Understanding and Improving Model Performance at Small Mass Flow Rates in Fluid System Models

Robert Flesch[1] Annika Kuhlmann[1] Johannes Brunnemann[1] Jörg Eiden[1]

[1]XRG Simulation GmbH, Germany, {flesch,kuhlmann,brunnemann,eiden}@xrg-simulation.de

Abstract

This paper provides a detailed analysis of the reasons behind the poor simulation performance observed when mass flow rates become very small, commonly referred to as zero mass flow issues. By using simple example models, we effectively demonstrate the underlying causes of these simulation performance issues. We highlight various contributing factors that play a significant role in exacerbating the problem.

Furthermore, we propose and examine countermeasures to mitigate these challenges. These countermeasures include modifications to the model itself, utilization of available settings in simulation tools, and adjustments to the solver. By implementing and evaluating these countermeasures, we illustrate their impact on improving simulation performance in scenarios involving low mass flow rates.

Keywords: zero mass flow issue, fluid dynamics, ODE integration, non-linear modeling

1 Introduction

Modelica is a great option to easily create models to simulate complex fluid systems. The created models can be simulated in one of the available tools. Thanks to advancements in algorithms and computational power, it is now possible to model these intricate systems within a short period of time.However, there is a challenge when it comes to systems that contain branches with no flow during certain periods or when the model is used to simulate ramp-up or shut-down sequences. In such cases, simulation time can drastically increase, resulting in what is commonly known as zero mass flow problems. From our experience in supporting several customers in creating and simulating models using different libraries, this issue causes large problems. Nevertheless, there is limited literature available on this subject. Dermont et al. (2016) presented an analysis of measures to improve robustness of models used to simulate air condition cycles. The demonstrated the impact of different measures including regularization of the mass flow pressure correlation at low mass flow rates, heat transfer modeling and choice of solver. The study (Li et al. 2020) highlighted the impact of an accurate and fast calculation of the system Jacobian matrix as part of the solution process. Qiao and Laughman (2022) analyzed the impact of different measures to improve performance of air conditioning models at low mass flow rates. They came up with a new regulation scheme for the pressure drop mass flow correlation around zero mass flows and an analysis of heat transfer handling methods.

In our opinion these article only scratch the surface of the numerical reasons which cause the slow simulations at zero mass flow rates. We strongly believe that gaining a deeper understanding of this phenomenon is crucial for developing effective strategies to improve simulation time in scenarios involving low mass flow rates. Therefore, the primary objective of this paper is to elucidate the underlying causes behind the sluggish performance observed during simulations with low mass flow rates.

To achieve this goal, we begin by examining and analyzing the solution process of a highly simplified fluid dynamic model. This analysis serves as a starting point to unveil the root causes of zero mass flow issues. Furthermore, we expand this model to incorporate additional complexities, thereby showcasing the impact of increased intricacy on simulation performance. By employing all these models, we illustrate the application of various countermeasures aimed at mitigating the challenges posed by zero mass flow problems.

2 Analysis for simple model

For demonstration of the underlying phenomenon we use a simple model of an isothermal ideal gas in a volume with the variable pressure p connected over a flow resistance to a boundary with fixed pressure p_b

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \frac{V}{R\cdot T}\frac{\mathrm{d}p}{\mathrm{d}t} = m_\mathrm{flow}\,. \tag{1}$$

When including a quadratic flow model with the parameter c but neglecting the dynamics of the momentum flow as often done for gas flows

$$m_\mathrm{flow} = c\cdot\mathrm{sign}(p_\mathrm{b}-p)\cdot\sqrt{|p_\mathrm{b}-p|} \tag{2}$$

we can derive the single ordinary differential equation

$$\frac{V}{R\cdot T}\cdot\frac{\mathrm{d}p}{\mathrm{d}t} = c\cdot\mathrm{sign}(p_\mathrm{b}-p)\cdot\sqrt{|p_\mathrm{b}-p|} \tag{3}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \frac{1}{\tau}\cdot\mathrm{sign}(p_\mathrm{b}-p)\cdot\sqrt{|p_\mathrm{b}-p|} = \frac{1}{\tau}\cdot F(\Delta p)\,. \tag{4}$$

In order simplify the equation we introduced $\Delta p = p_\mathrm{b} - p$ and $\tau = \frac{V}{R\cdot T\cdot c}$. Equation (4) could be integrated using an implicit Euler scheme

$$\frac{p_{t_i} - p_{t_{i-1}}}{\Delta t} = \frac{1}{\tau} \cdot F(\Delta p_{t_i}) \tag{5}$$

$$R(p_{t_i}) = p_{t_i} - p_{t_{i-1}} - \frac{\Delta t}{\tau} \cdot F(\Delta p_{t_i}) = 0 \tag{6}$$

in which p_{t_i} denotes the solution at the current and $p_{t_{i-1}}$ the solution at the previous time step. Equation (6) is a nonlinear equation which has to be solved to determine the pressure at the current step t_i. We have brought the equation to residual form. So we have to find the root of the residual function $R(p)$. For the solution solvers usually apply a Newton method, in which the linearized equation is repeatedly solved

$$p_{t_i}^j = p_{t_i}^{j-1} - \left(R'\left(p_{t_i}^{j-1}\right)\right)^{-1} \cdot R(p_{t_i}^{j-1}). \tag{7}$$

Here the prime denotes the derivative and $p_{t_i}^j$ is the updated solution for pressure at time step t_i in iteration j. The iteration is performed starting with an initial guess $p_{t_i}^0$ for the solution. Solving the equation requires solving the linear system $R'\left(p_{t_i}^{j-1}\right) \cdot (p_{t_i}^{j-1} - p_{t_i}^j) = R\left(p_{t_i}^{j-1}\right)$. For a single equation this can be done without much effort, but for a system of differential equations, the derivative of the residual equation system becomes a matrix (which is closely related to the Jacobian matrix of the system function F). Calculation of the derivative/Jacobian matrix and calculation of a decomposed form for solution comes at high computational effort. Additionally, in many systems the matrix does not change too much while advancing in time. Therefore, solvers usually never update the derivative during the iterative solution process for a time step and use $(R'(p_{t_i}^j))^{-1} = (R'(p_{t_i}^0))^{-1}$ (known as Chord method (Kelley 1995)). Additionally, solvers try to use the same derivative $(R'(\tilde{p}))^{-1}$ for solving multiple time steps. With this assumption the following iteration scheme is used with a constant $R'(\tilde{p}))^{-1}$

$$p_{t_i}^j = \Phi(p_{t_i}^{j-1}) = p_{t_i}^{j-1} - (R'(\tilde{p}))^{-1} \cdot R(p_{t_i}^{j-1})). \tag{8}$$

Figure 1 visualizes an exemplary successful iteration for the example problem from equation (4). The slope of the linearized approximations (red lines) do not match the actual local slope of the residual function (blue curve). Nevertheless, the iteration schemes converges, but it has lost its quadratic convergence due to the constant derivative. But as shown in Figure 2 the iteration might fail. The iterations schemes drifts off from the actual root of the residual equation and finally circles around the solution. In the following we will analyze this phenomenon in more detail and demonstrate that the iteration with not-up-to-date derivative becomes more difficult to solve when mass flow rate becomes small. If the solution diverges, as shown in Figure 2 or if due to the slower convergence no solution is found within a given number of iterations the step will be rejected. As reaction the solver will request an update of

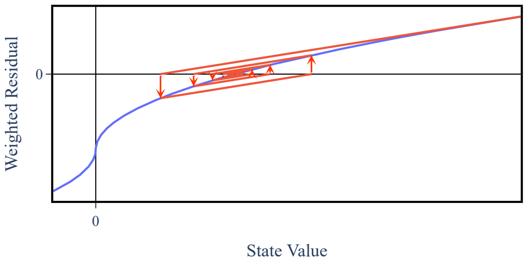

Figure 1. Example of a converging iterative solution for the example problem.

the Jacobian matrix and repeat the solution process. If the solution still fails, it might be necessary (as we will show later on) to reduce the step size. Then a solution with the outlined method can only be achieved for very small time steps and if the Jacobian matrix is updated at every time step. In this setting the time step is not chosen to fulfill the chosen tolerance anymore, but to make the system solvable with the approximated Jacobian matrix from the initial guess. This slows down the solution process and is the cause of slow models performance what is named zero mass flow issues.

Let us take a closer look on the iterative solution process of Equation 4. The Banach fixed-point theorem states that an iteration scheme $\Phi(p_{t_i}^j)$ will converge if it is contractive. In that case a contraction factor $\lambda < 1$ exists and the iterative schemes fulfills

$$\left|\Phi(p_{t_i}^j) - \Phi(p_{t_i}^{j-1})\right| \leq \lambda \cdot \left|p_{t_i}^j - p_{t_i}^{j-1}\right|. \tag{9}$$

For the Chord method we can calculate the contraction factor with

$$\lambda = \frac{\left|\Phi(p_{t_i}^j) - \Phi(p_{t_i}^{j-1})\right|}{\left|p_{t_i}^j - p_{t_i}^{j-1}\right|} = \left|\frac{\Phi(p_{t_i}^j) - \Phi(p_{t_i}^{j-1})}{p_{t_i}^j - p_{t_i}^{j-1}}\right|. \tag{10}$$

By applying the mean value theorem there exists a $p_\xi \in (p_{t_i}^{j-1}, p_{t_i}^j)$ such that

$$\lambda = |\Phi'(p_\xi)| = |1 - R'(\tilde{p}))^{-1} \cdot R'(p_\xi)|. \tag{11}$$

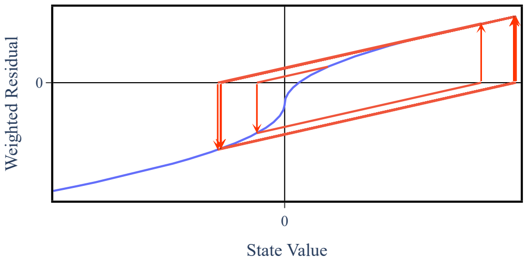

Figure 2. Example of a failing iterative solution attempt for the example problem.

For the second step we included Equation 8. As convergence requires $\lambda < 1$ we can obtain with Equation 11

$$0 < R'(\tilde{p}))^{-1} \cdot R'(p_\xi) < 2\,. \tag{12}$$

This criterion can be used for a more detailed analysis of the cause of zero mass flow issues and how it can be avoided, or its effects can be reduced.

But before applying this criterion to our demonstration problem Equation 1, we want to take a general look on the significance of Equation 12. For the reasons mentioned above a solver will try to reuse the derivative $R'(\tilde{p}))$ for multiple steps. But the actual derivative in the region of the solution $R'(p_\xi)$ might differ from the used derivative $R'(\tilde{p}))$ by a factor of two as demonstrated in Figure 2. In that case the iterative solution of the nonlinear system will fail. The solver will reject the step and repeat the solution with a reduced step size and an updated value for $R'(\tilde{p}))$. Unfortunately, the actual criterion for convergence is even stricter than that from Equation 12. If the condition is fulfilled, but with values close to two, the convergence is very slow. The integrator demand convergence within a few iteration, e.g. Hindmarsh et al. (2023), otherwise the step is rejected and the simulation performance decreases. But as the ratio $|R'(\tilde{p}))^{-1} \cdot R(p^j_{t_i})|$ represent somehow the convergence speed of the method, we can use it for an analysis. Smaller values for $|R'(\tilde{p}))^{-1} \cdot R(p^j_{t_i})|$ will lead to convergence within fewer steps. As convergence within few steps is demanded, a good initial guess is of great importance. The closer this initial guess is to the actual solution, the more likely is convergence of the method to the actual solution within the demanded solution tolerance. This has three consequences:

1. Methods which have a good approximation or forecast used for the initial guess will not be affected as much by the zero mass flow issue like method with no good forecast.

2. Reducing the time step improves the approximation of the initial guess. Therefore, reducing the step size has two positive aspects: it improves the initial guess and the ratio $|R'(p^0_{t_i}))^{-1} \cdot R(p^j_{t_i})|$ will be closer to one. But obviously this comes as the price of a slower simulation.

3. The stricter the tolerance demanded for the solution, the harder it becomes to reach the tolerance within the desired number of iteration. So decreasing the tolerance might cause more problems at low mass flow rates.

After that general analysis, we will take a closer look on that criterion for our demonstration problem.

2.1 Analysis of Convergence for Demonstration Problem

If we apply Equation 12 to our simple problem in Equation 1, we get

$$\frac{1+\frac{\Delta t}{2\tau\sqrt{p_\mathrm{b}-p_\xi}}}{R'(\tilde{p})} < 2\,. \tag{13}$$

First of all, one can see that as $p \rightarrow p_\mathrm{b}$ results in $p \rightarrow p_\xi$ the method has only a chance to convergence for $\Delta t \rightarrow 0$. Therefore, in almost all libraries the square-root in the mass flow pressure relation is replaced by an approximation which has a finite derivative when crossing zero. For the regulation named regRoot from the Modelica Standard Library (*Modelica Standard Library 4.0.0* 2023) eq. 13 becomes

$$\frac{1+\frac{\Delta t}{\tau}\frac{0.5\cdot(p_\mathrm{b}-p_\xi)^2+\Delta p_\mathrm{small}{}^2}{((p_\mathrm{b}-p_\xi)^2+\Delta p_\mathrm{small}{}^2)^{1.25}}}{R'(\tilde{p})} < 2\,. \tag{14}$$

But still convergence for $p \rightarrow p_\mathrm{b}$ can become problematic for small values of τ and if Δp_small is not chosen large enough.The solver will find a solution within the demanded number of iteration but only for small time steps. Additionally, the simple demonstration model can be used to explain some general phenomena which can be seen in more complex models concerning if zero mass flow issues occur or not. Therefore, the simple model has been implemented as Modelica model and was solved with Dymola 2022x (Dassault Systemes AB 2023) with the regulated form of the square root mentioned above. Two different solvers from Dymola were used: Dassl and Radau. Dassl as a general purpose solver with good performance and Radau, as it was recommend by Dermont et al. (2016), when encountering zero mass flow problems. Figure 3 shows the number of Jacobian matrix evaluation when solving the problem while varying the pressure level of the boundary p_b. The pressure was initialized at the same level as the boundary. After one second the pressure in the boundary was increased or decreased by 10 Pa and the system was simulated for further 10 s to settle. One can see that after exceeding a certain threshold in pressure

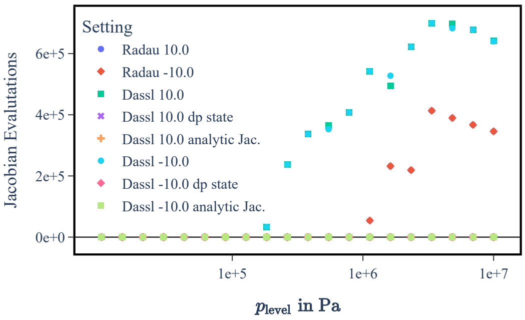

Figure 3. Number of Jacobian matrix evaluations for the example problem for different settings.

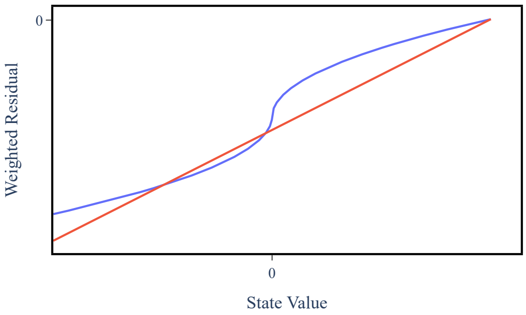

Figure 4. Demonstration of the approximate calculation of the derivative as secant (red) to the residual function (blue). If the two supporting points have different signs, the slope of the derivative approximation is much low than the slope in the region around $\dot{m} = 0$.

level the number of Jacobian matrix evaluation dramatically increases. This indicates presence of a zero mass flow problem as the integration steps are rejected, and a new Jacobian matrix must be calculated. This pressure level threshold depends on the chosen solver. Radau generally performs better in this case. But in general, it is surprising that the pressure level has an impact on the solution of the problem. The pressure level of p_b is only a constant offset for the solution and therefore on first look it should not have an impact. Additionally, if Dassl is chosen as solver, the number of Jacobian matrix evaluations depends on the sign of the pressure step. When Radau is used, the same number of Jacobian matrix evaluation is required independent of the sign of the pressure step.

Both phenomena can be explained if one focuses on how the derivative $R'(\tilde{p})$ is calculated in the solution process. If no special settings are applied the derivative is calculated numerically presumably with a given relative variation $\Delta p_\varepsilon = \varepsilon \cdot \tilde{p}$ with a small number ε. Typical methods are

$$R'(\tilde{p}) \approx \frac{R(\tilde{p}+\Delta p_\varepsilon) - R(\tilde{p})}{\Delta p_\varepsilon} \tag{15}$$

or

$$R'(\tilde{p}) \approx \frac{R(\tilde{p}+\Delta p_\varepsilon) - R(\tilde{p}-\Delta p_\varepsilon)}{2 \cdot \Delta p_\varepsilon} . \tag{16}$$

Firstly, if the pressure level increases, the absolute variation for calculation of the derivative increases. But the physical meaning of a changed pressure difference has not changed. For a solution around the steady solution $\tilde{p} \approx p_\mathrm{b}$ the variation Δp_ε used to approximate the derivative might flip the sign of $\tilde{p} - p_\mathrm{b}$. In that case the absolute value of approximation of $R'(\tilde{p})$ becomes smaller than the absolute value of the actual derivative (see Figure 4). As Equation 13 shows this decreases the speed of convergence. Secondly, the problem itself is symmetric to an increasing or decreasing step of the pressure. But using the approximation Equation 15 to calculate the derivative introduces an asymmetry. As only the Dassl solver is susceptible to the sign of the variation it suggests itself that Dassl uses a forward differencing scheme while Radau uses a central difference scheme like Equation 16. We created an external external function which is called in the model to track the value of states during all models calls: by analyzing the state variation one can identify the times at which the Jacobian matrix is updated. When using Radau as solver the perturbation is applied in both directions while for Dassl only a perturbation in one direction is applied - therefore we can verify the assumption. The central scheme is is more accurate and symmetric, and therefore the results to not show a dependence on the sign of the pressure step.

There are two options to improve model performance for zero mass flow rate. If possible one could use an analytic calculation of the Jacobian matrix to get rid of the underestimation of the derivative. Additionally, or solely the problem could be reformulated to use $\Delta p = p - p_\mathrm{b}$ as state. This measure improves the accuracy of the numeric approximation of the Jacobian matrix. Pressure differences are driving the mass flow rates and therefore the changes in pressure. If the pressure becomes large, the perturbation applied $\varepsilon \cdot \Delta p$ to calculated the derivative might be in the range of the driving pressure differences and the approximate Jacobian matrix is not accurately enough to solve the system with large steps. Choosing the pressure difference as state, leads to a perturbation much smaller used to calculate the derivative and the applied solution method is much more robust. In both cases the model performs much better as one can see in Figure 3. If one of the tweaks is applied, no issues occur even if pressure level is increased. Additionally, the number of evaluations is independent of the step sign.

2.2 Non-linearity as root of the problem

Furthermore, we want to use the simple model to emphasize the importance of the non-linearity of the problem. Dermont et al. (2016) gave the time constant of the problem which increases when decreasing the mass flow rate as reason for the poor model performance for small mass flow rates. We want to demonstrate that the time constant itself is not the root of the problem. For this test we run the model with Radau as solver and the pressure level of 1×10^6 Pa with a step of 10 Pa. For the second run the regRoot mass flow pressure relation was replaced with a linear relation which has the same slope for vanishing mass flow rates (Figure 5). Due to the linearity this slope remains the same for all mass flow rates. The slope of the mass flow pressure relation determines the time constant of the problem. The comparison is designed in a way that the time constant of the non-linear problem is at maximum the same as the time constant of the linear problem. Figure 6 shows the solver step when both models are simulated with the same settings. Though the linear problem should be tougher to solver from a simple perspective of the time constant, it is actually the other way round. A linear model cannot face zero mass flow

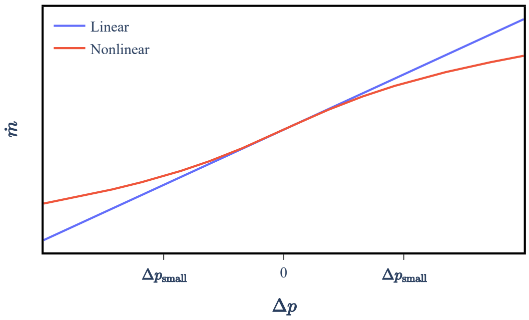

Figure 5. Linearized and actual non-linear mass flow pressure relation used in the example model to demonstrate the impact of the non-linearity.

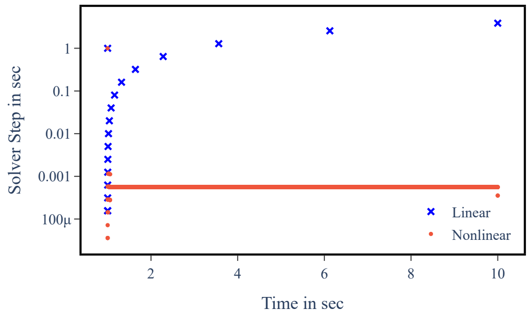

Figure 6. Solver step size for example model with linear and non-linear mass flow pressure relation as shown in Figure 5.

issues as the Jacobian matrix calculated at any position is valid for the whole domain. As a result the linear problem can be solved with much higher time steps, though it actually has the higher dynamics. Actually, the presented findings could have been derived from Equation 12. But the results are presented as they impressively show which solver steps are possible coming from the pure dynamic of the problem. The non-linearity of the problem which results in a change of the time constant, causes the Chord method to fail and as a result the possible time steps of the integration method cannot be exploited.

3 Extended model with transported scalar

Obviously, models are normally much more complex. One additional level of complexity are additional equations to be solved e.g. balances for balance, mass, composition etc. The form of these equations has a crucial impact on the resilience of the model against zero mass flow issues. In order to demonstrate and analyze this impact we extend our simple models with an additional equation

$$\frac{\mathrm{d}\psi}{\mathrm{d}t} = \begin{cases} \frac{m_{\text{flow}}}{m} \cdot (\psi_{\text{b}} - \psi), & \text{if } m_{\text{flow}} \geq 0 \\ 0, & \text{otherwise} \end{cases} . \tag{17}$$

Now we introduce the abbreviation $\Delta\psi = \psi_{\text{b}} - \psi$ and observe that $\text{sign}(m_{\text{flow}}) = \text{sign}(\Delta p)$ in order to use the Heaviside step function $\text{H}(\Delta p)$. Moreover from Equation 4 and Equation 2 we can replace $m_{\text{flow}} = c \cdot F(\Delta p)$ and $m = \frac{V}{R \cdot T} \cdot p$. Then

$$\frac{\mathrm{d}\psi}{\mathrm{d}t} = c^2 \tau \cdot \frac{G(\Delta p)}{p} \cdot \Delta\psi \tag{18}$$

where we have introduced $G(\Delta p) = \text{H}(\Delta p) \cdot F(\Delta p))$ and again use $\tau = \frac{V}{R \cdot T \cdot c}$.

The residual equation becomes the form

$$R(\psi_t) = \psi_t - \psi_{t-1} - \Delta t \cdot c^2 \tau \cdot \frac{G(\Delta p)}{p_t} \cdot \Delta\psi_t \tag{19}$$

The solution of equation $R(\psi_t) = 0$ depends on $G(\Delta p) = \text{H}(\Delta p) \cdot F(\Delta p)$ and hence on the flow direction through $\text{H}(\Delta p)$ and on the actual mass flow rate through $F(\Delta p)$. It is given by

$$\psi_t = \frac{\psi_{t-1} + \Delta t \cdot c^2 \tau \cdot \frac{G(\Delta p_t)}{p_t} \psi_{\text{b}}}{1 + \Delta t \cdot c^2 \tau \cdot \frac{G(\Delta p_t)}{p_t}} \tag{20}$$

Though in our model it would be possible to solve the equation for pressure and the equation for the passive scalar sequentially, in a Modelica tool these equations are solved simultaneously. So Equation 17 is added to the model from the previous section and solved again while tracking the number of Jacobian matrix evaluations as it was plotted in Figure 3. This time only Radau and a step of 10 Pa was used. The results are show in Figure 7. For reference purposes the results for only the pressure equations are shown additionally. Adding the equation for the transported scalar causes the zero mass flow issue to occur at lower pressure levels and from that pressure level additional Jacobian matrix evaluations are caused compared to the reference case. When integrating the solution of Equation 17 depends on the actual mass flow rate. Therefore, the iteration can only converge after the Equation 1 has converged reasonably close to the actual solution. If Equation 1 alone converges only just within the desired number of iteration, the system will not converge, causing the zero mass flow issue occurring at lower pressure levels. Figure 8 shows an exemplary course of the residuals

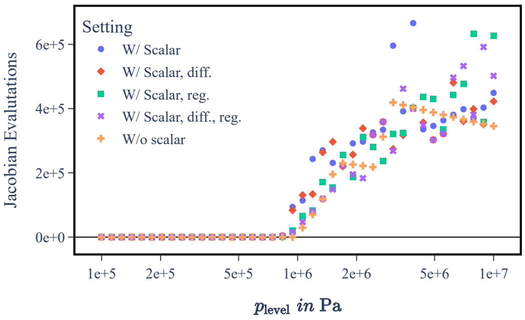

Figure 7. Number of Jacobian matrix evaluations for example model with (w/) and without (w/o) transported scalar and different settings.

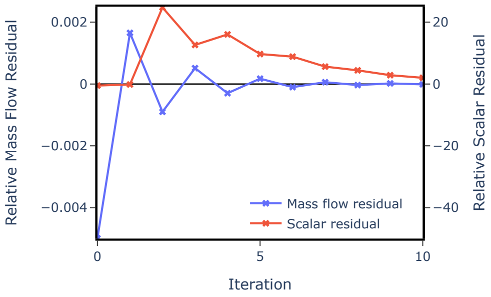

Figure 8. Residuals of pressure and transported scalar during iterative solution.

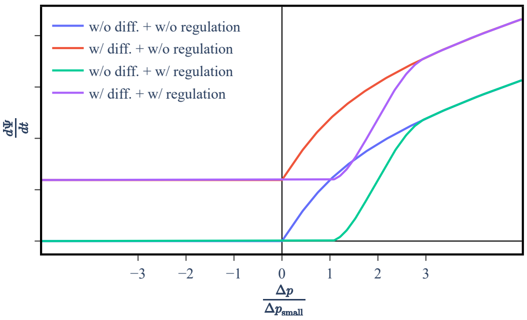

Figure 9. Transported scalar flow rate for different modeling approaches: all combinations of with (w/) and without (w/o) diffusion and convection with (w/) and without (w/o) regulation.

of the two equations over the iterations. One can clearly see that the residual of the scalar (Equation 17) even increases before it starts to converge. But convergence rate is slow, while the mass flow rates oscillates around its final value. Figure 7 contains measure which are intended to improve convergence around zero mass flow rate. One idea is to introduce diffusion into Equation 17 by adding a transport term which is independent of the mass flow rate

$$\frac{d\psi}{dt} = \left(c^2\tau \cdot \frac{G(\Delta p)}{p} + \mu\right) \cdot \Delta\psi \tag{21}$$

We use a artificial diffusion constant μ.

The second idea is to introduce a (nonphysical) regulation of the convective transport as depicted in Figure 9. The convective transport with is set to be zero if the mass flow is regulated. After the mass flow has left the regulation the convective flow is ramp up to its actual value.

From Figure 7 we can see that the pure diffusion does not improve the system convergence around zero. No improvement was observed for reasonable big values of the artificial diffusion constant μ. The problem is that the solution for ψ still depends on the mass flow rate, though it is shifted. Nevertheless, the scalar converges only after the mass flow has converged. For an improvement the solution should be independent of the mass flow rate for very small mass flow rates. That is the aim of the second measure, with the additional regulation of the convective flow. This method has to be implemented carefully to avoid a violation of conservation. When this measure is applied the system is more robust against zero mass flow issues, but it still performs worse than the model with only mass conservation.

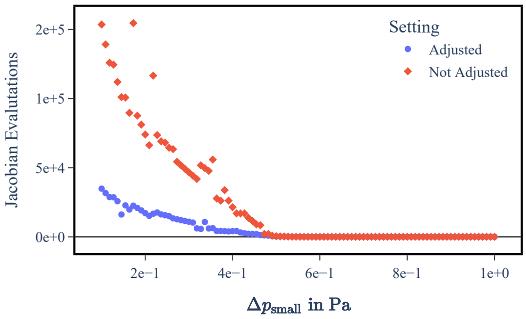

Figure 10. Number of Jacobian matrix evaluations for a multi-volume model with and without adjusted linearization interval.

4 Extended model with multiple volumes

In this step we want to extend the model from section 2 to a model with multiple connected control volumes.

4.1 Pressures

For the pressures we obtain

$$\frac{dp[k]}{dt} = \frac{1}{\tau[k]} \cdot \Big(F(\Delta p[k]) - F(\Delta p[k+1])\Big) \tag{22}$$

where we have extended the notation from Equation 4 to the control volume label $k = 1 \ldots N$ with $\Delta p[1] = p_b - p[1]$, $\Delta p[k] = p[k-1] - p[k]$, $\Delta p[N+1] = 0$ and control volume dependent time constant $\tau[k] = \frac{V[k]}{R \cdot T \cdot c}$.

Following the procedure in Equation 14, we now approximate

$$\begin{aligned} F(\Delta p[k]) &= \operatorname{sign}(\Delta p[k]) \cdot \sqrt{|\Delta p[k]|} \\ &\approx \operatorname{regRoot}(\Delta p[k], \Delta p_{\text{small}}[k]) \end{aligned} \tag{23}$$

In a numerical experiment with $n = 9$ the control volume $V[k]$ was chosen uniformly $V[k] = V$ except for volume five, in which it was chosen to be a tenth of the other values $V[5] = 0.1 \cdot V$, such that $\tau[5] = \tau/10$. The other parameter varied is $\Delta p_{\text{small}}[k]$. Two strategies are applied: in the first strategy all regularization intervals have the same size and in the second strategy all have the same size except for that of $k = 5$. The value is chosen to be $\Delta p_{\text{small}}[5] = 10 \cdot \Delta p_{\text{small}}$. The results are given in Figure 10. It is important to mention that the given level of Δp_{small} corresponds to the value applied for most of the equations. The first important thing which one can see is that increasing the regularization interval will improve the performance until a certain threshold. After that one could say that the zero mass flow problem is not present anymore. But the increase in Δp_{small} comes at a certain price, the pressure drop at low mass flow rates will be computed incorrectly. If this is an issue, the second strategy might be an interesting option. It is only required to apply a larger linearization for flow models which are connected to pressure state with the large values of τ. So the overall accuracy of the model will be better.

4.2 Passive Scalars

Using the same notation as in the previous section we can extend Equation 18 for the passive scalar to obtain

$$\begin{aligned}\frac{\mathrm{d}\psi[k]}{\mathrm{d}t} &= \left(c^2\tau[k]\cdot\frac{G(\Delta p[k])}{p[k]}+\mu\right)\cdot\Delta\psi[k] \\ &\quad -\left(c^2\tau[k]\cdot\frac{\widetilde{G}(\Delta p[k+1])}{p[k]}+\mu\right)\cdot\Delta\psi[k+1]\end{aligned} \tag{24}$$

with $\Delta\psi[1]=\psi_b-\psi[1]$ and $\Delta\psi[k]=\psi[k-1]-\psi[k]$ and $\widetilde{G}(\Delta p)=G(-\Delta p)=\big(H(\Delta p)-1\big)\cdot F(\Delta p)$, where we have used the symmetry properties of the Heaviside step function H and the function F due to Equation 4. Finally we approximate

$$\mathrm{H}(\Delta p)\approx \mathrm{SM}(\Delta p,\mathrm{func},\mathrm{nofunc}) \tag{25}$$

with the `stepsmoother` function SM(.) from `Modelica.Fluid.Dissipation.Utilities...` as regularization for the Heaviside step function. Here the difference $|\mathrm{func}-\mathrm{nofunc}|$ denotes the width of the rgulated step.

4.3 Generalized convergence criterium

In this section we would like to extend the computation of λ in Equation 11 and the resulting convergence criterium in Equation 12 to general state space systems with N states $(x[1],\cdots,x[N])=:\vec{x}$. For a given time step t_i we have

$$\frac{d}{dt}\vec{x}_{t_i}=\vec{f}(\vec{x}_{t_i}) \tag{26}$$

and for an implicit integration algorithm it holds that

$$\vec{x}_{t_i}=\vec{x}_{t_{i-1}}+\Delta t\cdot\vec{f}(\vec{x}_{t_i}) \tag{27}$$

with the non linear residuum equation

$$\vec{R}(\vec{x}_{t_i})\overset{!}{=}\vec{0}=\vec{x}_{t_i}-\vec{x}_{t_{i-1}}-\Delta t\cdot\vec{f}(\vec{x}_{t_i}) \tag{28}$$

which can be solved iteratively by a Newton-Raphson scheme

$$\vec{x}^{\,j+1}_{t_i}=\vec{\Phi}(\vec{x}^{\,j}_{t_i})=\vec{x}^{\,j}_{t_i}-\left[\mathrm{J}_{\vec{R}}(\vec{x}^{\,j}_{t_i})\right]^{-1}\vec{R}(\vec{x}^{\,j}_{t_i})\,. \tag{29}$$

Here $\mathrm{J}_{\vec{R}}$ denotes the Jacobian of the residuum vector $\vec{R}(\vec{x}^{\,j}_{t_i})$. In analogy to Equation 8 the Chord method attempts to solve Equation 29 with the inverse Jacobian fixed at some state $\widetilde{\vec{x}}$:

$$\vec{x}^{\,j+1}_{t_i}=\vec{\Phi}(\vec{x}^{\,j}_{t_i})=\vec{x}^{\,j}_{t_i}-\left[\mathrm{J}_{\vec{R}}(\widetilde{\vec{x}})\right]^{-1}\vec{R}(\vec{x}^{\,j}_{t_i})\,. \tag{30}$$

Now in analogy to Equation 9 this iteration converges if it is contractive, that is for some $\lambda<1$ it holds that

$$\left\|\vec{\Phi}(\vec{x}^{\,j}_{t_i})-\vec{\Phi}(\vec{x}^{\,j-1}_{t_i})\right\|\le\lambda\cdot\left\|\vec{x}^{\,j}_{t_i}-\vec{x}^{\,j-1}_{t_i}\right\| \tag{31}$$

In order to further develop this expression, notice that we re-write the difference on the left as the result of an integration along a straight line $\vec{x}(s)=\vec{x}^{\,j-1}_{t_i}+s\cdot(\vec{x}^{\,j}_{t_i}-\vec{x}^{\,j-1}_{t_i})$ in state space with curve parameter $s\in[0,1]$:

$$\vec{\Phi}(\vec{x}^{\,j}_{t_i})-\vec{\Phi}(\vec{x}^{\,j-1}_{t_i})=\int\limits_0^1 ds\left\{\mathrm{J}_{\vec{\Phi}}\big(\vec{x}(s)\big)\cdot\frac{\mathrm{d}\vec{x}(s)}{\mathrm{d}s}\right\}\quad, \tag{32}$$

where $\mathrm{J}_{\vec{\Phi}}\big(\vec{x}(s)\big)$ denotes the Jacobian of $\vec{\Phi}$ at position $\vec{x}(s)$. Now clearly the tangent vector $\mathrm{d}\vec{x}(s)/\mathrm{d}s$ is constant along the line and given by

$$\frac{\mathrm{d}\vec{x}(s)}{\mathrm{d}s}=\vec{x}^{\,j}_{t_i}-\vec{x}^{\,j-1}_{t_i}\quad. \tag{33}$$

Moreover from Equation 30 it follows that

$$\mathrm{J}_{\vec{\Phi}}\big(\vec{x}(s)\big)=\mathbb{1}-\left[\mathrm{J}_{\vec{R}}(\widetilde{\vec{x}})\right]^{-1}\mathrm{J}_{\vec{R}}\big(\vec{x}(s)\big) \tag{34}$$

with $\mathbb{1}$ denoting the N-dimensional identity matrix. So we can formally re-write Equation 32 as

$$\vec{\Phi}(\vec{x}^{\,j}_{t_i})-\vec{\Phi}(\vec{x}^{\,j-1}_{t_i})=A\cdot\left(\vec{x}^{\,j}_{t_i}-\vec{x}^{\,j-1}_{t_i}\right) \tag{35}$$

with the matrix A given by

$$A=\int\limits_0^1 ds\left\{\mathbb{1}-\left[\mathrm{J}_{\vec{R}}(\widetilde{\vec{x}})\right]^{-1}\mathrm{J}_{\vec{R}}\big(\vec{x}(s)\big)\right\} \tag{36}$$

With these ingredients we can replace Equation 31 similarly to Equation 11 by

$$\|A\|\le\lambda<1 \tag{37}$$

Here $\|A\|=\|A\|_2$ denotes the spectral norm of the matrix A as induced from the L^2 vector norm $\|\vec{x}\|$. The spectral norm is defined as the square root of the largest eigenvalue of A^TA, with A^T the conjugate transpose of A. We can further estimate $\|A\|$ as follows

$$\begin{aligned}\|A\| &= \left\|\int_0^1 ds\left\{\mathbb{1}-\left[\mathrm{J}_{\vec{R}}(\widetilde{\vec{x}})\right]^{-1}\mathrm{J}_{\vec{R}}\big(\vec{x}(s)\big)\right\}\right\| \\ &\le \max_{s\in[0,1]}\left\|\mathbb{1}-\left[\mathrm{J}_{\vec{R}}(\widetilde{\vec{x}})\right]^{-1}\mathrm{J}_{\vec{R}}\big(\vec{x}(s)\big)\right\| \\ &\le \left\|\left[\mathrm{J}_{\vec{R}}(\widetilde{\vec{x}})\right]^{-1}\right\|\cdot\max_{s\in[0,1]}\left\|\mathrm{J}_{\vec{R}}(\widetilde{\vec{x}})-\mathrm{J}_{\vec{R}}\big(\vec{x}(s)\big)\right\| \\ &\overset{!}{<} 1\end{aligned} \tag{38}$$

Although Equation 38 holds for any matrix norm, for practical application in the context of Equation 31 one has to use the spectral norm $\|\mathrm{J}\|_2$.

4.4 Explicit application

The bound obtained in Equation 38 can be used in order to give conditions for convergence and also recommendations for the regularization parameters of the regRoot and SM functions contained in $F(\Delta p)$ and $G(\Delta p)$.

4.4.1 General case

The matrix elements $\mathrm{J}_{\vec{R}}(\vec{x})[I,J]$of the Jacobian $\mathrm{J}_{\vec{R}}(\vec{x})$ can be computed from Equation 28 as follows:

$$\mathrm{J}_{\vec{R}}(\vec{x})[I,J] = \frac{\partial R[I](\vec{x})}{\partial x[J]} = \delta[I,J] - \Delta t \cdot \frac{\partial f[I](\vec{x})}{\partial x[J]} \tag{39}$$

Here $\delta[I,J]$ denotes the Kronecker delta. For the coupled pressure scalar system we have the time derivatives $f[I]$ given from Equation 22 and Equation 24. Let $(x[1],\ldots,x[N]) = (p[1],\ldots,p[N])$ and $(x[N+1],\ldots,x[2N]) = (\psi[1],\ldots,\psi[N])$. Then

$$\begin{aligned}
&\boxed{1 \leq I \leq N} \quad f[I] = f\big(x[I-1],x[I],x[I+1]\big)\\
&\boxed{N < K \leq 2N} \quad f[K] = f\big(x[K-1],x[K],x[K+1],\\
&\qquad\qquad x[K-N-1],x[K-N],x[K-N+1]\big)
\end{aligned}$$

Hence the resulting incidence matrix is tri-diagonal for the pressures and tri-diagonal for the scalars with an additional tri-band between pressures and scalars. This property may be used in order to arrive at an estimate to the spectral norms of Equation 38. An explicit computation of the eigenvalues can in principle be avoided by approximating the spectral norm by the general property:

$$\|A\|_2 \leq \sqrt{\|A\|_1 \cdot \|A\|_\infty} \tag{40}$$

where $\|A\|_1 = \max_j \sum_{i=1}^N |a_{ij}|$ is the maximum absolute column sum norm of A and $\|A\|_\infty = \max_i \sum_{j=1}^N |a_{ij}|$ is the the maximum absolute row sum norm of A. This also avoids computation of $A^T A$. An explicit symbolic analysis for the pressure system in the fashion of section 7 in J. Brunnemann (2008) will be subject to future work.

4.4.2 Single pressure

For the case of a single pressure $p[1]$ and boundary pressure p_b the Jacobian $\mathrm{J}_{\vec{R}}(\vec{x})$ has only one single element:

$$\mathrm{J}_{\vec{R}}(\vec{x})[1,1] \quad = \quad \frac{\partial R[1](\vec{x})}{\partial x[1]} =: R'(p) \tag{41}$$

For simplicity of notation we have dropped the discretization index on the right hand side. Plugging this into Equation 38 we obtain

$$\max_{s\in[0,1]} \left\|1 - [R'(\widetilde{p})]^{-1} \cdot R'(p(s))\right\| < 1c \tag{42}$$

In order to fulfill this inequality it must hold that

$$0 \leq \frac{R'\big(p(s)\big)}{R'(\widetilde{p})} \leq 2 \quad \forall s \in [0,1] \tag{43}$$

This re-produces the result of Equation 12.

4.4.3 Single pressure and passive scalar

For the case of a single pressure $p[1]$ with boundary pressure p_b acoupled to a single passive scalar $\psi[1]$ with boundary ψ_b the Jacobian $\mathrm{J}_{\vec{R}}(\vec{x})$ is a $2x2$ matrix with three non-zero elements:

$$\begin{aligned}
\mathrm{J}_{\vec{R}}(\vec{x})[1,1] &= \frac{\partial R[1](\vec{x})}{\partial x[1]} = \frac{\partial R[1](p)}{\partial p} =: a\\
\mathrm{J}_{\vec{R}}(\vec{x})[2,1] &= \frac{\partial R[2](\vec{x})}{\partial x[1]} = \frac{\partial R[2](p,\psi)}{\partial p} =: c\\
\mathrm{J}_{\vec{R}}(\vec{x})[2,2] &= \frac{\partial R[2](\vec{x})}{\partial x[2]} = \frac{\partial R[2](p,\psi)}{\partial \psi} =: d
\end{aligned}$$

Here we have left out the discretization indices for p, ψ on the left hand side for simplicity of notation.

$$\mathrm{J}_{\vec{R}}\big(p(s),\psi(s)\big) = \begin{pmatrix} a & 0 \\ c & d \end{pmatrix} \text{ and } \mathrm{J}_{\vec{R}}\big(\widetilde{p},\widetilde{\psi}\big) = \begin{pmatrix} a_0 & 0 \\ c_0 & d_0 \end{pmatrix}$$

In the sequel we will suppress the (s)-dependence of (a,c,d) for simplicity of notation. The above setting implies

$$\begin{aligned}
M &:= \mathbb{1} - \big[\mathrm{J}_{\vec{R}}(\widetilde{\vec{x}})\big]^{-1}\mathrm{J}_{\vec{R}}\big(\vec{x}(s)\big)\\
&= \begin{pmatrix} m_{11} & 0 \\ m_{21} & m_{22} \end{pmatrix} = \begin{pmatrix} -\frac{a}{a_0}+1 & 0 \\ -\frac{c}{d_0}+\frac{ac_0}{a_0 d_0} & -\frac{d}{d_0}+1 \end{pmatrix}
\end{aligned}$$

And finally

$$M^T M = \begin{pmatrix} {m_{11}}^2 & m_{11}m_{21} \\ m_{11}m_{21} & {m_{21}}^2 + {m_{22}}^2 \end{pmatrix} \tag{44}$$

From the symmetry of $M^T M$ it follows that the two eigenvalues λ_1, λ_2 are real. For the 2D case we can explicitly compute them. However one may also apply Gershgorins circle theorem (Gershgorin (1931)) $|\lambda_1 - {m_{11}}^2| \leq |m_{11}m_{21}|$ and $|\lambda_2 - {m_{21}}^2 - {m_{22}}^2| \leq |m_{11}m_{21}|$ for an upper bound of the eigenvalues:

$$\|A\|_2 \leq \sqrt{\max_{s\in[0,1]} \big(|\lambda_1(s)|, |\lambda_2(s)|\big)} \overset{!}{<} 1 \tag{45}$$

This is of particular use for higher dimensional cases of Equation 38.

5 Solver Modification

The adaptions shown which should lead to an improved simulation of the model where all on the model side. These modification can be applied by the simulation engineer or library developer and they work independently of the used tool. But these modifications cause deviations between model results and the physical correct or the expected results. These deviations might be tolerable in many cases, but in some cases they are not. Therefore, it would be interesting to have a solution process which completely avoids zero mass flow issues or reduces the impact.

Table 1. Performance key figures for the solution process for the presented models with and without damping strategy.

	Jacobian Evaluations		Function Calls		Integrator Steps	
	Damped	Normal	Damped	Normal	Damped	Normal
Simple Mass Flow (section 2)	7.1×10^1	2.0×10^4	7.0×10^3	1.4×10^5	2.1×10^3	3.4×10^4
Mass Flow + Scalar (section 3)	5.3×10^2	6.7×10^3	3.0×10^4	7.7×10^5	1.0×10^4	3.0×10^4
Multi volume (section 4)	2.2×10^1	1.1×10^4	2.7×10^3	1.3×10^5	1.1×10^3	4.6×10^4
BranchingDynamicPipes	7.3×10^1	7.1×10^1	4.3×10^3	4.3×10^3	3.4×10^3	3.4×10^3
PID	1.9×10^1	1.9×10^1	9.5×10^2	9.5×10^2	8.4×10^2	8.4×10^2
BatchPlant_StandardWater	1.7×10^2	1.9×10^2	6.8×10^3	6.7×10^3	5.1×10^3	5.2×10^3

Overshooting during the iteration process is a known problem even for normal Newton methods, if the starting point is far off from the actual solution and/or if the equation to solve is highly nonlinear. In that case damping of the solution can reduce the required number of iteration or even avoid divergence of the method. As shown above overshooting is as well the problem which causes the zero mass flow issue. Therefore, we tried to apply a damping strategy to the solution process. For this test we used OpenModelica and the Cvode solver as for both the full code is available and can easily be modified. The algorithm is taken from Dahmen and Reusken (2008) and modified to be usable for the Chord method. For a general residual equation

$$R(x_{t_i}) = 0 \tag{46}$$

a Chord step is done to calculate an update the solution for all states

$$\Delta x_{t_i}^j = -(R'(\tilde{x}))^{-1} \cdot R(x_{t_i}^{j-1})\,. \tag{47}$$

The update is not applied directly. Instead the condition

$$\begin{aligned} &||(R'(\tilde{x}))^{-1} \cdot R\left(x_{t_i}^{j-1} + \lambda \cdot \Delta x_{t_i}^j\right)|| \\ &\quad \leq C_\lambda \cdot ||(R'(\tilde{x}))^{-1} \cdot R(x_{t_i}^{j-1})||\,, \end{aligned} \tag{48}$$

is checked until it is fulfilled with the series $\lambda = 1, 0.5, 0.25, ...$ using $C_\lambda = 1 - \frac{\lambda}{4}$. When the condition is fulfilled the step is applied and the same procedure is repeated for the next step $\Delta x_{t_i}^{j+1}$. The right hand side of Equation 48 is (a fraction) of the norm of the full step calculated from Equation 47. The left hand side is the next full step which is calculated if the current damped step is applied. Therefore, by using this method we damp the calculated step until the norm of the next step is smaller than the current. As the size of the step depends of the residual $R(x_{t_i}^j))$, we enforce the residual to reduce. Checking the condition comes with no relevant extra computational effort. The price of a recalculation after a rejected Chord update costs the same as a normal step. So if the rejected steps are included in the total number of Chord updates, one could design a method which comes at almost no extra costs.

The method was applied to the example problems described above in configuration in which the zero mass flow issue occurs. The performance of the damped strategy is compared to the normal algorithm. The results are summarized in Table 1. Damping the steps causes a significant reduction in Jacobian matrix evaluations and functions calls. The number of integrator time steps is reduced as well, though in case of the problem from section 3 the reduction of integrator steps is not a high as in the other. But function calls and the decomposition of the Jacobian matrix are the main expenses during integration. Therefore, using the damped solving approach significantly reduces the solution time in all the cases. Beside the some models from the Modelica Standard Library were simulated with both methods, though zero mass flow rate was not a particular issues for these models. The performance indicator numbers are very similar. The slight difference can result from differences in the solution and the fact that with the used settings damping steps were not counted in the total step count for an iteration. Therefore, the number of function with damping exceed the number of calls without damping though less steps were taken and less Jacobian updates were required. These minor deviation could be tolerated. All in all the method looks promising to reduce the impact of zero mass flow issues.

6 Summary and Outlook

In this paper we demonstrated the reason for the slow integration process of fluid system close to vanishing mass flow rates: Due to the high non-linearity of the problem the solution of the non-linear system for calculating an integration step fails, as the simplified chord method is used. The system can only be solved for steps smaller than the steps required from the dynamics of the system itself. And additionally the Jacobian has to be updated almost every step. Therefore, more step which are itself even more computational expensive are required. The problem can be aggravated due to inaccuracies in the calculation of the Jacobian matrix depending on the choice of the states. Though we focused on models of fluid systems, this phenomenon is not restricted to this kind of problem. It occurs for any model with high non-linearity in state where a system should come to rest.

Some approaches which help to avoid the issues were given: If possible the non linearity of the model should be reduced e.g. by linearization close to the zero mass flow rate. The linearization interval can be adapted to the local model behavior. Additionally, it is helpful to make other variables independent of the solution of the mass flow rate if the mass flow rate becomes small. Furthermore, it is advisable to improve the accuracy of the Jacobian matrix calculation e.g. by modified state choice or by a analytic calculation of the Jacobian matrix. With the knowledge of the exact cause of the problem it might be possible to develop additional improvements or enhance the existing once. For example it might be possible to develop an automatic calculation of the linearization interval for the relation of mass flow rate and pressure drop.

As the zero mass flow issue comes from the non-linear solution process, a modification of that process was suggested as well. By including a damping procedure in the solution process, the solution process for small mass flow rates can be improved. Unfortunately, this method only improves solution process if the iterative solution "overshoots" the actual solution. Non-linearity might also lead to a slow iteration progress without overshooting: for example when calculating the root of $\dot{m}^2 = 0$. If a dynamic momentum balance is used, the problem to solve is in that nature and a damping of the steps is not helpful anymore. This problem should be analyzed as well as it is done here for models without mass flow rate state. It might be possible to improve the solution process as well, e.g. by including factors or solution processes which do an approximated update of the (inverted) Jacobian matrix during iteration (e.g. Broyden method). The usage of these methods might be useful in the presented cases as well.

References

Dahmen, W. and A. Reusken (2008). *Numerik für Ingenieure und Naturwissenschaftler*. Springer-Lehrbuch. Springer Berlin Heidelberg. ISBN: 9783540764939.

Dassault Systemes AB (2023). *Dymola Release Notes*.

Dermont, Pieter et al. (2016-05). “Advances of Zero Flow Simulation of Air Conditioning Systems using Modelica”. In: *Proceedings of the 1st Japanese Modelica Conference, Tokyo, Japan*. Vol. 144, pp. 139–144. DOI: 10.3384/ecp16124139.

Gershgorin, S. (1931). “Über die Abgrenzung der Eigenwerte einer Matrix”. In: *Bulletin de l’Académie des Sciences de l’URSS. Classe des sciences mathématiques et na* (6), pp. 749–754.

Hindmarsh, Alan C. et al. (2023). *User Documentation for CVODE*. https://sundials.readthedocs.io/en/latest/cvode. v6.5.1. URL: https://sundials.readthedocs.io/en/latest/cvode.

J. Brunnemann, D. Rideout (2008). “Properties of the Volume Operator in Loop Quantum Gravity II: Detailed Presentation”. In: *Class.Quant.Grav.* (25:065002).

Kelley, Carl T (1995). *Iterative methods for linear and nonlinear equations*. SIAM.

Li, Lixiang et al. (2020-11). “Fast Simulations of Air Conditioning Systems Using Spline-Based Table Look-Up Method (SBTL) with Analytic Jacobians”. In: *American Modelica Conference*. DOI: 10.3384/ECP2016964.

Modelica Standard Library 4.0.0 (2023-04-18). URL: https://github.com/modelica/ModelicaStandardLibrary.

Qiao, Hongtao and Christopher R Laughman (2022). “Performance Enhancements for Zero-Flow Simulation of Vapor Compression Cycles”. In: *Modelica Conferences*, pp. 128–135. DOI: 10.3384/ECP21186128.