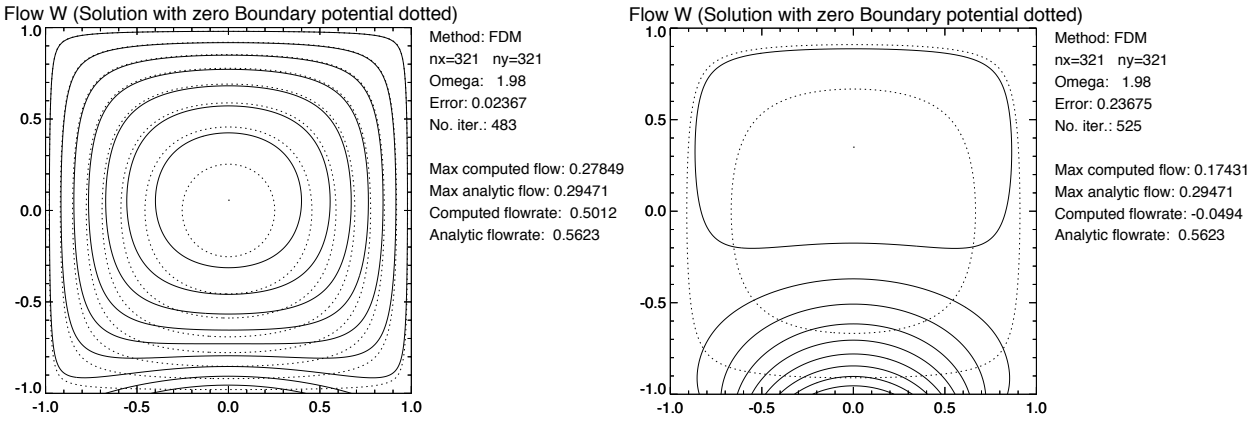
33. Alter the boundary condition for program duct at the minimum boundary at y to $w(x,-1) = -a\cos^2(\pi x/2)$. Run the program with SOR and finite differences for the grid sequence

21^2, 41^2, 81^2, and 161^2 for $a = 0.1$ and $a = 1$. Plot the solutions for the 161^2 grid for $a = 0.1$ and $a = 1$. What are the changes for the solution? Do the iteration properties change, i.e., the scaling of N_{iter} with N and the value of λ_{opt} and compare this with the results from Problem 30?

The program duct has no subroutine dedicated to assign boundary conditions because the original intent for it did not require either time-dependent or von Neumann boundary conditions. Since boundary conditions are Dirichlet conditions constant in time they are assigned through the initial conditions. In order to apply the new constant boundary condition it is sufficient to alter the value of $f(x,y)$at the y_{min} boundary in the initial conditions. This is done by adding a do loop after loop index 10 in subroutine initcon:

```
do 20 l = 1,nx
   f(l,1) = -a*cos(pi*x(l)/2.)**2
20 continue
```

with $a = 0.1$ and 1. The results for theses values and the altered boundary condition are plotted below.

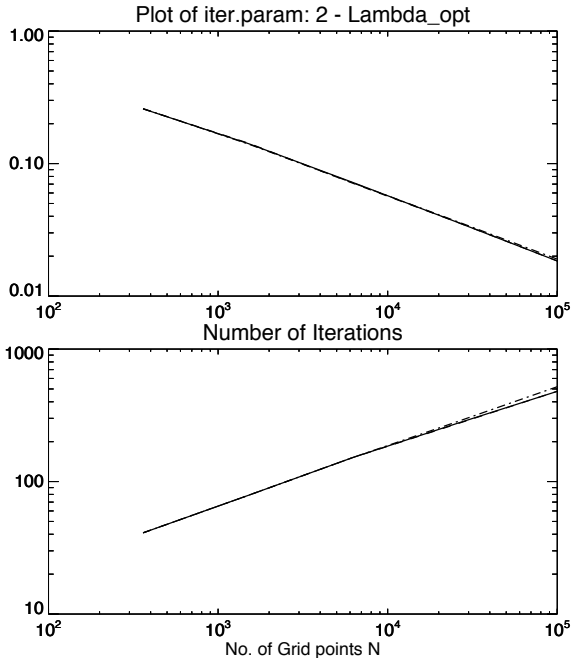


The dotted lines indicate the analytic solution for the unaltered (old) boundary condition. The new condition introduces a nonzero value of the flow at the minimum y boundary. This effect is small for $a = 0.1$and only alters the solution close to the boundary significantly. Further from the boundary and particularly for values of $y > 0$ the solution appears almost unaltered. This changes for a value of $a = 1$. Now the new boundary condition creates a major change in the solution. Note that the different number of contour lines is cause by the larger spacing of contour lines because the difference between maximum and minimum of the solution is much larger. Only close to the top boundary in y is the altered solution still similar to the old solution.

The table below shows the iteration parameters for the unaltered (old) boundary conditions (from homework 30) and for the new conditions with $a = 0.1$ and 1. The table already indicates that the changes are very minor. The optimum value for the SOR parameter λ changes only within the uncertainty of the iteration to determine λ. The number of iterations is almost unchanged for $a = 0.1$ and only increases slightly for large numbers of grid points for $a = 1$.

	N	λ_{opt}	$2-\lambda_{\text{opt}}$	N_{iter}	Predictor for λ_{opt}
fdm0021-0bd	361	1.74100	0.25900	41	1.87050
fdm0041-0bd	1521	1.85900	0.14100	79	1.92950
fdm0081-0bd	6241	1.92800	0.07200	152	1.96400
fdm0161-0bd	25281	1.96360	0.03640	273	1.98180
fdm0321-0bd	101761	1.98175	0.01825	483	1.99088
fdm0021-$a=-0.1$	361	1.74100	0.25900	41	1.87050
fdm0041-$a=-0.1$	1521	1.85800	0.14200	79	1.92900
fdm0081-$a=-0.1$	6241	1.92900	0.07100	152	1.96450
fdm0161-$a=-0.1$	25281	1.96355	0.03645	271	1.98178
fdm0321-$a=-0.1$	101761	1.98175	0.01825	484	1.99088
fdm0021-$a=-1$	361	1.74000	0.26000	41	1.87000
fdm0041-$a=-1$	1521	1.86000	0.14000	79	1.93000
fdm0081-$a=-1$	6241	1.92850	0.07150	152	1.96425
fdm0161-$a=-1$	25281	1.96320	0.03680	281	1.98160
fdm0321-$a=-1$	101761	1.98115	0.01885	524	1.99058

Plots of $\ln(2-\lambda_{opt})$ and $\ln N_{iter}$ versus $\ln N$ for the 3 cases of the table (see also homework 30). The unaltered boundary case (FDM SOR) uses the solid line, The $a = 0.1$ case a dashed and the $a = 1$ cases a dash-dotted line. As indicated by the table the results for these iteration are so close that the solid line masks the other lines for much of the plot. A difference is evident (but still very small) only for large numbers of grid points.



34. The program duct solves an elliptic partial differential equation that is equivalent to the Poisson equation. Interpreting the problem as an electrostatic problem, w becomes the electric potential.

(a) What is the physical meaning of $w = 0$ at the boundaries? How are contour lines of w related to the electric field?

$$\left(\frac{b}{a}\right)^2 \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + 1 = 0$$

For an electrostatic problem w becomes the electric potential and $\mathbf{E} = -\nabla w$ such that the electric field is perpendicular to equipotential contours. Specifically, a constant potential $w = 0$ at the boundary implies that the electric field is normal to the boundary such that the tangential magnetic field is 0. This is usually interpreted as a conducting boundary. A varying potential along the boundary prescribes the electric field component tangential to the boundary. In ionospheric physics convection in the high latitude ionosphere creates an electric field. according to Ohm's law $\mathbf{E} = -\mathbf{v} \times \mathbf{B}$. Since the dominant magnetic is in the vertical direction and the velocity is dominant in the horizontal direction, the electric field is also in the horizontal direction (perpendicular to the velocity). In a steady state the electric field can be considered a potential field. Since contour lines of the potential are perpendicular to the electric field, they must be aligned with the convection velocity. In this case iso-contours of the electric potential are streamlines of the plasma flow.

(b) Assume that the normal electric field is 0 at the boundary $x = -1$. Describe how you can implement this condition in the program duct. Alter the program according to this description and run the program for the 161×161 grid for FDM and FEM. Comment your results.

The normal electric field is given by $E_n = -\partial w/\partial n$ such that $E_n = 0$ corresponds to $\partial w/\partial n = 0$, i.e., a von Neumann boundary condition. To realize this boundary condition we need to extend the iteration to include the physical boundary. Note that for Dirichlet conditions it only is applied to the interior of the system, i.e., from grid point 2 to $n-1$. For instance using the FDM the iteration determines $w^*_{j,k}$(the Gauss-Seidel $n+1$ iteration step) from

$$w^*_{j,k} = \frac{1}{2c_0}\left[1 + \left(\frac{b}{a}\right)^2 \frac{w^{(n+1)}_{j-1,k} + w^{(n)}_{j+1,k}}{\Delta x^2} + \frac{w^{(n+1)}_{j,k-1} + w^{(n)}_{j,k+1}}{\Delta y^2}\right] \tag{1}$$

The zero derivative boundary condition (at the minimum x boundary implies $\partial w/\partial n|_{xmin} = 0 = \left(w_{2,k} - w_{0,k}\right)/(2\Delta_x)$ or $w_{0,k} = w_{2,k}$. Thus for $j = 1$ we obtain

$$\begin{aligned} w^*_{1,k} &= \frac{1}{2c_0}\left[1 + \left(\frac{b}{a}\right)^2 \frac{w^{(n+1)}_{0,k} + w^{(n)}_{2,k}}{\Delta x^2} + \frac{w^{(n+1)}_{1,k-1} + w^{(n)}_{1,k+1}}{\Delta y^2}\right] \\ &= \frac{1}{2c_0}\left[1 + 2\left(\frac{b}{a}\right)^2 \frac{w^{(n+1)}_{2,k}}{\Delta x^2} + \frac{w^{(n+1)}_{1,k-1} + w^{(n)}_{1,k+1}}{\Delta y^2}\right] \end{aligned} \tag{2}$$

In other words for $j = 1$ we need to include another iteration (loop) for k=1 and implement the change in w^* loop that uses the above expression.

The approach is the same for finite elements where in the following expression for $j = 1$ all terms with index $j-1 = 0$ should be replace with $j-1 = 2$.

$$\begin{aligned} w^*_{1,k} &= \frac{3}{4c_0}\left[1. + c_1\left(w_{j-1,k-1} + w_{j+1,k-1} + w_{j-1,k+1} + w_{j+1,k+1}\right)\right. \\ &\qquad \left. + c_2\left(w_{j-1,k} + w_{j+1,k}\right) + c_3\left(w_{j,k-1} + w_{j,k+1}\right)\right]_{j=1} \\ &= \frac{3}{4c_0}\left[1. + 2c_1\left(w_{2,k-1} + w_{2,k+1}\right) + 2c_2 w_{2,k} + c_3\left(w_{1,k-1} + w_{1k+1}\right)\right] \end{aligned}$$

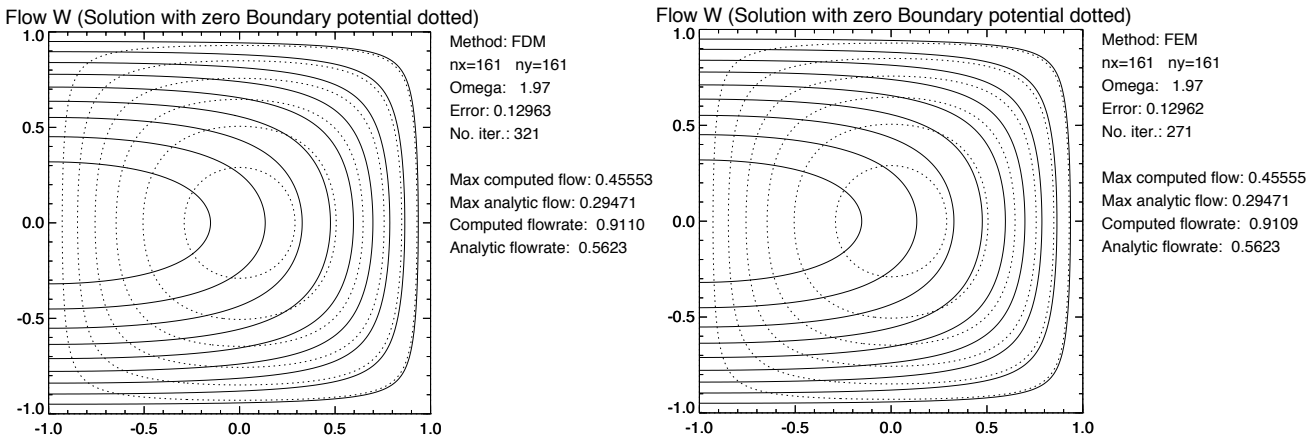
with

$$\begin{aligned} c_0 &= \left(\frac{b}{a}\right)^2 \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \\ c_1 &= \frac{1}{6} c_0 \\ c_2 &= \left(\frac{b}{a}\right)^2 \frac{2}{3\Delta x^2} - \frac{1}{3\Delta y^2} \\ c_3 &= -\left(\frac{b}{a}\right)^2 \frac{1}{3\Delta x^2} + \frac{2}{3\Delta y^2} \end{aligned}$$

The changes to implement these new conditions are actually very minor. Loops 13 and 14 in the subroutine *SOR* need to include the j=1 value and we need to make certain that for $\mathtt{j} = 1$ the variable jm uses a value of 2. In the program the first 4 lines of loop 13 in SOR now read

```
do 13 j = 1,nx-1
  jm = j-1
  jp = j+1
  if (j .eq. 1)  jm=2
```

and we use the same change for loop 14! The following two plots show the results for the altered von Neumann boundary conditions for FDM and FEM (and 321^2 grid points):



Note that the number of iterations is slightly (factor 1.7) larger than for the Dirichlet conditions.

(c) In the current version of duct the pressure gradient is constant and equivalent to a constant charge density. How do you need to alter duct to incorporate a varying charge density $n = n(x,y)$?

The constant pressure gradient term generates the constant 1 in the brackets of the expressions for $w^*_{j,k}$ (see equations (1) and (2) on prior page). Therefore we need to evaluate $n(x,y)$ on the grid, i.e., introduce a new two-dimensional variable $\texttt{dens}_{jk}$, assign density to it and

for finite differences replace 1 in (1) with $n_{j,k}$.

for finite elements replace 1 in (2) with

$$\begin{aligned} M_x \otimes M_y n_{jk} &= \frac{4}{9} n_{j,k} + \frac{1}{9}\left(n_{j-1,k} + n_{j+1,k} + n_{j,k-1} + n_{j,k+1}\right) \\ &\quad + \frac{1}{36}\left(n_{j-1,k-1} + n_{j+1,k-1} + n_{j-1,k+1} + n_{j+1,k+1}\right) \end{aligned}$$