Asymptotic Accuracy for Boundary Value Problems

Julian Köllermeier, Manuel Torrilhon
RWTH Aachen University, September 2011

1 Hyperbolic Gas Poiseuille-Flow Model

The model equation emerges from a 1D simplification of the R13 system and has the following form

$$\partial_t U + \partial_x F(U) = P(U) + G \tag{1}$$

Here, the variables are $U = (u, v, w)^T$ with velocity u, shear stress v and parallel heat flux w. Furthermore, we have

$$F(U) = AU \text{ with } A = \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix}, P(U) = \begin{pmatrix} 0 \\ -v/\epsilon \\ -w/\epsilon \end{pmatrix}, G = \begin{pmatrix} g \\ 0 \\ 0 \end{pmatrix} \tag{2}$$

Here the parameters g and ϵ are the external force and relaxation time.

For later use where write down the stationary PDE as follows

$$\partial_x v = g \tag{3}$$

$$\frac{1}{2}\partial_x u + \frac{1}{2}\partial_x w = -v/\epsilon \tag{4}$$

$$\partial_x v = -w/\epsilon \tag{5}$$

For real applications, we consider a bounded domain $x \in [-1, 1]$ where we have to prescribe boundary conditions. We only have a boundary condition for v as follows

$$v|_{x=\pm 1} = \pm(\alpha u + \beta w)_{x=\pm 1} \tag{6}$$

with some parameters $\alpha > \beta > 0$.

For our numerical experiments we chose example values

$$g = 1, \alpha = 0.7, \beta = 0.3, \epsilon = 0.01 \tag{7}$$

With the help of numerical solution methods we want to converge to the steady state, which is here

$$u_s(x) = g \cdot \left(\frac{1 + \epsilon\beta}{\alpha} + \frac{1}{\epsilon}(1 - x^2) \right), \quad v_s(x) = gx, \quad w_s(x) = -\epsilon g \tag{8}$$

In order to compare numerical solutions afterwards, we first compute the norms of the exact steady state solution as follows

$$||\cdot||_1 = \int_{-1}^{1} |\cdot| dx \quad , \quad ||\cdot||_\infty = max|\cdot| \tag{9}$$

Using this definition, we obtain the values for all the three components

$$||u_s||_1 = \frac{2g(1 + \epsilon\beta)}{\alpha} + \frac{4g}{3\epsilon} = \mathcal{O}(\epsilon^{-1}) \quad , ||v_s||_1 = g = \mathcal{O}(1) \quad , ||w_s||_1 = 2\epsilon g = \mathcal{O}(\epsilon) \tag{10}$$

$$||u_s||_\infty = \frac{g(1 + \epsilon\beta)}{\alpha} + \frac{g}{\epsilon} = \mathcal{O}(\epsilon^{-1}) \quad , ||v_s||_\infty = g = \mathcal{O}(1) \quad , ||w_s||_\infty = \epsilon g = \mathcal{O}(\epsilon) \tag{11}$$

At this moment, the difficulty of simulating the whole system can be seen due to the different scales of the three components.

2 First Solution Method and Results

We will first repeat the first solution method done by M.Torrilhon. Here the spatial discretization will lead to a semi-discrete formulation that will be solved using the Heun method and a *stifly accurate* IMEX-scheme in order to use a larger time step.

2.1 Spatial Discretization

We use a finite volume discretization with

$$x_i = -1 + (i - \frac{1}{2})\Delta x, \quad \Delta x = \frac{2}{N}, \quad i = 1, \ldots, N \tag{12}$$

and walls/boundaries located at $x_{1/2} = -1$ and $x_{N+1/2} = 1$. So we get for the evolution of the cell means U_i

$$\frac{dU_i}{dt} = R_i(U), \quad \text{with} \quad R_i(U) = \frac{1}{\Delta x}\left(F_{i-1/2} - F_{i+1/2}\right) + P(U_i) + G \tag{13}$$

The numerical flux $F_{1+1/2}$ is evaluated using upwinding with linear reconstruction

$$F_{1+1/2} = \widetilde{F}\left(\frac{U_{i+1} + 4U_i - U_{i-1}}{4}, \frac{-U_{i+2} + 4U_{i+1} + U_i}{4}\right) \tag{14}$$

$$\widetilde{F}(U_L, U_R) = \frac{1}{2}A(U_L + U_R) + \frac{1}{2}|A|(U_L - U_R) \tag{15}$$

Here the matrix $|A|$ has the same eigenvectors as A but the eigenvalues of A in modulus. Therefore $|A|$ can be computed offline before the simulations as

$$|A| = \begin{pmatrix} 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \end{pmatrix} \tag{16}$$

As we use linear reconstruction of the fluxes, we need two layers of ghost cells that can be obtained using the boundary values. This part of the discretization is most important, because the efficiency of the whole method heavily depends on the choice of the right boundary values and extrapolation methods. We will now present the original method, proposed by M.Torrilhon.

Knowing the values in the interior of the domain, we first need to get the values exactly at the boundary. This is done by a second order extrapolation as follows

$$U_{1/2} = \frac{15}{8}U_1 - \frac{5}{4}U_2 + \frac{3}{8}U_3, \quad U_{N+1/2} = \frac{15}{8}U_N - \frac{5}{4}U_{N-1} + \frac{3}{8}U_{N-2} \tag{17}$$

Additionally, boundary conditions have to be imposed on v to obtain the wall values

$$U^W_{\text{left}} = (u_{1/2}, -(\alpha u_{1/2} + \beta w_{1/2}), w_{1/2})^T, \tag{18}$$

$$U^W_{\text{left}} = (u_{N+1/2}, \alpha u_{N+1/2} + \beta w_{N+1/2}, w_{N+1/2})^T \tag{19}$$

Now we can use those values on the wall to obtain the values at the ghost cells by linear extrapolation

$$U_0 = 2U^W_{\text{left}} - U_1, \quad U_{-1} = 2U_0 - U_1 \tag{20}$$

$$U_{N+1} = 2U^W_{\text{left}} - U_N, \quad U_{N+2} = 2U_{N+1} - U_N \tag{21}$$

2.2 Time integration

The semi-discrete formulation is now integrated. As reference we use the fully explicit second order Heun method for the semi-discrete system given by

$$U_i^* = U_i^n + \Delta t R_i(U^n), \quad U_i^{n+1} = \frac{1}{2}\left(U_i^n + U_i^* + \Delta t R_i(U^*)\right) \tag{22}$$

As the stability constraint for the Heun method we have

$$\widetilde{\nu} = \frac{\Delta t \; a_{\max}}{\Delta x} + \frac{\Delta t}{2\epsilon} \leq 1 \tag{23}$$

Where $a_{\max}$ is the maximum eigenvalue of A, here it is $a_{\max} = 1$.

To improve the stability constraint in order to choose larger timesteps, the *stiffly accurate* IMEX-SSP2(3,3,2) is employed based on the explicit and implicit tableau

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & 1/2 & 1/2 & 0 \\ \hline & 1/3 & 1/3 & 1/3 \end{array} \qquad \begin{array}{c|ccc} 1/4 & 1/4 & 0 & 0 \\ 1/4 & 0 & 1/4 & 0 \\ 1 & 1/3 & 1/3 & 1/3 \\ \hline & 1/3 & 1/3 & 1/3 \end{array} \tag{24}$$

and the stability constraint $\nu = \Delta t \; a_{\max}/\Delta x \leq 1$ independent of ϵ. For more information about IMEX schemes or the finite volume discretizazion see [1].

2.3 First Results

Though we expected the IMEX method to give better results, there remains a significant error near the boundary. There the error begins to oscillate. As for the convergence order of the method, the desired second order can only be achieved for the first and second component u and v, whereas the third component seems to have a lower order of convergence. Furthermore one can see an offset for the error of the third component, which is not the case when applying Heun method.

So the question are: What spoils the desired behaviour of the second order IMEX method and is there a possibility to overcome those properties by treating the boundary values differently?

In order to ensure that the problem of loosing the second order convergence is because of the boundary behaviour, we also simulated a periodic test case. As further test settings, we chose $g = 0$. The initial conditions are $u = \sin(\pi x) + 0.5\sin(5\pi x), \quad v = 0, \quad w = 0$. We simulate until $t_{end} = 4$ and compare the results for a spatial discretization of $N = 50$ with $N = 100$. The approximative exact solution is calculated with the Heun method using a high spatial resolution of $N = 400$.

Using these settings, we obtained different orders of convergence for Heun and IMEX. For Heun, we have a order of 2 for all three components. With the IMEX method, we also got a convergence order of 2 for u and v, but for w we only found 1.5. So it seems that we cannot expect more than a convergence order of 1.5 for the third component, but for the second component v a second order convergence is possible.

3 Variants for Boundary Treatment

After the first attempt, several variants have been developed and numerically tested. We now want to present the various methods that handle the setting of boundary values while evaluating the numerical fluxes inside the IMEX or Heun method.

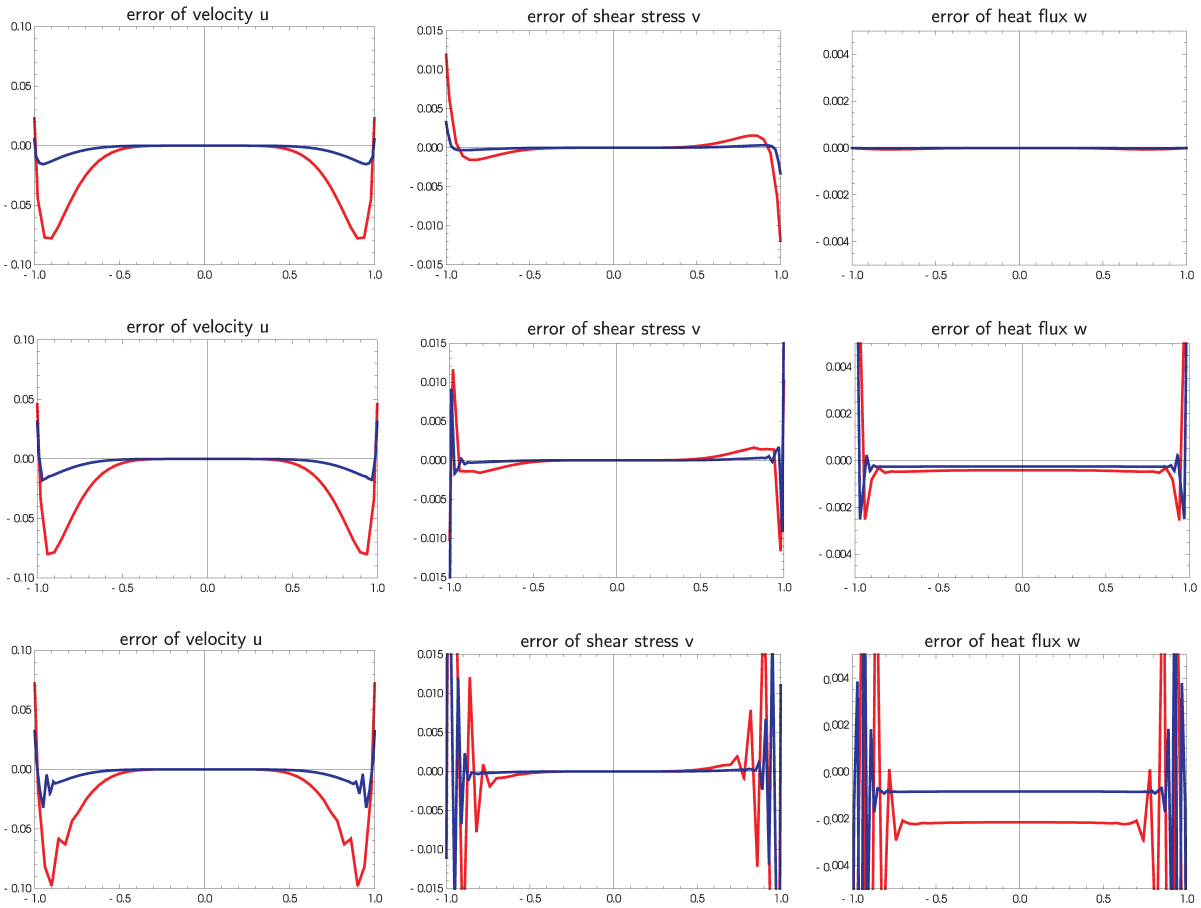

Figure 1: First results obtained with Heun method (top) and IMEX (middle) on grids $N = 50$ (420 time steps, $\nu = 0.3, \widetilde{\nu} = 0.89$) and $N = 100$ (560 time steps, $\nu = 0.45, \widetilde{\nu} = 0.9$) and IMEX with large time steps (bottom) $N = 50$ (140 time steps, $\nu = 0.89, \widetilde{\nu} = 2.68$) and $N = 100$ (280 time steps, $\nu = 0.89, \widetilde{\nu} = 1.79$). The simulation ends at time $t_{end} = 5$

3.1 Reference 1: first approach

As a reference for the other methods, we use the original method. For the spatial discretization near the boundary we used second order extrapolation to the boundary values, then imposed the boundary condition on v and obtained the two ghost cell layers via linear extrapolation from the boundary and inner cells.

3.2 Variant 2: BC for u instead of v

In this method, the boundary value is imposed on u instead of v. This is done by solving the boundary equation (6) for u and inserting the respective value after extrapolating from the interior.

$$u|_{x=\pm 1} = \pm \frac{1}{\alpha}(v - \beta w)_{x=\pm 1} \tag{25}$$

3.3 Variant 3: BC extrapolation 1st order

The extrapolation to the boundary values is done with first order. So we set

$$U_{1/2} = 1.5U_1 - 0.5U_2, \quad U_{N+1/2} = 1.5U_N - 0.5U_{N-1} \tag{26}$$

3.4 Variant 4: GC extrapolation 2nd order

As another attempt to achieve second order, we improve the extrapolation to the ghost cells. Instead of a first order approximation we here use second order.

$$U_0 = \frac{1}{3}(8U^W_{\text{left}} - 6U_1 + U_2) \tag{27}$$

$$U_{-1} = 8U^W_{\text{left}} - 9U_1 + 2U_2 \tag{28}$$

$$U_{N+1} = \frac{1}{3}(8U^W_{\text{right}} - 6U_N + U_{N-1}) \tag{29}$$

$$U_{N+2} = 8U^W_{\text{right}} - 9U_N + 2U_{N-1} \tag{30}$$

3.5 Variant 5: 1st layer GC extrapolated 2nd order, 2nd layer GC constantly extrapolated

In order to study the effect of the ghost cell extrapolation, we here differentiate the extrapolation for the two layers of ghost cells. For the first layer we use the second order extrapolation from variant 4 and the second layer is only constantly extrapolated.

$$U_0 = \frac{1}{3}(8U^W_{\text{left}} - 6U_1 + U_2) \tag{31}$$

$$U_{-1} = U_0 \tag{32}$$

$$U_{N+1} = \frac{1}{3}(8U^W_{\text{right}} - 6U_N + U_{N-1}) \tag{33}$$

$$U_{N+2} = U_{N+1} \tag{34}$$

3.6 Variant 6: BC for v, w from stationary PDE

To accelerate the convergence to the stationary solution, we try setting values from the stationary PDE on the boundary. The extrapolation to the boundary is no longer necessary. For the calculation of the boundary values, we discretize the stationary PDE with forward differences of first order and set u using the boundary condition afterwards.

$$v^W_{\text{left}} = v_1 - g \cdot \frac{\Delta x}{2} \tag{35}$$

$$w^W_{\text{left}} = -g \cdot \epsilon \tag{36}$$

$$u^W_{\text{left}} = -\frac{1}{\alpha}(v^W_{\text{left}} - \beta w^W_{\text{left}}) \tag{37}$$

$$v^W_{\text{right}} = v_N + g \cdot \frac{\Delta x}{2} \tag{38}$$

$$w^W_{\text{right}} = -g \cdot \epsilon \tag{39}$$

$$u^W_{\text{right}} = \frac{1}{\alpha}(v^W_{\text{right}} - \beta w^W_{\text{right}}) \tag{40}$$

Note that the v component has to be calculated first in order to impose the boundary condition for u afterwards.

3.7 Variant 7: BC on u according to method 2, v, w from stationary PDE

For the investigation of values from the stationary PDE, we try imposing the boundary condition on u first and set v, w from stationary PDE afterwards. Therefore we still need the extrapolation to the boundary, in order to impose the boundary condition on u in contrast to variant 6.

So first apply equation (17) and set u according to the boundary condition (25). Now update v, w according to

$$v_{\text{left}}^W = v_1 - g \cdot \frac{\Delta x}{2} \tag{41}$$

$$w_{\text{left}}^W = -g \cdot \epsilon \tag{42}$$

$$v_{\text{right}}^W = v_N + g \cdot \frac{\Delta x}{2} \tag{43}$$

$$w_{\text{right}}^W = -g \cdot \epsilon \tag{44}$$

Like in method 6.

3.8 Variant 8: GC u according to method 2, GC v, w directly from stationary PDE

Here we set the values of v, w on the ghost cells directly from the stationary PDE. The values for u are obtained like in method 2 using linear extrapolation first.

$$U_0 = \begin{pmatrix} 2 \cdot u_{\text{left}}^W - u_1 \\ v_1 - g \cdot \Delta x \\ -g \cdot \epsilon \end{pmatrix}, \quad U_{-1} = \begin{pmatrix} 2 \cdot u_0 - u_1 \\ v_1 - g \cdot 2 \cdot \Delta x \\ -g \cdot \epsilon \end{pmatrix} \tag{45}$$

$$U_{N+1} = \begin{pmatrix} 2 \cdot u_{\text{right}}^W - u_N \\ v_N + g \cdot \Delta x \\ -g \cdot \epsilon \end{pmatrix}, \quad U_{N+2} = \begin{pmatrix} 2 \cdot u_{N+1} - u_N \\ v_N + g \cdot 2 \cdot \Delta x \\ -g \cdot \epsilon \end{pmatrix} \tag{46}$$

3.9 Variant 9: same as variant 8 but with 2nd order approximation for v

The second equation of the stationary PDE includes a derivative of v with respect to x. In this method, we use a second order finite difference approximation to discretize the derivative. Apart from that, this variant uses the same formula than variant 8. That leads to

$$v_0 = \frac{3}{4} v_1 - \frac{1}{3} v_2 - g \cdot \Delta x \tag{47}$$

$$v_{-1} = \frac{3}{4} v_0 - \frac{1}{3} v_1 - g \cdot \Delta x \tag{48}$$

$$v_{N+1} = \frac{3}{4} v_N - \frac{1}{3} v_{N-1} + g \cdot \Delta x \tag{49}$$

$$v_{N+2} = \frac{3}{4} v_{N+1} - \frac{1}{3} v_N + g \cdot \Delta x \tag{50}$$

3.10 Variant 10: GC extrapolation 2nd order, v, w from stationary PDE

As a combination of variant 4 and 6, we here use a second order extrapolation to the ghost cells and boundary values from the stationary PDE. So we use equations (35) to (40) to get the boundary values and afterwards extrapolate to the ghost cells using equations (27) to (30) Similar to method 6, there is no need for any extrapolation to the boundary as the boundary values are set directly.

3.11 Variant 11: GC extrapolation 2nd order, v, w from stationary PDE, 2nd order approximation for v

The method is basically the same as variant 10, but here we use second order approximations for the v-derivative of the stationary PDE on the boundary. Note that this is not the same as method 9 as the distances between the neighbour points are different for ghost cell points and boundary points. The calculation of v on the boundary before applying the boundary condition on u is as follows

$$v^W_{\text{left}} = \frac{3}{8}(-\Delta x \cdot g + 3v_1 - \frac{1}{3}v_2) \tag{51}$$

$$v^W_{\text{right}} = \frac{3}{8}(\Delta x \cdot g + 3v_N - \frac{1}{3}v_{N-1}) \tag{52}$$

3.12 Variant 12: like 7 with u 2nd order extrapolation to ghost cells and 2nd order approximation of v-derivative

For the most complex method, we combine boundary conditions to v, w from stationary PDE using 2nd order approximation for the v-derivative and 2nd order extrapolation to the u value on ghost cells. The other values v, w are directly set on the ghost cells from the stationary PDE with 2nd order approximation again. Especially the v-derivative approximation on the ghost cells is difficult because of the different step sizes with respect to the boundary value. The formula for the ghost cell values read as follows

$$U_0 = \begin{pmatrix} \frac{1}{3}(8u^W_{\text{left}} - 6u_1 + u_2) \\ \frac{1}{3}(-\Delta x \cdot g - v^W_{\text{left}} + 4v_1) \\ -g \cdot \epsilon \end{pmatrix}, \quad U_{-1} = \begin{pmatrix} 8u^W_{\text{left}} - 9u_1 + 2u_2 \\ \frac{6}{7}(-\Delta x \cdot g + 1.5v_0 - \frac{1}{3}v^W_{\text{left}}) \\ -g \cdot \epsilon \end{pmatrix} \tag{53}$$

$$U_0 = \begin{pmatrix} \frac{1}{3}(8u^W_{\text{right}} - 6u_N + u_{N-1}) \\ \frac{1}{3}(\Delta x \cdot g - v^W_{\text{right}} + 4v_N) \\ -g \cdot \epsilon \end{pmatrix}, \quad U_{-1} = \begin{pmatrix} 8u^W_{\text{right}} - 9u_N + 2u_{N-1} \\ \frac{6}{7}(\Delta x \cdot g + 1.5v_{N+1} - \frac{1}{3}v^W_{\text{right}}) \\ -g \cdot \epsilon \end{pmatrix} \tag{54}$$

3.13 Variant 13: like 4 with BC imposed on u instead of v (combination of 4 and 2)

We here first extrapolate to the wall and then impose the boundary condition on u. Afterwards, we extrapolate to the ghost cells using 2nd order approximation. The formula are explained in variant 4 and 2 respectively.

3.14 Variant 14: like 10 but with BC on u from extrapolated inner values (combination of 10 and 7)

We here first extrapolate to the wall and then impose the boundary condition on u. Afterwards, we set the values for v, w from the stationary PDE and extrapolate to the ghost cells using 2nd order approximation.

3.15 Variant 15: like 10 but using the stationary PDE to get u and impose BC on v

This method uses the stationary solution. We first solve (5) using (3) to obtain the value for w:

$$w^W_{left/right} = -\epsilon g; \tag{55}$$

With $w = const$ we can impose the boundary condition on v and insert it into the second equation (4) to get

$$\partial_x u_{\pm 1} = -\frac{2}{\epsilon} v = -\frac{2}{\epsilon}(\pm(\alpha u + \beta w)_{\pm 1}; \tag{56}$$

We then discretize the derivative of u using a forward difference $\partial_x u_{-1} \approx \frac{u_1 - u_l^W eft}{1/2 \ \Delta x}$ to get an equation we can solve for u. This leads to

$$u_l^W eft = \frac{-u_1 + \frac{\Delta x}{\epsilon}\beta w_l^W eft}{-\frac{\Delta x}{\epsilon}\alpha - 1} \tag{57}$$

The equation for the other boundary value can be obtained analoguously

$$u_r^W ight = \frac{+u_1 - \frac{\Delta x}{\epsilon}\beta w_r^W ight}{\frac{\Delta x}{\epsilon}\alpha + 1} \tag{58}$$

4 Results

To test the different approaches, we performed several numerical simulations with the following settings.

We choose the stability parameter $\nu = 0.89$ for IMEX and $\widetilde{\nu} = 0.89$ for Heun as well. Note that this means larger time step sizes for IMEX method because of the improved time step constraint. Furthermore, we let the simulation run from the exact steady state (8) as initial condition and stop at time $t = 5$. We measure the error at time $t = 5$ in the L^1 as well as in the L^∞ norm in order to get the maximum error in the domain. For the spatial discretization we choose $N = 10, 20, 30, 40, 50$ corresponding to timesteps $\Delta t = 0.0263, 0.0137, 0.0127, 0.0119$ for Heun and $\Delta t = 0.1780, 0.0890, 0.0593, 0.0445, 0.0356$ for IMEX. Then, we calculate the order of convergence using linear regression of the corresponding double-log data.

For each time integration method (IMEX or Heun) we end up with the order of convergence for every variant $1, \ldots, 12$ and component u, v, w and the relative error for the highest resolution $N = 50$ for every method and component as well.

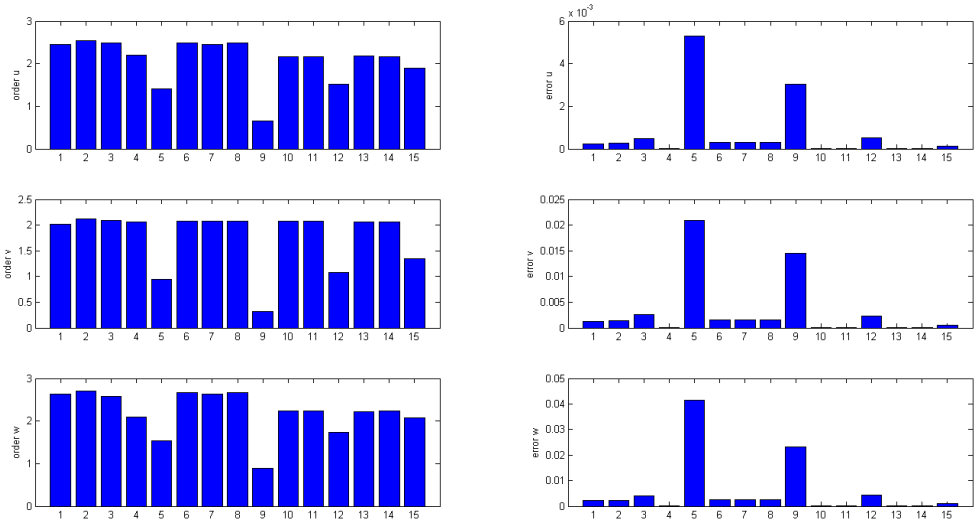

Figure 2: results for Heun method with error measured in L^1 norm. Left: orders of convergence. Right: relative errors for highest resolution ($N = 50$)

Figure 2 shows, that the simple Heun method comes up with the second order convergence for every component and almost every variant. Even the first approach leads to this desired order and this can hardly be improved using one of the other methods. It can be seen, that the third component w has the highest order of convergence of about 2.3 which will be different for the IMEX method.

As for the relative errors of the different methods, we observe a different behaviour. We can see, that most of the methods with second order extrapolation to the ghost cells show a lower error than the standard approach. At this point it seems obvious that method 5 (constant extrapolation of second ghost cell layer) and method 9 (second derivative approximation for stationary values) have a much larger error than each of the other methods. Summarizing, the errors have the same relative behaviour for all the three components, though the scales are quite different.

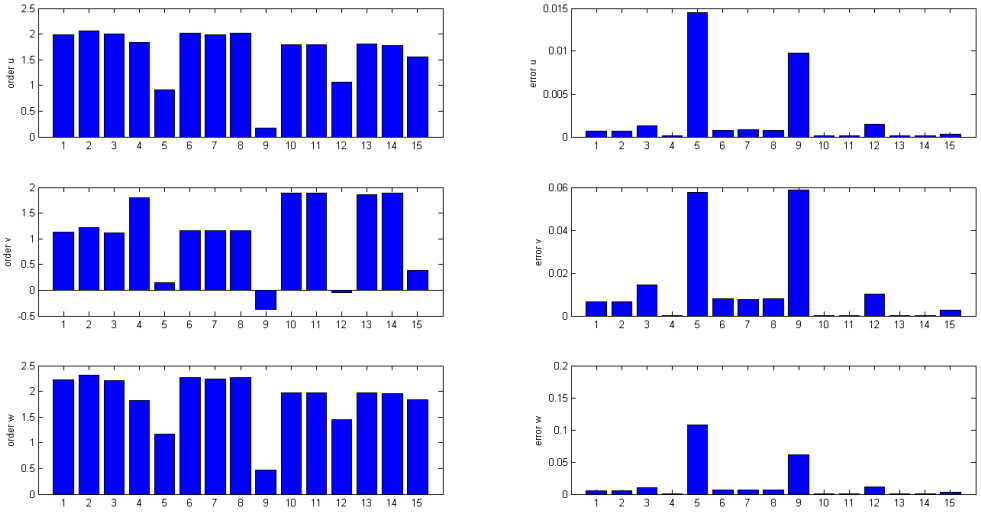

Figure 3: results for Heun method with error measured in L^∞ norm. Left: orders of convergence. Right: relative errors for highest resolution ($N = 50$)

In contrast Figure 3 shows the same simulation as Figure 2 but with the error measured in the discrete L^∞ norm. So this error shows how the maximum error in the domain is reduced. As we expect the maximum error to occur near the boundary, this norm shows us the quality of the boundary treatments.

Again, we have a second order convergence for u and w, but not for v. The results for the v-component only show a higher rate of convergence for methods $4, 10, 11, 13, 14$, each with second order ghost cell extrapolation. Though these methods come up with slightly reduced convergence orders in the first and third component.

The relative error shows the same behaviour as we have already seen in Figure 2 abain with best results for methods $4, 10, 11, 13, 14$.

As these methods obviously are the only one, that can guarantee a second order convergence in the L^∞ norm, we will from now on only examine those five methods compared to the usual boundary treatmend in method 1. That is the reason, why we only see those columns in the following figures.

Now we come to the tests with the IMEX method. As already said Figure 4 shows the results from the same test settings as the Heun method but the time steps are larger due to the

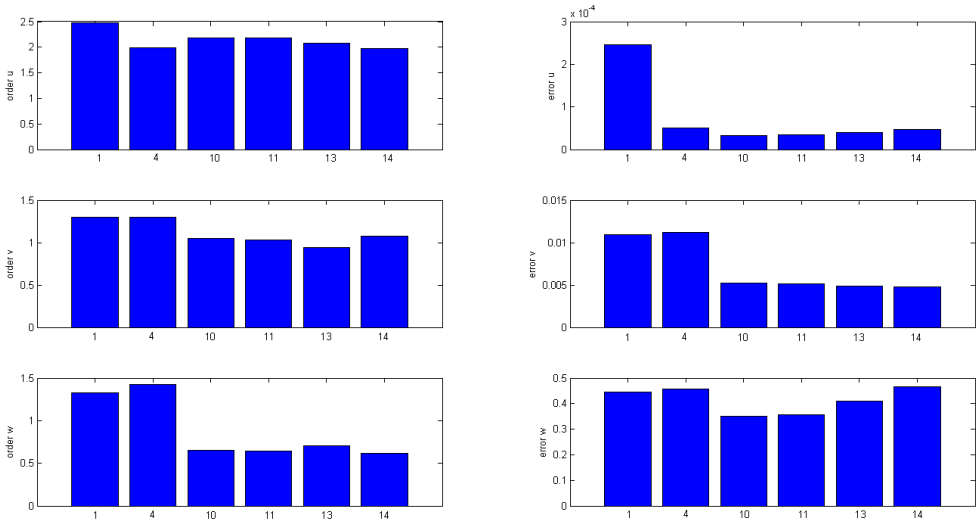

Figure 4: results for IMEX method with error measured in L^1 norm. Left: orders of convergence. Right: relative errors for highest resolution ($N = 50$)

improved time step constraint.

The convergence rates are only good for the first component. The second and third component come up with rates of about 1.3 which is much smaller than the values for the Heun method. Another interesting observation is that the order of convergence is much smaller for all the methods, that use values from the stationary PDE on the boundary or for the ghost cells (methods 6-12 cannot be seen in the figure). Nevertheless, method 4 gives the best results for the third component using only the second order ghost cell extrapolation.

The relative error is again very small for methods $4, 10, 11, 13$ and 14 at least for the first component while there are not many differences for the other two components. Compared to the Heun method, the scale for u is the same, while the second and third component v and w have a higher error than the Heun method results. Note again, that the Heun method takes many more time steps compared to the IMEX method.

Figure 5 finally shows the L^∞ convergence rate and error of the IMEX method. Here the disadvantages of the IMEX method are most obvious. The convergence rate for the u component is still quite good, but the second component shows a very slow convergence for the initial approach. This convergence rate can be improved by method 4 for example but will not be higher than 0.6. The w component does not show the desired error reduction, too. Again method 4 gives the best results but still has a quite small rate of convergence.

Compared to the L^1 norm in Figure 4 the error here has the same behaviour. Again especially methods $10, 11, 13$ and 14 are able to reduce the error in u significantly. In comparison to Heun method we observe a much higher error in v, w while the error in u is almost the same.

5 Conclusion

The different test results have shown the importance of the boundary treatment for this type of PDE. In general, the error can be reduced significantly by using the highest order extrapolation method. This has been demonstrated using method 4 or $10, 11$ respectively. Besides that, the Heun and IMEX method show very different results. The use of the stationary PDE for boundary and/or ghost cell values has not neither reduced the error nor really improved the convergence

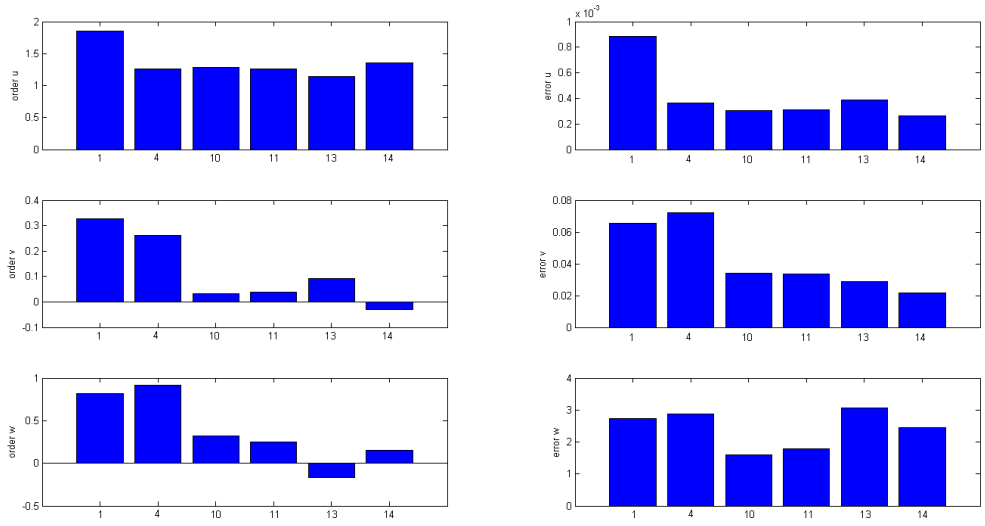

Figure 5: results for IMEX method with error measured in L^∞ norm. Left: orders of convergence. Right: relative errors for highest resolution ($N = 50$)

order. Furthermore, the methods that use the stationary PDE come up with a much slower convergence for the third component when using the IMEX method.

Even the combination of different variants did not lead to a better convergence rate and error for all the three components. Some methods were able to reduce certain components of the error while spoiling the accuracy for another component.

In summary it is a good choice to use second order extrapolation to the ghost cells (method 4). Due to that, the error in the first component can be reduced while improving the rate of convergence of the other components.

References

[1] L. Pareschi, G. Russo, Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation, 2003, (available online http://www.math.ntnu.no/conservation/2004/063.pdf or http://web.unife.it/utenti/lorenzo.pareschi/talks/imex_slides.pdf)