = \{x \in [0, 1]\},$$

with periodic boundary conditions, and exact solution

$$u(x, t) = \sin(2\pi(x + t)).$$

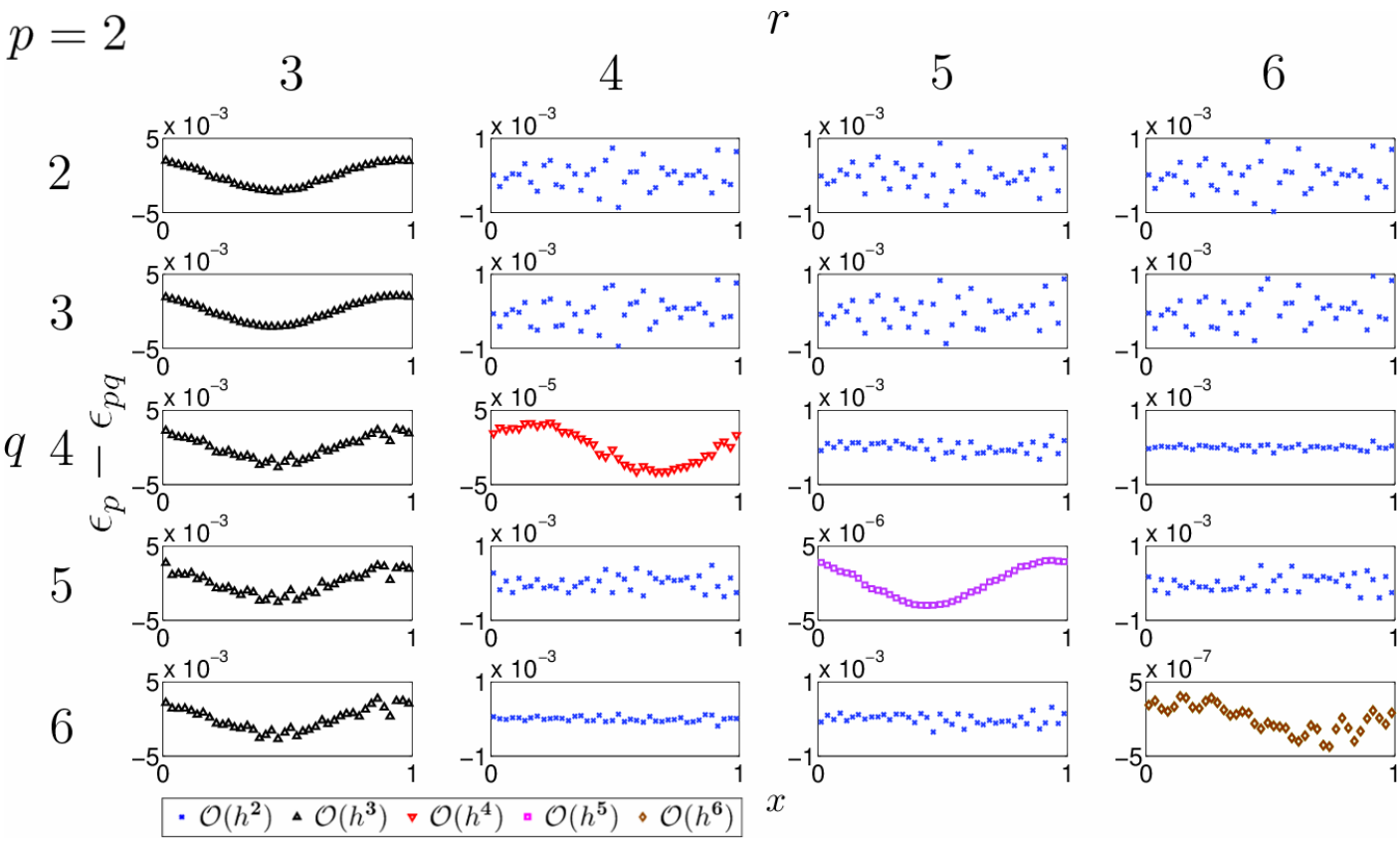

Figure 1: Error estimate difference for $(2, q, r)$ schemes. Plots are colored by different asymptotic behaviors.

2 Box and whiskers plots for Advection:

(included original representation, Figure 2, and revised grouped by p, Figure 3.)

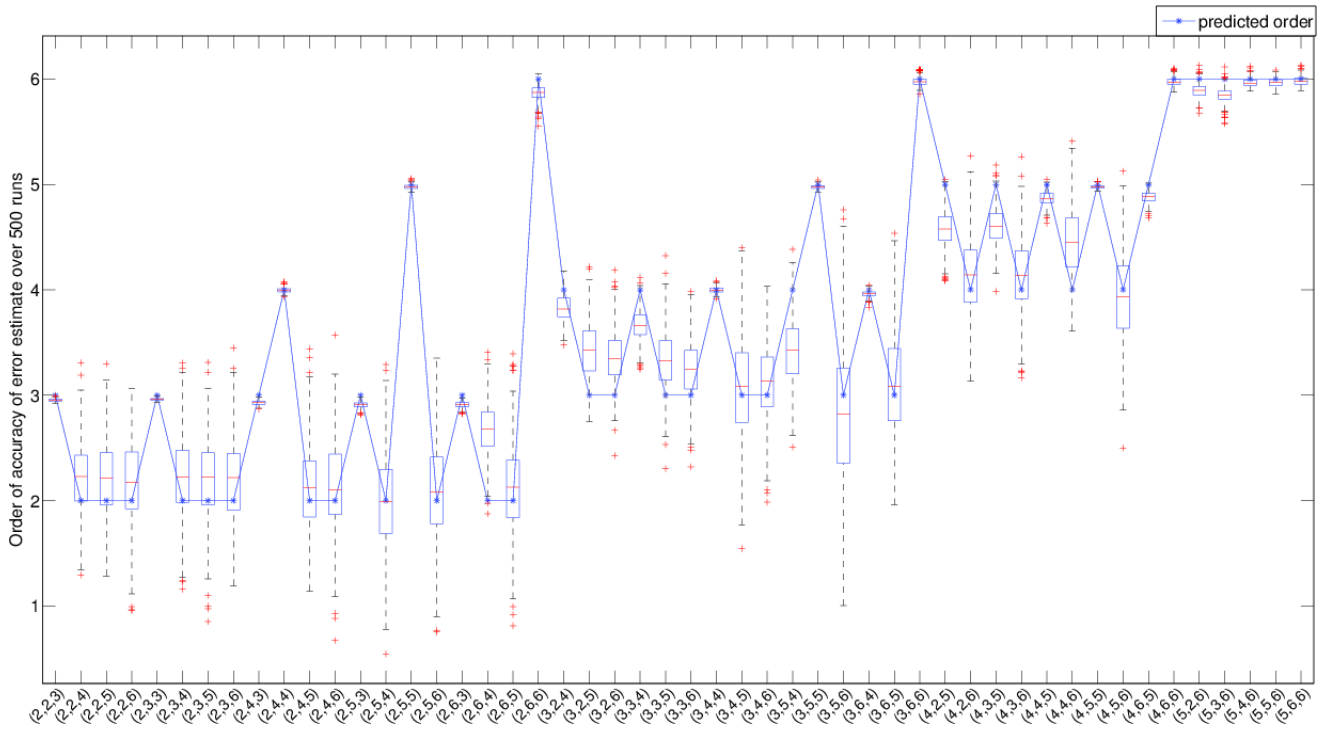

Figure 2: Box and whiskers plot of predicted model versus results. The 25^{th} and 75^{th} percentiles, outliers, and range of the data are shown.

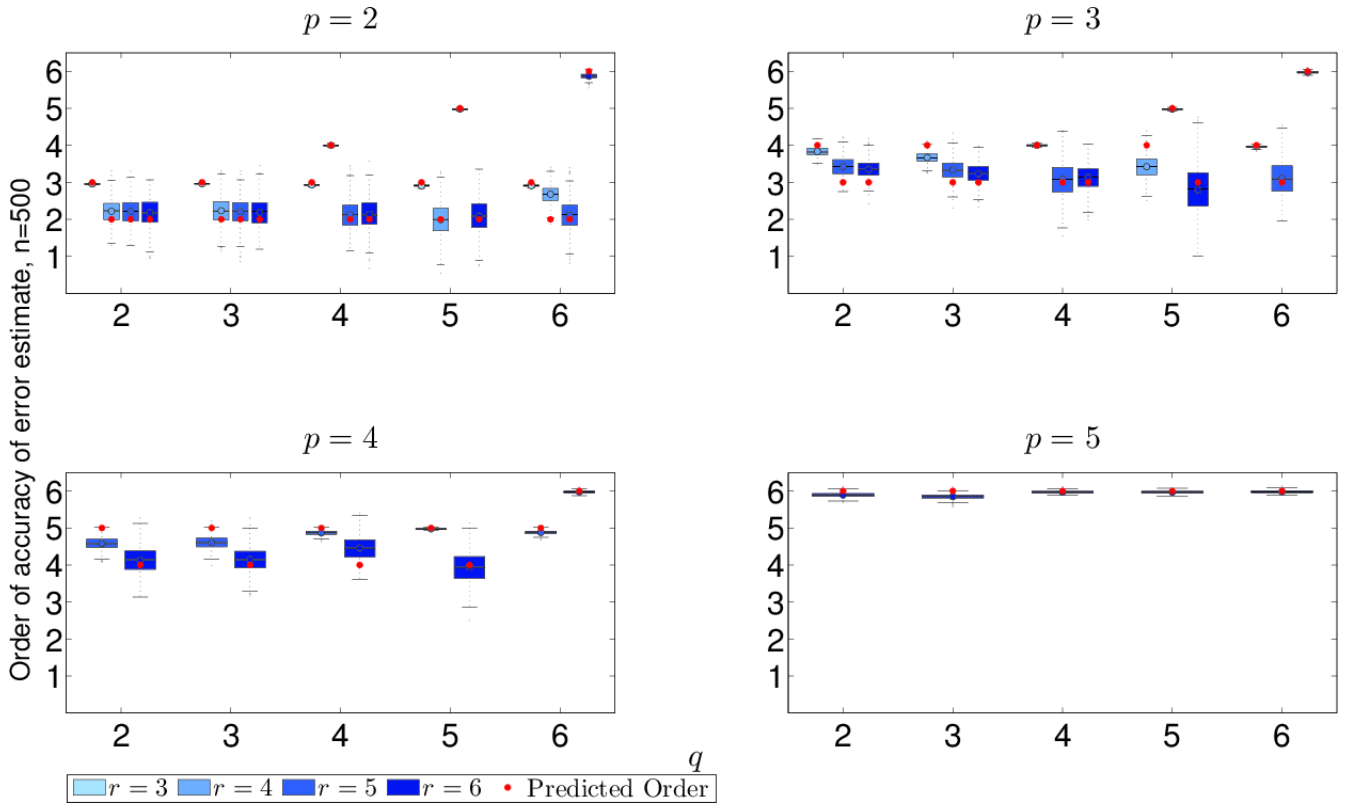

Figure 3: Box and whiskers plot of predicted model versus results. The 25^{th} and 75^{th} percentiles, outliers, and range of the data are shown.

-

3 Quasi-1D Euler Equations

$$\begin{pmatrix} \rho A \\ \rho u A \\ \rho E A \end{pmatrix}_t + \begin{pmatrix} \rho u A \\ (\rho u^2 + P)A \\ \rho u (E + \frac{P}{\rho}) A \end{pmatrix}_x = \begin{pmatrix} 0 \\ P A_x \\ 0 \end{pmatrix}$$

$$\partial_t \mathbf{u} + \partial_x \mathbf{f}(\mathbf{u}) = \mathbf{s}(\mathbf{u})$$

$$\partial_t \varepsilon + \partial_x (\mathbf{f}(\varepsilon + \tilde{\mathbf{u}}) - \mathbf{f}(\tilde{\mathbf{u}})) + \mathbf{s}(\varepsilon + \tilde{\mathbf{u}}) - \mathbf{s}(\tilde{\mathbf{u}}) = -\mathcal{R}(\tilde{\mathbf{u}})$$

$$\begin{aligned} \partial_t \varepsilon + \partial_x (f(\varepsilon + \tilde{u}) - f(\tilde{u})) &= s(\varepsilon + \tilde{u}) - s(\tilde{u}) - \mathcal{R}(\tilde{u}) \\ &= s(\varepsilon + \tilde{u}) - s(\tilde{u}) - (\partial_t \tilde{u} + \partial_x f(\tilde{u}) - s(\tilde{u})) \end{aligned}$$

The exact solution is purely subsonic with a throat.

The four methods attempted can be summarized as follows. Method 4 is solving primal and error equation purely on conserved variables, hence no translations whatsoever.

Methods 1, 2, and 3 all involve solving the primal problem via reconstruction in the primitive variables. The difference is in how the error equation is handled. In method 1, the errors in primitive and conserved variables are translated back and forth by adding and subtracting the converged quantities. Method 2 handles the error equation in essentially the same way as method 4: by only considering error in the conserved variables and computing flux as $f(\epsilon_U + U_p) - f(U_p)$. Method 3 takes the error in the conserved variables as a passive scalar and adds it to the primitive converged solution for reconstruction and computing fluxes

Method 2 and method 4 gave similar and better results than the other two. Boundary conditions are taken to be zero and weakly enforced.

3.1 Method 1

Algorithm 1 Conversion between primitive and conserved ϵ_U,ϵ_V

while $||R|| > \delta$
  $\frac{\partial \mathbf{R}}{\partial \epsilon}$=ComputeJacobian($\epsilon_V$)
  R = ComputeErrorFluxIntegral($\mathcal{I}_h^q \epsilon_V$)
  ϵ_U= TranslateToConserved($V_p + \epsilon_V$) $- U_p$
  $\Delta \epsilon_U = \left(\frac{\mathbf{1}}{\Delta t} + \frac{\partial \mathbf{R}}{\partial \epsilon} \right)^{-1} R$
  $\epsilon'_U = \epsilon_U + \Delta \epsilon_U$
  ϵ'_V=TranslateToPrimitive($U_p + \epsilon'_U$) $- U_p$
  $\epsilon_V = \epsilon'_V$
end

3.2 Method 2

Algorithm 2 Use only ϵ_U and U_p

while $||R|| > \delta$
$\frac{\partial \mathbf{R}}{\partial \epsilon}$=ComputeJacobian2($\epsilon_U$)
R = ComputeErrorFluxIntegral2(ϵ_U)=$\int f(\epsilon_U + U_p) - f(U_p) d\mathbf{x}$
$\Delta\epsilon_U = \left(\frac{\mathbf{1}}{\Delta t} + \frac{\partial \mathbf{R}}{\partial \epsilon}\right)^{-1} R$
$\epsilon'_U = \epsilon_U + \Delta\epsilon_U$
$\epsilon_U = \epsilon'_U$
end

3.3 Method 3

Algorithm 3 Treatϵ_U passive scalar, error flux as $f(\epsilon_U + V_p) - f(V_p)$

while $||R|| > \delta$
$\frac{\partial \mathbf{R}}{\partial \epsilon}$=ComputeJacobian2($\epsilon_U$)
R = ComputeErrorFluxIntegral3(ϵ_U)
$\Delta\epsilon_U = \left(\frac{\mathbf{1}}{\Delta t} + \frac{\partial \mathbf{R}}{\partial \epsilon}\right)^{-1} R$
$\epsilon'_U = \epsilon_U + \Delta\epsilon_U$
$\epsilon_U = \epsilon'_U$
end

3.4 Method 4

Algorithm 4 Everything done in conserved variables

SolvePrimalReconConserved
while $||R|| > \delta$
$\frac{\partial \mathbf{R}}{\partial \epsilon}$=ComputeJacobian2($\epsilon_U$)
R = ComputeErrorFluxIntegral4(ϵ_U)
$\Delta\epsilon_U = \left(\frac{\mathbf{1}}{\Delta t} + \frac{\partial \mathbf{R}}{\partial \epsilon}\right)^{-1} R$
$\epsilon'_U = \epsilon_U + \Delta\epsilon_U$
$\epsilon_U = \epsilon'_U$
end

Random Mesh results to compare methods:

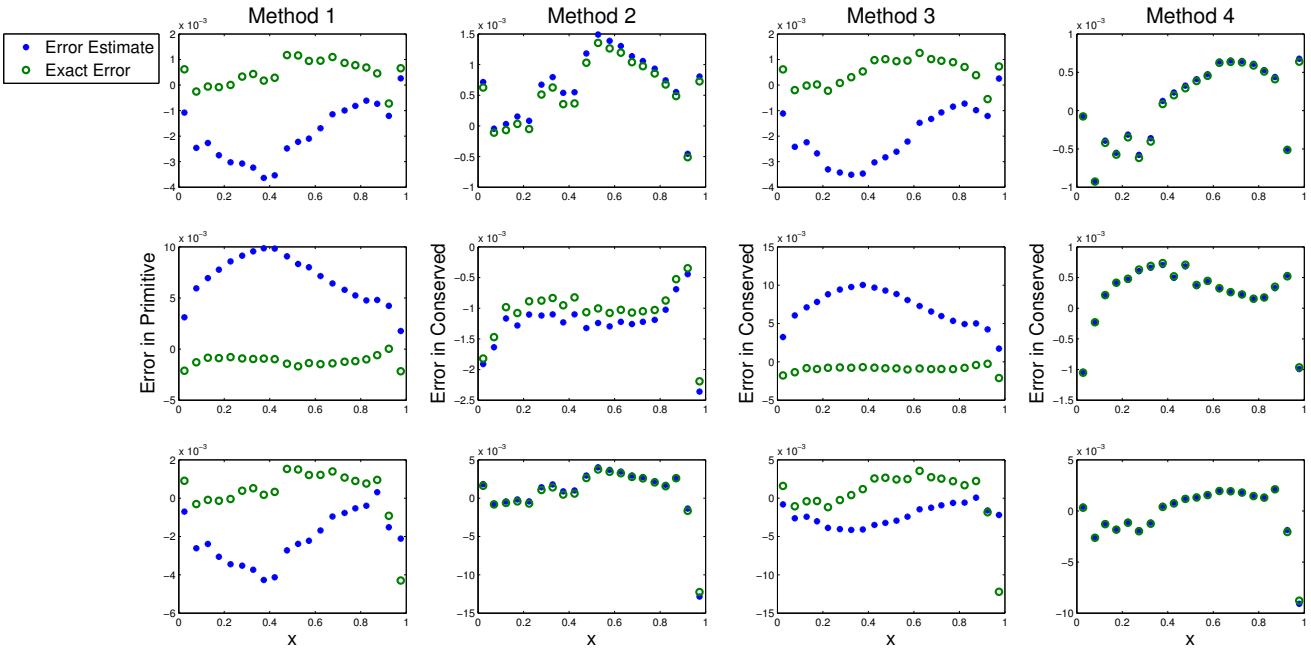

Figure 4: Comparison of four methods in $(2, 4, 4)$ scheme.

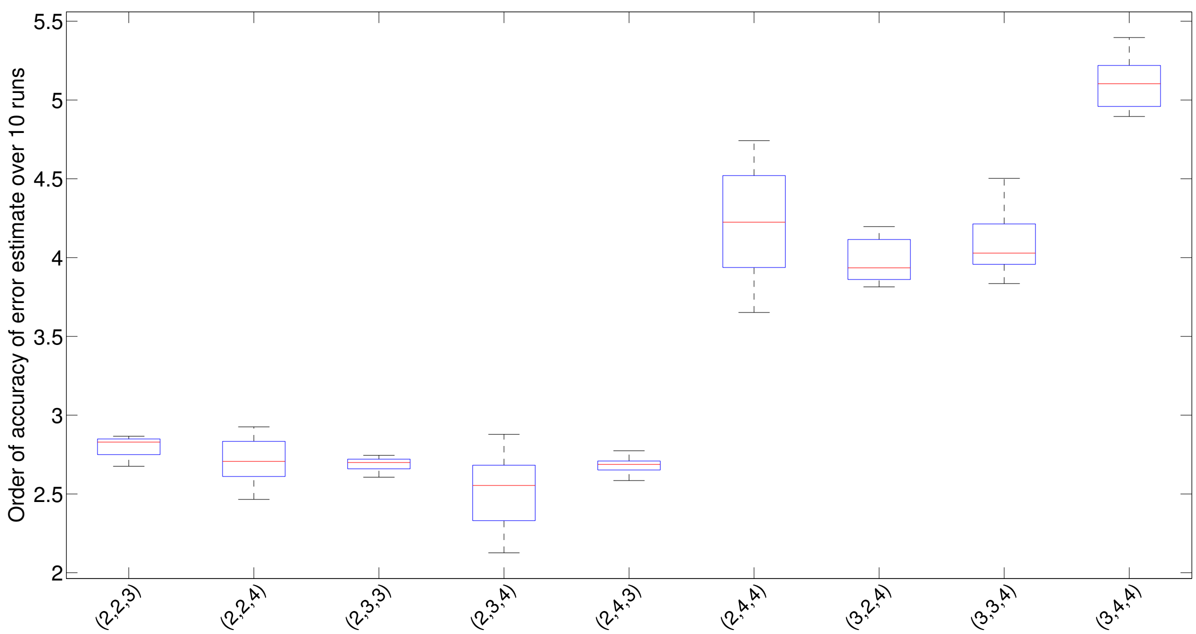

Figure 5: Box and whiskers plot of method 4 for quasi-1D Euler.

-

4 Linear Advection in 2D

Consider the steady problem

$$
\begin{aligned}
\mathbf{a} \cdot \nabla u &= 0, \text{ on } \Omega = \{(x, y) \in [0, 3] \times [0, 1]\}, \mathbf{a} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T. \\
u(x, 0) &= 0 \\
u(x, 1) &= 0 \\
u(0, y) &= 4 \log(1 + y(1 - y))
\end{aligned}
$$

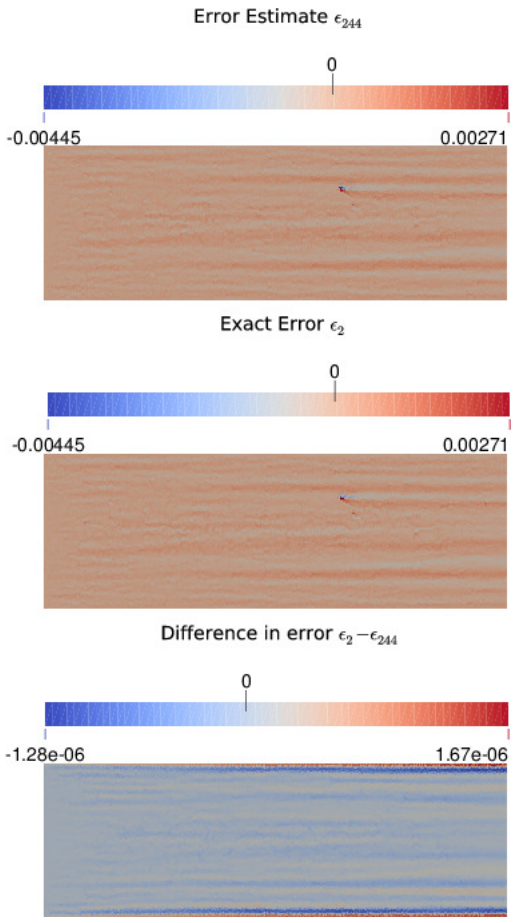

Figure 6: Error estimate, exact error, and difference for $(2, 4, 4)$ scheme for Advection. Convergence of error is close to $\mathcal{O}(h^q)$.

Remedied this by generating meshes with uniform degree (6) so this configuration is heuristically not present.

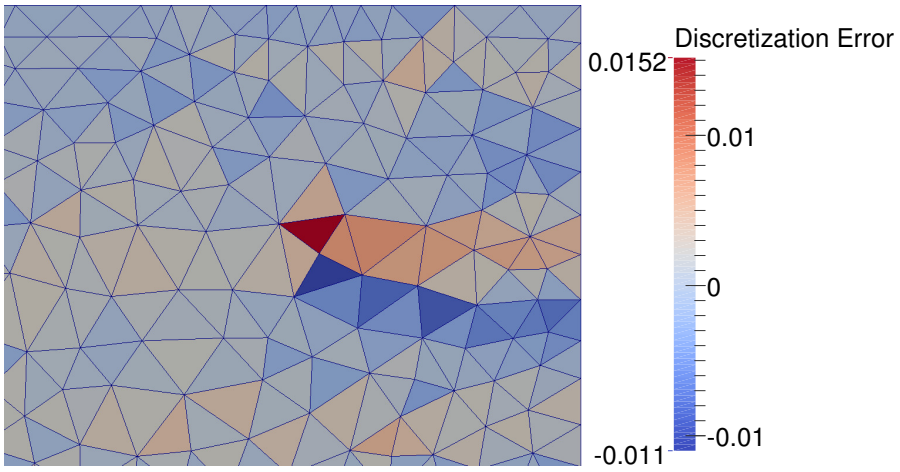

Figure 7:

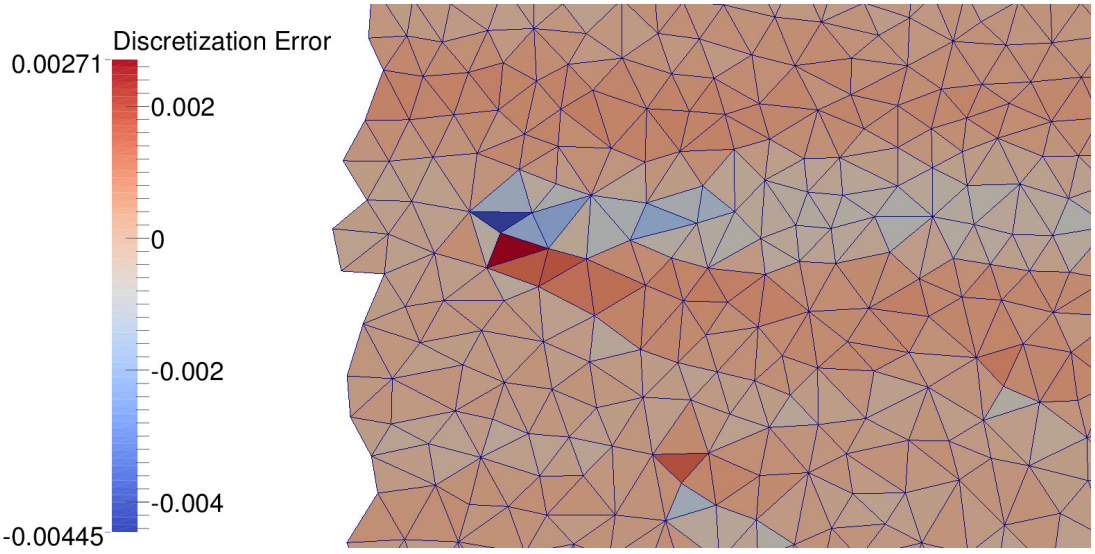

Figure 8: Close up of bad arrangement. Heuristically the configuration is a subset of a large angle incident to the wave direction.

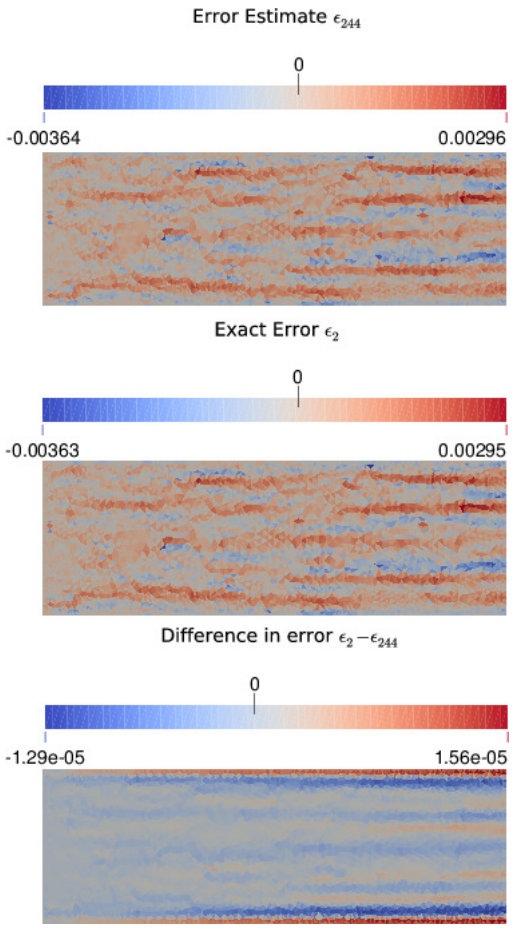

Figure 9: "Good Configuration": Error estimate, exact error, and difference for $(2, 4, 4)$ scheme for Advection. Convergence of error is close to $\mathcal{O}(h^q)$.

-

5 Burgers' Equation in 2D

Consider the steady problem

$$\begin{aligned} uu_x + u_y &= 0, \text{ on } \Omega = \{(x,y) \in [0,\pi] \times [0,0.5]\} \\ u(x,0) &= -\sin x \\ u(0,y) &= 0 \\ u(\pi,y) &= 0 \end{aligned}$$

The boundary conditions are compatible with the characteristics along the boundary $\frac{dx}{dy}(0,y) = \frac{dx}{dy}(\pi,y) = 0$. The exact solution is:

$$u(x,y) = -2\sum_{n=1}^{\infty} \frac{J_n(ny)}{ny} \sin nx,$$

where J_n is the Bessel function of the first kind of order n.

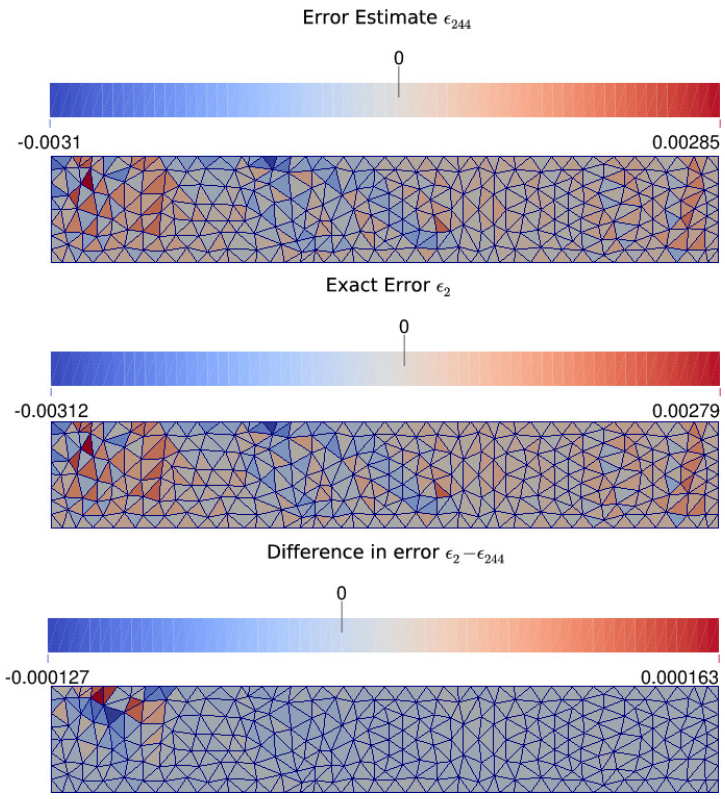

Figure 10: Error estimate, exact error, and difference for $(2,4,4)$ scheme for Burgers' Equation. Convergence of error is close to $\mathcal{O}(h^q)$.