

Report on weekly progress; meeting 02/04/2020

Perturbation Functions Considered

The one from last week:

$$f(t) = \frac{e^{-1/t}}{e^{-1/t} + e^{-1/(1-t)}}.$$

An new perturbation is given by:

$$g(t) = \frac{e^{-a/(t-t_0)}}{e^{-a/(t-t_0)} + e^{-a/(1-t-t_0)}} \times \frac{e^{a/(t-t_0)}}{e^{a/(t-t_0)} + e^{a/(1-t-t_0)}},$$

where $a = 0.7$ (mostly) and $t_0 = -0.01$. The translation by t_0 is due to the fact that otherwise you'd get NaNs for $t = 0$. The factor a is flattening or sharpening the peak. While $g(t)$ is perturbed up to 0.04 (see Figure 1, perturbation in time), this does not translate to the perturbation in w . For a perturbation in w of 0.1, you need $0.05g(t)$ (see Figure 1 for perturbation $0.1g(t)$).

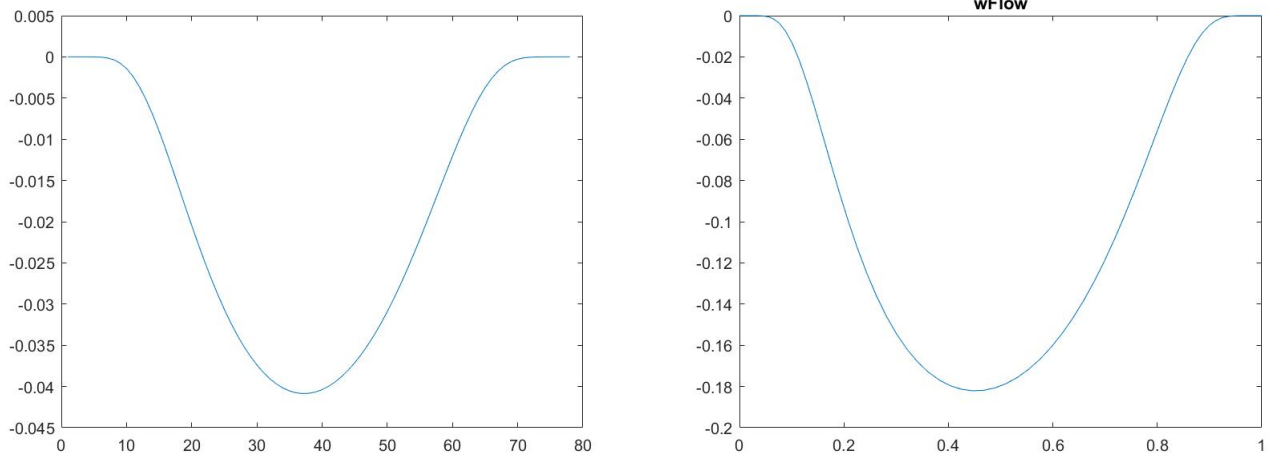


Figure 1: Perturbation $g(t)$ (left), w perturbed by $0.1g(t)$ (right).

Kalise Algorithm

Exact Solution Errors

I have noticed that when measuring the error in the L2Linfinity sense (as done in the past weeks), this algorithm shows a difference of up to 2 orders of magnitude between w, p and ρ . The most severe case is the mixed BC one, where the difference can go up to 4 orders of magnitude (depending on β - smaller β increase the error). This seems to be mitigated by measuring the error pointwise. Furthermore, distributing β in the exact solutions between ρ

and p instead of just putting it on p , seems to improve the situation in the mixed boundary condition case. This has not shown to be a problem in the Multiple Shooting algorithm. Using the exact solution that is linear in time for the Mixed BCs (see other document) solves the problem in the L2Linf norm as well. The algorithm error is much less than the ODE tolerance for all variables. This pattern is also not there for Dirichlet/Neumann solutions with linear time (i.e. trig in space, linear in time).

Neumann Flow Control (plus2) $\beta = 10^{-3}$

Further to last week, I have tested Neumann (plus 2) flow control with $\beta = 10^{-3}$. This converges in 7210 Iterations with $\lambda = 1$, tolerances = $10^{-9}/10^{-5}$ and $n = 74$, $N = 72$. The perturbation used is $0.1f(t)$, as before. The consistency error is $w = 9.9920 \times 10^{-6}$ and the exact errors are $w_{err} = 9.4017 \times 10^{-6}$, $p_{err} = 5.8640 \times 10^{-5}$ and $\rho_{err} = 8.0369 \times 10^{-6}$, see Figure 2.

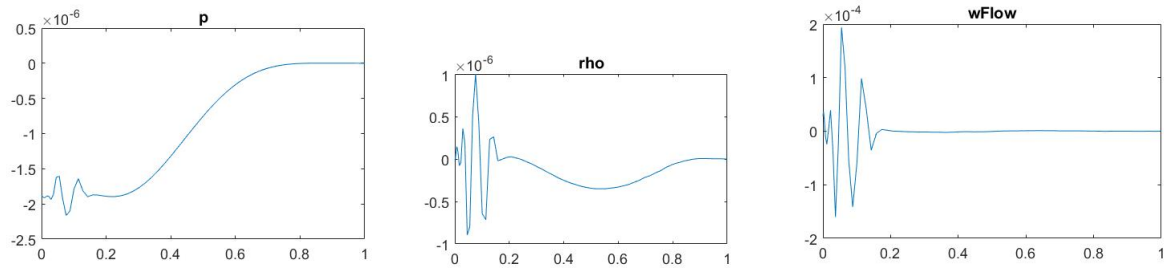


Figure 2: Exact Errors, Neumann Flow Control $\beta = 10^{-3}$, perturbation $0.1f(t)$.

Mixed Boundary Conditions

Using 'ADFlowMixedExact' as exact solution (see exact solution for mixed BCs). Here, $\beta = 10^{-1}$, tolerances are $10^{-9}/10^{-5}$, $\lambda = 0.01$, $n = 74$, $N = 72$. The perturbation is again $0.1f(t)$. This converges in 7463 Iterations, consistency error is 9.9958×10^{-6} , exact errors are $w_{err} = 2.4073 \times 10^{-5}$, $p_{err} = 2.4526 \times 10^{-5}$ and $\rho_{err} = 7.4239 \times 10^{-7}$, see Figure 2. Note that the errors are two orders of magnitude apart in the chosen norm, however not on the figure. Compare with exact solution above.

Investigating Pointwise Error for Kalise

Investigating Neumann (plus2) Flow Control shows that for $\beta = 10^{-1}$ with $\lambda = 0.01$ and 0.005 , the result diverges after a while. The relative error in p is 7.49.... Trying $\beta = 10^{-3}$ diverges after three iterations when trying $\lambda = 1, 0.5, 0.01$. Given that this converges in the L2Linfy norms, it doesn't seem like pointwise error is good for the convergence of the Kalise method. (However, it does fix the odd behaviour in exact solution - see above.)

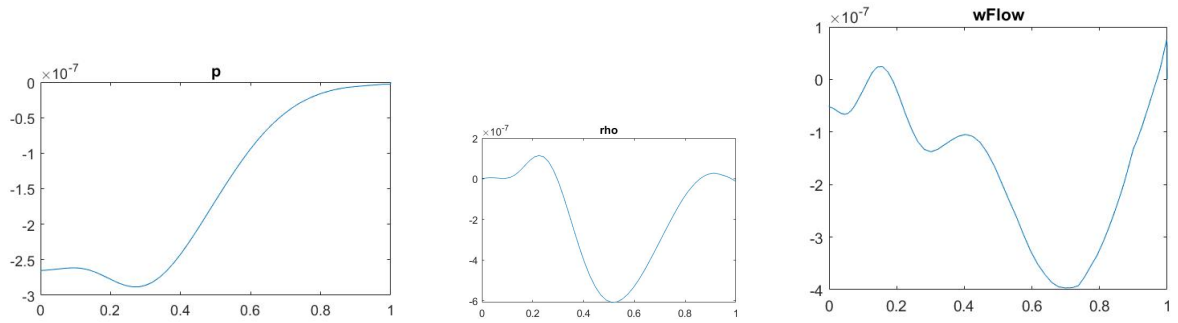


Figure 3: Exact Errors, Neumann Flow Control $\beta = 10^{-3}$, perturbation $0.1f(t)$.

Doing the same thing for Dirichlet flow with $\beta = 10^{-1}$ and $\lambda = 0.01$, it diverges at a consistency of 3.4502×10^{-4} at Iteration 2776. This is using the perturbation $0.1f(t)$, with tolerances $10^{-9}/10^{-5}$ and $n = 74$, $N = 72$. Same happens with 10^{-12} . When giving the diverged result to the multiple shooting algorithm, it diverges further, so that's not a solution.

Checking that Multiple Shooting converges given the Kalise solution

Using the Kalise result for the Neumann Flow Control Problem with $\beta = 10^{-1}$ and giving this to the multiple shooting algorithm as an initial guess for ρ and p . The Kalise algorithm converges in 7209 Iterations (compare with above), with $\lambda = 0.01$, tolerances $= 10^{-12}/10^{-5}$, $n = 74$, $N = 72$. The consistency error is $w = 9.9992 \times 10^{-6}$. Then the multiple shooting, with $\lambda = 0.1$, has an initial consistency error of 0.00001058, with exact errors $\rho_{err} = 0.00000802$ and $p_{err} = 0.00005765$. This converges in 20 Iterations to a consistency of 0.00000997, with exact errors $\rho_{err} = 0.00000821$ and $p_{err} = 0.00005745$. So the exact error in ρ diverges, while the one in p converges. Overall, the Kalise solution is a solution to the multiple shooting algorithm as well.

Multiple Shooting

Using Kalise as initial guess, then Multiple shooting

This is doing one Kalise step first to get an initial guess for p (instead of integrating). In the pointwise norm the error in p is 7.49588... and in ρ 0.3589.... This eventually diverges and the error is as erratic as we saw last week. This is Neumann (plus2) Flow Control, $\beta = 10^{-1}$ and perturbation $0.1f(t)$.

Multiple Shooting with Integration and PW error

Using the same technique as last week by getting an initial guess for p from integrating the gradient equation. This also converges for a while before diverging, as last week. This is

Neumann (plus2) Flow Control, $\beta = 10^{-1}$ and perturbation $0.1f(t)$.

Different Perturbations

Using the different perturbations introduced above (in very first section).

Kalise Algorithm

Using L2Linf norm (since PW doesn't seem to converge), $\text{tols} = 10^{-9}/10^{-5}$, $n = 74$, $N = 72$. Considering Neumann (plus2) flow control and a perturbation of $0.1g(t)$. For $\beta = 10^{-1}$ and $\lambda = 0.01$, it converges. It needs a bit more than 5000 Iterations and the exact errors are $w_{err} = 1.4136 \times 10^{-5}$, $p_{err} = 7.7223 \times 10^{-5}$ and $\rho_{err} = 1.3989 \times 10^{-5}$ (see Figure 4). So here, to get 10^{-5} accuracy, the consistency tolerance would have to be lower. When investigating this, we set the consistency tolerance to 10^{-6} and leave everything else the same. The algorithm diverges at $w = 5.1200 \times 10^{-6}$, with exact errors of $w_{err} = 1.8839 \times 10^{-5}$, $p_{err} = 4.5021 \times 10^{-5}$ and $\rho_{err} = 1.8175 \times 10^{-5}$. This shows that we cannot get to an error of $O(10^{-5})$ with these configurations.

The next thing to try is $\beta = 10^{-3}$ and $\lambda = 1$. This also converges in 5980 Iterations. Then the

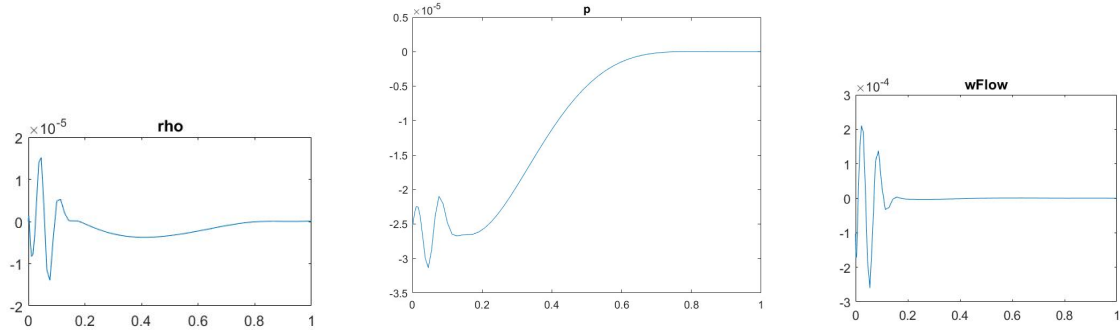


Figure 4: Exact Errors, Neumann Flow Control $\beta = 10^{-1}$, perturbation $0.1g(t)$.

exact errors are $w_{err} = 1.6386 \times 10^{-5}$, $p_{err} = 7.7389 \times 10^{-5}$ and $\rho_{err} = 1.5156 \times 10^{-5}$. This is very similar to the case when $\beta = 10^{-3}$.

Now it should be investigated how $0.05g(t)$ converges, considering that this is a 10% perturbation in w .

Dirichlet Flow control is not working as well as Neumann. Using the L2Linf errors, $\beta = 10^{-1}$, $\text{tols} = 10^{-9}/10^{-5}$ and $\lambda = 0.01$. When perturbing by $0.1g(t)$, the algorithm diverges after 1864 Iterations at a consistency of 2.4847×10^{-4} . The exact errors at that point are $w_{err} = 4.2662 \times 10^{-4}$, $p_{err} = 2.4504 \times 10^{-4}$ and $\rho_{err} = 3.5706 \times 10^{-4}$, see Figure 5. Measuring the pointwise error of this result is worse. When choosing $0.01g(t)$, the algorithm diverges at $w = 2.4800 \times 10^{-5}$ and

when choosing $0.01g(t)$ with $a = 0.5$ it diverges at 5.3647×10^{-5} at Iteration 1524. When taking the result and translating it to pointwise errors we get $w_{err} = 5.1272 \times 10^{-4}$, $p_{err} = 1.1953 \times 10^{-4}$ and $\rho_{err} = 2.6470 \times 10^{-4}$. When using pointwise error as the convergence measure from the start for this problem, it diverges at $w = 2.8653 \times 10^{-4}$ with exact errors of $w_{err} = 5.3481 \times 10^{-4}$, $p_{err} = 1.2412 \times 10^{-4}$ and $\rho_{err} = 2.8613$. This shows that using the pointwise error for this problem, the convergence is slightly worse than if we use L2Linf error. There is no apparent reason for this to not converge since the perturbation in w is small, see Figure 6.

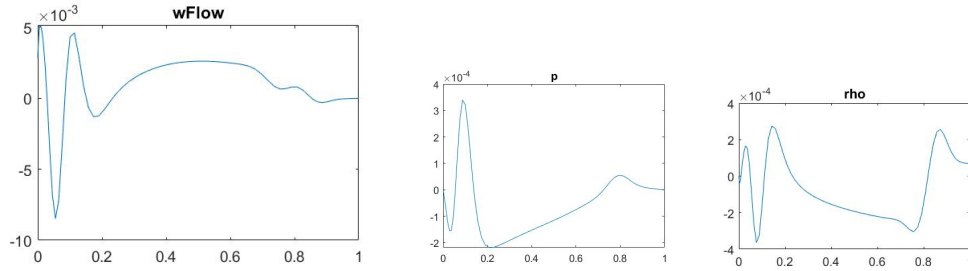


Figure 5: Exact Errors, Dirichlet Flow Control $\beta = 10^{-1}$, perturbation $0.1g(t)$.

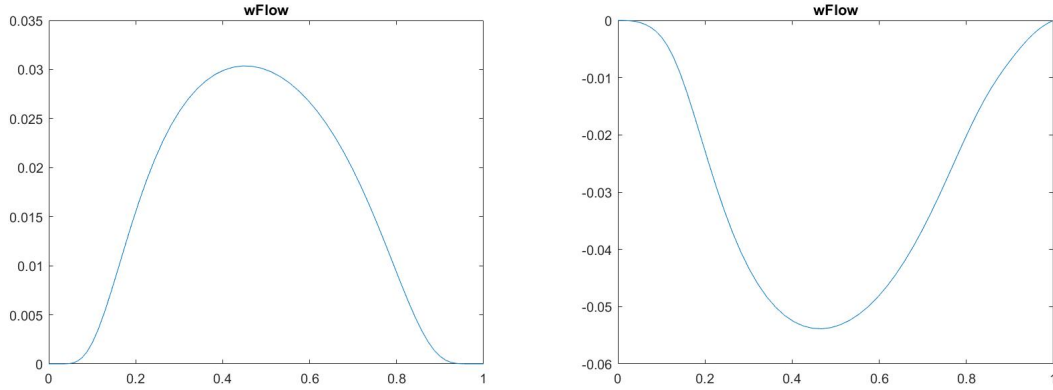


Figure 6: Perturbation in w (first and second vs exact solution), perturbation $0.01g(t)$, $a = 0.5$.

Mixed Boundary conditions with a linear time relationship (see document for exact solution).

Multiple Shooting

Using Neumann (plus2) and Flow control at first we try $0.1g(t)$ as perturbation, with again $N = 72$, $n = 74$, $\text{tols} = 10^{-9}/10^{-5}$, $\beta = 10^{-1}$ and $\lambda = 0.1$. Using the pointwise error measure, this converges to a consistency of 0.00514602 at Iteration 214, while $\rho_{err} = 0.00011263$ at Iteration 211 and $p_{err} = 0.06646699$ at Iteration 247 are the lowest individual exact errors. This is from an initial error of 0.05973065 with exact errors of $\rho_{errI} = 0.06184073$ and $p_{errI} = 0.02023870$. The initial guess for p is found by integrating the gradient equation. Finding the initial guess for

p using the Kalise algorithm makes it worse and it converges only to 0.03... consistency. Using the L2Linfinity error with a change in the perturbation $0.1g(t)$ with $a = 0.5$ instead (Doesn't significantly change errors in pointwise measure, makes it slightly worse), it gives better results. It converges to 0.00043827 at Iteration 218 and with $\rho_{err} = 0.00009809$ at Iteration 202 and $p_{err} = 0.00445544$ at Iteration 245. So at least this example suggests that the L2Linf error is a better fit. Investigating the possible perturbations that might converge, the next choice is $0.01g(t)$ and $a = 0.5$ with L2Linf norm. The initial error is 0.00282653 with $\rho_{errI} = 0.00198820$ and $p_{errI} = 0.00045162$ (p diverges before converging). This diverges at 0.00004813 at Iteration 196 with $\rho_{err} = 0.00001212$ and $p_{err} = 0.00048389$ at Iteration 197 with p converging further after that, see Figure 7 and 8. Next, try to decrease λ to see if it improves convergence. It actually makes the convergence slightly worse 0.000049... and takes a lot longer.

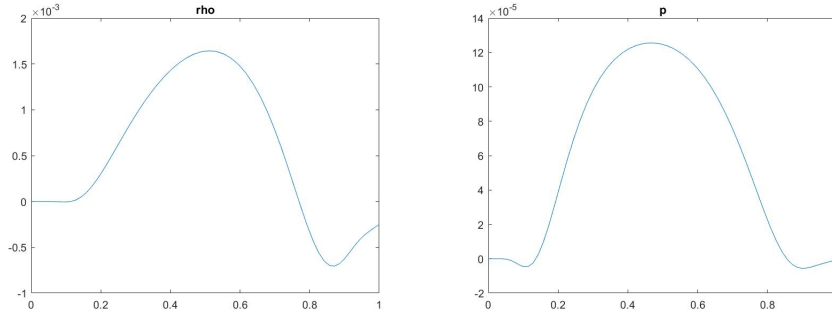


Figure 7: Initial Exact Errors, Neumann Flow Control $\beta = 10^{-1}$, perturbation $0.01g(t)$, $a = 0.5$.

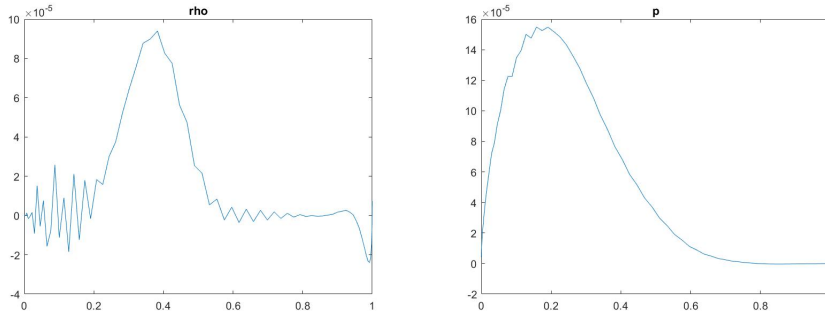


Figure 8: Exact Errors, Neumann Flow Control $\beta = 10^{-1}$, perturbation $0.01g(t)$, $a = 0.5$.

Linear time exact solution trials

Kalise

We are using the exact solution 'Mixed3' with Dirichlet and Neumann BCs. All errors are measured in L2Linf, tolerances and n, N as above. In the Dirichlet Case both $\beta = 10^{-1}$ ($\lambda = 0.1$)

and $\beta = 10^{-3}$ ($\lambda = 10$) converged. The perturbation was $10g(t)$ with $a = 0.7$, see Figure 9 and 10. For $\beta = 10^{-1}$ the initial exact error was $w_{errI} = 0.0324$, $p_{errI} = 0.0557$ and $\rho_{errI} = 0.0421$. Then after 1157 iterations it converged to $w_{err} = 2.2602 \times 10^{-7}$, $p_{err} = 4.2460 \times 10^{-7}$ and $\rho_{err} = 31175 \times 10^{-7}$. The errors for $\beta = 10^{-3}$ are similar.

The results for the Neumann case are very similar. The algorithm converges for both β values. For $\beta = 10^{-1}$ the initial exact errors are $w_{ErrI} = 0.0309$, $p_{errI} = 0.0755$ and $\rho_{ErrI} = 0.0466$. Then in 1175 Iterations, this converges to $w_{err} = 2.1936 \times 10^{-7}$, $p_{Err} = 6.5744 \times 10^{-7}$ and $\rho_{err} = 3.4265 \times 10^{-7}$. The results for $\beta = 10^{-3}$ are again similar.

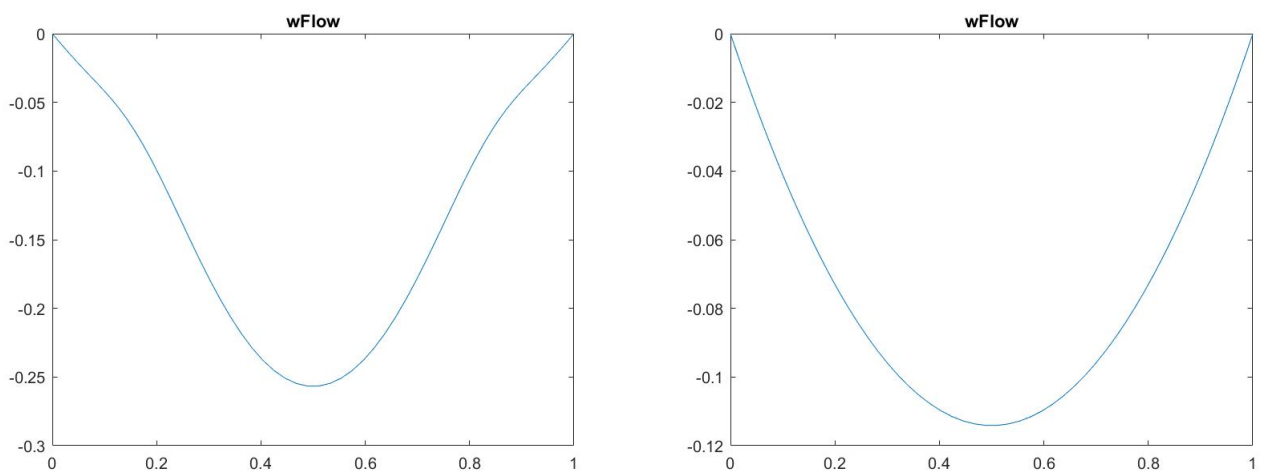


Figure 9: Perturbed w and exact w .

Now, investigating Neumann (plus 2) with linear time terms (i.e. exact solution with trig functions in space, and linear time) gives the following: With $\beta = 10^{-1}$ the initial errors are $w_{errI} = 0.2353$, $p_{errI} = 0.8576$ and $\rho_{errI} = 0.3192$. This converges in 1167 Iterations to $w_{err} = 6.9449 \times 10^{-7}$, $p_{err} = 8.3209 \times 10^{-6}$ and $\rho_{err} = 2.1389 \times 10^{-6}$. The results are very similar for $\beta = 10^{-3}$ and can be seen in Figure 12. The perturbation in w can be seen in Figure 11. Note this also converges to a tolerance of 10^{-6} instead of 10^{-5} .

Looking at Dirichlet Exact solutions with linear time terms (i.e. trig in space, linear in time), this also converges for the two β values. Accidentally the tolerance was set to 10^{-6} which converged as well ($\beta = 10^{-1}$). The perturbations in w can be seen in Figure 13. The results are $w_{err} = 8.6507 \times 10^{-7}$, $p_{err} = 2.9450 \times 10^{-7}$ and $\rho_{Err} = 10^{-7}$.

Considering $\beta + 10^{-3}$ seems to be a bit more difficult, simply because the scaling makes the perturbation much larger for this case. Therefore, different perturbations have to be tried, to see which one perturbs w a reasonable amount. This is most likely due to the fact that the

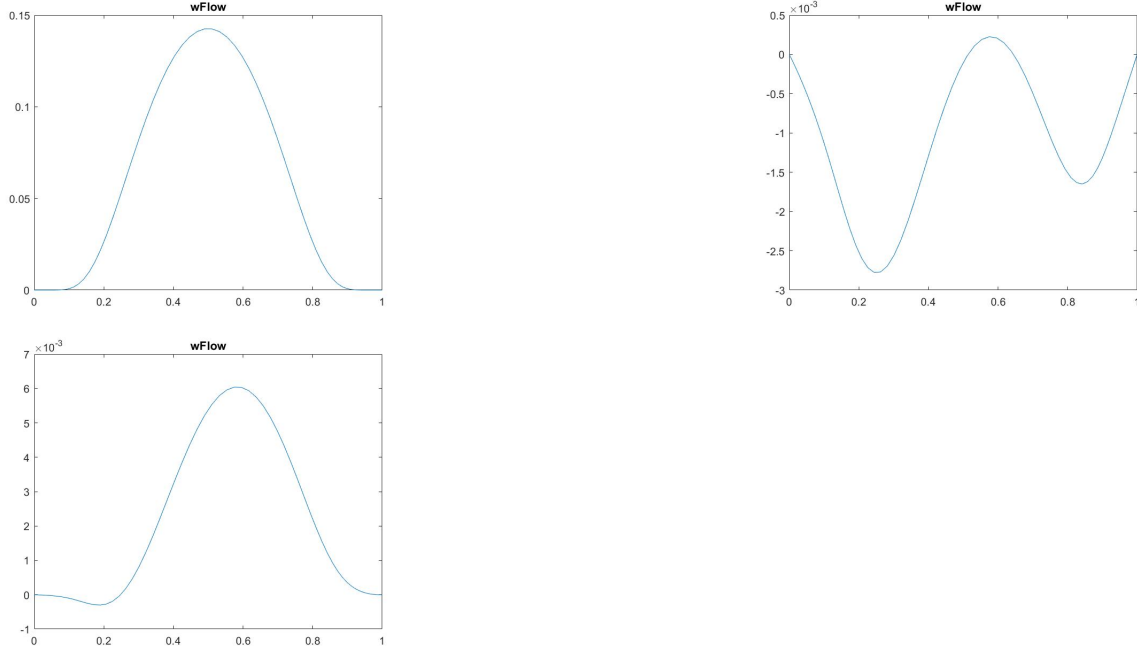


Figure 10: Perturbed w Error (first - left) (second: Dirichlet - Mid, Neumann - right).

Dirichlet exact solution for w scales like $1/\beta$ (to be consistent with the original exact solutions). This should be checked.

Multiple Shooting

Using 'Mixed3' again, and the perturbation $10g(t)$, with $\beta = 10^{-1}$ and measured in L2Linf norm, Multiple Shooting converges within 724 Iterations. The exact errors are then $\rho_{err} = 0.00000791$ and $p_{err} = 0.00015429$, see Figure 14. The initial errors were $cons_{err} = 0.02820414$, $\rho_{ErrI} = 0.03215880$ and $p_{err} = 0.01302147$.

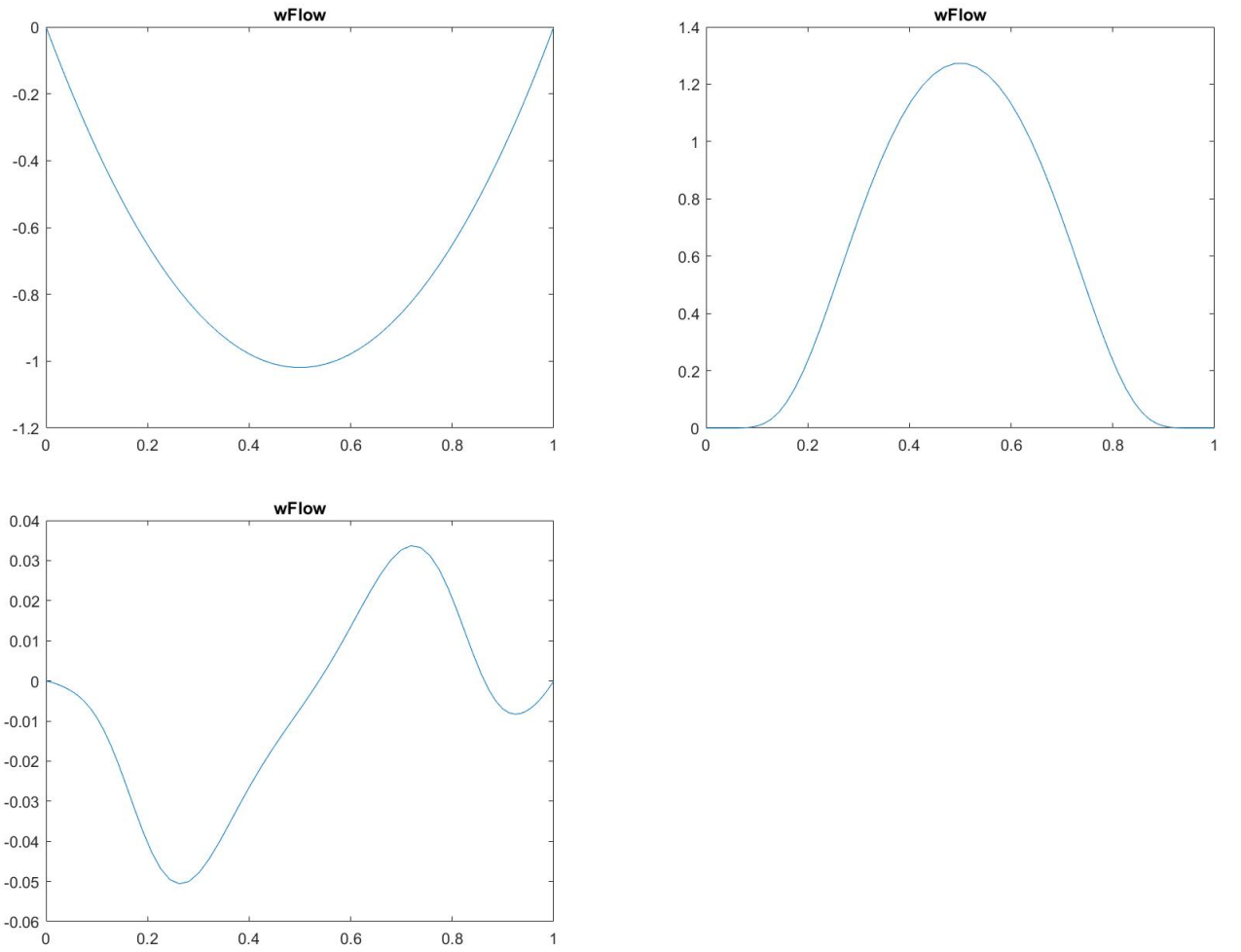


Figure 11: This is w exact (left), w first error (mid) and second error (right). Here with $\beta = 10^{-3}$ and perturbation $10g(t)$.

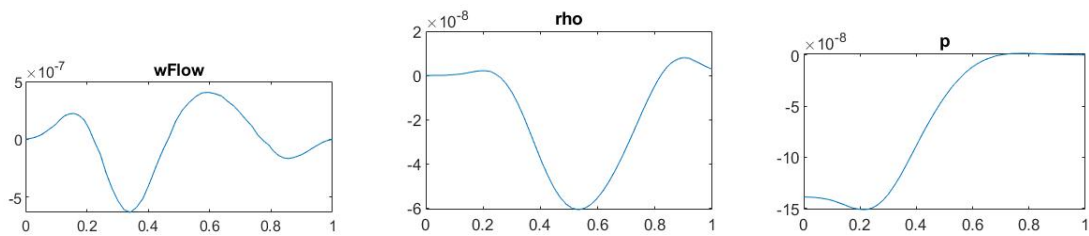


Figure 12: Final error in variables when Neumann with linear time converged. Here with $\beta = 10^{-3}$ and perturbation $10g(t)$.

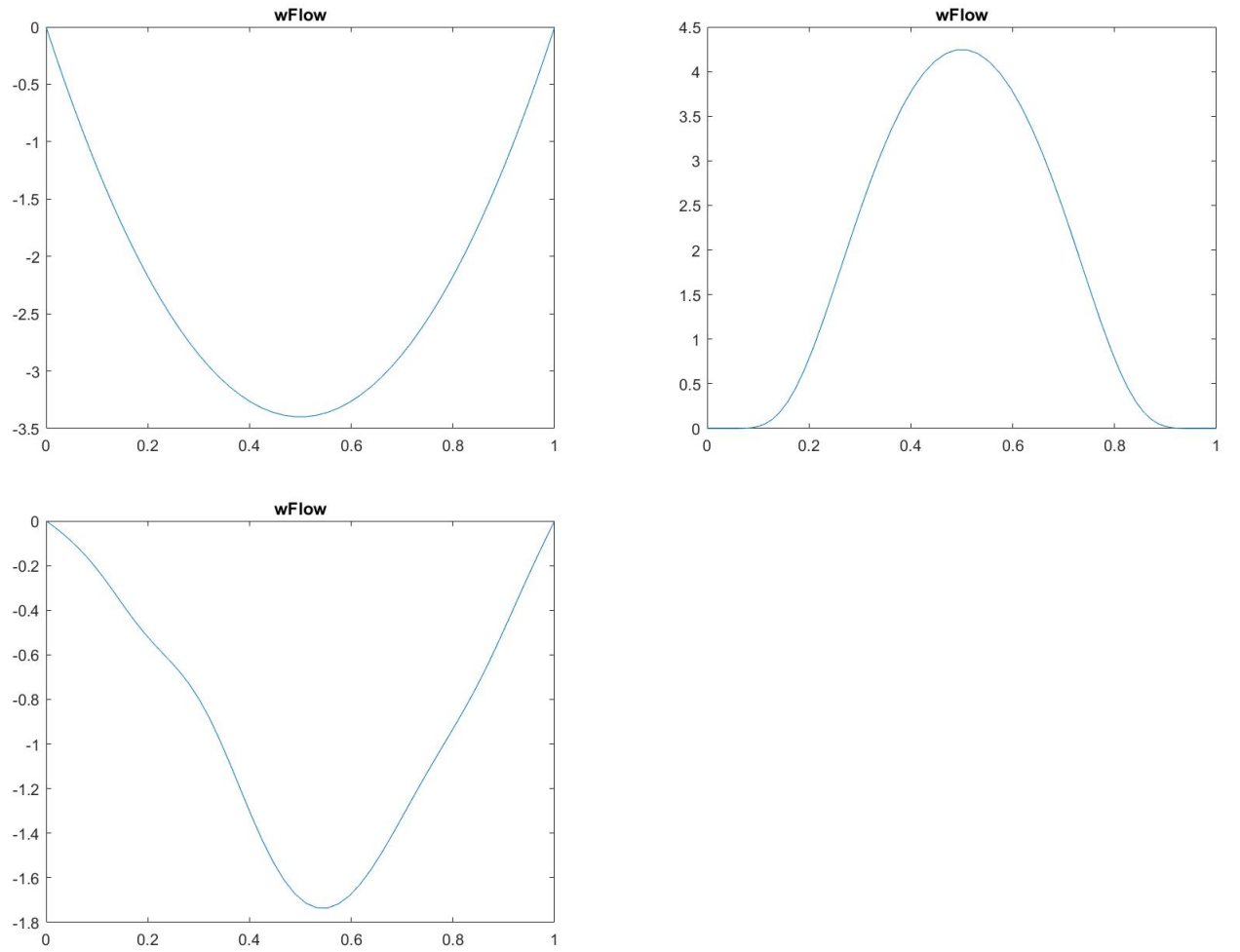


Figure 13: This is w exact (left), w first error (mid) and second error (right). Here with $\beta = 10^{-1}$ and perturbation $10g(t)$.

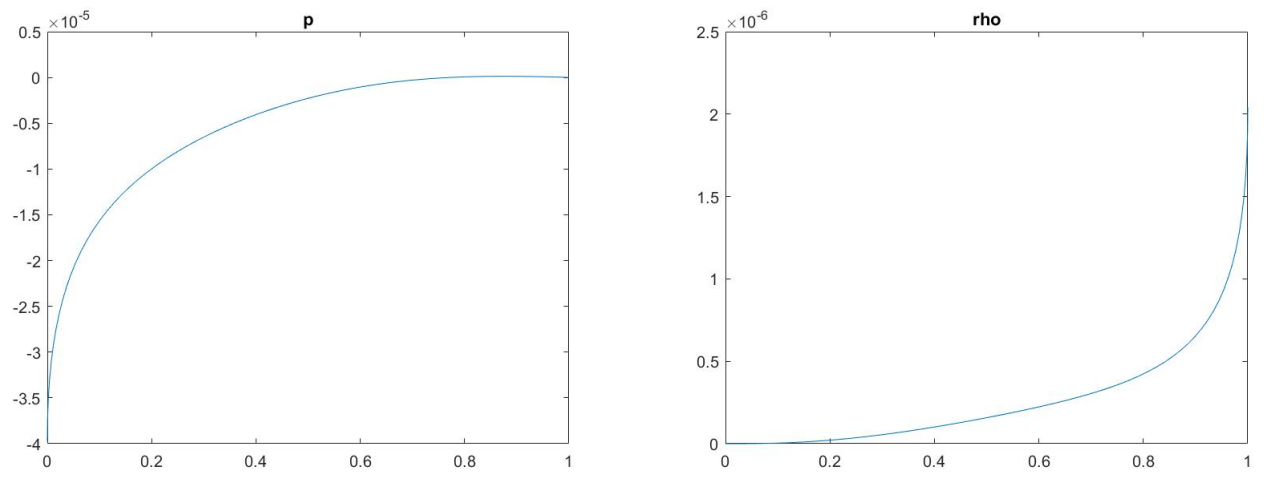


Figure 14: Final exact error in ρ and p . Here with $\beta = 10^{-1}$ and perturbation $10g(t)$.