Johannes Kasimir

Task 1: Q-N with Anderson Acceleration

Definitions:

$$\mathbf{x}^{k+1} := \mathbf{N}(\mathbf{D}(\mathbf{x}^k)) \tag{1}$$

is the DN iteration on the interface Dirichlet values $\mathbf{x}^k := \{\mathbf{u}_\Gamma^{L,k}, \mathbf{u}_\Gamma^{R,k}\}$ where L and R signifies the left respectively the right shared interface in the room geometry.

$$\mathbf{H}(\mathbf{x}) = \mathbf{N}(\mathbf{D}(\mathbf{x})) \tag{2}$$

is the relaxed DN iteration and

$$\mathbf{R}(\mathbf{x}) = \mathbf{H}(\mathbf{x}) - \mathbf{x} \tag{3}$$

is the residual.

Using a Quasi-Newton method we want to solve the problem $\mathbf{R}(\mathbf{x}) = 0$. Define the matrices

$$\mathbf{V}_k = \left[\mathbf{R}(\mathbf{x}^1) - \mathbf{R}(\mathbf{x}^0), \quad \dots, \quad \mathbf{R}(\mathbf{x}^k) - \mathbf{R}(\mathbf{x}^{k-1})\right] \tag{4}$$

$$\mathbf{W}_k = \left[\mathbf{x}^1 - \mathbf{x}^0, \quad \dots, \quad \mathbf{x}^k - \mathbf{x}^{k-1}\right]. \tag{5}$$

The "Newton-updates" are defined by

$$\Delta\mathbf{x}^k := \mathbf{W}_k \alpha \tag{6}$$

where

$$\alpha := argmin_\alpha ||\mathbf{V}_k \alpha - \mathbf{R}(\mathbf{x}^k)||_2. \tag{7}$$

and the next interface Dirichlet values are

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k. \tag{8}$$

My first impression of this method was that it is significantly more robust than using optimal relaxation parameters. Changing the time step size or heat conductivity and -capacity barely influences the number of iterations required to convergence. However, the mesh width does influence the convergence rate, reducing Δx increases the number of iterations required. We did not investigate the relationship between the convergence rate and Δx closer, but the number of iterations required appears to be approximately inversely proportional to Δx. The high number of iterations required might indicate a bug in the code.

From the convergence history in figure 3 we see that most of the convergence happens in the last few iterations, a behavior resembling GMRES. But in contrast to GMRES the residual increases in some iterations. This can be compared to the DN iteration in figure 4 which has the expected log-linear convergence history of a linear iterative method. The DN iteration using optimal smoothing parameters converges faster, but finding the optimal smoothing parameter can be difficult. When Δt grows large the

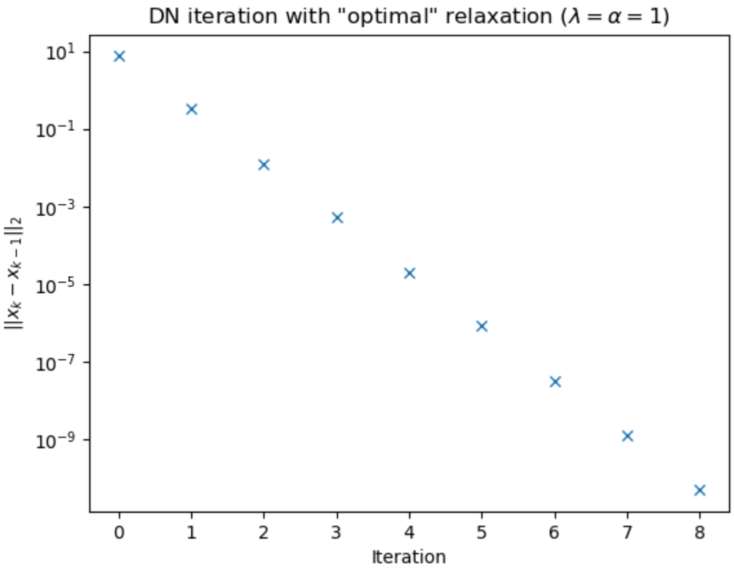

Figure 1: Convergence history for the DN iteration in the first time step. The relaxation parameter was not optimized exactly but by trial and error, $\theta = 0.64$.

optimal smoothing parameter approaches the expected limit value $\lambda_N/(\lambda_D + \lambda_N)$ but for smaller/intermediate size Δt the value might be far from either of the limits $c \to 0$ or $c \to \text{inf}$.

When $\lambda = \alpha = 1$ then the solution reaches the steady state of the heat equation in relatively few time steps. When this happens the convergence rates of the DN and QN iterations increase until they always converge in a single step. This is natural since in the stationary state nothing changes in-between time steps, and therefore our initial guess for the interface Dirichlet values is already at the fixed point of the iterations.

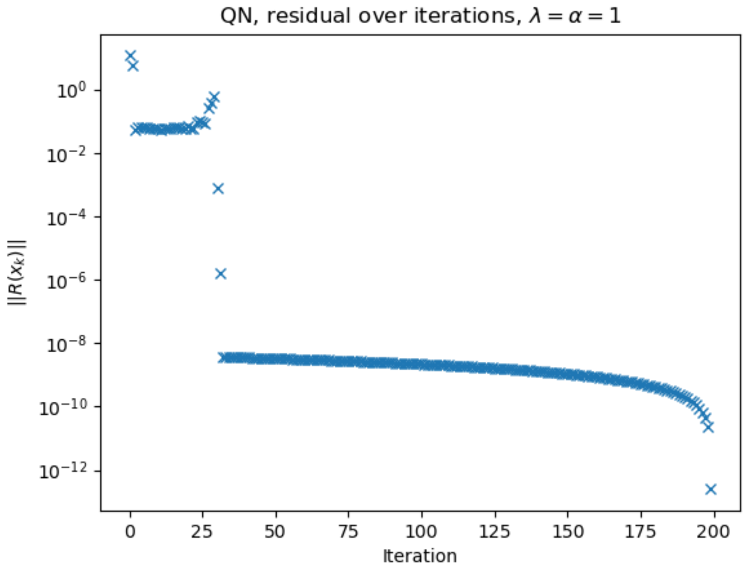

Figure 2: Convergence history for the QN iteration in the first time step.

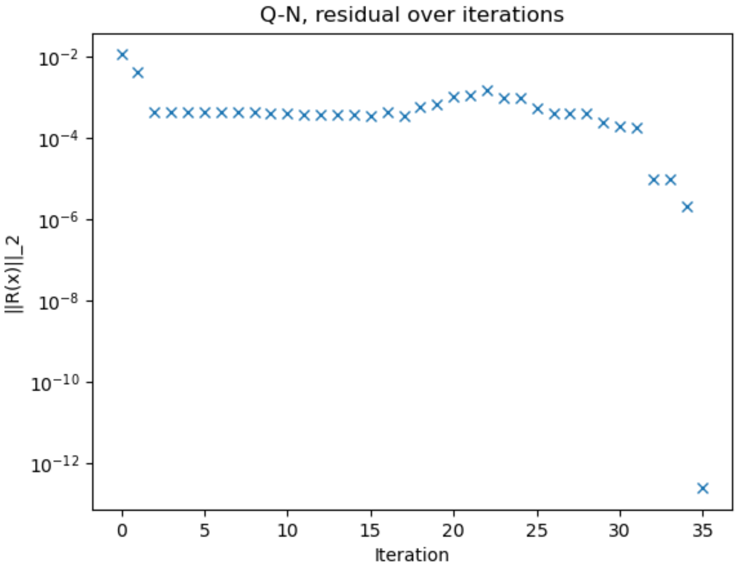

Figure 3: Convergence history for the QN iteration in a typical time step when $\lambda = 0.0234$ and $\alpha = 1300$.

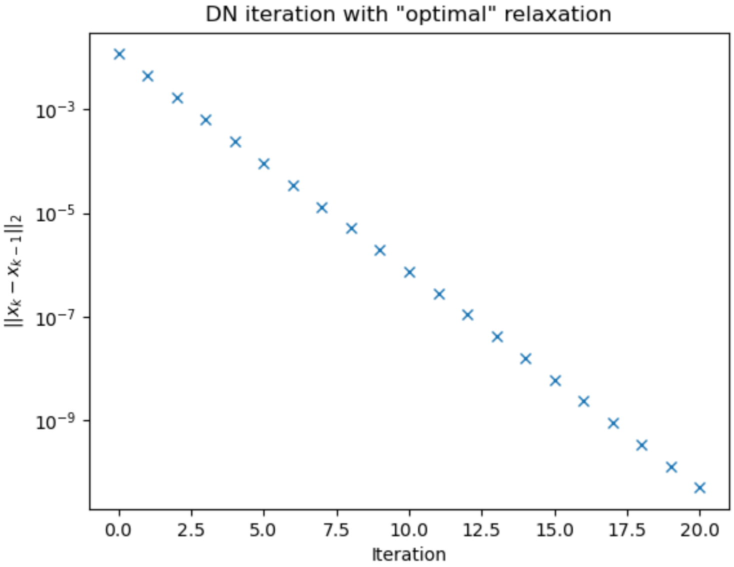

Figure 4: Convergence history for the DN iteration in a typical time step when $\lambda = 0.0234$ and $\alpha = 1300$. The relaxation parameter was not optimized exactly but by trial and error, $\theta = 0.62$.

1 Task 2: SDIRK for scalar ODE

We are solving the equation:

$$\frac{d}{dt}\mathbf{u}(t) = \mathcal{L}(t, \mathbf{u}(t)) \tag{9}$$

where $\mathcal{L}$ is a possibly nonlinear operator acting on $u(t)$. The time integration is done using an adaptive SDIRK, *diagonally implicit Runge Kutta* method, of the form

$$\mathbf{u}_{n+1} = \mathbf{u}_n + h\sum_{i=1}^{s} b_i \mathbf{k}_i \tag{10}$$

$$\mathbf{k}_i = \mathcal{L}(t_n + c_i h, \mathbf{u}_n + h\sum_{j=1}^{i} a_{ij}\mathbf{k}_j). \tag{11}$$

A DIRK scheme is executed by solving a series of nonlinear equation systems. At each stage i, solve

$$\mathbf{k}_i = \mathcal{L}(\bar{t}_i, \bar{\mathbf{u}}_i + \alpha_i \mathbf{k}_i) \tag{12}$$

where $\alpha_i = ha_{ii}$ and $\bar{t}_i$ is a prescribed time, $\bar{\mathbf{u}}_i$ is a linear combination of the steps from previous stages and the point $\mathbf{u}_n$.

In our case the operator is

$$\mathcal{L}(t, \mathbf{u}) = \begin{bmatrix} 0 & 1 \\ -k/m & 0 \end{bmatrix} \mathbf{u} + \frac{A}{m} \begin{bmatrix} 0 \\ p(t) - p_0 \end{bmatrix} \tag{13}$$

and the system we need to solve every stage is

$$(\mathbf{I} - \alpha_i \begin{bmatrix} 0 & 1 \\ -k/m & 0 \end{bmatrix})\mathbf{k}_i = \begin{bmatrix} 0 & 1 \\ -k/m & 0 \end{bmatrix} \bar{\mathbf{u}}_i + \frac{A}{m} \begin{bmatrix} 0 \\ p(t) - p_0 \end{bmatrix}. \tag{14}$$

The method should be time adaptive. This is achieved using an embedded RK method of order $\hat{p} < p$ where p is the order of the SDIRK method. At the end of every timestep, the error is estimated by

$$\hat{\mathbf{e}} = \sum_{i=1}^{s} d_i \mathbf{k}_i \tag{15}$$

for some constants d_i given by the method. The next time step is calculated as

$$\Delta t := 0.9\Delta t \left(\frac{TOL}{||\hat{\mathbf{e}}||}\right)^{\frac{1}{\hat{p}+1}}. \tag{16}$$

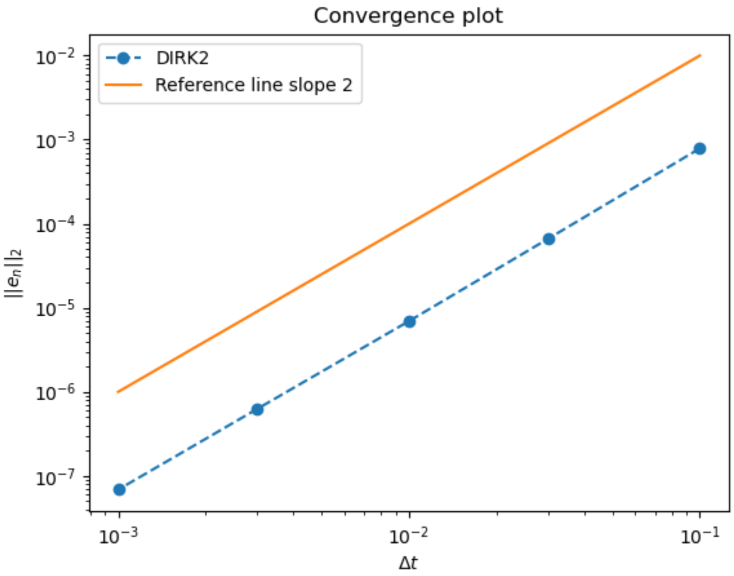

Figure 5: Error at last time step over Δt.

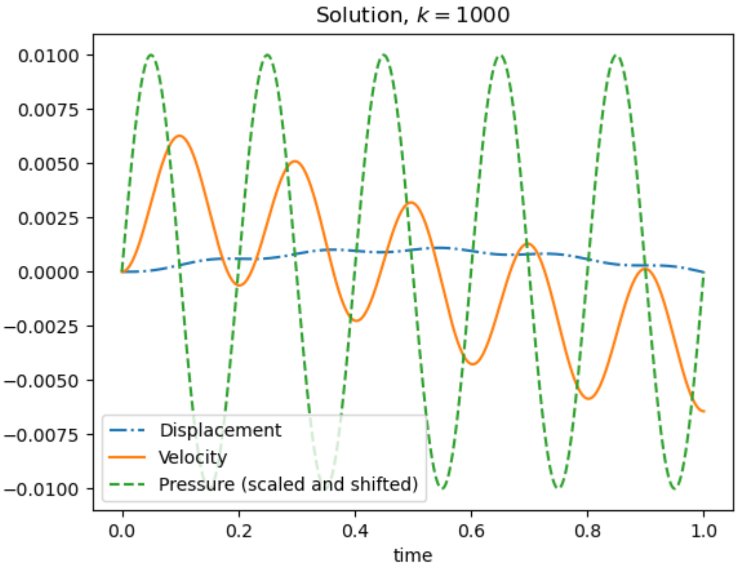

Figure 6:

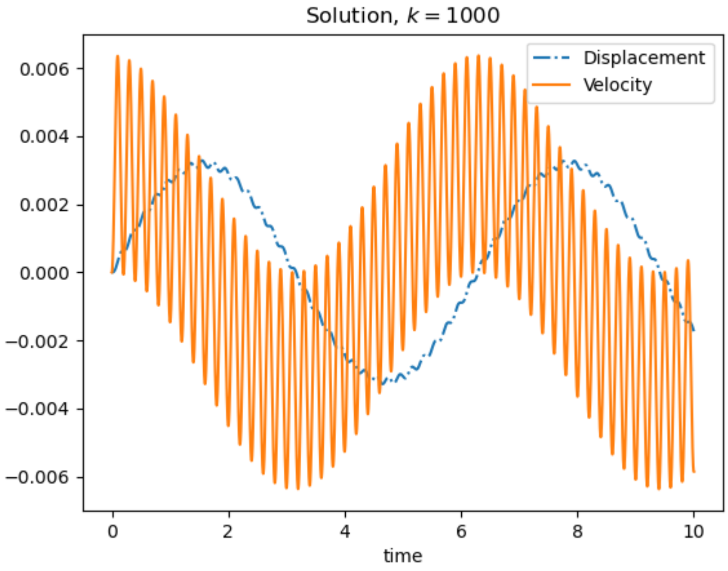

Figure 7: Lower stiffness leads to larger deformations and a lower characteristic frequency of the piston.