Spectral schemes on triangular elements

by Wilhelm Heinrichs and Birgit I. Loch

Abstract

The Poisson problem with homogeneous Dirichlet boundary conditions is considered on a triangle. The mapping between square and triangle is realized by mapping an edge of the square onto a corner of the triangle. Then standard Chebyshev collocation techniques can be applied. Numerical experiments demonstrate the expected high spectral accuracy. Further, it is shown that finite difference preconditioning can be successfully applied in order to construct an efficient iterative solver. Then the convection-diffusion equation is considered. Here finite difference preconditioning with central differences does not overcome instability. However, applying the first order upstream scheme, we obtain a stable method. Finally, a domain decomposition technique is applied to the patching of rectangular and triangular elements.

Keywords

spectral, collocation, triangle, preconditioning, Poisson, convection-diffusion, domain decomposition.

Introduction

Pseudospectral collocation methods give good approximations to smooth solutions of elliptic partial differential equations. However, there is a huge disadvantage as these methods are confined to rectangles. Additionally, the spectral operator is ill conditioned compared to finite difference or finite element operators and requires preconditioning to construct an effective iterative solver. Here, we apply the standard Chebyshev collocation method for solving partial differential equations on certain right triangles. We introduce a transformation between the triangle and the standard square where spectral collocation can be applied. This transformation maps one edge of the square onto one corner of the triangle so that the non-equally spaced collocation points cluster in that corner. In [6] a different approach has been examined. The results are compared. This method is then applied to the Poisson equation with homogeneous Dirichlet boundary conditions on a right triangle. It is numerically shown that for smooth solutions high spectral accuracy can be achieved. Then we introduce a singularity caused by the singular behaviour of the right-hand side leading to a somewhat slower convergence of the approximation. Preconditioning by finite differences yields a condition number increasing as O(N).
After that the convection-diffusion equation is considered. To overcome the instability for small ϵ we choose N to be odd (see [1]). Preconditioning by central finite differences yields an unbounded condition number such that an upwind method has to be applied.
Finally, domain decomposition problems are investigated. The Poisson problem is numerically solved on patchings of rectangular and triangular elements. A Dirichlet Neumann interface relaxation is iterated until continuity of normal derivatives is achieved. By numerical results the efficiency of this treatment is demonstrated.

Transformation of the right triangle

The standard Chebyshev collocation scheme (see [6]) is defined for the non-equally spaced Chebyshev-Gauss-Lobatto nodes $(s_i, t_j) = (\cos\frac{i\pi}{N}, \cos\frac{j\pi}{N})$ on the square $[-1,1]^2$. Using linear transforms, arbitrary rectangles can be considered. However, if we are interested in triangular domains the mapping is more complicated. In [6] a mapping applying polar coordinate transformation and bending of an edge of the triangle was introduced and analyzed. Numerical results showed the effectiveness of this method. Here we consider a new transformation between the standard square $R = \{(x,y) \mid -1 < x, y < 1\}$ and the right triangle $T = \{(x,y) \mid 0 < x, y < 1 \text{ and } x + y < 1\}$. The original mapping is given in [7] and has been changed for our purposes. The transformation reads

$$\begin{aligned} x &= \tfrac{1}{4}(x_R+1)(1-y_R), \quad y = \tfrac{1}{2}(y_R+1) \\ x_R &= \tfrac{2x}{1-y} - 1, \qquad\qquad y_R = 2y - 1 \end{aligned}$$

and is displayed in Figure 1.

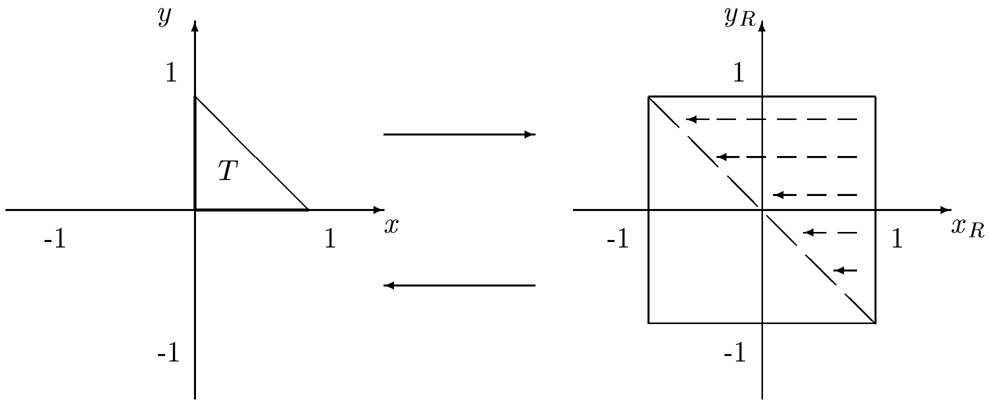

Figure 1: Horizontal transformation

We will call this the horizontal transform as every node is actually moved horizontally. The vertical transform is

$$\begin{aligned} x &= \tfrac{1}{2}(x_R+1), \quad y = \tfrac{1}{4}(y_R+1)(1-x_R) \\ x_R &= 2x - 1, \qquad y_R = \tfrac{2y}{1-x} - 1 \end{aligned}$$

and will be considered later.
This transformation is no longer injective. We will see that this does not disturb the accuracy of our approximation. The upper edge of R is mapped onto P(0,1) on T. As this edge belongs to the border of our domain boundary conditions are applicable which are treated separately anyway.

Partial derivatives must be transformed, too. Using the horizontal transform we derive

$$\begin{aligned}
u_x &= 2u_{x_1} &&= \tfrac{4}{1-y_R}u_{xR} \\
u_{xx} &= 4u_{x_1x_1} &&= \tfrac{16}{(1-y_R)^2}u_{xRxR} \\
u_y &= 2u_{y_1} &&= 2\tfrac{x_R+1}{1-y_R}u_{xR} + 2u_{yR} \\
u_{yy} &= 4u_{y_1y_1} &&= 4\tfrac{(x_R+1)^2}{(1-y_R)^2}u_{xRxR} + 8\tfrac{x_R+1}{1-y_R}u_{xRyR} + 8\tfrac{x_R+1}{(1-y_R)^2}u_{xR} + 4u_{yRyR}.
\end{aligned}$$

The Laplacian then reads as follows

$$\begin{aligned}
\Delta u &= u_{xx} + u_{yy} \\
&= 4\tfrac{4+(x_R+1)^2}{(1-y_R)^2}u_{xRxR} + 8\tfrac{x_R+1}{1-y_R}u_{xRyR} + 8\tfrac{x_R+1}{(1-y_R)^2}u_{xR} + 4u_{yRyR}.
\end{aligned}$$

The Poisson problem

Numerous spectral algorithms for the numerical simulation of physical phenomena demand the approximative solution of one or more Poisson problems in a bounded domain.
We now study the problem

$$\begin{aligned}
\Delta u &= f \text{ in } T, \\
u &= 0 \text{ on } \partial T,
\end{aligned}$$

where ∂T denotes the boundary of T. We apply the standard Chebyshev collocation scheme to the exact solution

$$u(x,y) = xy(e^{x+y} - e). \tag{1}$$

This function obviously fulfills the boundary condition.
Table 1 shows the discrete L_2 error $E_2 := \frac{\|u-u_N\|_2}{N}$. One observes the exponential decay of the error.

N	E_2	E_2 in [6]
4	$1.94 \cdot 10^{-5}$	$1.89 \cdot 10^{-4}$
8	$2.04 \cdot 10^{-11}$	$8.85 \cdot 10^{-7}$
16	$2.12 \cdot 10^{-16}$	$1.84 \cdot 10^{-11}$
32	$4.29 \cdot 10^{-16}$	$1.78 \cdot 10^{-16}$

Table 1: Error using horizontal transformation and [6]

As we see the high spectral accuracy can also be reached on the triangle T. We have the best approximation of the solution at P(0,1) as the collocation points cluster there. Figure 2 shows the position of the collocation points for N=16 on the triangle and on the square.

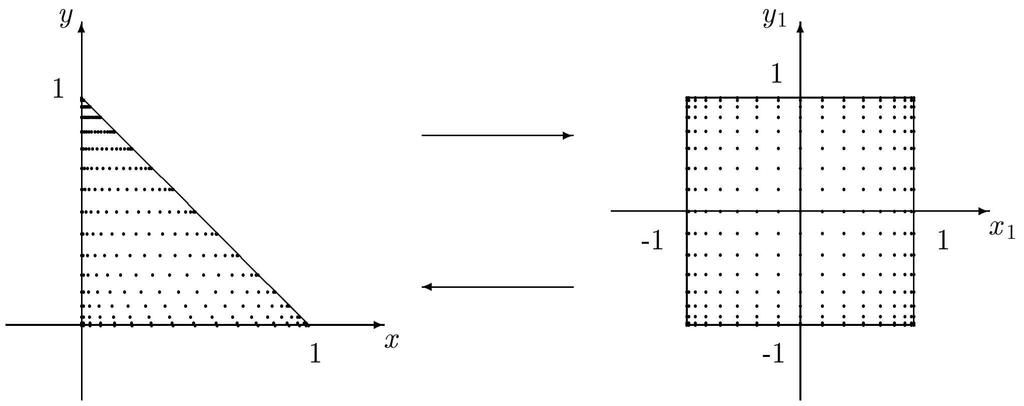

Figure 2: Positions of the Chebyshev collocation nodes for $N = 16$

Comparison of these results to those in [6] using polar coordinate transformation (see Table 1) shows that our mapping yields a faster convergence of the approximation. Here rounding error accuracy is already reached for N=16. N=4 and N=8 give results which are more exact by 1 or 5 digits. This can be explained by the position and way of numbering the collocation nodes. Figure 3 shows that the jumps occuring when changing the row (e.g. from third to fourth point) are decreasing while those in [6] seem to be larger. The speed of convergence is probably influenced by greater jumps.

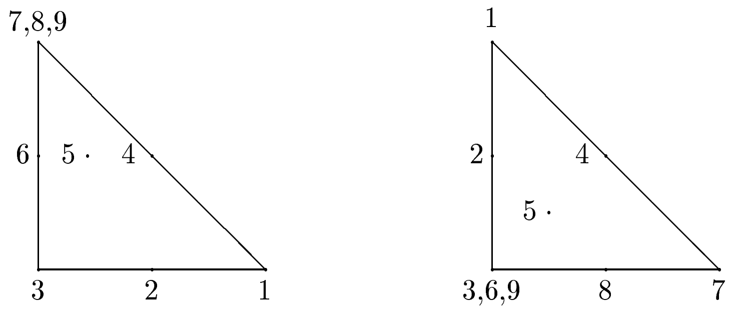

Figure 3: Order of collocation points for $N = 2$ compared to [6]

Next we consider a singular problem where $f \equiv -1$. We compare the results for N=4, 8, 16 and 32 to those obtained for N=36 at the fixed points displayed in Figure 4. These points are the collocation nodes for N=4 which are also used for larger N divisible by 4. We expect the error to be smallest close to y=1 because there the collocation nodes cluster. We deal with the following nodes:
$P_1(0,0)$, $P_2(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$, $P_3(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$, $P_4(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ and $P_5(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$.

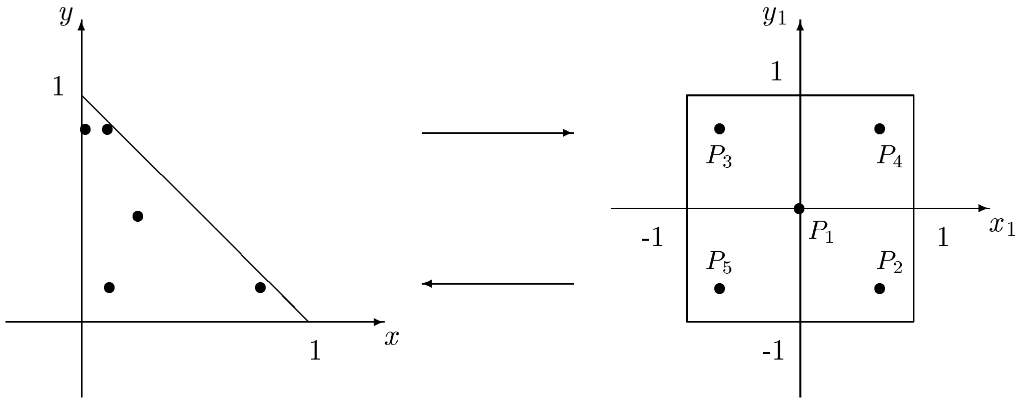

Figure 4: Positions of the five nodes

The approximation converges more slowly than in the last example. That makes sense because here the differential equation and its boundary condition are not compatible any more.
To get an overview we present $ER = |u_N - u_{36}|$ which is the absolute value of the difference for every node, in a diagram (Figure 5).

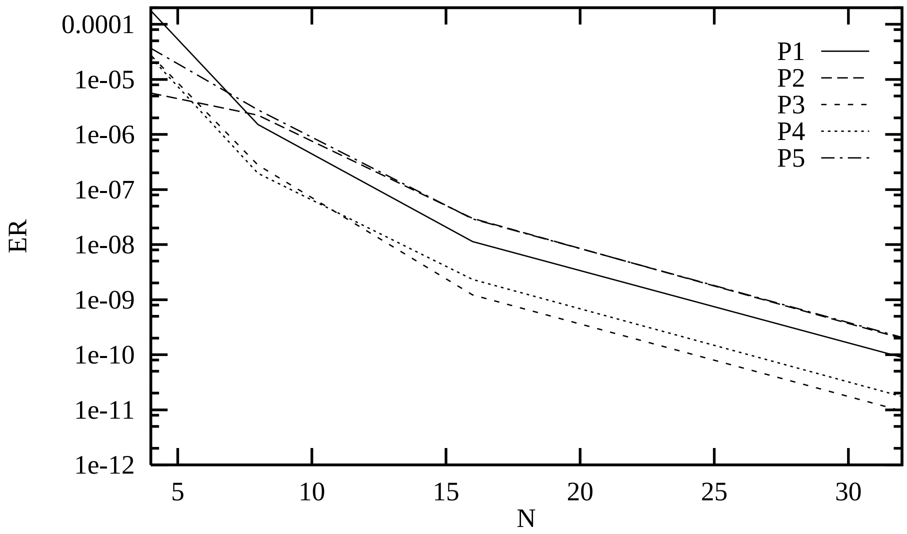

Figure 5: Poisson problem with constant f

Next we choose f discontinuous:

$$f(x,y) = \begin{cases} -1 & \text{for } y - x > 0 \\ 0 & \text{for } y - x \leq 0. \end{cases}$$

As Figure 6 shows the triangle is now bisected. The transformation of the line $y = x$ on the triangle gives the hyperbola $y = 2\frac{x+1}{x+3} - 1$ on the square.

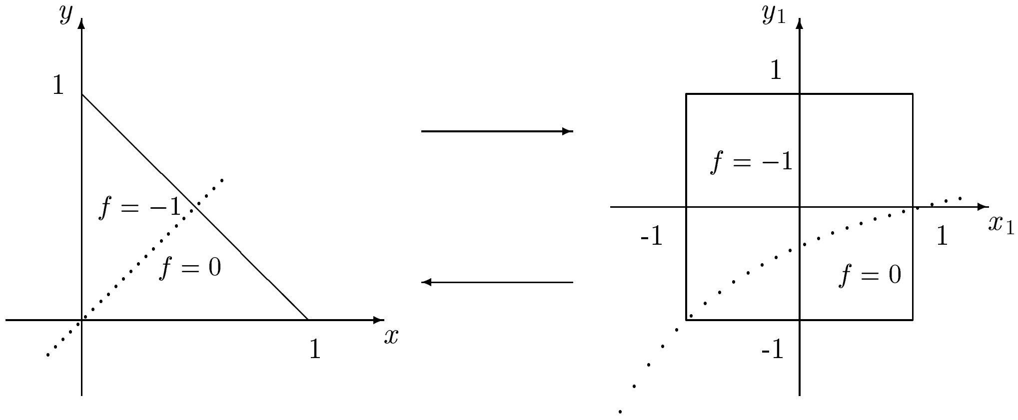

Figure 6: Transformation of the line

The results can be found in Figure 7.

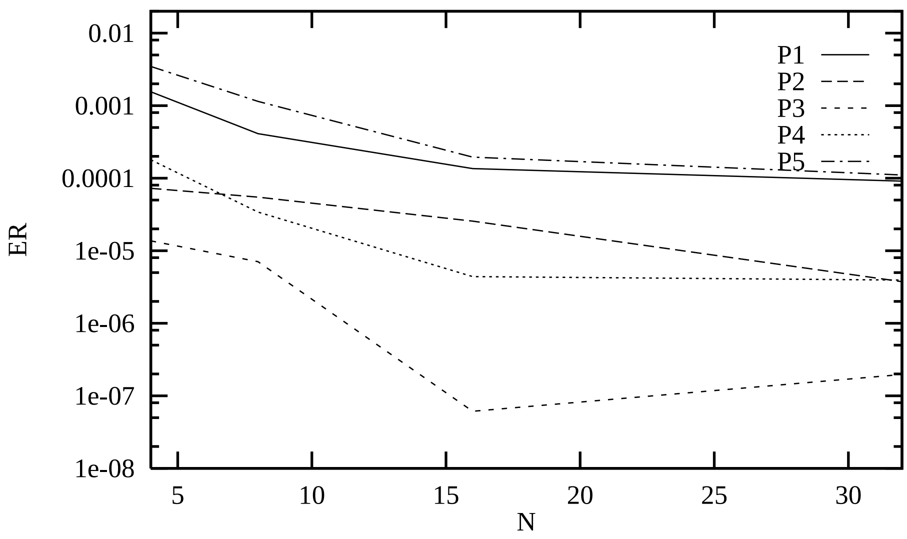

Figure 7: Poisson problem with discontinuous f

The approximation is relatively bad close to the separating line. Since f is discontinuous the solution of the partial differential equation is no longer smooth and there is no high spectral accuracy any more. We only have a first order method.

Preconditioning

We are interested in a good condition number of our spectral operator which does not increase too fast such that efficient iterative solvers can be found. Here the maximum eigenvalues of the spectral Laplacian on the triangle scale as $O(N^8)$ (Table 2). On the square one has $O(N^4)$ which is certainly preferrable. We are looking for a preconditioner to improve the condition so that it scales as $O(N)$ or even independently of N. A good preconditioner also has to be a good approximation of the inverse of the spectral operator. We found that the condition number is already

reduced if we multiply the operator by $(1-y_R)^2$. The partial derivatives contain this factor in the denominator. For y close to 1 the influence of the appropriate partial derivative is extremely high. The discretized operator is called $L_{2,SP}$. Table 3 shows $\lambda_{max} := max\{|\lambda| \mid \lambda\ eigenvalue\}$, $\lambda_{min} := min\{|\lambda| \mid \lambda\ eigenvalue\}$ and $cond \approx \frac{\lambda_{max}}{\lambda_{min}}$.
Here the condition number scales as $O(N^4)$.

N	λ_{max}	λ_{min}	$cond$	λ_{max}/N^8
4	$5.39 \cdot 10^3$	$5.36 \cdot 10^1$	$1.01 \cdot 10^2$	0.08
8	$1.10 \cdot 10^6$	$4.94 \cdot 10^1$	$2.23 \cdot 10^4$	0.07
16	$2.71 \cdot 10^8$	$4.93 \cdot 10^1$	$5.49 \cdot 10^6$	0.06
32	$6.86 \cdot 10^{10}$	$4.93 \cdot 10^1$	$1.39 \cdot 10^9$	0.06

Table 2: The spectral operator L_{SP}

N	λ_{max}	λ_{min}	$cond$	λ_{max}	λ_{min}	$cond$
4	$6.36 \cdot 10^2$	$5.41 \cdot 10^1$	$1.18 \cdot 10^1$	$1.15 \cdot 10^2$	5.10	$2.26 \cdot 10^1$
8	$9.77 \cdot 10^3$	$5.35 \cdot 10^1$	$1.83 \cdot 10^2$	$1.89 \cdot 10^3$	4.58	$4.15 \cdot 10^2$
16	$1.56 \cdot 10^5$	$5.32 \cdot 10^1$	$2.94 \cdot 10^3$	$3.07 \cdot 10^4$	4.39	$7.00 \cdot 10^3$
32	$2.50 \cdot 10^6$	$5.30 \cdot 10^1$	$4.71 \cdot 10^4$	$4.93 \cdot 10^5$	4.29	$1.15 \cdot 10^5$

Table 3: The spectral operator $L_{2,SP}$ and results in [6]

Our results are comparable to those in [6].
We now study the finite difference preconditioner L_{FD} which is the discretization of the Laplacian by second order finite differences. The first and second derivatives are

$$\begin{aligned} w'(s_j) &= \frac{1}{2}(-\gamma_{j-1}w(s_{j-1}) - (\gamma_j - \gamma_{j-1})w(s_j) + \gamma_j w(s_{j+1})), \\ w''(s_j) &= 2\delta_j(\gamma_{j-1}w(s_{j-1}) - (\gamma_j + \gamma_{j-1})w(s_j) + \gamma_j w(s_{j+1})) \end{aligned}$$

where

$$\begin{aligned} \delta_j &= \frac{1}{s_{j+1} - s_{j-1}}, \\ \gamma_j &= \frac{1}{s_{j+1} - s_j} \quad \text{for } j = 1, \ldots, N-1 \text{ (see [6]).} \end{aligned}$$

Table 4 shows the improved results.

N	λ_{max}	λ_{min}	$cond$	λ_{max}	λ_{min}	$cond$
4	1.73	1.00	1.73	1.71	0.99	1.73
8	2.13	0.89	2.41	2.12	0.99	2.13
16	2.50	0.71	3.53	2.41	0.80	3.01
32	2.91	0.60	4.89	2.83	0.66	4.31

Table 4: $(L_{FD})^{-1}L_{SP}$ and results in [6]

Now we obtained a condition number scaling as $O(N)$. We could construct an effective iterative solver now.
Figure 8 shows the positions of the eigenvalues for N=32. Their imaginary parts are fairly small and the real parts are contained in $[0.5, 3]$.

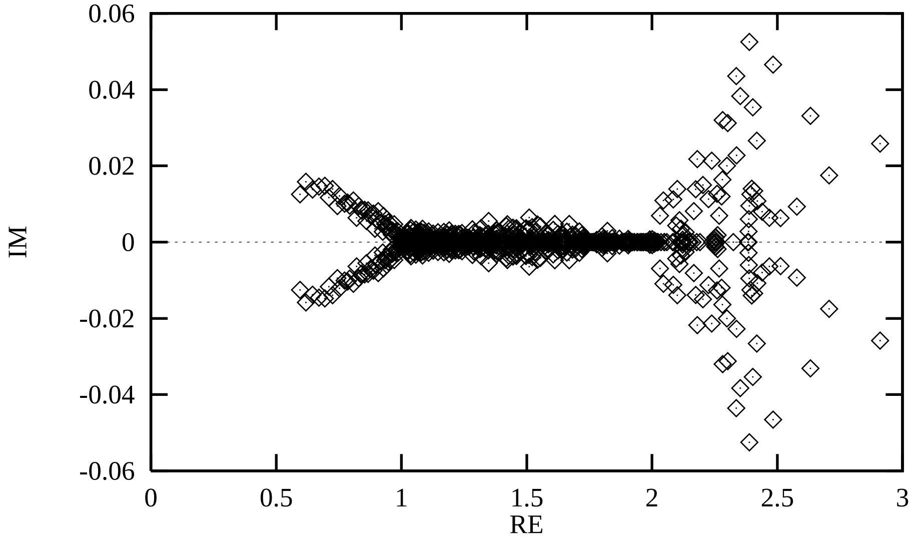

Figure 8: Eigenvalues of $(L_{FD})^{-1} L_{SP}$ for $N = 32$

One could apply higher order FD-methods for an even better condition number. However, this would result in an extended effort for solving the FD problem.
In summing up, we state that this transformation between triangle and square gives comparable or better results than the transformation by polar coordinates in [6].

The convection-diffusion equation

Modelling of purely convectional or convection dominated processes is a central problem in areas like e.g. meteorology or investigation of aerodynamical or geophysical flows. A model boundary value problem is the convection-diffusion equation

$$\begin{aligned} -\epsilon\Delta u + au_x + bu_y &= f \quad \text{in } T, \\ u &= 0 \quad \text{on } \partial T, \end{aligned}$$

which can be used for describing the expansion of temperature in a fluent. Temperature expands uniformly diffusive in every direction which is expressed by $-\epsilon\Delta u$. It is spread by current, too, called convection and is described by $au_x + bu_y$ (a and b being the velocities in x- and in y-direction).

As usual, ϵ is the viscosity of our material and represents a measure for interior friction. As the partial differential equation is of different type for $\epsilon > 0$ and $\epsilon = 0$ (in the first case it is elliptic and in the latter it is hyperbolic) we talk about singular behaviour. In the interior of our domain u_ϵ and u_0 are close together but getting to the boundary they differ extremely.

Homogeneous Dirichlet boundary conditions are not applicable to hyperbolic problems such that we have to deal with boundary layers now. Boundary layers are environments where derivatives of u_ϵ scale as $O(\frac{1}{\epsilon})$. Those systems are also called stiff systems. Unphysical oscillations occur in the numerical solution and the discretization is instable. Figure 9 shows the situation in 1D (see [4]).

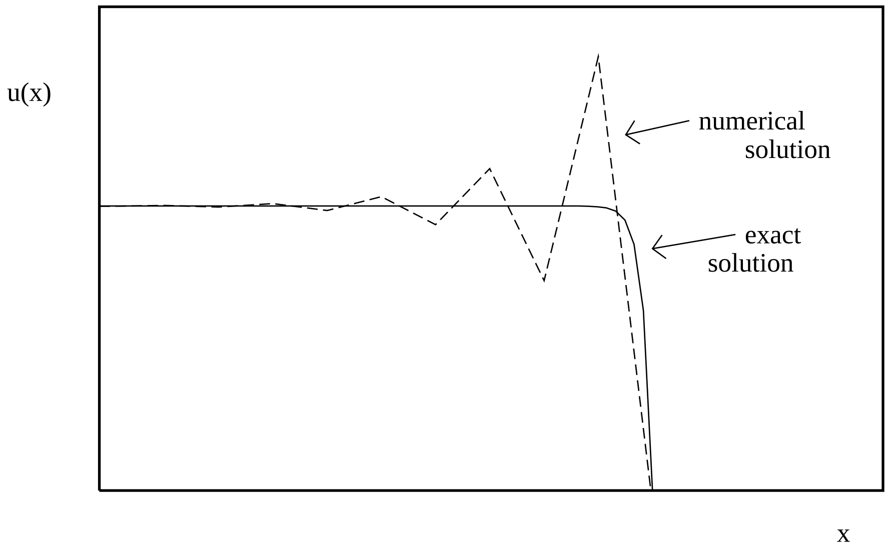

Figure 9: Boundary layer

We are looking for a method to resolve the boundary layers. There are schemes which still use spectral methods like adding artificial viscosity, spectral viscosity or streamline diffusion. However, here we only choose odd N. Oscillation always arises and is increased for even N with $\epsilon \ll N^{-2}$ while this is not the case if N is odd.

The following table contains the discrete L_2 error for decreasing ϵ which develops when discretizing the convection-diffusion equation by spectral collocation. Here we choose (a,b)=(1,1)

and (-1,1) as these two cases are good representatives for other choices of (a,b). We have tested the algorithm with example (1). In the case of pure convection ($\epsilon = 0$) the method is unstable. With decreasing ϵ the singular behaviour is increasing and one has to choose a finer grid (larger N) to obtain results comparable to $\epsilon = 1$.
As mentioned above, we now choose N odd which usually leads to a decreased error. This behaviour was analyzed in [1] in 1D on the square. It can be transferred to the triangle with only few restrictions concerning the choice of parameters. If N is even there exists an interpolation polynomial which fulfills the boundary conditions and whose derivative vanishes at the collocation points. This polynomial is responsible for the instability. On the contrary, if N is odd one finds the proof in [1] that this polynomial does not exist. Apparently, there are parameters (a,b) for which the spectral method is unstable even for odd N. For the stable case (1,0) we actually have the regular operator $\frac{\partial}{\partial x}$ on the square multiplied with a factor. For (-1,1) we have exactly that combination of the first derivatives on the square where there are at least two equal rows in the derivative matrix. The partial derivatives are based on the matrix D_N. As the collocation points on the square are symmetric (for every positive node we find a corresponding negative one) there is annulation in the derivative matrix. The following example for N=3 shows the connection.
$u_y = 2\frac{x_R+1}{1-y_R}u_{x_R} + 2u_{y_R}$ yields the derivative matrix

$$\begin{pmatrix} \frac{-s_1}{(1-s_1)^2} - \frac{s_1}{(1-s_1^2)} & \frac{-2(s_1+1)}{(1-s_1)(s_1-s_2)} & \frac{-2}{s_1-s_2} & 0 \\ \frac{-2(s_2+1)}{(1-s_1)(s_2-s_1)} & \frac{-s_2}{(1-s_1)(1-s_2)} - \frac{s_1}{1-s_1^2} & 0 & \frac{-2}{s_1-s_2} \\ \frac{-2}{s_2-s_1} & 0 & \frac{-s_1}{(1-s_2)(1-s_1)} - \frac{s_2}{1-s_2^2} & \frac{-2(s_1+1)}{(1-s_2)(s_1-s_2)} \\ 0 & \frac{-2}{s_2-s_1} & \frac{-2(s_2+1)}{(1-s_2)(s_2-s_1)} & \frac{-s_2}{(1-s_2)^2} - \frac{s_2}{1-s_2^2} \end{pmatrix}$$

As $s_1 = -s_2$ (symmetry) we have equal second and third row and the matrix is singular. (-1,1) shows the same behaviour.
For (1,1) we do not have annulations and the method is stable. Table 5 displays the results. (0,1) can be stabilized by using the vertical transformation where x and y are exchanged.

		E2 for				
(a,b)	N	$\epsilon = 1$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-6}$	$\epsilon = 0$
$(1,1)$	3	$2.75 \cdot 10^{-4}$	$2.21 \cdot 10^{-3}$	$2.15 \cdot 10^{-3}$	$2.15 \cdot 10^{-3}$	$2.15 \cdot 10^{-3}$
	7	$8.75 \cdot 10^{-10}$	$4.08 \cdot 10^{-9}$	$7.77 \cdot 10^{-9}$	$7.77 \cdot 10^{-9}$	$7.77 \cdot 10^{-9}$
	15	$3.77 \cdot 10^{-16}$	$5.32 \cdot 10^{-17}$	$1.81 \cdot 10^{-16}$	$1.50 \cdot 10^{-16}$	$1.14 \cdot 10^{-16}$
	31	$5.08 \cdot 10^{-16}$	$1.72 \cdot 10^{-16}$	$2.27 \cdot 10^{-16}$	$5.64 \cdot 10^{-16}$	$3.67 \cdot 10^{-16}$
$(-1,1)$	3	$2.56 \cdot 10^{-4}$	$3.75 \cdot 10^{-3}$	$1.69 \cdot 10^{-1}$	$1.68 \cdot 10^{1}$	$1.09 \cdot 10^{13}$
	7	$8.76 \cdot 10^{-10}$	$5.16 \cdot 10^{-9}$	$1.86 \cdot 10^{-7}$	$1.86 \cdot 10^{-5}$	$1.59 \cdot 10^{8}$
	15	$1.06 \cdot 10^{-16}$	$7.53 \cdot 10^{-17}$	$1.86 \cdot 10^{-16}$	$2.70 \cdot 10^{-14}$	$5.75 \cdot 10^{-1}$
	31	$4.25 \cdot 10^{-16}$	$2.27 \cdot 10^{-16}$	$4.74 \cdot 10^{-16}$	$2.90 \cdot 10^{-14}$	$4.36 \cdot 10^{0}$

Table 5: Error for the convection-diffusion equation

Next a constant right-hand side is considered. Differential equation and boundary condition are not compatible here, i.e.

$$-\epsilon\Delta u + au_x + bu_y \quad = \quad 1 \quad \text{in } T,$$

$$u = 0 \text{ on } \partial T.$$

Table 6 shows the difference ER between u_{36} and u_N at P1(0,0). P1(0,0) is in the center of the triangle and therefore far away from any boundary. It is the only collocation point (out of P1-P5) where stability is achieved for (1,1) for small ϵ.

		ER for				
(a,b)	N	$\epsilon = 1$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-6}$	$\epsilon = 0$
$(1,1)$	4	$1.34 \cdot 10^{-4}$	$1.99 \cdot 10^{-1}$	$9.53 \cdot 10^{-2}$	$2.72 \cdot 10^{-2}$	$4.11 \cdot 10^{-1}$
	8	$1.86 \cdot 10^{-6}$	$5.99 \cdot 10^{-2}$	$7.82 \cdot 10^{-2}$	$4.83 \cdot 10^{-3}$	$1.14 \cdot 10^{-1}$
	16	$1.12 \cdot 10^{-8}$	$1.24 \cdot 10^{-3}$	$6.20 \cdot 10^{-2}$	$7.03 \cdot 10^{-3}$	$8.06 \cdot 10^{-1}$
	32	$8.91 \cdot 10^{-11}$	$1.68 \cdot 10^{-7}$	$1.26 \cdot 10^{-2}$	$1.58 \cdot 10^{-3}$	$2.89 \cdot 10^{-2}$
$(-1,1)$	4	$6.80 \cdot 10^{-5}$	$2.20 \cdot 10^{-1}$	$2.34 \cdot 10^{1}$	$2.35 \cdot 10^{3}$	$1.79 \cdot 10^{15}$
	8	$1.56 \cdot 10^{-6}$	$6.71 \cdot 10^{-2}$	$1.43 \cdot 10^{0}$	$1.41 \cdot 10^{2}$	$1.27 \cdot 10^{15}$
	16	$1.11 \cdot 10^{-8}$	$2.10 \cdot 10^{-3}$	$1.75 \cdot 10^{-3}$	$2.45 \cdot 10^{1}$	$1.27 \cdot 10^{15}$
	32	$8.88 \cdot 10^{-11}$	$1.31 \cdot 10^{-5}$	$6.21 \cdot 10^{-2}$	$2.17 \cdot 10^{0}$	$2.06 \cdot 10^{15}$

Table 6: Error for constant f in P1

A discontinuous right-hand side

$$f(x,y) = \begin{cases} -1 & \text{for } y - x > 0 \\ 0 & \text{for } y - x \leq 0 \end{cases}$$

yields an even slower convergence rate than the last example (Table 7).

		ER for				
(a,b)	N	$\epsilon = 1$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-6}$	$\epsilon = 0$
$(1,1)$	4	$1.53 \cdot 10^{-3}$	$1.68 \cdot 10^{-1}$	$1.08 \cdot 10^{-1}$	$2.00 \cdot 10^{-2}$	$3.69 \cdot 10^{-1}$
	8	$4.29 \cdot 10^{-4}$	$2.00 \cdot 10^{-2}$	$7.42 \cdot 10^{-2}$	$1.66 \cdot 10^{-2}$	$8.56 \cdot 10^{-2}$
	16	$1.36 \cdot 10^{-4}$	$3.64 \cdot 10^{-4}$	$4.14 \cdot 10^{-2}$	$2.37 \cdot 10^{-2}$	$8.89 \cdot 10^{-1}$
	32	$9.35 \cdot 10^{-5}$	$5.88 \cdot 10^{-4}$	$1.89 \cdot 10^{-2}$	$1.73 \cdot 10^{-3}$	$3.08 \cdot 10^{-2}$
$(-1,1)$	4	$1.70 \cdot 10^{-3}$	$1.39 \cdot 10^{-1}$	$5.46 \cdot 10^{0}$	$5.48 \cdot 10^{2}$	$3.43 \cdot 10^{14}$
	8	$4.94 \cdot 10^{-4}$	$6.86 \cdot 10^{-2}$	$1.93 \cdot 10^{0}$	$1.96 \cdot 10^{2}$	$9.23 \cdot 10^{14}$
	16	$1.53 \cdot 10^{-4}$	$4.65 \cdot 10^{-3}$	$2.35 \cdot 10^{-1}$	$3.61 \cdot 10^{1}$	$5.63 \cdot 10^{14}$
	32	$1.02 \cdot 10^{-4}$	$1.79 \cdot 10^{-3}$	$3.64 \cdot 10^{-2}$	$3.08 \cdot 10^{-1}$	$8.96 \cdot 10^{14}$

Table 7: Error for discontinuous f in P1

Preconditioning

For the construction of an effective iterative solver we now examine the condition number of the spectral operator $L_{2,\epsilon}$ of $(1 - y_R)^2(-\epsilon\Delta + au_x + bu_y)$.

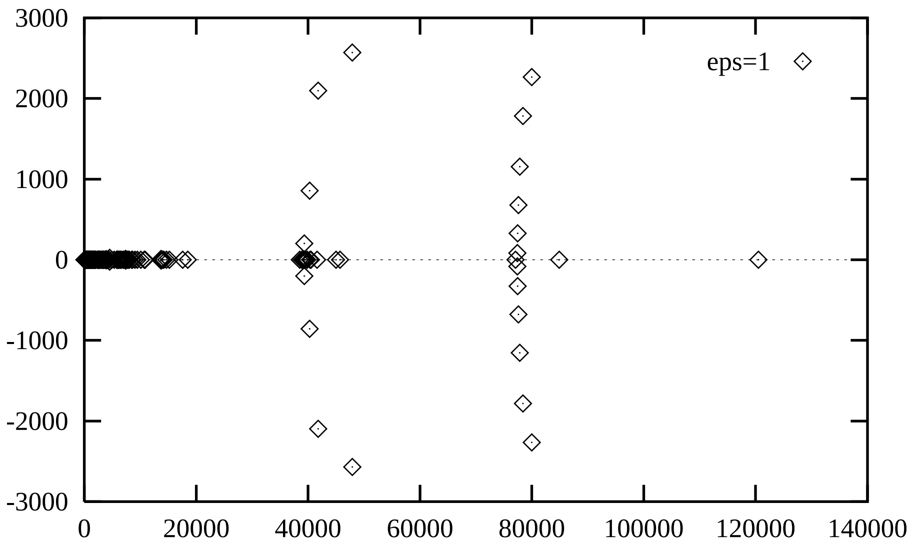

Figure 10: Eigenvalues of $L^{\epsilon}_{2,SP}$ for N=15, (a,b)=(1,1)

Figures 10 and 11 show the positions of the eigenvalues for $\epsilon = 1$ and $\epsilon = 10^{-6}$ for (a,b)=(1,1), N=15.

		$\epsilon\ = 1$			$\epsilon\ = 0$		
(a,b)	N	λ_{max}	λ_{min}	$cond$	λ_{max}	λ_{min}	$cond$
$(1,1)$	3	$2.20 \cdot 10^{2}$	$5.91 \cdot 10^{1}$	$3.72 \cdot 10^{0}$	$8.76 \cdot 10^{0}$	$2.43 \cdot 10^{0}$	$3.60 \cdot 10^{0}$
	7	$5.71 \cdot 10^{3}$	$5.36 \cdot 10^{1}$	$1.06 \cdot 10^{2}$	$8.63 \cdot 10^{1}$	$7.02 \cdot 10^{-1}$	$1.23 \cdot 10^{2}$
	15	$1.20 \cdot 10^{5}$	$5.32 \cdot 10^{1}$	$2.26 \cdot 10^{3}$	$4.43 \cdot 10^{2}$	$1.73 \cdot 10^{-2}$	$2.57 \cdot 10^{3}$
	31	$2.20 \cdot 10^{6}$	$5.30 \cdot 10^{1}$	$4.15 \cdot 10^{4}$	$1.96 \cdot 10^{3}$	$4.24 \cdot 10^{-2}$	$4.62 \cdot 10^{4}$
$(-1,1)$	3	$2.22 \cdot 10^{2}$	$5.82 \cdot 10^{1}$	$3.82 \cdot 10^{0}$	$3.27 \cdot 10^{0}$	$0.00 \cdot 10^{0}$	
	7	$5.75 \cdot 10^{3}$	$5.36 \cdot 10^{1}$	$1.07 \cdot 10^{2}$	$5.03 \cdot 10^{1}$	$2.60 \cdot 10^{-16}$	$1.93 \cdot 10^{17}$
	15	$1.21 \cdot 10^{5}$	$5.32 \cdot 10^{1}$	$2.27 \cdot 10^{3}$	$2.86 \cdot 10^{2}$	$6.61 \cdot 10^{-16}$	$4.33 \cdot 10^{17}$
	31	$2.20 \cdot 10^{6}$	$5.30 \cdot 10^{1}$	$4.15 \cdot 10^{4}$	$1.29 \cdot 10^{3}$	$6.52 \cdot 10^{-16}$	$1.98 \cdot 10^{18}$

Table 8: $L^{\epsilon}_{2,SP}$

Table 8 gives $\lambda_{max}, \lambda_{min}$ and *cond* and demonstrates that there really is an eigenvalue close to 0 for (-1,1).
Applying the inverse of the FD operator L^{ϵ}_{FD} as preconditioner, we observe decreased condition number if $\epsilon = 1$ while for small ϵ, λ_{max} is unbounded for (1,1). This preconditioner obviously does not stabilize.
Figures 12 and 13 show the positions of the eigenvalues. For small ϵ they are relatively dense positioned with few peak values.
In general, FD methods applied to singular disturbance problems are stable if the step size $h_i < 2\epsilon$. Contrary, if $h_i \gg \epsilon$ they are unstable. To obtain stability one could increase the number of collocation points which reduces the step size. A more promising attempt is the use

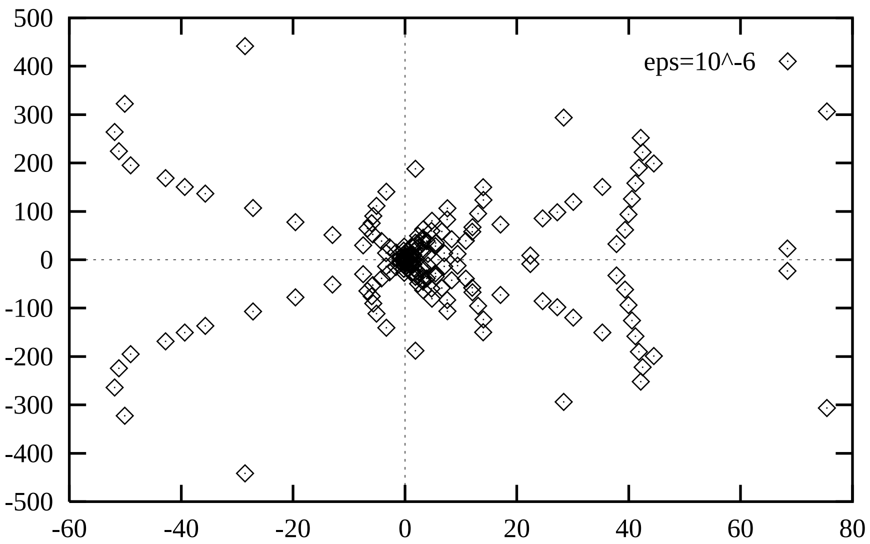

Figure 11: Eigenvalues of $L^{\epsilon}_{2,SP}$ for N=15, (a,b)=(1,1)

of the upwind method. The first derivatives $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$, the convectional part, is discretized by one-sided stream-directed finite differences while the diffusive part is treated with central differences. We lose one order in convergence but stability is achieved.
We have

$$\begin{aligned} a \cdot u_x &= a \cdot \frac{4}{1-y_R} \cdot u_{x_R} \text{ and} \\ b \cdot u_y &= b \cdot (2\frac{x_R+1}{1-y_R} \cdot u_{x_R} + 2u_{y_R}). \end{aligned}$$

According to the factor the derivatives u_{x_R} and u_{y_R} are discretized by left- or right-differences in stream direction:

$$u_{x_R}(x_i, y_j) \cong \begin{cases} \frac{u(x_{i+1},y_j)-u(x_i,y_j)}{x_{i+1}-x_i} & \text{if a } \geq 0 \\ \frac{u(x_i,y_j)-u(x_{i-1},y_j)}{x_i-x_{i-1}} & \text{if a } < 0 \end{cases}$$

for $i = 0, \ldots, N-1$ or $i = 1, \ldots, N$. Analogously for u_{y_R}. The upwind method is not uniformly convergent. An adaptive refinement might help here.
Figures 14 and 15 show that by applying the upstream method the positions of the eigenvalues have completely changed for small ϵ. They are complex, bounded and symmetric. Table 9 gives the numerical results for the upstream scheme.

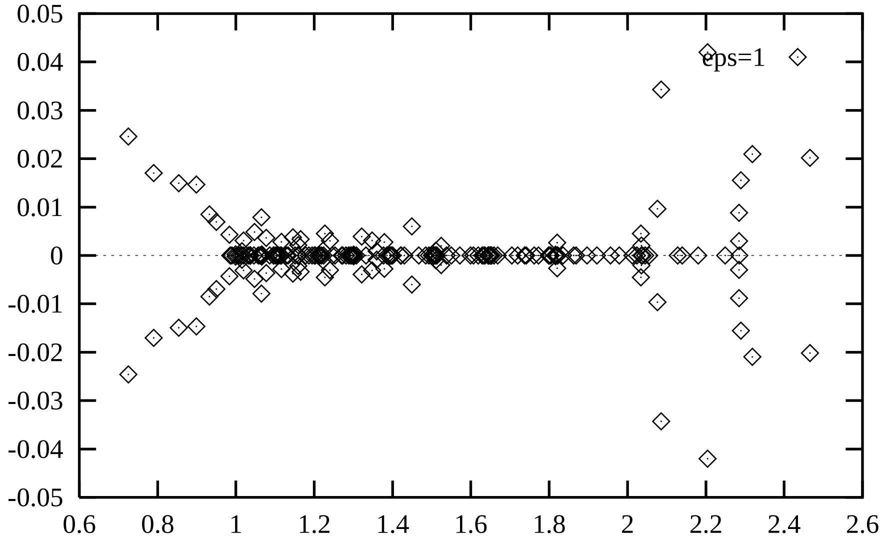

Figure 12: Eigenvalues of $L_{FD}^{-1}L_{SP}$ for N=15, (a,b)=(1,1)

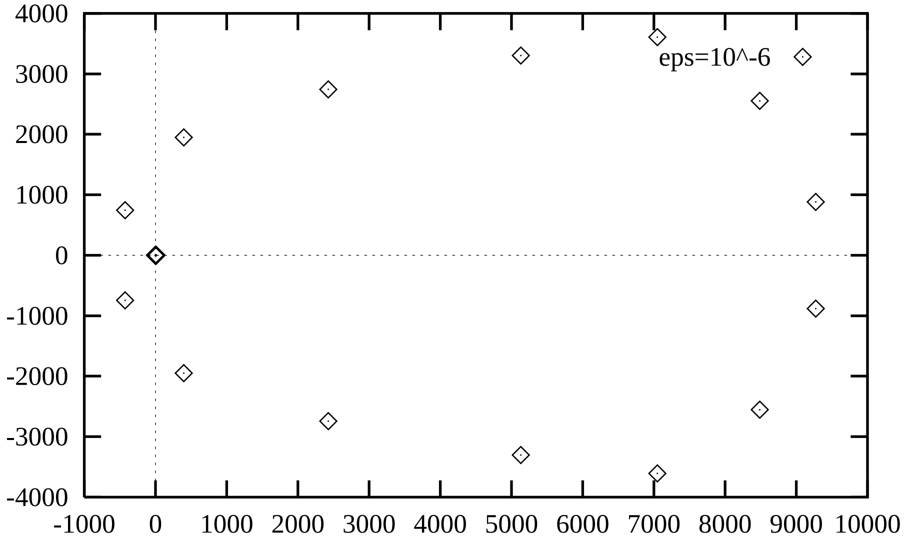

Figure 13: Eigenvalues of $L_{FD}^{-1}L_{SP}$ for N=15, (a,b)=(1,1)

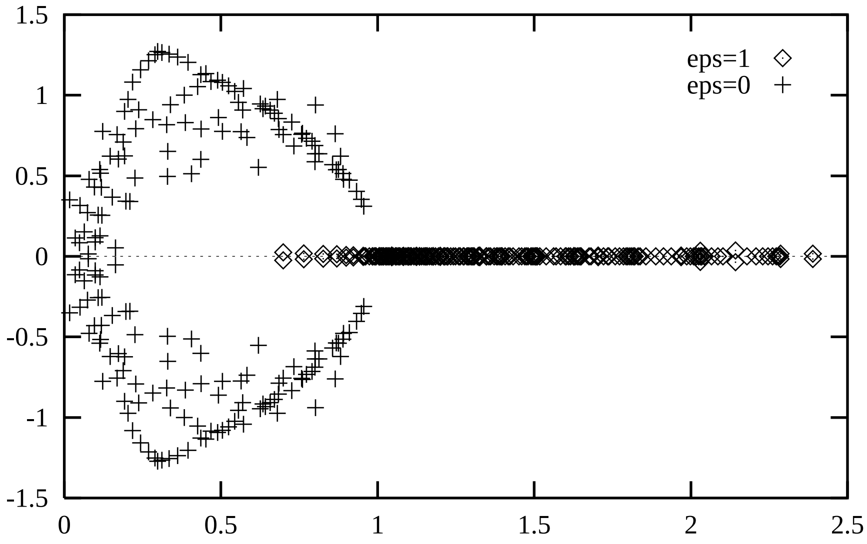

Figure 14: Eigenvalues of the upstream operator for N=15, (a,b)=(1,1)

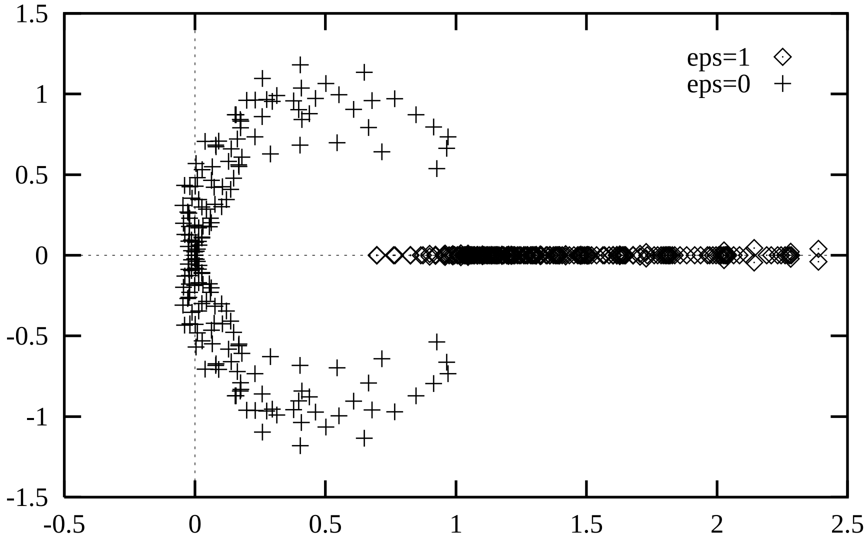

Figure 15: Eigenvalues of the upstream operator for N=15, (a,b)=(-1,1)

	(a,b)	$(1,1)$			$(-1,1)$		
ϵ	N	λ_{max}	λ_{min}	$cond$	λ_{max}	λ_{min}	$cond$
1	3	1.40	$9.28 \cdot 10^{-1}$	1.51	1.41	$9.35 \cdot 10^{-1}$	1.50
	7	2.06	$8.61 \cdot 10^{-1}$	2.39	2.06	$8.58 \cdot 10^{-1}$	2.40
	15	2.39	$6.99 \cdot 10^{-1}$	3.42	2.39	$6.97 \cdot 10^{-1}$	3.43
	31	2.85	$5.91 \cdot 10^{-1}$	4.82	2.85	$5.91 \cdot 10^{-1}$	4.82
10^{-2}	3	$6.04 \cdot 10^{-1}$	$3.04 \cdot 10^{-1}$	1.98	$2.86 \cdot 10^{-1}$	$1.11 \cdot 10^{-1}$	2.58
	7	1.37	$2.99 \cdot 10^{-1}$	4.56	1.46	$2.67 \cdot 10^{-1}$	5.49
	15	2.18	$4.09 \cdot 10^{-1}$	5.34	2.19	$4.81 \cdot 10^{-1}$	4.55
	31	2.37	$4.34 \cdot 10^{-1}$	5.47	2.37	$5.08 \cdot 10^{-1}$	4.67
10^{-4}	3	$6.66 \cdot 10^{-1}$	$3.15 \cdot 10^{-1}$	2.11	$3.33 \cdot 10^{-1}$	$1.41 \cdot 10^{-3}$	$2.35 \cdot 10^{2}$
	7	1.14	$1.50 \cdot 10^{-1}$	7.61	1.19	$3.10 \cdot 10^{-3}$	$3.84 \cdot 10^{2}$
	15	1.29	$8.44 \cdot 10^{-2}$	15.3	1.30	$8.37 \cdot 10^{-3}$	$1.56 \cdot 10^{2}$
	31	1.82	$8.08 \cdot 10^{-2}$	22.5	1.82	$1.74 \cdot 10^{-2}$	$1.05 \cdot 10^{2}$
10^{-6}	3	$6.67 \cdot 10^{-1}$	$3.15 \cdot 10^{-1}$	2.11	$3.33 \cdot 10^{-1}$	$1.42 \cdot 10^{-5}$	$2.34 \cdot 10^{4}$
	7	1.15	$1.50 \cdot 10^{-1}$	7.65	1.20	$3.11 \cdot 10^{-5}$	$3.85 \cdot 10^{4}$
	15	1.31	$7.79 \cdot 10^{-2}$	16.8	1.31	$8.41 \cdot 10^{-5}$	$1.55 \cdot 10^{4}$
	31	1.41	$5.72 \cdot 10^{-2}$	24.7	1.34	$1.79 \cdot 10^{-4}$	$7.50 \cdot 10^{3}$
0	3	$6.67 \cdot 10^{-1}$	$3.15 \cdot 10^{-1}$	2.11	$3.33 \cdot 10^{-1}$	$2.25 \cdot 10^{-17}$	$1.48 \cdot 10^{16}$
	7	1.15	$1.50 \cdot 10^{-1}$	7.65	1.20	$1.84 \cdot 10^{-18}$	$6.48 \cdot 10^{17}$
	15	1.31	$7.78 \cdot 10^{-2}$	16.8	1.31	$6.61 \cdot 10^{-18}$	$1.98 \cdot 10^{17}$
	31	1.41	$5.69 \cdot 10^{-2}$	24.8	1.34	$5.21 \cdot 10^{-17}$	$2.58 \cdot 10^{16}$

Table 9: Upstream method

It is not satisfactory that there are cases (eg. (-1,1)) in which no stability can be achieved. A possibility to overcome this may lie in the introduction of an additional collocation point. The system is then overdetermined. This method has been examined and successfully applied on the square in [1]. A further method may be the use of staggered grids which possibly leads to $\lambda_{min} > 0$. Two different sets of grids are used - one for the solution and the other one for its derivative. For the advection-diffusion equation there were positive results in [2].
Instead of using the Gauss algorithm for solving the linear systems, one could apply iterative methods. As many other iterative methods do not support complex eigenvalues we recommend the use of the GMRES method (see [5]) - a method of minimized residuals. The linear system $Bv = g$ where B is a non-symmetric and large matrix is solved as follows. v_0 is the initial solution, $r_0 = g - Bv_0$ and we define the m-th Krylov space $K_m := span\{r_0, Br_0, \ldots, B^{m-1}r_0\}$. Then we find the approximation $v_m \in v_0 + K_m$ such that the m-th residual r_m fulfills $|r_m| = min!$.

Domain decomposition

We are now interested in applying the spectral method to more complex domains. We use the patching method (see [3]) where the domain is separated into square or triangular subdomains on which Gauss-Lobatto nodes are defined. The differential equation is solved at the interior nodes. At the interface we require continuity of the solution and its normal derivative. We consider the Poisson equation with Dirichlet boundary condition

$$\begin{aligned} \Delta u &= f \text{ in } \Omega, \\ u &= g \text{ on } \partial\Omega. \end{aligned}$$

At the interface Γ between two subdomains, information is exchanged until continuity is reached. In one direction Dirichlet information is transfered and in the other direction it is Neumann information. We use an interface relaxation as proposed in [3] i.e. at the Dirichlet side we hand over a weighted sum of subsolutions at the interface. We iterate until the error at the interface is smaller than 10^{-14}. Thus we iteratively solve a sequence of Dirichlet Neumann problems. We begin with a domain composed of one patched triangle and square $\Omega = T \cup R$ while

$$\begin{aligned} T &= \{(x, y) \mid 0 < x, y < 1 \text{ and } x + y < 1\} \text{ and} \\ R &= \{(x, y) \mid 0 < x < 1 \text{ and } -1 < y < 0\}. \end{aligned}$$

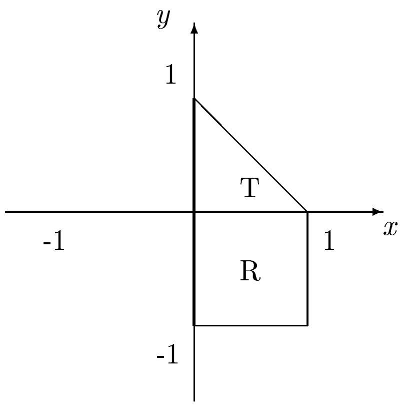

Figure 16: Domain Ω

The interface is $\Gamma = (0, 1) \times \{0\}$ (Figure 16). Initial conditions are $u_1^0 = u_2^0 \equiv 0$ on Ω and $u_1^1 = g$ on Γ. We then iterate

$$\begin{aligned} \Delta u_1^m &= f \text{ in } T, \\ u_1^m &= g \text{ on } \partial T \setminus \Gamma, \\ u_1^m &= \delta^{m-1} u_2^{m-1} + (1 - \delta^{m-1}) u_1^{m-1} \text{ on } \Gamma \end{aligned}$$

and

$$\begin{aligned} \Delta u_2^m &= f \text{ in } R, \\ u_2^m &= g \text{ on } \partial R \setminus \Gamma, \\ \frac{\partial}{\partial y} u_2^m &= \frac{\partial}{\partial y} u_1^m \text{ on } \Gamma. \end{aligned}$$

Here δ^m denotes the relaxation parameter which is chosen dynamically. This dynamical choice usually accelerates the convergence. $\delta^m = \delta$ is the unique number which minimizes $\| z_m(\delta) - z_{m-1}(\delta) \|_2^2$ where $z_m(\delta) = \delta u_2^m + (1 - \delta) u_1^m$. δ^m is calculated by

$$\delta^m = \frac{(e_1^m, e_1^m - e_2^m)}{\| e_1^m - e_2^m \|_2^2},$$

where $(.,.)$ denotes the discrete L_2 inner product and

$$e_i^m = u_i^m - u_i^{m-1} \text{ for } i = 1, 2$$

is the difference of two consecutive iterates on the two subdomains. δ^m should be in $(0, 1]$. We cannot use example (1) because this function vanishes at the interface. Therefore no new information is exchanged which makes an iterative method superfluous, as it converges after the first step. Thus we introduce the following oscillating example

$$u(x, y) = \sin(\pi x)\sin(\pi y + \frac{\pi}{4}). \tag{2}$$

N	It	$E2_T$	$E2_R$
4	15	$1.36 \cdot 10^{-2}$	$1.28 \cdot 10^{-2}$
8	17	$2.42 \cdot 10^{-5}$	$1.64 \cdot 10^{-5}$
16	17	$8.01 \cdot 10^{-13}$	$4.24 \cdot 10^{-13}$
32	17	$1.00 \cdot 10^{-14}$	$1.60 \cdot 10^{-14}$

Table 10: Ω with (2)

Table 10 shows the number of iterations and the discrete L_2 error on square and triangle. We reach the tolerance after relatively few steps. The convergence is fairly slow because of the oscillatory behaviour of the solution. The number of iterations is constant and independent of N. Machine accuracy is reached for N=16.

The second domain Ω_1 to be studied consists of Ω and an additional triangle T_1 attached to the already existing one (Figure 17).

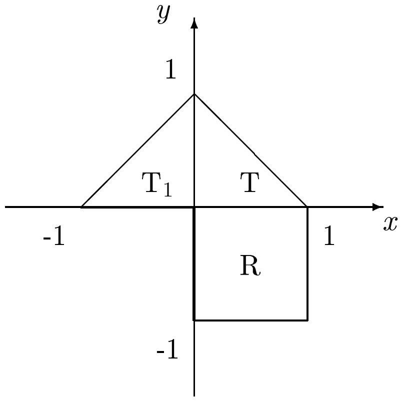

Figure 17: Domain Ω_1

We begin with triangle T with interface boundaries $\Gamma_1 = (0, 1) \times \{0\}$ and $\Gamma_2 = \{0\} \times (0, 1)$ and $\Gamma = \Gamma_1 \cup \Gamma_2$. Then we solve on R and T_1. This should be realized on a parallel computer. The algorithm reads

$$\Delta u_1^m \;=\; f \text{ in } T,$$

$$
\begin{aligned}
u_1^m &= g \text{ on } \partial T \setminus \Gamma, \\
u_1^m &= \delta_1^{m-1} u_2^{m-1} + (1-\delta_1^{m-1}) u_1^{m-1} \text{ on } \Gamma_1, \\
u_1^m &= \delta_2^{m-1} u_3^{m-1} + (1-\delta_2^{m-1}) u_1^{m-1} \text{ on } \Gamma_2,
\end{aligned}
$$

and

$$
\begin{aligned}
\Delta u_2^m &= f \text{ in } R, \\
u_2^m &= g \text{ on } \partial R \setminus \Gamma_1, \\
\frac{\partial}{\partial y} u_2^m &= \frac{\partial}{\partial y} u_1^m \text{ on } \Gamma_1,
\end{aligned}
$$

and

$$
\begin{aligned}
\Delta u_3^m &= f \text{ in } T_1, \\
u_3^m &= g \text{ on } \partial T_1 \setminus \Gamma_2, \\
\frac{\partial}{\partial x} u_3^m &= \frac{\partial}{\partial x} u_1^m \text{ on } \Gamma_2.
\end{aligned}
$$

N	It	$E2_T$	$E2_R$	$E2_{T1}$
4	65	$1.04 \cdot 10^{-2}$	$9.62 \cdot 10^{-3}$	$1.25 \cdot 10^{-2}$
8	83	$6.98 \cdot 10^{-6}$	$5.67 \cdot 10^{-6}$	$2.68 \cdot 10^{-5}$
16	84	$6.85 \cdot 10^{-13}$	$4.01 \cdot 10^{-13}$	$1.02 \cdot 10^{-12}$
32	90	$2.31 \cdot 10^{-13}$	$8.30 \cdot 10^{-14}$	$9.90 \cdot 10^{-14}$

Table 11: Ω_1 with (3)

Initial values are analogous to the last example. We apply this algorithm to the example

$$
u(x,y) = \sin(\pi x + \frac{\pi}{4}) \sin(\pi y + \frac{\pi}{4}). \tag{3}
$$

The results are listed in Table 11. The number of iterations is extremely increased if a further triangle is added. Unfortunately, δ_i^m tends to leave the interval $(0,1]$. Whenever this happens, the following approximation is worse than the one before. Nevertheless, the method finally converges. This dynamical choice of δ_i^m is not optimal. We have derived results for fixed $\delta_i^m = \frac{1}{2}$ in Table 12.

N	It	$E2_T$	$E2_R$	$E2_{T1}$
4	34	$4.89 \cdot 10^{-4}$	$2.82 \cdot 10^{-4}$	$1.51 \cdot 10^{-3}$
8	41	$9.29 \cdot 10^{-9}$	$2.30 \cdot 10^{-9}$	$6.82 \cdot 10^{-8}$
16	46	$1.78 \cdot 10^{-13}$	$3.70 \cdot 10^{-14}$	$1.83 \cdot 10^{-13}$
32	87	$9.68 \cdot 10^{-13}$	$8.40 \cdot 10^{-14}$	$1.01 \cdot 10^{-12}$

Table 12: Ω_1 with (3) and $\delta^m = 0.5$

The number of iterations is smaller and there are no 'backward steps' any more.

Finally, we study the domain $\Omega_2 = R \cup T_1 \cup T_2 \cup T_3 \cup T_4$ (T_i triangles) which is symmetric to the origin (Figure 18). We consider the following example

$$u(x,y) = \sin(3\pi x + \frac{\pi}{4}) \sin(3\pi y + \frac{\pi}{4}). \tag{4}$$

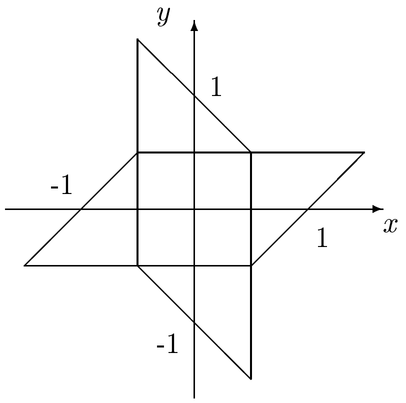

Figure 18: 'wind wheel' Ω_2

The algorithm is analogous to the last one and we first solve on the square and then on the triangles. The results in Table 13 are fairly good for symmetry reasons considered that we now deal with five subdomains. The number of iterations is constant and machine accuracy is reached for N=16.

N	It	$E2_R$	$E2_{T1}$	$E2_{T2}$	$E2_{T3}$	$E2_{T4}$
8	75	$9.00 \cdot 10^{-2}$	$5.86 \cdot 10^{-2}$	$7.55 \cdot 10^{-2}$	$4.89 \cdot 10^{-2}$	$7.13 \cdot 10^{-2}$
16	72	$4.33 \cdot 10^{-5}$	$2.08 \cdot 10^{-5}$	$4.70 \cdot 10^{-5}$	$1.79 \cdot 10^{-5}$	$4.63 \cdot 10^{-5}$
32	69	$3.83 \cdot 10^{-13}$	$5.10 \cdot 10^{-14}$	$1.06 \cdot 10^{-13}$	$6.60 \cdot 10^{-14}$	$8.60 \cdot 10^{-14}$

Table 13: Ω_2 with (4)

Summing up we constate that this spectral method is effective for domain decomposition problems, too. Now, we can also deal with partial differential equations on complex domains using spectral methods as long as those domains can be separated into rectangular and triangular elements.

Bibliography

[1] H. Eisen, W. Heinrichs, *A new method of stabilization for singular perturbation problems with spectral methods.* SIAM J. Numer. Anal. 29, pp. 107-122 (1992).

[2] D. Funaro, *A fast solver for elliptic boundary-value problems in the square.* Comput. Methods Appl. Engrg. 116, pp. 253-255 (1994).

[3] D. Funaro, A. Quarteroni, P. Zanolli, *An iterative procedure with interface relaxation for domain decomposition methods,* SIAM J. Num. Anal., 25, pp. 1213-1236 (1988)

[4] M. Griebel, T. Dornseifer, T. Neunhoeffer *Numerische Simulation in der Strömungsmechanik.* Vieweg Lehrbuch, 1995.

[5] W. Heinrichs, *Defect correction for convection-dominated flow.* SIAM J. Sci. Comput., 17, No. 5, pp. 1082-1091, 1996.

[6] W. Heinrichs, *Spectral collocation on triangular elements,* J. Comp. Ph., 145, pp. 743-757 (1998)

[7] S. Sherwin, G. Karniadakis, *Triangular and tetrahedral spectral elements.* ICOSAHOM'95: Proceedings of the third international conference on spectral and high order methods, 1996 Houston Journal of Mathematics.