Analysis of a semi–Toeplitz preconditioner for a convection-diffusion problem*

Samuel Sundberg[†]and Lina von Sydow[‡]

April 22, 2002

Abstract

We have defined and analyzed a semi-Toeplitz preconditioner for time-dependent and steady-state convection-diffusion problems. The preconditioner exhibits very good theoretical convergence properties. The analysis is corroborated by numerical experiments.

1 Introduction

We are interested in solving the linearised Navier-Stokes equations over a plate or in a 2D-channel. In this paper we study a scalar model problem which mimics this kind of flow in order to analyze our solution method. We consider two different types of problem, a time-dependent one where we want to solve the problem accurately in time and a stationary problem where we are only interested in the steady-state solution. For the time-dependent problem we are interested in low Mach number flows where the fast scale (the sound waves) is not present in the solution. An explicit time-discretization would yield a severe restriction on the time-step to be used due to this fast scale and hence we will employ an implicit scheme in time. Thus, we are faced with a large linear system of equations to solve (each time-step for the time-dependent problem)

$$Au = b, \tag{1.1}$$

in both problem settings.

The coefficient-matrix A in (1.1) is large and sparse, which means that a direct method to solve this system is not well suited due to the extensive amount of fill-in. Thus, we will employ an iterative method. Here, we will use GMRES (Generalized Minimal Residuals), [7], which is a Krylov subspace method well suited for this kind of problems.

*This work was supported by the Swedish Research Council for Engineering Sciences, under contracts No. 222-97-771 and 222-99-695

†Information Technology, Department of Scientific Computing, Uppsala University, Box 120, SE-751 04 Uppsala, Sweden, `Samuel.Sundberg@tdb.uu.se`

‡Information Technology, Department of Scientific Computing, Uppsala University, Box 120, SE-751 04 Uppsala, Sweden, `Lina.von.Sydow@tdb.uu.se`

When iterative methods are used to solve systems of equations in large scale computations two objectives are important: efficiency and reliability. In order to achieve these, preconditioning of the systems is a common technique, which often improves the convergence behaviour considerably. In this report we present an analysis of a semi-Toeplitz preconditioner for the scalar model problem mentioned above. The analysis is corroborated by numerical experiments.

We employ left preconditioning of (1.1) with a preconditioner M,

$$M^{-1}Au = M^{-1}b. \tag{1.2}$$

In e.g. [1] it is shown that the residual reduction depends on the location of the eigenvalues of $M^{-1}A$ and the conditioning of its eigenvector-matrix. If the matrix $M^{-1}A$ is diagonalizable, $M^{-1}A = V\Lambda V^{-1}$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ and λ_i are the eigenvalues of $M^{-1}A$, the residual reduction in step k is estimated by

$$\|r^k\|_2/\|r^0\|_2 \leq \mathrm{cond}_2\,(V) \min_{p_k(0)=1} \max_{i=1,\ldots,n} |p_k(\lambda_i)| \equiv \mathrm{cond}_2\,(V) \cdot \epsilon_k. \tag{1.3}$$

Here $r^k = M^{-1}(Au^k - b)$, $\mathrm{cond}_2\,(V) = \|V\|_2 \cdot \|V^{-1}\|_2$ is the condition number of the eigenvector-matrix V of $M^{-1}A$ and p_k is the k-th residual polynomial for the method. Thus, we are able to estimate the effect of the preconditioner once we have expressions for the eigenvalues and eigenvectors of the preconditioned system.

In the analysis we first focus on the spectrum of the preconditioned system and find analytical expressions for the eigenvalues. Then we let the number of grid points go to infinity to find out what happens asymptotically with the eigenvalues. We also find estimates for the condition number reduction of the eigenvector-matrix. Hence, we are able to say something about the convergence behaviour for GMRES applied to (1.2).

The outline of the paper is the following. In Section 2 we introduce the model problem studied, while Section 3 is devoted to the presentation of the semi-Toeplitz preconditioner. We derive the eigenvalues of the preconditioned system in Section 4 and go on with an asymptotic analysis in Section 5. In Section 6 we study the eigenvectors of the preconditioned system. Finally we present some numerical results in Section 7 to corroborate the analysis, and we end up with some conclusions in Section 8.

2 The model problem

We consider the model problem

$$\left(\frac{\partial u}{\partial t}\right) - \nu \frac{\partial^2 u}{\partial x_2^2} + \frac{\partial u}{\partial x_1} + v\frac{\partial u}{\partial x_2} = 0, \qquad 0 \leq x_1, x_2 \leq 1, \tag{2.1}$$

where we have Dirichlet boundary conditions at all boundaries except at $x_1 = 1$. If we impose $u(x_1, 0) = 0$ and

$$v = c_v\sqrt{\nu}, \quad c_v > 0, \tag{2.2}$$

this equation models the behaviour of laminar boundary layers for the Navier–Stokes equations, see [4]. If the free-stream value of u is $\mathcal{O}(1)$, the resulting boundary layer will have thickness $\mathcal{O}(\sqrt{\nu})$.

We will present results from both the time-dependent setting of the problem and the stationary version. Thus we separate the presentation of these two problem-settings.

2.1 The time-dependent problem

We start by defining a computational grid in space

$$x_{d,j} = jh_d, \quad j = 0, 1, \ldots, m_d, \quad d = 1, 2, \quad h_1 = 1/m_1, \quad h_2 = 1/(m_2 + 1). \tag{2.3}$$

We discretize (2.1) in space using D_0 for first-order terms except at $x_1 = 1$, where we use D_- instead. Second order terms are discretized with D_+D_-. We notice that we have second order terms in the x_2-direction only, where we have Dirichlet boundary conditions at $x_2 = 0$ and $x_2 = 1$ and hence no special precaution has to be taken for those terms. For temporal discretization we use the trapezoidal rule.

This yields the system

$$A^t u^{n+1} = b, \tag{2.4}$$

where $A^t \in \mathbb{R}^{m_1 m_2 \times m_1 m_2}$ is a block tridiagonal matrix

$$A^t = \begin{pmatrix} A_1^t & A_3^t & & \\ A_2^t & \ddots & \ddots & \\ & \ddots & \ddots & A_3^t \\ & & A_2^t & A_1^t \end{pmatrix}$$

where $A_d^t \in \mathbb{R}^{m_1 \times m_1}$,

$$A_1^t = \begin{pmatrix} 4+2\eta^t & \kappa_1^t & & & \\ -\kappa_1^t & \ddots & \ddots & & \\ & \ddots & \ddots & \kappa_1^t & \\ & & -\kappa_1^t & 4+2\eta^t & \kappa_1^t \\ & & & -2\kappa_1^t & 4+2\eta^t+2\kappa_1^t \end{pmatrix},$$

$A_2^t = -(\xi^t + \eta^t)I_{m_1}, \qquad A_3^t = (\xi^t - \eta^t)I_{m_1}.$

Here we have introduced the quantities

$$\kappa_d^t = \frac{\Delta t}{h_d}, \quad d = 1, 2, \tag{2.5}$$

and

$$\xi^t = v\kappa_2^t, \quad \eta^t = \frac{2\nu\kappa_2^t}{h_2}. \tag{2.6}$$

In order to resolve the boundary layer we need a number of grid-points within the layer. Hence we impose

$$h_2 = c_h\sqrt{\nu}, \quad 0 < c_h < 1. \tag{2.7}$$

In order to simulate the convergence behaviour for the linearised Navier-Stokes equation with low Mach numbers, we will employ large κ_d^t:s. In order to ensure time-accuracy we must thus assume that the solution u varies slowly in time compared to the variation in space. This is usually the case in laminar boundary layers.

2.2 The stationary problem

In [4], Holmgren et al. employ a semicirculant preconditioner in combination with pseudo time-stepping using Euler forward to solve this problem. They also compare this preconditioner with a multigrid technique. We will further discuss their method and try to make comparisons later on in this paper.

We will use the computational grid defined in (2.3) and the same spatial discretisation as for the time-dependent problem. This yields the following system of equations to solve

$$A^s u^{n+1} = b, \tag{2.8}$$

where $A^s \in \mathbb{R}^{m_1 m_2 \times m_1 m_2}$ is a block tridiagonal matrix

$$A^s = \begin{pmatrix} A_1^s & A_3^s & & \\ A_2^s & \ddots & \ddots & \\ & \ddots & \ddots & A_3^s \\ & & A_2^s & A_1^s \end{pmatrix}$$

where $A_d^s \in \mathbb{R}^{m_1 \times m_1}$,

$$A_1^s = \begin{pmatrix} 2\eta^s & \kappa_1^s & & & \\ -\kappa_1^s & \ddots & \ddots & & \\ & \ddots & \ddots & \kappa_1^s & \\ & & -\kappa_1^s & 2\eta^s & \kappa_1^s \\ & & & -2\kappa_1^s & 2\eta^s + 2\kappa_1^s \end{pmatrix},$$

$A_2^s = -(\xi^s + \eta^s) I_{m_1}, \qquad A_3^s = (\xi^s - \eta^s) I_{m_1}.$

This time we have defined

$$\kappa_d^s = \frac{1}{h_d}, \quad d = 1, 2,$$

and

$$\xi^s = \frac{v}{h_2}, \quad \eta^s = \frac{2\nu}{h_2^2}. \tag{2.9}$$

We impose the restriction (2.7) also this time in order to resolve the boundary layer.

3 Preconditioning

3.1 Definition

We will use left preconditioning and solve (1.2) with GMRES. The preconditioning matrices M^t and M^s for the time-dependent and stationary problem respectively are defined by

$$M^{t,s} = \begin{pmatrix} M_1^{t,s} & M_3^{t,s} & & \\ M_2^{t,s} & \ddots & \ddots & \\ & \ddots & \ddots & M_3^{t,s} \\ & & M_2^{t,s} & M_1^{t,s} \end{pmatrix}$$

where $M_d^{t,s} \in \mathbb{R}^{m_1 \times m_1}$,

$$M_1^t = \begin{pmatrix} 4+2\eta^t & \kappa_1^t & & \\ -\kappa_1^t & \ddots & \ddots & \\ & \ddots & \ddots & \kappa_1^t \\ & & -\kappa_1^t & 4+2\eta^t \end{pmatrix}, M_1^s = \begin{pmatrix} 2\eta^s & \kappa_1^s & & \\ -\kappa_1^s & \ddots & \ddots & \\ & \ddots & \ddots & \kappa_1^s \\ & & -\kappa_1^s & 2\eta^s \end{pmatrix},$$

$M_2^{t,s} = A_2^{t,s}, \qquad M_3^{t,s} = A_3^{t,s}$.

In order to analyze the spectrum of the coefficient-matrix $\left(M^{t,s}\right)^{-1} A^{t,s}$ we introduce the error matrix $E^{t,s}$ defined by $E^{t,s} = A^{t,s} - M^{t,s} = I_{m_2} \otimes E_1^{t,s}$, where

$$E_1^{t,s} = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & 0 & 0 \\ 0 & & -\kappa_1^{t,s} & 2\kappa_1^{t,s} \end{pmatrix},$$

and $\otimes$ denotes the Kronecker-product.

Using the relation given by the definition of the error matrix we rewrite the preconditioned coefficient-matrix

$$\begin{aligned} &\left(M^{t,s}\right)^{-1} A^{t,s} = \left(M^{t,s}\right)^{-1} \left(M^{t,s} + E^{t,s}\right) = I + \left(M^{t,s}\right)^{-1} E^{t,s} = \\ &I + \left(M^{t,s}\right)^{-1} \left(I_{m_2} \otimes E_1^{t,s}\right). \end{aligned} \tag{3.1}$$

As a consequence of (3.1) the spectrum of $\left(M^{t,s}\right)^{-1} A^{t,s}$ is a simple translation of the spectrum of $\left(M^{t,s}\right)^{-1} E^{t,s}$. This simplifies the analysis as we will see in the sequel.

3.2 Existence of $\left(M^{t,s}\right)^{-1}$

Next we will show that the inverse of $M^{t,s}$ exists for both problem settings.

3.2.1 The time-dependent problem

We can find explicit expressions for the eigenvalues and eigenvectors of M^t using the formulas for tridiagonal Toeplitz matrices found in [8]. The preconditioner can be written

$$M^t = I_{m_2} \otimes M_1^t + L^t \otimes I_{m_1},$$

where $L^t \in \mathbb{R}^{m_2 \times m_2}$ is a tridiagonal matrix

$$L^t = \begin{pmatrix} 0 & (\xi^t - \eta^t) & & \\ -(\xi^t + \eta^t) & \ddots & \ddots & \\ & \ddots & \ddots & (\xi^t - \eta^t) \\ & & -(\xi^t + \eta^t) & 0 \end{pmatrix}. \tag{3.2}$$

Both L^t and M_1^t can be decomposed by different sine matrices

$$S_{m_1}^H M_1^t S_{m_1} = \Lambda_1^t = \operatorname{diag}(\lambda_{1,1}^t, \lambda_{1,2}^t, \ldots, \lambda_{1,m_1}^t),$$

$$\widetilde{S}_{m_2}^{-1} L^t \widetilde{S}_{m_2} = \Lambda_2^t = \operatorname{diag}(\lambda_{2,1}^t, \lambda_{2,2}^t, \ldots, \lambda_{2,m_2}^t),$$

where

$$\lambda_{1,j}^t = 4 + 2\eta^t + 2i\kappa_1^t \cos\left(\tfrac{j\pi}{m_1+1}\right), \qquad j = 1, 2, \ldots, m_1,$$

$$S_{m_1}(j,k) = \sqrt{\tfrac{2}{m_1+1}}\, i^{j+k+1} \sin\left(\tfrac{jk\pi}{m_1+1}\right), \qquad j,k = 1, 2, \ldots, m_1,$$

and

$$\begin{aligned} \lambda_{2,k}^t &= -2\sqrt{(\eta^t)^2 - (\xi^t)^2} \cos\left(\frac{k\pi}{m_2+1}\right), \qquad k = 1, 2, \ldots, m_2, \\ \widetilde{S}_{m_2}(j,k) &= \left(\frac{-(\xi^t + \eta^t)}{(\xi^t - \eta^t)}\right)^{\frac{j}{2}} \sin\left(\frac{jk\pi}{m_2+1}\right), \qquad j,k = 1, 2, \ldots, m_2. \end{aligned} \tag{3.3}$$

We now define $D^t = (\widetilde{S}_{m_2}^{-1} \otimes S_{m_1}^H) M^t (\widetilde{S}_{m_2} \otimes S_{m_1})$, yielding

$$\begin{aligned} D^t &= (\widetilde{S}_{m_2}^{-1} \otimes S_{m_1}^H)(I_{m_2} \otimes M_1^t + L^t \otimes I_{m_1})(\widetilde{S}_{m_2} \otimes S_{m_1}) \\ &= I_{m_2} \otimes S_{m_1}^H M_1^t S_{m_1} + \widetilde{S}_{m_2}^{-1} L^t \widetilde{S}_{m_2} \otimes I_{m_1} \\ &= I_{m_2} \otimes \Lambda_1^t + \Lambda_2^t \otimes I_{m_1}. \end{aligned}$$

Hence, the matrix D^t is a diagonal matrix with the diagonal values given by the eigenvalues of M_1^t and L^t. We introduce the notation

$$D^t = \operatorname{diag}(D_1^t, \ldots, D_{m_2}^t), \qquad D_k^t = \operatorname{diag}\left(d_{1,k}^t, \ldots, d_{m_1,k}^t\right),$$

where

$$d^t_{j,k} = 4 + 2\eta^t + 2i\kappa^t_1 \cos\left(\tfrac{j\pi}{m_1+1}\right) - 2\sqrt{(\eta^t)^2 - (\xi^t)^2}\cos\left(\tfrac{k\pi}{m_2+1}\right),$$

$$\begin{cases} j = 1, 2, \dots, m_1, \\ k = 1, 2, \dots, m_2. \end{cases}$$

We now have explicit expressions for the eigenvalues and eigenvectors of M^t, and we have

$$\mathcal{R}e(d^t_{j,k}) = 4 + 2\eta^t - 2\sqrt{(\eta^t)^2 - (\xi^t)^2}\cos\left(\frac{k\pi}{m_2+1}\right) \geq 4, \quad \forall c_v c_h \leq 2, \tag{3.4}$$

where c_v and c_h are defined in (2.2) and (2.7) respectively. The condition $c_v c_h \leq 2$ makes sure that the term $2\sqrt{(\eta^t)^2 - (\xi^t)^2}\cos(k\pi/(m_2+1))$ is real.

We thus know that the inverse of M^t exists and can be written

$$\left(M^t\right)^{-1} = (\widetilde{S}_{m_2} \otimes S_{m_1})\left(D^t\right)^{-1}(\widetilde{S}^{-1}_{m_2} \otimes S^H_{m_1}).$$

3.2.2 The stationary problem

The preconditioner for the stationary problem can also be decomposed as

$$M^s = I_{m_2} \otimes M^s_1 + L^s \otimes I_{m_1},$$

where $L^s \in \mathbb{R}^{m_2 \times m_2}$ is a tridiagonal matrix

$$L^s = \begin{pmatrix} 0 & (\xi^s - \eta^s) & & \\ -(\xi^s + \eta^s) & \ddots & \ddots & \\ & \ddots & \ddots & (\xi^s - \eta^s) \\ & & -(\xi^s + \eta^s) & 0 \end{pmatrix}.$$

Following the steps in the previous section we end up with

$$(M^s)^{-1} = (\widetilde{S}_{m_2} \otimes S_{m_1})(D^s)^{-1}(\widetilde{S}^{-1}_{m_2} \otimes S^H_{m_1}).$$

where

$$D^s = \operatorname{diag}(D^s_1, \dots, D^s_{m_2}), \qquad D^s_k = \operatorname{diag}\left(d^s_{1,k}, \dots, d^s_{m_1,k}\right),$$

where

$$d^s_{j,k} = 2\eta^s + 2i\kappa^s_1 \cos\left(\tfrac{j\pi}{m_1+1}\right) - 2\sqrt{(\eta^s)^2 - (\xi^s)^2}\cos\left(\tfrac{k\pi}{m_2+1}\right),$$

$$\begin{cases} j = 1, 2, \dots, m_1, \\ k = 1, 2, \dots, m_2. \end{cases}$$

Hence, we have explicit expressions for the eigenvalues and eigenvectors of M^s and since

$$\mathcal{R}e(d^s_{j,k}) = 2\eta^s - 2\sqrt{(\eta^s)^2 - (\xi^s)^2}\cos\left(\frac{k\pi}{m_2+1}\right) \geq 0, \quad \forall c_v c_h < 2$$

we know that the inverse of M^s exists.

3.3 Preconditioner solve

The preconditioner system, $M^{t,s}x = y$, is in both cases solved using a Fast Modified Sine Transform and the solution of small tridiagonal systems. This solution procedure which requires $\mathcal{O}(m_1 m_2 \log_2(m_1))$ arithmetic operations can be described by the following three steps:

1. Perform Fourier transforms of vectors $y \in \mathbb{C}^{\frac{m_1+1}{2}}$.
2. Solve $\frac{m_1+1}{2}$ tridiagonal systems of order m_2.
3. Perform invers Fourier transforms of vectors $y \in \mathbb{C}^{\frac{m_1+1}{2}}$.

For further details concerning the preconditioner solve we refer to [2].

4 Eigenvalues of $\left(M^{t,s}\right)^{-1} A^{t,s}$

4.1 Introduction

We would like to find analytical expressions for the eigenvalues of $\left(M^{t,s}\right)^{-1} A^{t,s}$. The way this is done is somewhat unintuitive. As we noted in the previous section the spectrum of $\left(M^{t,s}\right)^{-1} A^{t,s}$ is related to the spectrum of $\left(M^{t,s}\right)^{-1} E^{t,s}$ by (3.1). We introduce a similarity transformation $W^{t,s}$ of $\left(M^{t,s}\right)^{-1} E^{t,s}$ and find the eigenvalues of $W^{t,s}$, which is relatively easy due to the simple structure of the matrix. We also make use of a lemma in [3] to reduce the complexity in computing the entries of $W^{t,s}$. This finally gives us an expression for the eigenvalues of $\left(M^{t,s}\right)^{-1} A^{t,s}$.

We now look at the matrix $W^{t,s}$ defined by

$$\begin{aligned}
W^{t,s} &= (\widetilde{S}_{m_2}^{-1} \otimes I_{m_1})\left(M^{t,s}\right)^{-1} E^{t,s} (\widetilde{S}_{m_2} \otimes I_{m_1}) \\
&= (\widetilde{S}_{m_2}^{-1} \otimes I_{m_1})(\widetilde{S}_{m_2} \otimes S_{m_1})\left(D^{t,s}\right)^{-1} (\widetilde{S}_{m_2}^{-1} \otimes S_{m_1}^H) E^{t,s} (\widetilde{S}_{m_2} \otimes I_{m_1}) \\
&= (I_{m_2} \otimes S_{m_1})\left(D^{t,s}\right)^{-1} (I_{m_2} \otimes S_{m_1}^H E_1^{t,s}) \\
&= \operatorname{diag}(S_{m_1}\left(D_1^{t,s}\right)^{-1} S_{m_1}^H E_1^{t,s}, \ldots, S_{m_1}\left(D_{m_2}^{t,s}\right)^{-1} S_{m_1}^H E_1^{t,s}) \\
&\equiv \operatorname{diag}(W_1^{t,s}, \ldots, W_{m_2}^{t,s}),
\end{aligned}$$

where $W_k^{t,s} \in \mathbb{R}^{m_1 \times m_1}$ only has two non-zero columns,

$$W_k^{t,s} = \begin{pmatrix} 0 & \cdots & 0 & W_k^{t,s}(1, m_1-1) & W_k^{t,s}(1, m_1) \\ \vdots & & \vdots & W_k^{t,s}(2, m_1-1) & W_k^{t,s}(2, m_1) \\ \vdots & & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & W_k^{t,s}(m_1, m_1-1) & W_k^{s}(m_1, m_1) \end{pmatrix}.$$

In the following two subsections we will show how to compute the eigenvalues of $W_k^{t,s}$.

4.2 The time-dependent problem

The entries of each block W_k^t can be computed as a sum given by

$$
\begin{aligned}
W_k^t(l, m_1 - 1) &= -\kappa_1^t \sum_{j=1}^{m_1} \frac{S_{m_1}(l,j) S_{m_1}^H(j, m_1)}{d_{j,k}^t}, \\
W_k^t(l, m_1) &= 2\kappa_1^t \sum_{j=1}^{m_1} \frac{S_{m_1}(l,j) S_{m_1}^H(j, m_1)}{d_{j,k}^t}.
\end{aligned}
\tag{4.1}
$$

Using Lemma 5.2 in [3] we find that the sums in (4.1) can be computed as

$$
\begin{aligned}
&\textstyle\sum_{j=1}^{m_1} S_{m_1}(l,j) d_{j,k}^{-1} S_{m_1}^H(j, m_1) = \\
&i^{l-m_1} \Phi_{m_1,l}\left(2 + \eta^t - \sqrt{(\eta^t)^2 - (\xi^t)^2} \cos\left(\tfrac{k\pi}{m_2+1}\right), \kappa_1^t\right)
\end{aligned}
\tag{4.2}
$$

where the function Φ is defined as

$$
\Phi_{m_1,l}(\alpha, \beta) = \frac{i}{\beta}\left(\frac{z^{m_1+l} - z^{|m_1-l|}}{z - z^{-1}} + \frac{(z^{m_1} - z^{-m_1})(z^l - z^{-l})}{z - z^{-1}} \frac{z^{2m_1+2}}{1 - z^{2m_1+2}}\right)
$$

and z is given by

$$
z = i\left(\frac{\alpha}{\beta} - \sqrt{1 + \left(\frac{\alpha}{\beta}\right)^2}\right). \tag{4.3}
$$

In Equation (4.2) we have used that $m_1 = 2^k - 1$ for some integer k. This relation is due to the fact that the preconditioner solve includes Fast Modified Sine Transforms in the x_1-direction.

We can now derive an analytical expression for the eigenvalues of W^t. The characteristic equation for W_k^t is

$$
\begin{aligned}
0 &= \det(\lambda I_{m_1} - W_k^t) \\
&= \lambda^{m_1-2} \begin{vmatrix} \lambda - W_k^t(m_1-1, m_1-1) & -W_k^t(m_1-1, m_1) \\ -W_k^t(m_1, m_1-1) & \lambda - W_k^t(m_1, m_1) \end{vmatrix} \\
&= \lambda^{m_1-1}\Bigg((\lambda - i\kappa_1^t \Phi_{m_1,m_1-1}\left(2 + \eta^t - \sqrt{(\eta^t)^2 - (\xi^t)^2}\cos\left(\tfrac{k\pi}{m_2+1}\right), \kappa_1^t\right) - \\
&\quad 2\kappa_1^t \Phi_{m_1,m_1}\left(2 + \eta^t - \sqrt{(\eta^t)^2 - (\xi^t)^2}\cos\left(\tfrac{k\pi}{m_2+1}\right), \kappa_1^t\right)\Bigg))
\end{aligned}
$$

and we find that W_k^t only has one non-zero eigenvalue given by

$$
\begin{aligned}
\lambda_k^t &= i\kappa_1^t \Phi_{m_1,m_1-1}\left(2 + \eta^t - \sqrt{(\eta^t)^2 - (\xi^t)^2}\cos\left(\tfrac{k\pi}{m_2+1}\right), \kappa_1^t\right) + \\
&\quad 2\kappa_1^t \Phi_{m_1,m_1}\left(2 + \eta^t - \sqrt{(\eta^t)^2 - (\xi^t)^2}\cos\left(\tfrac{k\pi}{m_2+1}\right), \kappa_1^t\right) = \\
&= -\left(z_k^t\right)^2 \tfrac{1-\left(z_k^t\right)^{2m_1-2}}{1-\left(z_k^t\right)^{2m_1+2}} + 2iz_k^t \tfrac{1-\left(z_k^t\right)^{2m_1}}{1-\left(z_k^t\right)^{2m_1+2}},
\end{aligned}
\tag{4.4}
$$

where

$$
\begin{aligned}
z_k^t &= i\left(\chi_k^t - \sqrt{1+(\chi_k^t)^2}\right), \\
\chi_k^t &= \left(2+\eta^t - \sqrt{(\eta^t)^2 - (\xi^t)^2}\cos\left(\tfrac{k\pi}{m_2+1}\right)\right)/\kappa_1^t.
\end{aligned}
\tag{4.5}
$$

Since $(M^t)^{-1}A^t = I + (M^t)^{-1}E^t$ and W^t is a similarity transformation of $(M^t)^{-1}E^t$, we get the eigenvalues of $(M^t)^{-1}A^t$ by simply adding 1 to the eigenvalues of W^t.

We are now ready to state the following theorem regarding the eigenvalues of $(M^t)^{-1}A^t$.

Theorem 4.1 *The preconditioned coefficient-matrix $(M^t)^{-1}A^t$ has $(m_1-1)m_2$ eigenvalues that are identically 1 and m_2 eigenvalues μ_k^t that are given by*

$$
\mu_k^t = 1 - \left(z_k^t\right)^2 \frac{1-\left(z_k^t\right)^{2m_1-2}}{1-\left(z_k^t\right)^{2m_1+2}} + 2iz_k^t\frac{1-\left(z_k^t\right)^{2m_1}}{1-\left(z_k^t\right)^{2m_1+2}},
\tag{4.6}
$$

where z_k^t is defined in (4.5).

We note that due to the condition in (3.4), χ_k^t is real. This implies that z_k^t is imaginary and furthermore that μ_k^t is real. Thus the spectrum of $(M^t)^{-1}A^t$ lies on the real axis.

4.3 The stationary problem

We now proceed with the stationary problem in exactly the same manner as for the time-dependent problem and state the equivalence to Theorem 4.1.

Theorem 4.2 *The preconditioned coefficient-matrix $(M^s)^{-1}A^s$ has $(m_1-1)m_2$ eigenvalues that are identically 1. The remaining m_2 eigenvalues are given by*

$$
\mu_k^s = 1 - (z_k^s)^2 \frac{1-(z_k^s)^{2m_1-2}}{1-(z_k^s)^{2m_1+2}} + 2iz_k^s\frac{1-(z_k^s)^{2m_1}}{1-(z_k^s)^{2m_1+2}}.,
\tag{4.7}
$$

where

$$
z_k^s = i\left(\chi_k^s - \sqrt{1+(\chi_k^s)^2}\right) \quad , \quad \chi_k^s = h_1\left(\eta^s - \sqrt{(\eta^s)^2-(\xi^s)^2}\cos\left(\tfrac{k\pi}{m_2+1}\right)\right).
\tag{4.8}
$$

By the same argument as in the previous subsection we conclude that μ_k^s are all real.

5 Asymptotic spectrum of $\left(M^{t,s}\right)^{-1} A^{t,s}$

We now look at the spectrum of the preconditioned system when $m_1 \to \infty$. The eigenvalues of the preconditioned systems that are different from 1 are given in (4.6) and (4.7), and in order to analyze the asymptotic behaviour we define a relation between the number of gridpoints in the x_1 and the x_2 direction

$$m_2 = \phi m_1, \qquad 0 < \phi < \infty. \tag{5.1}$$

A short remark regarding the asymptotic analysis is needed here. In the limit $m_1 = \infty$ we have $\nu = 0$ and thus the model problem degenerates to a first order problem where we can't impose Dirichlet boundary conditions at $x_2 = 1$. This is not a problem in real computations, however, as we always have a finite number of grid points. For a thorough discussion of this phenomenon we refer to a paper by Kim and Parter, [5].

We now go over to study the two different problem settings.

5.1 The time-dependent problem

Using (5.1), (2.2), and (2.7) together with the relations (2.5) and (2.6) we get

$$\begin{aligned} \xi^t &= v\kappa_2^t = \frac{c_v \kappa_1^t}{c_h m_1} \\ \eta^t &= \frac{2\nu\kappa_2^t}{h_2} = \frac{2\kappa_1^t}{c_h^2 m_1}. \end{aligned} \tag{5.2}$$

With (5.2) in (4.5) we get

$$\chi_k^t = \frac{2}{\kappa_1^t} + \frac{2}{c_h^2 m_1} - \sqrt{\left(\frac{2}{c_h^2 m_1}\right)^2 - \left(\frac{c_v}{c_h m_1}\right)^2} \cos\left(\frac{k\pi}{m_2+1}\right).$$

Now assume that κ_1^t, ϕ, c_v and c_h are fixed and let $m_1 \to \infty$. This gives that $\lim_{m_1\to\infty} \chi_k^t = \frac{2}{\kappa_1^t}$ and we obtain

$$z_\infty^t \equiv \lim_{m_1\to\infty} z_k^t = i\left(\frac{2}{\kappa_1^t} - \sqrt{1+\left(\frac{2}{\kappa_1^t}\right)^2}\right).$$

If we assume that κ_1^t is large we can write

$$\begin{aligned} (z_\infty^t)^2 &= -1\left(\tfrac{2}{\kappa_1^t} - \sqrt{1+\left(\tfrac{2}{\kappa_1^t}\right)^2}\right)^2 = \\ &= -1 + \tfrac{4}{\kappa_1^t}\sqrt{1+\left(\tfrac{2}{\kappa_1^t}\right)^2} - \tfrac{8}{\left(\kappa_1^t\right)^2} = -1+\delta, \quad 0<\delta<1. \end{aligned}$$

Hence, $\lim_{m_1\to\infty}\left(z^t_\infty\right)^{2m_1} = 0$ and we finally get that we asymptotically have only two eigenvalues which are 1 and

$$\mu^t_\infty = 1 + \left(\frac{2}{\kappa^t_1} - \sqrt{1+\left(\frac{2}{\kappa^t_1}\right)^2}\right)^2 - 2\left(\frac{2}{\kappa^t_1} - \sqrt{1+\left(\frac{2}{\kappa^t_1}\right)^2}\right)$$

By Taylor expanding μ^t_∞ around $1/\kappa^t_1 = 0$ we are ready to state the following theorem.

Theorem 5.1 *In the limit $m_1 \to \infty$, $\left(M^t\right)^{-1} A^t$ has only two eigenvalues given by 1 and*

$$\mu^t_\infty = 4 - \frac{8}{\kappa^t_1} + \frac{12}{\left(\kappa^t_1\right)^2} + \mathcal{O}\left(\frac{1}{\left(\kappa^t_1\right)^3}\right).$$

A comment regarding the assumption that κ^t_1 is large might be needed here. The reason for this is that we are interested in solving low Mach number flows as we said in Section 1. The reasonable way to mimic this situation for this scalar model problem is that we assume that the flow varies slowly in time, while we might have a faster variation in space. Hence, we can use a much larger time-step compared to the space-step and still expect high accuracy in time.

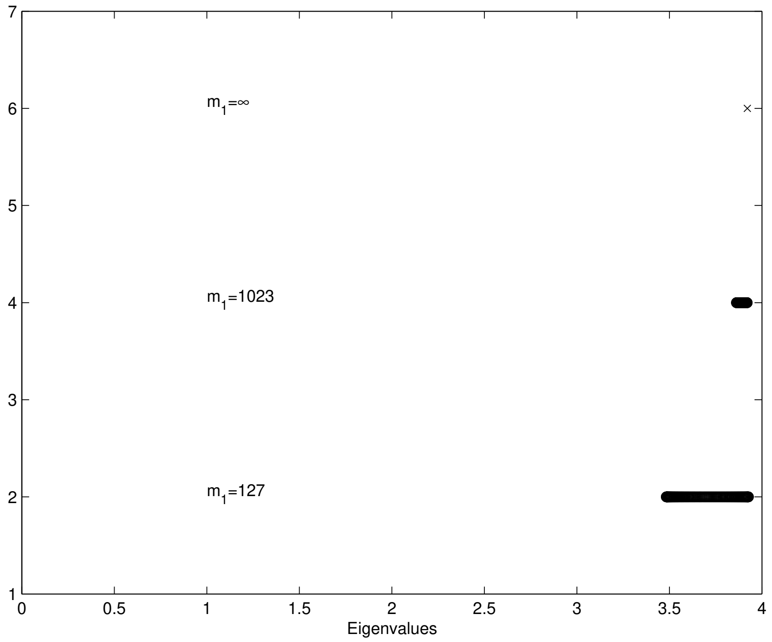

Figure 5.1: Plot of eigenvalues of $\left(M^t\right)^{-1} A^t$ (circles) different from 1 for $m_1 = 127$ and $m_1 = 1023$, when $\kappa^t_1 = 100.0$, $\phi = 1.0$, $c_v = 1.0$, and $c_h = 0.5$ together with theoretical eigenvalue μ^s_∞ (cross) for $m_1 = \infty$.

In Figure 5.1 we see two plots of the spectrum for different problem-sizes, and we note that the eigenvalues appear to converge to the theoretical limit as the grid is refined.

From this analysis we can draw the conclusion that the iteration will converge in only two iterations for a problem that is large enough. In the limit $m_1 \to \infty$ the iteration is independent of the constants κ_1^t, ϕ, c_v, and c_h. Moreover, the spectrum seems to contract the larger problem we solve, i.e. the iteration will probably converge faster for larger problem-sizes. This is a very good property that we will numerically verify in Section 7.

5.2 The stationary problem

Using (5.1), (2.2), and (2.7) together with (2.9) we get

$$\begin{aligned} \xi^s &= \frac{c_v}{c_h} \\ \eta^s &= \frac{2}{c_h^2}. \end{aligned} \tag{5.3}$$

With (5.3) in (4.8) we get

$$\begin{aligned} z_k^s &= i\left(\tfrac{\alpha_k}{m_1} - \sqrt{1+\left(\tfrac{\alpha_k}{m_1}\right)^2}\right), \\ \alpha_k &= \tfrac{2}{c_h^2}\left(1-\sqrt{1-\tfrac{(c_v c_h)^2}{4}\cos\left(\tfrac{k\pi}{m_2+1}\right)}\right). \end{aligned}$$

Now assume that ϕ, c_v and c_h are fixed yielding

$$\alpha_\infty(\theta) = \lim_{m_1\to\infty}\alpha_k = \tfrac{2}{c_h^2}\left(1-\sqrt{1-\tfrac{(c_v c_h)^2}{4}\theta}\right) \quad , \quad -1 \le \theta \le 1 \tag{5.4}$$

and we obtain

$$\begin{aligned} &\lim_{m_1\to\infty} z_k^s = -i, \\ &\lim_{m_1\to\infty} (z_k^s)^2 = -1, \\ &\lim_{m_1\to\infty} (z_k^s)^{2m_1} = \lim_{m_1\to\infty}\left(-1+\tfrac{2\alpha_k}{m_1}+\mathcal{O}\left(\tfrac{1}{m_1^2}\right)\right)^{m_1} = -e^{-2\alpha_\infty}, \end{aligned}$$

where we have Taylor expanded around $1/m_1 = 0$ and used that m_1 is odd in the last expression. We sum this up in the following theorem.

Theorem 5.2 *In the limit $m_1 \to \infty$ the eigenvalues to $(M^s)^{-1} A^s$ are identically 1 or given by*

$$\mu_\infty^s(\theta) = 2\left(1+\frac{1+e^{-2\alpha_\infty(\theta)}}{1-e^{-2\alpha_\infty(\theta)}}\right),$$

where $\alpha_\infty(\theta)$ is defined in (5.4).

From Theorem 5.2 we can immediately state another theorem.

Theorem 5.3 *In the limit $m_1 \to \infty$, the eigenvalues to $\left(M^s\right)^{-1} A^s$ are identically 1 or lie on the real axis between a and b, $a < b$ defined by*

$$\begin{aligned} a &= 2\left(1 + \tfrac{1+e^{-2A}}{1-e^{-2A}}\right) \quad , \quad A = \tfrac{2}{c_h^2}\left(1 + \sqrt{1 - \tfrac{(c1 \cdot c2)^2}{4}}\right), \\ b &= 2\left(1 + \tfrac{1+e^{-2B}}{1-e^{-2B}}\right) \quad , \quad B = \tfrac{2}{c_h^2}\left(1 - \sqrt{1 - \tfrac{(c1 \cdot c2)^2}{4}}\right). \end{aligned} \tag{5.5}$$

Proof The proof follows directly from Theorem 5.2. ■

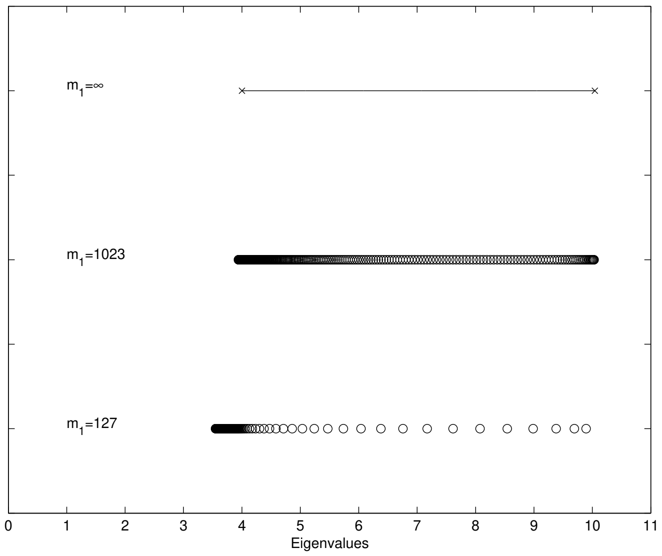

Figure 5.2: Plot of eigenvalues of $\left(M^s\right)^{-1} A^s$ (circles) different from 1 for $m_1 = 127$ and $m_1 = 1023$, when $\phi = 1.0$, $c_v = 1.0$, and $c_h = 0.5$ together with theoretical eigenvalues $\mu^s_\infty(\theta)$ (line with endpoints marked with crosses) for $m_1 = \infty$.

From Figure 5.2 we see that the eigenvalues μ^s_k of $\left(M^s\right)^{-1} A^s$ very well agree with the asymptotic values $\mu^s(\theta)$ even for finite problem-sizes. Hence, we can hope that the convergence rate for the iteration for finite problem-sizes will agree with the theory below for the asymptotic convergence rate.

Next we define the asymptotic convergence factor ρ by

$$\rho \equiv \lim_{k \to \infty} \epsilon_k^{1/k} \tag{5.6}$$

where ϵ_k is defined in (1.3).

Following e.g. Saad [6] we get an upper bound on ρ which we now state and prove.

Theorem 5.4 *The asymptotic convergence factor* ρ *defined in (5.6) satisfies*

$$\rho \leq \frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}. \tag{5.7}$$

Proof Relation (5.7) follows immediately from Theorem 5.2 and [6]. ■

We also define the residual reduction after 20 iterations $\tilde{\rho}^{20}$ by

$$\tilde{\rho}^{20} = \left(\frac{\|r^{20}\|_2}{\|r^0\|_2} \right)^{1/20} \tag{5.8}$$

In Figure 5.3 we have plotted the upper bound on the asymptotic convergence factor defined in (5.7) and the residual reduction $\tilde{\rho}^{20}$, (5.8).

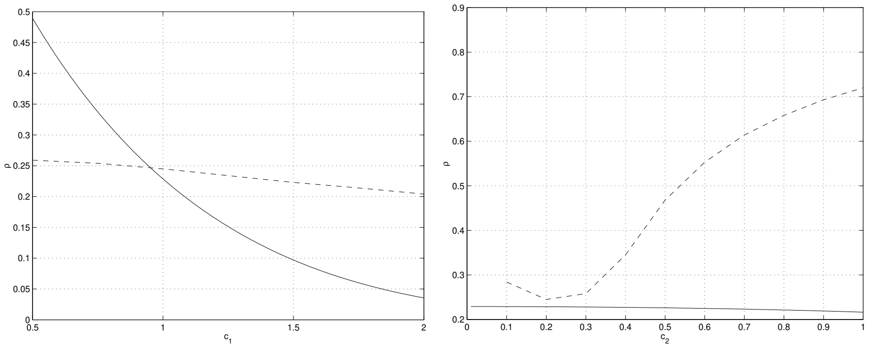

Figure 5.3: Plot of ρ (solid) and $\tilde{\rho}^{20}$ (dashed) as functions of c_v and c_h for $m_1 = 127$ and $\phi = 1.0$. In the left picture we have used $c_h = 0.2$ and in the right $c_v = 1.0$.

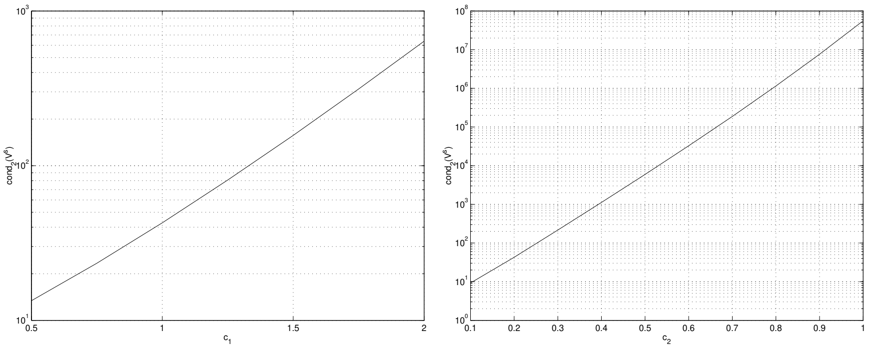

Figure 5.4: Plot of $\mathrm{cond}_2(V^s)$ as a function of c_v and c_h. In the left picture we have used $c_h = 0.2$ and in the right $c_v = 1.0$.

We see that the residual reduction for the preconditioned iteration is less than 1 in all cases which is necessary for convergence. Clearly, the non-normality of the eigenvector-matrix V^s of $(M^s)^{-1} A^s$ defined in (6.1) cannot be neglected. In Figure 5.4 we display $\mathrm{cond}_2(V^s)$ as a function of c_v and c_h. The extreme increase in $\mathrm{cond}_2(V^s)$ with increasing c_h probably explains the fairly large discrepancy between ρ and $\tilde{\rho}^{20}$ for "large" c_h's.

6 Condition number reduction of the eigenvector-matrix

We now turn our attention to the eigenvector-matrix of $(M^{t,s})^{-1} A^{t,s}$. It turns out that we can use essentially the same analysis as Hemmingsson and Otto use in [3], but we have included some theorems and conclusions here as to make the presentation complete.

By Theorem 5.4 in [3] we know that the eigenvector-matrix $V^{t,s}$ of $(M^{t,s})^{-1} A^{t,s}$ can be written as

$$V^{t,s} = \left(\widetilde{S}_{m_2} \otimes I_{m_1}\right) \mathrm{diag}\left(U_1^{t,s}, \ldots, U_{m_2}^{t,s}\right), \tag{6.1}$$

where $\widetilde{S}_{m_2}$ is defined in Equation (3.3) and $U_k^{t,s} \in \mathbb{C}^{m_1 \times m_1}$ is given by

$$U_k^{t,s} = \left[e_1, \ldots, e_{m_1-2}, (2e_{m_1-1} + e_{m_1})/\sqrt{5}, \left(u^{t,s}\right)^{(k)} / \| \left(u^{t,s}\right)^{(k)} \|_2\right], k = 1, \ldots, m_2, \tag{6.2}$$

where e_j denotes the jth canonical column vector and

$$\left(u^{t,s}\right)_l^{(k)} = i^l \left(\left(z_k^{t,s}\right)^l - \left(z_k^{t,s}\right)^{-l}\right), \qquad l = 1, \ldots, m_1, \tag{6.3}$$

where $z_k^{t,s}$ is defined in (4.5) and (4.8) respectively.

We are now able to estimate the *condition number reduction* for the eigenvector-matrix defined by

$$\psi \equiv \mathrm{cond}_2\left(V^{t,s}\right) / \mathrm{cond}_2\left(\tilde{V}^{t,s}\right), \tag{6.4}$$

where $\tilde{V}^{t,s}$ is the eigenvector-matrix to $A^{t,s}$ in (2.4) and (2.8). From [3] we get

$$\psi^{t,s} \leq \psi_0^{t,s} = \max_{1 \leq k \leq m_2} \|U_k^{t,s}\|_2 \cdot \max_{1 \leq k \leq m_2} \| \left(U_k^{t,s}\right)^{-1} \|_2 \Big/ \mathrm{cond}_2\left(V_1^{t,s}\right), \tag{6.5}$$

where $V_1^{t,s}$ is the eigenvector-matrix of $A_1^{t,s}$ and $\|U_k^{t,s}\|_2$ and $\| \left(U_k^{t,s}\right)^{-1} \|_2$ can be computed using Theorem 5.5 in [3], which we restate for readability.

Theorem 6.1 *Assume that the eigenvalues λ_k^t defined in (4.4) and λ_k^s (defined similarly for the stationary problem) are nonzero. Then*

$$\begin{aligned} \|U_k^{t,s}\|_2 &= \left(1 + \sqrt{1 - a_k^{t,s}}\right)^{1/2}, \\ \| \left(U_k^{t,s}\right)^{-1} \|_2 &= \left(1 - \sqrt{1 - a_k^{t,s}}\right)^{-1/2}, \end{aligned} \qquad k = 1, \ldots, m_2,$$

where

$$a_k^{t,s} = \frac{\zeta_k^{t,s}}{\varepsilon_k^{t,s}},$$

and

$$\begin{aligned}
\zeta_k^{t,s} &= \tfrac{1}{5}\left| \left(z_k^{t,s}\right)^{m_1-1} - \left(z_k^{t,s}\right)^{-m_1+1} \right|^2 + \tfrac{4}{5}\left| \left(z_k^{t,s}\right)^{m_1} - \left(z_k^{t,s}\right)^{-m_1} \right|^2 - \\
&\quad \tfrac{4}{5}\mathcal{I}m\left(\left(\left(z_k^{t,s}\right)^{m_1-1} - \left(z_k^{t,s}\right)^{-m_1+1} \right) \left(\left(\overline{z_k^{t,s}}\right)^{m_1} - \left(\overline{z_k^{t,s}}\right)^{-m_1} \right) \right) \\
\varepsilon_k^{t,s} &= \frac{|z_k^{t,s}|^2 - |z_k^{t,s}|^{2m_1+2}}{1-|z_k^{t,s}|^2} + \frac{|z_k^{t,s}|^{-2} - |z_k^{t,s}|^{-2m_1-2}}{1-|z_k^{t,s}|^{-2}} - \\
&\quad 2\mathcal{R}e\left(\frac{z_k^{t,s}\left(\overline{z_k^{t,s}}\right)^{-1} - \left(z_k^{t,s}\left(\overline{z_k^{t,s}}\right)^{-1}\right)^{m_1+1}}{1-z_k^{t,s}\left(\overline{z_k^{t,s}}\right)^{-1}} \right).
\end{aligned}$$

In Figure 6.1 we have plotted ψ_0^t defined in (6.5) as a function of problem-size. Clearly the condition number reduction improves with increasing problem-size. Moreover, the condition number reduction seems to be independent on c_v and c_h.

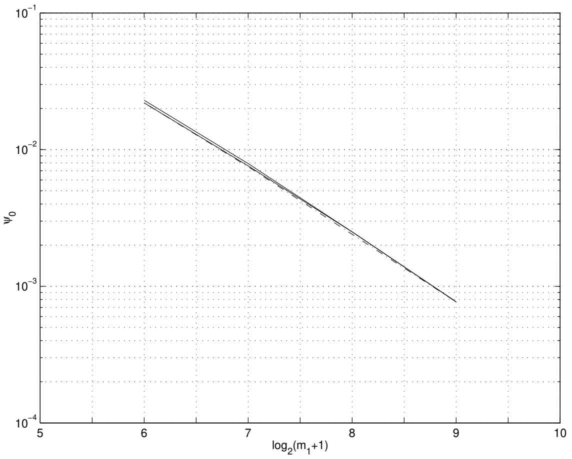

Figure 6.1: Plot of ψ_0^t for increasing number of grid points. We have used $\kappa_1^t = 100.0$ and $\phi = 1$. Solid line represents $c_v = 1.0$, $c_h = 0.5$, dashed-dotted line $c_v = 1.0$, $c_h = 1.0$, and dashed line $c_v = 0.5$, $c_h = 0.5$.

Figure 6.2 shows ψ_0^s defined in (6.5) as a function of problem-size. Again, the condition number reduction improves with increasing problem-size, but the decrease is clearly less compared to the time-dependent case. Finally, the condition number reduction for the stationary problem seems to be fairly independent on c_v and c_h.

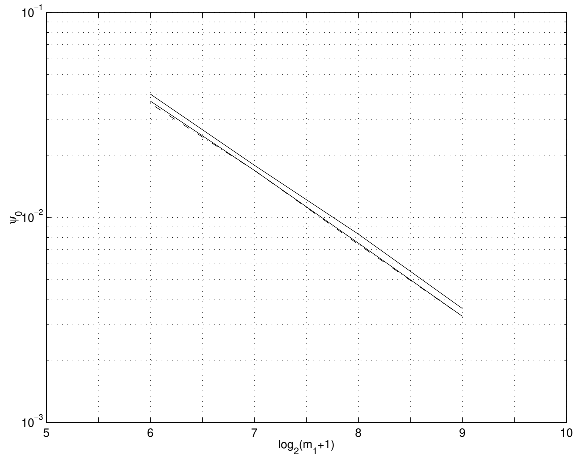

Figure 6.2: Plot of ψ_0^s for increasing number of grid points when $\phi = 1$. Solid line represents $c_v = 1.0$, $c_h = 0.5$, dashed-dotted line $c_v = 1.0$, $c_h = 1.0$, and dashed line $c_v = 0.5$, $c_h = 0.5$.

7 Numerical results

To study the effect of the preconditioner we have solved the model problem (both settings) with GMRES(ℓ). For the time-dependent problem we have used $\ell = 20$, while we have used $\ell = 30$ for the steady-state problem. The reason for choosing this larger restarting length for the steady-state problem is that this is a much harder problem that suffers quite a lot from restarting. During the experiments the relations (2.2) and (2.7) were kept to ensure that the model problem mimics the behaviour of boundary layers for the Navier–Stokes equations and that the layers are sufficiently resolved.

The computer used for the experiments was a Sun Ultra-60 with UltraSPARC-II CPUs at 450 MHz with 4 MB of L2 Cache memory. The total amount of primary RAM was 1 GB. The times presented are wall-clock time.

We have run a number of experiments to numerically verify the good properties of the semi-Toeplitz preconditioner. We have separated the presentation in two parts, one for the time-dependent setting of the problem, and one for the steady-state problem.

7.1 The time-dependent problem

We have run a problem with the following initial and boundary conditions

$$\begin{aligned} u(x_1, 0, t) &= 0, \\ u(x_1, 1, t) &= 1, \\ u(0, x_2, t) &= 2x_2 - x_2^2, \\ u(x_1, x_2, 0) &= 2x_2 - x_2^2. \end{aligned}$$

With this set-up, the solution does not vary very much in time, and we can hope to have good time-accuracy even for large κ_d^t:s. The main issue of these numerical experiments is *not* to present a numerical method with good accuracy-properties but rather to simulate the behaviour of the iterative method in combination with the semi-Toeplitz preconditioner for a low Mach number flow problem. We believe that by using this set-up we achieve this.

We consider the iteration to have converged when

$$\frac{\| (M^t)^{-1} \left(A^t u^{n+1,k} - b\right) \|_2}{\| (M^t)^{-1} b\|_2} < 10^{-6}.$$

In Table 7.1 we display the number of iterations and time to reach convergence, with and without preconditioning for $u^{n+1,0} = u^n$. When we employ the semi-Toeplitz preconditioner, a factorization phase of the preconditioner is included in the time.

	Preconditioned with M^t		Unpreconditioned	
m_1	*#it.*	*time(s.)*	*#it.*	*time(s.)*
31	7	0.0495	158	0.150
63	5	0.145	202	0.830
127	4	0.496	240	4.11
255	3	1.79	321	31.5
511	2	4.68	298	136
1023	2	18.6	222	544

Table 7.1: Number of iterations and time for $\kappa_1^t = 100.0$, $c_v = 1.0$, $c_h = 0.5$, and $\phi = 1.0$.

From Table 7.1 we conclude that the predicted convergence behaviour with the preconditioner is satisfied, i.e. the number of iterations decreases when the problem-size increases, and the theoretical limit of 2 iterations per time-step seems to hold for large problems. Furthermore, we conclude that employing the semi-Toeplitz preconditioner is beneficial in terms of both iteration count and timings. Note that the preconditioned iteration does not have to restart for any of the problem-sizes.

In Table 7.2 we display the number of preconditioned iterations for different initial guesses and right-hand sides. We have used both $u^{n+1,0} = u^n$ and $u^{n+1,0} = 0$. Moreover, we have used the b that we obtain when we solve (2.1) which we denote "flow", and a right-hand side that is equal to 1.

Table 7.2 shows that the number of iterations seems to be independent of right-hand side and fairly independent of initial guess. The first property corroborates that the model problem studied is the right one, i.e. if we get a time-accurate solution or not does not affect the number of iterations. In the sequel we will use $u^{n+1,0} = u^n$ and solve (2.1).

$u^{n+1,0}$	u^n	0	0
b	flow	flow	1
m_1	$\#it.$	$\#it.$	$\#it.$
31	7	8	7
63	5	6	5
127	4	5	5
255	3	4	3
511	2	4	2
1023	2	3	2

Table 7.2: Number of iterations for $\kappa_1^t = 100.0$, $c_v = 1.0$, $c_h = 0.5$, and $\phi = 1.0$.

Next, we will vary κ_1^t, keeping c_v, c_h and ϕ fixed.

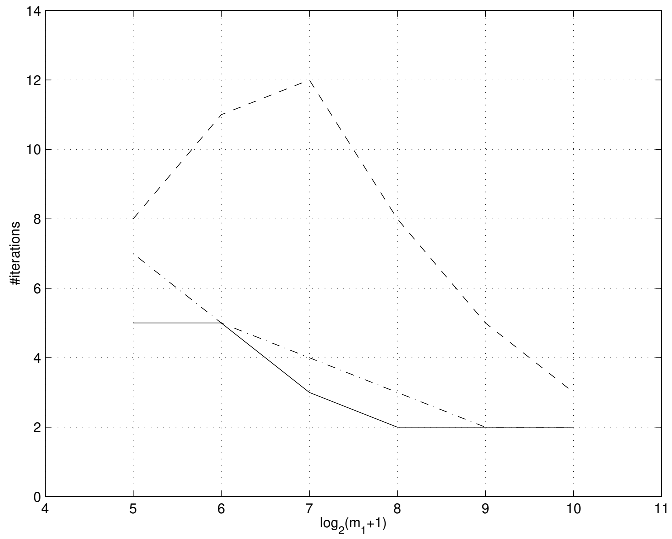

Figure 7.1: Number of iterations for $\kappa_1^t = 10.0$ (solid line), $\kappa_1^t = 100.0$ (dashed-dotted line) and $\kappa_1^t = 1000.0$ (dashed line), when $c_v = 1.0$, $c_h = 0.5$, and $\phi = 1.0$.

From Figure 7.1 it seems as if the number of iterations will end up being only 2 per time-step for large problem-sizes, independent of the time-step. I.e., we conclude that when we solve the low Mach number flow problem using this method, we should use as large time-step as possible that still gives good time-accuracy.

Another parameter that we will vary is c_v, i.e. how large v is compared to $\sqrt{\nu}$, see (2.2). In Figure 7.2 we have varied c_v between 0.5 and 2, keeping κ_1^t, c_h, and ϕ fixed.

It is clear from Figure 7.2 that the number of iterations does not depend much on c_v, at least not in this range.

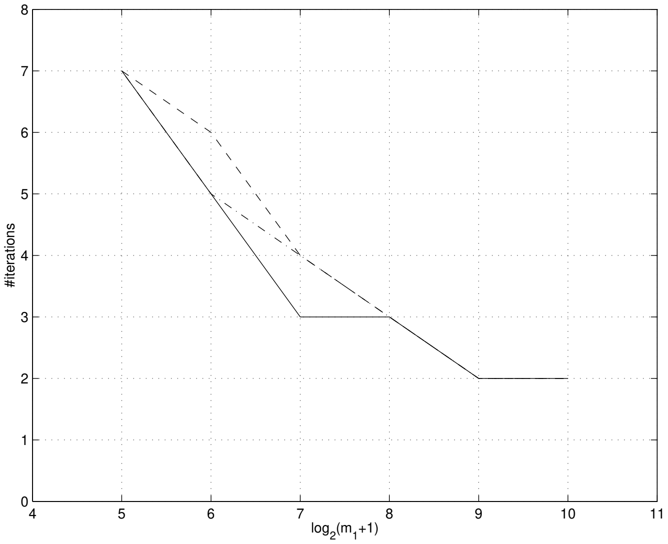

Figure 7.2: Number of iterations for $c_v = 0.5$ (solid line), $c_v = 1.0$ (dashed-dotted line) and $c_v = 2.0$ (dashed line), when $\kappa_1^t = 100.0$, $c_h = 0.5$, and $\phi = 1.0$.

Now, we will vary the parameter that regulates the resolution of the boundary layer, i.e. c_h defined in (2.7), which we display in Figure 7.3.

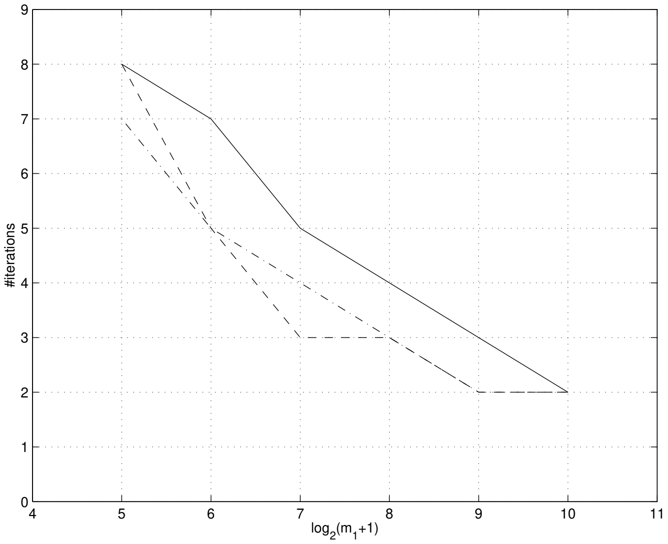

Figure 7.3: Number of iterations for $c_h = 0.1$ (solid line), $c_h = 0.5$ (dashed-dotted line) and $c_h = 1.0$ (dashed line), when $\kappa_1^t = 100.0$, $c_v = 1.0$, and $\phi = 1.0$.

Again, we can conclude that the iteration seems to be independent also on the parameter c_h. Larger c_h:s render slightly fewer iterations. This can partly be an effect of the fact that a larger c_h gives less variance between u^n and u^{n+1} than a smaller c_h does. The trend is in all cases that the number of iterations goes down to 2 with increasing problem-size.

Next, we will examine the dependence on ϕ. Since we have a boundary layer in the x_2-direction, we wish to have a better resolution in this direction compared to the x_1-direction. Therefore, we will vary ϕ defined in (5.1) between 1.0 and 10.0 to see how this affects the number of iterations. Note, that we were not able to fit the largest problem into the computer for $\phi = 10.0$.

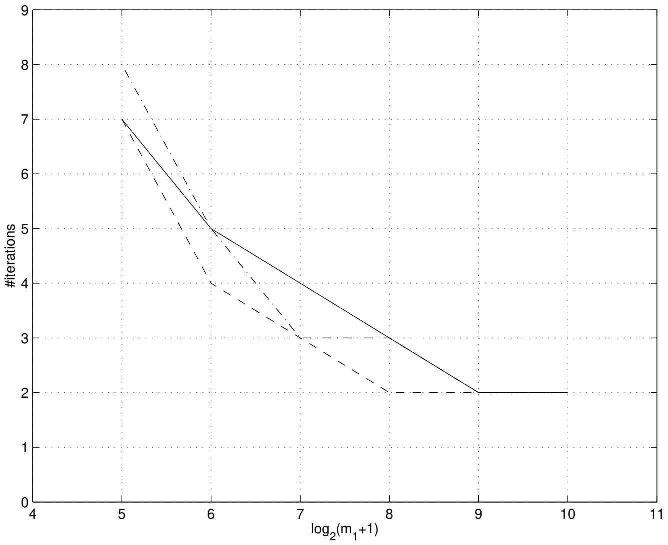

Figure 7.4: Number of iterations for $\phi = 1.0$ (solid line), $\phi = 2.0$ (dashed-dotted line) and $\phi = 10.0$ (dashed line), when $\kappa_1^t = 100.0$, $c_v = 1.0$, and $c_h = 0.5$.

The number of iterations seems to be rather independent also on ϕ. However, there is slightly less iterations for larger ϕ:s. This agrees well with the theory since we solve a larger problem for a larger ϕ if we keep m_1 fixed.

Another way to get a better resolution of the boundary layer is to introduce stretching in the grid. To study the performance of the preconditioner for this case we introduce a stretching in the x_2-direction. The location of the horisontal gridlines is now given by

$$x_{2,k}(\beta_2) = \frac{\exp(\beta_2 k/(m_2+1)) - 1}{\exp \beta_2 - 1}, \tag{7.1}$$

where β_2 is constant. With this distribution of gridlines we get a constant stretching factor, i.e. $h_{2,k+1}/h_{2,k} = \exp(\beta_2/(m_2+1))$. The appearance of the grid for different β_2 is shown in Figure 7.5.

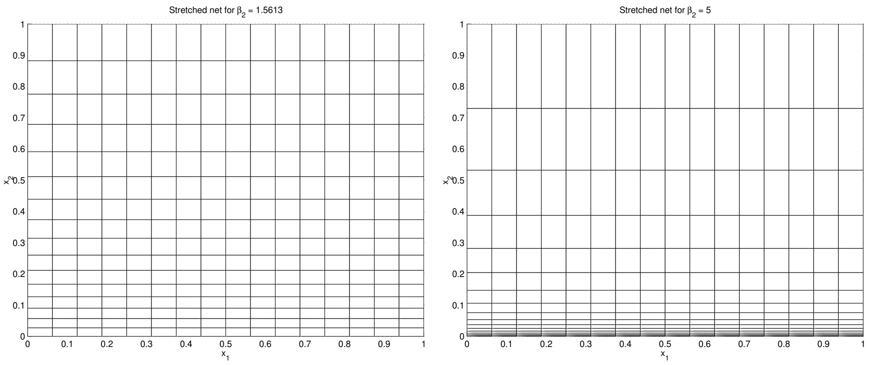

Figure 7.5: The grids generated using $\beta_2 = 1.5613$ and $\beta_2 = 5$.

In Figure 7.6 we display the number of iterations for the the β_2:s shown in Figure 7.5.

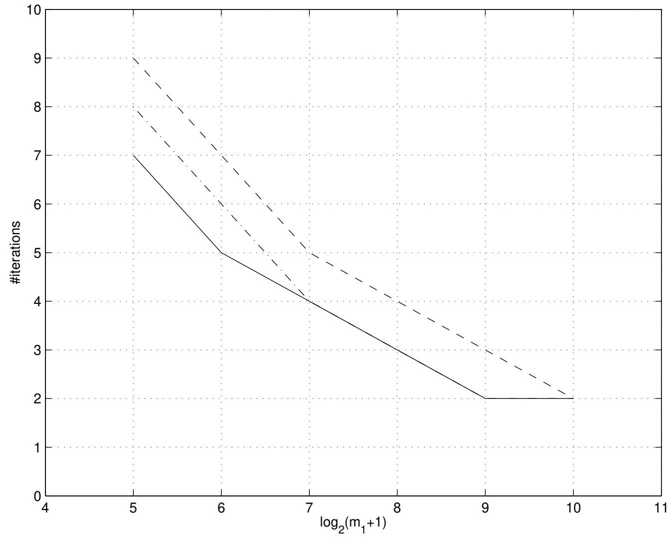

Figure 7.6: Number of iterations for $\beta_2 = 0.0$ (solid line), $\beta_2 = 1.5613$ (dashed-dotted line) and $\beta_2 = 5.0$ (dashed line), when $\kappa_1^t = 100.0$, $c_v = 1.0$, $c_h = 0.5$, and $\phi = 1.0$.

A larger β_2 renders a slight increase in the number of iterations. However, it is clear that this increase is so small that it is beneficial in terms of timings to use stretching in the grid compared to increasing the number of gridpoints to get higher resolution of the boundary layer.

7.2 The stationary problem

We will start by trying to make a comparison with [4]. They have solved (2.1) using Euler forward in time in combination with a semi-circulant convergence accelerator, i.e. a sort of pseudo time-stepping. In their method they have to estimate a parameter Δt, and the number of iterations to reach steady-state will depend on this parameter. It is hard to tell from [4] how strong this dependence is.

It is difficult to make a fair comparison with their method, among other things we don't know what initial guess they use and we use a different convergence criterion.

We have run a problem with the following boundary conditions which are the same as they use in [4].

$$\begin{aligned} u(x_1, 0) &= 0, \\ u(x_1, 1) &= 1, \\ u(0, x_2) &= 2x_2 - x_2^2. \end{aligned}$$

As initial guess we have used $u^0(x_1, x_2) = 2x_2 - x_2^2$. We consider the iteration to have converged when

$$\frac{\| (M^s)^{-1} \left(A^s u^k - b \right) \|_2}{\| (M^s)^{-1} b \|_2} < 10^{-6}.$$

In Table 7.3 we display the number of iterations to reach convergence with the semi-Toeplitz preconditioner. We have chosen $c_v = 1.0$ and $c_h = 1.0$ in accordance with Table 5.3 in [4]. The number of grid-points in the x_1-direction is kept constant $m_1 = 63$.

m_1	ν	$\#it.$
63	$2.4 \cdot 10^{-4}$	15
127	$6.1 \cdot 10^{-5}$	20
255	$1.5 \cdot 10^{-5}$	26
511	$3.8 \cdot 10^{-6}$	37
1023	$9.5 \cdot 10^{-7}$	54

Table 7.3: Number of iterations for $m_1 = 63$, $c_v = 1.0$, and $c_h = 1.0$.

If we compare the results in Table 7.3 with those in Table 5.3 in [4] it is clear that for this particular case the number of iterations is less for the method presented in this paper. However, since the number of iterations depends on the initial guess, the convergence criterion and the restarting length ℓ that we use in GMRES we can't draw any far-reaching conclusions from this result. What we can say, is that the method presented here seems to be competitive with the method presented in [4] in terms of number of iterations (and hence in arithmetic operations since the operation count per iteration for the two methods are similar) *and* that our method does not use any parameter that we need to estimate. The latter property may be very useful for more realistic applications.

Now we will go over to solve a slightly harder problem. The boundary conditions that we use are

$$\begin{aligned} u(x_1, 0) &= 0, \\ u(x_1, 1) &= 1, \\ u(0, x_2) &= 1. \end{aligned}$$

These boundary conditions yields the following picture (Figure 7.7) of the steady-state flow for $m_1 = 127$, $c_v = 1.0$, $c_h = 0.2$, and $\phi = 1.0$. To ensure good resolution of the boundary layer for real applications, we need approximately 25 grid-points within the layer, [9]. For this problem, this is achieved for this parameter-setting, independent of m_1 (and hence m_2).

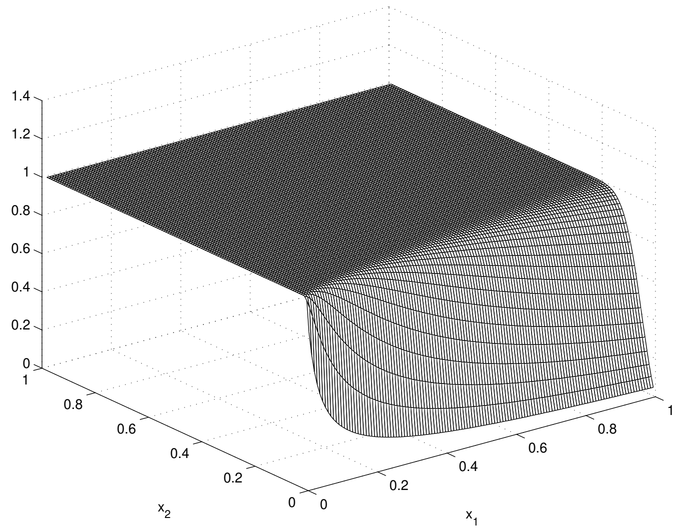

Figure 7.7: Steady-state flow for $m_1 = 127$, $c_v = 1.0$, $c_h = 0.2$, and $\phi = 1.0$.

In order to make sure that preconditioning is necessary for this problem we display the number of iterations and time to reach convergence, with and without preconditioning for in Table 7.4. As initial guess we have used $u^0 = 1.0$. Again, a factorization phase of the preconditioner is included in the time when we employ the semi-Toeplitz preconditioner.

From Table 7.4 we conclude that it is clearly beneficial to use preconditioning for the steady-state problem. However, the number of preconditioned iterations increases with increasing problem-size. We believe that this is due to the non-normality of V^s, and that $\text{cond}_2(V^s)$ increases with increasing problem-size. This can clearly affect the convergence rate according to (1.3).

	Preconditioned with M^t		Unpreconditioned	
m_1	$\#it.$	$time(s.)$	$\#it.$	$time(s.)$
31	10	0.0607	292	0.321
63	10	0.231	447	2.44
127	11	0.961	440	10.24
255	16	5.90	485	126
511	22	28.8	744	2283
1023	30	191	> 1000	-

Table 7.4: Number of iterations and time for $c_v = 1.0$, $c_h = 0.2$, and $\phi = 1.0$.

Next, we will vary the parameter c_v defined in (2.2). In Figure 7.8 we have varied c_v between 0.5 and 2, keeping c_h, and ϕ fixed.

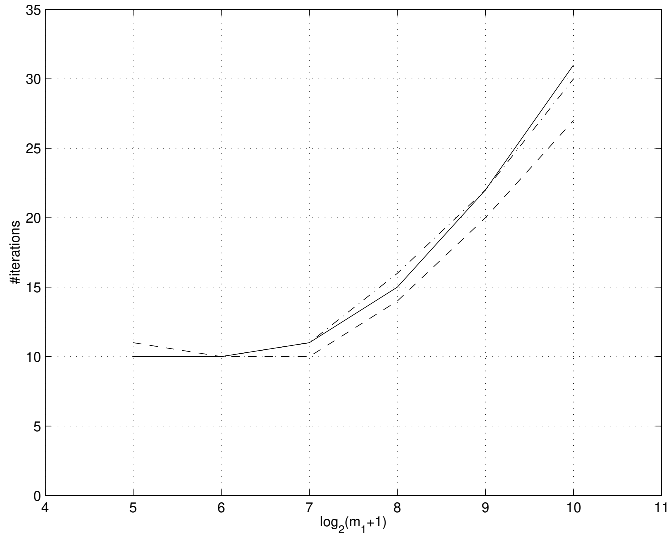

Figure 7.8: Number of iterations for $c_v = 0.5$ (solid line), $c_v = 1.0$ (dashed-dotted line) and $c_v = 2.0$ (dashed line), when $c_h = 0.2$, and $\phi = 1.0$.

It is clear from Figure 7.8 that the number of iterations does not depend much on c_v, at least not in this range.

Now, we will vary the parameter that regulates the resolution of the boundary layer, i.e. c_h defined in (2.7), which we display in Figure 7.9.

From Figure 7.9 it is clear that the number of iterations depends quite strongly on c_h, i.e. the resolution of the boundary layer. The better we resolve the boundary layer, the fewer iterations we get. This is probably due to the increasing non-normality of the eigenvector-matrix to $(M^s)^{-1} A^s$ with increasing c_h that we saw in Section 5.2. Hence, we shall be sure that we resolve the boundary layer well in order to ensure fast convergence.

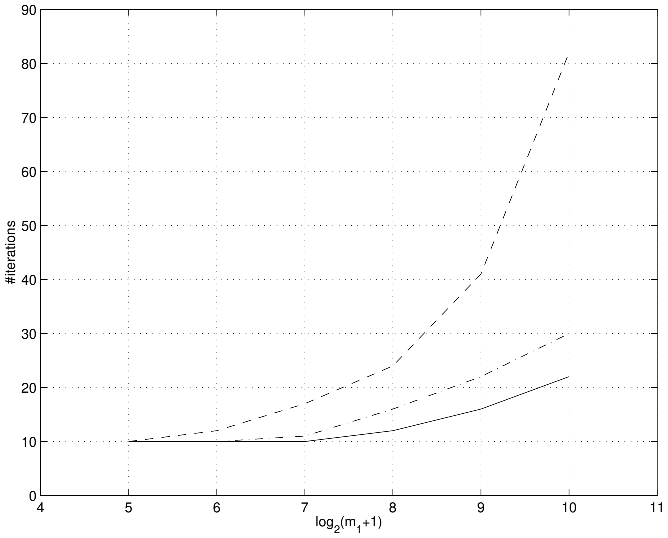

Figure 7.9: Number of iterations for $c_h = 0.1$ (solid line), $c_h = 0.2$ (dashed-dotted line) and $c_h = 0.5$ (dashed line), when $c_v = 1.0$, and $\phi = 1.0$.

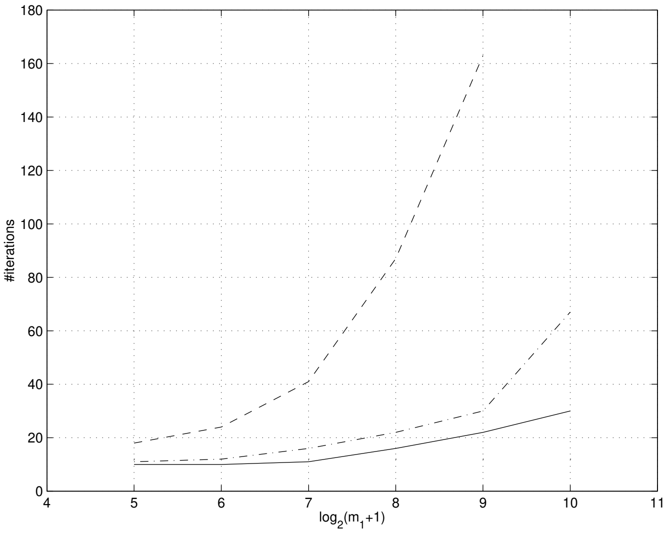

Figure 7.10: Number of iterations for $\phi = 1.0$ (solid line), $\phi = 2.0$ (dashed-dotted line) and $\phi = 10.0$ (dashed line), when $c_v = 1.0$, and $c_h = 0.2$.

Now, we will study the dependence on ϕ. Again, we will vary ϕ between 1.0 and 10.0. Note, that we were not able to fit the largest problem into the computer for $\phi = 10.0$.

The number of iterations depends strongly on the parameter ϕ. This is probably due to the fact that we solve a larger problem for larger ϕ:s, and from Table 7.4 and Figures 7.8 and 7.9 it is clear that the number of iterations increases with increasing problem-size.

Next, we will introduce the stretching of the grid in the x_2-direction defined in (7.1). In Figure 7.11 we display the number of iterations for the β_2:s shown in Figure 7.5.

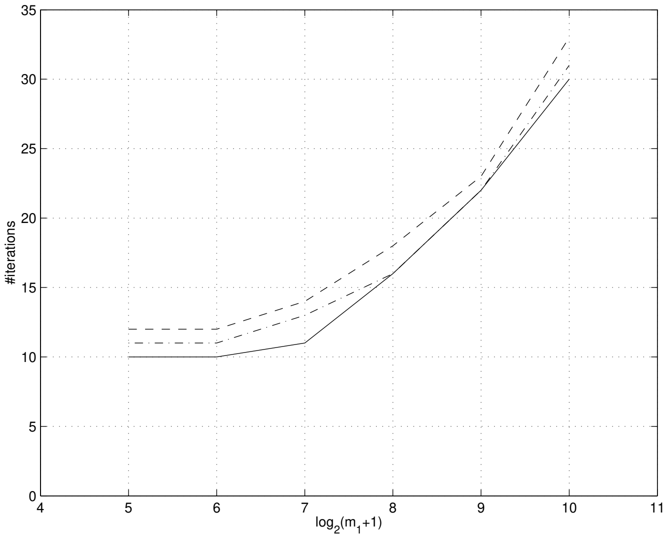

Figure 7.11: Number of iterations for $\beta_2 = 0.0$ (solid line), $\beta_2 = 1.5613$ (dashed-dotted line) and $\beta_2 = 5.0$ (dashed line), when $c_v = 1.0$, $c_h = 0.2$, and $\phi = 1.0$.

Again, a larger β_2 renders a slight increase in the number of iterations. Since we have such a small increase with increasing β_2, it is clear that we benefit from solving a smaller problem with stretching in the x_2-direction to resolve the boundary layer sufficiently compared to increasing the problem-size.

8 Conclusions

In this paper a scalar convection-diffusion equation is studied. Both a time-dependent and a stationary problem are considered. We have performed an analysis of a semi-Toeplitz preconditioner which reveals that the preconditioner has a number of promising features for the model problem considered. In particular we found analytical expressions for the spectra of the preconditioned systems. Asymptotically we find that these eigenvalues cluster around two values for the time-dependent problem and in one point and on a real line for the stationary problem. Both these properties are favorable when using Krylov subspace methods as the convergence of such methods is related to the spectrum of the problem. Furthermore estimates for the condition number reduction of the eigenvector-matrices of the preconditioned systems were found. It is shown that the condition number reduction increases as the problem-size increases, which is a good property.

The numerical experiments show that the preconditioner is efficient both in terms of iteration count and total runtime and shows a behaviour that comply with the theoretical analysis. Furthermore, it maintains this stable behaviour in cases where a moderately stretched net is used.

Due to the promising features of this preconditioner future work regarding the linearized Navier–Stokes equations is planned.

Acknowledgments

The authors wish to thank Dr. Sverker Holmgren and Prof. Per Lötstedt for valuable discussions.

References

[1] Roland W. Freund, Gene H. Golub, and Noël M. Nachtigal. Iterative solution of linear systems. *Acta Numerica*, 1:57–100, 1992.

[2] Lina Hemmingsson. A fast modified sine transform for solving block-tridiagonal systems with Toeplitz blocks. *Numer. Algorithms*, 7:375–389, 1994.

[3] Lina Hemmingsson and Kurt Otto. Analysis of semi-Toeplitz preconditioners for first-order PDEs. *SIAM J. Sci. Comput.*, 17:47–64, 1996.

[4] Sverker Holmgren, Henrik Brandén, and Erik Sterner. Convergence acceleration for the linearized Navier–Stokes equations using semicirculant approximations. *SIAM J. Sci. Comput.*, 21:1524–1550, 2000.

[5] Sang Dong Kim and Seymour V. Parter. Semi-circulant preconditioners of elliptic operators. *SIAM J. Numer. Anal.*, 2002. Submitted for publication.

[6] Youcef Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Math. Comp.*, 37:105–126, 1981.

[7] Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.

[8] G. D. Smith. *Numerical solution of partial differential equations, finite difference methods*. Clarendon Press, Oxford, 1978.

[9] Shigefumi Tatsumi, Luigi Martinelli, and Antony Jameson. Flux-limited schemes for the compressible Navier–Stokes equations. *AIAA J.*, 33:252–261, 1995.