

MASTER

A numerical calculation of the flow through the actuator strip

van Helden, W.G.J.

*Award date:*  
1988

[Link to publication](#)

**Disclaimer**

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

**Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A NUMERICAL CALCULATION OF THE FLOW  
THROUGH THE ACTUATOR STRIP

Wim van Helden

March, 1988

R-898-A

Thesis of the graduationwork, accomplished between  
April, 1987 and March, 1988.  
W.G.J. van Helden.

coaches: Gijs van Kuik  
Paul Smulders.

Faculty of Physics, Fluid Dynamics, Wind Energy Group,  
Eindhoven University of Technology.

---

'Rommelig,' besloot mijnheer Charles C.M. Carlier. 'En slordig!' Het was, als je 't goed naging, eigenlijk allemaal nogal rommelig, slordig eigenlijk, en onafgewerkt; - de schepping en zo, de sterrenhemel en zo, zelfs al was het de zomerse sterrenhemel van zijn geliefd Provence.

Havank: Lijk Halfstok.

Dit afstudeer verslag draag ik op aan mijn ouders.

---

Contents.

page nr.	Chapter	Section
1		Summary
2	-	Introduction
4	1	The Actuator Strip
8	2	The Flow Equations
8	2.1	The Equations
11	2.2	The Differential Schemes
17	2.3	The Boundary Conditions.
19	3	The Half-infinite Actuator Strip
19	3.1	Model, Source Term and Grid
24	3.2	The convective Term in the Differential Scheme
25	3.3	Calculations and Results
31	3.4	Convergence.
34	3.5	Comparison with an Analytical Solution.
37	4	The (finite) Actuator Strip; Grid Generation
37	4.1	Model of the flow through the actuator strip.
40	4.2	Numerical grid generation.
47	4.3	Application of grid generation to the actuator strip.
54	5	The (finite) Actuator Strip; Calculations, Results
54	5.1	Equations, Boundary- and Starting Conditions.
56	5.2	Flow Chart, Starting Grid.
61	6	Conclusions & Discussion
62		List of References.
64	A1	AppendixA1 (in Dutch) De sterk Impliciete Methode van Stone.
68	A2	Appendix2 Lay-out of the computer-programmes.

---

## Summary.

The actuator strip is a massless strip of infinite length which is covered with a uniform force distribution. It is used as a two-dimensional model for the flow through propellers or wind turbines. Two different computer programmes have been developed; one for the calculation of the unsteady flow through the actuator strip with only one edge ( the half-infinite actuator strip ), the other for the calculation of the unsteady flow through the finite actuator strip. In both cases, the flow at a particular time is found by solving the vorticity diffusion equation and the Poisson-PSI equation for the streamfunction.

The results for the half-infinite strip are good. The strip, switched on at  $t = 0$ , produces a velocity field with concentric, circular streamlines. The edge of the strip is the centre of the circles. In a great part of the domain the velocity field resembles the velocity field of a point vortex.

Also a comparison is made between the calculations and an analytical solution.

For the calculation of the flow through the finite actuator strip, a form of adaptive grid generation has been used. At each time step in the calculation the grid points are redistributed dependent on the solution of the differential equations in the previous time step. The adaptive grid generation has been made as simple as possible. In this way the principles of grid generation have become much clearer.

Because of the complexity of the computer programme it wasn't possible to finish the work completely. Therefore only preliminary results concerning the computed flow through the finite actuator strip are given.

The importance of windturbines in electricity generation or water pumping is still increasing, while helicopters and propellor-driven aircraft already play a very important role in air traffic and air transport. All three types of machines are rotating lift machines, they exchange energy with the surrounding air by revolving several rotorblades.

The goal of the designers of for instance windturbines is to make a windturbine with maximum power output at minimum cost. In order to obtain this goal it is necessary to have a good theory of the flow through a rotor. With such a theory one should be able to make an accurate prediction of the maximum power that can be extracted from the wind by a rotor before the rotor is built.

The ratio of the extracted power to the total power of the undisturbed wind going through the area of the rotor is called  $C_p$ . The value of  $C_p$  is given by the theory of Betz & Lancaster and has a maximum of  $\approx 59\%$ . There are some indications, however, that this Lancaster maximum and therefore this theory needs some modification.[ref.12]

The rotating movement of the rotor aggravates a straightforward theoretical modeling of the problem. For this reason some simplifications are used.

An example of a commonly used model is that of the actuator disc. This is a disc without mass that is covered with a force distribution. An imaginable example for a braking actuator disc is a circular screen or wire gauze.

This actuator disc model is often used to calculate the pattern of the flow through a rotor and especially the formation of vortices at the edge of the disc.

One suggestion is that discrete forces resulting from a flow singularity at the edge of the disc, cause a higher  $C_p$ . [ref.12,ref.6].

The problem of calculating the flow through an actuator disc can be tackled in different ways. For instance, Alex van der Spek [ref.13] gives a first order solution of Wu's equations [ref.14].

In this work, the 2-dimensional analogy of the actuator disc, the actuator strip, is used. The flow through the actuator strip is only a function of the x- and y-coordinates and thus the vorticity has only a

z-component. Therefore, only the component for the z-direction of the vorticity diffusion equation has to be solved to obtain the vorticity. Three relations, the vorticity diffusion equation, the Poisson-PSI equation to calculate the streamfunction from the vorticity, and the relation between velocity and streamfunction, give a closed set of equations to determine the flow as a function of time.

The aim of this work is to get insight into the behaviour of the flow and perhaps obtain an estimate for the strength of the discrete vortices at the edges of the strip.

To avoid large gradients in the force density distribution at the edges of the strip, the force density is given by a function whose gradient at the edges can be adjusted. In the computations of the flow through the actuator strip a simple form of adaptive grid generation is used.

At each time step in the computation the grid points in the physical grid are redistributed according to the solution of the problem found in the preceding time step.

Two different configurations were calculated, the first being the half-infinite actuator strip. This is a strip with only one edge. The calculations with this configuration were used to test the convergence and the validity of the difference schemes that were used.

The second configuration, the finite actuator strip, is going to be used to determine the starting behaviour of the flow through the strip and to make an estimate of the value of the discrete vortices at the edges of the strip. This value is the yet missing link in the new theory presented in [ref.6].

The calculations are restricted to the situation with zero undisturbed velocity. This situation can be compared with the flow caused by a helicopter in hover.

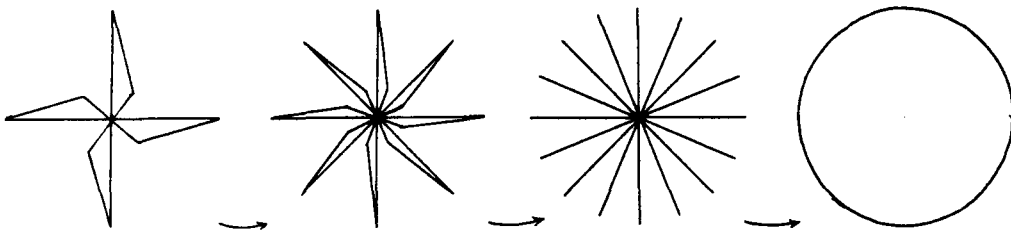
The complexity of the programme to calculate the flow through the finite actuator strip made it not possible to complete the work within the normal time. Only some results of testing programmes can be presented in this report. However, the whole programme is complete now and is in the final state of debugging. The rest of the calculations will be presented in an addendum.

## 1. The Actuator Strip.

Solving the flow equations for flows around wings, propellers and bodies in general is very difficult because of the complexity of the flow.

Things could be made easier, if idealized models were used. Such a model for the flow through a rotor is the flow through an actuator disc.

The actuator disc is obtained from a rotor by taking limits: the number of blades is increased and the chord of the blades decreased in such a way, that their product is kept constant. The total area of the rotorblades that is "seen" by the flow, is kept constant.



*Fig.(1.1): Transition from real rotor to actuator disc.*

The number of blades is infinite in the limit, the chord is zero. The force exerted on the air by the rotor (propellor) or vice versa (windturbine) is equally spread over the actuator strip. The force distribution over one blade divided by the radial distance to the axis equals the radial force distribution on the disc. This force is not necessarily constant along the radius of the disc. From now on we, however, assume a constant distribution of force density on the disc. The assumed force distribution on the blade then is proportional to the radial distance from the axis.



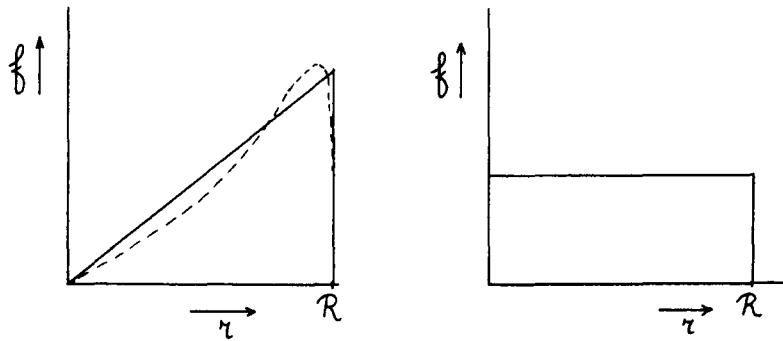
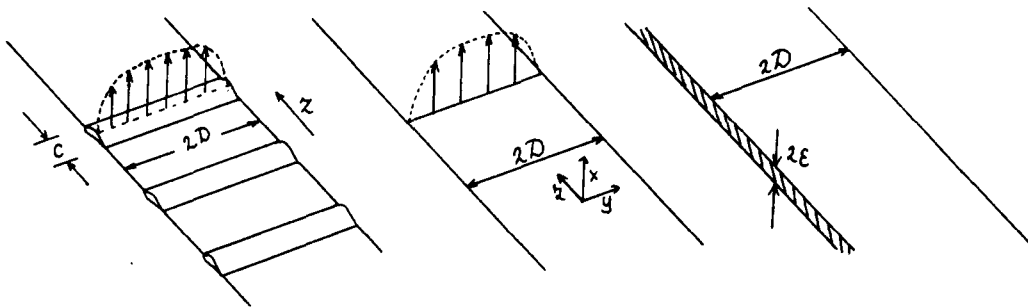


Fig.(1.2): Force distribution on rotor blade (left) in  $N/m$  and on the actuator disc in  $N/m^2$ . The dashed line gives the force distribution on a real blade.

Another way of obtaining the actuator strip is to place an infinite number of wing sections in succession in  $Z$ -direction, each with the same lift force distribution. By again increasing the number of wing sections while reducing the chord  $c$ , the actuator strip is born.



Figs.(1.3), (1.4a) and (1.4b): Placing infinitely many wingsections in a line while decreasing the chord  $c$  gives the actuator strip. The actuator strip that is used in the calculations actually has thickness  $2\epsilon$ .

Because the force density distribution is independent of the  $z$ -coordinate the flow around and through the actuator strip is a 2-dimensional one. This flow can be studied in an  $X$ - $Y$  plane, with  $z = 0$  for instance. If we again assume a constant force density distribution, we get a situation as depicted in figure 1.5. In the remainder we assume, that the force exerted by the strip, points only in the

$\pm X$ -direction. The edges of the strip are at  $y = \pm D$ . The unit of force density is  $(N/m^3)$ . The force density is also spread over a small layer with thickness  $2\epsilon$ , from  $x = -\epsilon$  to  $x = +\epsilon$ .

Mathematically we can take the limit for  $\epsilon \rightarrow 0$  and arrive at a force area 'density' in  $(N/m^2)$ .

If we want to solve the problem numerically, however, the finite thickness of the layer has to be maintained, as shall be explained in chapter 3.

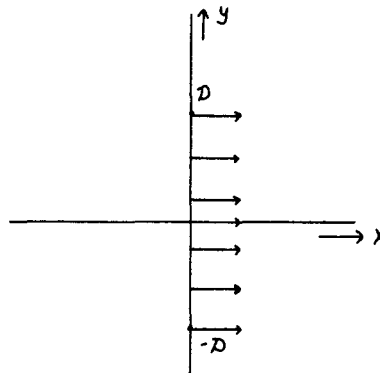


Fig.(1.5): Cross section of the actuator disc in the  $x$ - $y$  plane. The force distribution is spread from  $x = -\epsilon$  to  $x = \epsilon$ .

In  $X$ -direction a distribution of force density is chosen which is a well-known starting function to obtain the Dirac  $\delta$ -function.

The function reads:

$$f(x,y) = f(y) \cdot f_0 \cdot \frac{n}{2 \cosh^2(nx)} \quad (1.1)$$

The area under the function is given by:

$$\int_{-\infty}^{+\infty} f(x,y) dx = f(y) \cdot f_0 \cdot \left. \frac{1}{2} \tanh(nx) \right|_{-\infty}^{+\infty} = f(y) \cdot f_0 \quad (1.2)$$

By altering the parameter  $n$  one is able to vary the peakheight to width ratio without changing the area beneath the function. As we shall see

later on, this is a useful possibility when calculating the flow numerically. Letting  $n$  go to infinite gives the Dirac  $\delta$ -function.

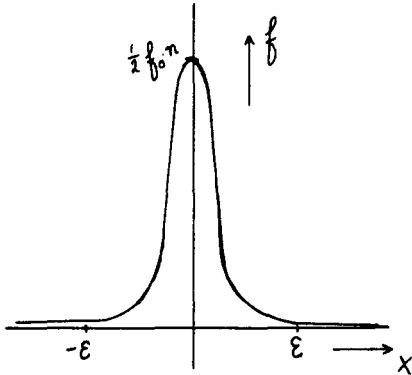


Fig.(1.6): Force density distribution as a function of  $x$  for  $y = 0$ .

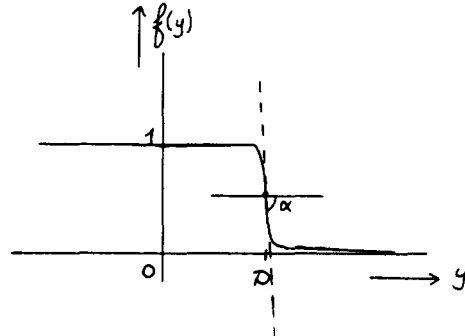


Fig.(1.7): Force density distribution as a function of  $y$  for  $x = 0$ .

At the edges, the force density falls gradually from maximum value to zero, but within a short distance. The  $Y$ -dependence at  $Y = + D$  is given by:

$$f(y) = 1/2 \cdot ( 1 - \tanh(m(y - D)) ) \tag{1.3}$$

Here  $m$  determines the distance over which the function falls from nearly maximum to nearly zero. The maximum gradient of the function is at  $y = D$  and equals  $-1/2 \cdot m$ .

For the edge at  $y = -D$ , the derivation of the function defining the force is similar.

## 2. The Flow Equations.

### 2.1 Equations.

A flow field in general is determined by the direction and absolute value of the velocity of elementary cells in the entire domain.

In other words: solving the flow equations comes down to determining the velocity field. One of the methods to determine the velocity field is to first solve the vorticity-diffusion equation, and then to calculate the streamfunction which gives the velocity field.

The flow through the actuator strip is caused by the action of the force density distribution in the strip.

The mathematical relationship between forces and velocity is given by the Navier-Stokes equation. This equation, assuming that air is incompressible and has only linear, isotropic shear, reads:

$$\frac{\partial \vec{u}}{\partial t} = \frac{1}{\rho_0} \cdot \vec{f}(x,y) - (\vec{u} \cdot \vec{\nabla}) \cdot \vec{u} + \nu \cdot \vec{\nabla}^2 \cdot \vec{u} \quad (2.1)$$

If we are interested in the production and transportation of vorticity, we take the rotation (the outer product with the differential operator  $\vec{\nabla}$ ) of equation (2.1) and arrive at the Vorticity-Diffusion Equation (VDE); the Z-component of this equation reads:

$$\frac{\partial \omega_z}{\partial t} = \frac{1}{\rho_0} \left[ - \frac{\partial f_x}{\partial y} + \frac{\partial f_y}{\partial x} \right] - (\vec{u} \cdot \vec{\nabla}) \omega_z + \nu \vec{\nabla}^2 \omega_z \quad (2.2)$$

From now on,  $\omega$  shall be used instead of  $\omega_z$ .

The term on the left-hand side gives the change of the vorticity as a function of time, this change results from the terms on the right-hand

side. The first one of these gives the production of vorticity by external forces, in our case by the gradient of volumetric force density at the edge(s) of the strip. The actuator strip is switched on at  $t = 0$ , this means that  $f(x,y) = 0$  for  $t \leq 0$ . The second term gives the transportation of vorticity by the local velocity and is called the convective term. The latter term gives the transportation of vorticity by diffusion, the diffusion term.

The reader may ask himself, why equation (2.1) is not directly solved in order to get the velocity field. The reason for this is that we are also interested in the distribution of vorticity. As it turns out, the actuator strip produces vorticity at the edges. These vortices can be seen directly as peaks in the vorticity distribution. Working numerically with peaked functions is only possible, if the grid points are distributed in a special way. For this grid point distribution the vorticity distribution has to be known.

Back to the equations. Because the flow is incompressible and 2-dimensional, a streamfunction  $\psi$  can be defined with:

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x} \quad (2.3)$$

where  $u$  is the velocity in X-direction,  $v$  the velocity in Y-direction, and with the definition of  $\omega$ , the vorticity in Z-direction:

$$\omega = (\vec{\nabla} \times \vec{u}) = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \quad (2.4)$$

we find the relationship between  $\omega$  and  $\psi$ , the so-called Poisson-PSI equation (PPE):

$$\nabla^2 \psi = -\omega \quad (2.5)$$

By solving this equation for a given vorticity distribution, the accompanying streamfunction is found. With a known streamfunction, the

velocity is found by applying the equations (2.3). Now the circle of equations, to be used in one time step, is closed.

In principle, a solution to equation (2.2) for a given time can be found. This could be done, for example, by an iterative process; an initial estimate of the velocity is used to solve the vorticity diffusion equation and the Poisson-PSI equation, which gives a new velocity distribution. This process is repeated until the difference between initial and found velocity are small enough. Choosing this method would increase the time, needed to solve the time-dependent problem, very much.

So an additional assumption has to be made to override this difficulty. By only allowing small time increments  $\Delta t$ , the velocity is assumed to be quasi-stationary, so equation (2.2) can be solved at  $t_1$  using the velocity at time  $t_0$ . This introduces an error in the computations, but this error can be reduced by decreasing the time step.

The equations (2.2) and (2.5) can only be solved, if some additional relationships for  $\omega$  and  $\psi$ , respectively, are given at the boundaries of the domain. These relationships are called the boundary conditions. For the flow through a half-infinite actuator strip the far-field boundary condition has been used: vorticity and velocity are both zero at the boundaries. These boundaries are taken far away from the origin. In calculating the flow through the finite actuator strip the far field condition has been used for three boundaries. The condition for the fourth boundary was the mirror-condition: because the flow through the actuator strip is symmetrical only half the field has to be computed. At crossing the symmetry line the velocity in x-direction stays the

same, while the velocity in y-direction and the vorticity change sign.

2.2 The differential schemes.

Because each domain has infinitely many points, it's impossible to solve any differential equation for each point numerically. For this reason, a grid is laid over the domain and the differential equation is transformed into a so-called differential scheme, a relationship between the function values of adjoining grid points. Let's take the function  $\omega(x,y)$  as an example. The computational grid is regular ( the distance between grid points is constant) and orthogonal (the lines linking grid points with the same X-index cross lines that link points with the same Y-index at right angles). There are m points in X- and n points in Y-direction. Each grid point is identified by an index l, with  $l = (j-1) \cdot m + i$ , where j is the number of the row and i the number of the column the grid point lies on. l ranges from 1 until  $m \cdot n = L$ . See fig. (2.2.1).

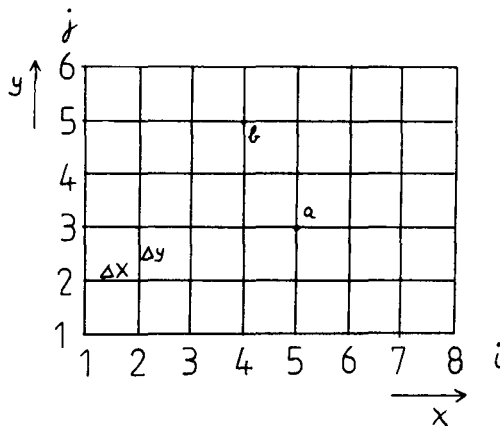


Fig.(2.2.1): Regular grid point distribution in a 8x6 points grid.  $m = 8$ ,  $n = 6$ . In a  $l = 21$ , in b  $l = 36$ .  $l$  is given by :  $l = (j-1) \cdot m + i$ .

If we consider a grid point  $l$  with its 8 direct adjacent points (fig.2.2.2), the vorticity in one of these points can be written as a Taylor series in the vorticity of the central point. For instance, if  $l$  is the index of the central point:

$$\omega_{l+1} = \omega_l + \Delta x \cdot \left. \frac{\partial \omega}{\partial x} \right|_l + \frac{1}{2} \Delta x^2 \cdot \left. \frac{\partial^2 \omega}{\partial x^2} \right|_l + \frac{1}{3!} \Delta x^3 \cdot \left. \frac{\partial^3 \omega}{\partial x^3} \right|_l + O(\Delta x^4) \quad (2.6)$$

and

$$\omega_{l-1} = \omega_l - \Delta x \cdot \left. \frac{\partial \omega}{\partial x} \right|_l + \frac{1}{2} \Delta x^2 \cdot \left. \frac{\partial^2 \omega}{\partial x^2} \right|_l - \frac{1}{3!} \Delta x^3 \cdot \left. \frac{\partial^3 \omega}{\partial x^3} \right|_l + O(\Delta x^4) \quad (2.7)$$

where  $O(\Delta x^4)$  means, that the remainder of the series is of size  $\Delta x^4$ .

This is the so-called truncation error of the Taylor series. If we add (2.6) and (2.7) we get:

$$\omega_{l+1} + \omega_{l-1} = 2 \cdot \omega_l + \Delta x^2 \cdot \left. \frac{\partial^2 \omega}{\partial x^2} \right|_l + O(\Delta x^4) \quad (2.8)$$

Reordering of the terms gives:

$$\left. \frac{\partial^2 \omega}{\partial x^2} \right|_l = \frac{\omega_{l+1} + \omega_{l-1} - 2 \cdot \omega_l}{\Delta x^2} - O(\Delta x^2) \quad (2.9)$$

Here we have the second derivative of the vorticity with respect to  $x$ , written in terms of the value of  $\omega$  in the points  $l-1$ ,  $l+1$  and  $l$  itself.

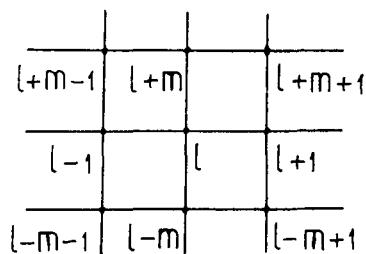


Fig.(2.2.2): Central grid point  $l$  with its 8 direct adjacent points.

Writing down Taylor series for the other grid points in the same manner and subtracting or adding them in such a way that certain terms vanish, gives us the discretisation schemes for a variety of derivatives:



$$\left. \frac{\partial \omega}{\partial x} \right|_1 = \frac{\omega_{1+1} - \omega_{1-1}}{2 \cdot \Delta x} + O(\Delta x^2) \quad (2.10)$$

$$\left. \frac{\partial \omega}{\partial y} \right|_1 = \frac{\omega_{1+m} - \omega_{1-m}}{2 \cdot \Delta y} + O(\Delta y^2) \quad (2.11)$$

$$\left. \frac{\partial^2 \omega}{\partial y^2} \right|_1 = \frac{\omega_{1+m} + \omega_{1-m} - 2 \cdot \omega_1}{\Delta y^2} + O(\Delta y^2) \quad (2.12)$$

$$\left. \frac{\partial^2 \omega}{\partial x \partial y} \right|_1 = \frac{\omega_{1-m-1} - \omega_{1-m+1} - \omega_{1+m-1} + \omega_{1+m+1}}{4 \cdot \Delta x \cdot \Delta y} + O(\Delta x^2 + \Delta y^2) \quad (2.13)$$

If we now want to turn the Vorticity Diffusion Equation (VDE) into a differential scheme, all the derivatives in the equation have to be replaced by their discrete approximation, for instance by equations (2.9) up to and including (2.12), if a second order truncation error is satisfactory.

The time derivative in the VDE is replaced by an approximation with first order truncation error:

$$\left. \frac{\partial \omega}{\partial t} \right|_{1,t} = \frac{\omega_1^n - \omega_1^{n-1}}{\Delta t} + O(\Delta t) \quad (2.14)$$

This is called the backward time difference. The upper index refers to the time:  $t_n = t_0 + n \cdot \Delta t$ , if no upper index is used, a time  $t_n$  (upper index  $n$ ) is assumed.

For the convective term in the VDE a special differential scheme is used in the programmes for the half infinite actuator strip. This scheme is described in section 3.2: 'The convective term in the differential scheme'.

The complete substitution in the VDE gives:

$$\begin{aligned} \frac{\omega_1 - \omega_1^{n-1}}{\Delta t} = \frac{1}{\rho_0} \left[ \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right] - u \cdot \frac{\omega_{1+1} - \omega_{1-1}}{2 \cdot \Delta x} - v \cdot \frac{\omega_{1+m} - \omega_{1-m}}{2 \cdot \Delta y} + \\ + v \cdot \frac{\omega_{1+1} + \omega_{1-1} - 2 \cdot \omega_1}{\Delta x^2} + v \cdot \frac{\omega_{1+m} + \omega_{1-m} - 2 \cdot \omega_1}{\Delta y^2} + O(\Delta x^2 + \Delta y^2 + \Delta t) \end{aligned} \quad (2.15)$$

Regrouping of the terms gives:

$$\begin{aligned} - \omega_1^{n-1} - \frac{\Delta t}{\rho_0} \left[ \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right] = \Delta t \cdot \left[ \frac{v}{2 \cdot \Delta y} + \frac{v}{\Delta y^2} \right] \cdot \omega_{1-m} + \Delta t \cdot \left[ \frac{u}{2 \cdot \Delta x} + \frac{v}{\Delta x^2} \right] \cdot \omega_{1-1} \\ + \left[ \Delta t \cdot \left[ \frac{-2 \cdot v}{\Delta x^2} - \frac{2 \cdot v}{\Delta y^2} \right] - 1 \right] \cdot \omega_1 + \Delta t \cdot \left[ \frac{-u}{2 \cdot \Delta x} + \frac{v}{\Delta x^2} \right] \cdot \omega_{1+1} + \Delta t \cdot \left[ \frac{-v}{2 \cdot \Delta y} + \frac{v}{\Delta y^2} \right] \cdot \omega_{1+m} \end{aligned} \quad (2.16)$$

With general coefficients the differential scheme for the Vorticity Diffusion Equation reads:

$$BO_1 \cdot \omega_{1-m} + DO_1 \cdot \omega_{1-1} + EO_1 \cdot \omega_1 + FO_1 \cdot \omega_{1+1} + HO_1 \cdot \omega_{1+m} = JO_1 \quad (2.17)$$

This is a linear function that connects the value of  $\omega$  in each grid point  $l$  with the value of  $\omega$  in its four direct adjoining grid points.

For this reason the scheme is called the five-point molecule scheme.

The naming of the coefficients is systematically: the coefficients belonging to the value in the grid point  $l-m-1$  begin with an A, the coefficients belonging to the value in grid point  $l-m$  with B etcetera.

The second letter is either an O for coefficients of the vorticity diffusion equation (omega) or a P for the Poisson-PSI equation (psi).

The whole grid consists of  $L$  grid points, meaning that  $L$  equations have to be solved. If we write these set of  $L$  equations as one matrix equation, we obtain:

$$\bar{A} \cdot \vec{\omega} = \vec{j} \quad (2.18)$$

With  $\bar{A}$  a  $L \times L$  matrix and  $\vec{\omega} = ( \omega_1, \omega_2, \omega_3, \omega_4, \dots, \omega_L )$  and  $\vec{J} = ( J_{O_1}, J_{O_2}, J_{O_3}, J_{O_4}, \dots, J_{O_L} )$ .

The matrix  $\bar{A}$  is a so-called pentadiagonal matrix; this is a sparse matrix with only five diagonals nonzero: the main diagonal and its 2 direct adjacent diagonals, plus the two diagonals at a distance  $\pm m$  from the main diagonal. ( see fig.2.2.3) Each row  $l$  in the matrix is filled with the coefficients belonging to the discretized differential equation for gridpoint  $l$ .

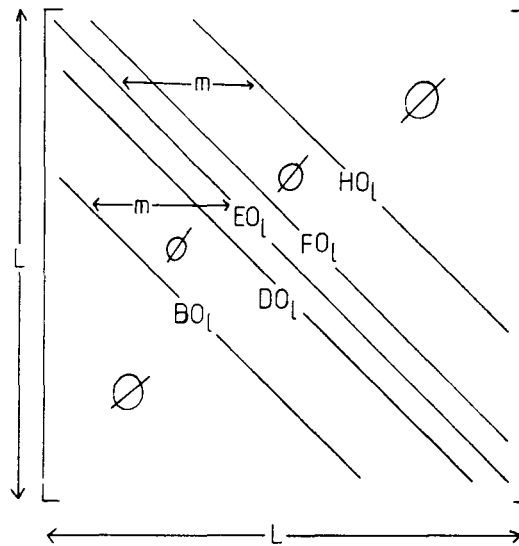


Fig.(2.2.3): Schematic representation of pentadiagonal matrix with  $L$  rows and  $L$  columns. Each row contains the coefficients of eq. 2.17 for the  $l$ -th gridpoint.

There are several numerical methods for solving these special kinds of matrix equations very quickly. The method used in this work is a method based on Stone's Strongly Implicit Method.(see [ref.3]).The flow through the half infinite strip is solved with the symmetric, strongly implicit method as described in [ref.4], and the flow through the finite actuator strip is solved using the modified strongly implicit method, described in [ref.5].

The replacement of the Poisson-PSI equation (PPE) by its differential scheme ( also a five-point molecule scheme) is done in the same way.

The result is:

$$-\psi_1 = \frac{1}{\Delta y^2} \psi_{1-m} + \frac{1}{\Delta x^2} \psi_{1-1} + \left[ \frac{-2}{\Delta x^2} - \frac{2}{\Delta y^2} \right] \cdot \psi_1 + \frac{1}{\Delta x^2} \psi_{1+1} + \frac{1}{\Delta y^2} \psi_{1+m} \quad (2.19)$$

This set of equations is solved with the same method.

The velocity components, finally, are calculated with:

$$u_1 = \frac{\psi_{1+m} - \psi_{1-m}}{2 \cdot \Delta y} + O(\Delta y^2) \quad (2.20)$$

$$v_1 = \frac{\psi_{1-1} - \psi_{1+1}}{2 \cdot \Delta x} + O(\Delta x^2) \quad (2.21)$$

The differential schemes all have a truncation error of order  $(\Delta x^2 + \Delta y^2 + \Delta t)$ . This truncation error provides us with a means to check the correctness of the differential scheme itself. If, for instance, the time step is halved, keeping  $\Delta x$  and  $\Delta y$  constant, the truncation error also has to reduce to 50%. If we again reduce the time step with 50% and call the solutions to the equations with different time step  $a_r$ ,  $a_{r+1}$  and  $a_{r+2}$  respectively, we can compare the differences in truncation error by subtracting the solutions.

If the errors are called  $\epsilon$ , we have:

$$\epsilon_{r+2} - \epsilon_{r+1} = a_{r+2} - a_{r+1} \quad (2.22)$$

$$\epsilon_{r+1} - \epsilon_r = a_{r+1} - a_r \quad (2.23)$$

And if, as assumed  $\epsilon_{r+1} = \frac{1}{2} \epsilon_r$ ,  $\epsilon_{r+2} = \frac{1}{2} \epsilon_{r+1}$ , then

$$\frac{\epsilon_{r+2} - \epsilon_{r+1}}{\epsilon_{r+1} - \epsilon_r} = \frac{1}{2} \quad (2.23a)$$

The convergence rate of an iterative process is defined as:

$$C_f = \frac{a_{r+2} - a_{r+1}}{a_{r+1} - a_r} \quad (2.24)$$

If  $C_f = \frac{1}{2}$ , then we can say that the method is of order zero with respect to time and the truncation error is of order 1.

The same procedure can be used to find the order of the solution method with respect to the spatial coordinates. The differential scheme suggests a truncation error of order 2 for both X- and Y-direction.

### 2.3 The boundary conditions.

As has already been mentioned, the partial differential equations can only be solved, if the behaviour of the function is given at the boundary. This can be understood quite easily, if we regard a grid point on the boundary ( fig. 2.3.1). If some equation (e.g.(2.17)) for the point  $l$  is to be solved, the value of  $\omega$  at  $l-1$  has to be known.

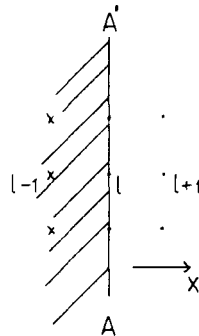


Fig.(2.3.1): Grid point  $l$  on boundary  $A-A'$ . The function value in points beyond the boundary can be expressed in function values of points inside by use of the boundary condition.

Normally, this value can be derived from values of points inside the domain by application of the boundary condition. For instance, if the line  $A-A'$  is a symmetry line the boundary condition would be  $\frac{\partial \omega}{\partial x} = 0$  on

the boundary.

In the computation of the flow through the finite actuator strip (ch.4), a similar boundary condition is used: there the X-axis is an anti-symmetry line and  $\omega_{1-m} = -\omega_{1+m}$ , if  $l$  is the index of a grid point that lies on the X-axis. The boundary conditions for the velocity are symmetrical:  $v = 0$  and  $\frac{\partial u}{\partial y} = 0$ .

This leads to  $\psi = 0$  on the X-axis.

The other boundary condition that has been used is the far-field condition: the vorticity as well as the velocity are zero far away from the actuator strip. This means  $\omega_l = 0$ ,  $u_l = 0$ ,  $v_l = 0$ ,  $\psi_l = 0$  if grid point  $l$  on the boundary, plus  $\psi_l = 0$ , if grid point  $l$  is immediately next to the boundary.

In case of nonzero undisturbed velocity  $\vec{U} = U \cdot \vec{e}_x + V \cdot \vec{e}_y$  the boundary condition for the streamfunction becomes:

$$\psi = U \cdot y - V \cdot x \quad \text{with } \psi(0,0) = 0 \quad (2.25)$$

The situation of nonzero undisturbed velocity can be compared with e.g. an ascending or descending helicopter or a windturbine.

3. The half-infinite actuator strip.

3.1 Model, source term and grid.

Model, source term.

If one edge of an actuator strip is held fixed on the origin and the other is moved to  $y = -\infty$  while keeping the mean force density distribution constant, we get the half-infinite actuator strip. See fig.(3.1.1)

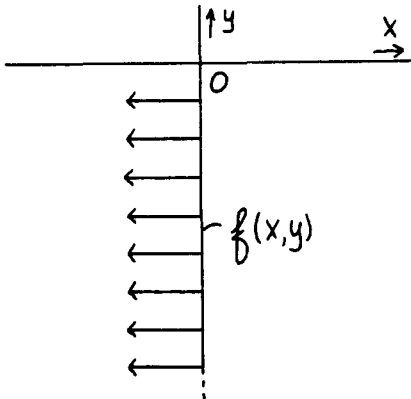


Fig.(3.1.1): Place of the edge and direction of the force for the half-infinite actuator strip

The first term on the right-hand side of equation (2.2) is the source term: we see that vorticity is only produced where a change of force density occurs. Because the half-infinite actuator strip only has one edge and the force has only an X-component, the only place where vorticity is produced, is at the origin. This production is given by:

$$\text{vorticity production} = - \frac{\partial f(x,y)_x}{\rho_o \cdot \partial y} \quad (3.1)$$

The functional form of  $f(x,y)$  is already mentioned in chapter 1, formulae (1.1) and (1.3). Combined they give:

$$f(x,y) = - \frac{f_o \cdot n \cdot (1 - \tanh(n \cdot y))}{\rho_o \cdot 4 \cdot \cosh^2(n \cdot x)} \quad (3.2)$$

If the value of  $n$  is taken 4.6 , then at distances 2 from the centre of the source the force density has diminished to less than 1/1000 th. of the maximum value.

If we for a moment only consider the production of vorticity by the source and leave out the influence of diffusion and convection, the produced vorticity field will have the shape of a peak in the origin. This peak is an approximation of the vorticity field of a point vortex. So the corresponding velocity field is approximately the field of a point vortex in the origin, especially in regions further away from the origin. Because the streamlines of a point vortex are concentric circles there will be no transportation of vorticity in radial direction by convection. This is true only as a result of diffusion.

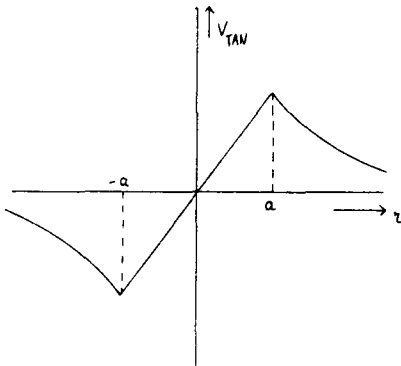


Fig.(3.1.2): Velocity field of a vortex with 'solid core'.

Another functional form of the source term is proposed by Gijs van Kuik [ref.6] and reads:

$$f = f_o \cdot \frac{1}{2} \left[ 1 - \frac{y \cdot y^2}{|y| a^2} \right] \quad \text{for } -a < y < a \quad (3.3)$$

$$f = f_o \quad \text{for } y \leq -a \quad (3.4)$$

$$f = 0 \quad \text{for } y \geq a \quad (3.5)$$

This form is chosen because application in the inviscid flow equations gives a velocity field that is identical to the velocity field of a



vortex with a 'solid core'. Only the tangential velocity is nonzero and is given in fig.(3.1.2). In the core with radius  $a$  the velocity is proportional to the distance from the origin, this is the velocity pattern of a rotating, massive body. Outside the core the velocity decreases proportional to  $1/r$ ,  $r$  being the distance to the origin. This is the velocity field of a point vortex in the origin. In the calculations a value for  $a$  of 1.5625 is taken, this is the distance between the third gridpoint and the origin. The source term for the source with solid core now reads:

$$\frac{\partial f}{\partial y} = - \frac{2.3}{\cosh^2(4.6 \cdot x)} \cdot \frac{y \cdot y^2}{|y| a^2} \quad (3.6)$$

this formula from now on will be referred to as the solid core source term.

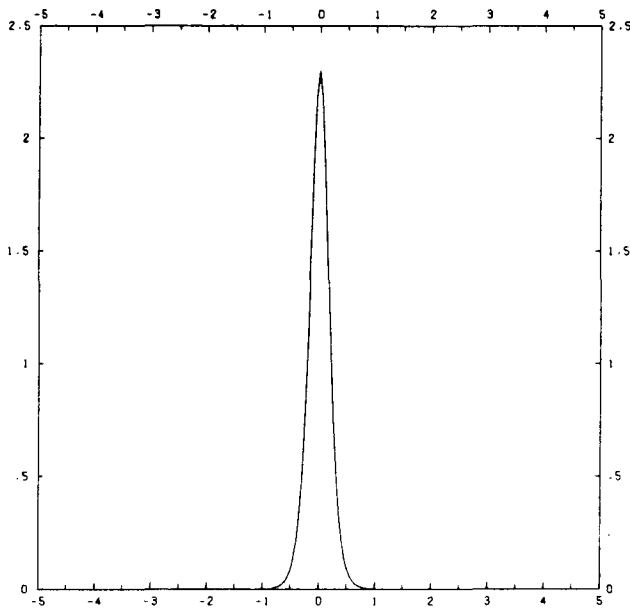


Fig.(3.1.3): Force density distribution in y-direction for  $\cosh^2$  source term.

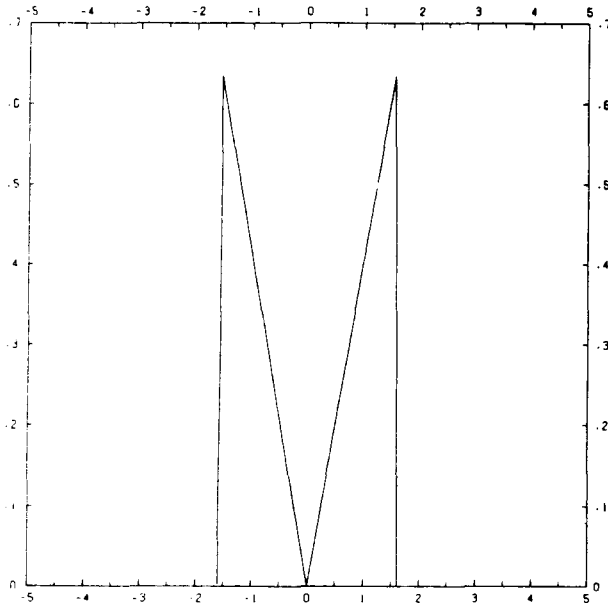


Fig.(3.1.4): Force density distribution in y-direction for solid core source term.

In the X-direction, a  $\cosh^2$ -dependence of the force density is introduced, as in the  $\cosh^2$  source term. The figures (3.1.3) and (3.1.4) give the force density as a function of y for the  $\cosh^2$ -source and the solid core source respectively. Notice that the area beneath the function in both cases is 1.

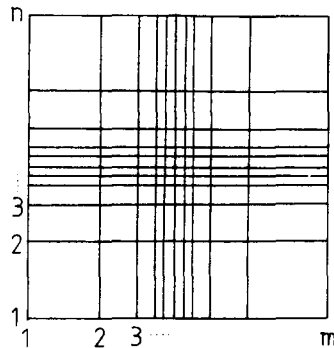


Fig.(3.1.5): Grid point distribution for computation of the flow through the half infinite actuator strip.

Grid.

The grid points in the computational grid are arranged in a special way. The grid consists of straight lines connecting points with same X- or Y-index. The distance between grid points decreases when the gridpoints approach either X- or Y-axis (see fig.3.1.1). The x- and y-values range between -100 and +100. Each row of gridpoints has y-coordinate  $y_j$ ,  $j$  between 1 and  $n$ , and each column has x-coordinate  $x_i$ ,  $i$  between 1 and  $m$ .  $x_i$  is given by:

$$x_i = 100 \cdot \left[ \frac{i-(m+1)/2}{(m-1)/2} \right]^{(2 \cdot NVF + 1)} \quad (3.7)$$

where NVF is a positive integer.

The same relation holds for  $y_j$ :

$$y_j = 100 \cdot \left[ \frac{j-(n+1)/2}{(n-1)/2} \right]^{(2 \cdot NVF + 1)} \quad (3.8)$$

When NVF = 1 we already have a strong decrease in distance between gridpoints when approaching the axes. (square proportionality). For a 33 x 33 point grid, as was used in the computations, the x- (and y-) values for the consequent gridpoints are given in table 3.1.

$i, (j)$	$x_i (y_j)$
1 (33)	-100
2 (32)	-82.998
3 (31)	-66.992
4 (30)	-53.638
5 (29)	-42.188
6 (28)	-32.495
7 (27)	-24.414
8 (26)	-17.798
9 (25)	-12.5
10 (24)	-8.374
11 (23)	-5.273
12 (22)	-3.052
13 (21)	-1.563
14 (20)	-0.659
15 (19)	-0.195
16 (18)	-0.024
17	0

Table 3.1: Values of  $x_i$  and  $y_j$  for the gridpoints in a 33 x 33 grid with NVF=1. For the points 18 until 33 the values of the coordinates are positive.

3.2 The convective term in the differential scheme.

For calculation of the flow through the half-infinite actuator strip, the differential schemes of the flow equation, as described in section 2.2, are used, with one exception though. The convective term in the vorticity-diffusion equation has been discretised according to Roache [ref.1,p.73] in the so-called second upwind differencing scheme. This method assures conservation of vorticity as well as conservation of direction of convection.

The convective term is assumed to be quasi-stationary. The vorticity in one grid point is developed as a series in the vorticity of neighbouring points in a way depending on the direction of the velocities in these points. The convective term can be written as:

$$(\vec{u} \cdot \vec{\nabla}) \cdot \omega = u \cdot \frac{\partial \omega}{\partial x} + v \cdot \frac{\partial \omega}{\partial y} \quad (3.9)$$

If we take the first term on the right-hand side and write it according to the second upwind differencing scheme:

$$u \cdot \frac{\partial \omega}{\partial x} = \frac{u_r \omega_r - u_l \omega_l}{KX} \quad (3.10)$$

with

$$u_r = \frac{1}{2} [u_{i+1,j} - u_{i,j}] \quad (3.11)$$

$$u_l = \frac{1}{2} [u_{i,j} - u_{i-1,j}] \quad (3.12)$$

see fig.(3.2.1).

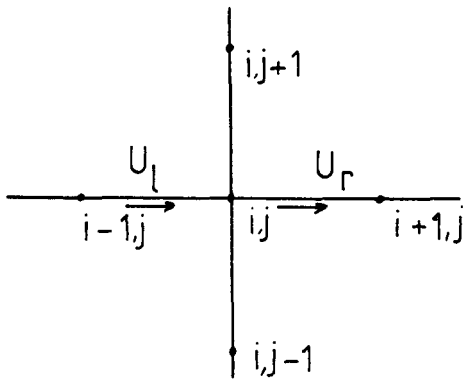


Fig.(3.2.1): definition of  $U_L$  and  $U_R$

The values of  $KX$ ,  $\omega_r$  and  $\omega_l$  are chosen depending on  $u_l$  and  $u_r$  :

$$u_r \geq 0 \text{ and } u_l \geq 0 : KX = x_i - x_{i-1}$$

$$u_r < 0 \text{ and } u_l < 0 : KX = x_{i+1} - x_i$$

$$\text{otherwise: } KX = \frac{1}{2}(x_{i+1} - x_{i-1})$$

and:

$$u_r \geq 0 : \omega_r = \omega_{i,j}$$

$$u_r < 0 : \omega_r = \omega_{i+1,j}$$

$$u_l \geq 0 : \omega_l = \omega_{i-1,j}$$

$$u_l < 0 : \omega_l = \omega_{i,j}$$

The scheme for  $v \cdot \frac{\partial \omega}{\partial y}$  is the same, with  $i$  and  $j$  interchanged and  $x$  replaced by  $y$ .

### 3.3 Calculations and results.

On the half infinite actuator strip several calculations have been made. The bulk of these calculations was used to determine the convergence of the differential scheme. This is described in the next section.

For the strips with  $\cosh^2$ -source term and solid-core source term (see section 3.1) calculations have been made in a  $33 \times 33$  point grid, with  $NVF = 1$ . (see section 3.1).

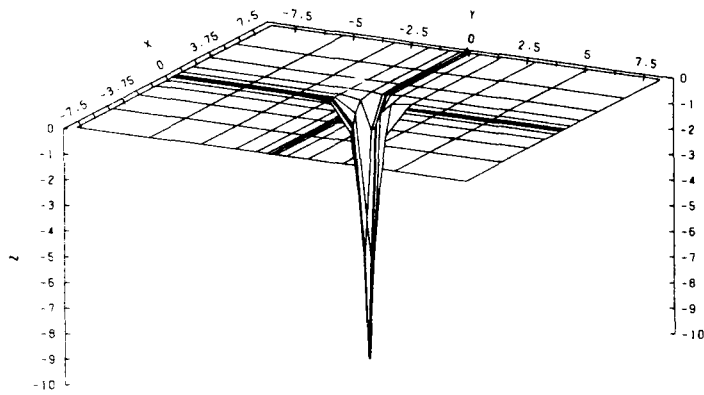


Fig.(3.3.1): Vorticity field at  $t=1.5$  s. for the  $\cosh^2$  source term. Notice: vorticity is negative

Let's first turn to the strip with  $\cosh^2$ -source term.

Figure (3.3.1) gives a perspective view of the vorticity field that was calculated at  $t = 1.5$  seconds in 10 time steps. The strength of the source was 100; this means that  $f_0$  in formula (3.2) is 100. Notice that the grid is not completely shown, both  $x$  and  $y$  extend from  $-100$  to  $+100$ . The height of the peak is normalized to 10; the real value of the vorticity at the peak is  $\approx 40$ . Fig.(3.3.2) shows the corresponding streamline pattern. As expected, the streamlines clearly form concentric circles.

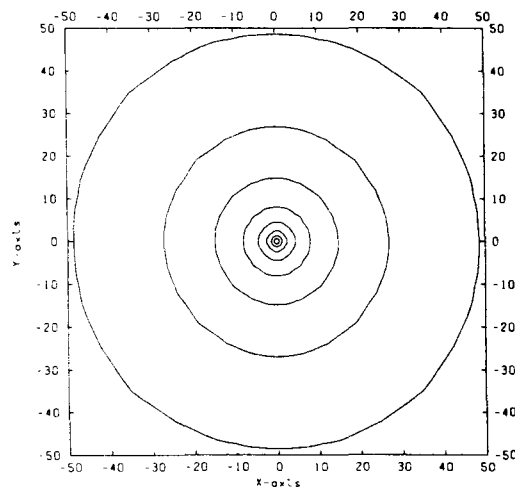
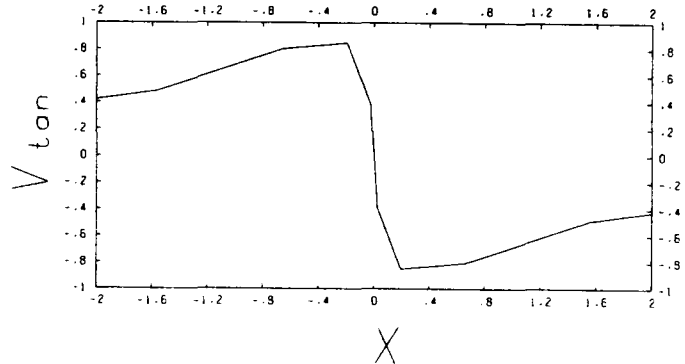
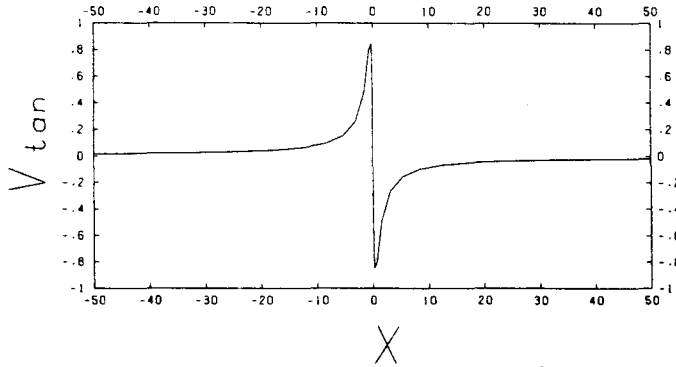


Fig.(3.3.2): Streamline pattern at  $t = 1.5$  s. for  $\cosh^2$  source.

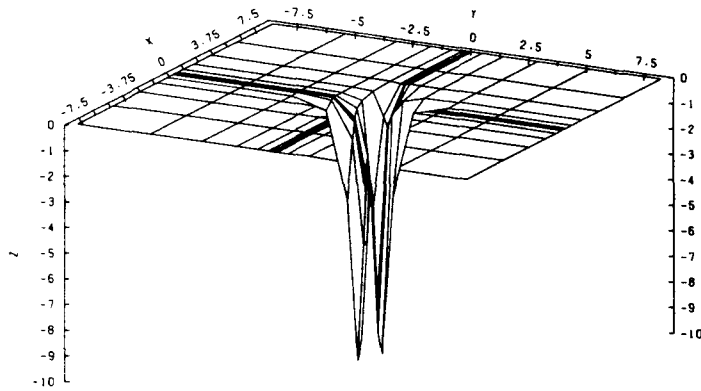
To find out, whether the tangential velocity has a reciprocal dependence of radial distance to the source, as is the case if the source is a real point vortex, we view figs. (3.3.3) and (3.3.4), which show the tangential velocity on the X-axis.



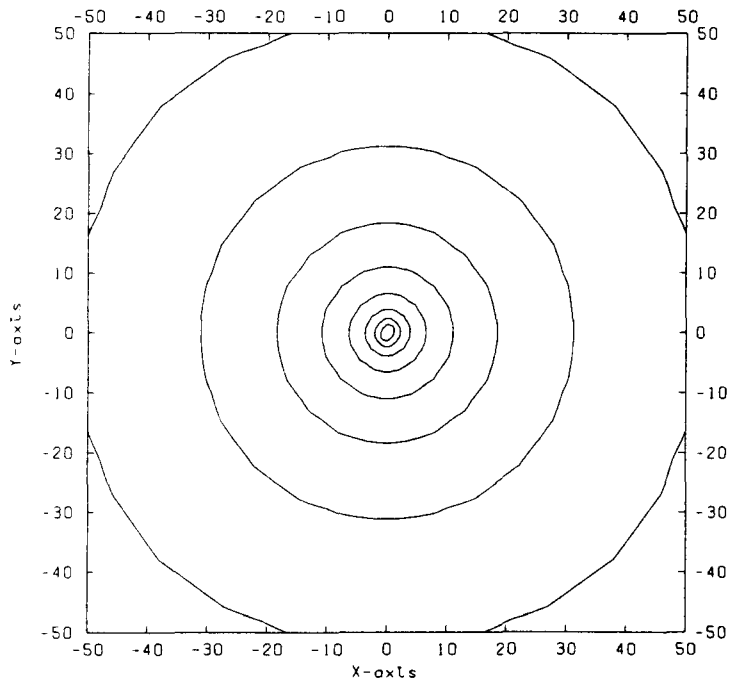
Figs. (3.3.3) and (3.3.4): Tangential velocity at  $t = 1.5$  seconds as a function of  $x$  for the  $\cosh^2$  source.

The same results for the strip with a solid core source term are shown in figs. (3.3.5) up to and including (3.3.8). Here the calculations are performed until  $t = 1.5$  seconds, in again 10 time steps and with the same strength of the source. Notice the double-peaked vorticity distribution in fig.(3.3.5).

*Fig.(3.3.5): Vorticity field at  $t = 1.5$  s for the solid core source term.*

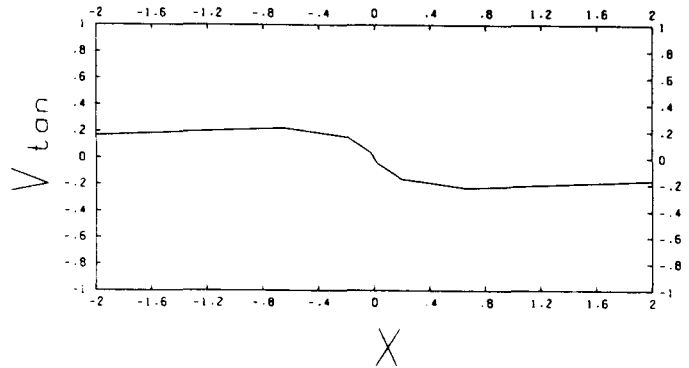
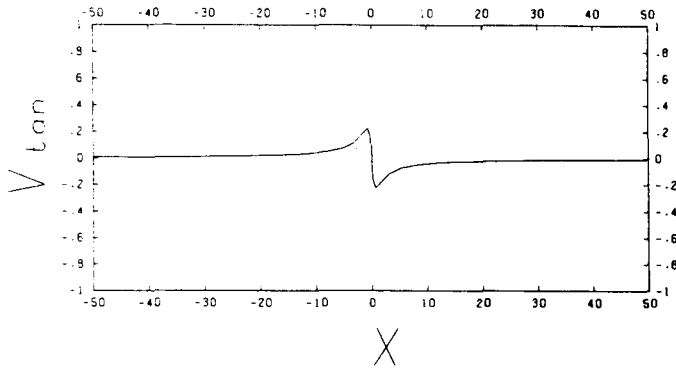


*Fig.(3.3.6): Streamline pattern at  $t = 1.5$  s. for the solid core source term.*





### 3. half-infinite strip



Figs.(3.3.7) and (3.3.8): Tangential velocity at  $t = 1.5$  s. for the solid core source.

Because the dependency of  $v_{tan}$  from  $r$ , the radial distance to the source, is rather difficult to derive from fig.(3.3.3) or (3.3.7), figure (3.3.9) gives the product of  $v_{tan}$  and  $r$  as a function of  $r$  for both types of source. We see that in a large area the product hardly changes and that the departure from a straight line at small  $r$  is earlier for the solid core source than for the  $\cosh^2$  source.

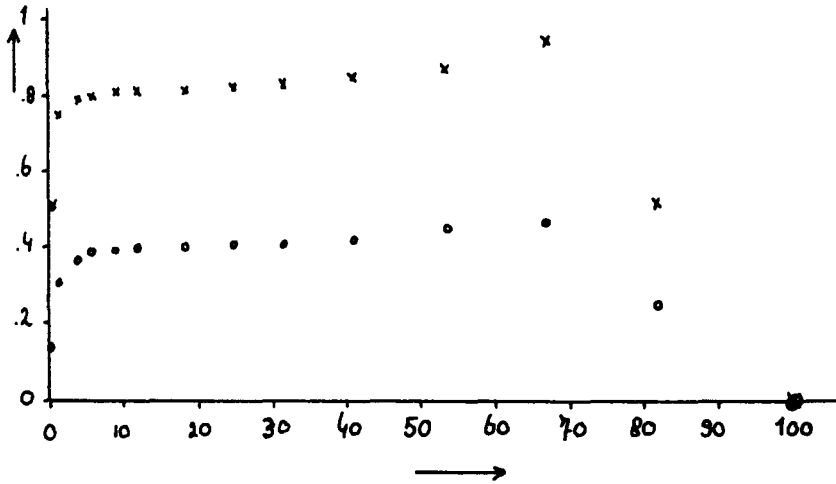


Fig.(3.3.9):  $V_{tan} \cdot r$  as a function of  $r$  for two different source terms in the half infinite actuator strip. crosses: solid core; dots:  $\cosh^2$ .

For large  $r$  the line is slowly inclining, but above  $r \approx 70$  the curve breaks down. This downfall for large  $r$  can be explained by the boundary condition for the velocity. These are both set to zero at the boundaries, so the velocity has to decrease at some point more quickly than the increase in  $r$ .

Here we arrive also at a possible explanation for the slight rise in the product  $v_{tan} \cdot r$  for intermediate  $r$ . Because the vorticity is always set zero at the boundaries, there is no possibility for vorticity to escape the domain. Consequently, the velocity will increase more than it would, if vorticity could diffuse or convect out of the domain.

This effect could be avoided, if either a large domain were chosen, or if a smaller ending time of the calculations were chosen.

If we look at the course of  $v_{tan} \cdot r$  for small  $r$  in fig.(3.3.9), we see that the departure from constant  $v_{tan} \cdot r$  happens at larger  $r$  for the solid core source. This indicates that, although the same grid is used

for both types of source, the solid core source has a larger core (i.e. a region where  $v_{tan} \cdot r$  is not constant) than the  $\cosh^2$ -core. Probably, the size of the latter is dictated by the size of the mesh near the origin.

0.100	-1.433	-14.330
0.200	-2.866	-14.329
0.300	-4.299	-14.329
0.400	-5.731	-14.328
0.500	-7.164	-14.328
0.600	-8.597	-14.328
0.700	-10.029	-14.327
0.800	-11.461	-14.327
0.900	-12.894	-14.326
1.000	-14.326	-14.326
1.100	-15.757	-14.325
1.200	-17.189	-14.324
1.300	-18.620	-14.323
1.400	-20.050	-14.321
1.500	-21.478	-14.319
1.600	-22.904	-14.315
1.700	-24.328	-14.310
1.800	-25.744	-14.302
1.900	-27.150	-14.290
2.000	-28.551	-14.275

Table 3.2: Value of the vorticity as a function of time. First column: time; second column: vorticity; third column: vorticity over time.

As a final result, table 3.2 is given. Here the vorticity in the origin as a function of time is given. The calculations were made in a 11x11 point grid. As can be read in the third column, the vorticity increases linearly as a function of time.

### 3.4 Convergence.

The calculations on the half-infinite actuator strip have also been performed to gain some insight into the convergence of the iteration process. According to the differential scheme, the order of the method to be expected is 1 for the spatial coordinates and 0 for the time

coordinate. This order can be calculated by comparing the results of computations with consequently diminishing steps in x- or y- or t-direction.

For instance, if we want to know the order of the truncation error of the method for the x-coordinate, we first solve the problem in a  $9 \times 9$  point grid and in, e.g., 2 time steps. Then we again solve the problem with the same parameters, except for the number of steps in x-direction, which is doubled. The grid is then  $17 \times 9$  points. Then the problem is solved for the third time, now in a  $33 \times 9$  point grid.

For each point in the original  $9 \times 9$  point grid the convergence rate  $C_f$  can be calculated by applying equation (2.24). The order is minus the second logarithm of the convergence rate. For order 1 and 2 of the truncation error this factor has to be  $1/2$  and  $1/4$ , respectively.

The convergence rate in a grid point was only calculated if the difference between consequent solutions was larger than the accuracy in the calculation.

These accuracies are EPSOM and EPSPSI in the calculation of the vorticity diffusion equation and the Poisson-PSI equation, respectively.

In the following three tables, the mean value of the calculated convergence rates, the standard deviations and the number of points in which the rates could be determined are printed.

For the determination of a set of convergence rates three calculations had to be performed, each next calculation with a double number of grid points in one direction.

An asterisk means, that the convergence rate in at least one grid point

was negative.

At the top of each column the number of grid points that was used to determine the convergence rates is printed.

x-direction (17 points in y-direction, 2 time steps)

	9 /17/33	17/33/65	33/65/129	Cf expected	
$\omega$	$\overline{Cf}$	9.2	0.45*	0.17*	0.25
	$\sigma(Cf)$	9.2	1.43	0.42	
	n	5	5	3	
$u$	$\overline{Cf}$	0.06*	1.39	0.33	0.25
	$\sigma(Cf)$	0.08	1.86	0.04	
	n	44	44	40	
$v$	$\overline{Cf}$	0.23*	0.75	0.29	0.25
	$\sigma(Cf)$	0.26	1.03	0.3	
	n	43	44	41	

$$\begin{aligned} \epsilon_{\omega} &= 10^{-8} \\ \epsilon_{\psi} &= 10^{-5} \end{aligned}$$

Table 3.3: Convergence rates in x-direction.

y-direction (33 points in x-direction, 2 time steps)

	9/17/33	17/33/65	Cf expected	
$\omega$	$\overline{Cf}$	0.134	4.454	0.25
	$\sigma(Cf)$	0.16	-	
	n	3	1	
$u$	$\overline{Cf}$	0.27	0.52	0.25
	$\sigma(Cf)$	0.30	0.18	
	n	38	30	
$v$	$\overline{Cf}$	0.11	0.62	0.25
	$\sigma(Cf)$	0.01	0.13	
	n	30	26	

$$\begin{aligned} \epsilon_{\omega} &= 10^{-5} \\ \epsilon_{\psi} &= 10^{-4} \end{aligned}$$

Table 3.4: Convergence rates in y-direction.

### 3. half-infinite strip

time ( 17x17 grid points)

	4/8/16	8/16/32	16/32/64	Cf expected	
$\omega$	$\overline{Cf}$	0.87	0.02	0.68	0.5
	$\sigma(Cf)$	0.57	0.8	0.18	
	n	13	13	9	
$u$	$\overline{Cf}$	0.376	0.420	0.455	0.5
	$\sigma(Cf)$	0.003	0.005	0.005	
	n	44	44	44	
$v$	$\overline{Cf}$	0.376	0.420	0.454	0.5
	$\sigma(Cf)$	0.003	0.005	0.005	
	n	44	44	44	

$$\begin{aligned} \epsilon_{\omega} &= 10^{-10} \\ \epsilon_{\psi} &= 10^{-6} \end{aligned}$$

Table 3.5: Convergence rates in t-direction.

For diminishing step in x-direction, as well as in y-direction, the convergence rates in the calculation of the vorticity are not as good as could be expected from the order of the method. Probably this is due to the small number of grid points included in the calculation of the convergence rate.

The convergence rates for the velocities approach the expected values. In case of diminishing time step, the convergence rate for the vorticity are better, and the rates for the velocities are even very good.

#### 3.5 Comparison with an analytical solution.

If the vorticity diffusion equation is written in cylindrical coordinates  $(r, \theta, z)$  without convection term, it reads:

$$\frac{\partial \omega}{\partial t} = -\nu \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial \omega}{\partial r} \right) = f(r, \theta, t) \quad (3.5.1)$$

If  $f(r, \theta, t)$  is a unit point source in time and in space ( $r=0, t=0$ ), the solution to (3.5.1) is:

$$\omega(r, \vartheta, t) = \frac{1}{v t} \exp(-r^2/4vt) \quad (3.5.2)$$

(see [ref.15], pp.258-)

If the  $\delta$ -function form is maintained in spatial coordinates, but the source is turned on from  $t = 0$  until  $t = t'$ , then the source term reads:

$$f(r, \vartheta, t) = \delta_r(0)H(0, t') \quad (3.5.3)$$

where  $H(0, t')=1$  for  $0 \leq t \leq t'$  and  $H(0, t')=0$  otherwise.

The solution to the equation (3.5.1) then is:

$$\omega(r, \vartheta, t) = \int_0^{t'} \frac{\exp(-r^2/4vt)}{vt} dt = \frac{1}{v} \int_{-r^2/4vt'}^{\infty} \frac{\exp(-p)}{p} dp = \frac{1}{v} E_1(r^2/4vt') \quad (3.5.4)$$

$E_1$  is the exponential integral. This function is tabulated in e.g. [ref.16], pp.228-.

For  $p = 10$ ,  $E_1(p) = 4.1 \times 10^{-6}$ . This means, that for  $t' = 1$ , as in the numerical calculations, the vorticity at places with  $r^2/4v = 10$  has a value of about  $2.8 \times 10^{-1}$ . This is for  $r = 2.4 \times 10^{-2}$ .

In a  $33 \times 33$  point grid the smallest distance is approximately  $2 \times 10^{-2}$ . Here the calculated vorticity is  $-5.0$ . This is much larger than the value caused by a pure delta-function in the origin. The difference between these two values can be explained partly by the absence of diffusion in the analytical solution, and partly by the fact that the source term used in the calculations is an approximation of the Dirac  $\delta$ -function. Therefore the gradients in the solution can be expected to be much smaller.

If  $f(r, \vartheta, t)$  is given by:

$$f(r, \vartheta, t) = \frac{n}{2 \cosh^2(nr')} H(0, t') \quad (3.5.5)$$

the solution will be given by:

$$\omega(r, \vartheta, t) = \int_0^{\infty} \int_0^{t'} \int_0^{2\pi} \frac{n \cdot \exp(-r^2 - r'^2 + 2rr' \cos(\vartheta - \vartheta') / 4vt')}{2 \cdot \cosh^2(nr')} d\vartheta' dt' dr'$$

(3.5.6)

This solution isn't worked out further.



#### 4.The (finite) actuator strip; Grid Generation.

##### 4.1 Model of the flow through the actuator strip.

The finite actuator strip has two edges: at  $x=0$ ,  $y= \pm D$ . In the ideal case, the force density at the edges falls from maximum to zero in an infinitesimal distance. The vorticity distribution formed in this case would consist of two infinitely high peaks at the edges.

In this model I use an approximation for the step in force density at the edges, given by eq.(1.3).

The vorticity will be formed in a small region around the edges. The area of this region depends on the parameter  $m$  in eq.(1.3): for increasing  $m$ , the area decreases and the maximum of the produced vorticity distribution increases.

The force exerted by the actuator strip will be assumed to point in positive  $x$ -direction and is turned on at  $t = 0$ . Then, according to equation (2.2), the vorticity produced at  $y = + D$  is positive and negative at  $y = -D$ .

The initial vorticity distribution has an accompanying velocity distribution: the peak of vorticity at one edge will create a velocity in the  $+ x$ -direction at the opposite edge and vice versa. Thus the vorticity produced at the edges of the strip will initially be transported in  $+x$ -direction.

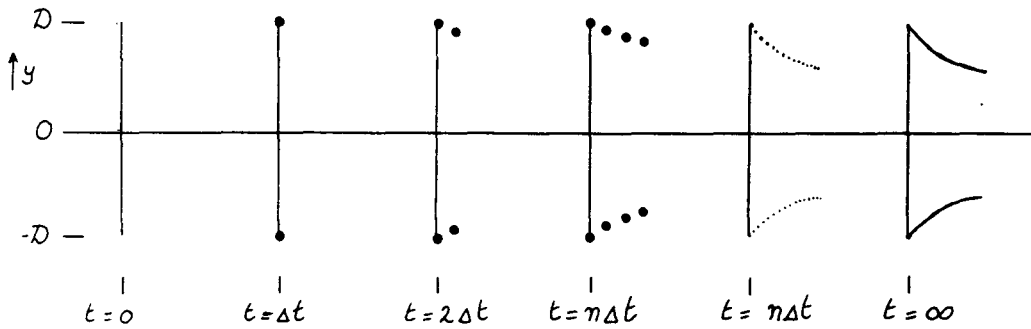
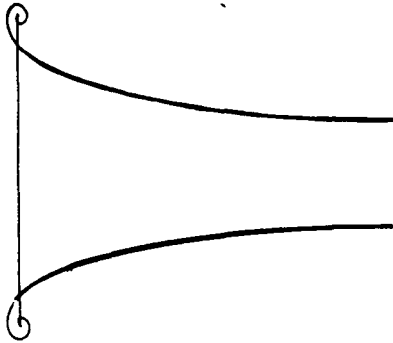


Fig.(4.1.1): Process of formation of vortex sheets at the edges of the actuator strip

This process can also be made clear as follows. If we for a moment think that the vorticity that is produced at one edge in one time step, is concentrated in a single point vortex, it will be moved in  $+x$ -direction by the velocity field of the point vortex at the opposite edge. In the following time step, two new point vortices will be created at the edges and each vortex will be transported by the sum of velocities of the other three vortices. After a while, a whole chain of point vortices is formed to the right of each edge. ( see fig.(4.1.1.)). If we let the time step decrease to zero, the distance between the individual point vortices will also become zero and the vortices will form a vortex sheet. This vortex sheet has zero thickness and infinite vorticity. In the model the vorticity is nowhere infinite and the vortex sheet has finite thickness, and is represented by a line of local maxima in the vorticity distribution.

After some time, a steady state situation will occur and the vortex sheets formed will have a constant shape. In this situation, the velocity at the edge of the strip is not unambiguously pointing in  $x$ -direction; the vortex sheet has a spiraling form in the vicinity of

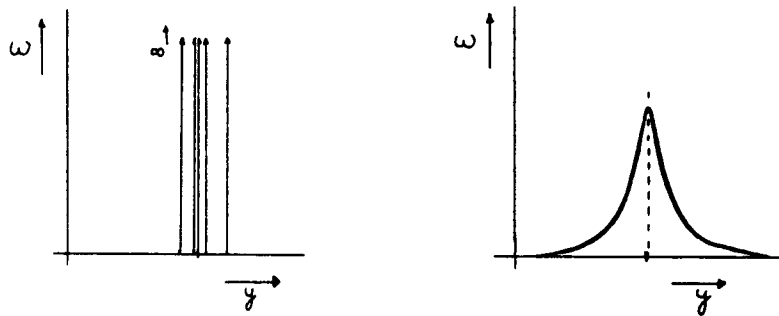
the edge.



*Fig.(4.1.2): Shape of vortex sheets in steady state.*

The flow through an actuator strip without diffusion in the stationary state can also be calculated analytically. The problem then is transformed into a complex plane. For an actuator strip with constant force density distribution Schmidt & Sparenberg [ref.10] find as a solution a vortex sheet that spirals into the edge of the actuator strip (see fig.4.1.2). This solution is discussed by Gijs van Kuik [ref.6] who also finds a spiraling solution but with another behaviour in the limit of going to the edge. In these works, further references can be found to other works on this subject.

With the underlying model one is not able to obtain such a spiraling solution at the edges because of the minimum mesh size. The vorticity in a grid point will always be some average value of the analytical solution. The vorticity distribution for  $x = 0$ ,  $t = \infty$  as a function of  $y$  is sketched in figs.(4.1.3) and (4.1.4). The first gives the analytical solution, the latter the average of the first. Figure (4.1.3) is a cross-section through the different branches of the vortex sheet, that spirals into the edge. The solution in (4.1.4) will be more accurate, if more grid points are used in those regions where the vorticity changes most.



Figs.(4.1.3) and (4.1.4): Vorticity distribution as a function of  $y$  for  $x = 0$ . left: analytical solution; infinitely high peaks; right: mean value of analytical solution.

Because we are interested in the starting behaviour of the flow through the actuator strip, we have to use the time-dependent equation (2.2). In this non-stationary case the vorticity produced at the edges is transported by the induced velocity field, so the vorticity distribution changes every time step.

If the flow is to be computed with maximum accuracy, the grid points have to be redistributed every time step in order to keep track of the moving vorticity distribution. So we would like to have a method that makes a grid of which the gridpoints are distributed dependent on the vorticity distribution. This method is called adaptive grid generation and is described in general in the next section.

#### 4.2 Numerical Grid Generation.

If a problem is to be solved numerically in a domain with complicated boundaries, numerical grid generation can be used to transform the domain into another domain with simple boundaries. For instance, in fig.(4.2.1) a domain with curved boundaries is transformed into a

rectangular domain.

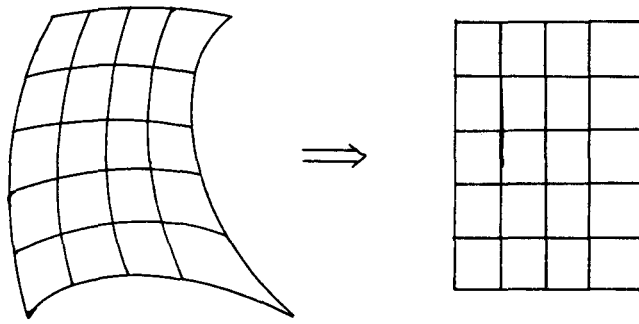


Fig.(4.2.1): Grid generation used to transform curved boundaries into straight ones.

Another possibility to use grid generation is in those situations where the solution is strongly varying throughout the domain. Grid generation then can be used to concentrate the gridpoints at places where for instance the solution has a large gradient (see fig.(4.2.2)).

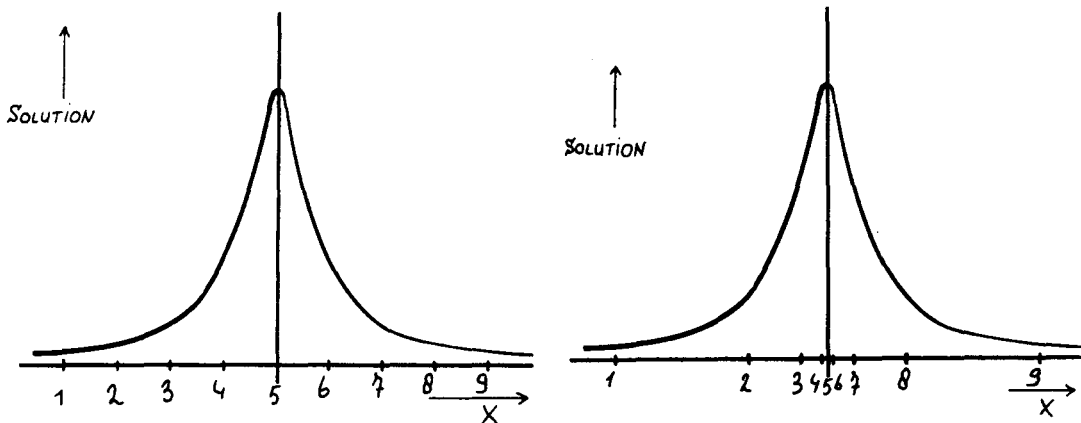


Fig.(4.2.2): Grid generation used to redistribute grid points on x-axis according to solution.

In general, grid generation leads to a quicker solution of the problem. In some cases grid generation offers a solution, and working with a regular grid doesn't.

Some general information about grid generation is given in [ref.7] and [ref.8]. However, the approach used in this work is much more simple and straightforward.

In the underlying problem, grid generation is used to redistribute the grid points over the domain. This redistribution is done every time step dependent on the solution obtained in the previous step and is therefore called adaptive grid generation.

Grid generation implies a transformation from a grid with grid points distributed in a special way (the physical grid) into a regular, orthogonal grid (the computational grid). The coordinates of the transformed grid  $(\xi, \eta)$  are written as a function of the coordinates in the physical grid  $(x, y)$ :

$$\begin{aligned} \xi &= \xi(x, y) \\ \eta &= \eta(x, y) \end{aligned} \quad \Bigg| \quad (4.2.1)$$

These two functions define the transformation.

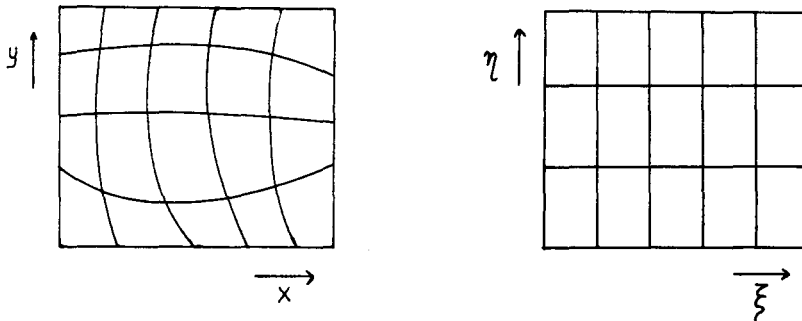


Fig.(4.2.3): Transformation of physical grid (left) into computational grid.

In figure (4.2.3) at the left, the physical grid is drawn, with drawn in it some lines  $\xi = \text{constant}$  and  $\eta = \text{constant}$ . These lines are transformed into straight lines in the computational grid. The difficulty is to construct functions  $\xi$  and  $\eta$  properly.

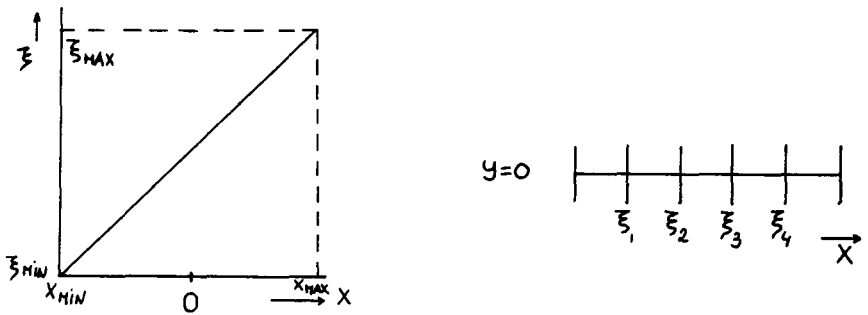
In analytical grid generation,  $\xi$  and  $\eta$  are written explicitly as a combination of analytical functions of  $x$  and  $y$ , for instance:

$$\xi = x^{1/n} \quad , \quad \eta = y \quad (4.2.2)$$

Because  $\xi$  and  $\eta$  are explicit functions of  $x$  and  $y$ , it's possible to

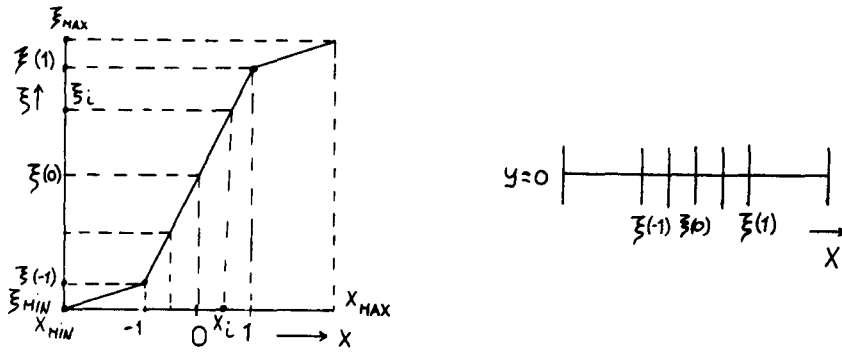
find the inverse transformations  $x = x(\xi, \eta)$  and  $y = y(\xi, \eta)$  in a general way. If then the computational grid is set up, the physical grid can be calculated by applying the inverse transformation.

Another way of generating a grid is numerical grid generation. Now only relationships between individual points in physical and computational grid are given. The inverse transformation can be found by interpolation or by fitting a general function (e.g. a power series) to the set of points.



Figs.(4.2.4) and (4.2.5): Relation between  $\xi$  and  $x$  linear.  $\xi$  is equally distributed on the  $x$ -axis.

Let's regard the relationship between  $\xi$  and  $x$  for fixed  $y$  (e.g.  $y = 0$ ). For  $\xi$  and  $x$  both regularly distributed over the domain, the transformation is linear and can be depicted as a straight line in the  $x$ - $\xi$  plane.( fig.(4.2.4)). If  $\xi_{min} = x_{min}$  and  $\xi_{max} = x_{max}$  , the partial derivative of  $\xi$  with respect to  $x$  is 1 everywhere, meaning that the lines  $\xi = \text{constant}$  are equally distributed in the physical grid for  $y = 0$ . (fig.4.2.5).



Figs. (4.2.6) and (4.2.7): Contraction of grid points between -1 and 1. Here the partial derivative of  $\xi$  with respect to  $x$  is larger than in the rest of the domain.

If we want to have more grid points in the physical grid between, for example,  $x = -1$  and  $x = 1$ , we can choose the partial derivative of  $\xi$  with respect to  $x$  larger in this interval. This also implies a smaller  $\xi_x$  in the rest of the domain because:

$$\int_{x_{min}}^{x_{max}} \xi_x dx = \xi_{max} - \xi_{min} = \text{constant} \quad (4.2.3)$$

If only the values of  $\xi$  and  $x$  are known in the points  $x = x_{min}$ ,  $-1$ ,  $1$ , and  $x_{max}$ , the  $x$ -values of the other points in the computational grid can be found by interpolating between the known values.

For instance, linear interpolation in fig.(4.2.6) gives for  $x_1$ :

$$x_1 = -1 + \frac{\xi_1 - \xi(-1)}{\xi(1) - \xi(-1)} \cdot (1 - (-1)) \quad (4.2.4)$$

In general, if  $i$  is the index for the subsequent points on the  $x$ - or  $\xi$ -axis:

$$x' = x_{i-1} + \frac{\xi' - \xi_{i-1}}{\xi_i - \xi_{i-1}} (x_i - x_{i-1}) \quad (4.2.5)$$

where  $\xi'$  lies between  $\xi_i$  and  $\xi_{i-1}$ .

We see that in numerical grid generation choosing the transformations amounts in essence to choosing the partial derivatives ( $\xi_x$  and  $\eta_y$ ). This can be done in many ways, as long as the integral ( as in



eq.(4.2.3)) of the partial derivative is constant and the partial derivative itself is nonzero everywhere.

For example, if more grid points are wanted in those regions where the gradient of the solution is large, we take the partial derivatives proportional to the absolute value of the gradient:

$$\xi_x = \alpha \cdot \left| \frac{\partial \omega}{\partial x} \right| + \beta \tag{4.2.6}$$

$\alpha$  is determined by the integral in eq.(4.2.3),  $\beta$  is added to ascertain that  $\xi_x > 0$  everywhere.

I will return to the choice of the transformation in the next section.

In the following, I will assume that in the physical grid for each grid point  $l$  the coordinates  $x_l, y_l, \xi_l$  and  $\eta_l$  are known, plus the values of the partial derivatives  $\xi_x, \xi_y, \xi_{xx}, \xi_{yy}, \xi_{xy}$  and  $\eta_x, \eta_y, \eta_{xx}, \eta_{yy}, \eta_{xy}$  are known. I defer to the next section the explanation of how the partial derivatives can be found.

Before a differential equation, for instance the vorticity diffusion equation, can be solved in the computational grid it has also to be transformed. With given partial derivatives of the transformation, the partial derivatives of the equation transform as:

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} \tag{4.2.7}$$

$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} \tag{4.2.8}$$

$$\frac{\partial^2}{\partial x^2} = \xi_{xx} \frac{\partial}{\partial \xi} + \xi_x^2 \frac{\partial^2}{\partial \xi^2} + \eta_{xx} \frac{\partial}{\partial \eta} + \eta_x^2 \frac{\partial^2}{\partial \eta^2} + 2 \cdot \xi_x \eta_x \frac{\partial^2}{\partial \xi \partial \eta} \tag{4.2.9}$$

$$\frac{\partial^2}{\partial y^2} = \xi_{yy} \frac{\partial}{\partial \xi} + \xi_y^2 \frac{\partial^2}{\partial \xi^2} + \eta_{yy} \frac{\partial}{\partial \eta} + \eta_y^2 \frac{\partial^2}{\partial \eta^2} + 2 \cdot \xi_y \eta_y \frac{\partial^2}{\partial \xi \partial \eta} \quad (4.2.10)$$

$$\frac{\partial}{\partial t} = \xi_t \frac{\partial}{\partial \xi} + \eta_t \frac{\partial}{\partial \eta} + \tau_t \frac{\partial}{\partial \tau} \quad (4.2.11)$$

I assume that the velocity of the grid is negligible and  $\tau = t$ .  $\xi_t$  and  $\eta_t$  are therefore zero and the partial derivative with respect to time transforms into itself. This leads to an error in the solution. This error can be minimised by choosing the time steps small enough.

The vorticity diffusion equation (VDE) becomes transformed:

$$\begin{aligned} \frac{\partial \omega}{\partial t} = & Q(x,y) + \left[ -u' \xi_x - v' \xi_y + v \xi_{xx} + v \xi_{yy} \right] \cdot \frac{\partial \omega}{\partial \xi} + \\ & + \left[ -u' \eta_x - v' \eta_y + v \eta_{xx} + v \eta_{yy} \right] \cdot \frac{\partial \omega}{\partial \eta} + \left[ v \xi_x^2 + v \xi_y^2 \right] \cdot \frac{\partial^2 \omega}{\partial \xi^2} + \\ & + \left[ v \eta_x^2 + v \eta_y^2 \right] \frac{\partial^2 \omega}{\partial \eta^2} + 2 \cdot v \cdot \left[ \xi_x \eta_x + \xi_y \eta_y \right] \cdot \frac{\partial^2 \omega}{\partial \xi \partial \eta} \end{aligned} \quad (4.2.12)$$

where  $Q(x,y)$  is the source term; this is a scalar and isn't changed by the transformation.  $u'$  and  $v'$  are the transformed velocities:

$$u' = \frac{\partial \Psi}{\partial y} = \xi_y \frac{\partial \Psi}{\partial \xi} + \eta_y \frac{\partial \Psi}{\partial \eta} \quad (4.2.13)$$

$$v' = - \frac{\partial \Psi}{\partial x} = - \xi_x \frac{\partial \Psi}{\partial \xi} - \eta_x \frac{\partial \Psi}{\partial \eta} \quad (4.2.14)$$

If  $Q$ ,  $u'$ ,  $v'$  and the partial derivatives of  $\xi$  and  $\eta$  are known, the coefficients of the partial derivatives of  $\omega$  can be calculated. Then the equation can be substituted by a differential scheme and solved in the same way as described in chapter 2, but now in the  $(\xi, \eta)$  grid. Because in the transformed equation (4.2.12) a term with the mixed derivative  $\frac{\partial^2}{\partial \xi \partial \eta}$  appears, eq.(2.13) also has to be used in the

differential scheme. The solution of the set of equations then needs a nine-point molecule difference scheme, because the values of  $\omega$  in the grid points with index  $l-m-1$ ,  $l-m+1$ ,  $l+m-1$  and  $l+m+1$  are also included. In [ref.5], a fast solution method for a nine-point molecule difference scheme is introduced.

#### 4.3 Application of grid generation to the actuator strip.

In this section I describe the practical realisation of the numerical adaptive grid, used to calculate the flow through the actuator strip.

The first choice was to let  $\xi$  only depend on  $x$ , this means that lines of constant  $\xi$  are allways straight lines and parallel to the  $y$ -axis if drawn in the  $x$ - $y$  plane.  $\eta$  is a function of both  $x$  and  $y$ .

Furthermore, the boundaries of the physical and the computational grid have been chosen the same.

The grid is set up in the following way:

first the vorticity diffusion equation is solved in a regular grid, the solution  $\omega(x,y)$  serves as a base for the set-up of the new grid.

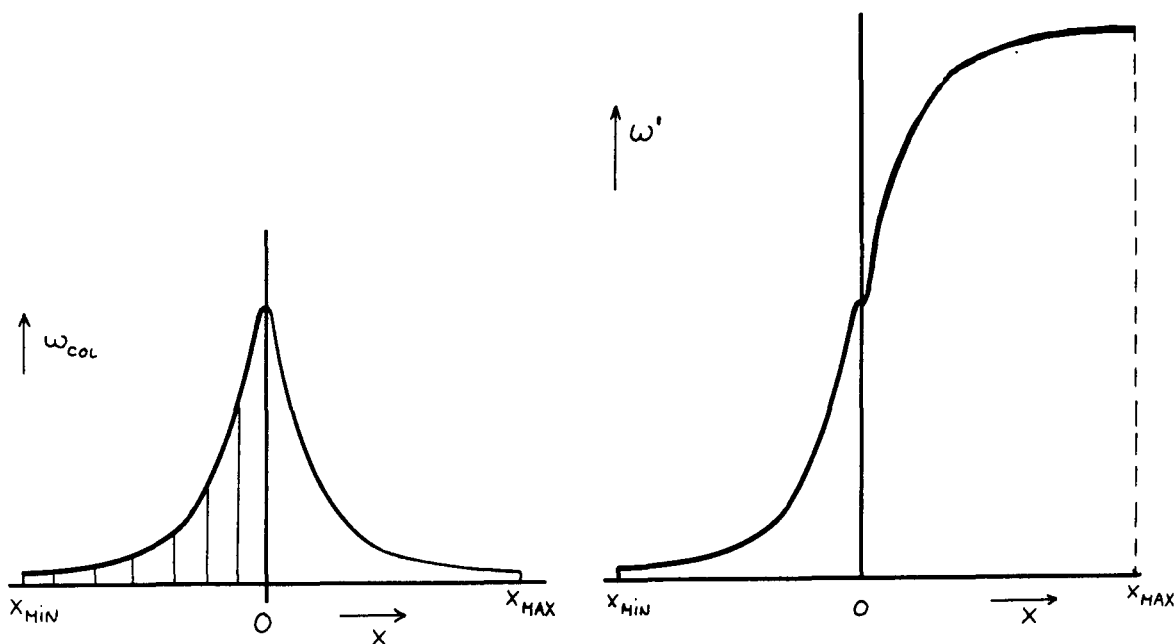


Fig.(4.3.1): The sum of vorticity in a column of grid points as a function of  $x$ .

Fig.(4.3.2): Same figure, but now with only positive gradient.

The vorticities in the grid points in each column  $i$  ( $x = x_i =$  constant) are made positive and added, giving an  $\omega_{col,i}$  for each  $i$ . This  $\omega_{col}$  is not a measure for vorticity, but only a quantity that relates the vorticity with the grid. The values of  $\omega_{col}$  are now altered in such a way, that the differences are always positive, see figs.(4.3.1) and (4.3.2). The difference between every two consequent  $\omega_{col}$  is calculated and negated if negative. In this way, the consequent  $\omega_{col}$  are always increasing. Simultaneously the difference is altered if it's too small or too large; in other words: in the whole domain the partial derivative of  $\xi$  with respect to  $x$  is checked. This  $\xi_x$  has to lie between a  $KSIXMIN$  and a  $KSIXMAX$ , the last two are given by the user, but both have to be positive. The altered values of  $\omega_{col}$  are called  $\omega'$ .

After this, the coupling between  $\xi$  and the  $\omega'$  is made:

$$\xi_i = A \cdot \omega_i' + B \tag{4.3.1}$$

A and B are determined in such a way that  $\xi$  ranges from XMIN to XMAX.

If we take the derivative of equation (4.3.1) with respect to x we get:

$$\frac{\partial \xi_i}{\partial x} = A \cdot \frac{\partial \omega_i'}{\partial x} = A \cdot \frac{\partial |\omega_{col i}|}{\partial x} \tag{4.3.2}$$

stated in words: the density of grid points in x-direction is proportional to the absolute value of the gradient of the vorticity summed over a column. This means, that the grid points are placed where they are needed the most: in those regions where the gradient in the solution is large.

$\eta$  and the vorticity are coupled in a similar manner. There are, however, a few differences, which will be discussed in the following.

The expected solution of the problem is a vortex sheet that travels in positive x-direction. (see section 4.1). At the vortex sheet, the vorticity has a local maximum. Because we want to have more grid points in the neighbourhood of these maxima a function  $g(x)$  is constructed that connects the local maxima, so  $g(x)$  follows the shape of the vortex sheet.  $g(x)$  is constructed as follows: for every  $x_i$ , the y-coordinate of the maximum of  $\omega$  is calculated.  $g(x)$  is fitted to the y-coordinate of these maxima by a least-squares fit procedure.  $g(x)$  is written as the truncated series of  $D \cdot (1 - \tanh(x))$ ; if the truncation begins at the NPAR +1rst term,  $g(x)$  reads:

$$g(x) = D \cdot \left[ 1 - \sum_{i=1}^{NPAR} \frac{2 \cdot PAR(i) \cdot x}{\pi^2 (2 \cdot i - 1)^2 / 4 + x^2} \right] \tag{4.3.3}$$

NPAR is the number of terms in the series, PAR(i) is an array containing the values of the parameters for each term. The value of

PAR(i) is set by the least-squares fit procedure. If NPAR is infinite and PAR(i) = 1 for every i,  $g(x) = D \cdot (1 - \tanh(x))$ .

Figure (4.3.3) gives a possible shape for  $g(x)$ . Note that  $g(x)$  is point-symmetrical in  $x = 0, y = D$ .

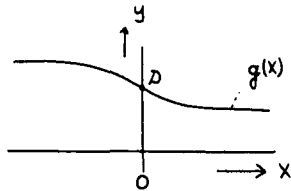


Fig.(4.3.3): Shape of  $g(x)$ , the line connecting local maxima in the vorticity distribution.

$\eta$  is coupled to the vorticity in such a way that the number of gridpoints below the line  $y = g(x)$  is equal to the number of gridpoints above the line  $y = g(x)$ . So  $y = g(x)$  is transformed to the line  $\eta = YMAX/2$ .

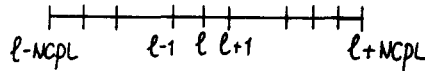


Fig.(4.3.4): The vorticity in NCPL neighbouring grid points is added to the vorticity in each grid point.

In order to give the lines  $\eta = \text{constant}$  in the physical grid a smooth course as a function of  $x$ , the vorticity in one grid point is increased with the vorticity in neighbouring points ( same row, other columns), according to:

$$\omega_l = \left[ \sum_{i=1}^{NCPL} \frac{\omega_{l+i}}{(i+1)^2} + \sum_{i=-1}^{-NCPL} \frac{\omega_{l+i}}{(-i+1)^2} \right]. \quad (4.3.4)$$

The larger NCPL (N couple), the smoother  $\eta = \text{constant}$  will be.

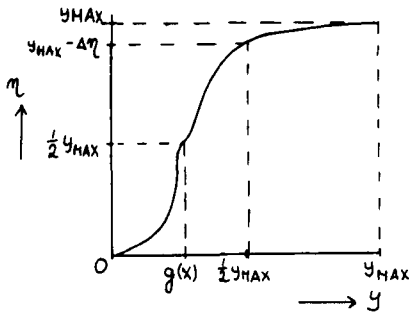


Fig.(4.3.5): Relation between  $\eta$  and  $y$ . The values in four points are fixed.

For the coupling the domain is split into two parts, above and below  $y = g(x)$ . In each column, four transformation values are fixed:

$$\left. \begin{aligned} \eta(x,0) &= 0 \\ \eta(x,g(x)) &= YMAX/2 \\ \eta(x,YMAX/2) &= YMAX - \Delta\eta \\ \eta(x,YMAX) &= YMAX \end{aligned} \right\} \quad (4.3.5)$$

These values are used to determine the coefficients  $\alpha_{1,i}$ ,  $\alpha_{2,i}$ ,  $\beta_{1,i}$  and  $\beta_{2,i}$  in the same way as A and B were determined in eq.(4.3.1):

Per column  $\omega_{i,j}$  is turned into  $\omega'_{i,j}$ , with the latter only increasing for increasing  $y$  and  $\frac{\partial \omega'}{\partial y}$  lying between a minimum and a maximum value (ETYMIN and ETYMAX). For  $y \leq g(x)$ :

$$\eta(x_i, y_j) = \alpha_{1,i} \cdot \omega'_{i,j} + \beta_{1,i} \quad (4.3.6)$$

and  $y \geq g(x)$ :

$$\eta(x_i, y_j) = \alpha_{2,i} \cdot \omega'_{i,j} + \beta_{2,i} \quad (4.3.7)$$

Again  $\alpha$  and  $\beta$  are chosen in such a way that  $\eta$  ranges between 0 and  $YMAX/2$  for  $\alpha_1$ ,  $\beta_1$  and between  $YMAX/2$  and  $YMAX$  for  $\alpha_2$  and  $\beta_2$ .

We see that the columns with  $i = 1, 2, (m+1)/2, m-1$  and  $m$  have always  $x$ -values  $XMIN, XMIN/2, 0, XMAX/2$  and  $XMAX$  respectively and that the rows with  $j = 1, (n+1)/2, n-1$  and  $n$  have  $y$ -coordinates  $0, g(x), YMAX/2$  and  $YMAX$ . Figure (4.3.6) illustrates this.

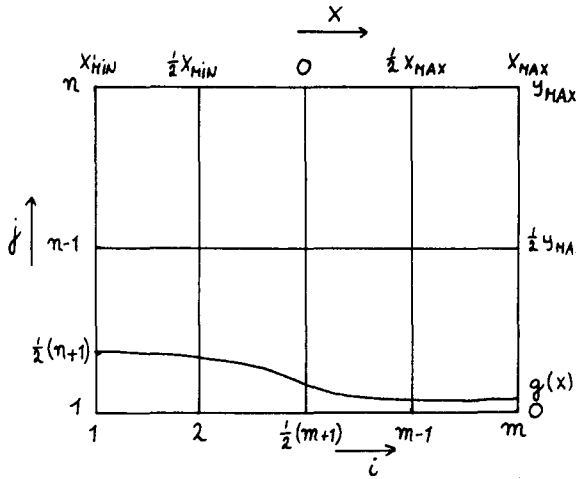
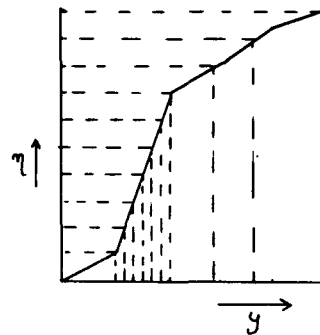
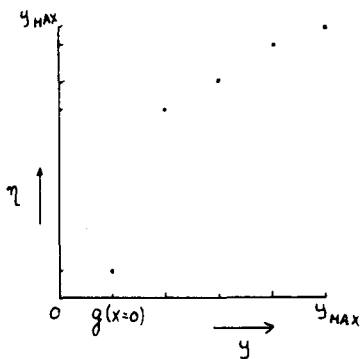


Fig.(4.3.6):  $\xi = \text{const.}$  and  $\eta = \text{constant}$  lines in the physical grid that are held fixed.

If the transformations of the individual grid points are known, the x- and y-coordinates of the new physical grid can be calculated. Figure (4.3.7) gives the relation between old y-coordinates (e.g. for  $x = 0$ ) and the new, computed  $\eta$ -coordinates. In figure (4.3.8) these points are connected by straight line pieces. Here we see that the y-values of the new physical grid belonging to a regular computational grid are interpolated from the  $\eta$ -values of the old grid.



Figs.(4.3.7) and (4.3.8): Method for calculation of new gridpoints. The solution in a regular physical grid are connected by straight line pieces. The new y-values of a regular computational grid are found by interpolation.

This interpolation is done  $m+1$  times: once for the  $\xi$ -x coordinates and  $m$  times for the  $\eta$ -y coordinates.



If the new grid is set up, some things have to be done before the calculation of the Poisson-PSI equation in the new grid can begin.

First, the vorticity in the new grid points has to be interpolated from the vorticity in the old points.

Second, the partial derivatives of  $\xi$  and  $\eta$  with respect to  $x$  and  $y$  plus higher order derivatives have to be determined for each grid point.

If 0 is the index for the grid points whose derivatives have to be calculated and 1 the index for a neighbouring point, Taylor development in 0 gives:

$$\begin{aligned} \eta_1 = \eta_0 + (x_1 - x_0) \left. \frac{\partial \eta}{\partial x} \right|_0 + (y_1 - y_0) \left. \frac{\partial \eta}{\partial y} \right|_0 + 1/2 \cdot (x_1 - x_0)^2 \left. \frac{\partial^2 \eta}{\partial x^2} \right|_0 + \\ + 1/2 \cdot (x_1 - x_0)^2 \left. \frac{\partial^2 \eta}{\partial y^2} \right|_0 + (x_1 - x_0) \cdot (y_1 - y_0) \left. \frac{\partial^2 \eta}{\partial x \partial y} \right|_0 + \dots \quad (4.3.8) \end{aligned}$$

If we write down the Taylor series for four other points ( index 2-5) the  $\eta_x$ ,  $\eta_{xx}$ ,  $\eta_y$ ,  $\eta_{yy}$  and  $\eta_{xy}$  can be solved from these five equations because  $\eta_1 - \eta_5$ ,  $x_1 - x_5$  and  $y_1 - y_5$  are known. The same is done for the partial derivatives of  $\xi$ . This means that for each grid point two sets of five equations, each set with five unknowns, have to be solved. However,  $\xi$  is a function of  $x$  only, so  $\xi_y$ ,  $\xi_{yy}$  and  $\xi_{xy}$  are zero everywhere. Furthermore  $\xi_x$  and  $\xi_{xx}$  are the same for every grid point in one column; this reduces the number of equations to be solved considerably.

5. The (finite) actuator strip; Calculations and results.

5.1 Equations, boundary- and starting conditions.

The transformed differential equations (eqs. 4.2.12-4.2.14) become more simplified because  $\xi$  is a function of  $x$  only. The transformed vorticity diffusion equation reads:

$$\begin{aligned} \frac{\partial \omega}{\partial t} = Q(x,y) + \left[ -u' \xi_x + v \xi_{xx} \right] \cdot \frac{\partial \omega}{\partial \xi} + \left[ -u' \eta_x - v' \eta_y + v \eta_{xx} + v \eta_{yy} \right] + \\ + v \xi_x^2 \frac{\partial^2 \omega}{\partial \xi^2} + \left[ v \eta_x^2 + v \eta_y^2 \right] \frac{\partial^2 \omega}{\partial \eta^2} + 2 \cdot v \cdot \xi_x \eta_x \frac{\partial^2 \omega}{\partial \xi \partial \eta} \end{aligned} \quad (5.1.1)$$

The transformed Poisson-PSI equation is:

$$\begin{aligned} -\omega = \xi_{xx} \frac{\partial \psi}{\partial \xi} + \xi_x^2 \frac{\partial^2 \psi}{\partial \xi^2} + \left[ \eta_{xx} + \eta_{yy} \right] \frac{\partial \psi}{\partial \eta} + \left[ \eta_x^2 + \eta_y^2 \right] \frac{\partial^2 \psi}{\partial \eta^2} + \\ + 2 \cdot \xi_x \eta_x \frac{\partial^2 \psi}{\partial \xi \partial \eta} \end{aligned} \quad (5.1.2)$$

and the transformed velocities:

$$u' = \eta_y \cdot \frac{\partial \psi}{\partial \eta} \quad (5.1.3)$$

$$v' = -\xi_x \cdot \frac{\partial \psi}{\partial \xi} - \eta_x \cdot \frac{\partial \psi}{\partial \eta} \quad (5.1.4)$$

The vorticity diffusion equation is solved on the old transformed grid, then a new grid transformation is set up on the basis of the found solution. On the new transformed grid the Poisson-PSI equation is solved and from the streamfunction the velocities are computed.

If the end time isn't reached and again a time step has to be taken, the loop of solving equations starts from the beginning.

The boundary conditions also have to be transformed. These conditions are straightforward for the boundaries  $x = XMIN$ ,  $x = XMAX$  and  $y = YMAX$ ; here  $\omega = 0$  on the boundary. Because  $u'$  and  $v'$  are taken zero on the boundary, the streamfunction  $\psi$  on, as well as next to the boundary is zero.

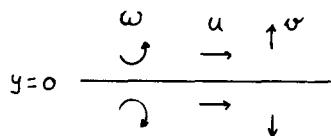


Fig.(5.1.1):  $\omega$  ,  $u$  and  $v$  reflected in the boundary  $y = 0$ .

The conditions on  $y = 0$  are different. This boundary acts as a mirror, in order to simulate the source at  $y = -D$ . See fig.(5.1.1). As can be seen in this figure, the vorticity below is minus the vorticity above the line  $y = 0$ . The same holds for  $v$ , the velocity in  $y$ -direction. The velocity in  $x$ -direction,  $u$ , doesn't change when crossing  $y = 0$ .

The foregoing leads to  $\omega = 0$ ,  $v = 0$  and  $\frac{\partial u}{\partial y} = 0$  on the boundary.

$\frac{\partial u}{\partial y} = 0$  leads to:

$$\eta_{yy} \frac{\partial \psi}{\partial \eta} + \eta_y^2 \cdot \frac{\partial^2 \psi}{\partial \eta^2} = 0, \quad \eta = 0 \quad (5.1.5)$$

As a consequence of this mirror boundary condition, a whole class of solutions to the equations are excluded. This is for example a solution with asymmetrical vortex shedding from the edges of the strip.

The starting conditions on  $t = 0$  are  $\omega = 0$ ,  $u = 0$ ,  $v = 0$  and  $\psi = 0$  on the whole domain.

The force density on the actuator strip is taken 1 ( $N/m^3$ ). The time step  $\Delta t$  is used as parameter to guide the tempo of vorticity production at the edge of the strip. The gradient of the force density at the edge of the strip can be altered by giving  $m$  in eq.(1.3) another value. This has the effect of broadening or tightening the peak in the vorticity

production at the edge of the actuator strip.

### 5.1 Flow chart, starting grid.

Figure (5.2.1) gives the flow chart of the numerical calculation of the flow through the finite actuator strip. Each square contains a separate action done by a subroutine. The name of the subroutine is placed between brackets. More information about these subroutines can be found in Appendix A2. The whole calculation can be directed by a set of calculation parameters, which are stored in a separate file. The use of these parameters makes it possible for instance first to calculate until  $t = 5$  seconds, then to view the results.

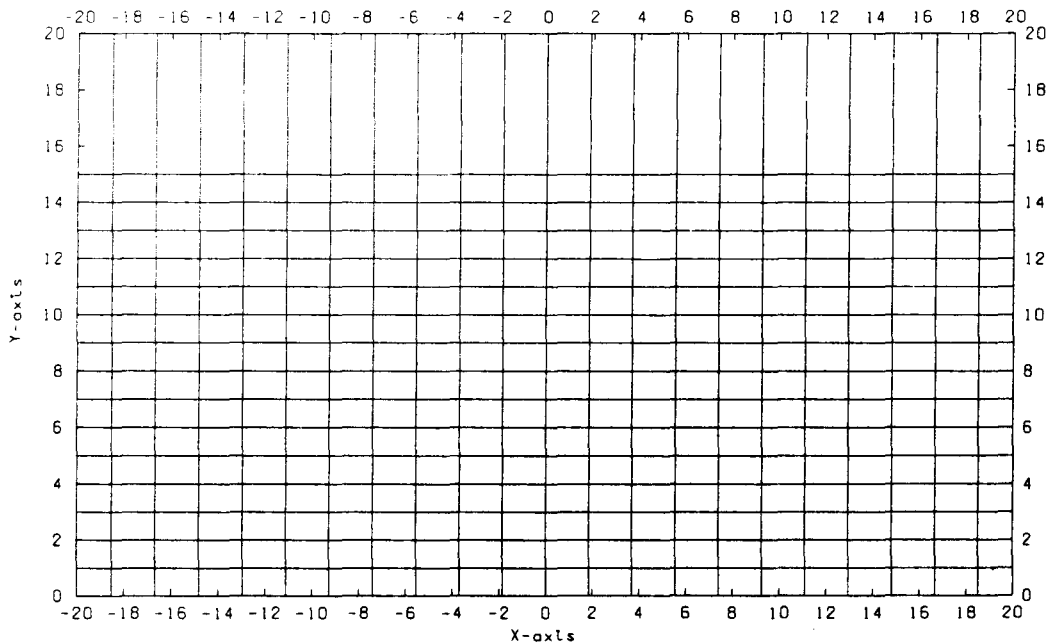


Fig.(5.2.2): Part of the starting grid with 41 by 21 grid points.

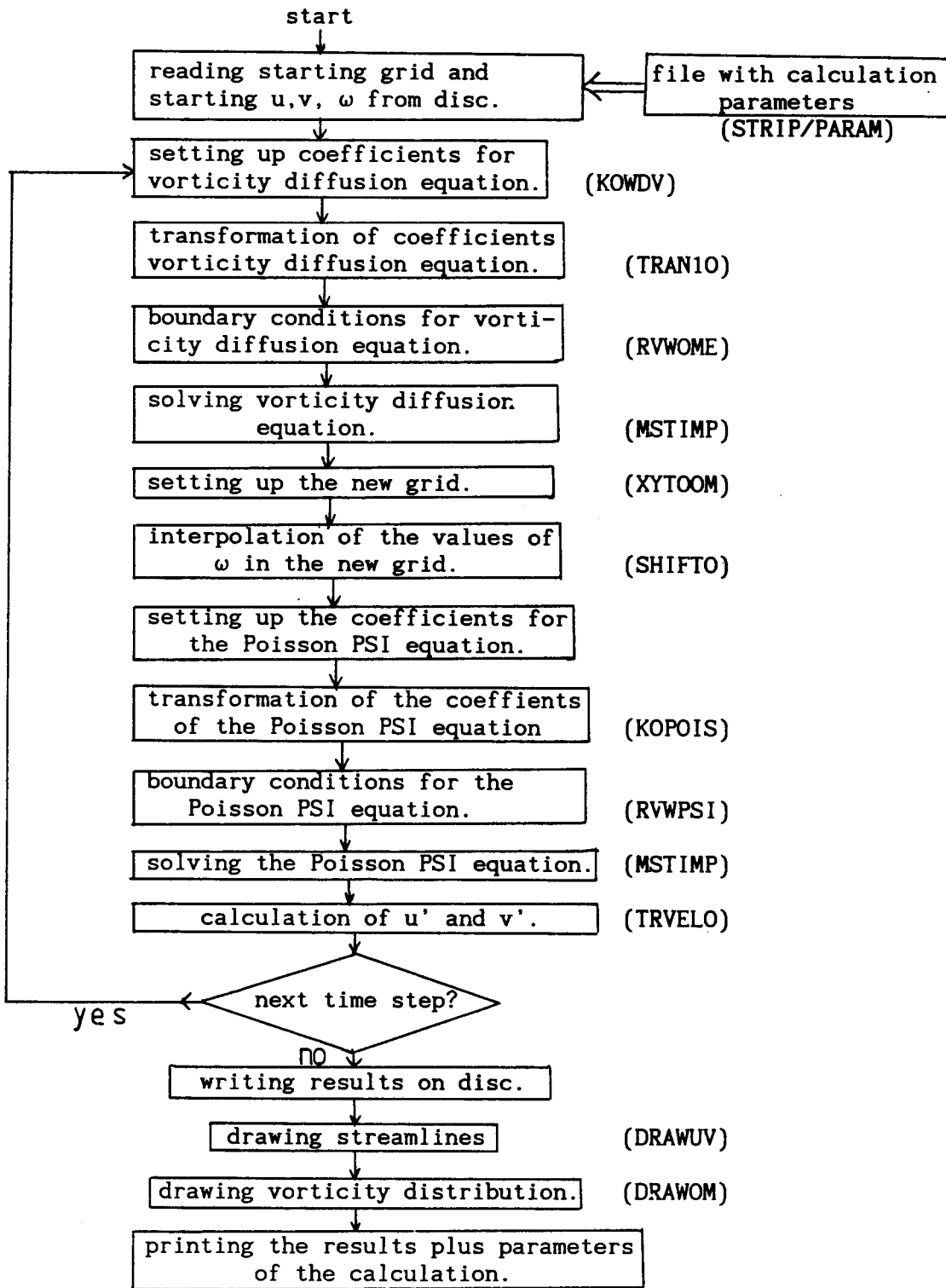


Fig.(5.2.1): Flow chart of the calculation of the flow through the finite actuator strip in the programme STRIP/DEVELOP.

If the results are satisfactory, a second calculation can be performed with the results of the first calculation as starting values.

Figure (5.2.2) gives the starting grid of 21 by 21 points. This grid is set up by the subroutine STAROP. Figure (5.2.3) gives a perspective view of the vorticity distribution at  $t = 0.1$  seconds, calculated in the starting grid.

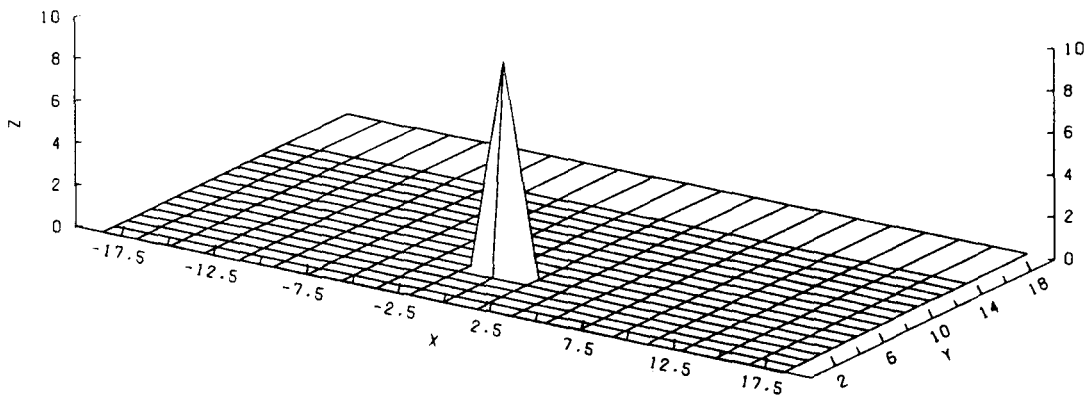


Fig.(5.2.3): Vorticity distribution in a 41 by 21 point regular grid at  $t=0.1$  seconds

In figure (5.2.4), the physical grid is drawn, that results from the grid generation subroutine XYTOOM. This 41 by 21 point grid is adapted to the vorticity distribution, shown in fig.(5.2.3).

The last figure, (5.2.5) gives the vorticity distribution in the adapted grid. This distribution results from linear interpolation of the vorticity distribution in fig.(5.2.3).

The results of calculations with the complete programme will be written in an addendum.

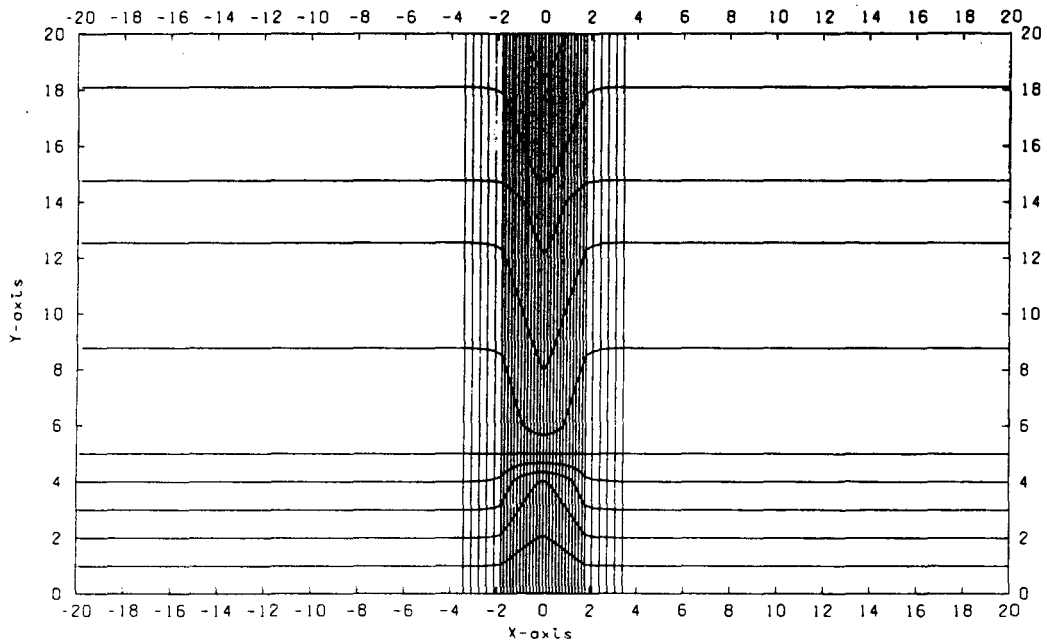


Fig.(5.2.4): Physical grid adapted to the vorticity distribution in the previous figure

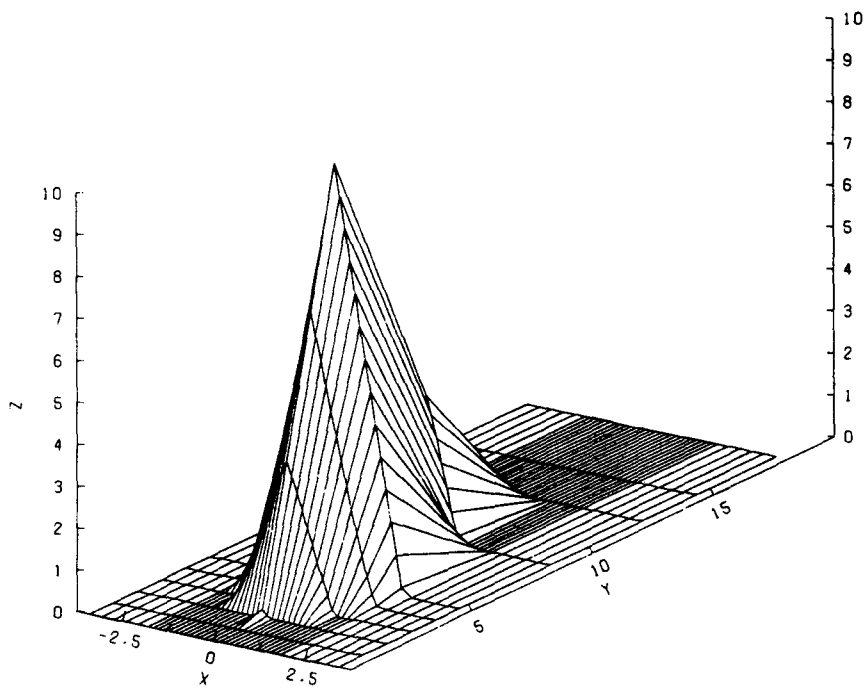


Figure (5.2.5): Vorticity distribution in the adapted grid.



---

## 6. Conclusions and Discussion.

The following conclusions can be drawn with respect to the calculation of the flow through the half infinite actuator strip.

- The calculated results conform to the theoretical expected situation; the vorticity, produced at the edge of the strip, is only transported as a result of diffusion and increases as a function of time. The streamlines are concentric circles around the edge of the strip.

- Only in regions near the edge of the strip the value of the vorticity is larger than the machine accuracy ( $\approx 1.2 \cdot 10^{-12}$ ). In spite of the fact that in the rest of the domain the value of the vorticity is smaller than this accuracy, the streamlines found here are also concentric circles.

- The convergence rates found for the Poisson-PSI equation (velocities) are good. However, the convergence rates found for the vorticity diffusion equation, are not as good as could be expected from the order of the method, possibly because too few gridpoints are involved in the calculation of the convergence rates.

- Both solid core source and  $\cosh^2$  source cause a velocity field that resembles the velocity field of a point vortex, in regions not close to the source.

When writing this report, the programme used to determine the flow through the finite actuator strip was still in the phase of debugging. The following conclusions are drawn from the results of several testing programmes.

- The adaptation of the grid to the vorticity distribution is satisfactory, but the computing time needed for the grid generation is relatively long; for a 41x21 point grid the computation takes  $\approx 4s$ , the grid generation  $\approx 13s$ .

- The block structure of the programme facilitates changes in parts of the programme. For instance, it's relatively easy to build in an undisturbed velocity  $U_0$ .

LIST OF REFERENCES.

1. Patrick J. Roache, "Computational Fluid Dynamics"  
Hermosa Publishers, Albuquerque. 1976
2. J.A. Sparenberg, "Elements of Hydrodynamic Propulsion."  
Martinus Nijhoff Publishers. 1984
3. H.L. Stone, "Iterative Solutions of Implicit  
Approximations of Multidimensional Partial  
Differential Equations".  
SIAM J.Numer.Anal. 5 nr 3 sept'68 p530-
4. D.A.H. Jacobs, "The Strongly Implicit Procedure for  
Biharmonic problems."  
J.of Comp.Phys. 13, 303-315 (1973).
5. G.E. Schneider & M. Zedan, "A Modified Strongly Implicit Procedure for  
the Numerical Solution of Field Problems."  
Numerical Heat Transfer, vol.4, pp 1-19, 1981
6. G.A.M. van Kuik,  
PhD. Thesis, to be published in the course of 1988,  
Eindhoven University of Technology, Dept. of Physics.
7. Z.U.A. Warsi & J.F. Thompson, "The Importance of Dynamically Adaptive  
Grids in the Numerical Solution of Partial  
Differential Equations."  
von Karman Institute for Fluid Dynamics. Lecture  
series 1984-04. Computational Fluid Dynamics.  
March 12-16, 1984.
8. J.F. Thompson, "General Curvilinear Coordinate Systems."  
in: "Numerical Grid Generation.",  
editor Joe F. Thompson,  
Elsevier Science Publishing Company Inc. 1982.
9. Anderson, Tannehill & Pletcher, "Computational Fluid Dynamics and Heat  
Transfer."  
Hemisphere, 1984-XII
10. G.H. Schmidt & J.A. Sparenberg, "On the Edge Singularity of an  
Actuator Disk with Large Constant Normal  
Load."  
J.of Ship Research, Vol.21 nr.2 june 1977 pp125-131
11. M.D. Greenberg & J.H.W. Lee, "Line Momentum Source in Shallow,  
Inviscid Fluid."  
J.of Fluid Mechanics, Vol.145, 1984

12. G.A.M. van Kuik, "The Actuator Disc Edge Singularity: the Key to a Revised Actuator Disc Concept and Momentum Theory."  
paper no.2-16, 13th European Rotorcraft Forum, sept.8-11, 1987, Arles, France.
13. A.M. van der Spek, "Linear and Non-linear Theories of Flow Through an Actuator Disc."  
Master's Thesis, Eindhoven Technical University, Dept. of Physics, report no. R-803-A.
14. T.Y. Wu, "Flow Through a Heavily Loaded Actuator Disc."  
Schiffstechnik, Bd.9-1962, Heft 47.
15. H.S. Carslaw and J.C. Jaeger, "Conduction of Heats in Solids."  
Clarendon Press, Oxford, 1986.
16. M. Abramowitz and I.A. Stegun, "Handbook of Mathematical Functions."  
Dover Publications, I Inc., New York 1965.

Appendix A1

De sterk impliciete methode van Stone.[lit.3]

Dit is een methode voor het oplossen van grote stelsels vergelijkingen, die geschreven kunnen worden als:

$$S T = q \quad (1)$$

dit is een stelsel vergelijkingen, dat verkregen wordt door bijvoorbeeld een tweede orde partiele differentiaalvergelijking te diskretiseren in een rooster van twee dimensies.

Voor een rooster met  $M \times N$  punten is  $S$  een vierkante matrix met  $(M \times N)^2$  elementen.

$S$  is pentadiagonaal in het geval van een 5-punts diskretisatie, hier verder het geval.

$S$  ziet er als volgt uit:

Uitgeschreven voor een roosterpunt ziet de vergelijking er als volgt uit:

$$q_{j,k} = B_{j,k} T_{j,k-1} + D_{j,k} T_{j-1,k} + D_{j,k} T_{j,k} + F_{j,k} T_{j+1,k} + H_{j,k} T_{j,k+1}$$

$j$  is de index voor de  $x$ -koordinaat van een roosterpunt,  $k$  de index voor de  $y$ -koordinaat.

Een manier om het stelsel (1) op te lossen is om de matrix  $S$  te schrijven als het produkt van twee matrices  $L$  en  $U$ , met  $L$  een linksonderdriehoeksmatrix en  $U$  een rechtsbovendriehoeksmatrix.

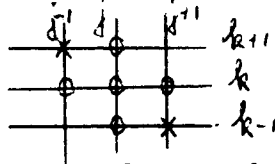
In het algemeen kost deze zogenaamde LU-dekompositie veel rekenwerk, omdat de matrix  $S$  meestal geen ijle matrices  $L$  en  $U$  oplevert. Onder een ijle matrix wordt verstaan een matrix, waarvan alle rijen op enkele na met nullen gevuld zijn.

Willen we een ijle L en U vinden, bijvoorbeeld L en U beiden tridiagonaal, dan moet bij S een matrix P opgeteld worden met P zodanig dat:

$$S + P = LU \tag{2}$$

Om te zien aan welke voorwaarden P moet voldoen, schrijven we LU uit:

S + P is een heptadiagonale matrix, de twee ten opzichte van S toegevoegde diagonalen kunnen gezien worden als coëfficiënten, horend bij de roosterpunten  $j+1, k-1$  en  $j-1, k+1$  (kruisjes in figuur a1).



Deze coëfficiënten noemen we  $C_{j,k}$  en  $G_{j,k}$  respectievelijk.

De funktiewaarden in de punten  $j+1, k-1$  en  $j-1, k+1$  kunnen geschreven worden als een lineaire combinatie van de funktiewaarden in de drie naastliggende punten. Dit kan door Taylorreeksontwikkeling met weglating van termen van de orde grootte  $x^3$ ,  $y^3$ , en  $xy$ . Dan geldt:

$$T_{j-1, k+1} = -T_{j, k} + T_{j, k+1} + T_{j-1, k} \tag{3}$$

en:

$$T_{j+1, k-1} = -T_{j, k} + T_{j+1, k} + T_{j, k-1} \tag{4}$$

Om de coëfficiënten  $C_{j,k}$  en  $G_{j,k}$  zo klein mogelijk te houden (S+P lijkt dan veel op S) kan het beste een ongeveer even grote waarde van  $T_{j-1, k+1}$  en  $T_{j+1, k-1}$  worden afgetrokken.

Het differentieschema voor S + P wordt dan:

$$\begin{aligned}
 & B_{j,k} T_{j,k-1} + D_{j,k} T_{j-1,k} + E_{j,k} T_{j,k} + F_{j,k} T_{j+1,k} + H_{j,k} T_{j,k+1} + \\
 & + C_{j,k} [ T_{j+1,k-1} - \alpha(-T_{j,k} + T_{j+1,k} + T_{j,k-1}) ] + \\
 & + G_{j,k} [ T_{j-1,k+1} - \alpha(-T_{j,k} + T_{j-1,k} + T_{j,k+1}) ] = q_{j,k} \quad (5)
 \end{aligned}$$

Gelijkstelling van de coëfficiënten van S+P en LU levert:

$$b_{j,k} = B_{j,k} - \alpha C_{j,k} \quad (6)$$

$$c_{j,k} = D_{j,k} - \alpha G_{j,k} \quad (7)$$

$$d_{j,k} + b_{j,k} f_{j,k-1} + c_{j,k} e_{j-1,k} = E_{j,k} + \alpha C_{j,k} + \alpha G_{j,k} \quad (8)$$

$$d_{j,k} e_{j,k} = F_{j,k} - \alpha C_{j,k} \quad (9)$$

$$d_{j,k} f_{j,k} = H_{j,k} - \alpha G_{j,k} \quad (10)$$

Uit deze vergelijkingen zijn iteratief de onbekenden  $b_{j,k}$ ,  $c_{j,k}$ ,  $d_{j,k}$ ,  $e_{j,k}$  en  $f_{j,k}$  op te lossen. Op de coëfficiënt  $\alpha$  komen we later terug.

Het bepalen van de coëfficiënten kan op twee manieren:

- 1) j oplopende van 1 tot M, k oplopend van 1 tot N
- 2) j weer oplopend, k afnemend van N tot 1, waarbij in de vergelijkingen (6) t/m (10) de j+1 wordt vervangen door j-1 en andersom.

De tweede manier komt neer op het doorlopen van het rooster van linksboven naar rechtsbeneden.

Er wordt steeds gebruik gemaakt van het feit dat de funktie buiten het rooster de waarde nul heeft:

$$c_{1,1} = 0 \quad \text{en} \quad C_{1,1} = 0 \quad (11)$$

$$b_{j,1} = 0 \quad \text{en} \quad B_{j,1} = 0 \quad (12)$$

$$f_{j,N} = 0 \quad \text{en} \quad H_{j,N} = 0 \quad (13)$$

$$F_{M,k} = 0 \quad (14)$$

Als de elementen van L en U bepaald zijn, kan het stelsel vergelijkingen (1) worden opgelost. In vergelijking (1) wordt aan beide zijden PT opgeteld:

$$(S + P)T = NT + q \quad (15)$$

Dit is ook gelijk aan:

$$(S + P)T = (S + P)T - (ST - q) \quad (16)$$

Deze vergelijkingen worden gebruikt voor het oplossen van het stelsel (1); stel  $T^{(n-1)}$  is de gevonden oplossing na de n-de iteratie; de nieuwe oplossing is dan gelijk aan:

$$(S + P)T^{(n)} = (S + P)T^{(n-1)} - (ST^{(n-1)} - q) \quad (17)$$

met  $\delta^{(n)} = T^{(n)} - T^{(n-1)}$  en  $R^{(n-1)} = q - ST^{(n-1)}$  wordt het op te lossen stelsel:

$$(S + P)\delta^{(n)} = R^{(n-1)} \quad (18)$$

of:

$$LU\delta^{(n)} = R^{(n-1)} \quad (19)$$

Om tot een snellere convergentie van de oplossing te komen kan de coëfficiënt  $\alpha$  uit de vergelijkingen (6) t/m (10) het beste worden gevarieerd. De maximale waarde van  $\alpha$  volgt volgens Jacobs [lit.4] uit de afmetingen van het rooster:

$$\alpha_{\max} = 1 - \frac{2}{(M-1)^2 + (N-1)^2} \quad (20)$$

Voor elke iteratie wordt een nieuwe  $\alpha$  genomen volgens

$$\alpha_i = 1 - (1 - \alpha_{\max})^{(i-1)/8} \quad i=1,9 \quad (21)$$

De volgorde waarin i wordt verwisseld is 9,6,3,8,5,2,7,4,1. Op deze manier omspant  $\alpha_i$  een groot bereik tussen 0 en  $\alpha_{\max}$  binnen een groepje van 3 opeenvolgende i.

WEG/WAM/STRIP/HO

DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 12:45:15.

```

10000 FILE 6(KIND='REMOTE')
10100 FILE 8(KIND='PRINTER')
10200 BLOCK GLOBALS
10300 C
10400 C This program calculates the vorticity distribution, the streamfunc-
10500 C tion and the velocity field of a half-infinite actuator strip with
10600 C its edge on the origin.
10700 C It uses the subroutines KOEFAP, KOEFRP and BEGWRD in the library LIB
10800 C and GRIDOP, VELOD2 and STIMSY in the library MAINLI.
10900 C The maximum grid size is 20000 gridpoints.
11000 C
11100 LIBRARY LIB(TITLE='OBJECT/WEG/WAM/STRIP/HO/STRIP/LIBRARY.')
11200 LIBRARY MAINLI(TITLE='OBJECT/WEG/WAM/STRIP/HO/MAIN/LIBRARY/
11300 * SOURCE.')
11400 END
11500 INTEGER I,J,K,L,M,N,NT,NTIT,NX,NY,NVF,TELLER,TELOME,TELPSI,LOOP,
11600 * IFALSE,XDELER,YDELER,TSOURC
11700 REAL A(20000),B(20000),C(20000),D(20000),
11800 * E(20000),F(20000),G(20000),
11900 * H(20000),KA(20000),P(20000),R(20000),
12000 * X(20000),Y(20000),U(20000),
12100 * V(20000),PSI(20000),OMO(20000),OMN(20000),
12200 * VHULP(20000),OMARKS(20000),
12300 * OM1(20000),Q(20000),FAC1,FAC2,CONV,RELRES,
12400 * H1(20000),H2(20000),H3(20000),H4(20000),
12500 * H5(20000),H6(20000),H7(20000),
12600 * H8(20000),FSTR,RHOO,NU,DT,EPS1,EPS2,OMEGA1,
12700 * OMEGA2,DEELT,NTQMAX,
12800 * NORM,NORMAK,ALFA,AKERN
12900 C
13000 C The computation is directed by parameters, that are stored in the
13100 C file 'STRIP/HO/PARAMETERS'.
13200 C
13300 OPEN(10,FILE='STRIP/HO/PARAMETERS',STATUS='OLD')
13400 OPEN(20,FILE='ROTATIEVELD',STATUS='NEW')
13500 OPEN(30,FILE='SNELHEIDSVELD',STATUS='NEW')
13600 OPEN(40,FILE='OMPSIUUV',STATUS='NEW')
13700 C
13800 C NT: number of time steps
13900 C TBEGIN: beginning time of the calculation
14000 C TEIND: ending time of calculation
14100 C M: number of grid points in X-direction
14200 C N: number of grid points in Y-direction
14300 C FSTR: value of the force on the actuator strip
14400 C NX: outer boundary of the grid in X-direction
14500 C NY: " " " " " " " Y- "

```

```

14600 C NVF: parameter to determine the strength of the non-regular
14700 C distribution of grid points, used in the subroutine GRIDOP
14800 C EPS1: accuracy for solving the vorticity diffusion equation
14900 C OMEGA1: iteration parameter
15000 C EPS2: same as EPS1, for solving the Poisson-PSI equation.
15100 C OMEGA2: iteration parameter
15200 C NTQMAX: time at which the force on the actuator strip is at
15300 C maximum strength. The strip is turned on at t=0.
15400 C TSOURC: type of source used to describe the behaviour of the
15500 C force density at the edge of the strip.
15600 C AKERN: distance to the core of the source where the force has
15700 C decreased to about 1/1000 th of its maximum.
15800 C
15900 RHOO=1.2
16000 NU=1.45E-5
16100 READ(10,*)NT,TBEGIN,TEIND,M,N,FSTR,NX,NY,NVF,EPS1,OMEGA1,EPS2,
16200 * OMEGA2,NTQMAX,TSOURC,AKERN
16300 WRITE(20,*)NT,TBEGIN,TEIND,M,N,FSTR,NX,NY,NVF,EPS1,OMEGA1,EPS2,
16400 * OMEGA2,NTQMAX,TSOURC,AKERN
16500 WRITE(30,*)NT,TBEGIN,TEIND,M,N,FSTR,NX,NY,NVF,EPS1,OMEGA1,EPS2,
16600 * OMEGA2,NTQMAX,TSOURC,AKERN
16700 WRITE(40,*)NT,TBEGIN,TEIND,M,N,FSTR,NX,NY,NVF,EPS1,OMEGA1,EPS2,
16800 * OMEGA2,NTQMAX,TSOURC,AKERN
16900 L=M*N
17000 C
17100 C The starting values for the various variables are determined in
17200 C BEGWRD.
17300 C
17400 DT=TEIND/NT
17500 CALL BEGWRD(L,M,N,U,V,OMO,OMN,PSI,DT,0.)
17600 C
17700 C The coordinates of the grid points are calculated in GRIDOP.
17800 C
17900 CALL GRIDOP(X,Y,M,N,NVF,NX,NY)
18000 C
18100 C beginning of the calculation; the loop is repeated for every time
18200 C step.
18300 C
18400 DO 200 NTIT=1,NT
18500 C
18600 C DEELT is smaller than 1 if the force hasn't reached its maximum
18700 C strenght.
18800 C
18900 IF(NTIT*DT.GT.NTQMAX)DEELT=1.
19000 IF(NTIT*DT.LE.NTQMAX)DEELT=NTIT*DT/NTQMAX
19100 C
19200 C In KOEFAP the coefficients of the discretised vorticity diffusion
19300 C equation are determined.
19400 C
19500 CALL KOEFAP(A,B,C,D,E,F,OMO,L,M,N,DT,U,V,X,Y,RHOO,NU,FSTR,
19600 * DEELT,AKERN,TSOURC)

```



```

19700 C
19800 C The vorticity diffusion equation is solved by the strongly implicit,
19900 C symmetric procedure.
20000 C
20100 C IFALSE=0
20200 CALL STIMSY(A,B,C,D,E,OMN,F,L,M,N,EPS1,TELOME,H1,H2,H3,H4,H5,H6,
20300 * H7,H8,VHULP)
20400 C
20500 C In KOEFRP the coefficients of the discretised Poisson-PSI equation
20600 C are determined.
20700 C
20800 CALL KOEFRP(X,Y,L,M,N,R,G,H,KA,P,Q,OMN)
20900 C
21000 C Again STIMSY is used, now to solve the Poisson-PSI equation.
21100 C
21200 CALL STIMSY(R,G,H,KA,P,PSI,Q,L,M,N,EPS2,TELPSI,H1,H2,H3,
21300 * H4,H5,H6,H7,H8,VHULP)
21400 C
21500 C In VFLOD2 the velocity components are calculated from the stream-
21600 C function.
21700 C
21800 CALL VFLOD2(U,V,X,Y,PSI,L,M,N)
21900 C
22000 C For the calculation in the next time step, the values of OMN (omega
22100 C new) are copied to OMO (omega old).
22200 C
22300 DO 160 I=1,L
22400 OMO(I)=OMN(I)
22500 160 CONTINUE
22600 C
22700 C If TEIND isn't reached yet, the whole calculation is repeated for
22800 C the next time step.
22900 C
23000 200 CONTINUE
23100 WRITE(6,115)TELOME
23200 115 FORMAT(IX,"Werveldiffusievergelijking opgelost.",/ ,IX,"Aantal",
23300 * "verrichte iteraties: ",I4,/ )
23400 C
23500 C The results of the calculation are stored in FILE40 (OMPSIU.V.).
23600 C By using other programmes it's possible to make a plot of the
23700 C streamlines, of the grid or of the vorticity distribution, or to
23800 C get a hard copy of the results.
23900 C
24000 DO 130 I=1,L
24100 130 WRITE(40,*)X(I),Y(I),OMN(I),PSI(I),U(I),V(I)
24200 C
24300 C For use in e.g. convergence calculations, the number of grid points
24400 C is reduced to 9x9 and the values of the vorticity and the velocities
24500 C are stored in FILE20 and FILE30, respectively.
24600 C

```

```

24700 XDELER=(M-1)/8
24800 YDELER=(N-1)/8
24900 DO 120 K=1,9
25000 DO 120 I=1,9
25100 OMN(I+(K-1)*9)=OMN((I-1)*XDELER+I+((K-1)*YDELER)*M)
25200 U(I+(K-1)*9)=U((I-1)*XDELER+I+((K-1)*YDELER)*M)
25300 V(I+(K-1)*9)=V((I-1)*XDELER+I+((K-1)*YDELER)*M)
25400 120 CONTINUE
25500 C
25600 C wegschrijven van de snelheidswaarden voor de laatste tijdstap:
25700 C
25800 DO 150 K=9-1,0,-1
25900 WRITE(20,*)(OMN(K*9+I),I=1,9)
26000 WRITE(30,*)(U(K*9+I),I=1,9)
26100 WRITE(30,*)(V(K*9+I),I=1,9)
26200 150 CONTINUE
26300 WRITE(20,*)TELOME
26400 WRITE(30,*)TELPSI
26500 WRITE(40,*)TELOME,TELPSI
26600 C
26700 C
26800 C afsluiten files
26900 C
27000 CLOSE(20,STATUS="CRUNCH")
27100 CLOSE(10,STATUS="CRUNCH")
27200 CLOSE(30,STATUS="CRUNCH")
27300 CLOSE(40,STATUS="CRUNCH")
27400 WRITE(6,300)TELPSI
27500 300 FORMAT(IX,"Aantal verrichte iteraties bij het oplossen van de",
27600 * "Poisson-PSI vgl: ",I4,/ )
27700 END
27800 SUBROUTINE GRIDOP(X,Y,M,N,NVF,NX,NY)
27900 INTEGER M,N,NVF,NX,NY
28000 REAL X(M*N),Y(M*N)
28100 IN LIBRARY MAINLI
28200 END
28300 SUBROUTINE SSOR(A,B,C,D,E,F,V,VH,L,M,EPS,OMEGA,TELLER,IFALSE)
28400 INTEGER L,M,TELLER,IFALSE
28500 REAL A(L),B(L),C(L),D(L),E(L),F(L),V(L),VH(L),EPS,OMEGA
28600 IN LIBRARY MAINLI
28700 END
28800 SUBROUTINE KOEFAP(A,B,C,D,E,F,OMO,L,M,N,DT,U,V,X,Y,RHO,NU,FSTR,
28900 * DEELT,AKERN,TSOURC)
29000 INTEGER L,M,N,TSOURC
29100 REAL A(L),B(L),C(L),D(L),E(L),F(L),OMO(L),U(L),V(L),X(M*N),
29200 * Y(M*N),DT,RHO,NU,FSTR,DEELT,AKERN
29300 IN LIBRARY LIB
29400 END
29500 SUBROUTINE KOEFRP(X,Y,L,M,N,R,G,H,KA,P,Q,OMO)
29600 INTEGER L,M,N

```

69

```

29700 REAL X(M*N),Y(M*N),R(L),G(L),H(L),KA(L),P(L),Q(L),OMO(L)
29800 IN LIBRARY LIB
29900 END
30000 SUBROUTINE VELOD2(U,V,X,Y,PSI,L,M,N)
30100 INTEGER L,M,N
30200 REAL U(L),V(L),X(L),Y(L),PSI(L)
30300 IN LIBRARY MAINLI
30400 END
30500 SUBROUTINE BEGWRD(L,M,N,U,V,OMO,OMN,PSI,DT,FSTR)
30600 INTEGER L,M,N
30700 REAL U(M),V(N),OMO(L),OMN(L),PSI(L),DT,FSTR
30800 IN LIBRARY LIB
30900 END
31000 SUBROUTINE STIMP1(R,G,H,KA,P,PSI,OM,L,M,N,ALFA,EPS,TELPSI,
31100 * A,B,C,D,E,F,VH,RH,DE)
31200 INTEGER L,M,N,TELPSI
31300 REAL R(L),G(L),H(L),KA(L),P(L),PSI(L),OM(L),A(L),B(L),C(L),
31400 * D(L),E(L),F(L),VH(L),RH(L),DE(L),ALFA,EPS
31500 IN LIBRARY MAINLI
31600 END
31700 SUBROUTINE STIMSY(R,G,H,KA,P,PSI,OM,L,M,N,EPS,TELPSI,
31800 * A,B,C,D,E,F,VH,RH,DE)
31900 INTEGER L,M,N,TELPSI
32000 REAL R(L),G(L),H(L),KA(L),P(L),PSI(L),OM(L),A(L),B(L),C(L),
32100 * D(L),E(L),F(L),VH(L),RH(L),DE(L),EPS
32200 IN LIBRARY MAINLI
32300 END

```

WEG/WAM/STRIP/HO/MAIN/LIBRARY/SOURCE  
DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 13:13:08.

```

10000 FILE 6(KIND='REMOTE')
10100 BLOCK GLOBALS
10200 EXPORT(GRIDOP,SSOR,VELOD3,VELOD2,STIMP1,STIMSY)
10300 END
10400 CALL FREEZE('TEMPORARY')
10500 END
10600 SUBROUTINE GRIDOP(X,Y,M,N,NVF,NX,NY)
10700 INTEGER M,N,NVF,NX,NY,I,J
10800 REAL X(M*N),Y(M*N)
10900 C
11000 C This subroutine generates a non linear distribution of grid points.
11100 C The distance between grid points decreases if the grid points lie
11200 C nearer to the coordinate axes. The rate of decrease is determined
11300 C lie in the centre of the grid.
11400 C
11500 DO 10 J=1,N
11600 DO 10 I=1,M
11700 K=(J-1)*M+I
11800 X(K)=NX*((2.*(I-1)/(M-1)-1.))**(2.*NVF+1))
11900 10 Y(K)=NY*((2.*(J-1)/(N-1)-1.))**(2.*NVF+1))
12000 RETURN
12100 END

```

70

```

27400 SUBROUTINE VELOD2(U,V,X,Y,PSI,L,M,N)
27500 INTEGER J,I,K,L,M,N
27600 REAL KI,KJ,S1,S2,U(L),V(L),X(L),Y(L),PSI(L)
27700 C
27800 C Determination of the velocity components in X- and in Y-direction
27900 C from the streamfunction.
28000 C Discretisation schemes with truncation errors of order 2 are used.
28100 C
28200 DO 100 J=2,N-1
28300 DO 100 I=2,M-1
28400 K=I+(J-1)*M
28500 KI=X(K+1)-X(K-1)
28600 V(K)=(PSI(K-1)-PSI(K+1))/KI
28700 KJ=Y(K+M)-Y(K-M)
28800 U(K)=(PSI(K+M)-PSI(K-M))/KJ
28900 100 CONTINUE
29000 DO 200 J=1,N
29100 DO 200 I=1,M
29200 K=I+(J-1)*M
29300 IF (I.EQ.1)THEN
29400 KI=X(K+1)-X(K)
29500 V(K)=(PSI(K)-PSI(K+1))/KI
29600 END IF
29700 IF (I.EQ.M)THEN
29800 KI=X(K)-X(K-1)
29900 V(K)=(PSI(K-1)-PSI(K))/KI
30000 END IF
30100 IF (J.EQ.1)THEN
30200 KJ=Y(K+M)-Y(K)
30300 U(K)=(PSI(K+M)-PSI(K))/KJ
30400 END IF
30500 IF (J.EQ.N)THEN
30600 KJ=Y(K)-Y(K-M)
30700 U(K)=(PSI(K)-PSI(K-M))/KJ
30800 END IF
30900 200 CONTINUE
31000 RETURN
31100 END
31200 SUBROUTINE STIMSY(R,G,H,KA,P,PSI,OM,L,M,N,EPS,TELPSI,A,B,C,
31300 * D,E,F,VH,RH,DE)
31400 INTEGER L,M,N,J,TELPSI,NT,NITV(9),NIT
31500 REAL R(L),G(L),H(L),KA(L),P(L),PSI(L),OM(L),A(L),B(L),C(L),D(L),
31600 * E(L),F(L),VH(L),RH(L),DE(L),ALFA,EPS,REST,Q,S,ITPAR(9),ALMAX
31700 C
31800 C This subroutine applies the strongly implicit, symmetric method of
31900 C Stone to solve a discretised differential equation in two
32000 C dimensions.
32100 C The equations, to be solved, read:
32200 C
32300 R1 PSI I-m + G1 PSI I-1 + H1 PSI I + KA1 PSI I+1 + P1 PSI I+m = OMI

```

```

39800 C
39900 C This is written in matrix form as:
40000 C
40100 C MA PSI = OM
40200 C
40300 C PSI is the value to be solved.
40400 C To MA a sparse matrix NA is added, in such a way that: MA + NA = L
40500 C L and U both are triangular.
40600 C
40700 C The equation then can be written as:
40800 C
40900 C (MA + NA)PSI' = (MA + NA)PSI + OM - NA PSI
41000 C
41100 C or:
41200 C
41300 C L U (PSI' - PSI) = RH
41400 C I U DE = RH
41500 C L VH = RH
41600 C L VH = RH
41700 C and:
41800 C U DE = VH
41900 C
42000 C RH is a residual vector
42100 C VH is a help vector
42200 C
42300 C First a set of iteration parameters ALFA is made, varying between
42400 C zero and ALMAX.
42500 C
42600 C DO 10 I=3,1,-1
42700 C DO 10 J=0,2
42800 10 NITV(10-3*I+J)=6+I-3*J
42900 ALMAX=1.-2./((M-1)**2+(N-1)**2)
43000 DO 20 I=1,9
43100 20 ITPAR(I)=1.-((1.-ALMAX)**((I-1.)/8.))
43200 NT=0
43300 NIT=0
43400 50 NIT=NIT+1
43500 NT=NT+1
43600 IF(NIT.EQ.10)NIT=1
43700 ALFA=ITPAR(NITV(NIT))
43800 C
43900 C calculating the residual vector RH(I):
44000 C
44100 C DO 100 J=1,L
44200 C RH(J)=OM(J)-H(J)*PSI(J)
44300 C IF(J.GE.2)RH(J)=RH(J)-G(J)*PSI(J-1)
44400 C IF(J.GE.M+1)RH(J)=RH(J)-R(J)*PSI(J-M)
44500 C IF(J.LE.L-M)RH(J)=RH(J)-P(J)*PSI(J+M)
44600 C IF(J.LE.L-1)RH(J)=RH(J)-KA(J)*PSI(J+1)
44700 100 CONTINUE

```

```

44800 C
44900 C Determination of the coefficients of the L and U matrices, for both
45000 C increasing row number and column number.
45100 C
45200 C DO 200 J=1,L
45300 C R(J)=0.
45400 C IF(J.GT.M)R(J)=R(J)/(1.+ALFA*E(J-M))
45500 C Q=0.
45600 C IF(J.GT.M)Q=R(J)*E(J-M)
45700 C C(J)=0.
45800 C IF(J.GT.1.AND.J.LE.L-M+1)C(J)=C(J)/(1.+ALFA*F(J-1))
45900 C IF(J.GT.L-M+1)C(J)=G(J)
46000 C S=0.
46100 C IF(J.GT.1.AND.J.LE.L-M+1)S=C(J)*F(J-1)
46200 C D(J)=H(J)+ALFA*(Q+S)
46300 C IF(J.GE.M+1)D(J)=D(J)-B(J)*F(J-M)
46400 C IF(J.GE.2)D(J)=D(J)-C(J)*E(J-1)
46500 C IF(J.EQ.L)E(J)=0.
46600 C IF(J.LT.L)E(J)=(-ALFA*Q+KA(J))/D(J)
46700 C F(J)=-ALFA*S/D(J)
46800 C IF(J.LE.L-M)F(J)=F(J)+P(J)/D(J)
46900 200 CONTINUE
47000 C
47100 C
47200 C Solution with lower triangular matrix.
47300 C
47400 C DO 300 J=1,L
47500 C IF(J.GE.2)RH(J)=RH(J)-C(J)*VH(J-1)
47600 C IF(J.GE.M+1)RH(J)=RH(J)-B(J)*VH(J-M)
47700 300 VH(J)=RH(J)/D(J)
47800 C
47900 C Solution with upper triangular matrix; DE(I) is the difference
48000 C new and old solution.
48100 C
48200 C DO 400 J=L,1,-1
48300 C IF(J.LT.L)VH(J)=VH(J)-E(J)*DE(J+1)
48400 C IF(J.LE.L-M)VH(J)=VH(J)-F(J)*DE(J+M)
48500 400 DE(J)=VH(J)
48600 C
48700 C Determination of new solution PSI, plus the maximum difference
48800 C new and old solution.
48900 C
49000 C REST=0.
49100 C DO 500 J=1,L
49200 C PSI(J)=PSI(J)+DE(J)
49300 C IF(ABS(DE(J)).GT.REST)REST=ABS(DE(J))
49400 500 CONTINUE
49500 C IF(REST.LT.EPS)GO TO 1000
49600 C NT=NT+1

```

```

49700 C
49800 C calculation of the residual vector RH(I)
49900 C
50000 DO 450 J=1,L
50100 RH(J)=OM(J)-H(J)*PSI(J)
50200 IF(J.GE.2)RH(J)=RH(J)-G(J)*PSI(J-1)
50300 IF(J.GE.M+1)RH(J)=RH(J)-R(J)*PSI(J-M)
50400 IF(J.LF.L-M)RH(J)=RH(J)-P(J)*PSI(J+M)
50500 IF(J.LF.L-1)RH(J)=RH(J)-KA(J)*PSI(J+1)
50600 450 CONTINUE
50700 C
50800 C determination of the coefficients of the L and U matrices, for
50900 C decreasing row number and increasing column number.
51000 C
51100 600 DO 700 K=N,1,-1
51200 DO 700 I=1,M
51300 J=I+(K-1)*M
51400 IF(K.LT.N.AND.K.GT.1)R(J)=R(J)/(1.+ALFA*E(J+M))
51500 IF(K.EQ.1)B(J)=0.
51600 IF(K.EQ.N)B(J)=R(J)
51700 IF(J.EQ.1)C(J)=0.
51800 IF(I.EQ.1.AND.J.GT.1)C(J)=G(J)
51900 IF(I.GT.1)C(J)=G(J)/(1.+ALFA*F(J-1))
52000 D(J)=H(J)
52100 IF(J.GT.1)D(J)=D(J)-C(J)+G(J)
52200 IF(I.GT.1)D(J)=D(J)-C(J)*E(J-1)
52300 IF(K.GT.1)D(J)=D(J)+R(J)-B(J)
52400 IF(K.GT.1.AND.K.LT.N)D(J)=D(J)-B(J)*F(J+M)
52500 E(J)=0.
52600 IF(J.LT.L)E(J)=KA(J)
52700 IF(K.GT.1.AND.J.LT.L)E(J)=E(J)+B(J)-R(J)
52800 E(J)=E(J)/D(J)
52900 F(J)=0.
53000 IF(J.GT.1.AND.K.LT.N)F(J)=C(J)-G(J)
53100 IF(K.LT.N)F(J)=F(J)+P(J)
53200 F(J)=F(J)/D(J)
53300 700 CONTINUE
53400 C
53500 C Solution with lower triangular matrix.
53600 C
53700 C
53800 DO 850 J=1,L
53900 IF(J.GE.2)RH(J)=RH(J)-C(J)*VH(J-1)
54000 IF(J.GE.M+1)RH(J)=RH(J)-B(J)*VH(J-M)
54100 850 VH(J)=RH(J)/D(J)
54200 C
54300 C Solution with upper triangular matrix; DE is the difference between
54400 C new and old solution.
54500 C

```

```

54600 DO 900 J=L,1,-1
54700 IF(J.LT.L)VH(J)=VH(J)-E(J)*DE(J+1)
54800 IF(J.LF.L-M)VH(J)=VH(J)-F(J)*DE(J+M)
54900 900 DE(J)=VH(J)
55000 C
55100 C Determination of the new solution PSI and the maximum difference
55200 C between new and old solution.
55300 C
55400 REST=0.
55500 DO 950 J=1,L
55600 PSI(J)=PSI(J)+DE(J)
55700 IF(ABS(DE(J)).GT.REST)REST=ABS(DE(J))
55800 950 CONTINUE
55900 IF(REST.LT.EPS)GO TO 1000
56000 GO TO 50
56100 1000 CONTINUE
56200 TELPS I=TELPS I+NT
56300 RETURN
56400 END
WEG/WAM/STRIP/HO/STRIP/LIBRARY
DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 13:23:50.
10000 FILE 6(KIND='REMOTE')
10100 FILE 8(KIND='PRINTER')
10200 BLOCK GLOBALS
10300 EXPORT(KOEFAP,KOEFAP,BEGWRD)
10400 END
10500 CALL FREEZE('TEMPORARY')
10600 END
10700 C
10800 C
10900 C
11000 SUBROUTINE KOEFAP(A,B,C,D,E,F,OMO,L,M,N,DT,U,V,X,Y,RHOO,NU,FSTR,
11100 * DEELT,AKERN,TSOURC)
11200 INTEGER L,M,N,I,J,K,TSOURC
11300 REAL DEELT,UR,UL,VR,VL,KX,KY,KI,KJ,S1,S2,AKERN,
11400 * A(L),B(L),C(L),D(L),E(L),F(L),OMO(L),U(L),V(L),X(L),
11500 * Y(L),DT,RHOO,NU,FSTR
11600 C
11700 C
11800 C Calculation of the value of the coefficients in the discretised
11900 C vorticity diffusion equation.
12000 C Boundary condition: the vorticity at the boundaries is zero.
12100 C The convective term is discretised according to Roache's second
12200 C upwind differencing scheme.
12300 C
12400 C
12500 DO 100 J=2,N-1
12600 DO 100 I=2,M-1

```

```

12700      K=I+(J-1)*M
12800      UR=(U(K+1)+U(K))/2.
12900      UL=(U(K)+U(K-1))/2.
13000      VR=(V(K+M)+V(K))/2.
13100      VL=(V(K)+V(K-M))/2.
13200      KX=(X(K+1)-X(K-1))/2.
13300      IF(UR.GE.0..AND.UL.GE.0.)KX=X(K)-X(K-1)
13400      IF(UR.LT.0..AND.UL.LT.0.)KX=X(K+1)-X(K)
13500      KY=(Y(K+M)-Y(K-M))/2.
13600      IF(VR.GE.0..AND.VL.GE.0.)KY=Y(K)-Y(K-M)
13700      IF(VR.LT.0..AND.VL.LT.0.)KY=Y(K+M)-Y(K)
13800      KI=X(K)-X(K-1)
13900      KJ=Y(K)-Y(K-M)
14000      S1=(X(K+1)-X(K))/KI
14100      S2=(Y(K+M)-Y(K))/KJ
14200      IF(J.EQ.2)THEN
14300         A(K)=0.
14400      ELSE
14500         A(K)=(2*NU*DT)/((1+S2)*(KJ**2))
14600         IF(VL.GE.0.)A(K)=A(K)+(VL*DT)/KY
14700      END IF
14800      IF(I.EQ.2)THEN
14900         B(K)=0.
15000      ELSE
15100         B(K)=(2*NU*DT)/((1+S1)*(KI**2))
15200         IF(UL.GE.0.)B(K)=B(K)+(UL*DT)/KX
15300      END IF
15400      C(K)=-1.-(2*NU*DT)*(1./(S1*(KI**2))+1./(S2*(KJ**2)))
15500      IF(UR.GE.0.)C(K)=C(K)-(UR*DT)/KX
15600      IF(UL.LT.0.)C(K)=C(K)+(UL*DT)/KX
15700      IF(VR.GE.0.)C(K)=C(K)-(VR*DT)/KY
15800      IF(VL.LT.0.)C(K)=C(K)+(VL*DT)/KY
15900      IF(I.EQ.M-1)THEN
16000         D(K)=0.
16100      ELSE
16200         D(K)=(2*NU*DT)/(S1*(1+S1)*(KI**2))
16300         IF(UR.LT.0.)D(K)=D(K)-(UR*DT)/KX
16400      END IF
16500      IF(J.EQ.N-1)THEN
16600         E(K)=0.
16700      ELSE
16800         E(K)=(2*NU*DT)/(S2*(1+S2)*(KJ**2))
16900         IF(VR.LT.0.)E(K)=E(K)-(VR*DT)/KY
17000      END IF
C
C QP and QG give the strength of the source at the edge of the
C actuator strip as a function of time and place.
C
17100      *
17200      IF(TSOURC.EQ.1)F(K)=-OMO(K)-DT*QP(RH00,FSTR,X(K),Y(K),DEELT,
17300      AKERN)
17400
17500
17600
17700
17800
17900
18000
18100
18200
18300
18400
18500
18600
18700
18800
18900
19000
19100
19200
19300
19400
19500
19600
19700
19800
19900
20000
20100
20200
20300
20400
20500
20600
20700
20800
20900
21000
21100
21200
21300
21400
21500
21600
21700
21800
21900
22000
22100
22200
22300
22400
22500
22600
22700
22800
22900
23000
23100
23200
23300
23400
23500
23600
23700
23800
23900
24000
24100
24200
24300
24400
24500
24600
24700
24800
24900
25000
25100
25200
25300
25400
25500
25600
25700
25800
25900
26000
26100
26200
26300
26400
26500
26600
26700
26800
26900
27000
27100
27200
27300
27400
27500
27600
27700
27800
27900
28000
28100
28200
28300
28400
28500
28600
28700
28800
28900
29000
29100
29200
29300
29400
29500
29600
29700
29800
29900
30000
30100
30200
30300
30400
30500
30600
30700
30800
30900
31000
31100
31200
31300
31400
31500
31600
31700
31800
31900
32000
32100
32200
32300
32400
32500
32600
32700
32800
32900
33000
33100
33200
33300
33400
33500
33600
33700
33800
33900
34000
34100
34200
34300
34400
34500
34600
34700
34800
34900
35000
35100
35200
35300
35400
35500
35600
35700
35800
35900
36000
36100
36200
36300
36400
36500
36600
36700
36800
36900
37000
37100
37200
37300
37400
37500
37600
37700
37800
37900
38000
38100
38200
38300
38400
38500
38600
38700
38800
38900
39000
39100
39200
39300
39400
39500
39600
39700
39800
39900
40000
40100
40200
40300
40400
40500
40600
40700
40800
40900
41000
41100
41200
41300
41400
41500
41600
41700
41800
41900
42000
42100
42200
42300
42400
42500
42600
42700
42800
42900
43000
43100
43200
43300
43400
43500
43600
43700
43800
43900
44000
44100
44200
44300
44400
44500
44600
44700
44800
44900
45000
45100
45200
45300
45400
45500
45600
45700
45800
45900
46000
46100
46200
46300
46400
46500
46600
46700
46800
46900
47000
47100
47200
47300
47400
47500
47600
47700
47800
47900
48000
48100
48200
48300
48400
48500
48600
48700
48800
48900
49000
49100
49200
49300
49400
49500
49600
49700
49800
49900
50000
50100
50200
50300
50400
50500
50600
50700
50800
50900
51000
51100
51200
51300
51400
51500
51600
51700
51800
51900
52000
52100
52200
52300
52400
52500
52600
52700
52800
52900
53000
53100
53200
53300
53400
53500
53600
53700
53800
53900
54000
54100
54200
54300
54400
54500
54600
54700
54800
54900
55000
55100
55200
55300
55400
55500
55600
55700
55800
55900
56000
56100
56200
56300
56400
56500
56600
56700
56800
56900
57000
57100
57200
57300
57400
57500
57600
57700
57800
57900
58000
58100
58200
58300
58400
58500
58600
58700
58800
58900
59000
59100
59200
59300
59400
59500
59600
59700
59800
59900
60000
60100
60200
60300
60400
60500
60600
60700
60800
60900
61000
61100
61200
61300
61400
61500
61600
61700
61800
61900
62000
62100
62200
62300
62400
62500
62600
62700
62800
62900
63000
63100
63200
63300
63400
63500
63600
63700
63800
63900
64000
64100
64200
64300
64400
64500
64600
64700
64800
64900
65000
65100
65200
65300
65400
65500
65600
65700
65800
65900
66000
66100
66200
66300
66400
66500
66600
66700
66800
66900
67000
67100
67200
67300
67400
67500
67600
67700
67800
67900
68000
68100
68200
68300
68400
68500
68600
68700
68800
68900
69000
69100
69200
69300
69400
69500
69600
69700
69800
69900
70000
70100
70200
70300
70400
70500
70600
70700
70800
70900
71000
71100
71200
71300
71400
71500
71600
71700
71800
71900
72000
72100
72200
72300
72400
72500
72600
72700
72800
72900
73000
73100
73200
73300
73400
73500
73600
73700
73800
73900
74000
74100
74200
74300
74400
74500
74600
74700
74800
74900
75000
75100
75200
75300
75400
75500
75600
75700
75800
75900
76000
76100
76200
76300
76400
76500
76600
76700
76800
76900
77000
77100
77200
77300
77400
77500
77600
77700
77800
77900
78000
78100
78200
78300
78400
78500
78600
78700
78800
78900
79000
79100
79200
79300
79400
79500
79600
79700
79800
79900
80000
80100
80200
80300
80400
80500
80600
80700
80800
80900
81000
81100
81200
81300
81400
81500
81600
81700
81800
81900
82000
82100
82200
82300
82400
82500
82600
82700
82800
82900
83000
83100
83200
83300
83400
83500
83600
83700
83800
83900
84000
84100
84200
84300
84400
84500
84600
84700
84800
84900
85000
85100
85200
85300
85400
85500
85600
85700
85800
85900
86000
86100
86200
86300
86400
86500
86600
86700
86800
86900
87000
87100
87200
87300
87400
87500
87600
87700
87800
87900
88000
88100
88200
88300
88400
88500
88600
88700
88800
88900
89000
89100
89200
89300
89400
89500
89600
89700
89800
89900
90000
90100
90200
90300
90400
90500
90600
90700
90800
90900
91000
91100
91200
91300
91400
91500
91600
91700
91800
91900
92000
92100
92200
92300
92400
92500
92600
92700
92800
92900
93000
93100
93200
93300
93400
93500
93600
93700
93800
93900
94000
94100
94200
94300
94400
94500
94600
94700
94800
94900
95000
95100
95200
95300
95400
95500
95600
95700
95800
95900
96000
96100
96200
96300
96400
96500
96600
96700
96800
96900
97000
97100
97200
97300
97400
97500
97600
97700
97800
97900
98000
98100
98200
98300
98400
98500
98600
98700
98800
98900
99000
99100
99200
99300
99400
99500
99600
99700
99800
99900
100000
100100
100200
100300
100400
100500
100600
100700
100800
100900
101000
101100
101200
101300
101400
101500
101600
101700
101800
101900
102000
102100
102200
102300
102400
102500
102600
102700
102800
102900
103000
103100
103200
103300
103400
103500
103600
103700
103800
103900
104000
104100
104200
104300
104400
104500
104600
104700
104800
104900
105000
105100
105200
105300
105400
105500
105600
105700
105800
105900
106000
106100
106200
106300
106400
106500
106600
106700
106800
106900
107000
107100
107200
107300
107400
107500
107600
107700
107800
107900
108000
108100
108200
108300
108400
108500
108600
108700
108800
108900
109000
109100
109200
109300
109400
109500
109600
109700
109800
109900
110000
110100
110200
110300
110400
110500
110600
110700
110800
110900
111000
111100
111200
111300
111400
111500
111600
111700
111800
111900
112000
112100
112200
112300
112400
112500
112600
112700
112800
112900
113000
113100
113200
113300
113400
113500
113600
113700
113800
113900
114000
114100
114200
114300
114400
114500
114600
114700
114800
114900
115000
115100
115200
115300
115400
115500
115600
115700
115800
115900
116000
116100
116200
116300
116400
116500
116600
116700
116800
116900
117000
117100
117200
117300
117400
117500
117600
117700
117800
117900
118000
118100
118200
118300
118400
118500
118600
118700
118800
118900
119000
119100
119200
119300
119400
119500
119600
119700
119800
119900
120000
120100
120200
120300
120400
120500
120600
120700
120800
120900
121000
121100
121200
121300
121400
121500
121600
121700
121800
121900
122000
122100
122200
122300
122400
122500
122600
122700
122800
122900
123000
123100
123200
123300
123400
123500
123600
123700
123800
123900
124000
124100
124200
124300
124400
124500
124600
124700
124800
124900
125000
125100
125200
125300
125400
125500
125600
125700
125800
125900
126000
126100
126200
126300
126400
126500
126600
126700
126800
126900
127000
127100
127200
127300
127400
127500
127600
127700
127800
127900
128000
128100
128200
128300
128400
128500
128600
128700
128800
128900
129000
129100
129200
129300
129400
129500
129600
129700
129800
129900
130000
130100
130200
130300
130400
130500
130600
130700
130800
130900
131000
131100
131200
131300
131400
131500
131600
131700
131800
131900
132000
132100
132200
132300
132400
132500
132600
132700
132800
132900
133000
133100
133200
133300
133400
133500
133600
133700
133800
133900
134000
134100
134200
134300
134400
134500
134600
134700
134800
134900
135000
135100
135200
135300
135400
135500
135600
135700
135800
135900
136000
136100
136200
136300
136400
136500
136600
136700
136800
136900
137000
137100
137200
137300
137400
137500
137600
137700
137800
137900
138000
138100
138200
138300
138400
138500
138600
138700
138800
138900
139000
139100
139200
139300
139400
139500
139600
139700
139800
139900
140000
140100
140200
140300
140400
140500
140600
140700
140800
140900
141000
141100
141200
141300
141400
141500
141600
141700
141800
141900
142000
142100
142200
142300
142400
142500
142600
142700
142800
142900
143000
143100
143200
143300
143400
143500
143600
143700
143800
143900
144000
144100
144200
144300
144400
144500
144600
144700
144800
144900
145000
145100
145200
145300
145400
145500
145600
145700
145800
145900
146000
146100
146200
146300
146400
146500
146600
146700
146800
146900
147000
147100
147200
147300
147400
147500
147600
147700
147800
147900
148000
148100
148200
148300
148400
148500
148600
148700
148800
148900
149000
149100
149200
149300
149400
149500
149600
149700
149800
149900
150000
150100
150200
150300
150400
150500
150600
150700
150800
150900
151000
151100
151200
151300
151400
151500
151600
151700
151800
151900
152000
152100
152200
152300
152400
152500
152600
152700
152800
152900
153000
153100
153200
153300
153400
153500
153600
153700
153800
153900
154000
154100
154200
154300
154400
154500
154600
154700
154800
154900
155000
155100
155200
155300
155400
155500
155600
155700
155800
155900
156000
156100
156200
156300
156400
156500
156600
156700
156800
156900
157000
157100
157200
157300
157400
157500
157600
157700
157800
157900
158000
158100
158200
158300
158400
158500
158600
158700
158800
158900
159000
159100
159200
159300
159400
159500
159600
159700
159800
159900
160000
160100
160200
160300
160400
160500
160600
160700
160800
160900
161000
161100
161200
161300
161400
161500
161600
161700
161800
161900
162000
162100
162200
162300
162400
162500
162600
162700
162800
162900
163000
163100
163200
163300
163400
163500
163600
163700
163800
163900
164000
164100
164200
164300
164400
164500
164600
164700
164800
164900
165000
165100
165200
165300
165400
165500
165600
165700
165800
165900
166000
166100
166200
166300
166400
166500
166600
166700
166800
166900
167000
167100
167200
167300
167400
167500
167600
167700
167800
167900
168000
168100
168200
168300
168400
168500
168600
168700
168800
168900
169000
169100
169200
169300
169400
169500
169600
169700
169800
169900
170000
170100
170200
170300
170400
170500
170600
170700
170800
170900
171000
171100
171200
171300
171400
171500
171600
171700
171800
171900
172000
172100
172200
172300
172400
172500
172600
172700
172800
172900
173000
173100
173200
173300
173400
173500
173600
173700
173800
173900
174000
174100
174200
174300
174400
174500
174600
174700
174800
174900
175000
175100
175200
175300
175400
175500
175600
175700
175800
175900
176000
176100
176200
176300
176400
176500
176600
176700
176800
176900
177000
177100
177200
177300
177400
177500
177600
177700
177800
177900
178000
178100
178200
178300
178400
178500
178600
178700
178800
178900
179000
179100
179200
179300
179400
179500
179600
179700
179800
179900
180000
180100
180200
180300
180400
180500
180600
180700
180800
180900
181000
181100
181200
181300
181400
181500
181600
181700
181800
181900
182000
182100
182200
182300
182400
182500
182600
182700
182800
182900
183000
183100
183200
183300
183400
183500
183600
183700
183800
183900
184000
184100
184200
184300
184400
184500
184600
184700
184800
184900
185000
185100
185200
185300
185400
185500
185600
185700
185800
185900
186000
186100
186200
186300
186400
186500
186600
186700
186800
186900
187000
187100
187200
187300
187400
187500
187600
187700
187800
187900
188000
188100
188200
188300
188400
188500
188600
188700
188800
188900
189000
189100
189200
189300
189400
189500
189600
189700
189800
189900
190000
190100
190200
190300
190400
190500
190600
190700
190800
190900
191000
191100
191200
191
```

```

22700      DO 200 I=1,M
22800          K=I+(J-1)*M
22900          IF(I.EQ.2.OR.I.EQ.M-1.OR.J.EQ.2.OR.J.EQ.N-1)THEN
23000              R(K)=0.
23100              G(K)=0.
23200              H(K)=1.
23300              KA(K)=0.
23400              P(K)=0.
23500              Q(K)=0.
23600          END IF
23700          IF(I.EQ.1.OR.I.EQ.M.OR.J.EQ.1.OR.J.EQ.N)THEN
23800              P(K)=0.
23900              R(K)=0.
24000              G(K)=0.
24100              KA(K)=0.
24200              H(K)=1.
24300              Q(K)=0.
24400          END IF
24500      200 CONTINUE
24600      RETURN
24700      END
24800      C
24900      C
25000      REAL FUNCTION QF(RHOO,FSTR,XI,YI,DEELT,AKERN)
25100      REAL RHOO,FSTR,DEELT,XI,YI,H,K
25200      C
25300      C This function gives the strenght of the cosh**2 source.
25400      C
25500          K=4.14677/AKERN
25600          IF(ABS(K*XI).GT.10..OR.ABS(K*YI).GT.10.)THEN
25700              QF=0.
25800          ELSE
25900              H=((K**2)/(4.*RHOO))/((COSH(K*XI))**2)
26000              QF=-H*FSTR*DEELT/((COSH(K*YI))**2)
26100          END IF
26200      RETURN
26300      END
26400      C
26500      C
26600      C
26700      REAL FUNCTION QG(RHOO,FSTR,A,B,DEELT,AKERN)
26800      REAL A,B,DEELT,AKERN,RHOO,FSTR
26900      C
27000      C This function gives the strength of the source, proposed by Gij's van
27100      C Kuik.
27200      C
27300          IF(ABS(A).LT.16.9.AND.ABS(B).LT.AKERN.AND.ABS(B).GT.1.E-15)THEN
27400              QG=-2.3*FSTR*DEELT*(B**2)/(RHOO*ABS(B)*((COSH(4.6*A)*AKERN)**2))
27500          ELSE
27600              QG=0.
27700          END IF
27800      END

```

```

28200      SUBROUTINE BEGWRD(L,M,N,U,V,OMO,OMN,PSI,DT,FSTR)
28300      INTEGER I,J,K,L,M,N
28400      REAL C,A,B,U(L),V(L),OMO(L),OMN(L),PSI(L)
28500      C
28600      C Here, the vorticity, streamfunction and velocities obtain starting
28700      C values.
28800      C
28900          DO 10 I=1,L
29000              OMO(I)=0.
29100              OMN(I)=0.
29200      10      PSI(I)=0.
29300          DO 20 J=1,N
29400              B=2.*(J-1)/(N-1)-1.
29500              DO 20 I=1,M
29600                  C=500.*DT*FSTR
29700                  A=2.*(I-1)/(M-1)-1.
29800                  K=I+(J-1)*M
29900                  IF(B**2.GT..0001)U(K)=C*B*(1.-B**2)/SQRT(B**2)*(1.-A**2)
30000                  IF(B**2.LE..0001)U(K)=C*99.99*B*(1.-A**2)
30100                  IF(A**2.GT..0001)V(K)=-C*A*(1.-A**2)/SQRT(A**2)*(1.-B**2)
30200      20      IF(A**2.LE..0001)V(K)=-C*99.99*A*(1.-B**2)
30300      RETURN
30400      END

```

STRIP/DEVELOP

DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:13:32.

```

10000 C Strip/develop calculates the time-dependent velocity field.
10100 C generated by a finite actuator strip.
10200 C The theory behind this program is discussed in the report R-898-A
10300 C of the faculty of Physics, Eindhoven Technical University.
10400 C
10500 FILE 8(KIND='PRINTER')
10600 FILE 5(KIND='REMOTE')
10700 BLOCK GLOBALS
10800 $INCLUDE 'PLOTTER/F77/DECLARATION ON APPL.'
10900 C
11000 C LIB contains the subroutine MARQUA,
11100 C MLIB contains the subroutine MSTIMP.
11200 C
11300 LIBRARY LIB(TITLE='(RCLC1)OBJECT/FSLIB/CP ON USER1.')
11400 LIBRARY NAGP(TITLE='LIBRARY/NAGFLIB/F ON APPL.')
11500 LIBRARY MLIB(TITLE='OBJECT/STRIP/MAINBIEB.')
11600 END
11700 $INCLUDE 'NAGFLIB/F/FORTRAN/F04ATF ON APPL.'
11800 $INCLUDE 'PLOTTER/F77/SKIPP ON APPL.'
11900 $INCLUDE 'PLOTTER/F77/JOINOR ON APPL.'
12000 $INCLUDE 'PLOTTER/F77/SURF2 ON APPL.'
12100 $INCLUDE 'PLOTTER/F77/DISPOB ON APPL.'
12200 $INCLUDE 'PLOTTER/F77/FTEXT ON APPL.'
12300 $INCLUDE 'PLOTTER/F77/MAPOB1 ON APPL.'
12400 $INCLUDE 'PLOTTER/F77/CLEARO ON APPL.'
12500 $INCLUDE 'PLOTTER/F77/DRAWOB ON APPL.'
12600 $INCLUDE 'PLOTTER/F77/FAXIS ON APPL.'
12700 $INCLUDE 'PLOTTER/F77/POLYGN ON APPL.'
12800 $INCLUDE 'PLOTTER/F77/AXISCO ON APPL.'
12900 $INCLUDE 'PLOTTER/F77/NEWORJ ON APPL.'
13000 C
13100 C In the following, the text of the various subroutines is included
13200 C in the text of this programme.
13300 C This is done in this way, because in most of the subroutines the
13400 C same variables are used as in the main programme by use of COMMON-
13500 C blocks. This reduces the computing time.
13600 C
13700 $INCLUDE 'STRIP/BIEB/DERIVA'
13800 $INCLUDE 'STRIP/BIEB/WERVLV'
13900 $INCLUDE 'STRIP/BIEB/MARQUA'
14000 $INCLUDE 'STRIP/BIEB/STAROP'
14100 $INCLUDE 'STRIP/BIEB/STAREX'
14200 $INCLUDE 'STRIP/BIEB/SETCOE'
14300 $INCLUDE 'STRIP/BIEB/TRANIO'
14400 $INCLUDE 'STRIP/BIEB/KOWDV'
14500 $INCLUDE 'STRIP/BIEB/RVWOME'
14600 $INCLUDE 'STRIP/BIEB/MSTIMP'

```

```

14700 $INCLUDE 'STRIP/BIEB/XYTOOM'
14800 $INCLUDE 'STRIP/BIEB/FIE'
14900 $INCLUDE 'STRIP/BIEB/TRANIP'
15000 $INCLUDE 'STRIP/BIEB/KOPOIS'
15100 $INCLUDE 'STRIP/BIEB/RVWPSI'
15200 $INCLUDE 'STRIP/BIEB/TRVELO'
15300 $INCLUDE 'STRIP/BIEB/REGGRI'
15400 $INCLUDE 'STRIP/BIEB/DRWGRD'
15500 $INCLUDE 'STRIP/BIEB/DRAWUV'
15600 $INCLUDE 'STRIP/BIEB/DRAWOM'
15700 $INCLUDE 'STRIP/BIEB/UITVOE'
15800 LOGICAL KOPJE, LDGRID, LDOME, LDVEL, LSTART, LCOME, LCGRID, LCPUV,
15900 * LOFILE, LOUTPU
16000 CHARACTER*30 NIFGRI, NIFRES, NOFGRI, NOFRES
16100 INTEGER FILENO, NCONLN, NDUM, NITMO, NITMP, NITTO, NITTP
16200 REAL XWMIN, XWMAX, YWMIN, YWMAX, DUM
16300 $INCLUDE 'STRIP/BIEB/COMMON'
16400 C
16500 C STRIP/BIEB/COMMON contains the declaration of the variables and
16600 C the common blocks.
16700 C
16800 C STRIP/PARAM contains the parameters that direct the calculations:
16900 C XMIN: x-coordinate of left boundary
17000 C XMAX: " " " right "
17100 C YMAX: y-coordinate of upper boundary
17200 C D: y-coordinate of upper edge of actuator strip.
17300 C DAKS: ETA-coordinate of upper edge.
17400 C M: number of grid points in x-direction.
17500 C N: number of grid points in y-direction.
17600 C TBEGIN: starting time of the calculation.
17700 C TFIN: ending time.
17800 C NT: number of time steps.
17900 C EPOMME: mean accuracy in the calculation of the vorticity
18000 C EPOMMA: maximum " " " " " " "
18100 C EPSIME: mean accuracy " " " " " streamfunction
18200 C EPSIMA: maximum " " " " " "
18300 C NITMO: maximum number of iterations in calculation of vorticity
18400 C NITMP: " " " " " " streamfunction
18500 C XTIMES: ratio between mean vorticity and vorticity of vortex sheet
18600 C STOPCR: stopcriterion in the determination of least-squares fit to
18700 C the points with local maximum vorticity.
18800 C LSTART: logical: if true, a starting grid is computed and no input
18900 C file is read from disc.
19000 C LCOME: logical: if true, the vorticity is computed.
19100 C LCGRID: " " " a new grid is set up.
19200 C LCPUV: " " " streamfunction and velocity are calculated.
19300 C LOFILE: " " " the results are saved on disc.
19400 C LOUTPU: " " " the results are printed.
19500 C KOPJE: " " " the calculation parameters are printed if
19600 C the results are printed.
19700 C NIFGRI, NIFRES, NOFGRI, NOFRES: Names of Input or Output Files for the

```

A2. Programme-source

```

19800 C GRId and partial derivatives or for the 24800
19900 C REsults. 24900
20000 C 25000
20100 OPEN(10, FILE='STRIP/PARAM.', STATUS='OLD') 25100
20200 READ(10,*)XMIN, XMAX, YMAX, D, DAKS, M, N, NHETA, TBEGIN, TEIND, NT 25200
20300 READ(10,*)E, POMME, EPOMMA, EPSIME, EPSIMA, NITMO, NITMP 25300
20400 READ(10,*)XTIMES, STOPCR, NOUTQF, EA, ER, ALFAO, ALFAP, NPAR, WIDTH 25400
20500 READ(10,*)LDGRID, LDOME, LDVEL, NCONLN, DIAGNO 25500
20600 READ(10,*)XIMIN, XIMAX, ETYMIN, ETYMAX, NCPL, NITPMX, NITPMY 25600
20700 READ(10,*)XWMIN, XWMAX, YWMIN, YWMAX 25700
20800 READ(10,*)LSTART, LCOME, LCGRID, LCPUV, LOFILE, LOPTU, KOPJE 25800
20900 READ(10,*)NIFGRI 25900
21000 READ(10,*)NIFRES 26000
21100 READ(10,*)NOFGRI 26100
21200 READ(10,*)NOFRES 26200
21300 L=M*N 26300
21400 C 26400
21500 C STAROP sets up a physical grid plus a computational grid and sets 26500
21600 C OM, PSI, U and V zero. 26600
21700 C 26700
21800 C STAREX only sets up a computational grid. 26800
21900 C 26900
22000 IF(LSTART)CALL STAROP 27000
22100 IF(.NOT.LSTART)CALL STAREX 27100
22200 DT=(TEIND-TBEGIN)/NT 27200
22300 FILENO=8 27300
22400 RHO0=1.2 27400
22500 NU=1.45E-5 27500
22600 IF(LSTART)GO TO 70 27600
22700 OPEN(20, STATUS='OLD', FILE=NIFGRI) 27700
22800 OPEN(30, STATUS='OLD', FILE=NIFRES) 27800
22900 READ(20,*)DUM, DUM, DUM, DUM, DUM, NDUM, NDUM, DUM, DUM, NDUM 27900
23000 READ(20,*)DUM, DUM, DUM, DUM, DUM, NDUM, DUM 28000
23100 READ(20,*)DUM, DUM, DUM, DUM, DUM, NDUM, DUM 28100
23200 READ(30,*)DUM, DUM, DUM, DUM, DUM, NDUM, DUM, DUM, NDUM 28200
23300 READ(30,*)DUM, DUM, DUM, DUM, DUM, NDUM, DUM 28300
23400 READ(30,*)DUM, DUM, DUM, DUM, DUM, NDUM, DUM 28400
23500 DO 50 J=0, N-1 28500
23600 READ(20,*)(X(J*M+I), Y(J*M+I), KSIX(J*M+I), KSIXX(J*M+I), EX(J*M+I), 28600
23700 * EY(J*M+I), EXX(J*M+I), EYY(J*M+I), I=1, M) 28700
23800 READ(30,*)(OM(J*M+I), PSI(J*M+I), U(J*M+I), V(J*M+I), I=1, M) 28800
23900 50 CONTINUE 28900
24000 CLOSE(20, STATUS='CRUNCH') 29000
24100 CLOSE(30, STATUS='CRUNCH') 29100
24200 70 CONTINUE 29200
24300 C 29300
24400 C Beginning of the loop, that is repeated every time step. 29400
24500 C 29500
24600 C TIT=0 29600
24700 100 CONTINUE 29700

```

```

TIT=TIT+I
IF(.NOT.LCOME)GO TO 107
C
C SETCOE calculates the coefficients for the vorticity diffusion
C equation.
C TRANIO transforms this coefficients into the computational plane.
C KOWDV determines the coefficients for the discretised, transformed
C vorticity diffusion equation.
C RVWOME determines boundary conditions.
C
CALL SETCOE
CALL TRANIO
CALL KOWDV
CALL RVWOME
NITO=0
C
C MSTIMP caculates a solution to a nine point molecule difference
C scheme, according to the modified strongly implicit procedure
C
CALL MSTIMP(M, N, OM, AO, BO, CO, DO, EO, FO, GO, HO, IO, JO, H1, H2, H3, H4, H5,
* H6, H7, H8, H9, H10, H11, H12, ALFAO, EPOMME, EPOMMA, NITMO, NITO)
NITTO=NITTO+NITO
WRITE(5, 105)TIT, NITO
105 FORMAT(1X, 'Time iteration number ', I4, '/', I4,
* 'Vorticity diffusion equation solved, ', I4, ' iterations used.')
107 CONTINUE
IF(.NOT.LCGRID)GO TO 110
C
C WERVLV computes the coordinates of the local maxima in the vorticity
C distribution and fits a function to these points.
C This function then serves as a base for the transformation of
C the grid.
C XYTOOM generates a grid by calculation of the transformation from
C physical to computational grid.
C
CALL WERVLV(XTIMES, STOPCR, NOUTQF, EA, ER)
CALL XYTOOM
WRITE(5, 120)TIT
120 FORMAT(1X, 'Time iteration number ', I4, '/', I4, 'New grid plus ',
* 'partial derivatives determined.')
110 CONTINUE
IF(.NOT.LCPUV)GO TO 115
DO 112 I=1, L
112 ZP1(I)=-OM(I)
C
C TRANIP, KOPOIS and RVWPSI: same as TRANIO, KOWDV and RVWOME,
C respectively, but working on the Poisson-PSI equation.
C
CALL TRANIP
CALL KOPOIS

```



```

29800 CALL RWVPS I
29900 NITP=0
30000 CALL MSTIMP(M,N,PSI,AP,BP,CP,DP,EP,FP,GP,HP,IP,JP,HI,H2,H3,H4,H5,
30100 * H6,H7,H8,H9,H10,H11,H12,ALFAP,EPSTME,EPSTMA,NITMP,NITP)
30200 NITTP=NITTP+NITP
30300 C
30400 C TRVELO computes the transformed velocity components from the
30500 C streamfunction.
30600 C
30700 CALL TRVELO
30800 WRITE(5,114)TIT,NITP
30900 114 FORMAT(1X,'Time iteration number',I4,/,1X,
31000 * 'streamfunction calculated: ',I4,' iterations used.')
31100 115 CONTINUE
31200 IF(TIT.LT.NT)THEN
31300 DO 140 I=1,L
31400 140 OMO(I)=OM(I)
31500 END IF
31600 IF(TIT.LT.NT)GO TO 100
31700 150 CONTINUE
31800 IF(.NOT.LOFILE)GO TO 205
31900 OPEN(20,STATUS='NEW',FILE=NOPGRI)
32000 OPEN(30,STATUS='NEW',FILE=NOPRES)
32100 C
32200 C In FILE20, the coordinates of the gridpoints and the partial
32300 C derivatives of the transformation are stored.
32400 C FILE30 is filled with the vorticity, streamfunction and velocity
32500 C components in each grid point.
32600 C
32700 WRITE(20,*)XMIN,XMAX,YMAX,D,DAKS,M,N,TBEGIN,TEIND,NT
32800 WRITE(30,*)XMIN,XMAX,YMAX,D,DAKS,M,N,TBEGIN,TEIND,NT
32900 WRITE(20,*)EPOMME,EPOMMA,EPSTME,EPSTMA
33000 WRITE(30,*)EPOMME,EPOMMA,EPSTME,EPSTMA
33100 WRITE(20,*)XTIMES,STOPCR,EA,ER,ALFAO,ALFAP,NPAR,WIDTH
33200 WRITE(30,*)XTIMES,STOPCR,EA,ER,ALFAO,ALFAP,NPAR,WIDTH
33300 DO 200 J=0,N-1
33400 WRITE(20,*)(X(J*M+I),Y(J*M+I),KSIX(J*M+I),KSIXX(J*M+I),EX(J*M+I),
33500 * EY(J*M+I),EXX(J*M+I),EYY(J*M+I),I=1,M)
33600 WRITE(30,*)(OM(J*M+I),PSI(J*M+I),U(J*M+I),V(J*M+I),I=1,M)
33700 200 CONTINUE
33800 CLOSE(20,STATUS='CRUNCH')
33900 CLOSE(30,STATUS='CRUNCH')
34000 205 CONTINUE
34100 IF(.NOT.LDGRID)GO TO 210
34200 C
34300 C DRWGRD makes a plot of the physical grid with lines KSI=constant and

```

```

34400 C ETA=constant.
34500 C
34600 CALL DRWGRD(X,XWMIN,XWMAX,M,Y,YWMIN,YWMAX,N,'X-axis',
34700 * 'Y-axis',NOFGRI)
34800 210 CONTINUE
34900 IF(.NOT.LDVFL)GO TO 220
35000 C
35100 C DRAWUV makes a plot of streamlines (PSI=constant): the number of
35200 C streamlines drawn is NCONLN.
35300 C
35400 CALL DRAWUV(X,Y,PSI,M,N,XWMIN,XWMAX,YWMIN,YWMAX,TBEGIN,TEIND,
35500 * NT,NCONLN)
35600 220 CONTINUE
35700 IF(.NOT.LDOME)GO TO 230
35800 C
35900 C REGGRI,SHIFTO and DRAWOM produce a plot of the vorticity in a
36000 C perspective view.
36100 C
36200 CALL REGGRI
36300 CALL SHIFTOY
36400 CALL DRAWOM(X,Y,M,N,OM,XWMIN,XWMAX,YWMIN,YWMAX,NOFGRI)
36500 230 CONTINUE
36600 IF(.NOT.LOUTPU)GO TO 300
36700 C
36800 C UITVOE prints the results, and, if KOPJE is true, also a
36900 C list of the calculation parameters.
37000 C
37100 CALL UITVOE(KOPJE,L,X,Y,OM,PSI,U,V,FILENO)
37200 300 CONTINUE
37300 1000 CONTINUE
37400 END
STRIP/PARAM
DATE & TIME PRINTED: MONDAY, FEBRUARY 29, 1988 @ 13:28:37.
100 -100.,100.,100.,5.,50.,41,21,0.000,0.001,1
200 1.E-5,1.E-5,1.E-5,1.E-5,200,200
300 2.0,12,1,1.E-3,1.E-2,.5,.5,5,-25
400 F,F,F,10
500 1.E-1,10.,1.E-1,10.,5,500,1000,6
600 -20.,20.,0.,20.
700 F,T,T,T,T,T,T
800 'AS/GRID/41B21/1/1000.'
900 'AS/OPUV/41B21/1/1000.'
1000 'AS/GRID/41B21/1/1000.'
1100 'AS/OPUV/41B21/1/1000.'
1200 'DATA FOR THE CALCULATION OF FLOW THROUGH FINITE ACTUATOR STRIP'
1300 'LINE 1: XMIN,XMAX,YMAX,D,DAKS,M,N,TBEGIN,TEIND,NT'
1400 'LINE 2: EPOMME,EPOMMA,EPSTME,EPSTMA,NITMO,NITMP'
1500 'LINE 3: XTIMES,STOPCR,NOUTQF,EA,ER,ALFAO,ALFAP,NPAR,WIDTH'

```

```

1600  "LINE 4: LDGRID, LDOME, LDVEL, NCONLN"
1700  "LINE 5: XIXMIN, XIXMAX, ETYMIN, ETYMAX, NCPL, NITPMX, NITPMY, NHETA"
1800  "LINE 6: XWMIN, XWMAX, YWMIN, YWMAX"
1900  "LINE 7: LSTART, LCOME, LCGRID, LCPUV, LOFILE, LOUPTU, KOPJE"
2000  "LINE 8: NIFGR1"
2100  "LINE 9: NIFRES"
2200  "LINE 10: NOFGR1"
2300  "LINE 11: NOFRES"

```

STRIP/BIEB/MARQUA  
DATE & TIME PRINTED: FRIDAY, DECEMBER 4, 1987 @ 15:49:31.

```

100      SUBROUTINE MARQUA(N,R,M,P,RES,STOPCR,EA,ER,ITMAX,OUTPUT,FNR)
110      INTEGER N,M,STOPCR,ITMAX,OUTPUT,FNR
120      REAL EA,ER,R(N),P(M)
130      EXTERNAL RES
140      IN LIBRARY LIB
150      END

```

STRIP/BIEB/STAROP  
DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:15:00.

```

1000      SUBROUTINE STAROP
1010      C
1020      C This routine sets up a starting grid.
1030      C The lines ksi=constant and eta=constant are straight lines.
1040      C In x-direction the grid points are distributed in the following way:
1050      C m-4 grid points are equally distributed between x=xmin/3 and
1060      C x=xmax/3.
1070      C The outer 4 points have x-coordinates xmin,xmin/2,xmax/2 and xmax.
1080      C
1090      C In y-direction: between y=0 and y=3*D 3*NY3 points are equally
1100      C distributed. The other points have coordinates ymax, ymax/2,
1110      C ymax/3, etcetera.
1120      C
1130      C The partial derivatives ksix, ksixx, ey and eyy are determined from
1140      C a second order Taylor series expansion. ksixx and eyy are set zero
1150      C on the boundaries.
1160      C
1170      C Finally, the starting values of PSI, OM, U, V and EXY are set zero.
1180      C
1185      REAL S1,S2,S22
1190      INCLUDE "STRIP/BIEB/COMMON"
1200      MH=(M+1)/2
1210      NH=(N+1)/2
1220      NY3=3*(NHETA-1)+1
1230      NX3=M-6
1240      DEX=(XMAX-XMIN)/(3.*FLOAT(NX3+1))
1250      DEY=3*D/FLOAT(NY3-1)
1260      DEXI=(XMAX-XMIN)/(M-1)

```

```

1270      DEET=YMAX/(N-1)
1280      DO 50 J=1,N
1290      DO 50 I=1,M
1300      K=(J-1)*M+I
1310      KSI(K)=XMIN+(I-1)*DEXI
1320      ETA(K)=(J-1)*DEET
1330
50      CONTINUE
1340      DO 55 I=4,MH-1
1350      X(I)=XMIN/3+(I-3)*DEX
1360      DO 57 I=MH+1,M-3
1370      X(I)=XMAX/3-(NX3-I+4)*DEX
1380      X(1)=XMIN
1390      X(2)=XMIN/2
1400      X(3)=XMIN/3
1410      X(MH)=0.
1420      X(M-2)=XMAX/3
1430      X(M-1)=XMAX/2
1440      X(M)=XMAX
1450      DO 60 J=2,NY3
1460      Y((J-1)*M+1)=(J-1)*DEY
1470      Y(1)=0.
1480      DO 65 I=1,N-NY3
1490      Y((N-I)*M+1)=YMAX/I
65      DO 70 I=2,M-1
1500      S1=X(I)-X(I-1)
1510      S2=(X(I+1)-X(I))/S1
1520      S22=S2*S2
70      KSI(X(I))=(KSI(X(I+1))-S22*KSI(X(I-1))+(S22-1)*KSI(X(I)))/(S1*S2*(1+S2))
1530      KSI(X(1))=KSI(X(2))+X(1)-X(2)*(KSI(X(3))-KSI(X(2)))/(X(3)-X(2))
1540      KSI(X(M))=KSI(X(M-1))+X(M)-X(M-1)*(KSI(X(M-1))-KSI(X(M-2)))/
1550      * (X(M-1)-X(M-2))
1560      DO 80 J=2,N-1
1570      K=(J-1)*M+1
1580      S1=Y(K)-Y(K-M)
1590      S2=(Y(K+M)-Y(K))/S1
80      EY(K)=(ETA(K+M)-S22*ETA(K-M)+(S22-1)*ETA(K))/(S1*S2*(1+S2))
1600      EY(1)=EY(M+1)+(Y(1)-Y(M+1))*(EY(1+2*M)-EY(1+M))/(Y(1+2*M)-Y(1+M))
1610      K=(N-1)*M+1
1620      EY(K)=EY(K-M)+(Y(K)-Y(K-M))*(EY(K-M)-EY(K-2*M))/(Y(K-M)-Y(K-2*M))
1630      DO 85 I=2,M-1
1640      S1=X(I)-X(I-1)
1650      S2=(X(I+1)-X(I))/S1
85      KSI(X(I))=(KSI(X(I+1))-S22*KSI(X(I-1))+(S22-1)*KSI(X(I)))/(S1*S2*(1+S2))
1660      KSI(X(1))=0.
1670      KSI(X(M))=0.
1680      DO 90 J=2,N-1
1690      K=(J-1)*M+1
1700      S1=Y(K)-Y(K-M)
1710      S2=(Y(K+M)-Y(K))/S1
90      DEET=YMAX/(N-1)
1720      DO 50 J=1,N
1730      DO 50 I=1,M
1740      K=(J-1)*M+I
1750      KSI(K)=XMIN+(I-1)*DEXI
1760      ETA(K)=(J-1)*DEET

```

```

1770 90) EYY(K)=(EY(K+M)-S22*EY(K-M)+(S22-1)*EY(K))/(S1*S2*(1+S2))
1780 EYY(I)=0.
1790 EYY((N-1)*M+1)=0.
1800 DO 170 J=1,N
1810 DO 170 I=1,M
1820 K=(J-1)*M+I
1830 X(K)=X(I)
1840 Y(K)=Y((J-1)*M+1)
1850 KSIX(K)=KSIX(I)
1860 KSIXX(K)=KSIXX(I)
1870 EY(K)=EY((J-1)*M+1)
1880 EYY(K)=EYY((J-1)*M+1)
1890 170 CONTINUE
1900 DO 100 I=1,L
1910 EXY(I)=0.
1920 OMO(I)=0.
1930 PSI(I)=0.
1940 U(I)=0.
1950 100 V(I)=0.
1960 END

```

STRIP/BIEB/STAREX  
DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:16:50.

```

500 SUBROUTINE STAREX
510 C
520 C setting values of the eta- and ksi-coordinates of the computational
530 C grid.
540 C
550 INTEGER L,I,M,N,J,K
560 REAL XMIN,XMAX,YMAX,
570 * DEET,DEXT,KSIX(10000),ETA(10000)
580 COMMON /SIZE/M,N,L,/CONFIG/XMIN,XMAX,YMAX
590 COMMON /TRCOOR/KSIX,ETA
600 DEXT=(XMAX-XMIN)/(M-1)
610 DEET=YMAX/(N-1)
620 DO 50 J=1,N
630 DO 50 I=1,M
640 K=(J-1)*M+I
650 KSIX(K)=XMIN+(I-1)*DEXT
660 ETA(K)=(J-1)*DEET
670 50 CONTINUE
680 END

```

STRIP/BIEB/SETCOE  
DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:23:18.

79

```

1000 C
1010 C In this subroutine, the coefficients for the ordinary vorticity
1020 C diffusion equation are given a value: in grid point I the equation
1030 C reads:

```

```

1040 C
1050 C  $d\Omega/dt = Z01(I) + Z02(I)d\Omega/dx + Z03(I)d\Omega/dy + Z04(I)d\Omega/dx dx +$ 
1060 C  $+ Z05(I)d\Omega/dy dy$ 
1070 C
1080 C  $\Omega$  is the vorticity.
1090 SUBROUTINE SETCOE
1100 $INCLUDE 'STRIP/BIEB/COMMON'
1110 DO 100 I=1,L
1120 Z01(I)=(-1./RHO0)*SOURCE(X(I),Y(I))
1130 Z02(I)=-U(I)
1140 Z03(I)=-V(I)
1150 Z04(I)=NU
1160 Z05(I)=NU
1170 100 CONTINUE
1180 END

```

C\*\*\*\*\*

```

1190 C
1200 REAL FUNCTION SOURCE(X,Y)
1210 REAL X,Y,W,D,K
1220 C
1230 C The strength of the vorticity-source at X,Y is given by this
1240 C function. W is the width of the point-like source.
1250 C
1260 C Strength of the source at  $|X|=W$  or  $|Y-D|=W$  is 1/1000 of the maximum
1270 C
1280 COMMON /DIAM/D,W
1290 K=4.14677/W
1300 IF (ABS(K*X).GT.10..OR.ABS(K*(Y-D)).GT.10.)THEN
1310 SOURCE=0.
1320 ELSE
1330 SOURCE=(-K**2)/(4.*(COSH(K*X)*COSH(K*(Y-D)))**2)
1340 END IF
1350 END
1360

```

STRIP/BIEB/TRANIO  
DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:25:50.

```

2000 SUBROUTINE TRANIO
2010 $INCLUDE 'STRIP/BIEB/COMMON'
2020 C
2030 C transformation of the coefficients in the vorticity-diffusion equation
2040 C into the computational plane.
2050 C
2060 DO 100 I=1,L
2070 ZOT1(I)=Z01(I)
2080 ZOT2(I)=Z02(I)*KSIX(I)+Z04(I)*KSIXX(I)
2090 ZOT3(I)=Z02(I)*EX(I)+Z03(I)*EY(I)+Z04(I)*EXX(I)+Z05(I)*EYY(I)
2100 ZOT4(I)=Z04(I)*(KSIX(I)**2)
2110 ZOT5(I)=Z04(I)*(EX(I)**2)+Z05(I)*(EY(I)**2)
2120 ZOT6(I)=2*Z04(I)*KSIX(I)*EX(I)
2130 100 CONTINUE
2140 END

```

STRIP/BIEB/KOWDV  
 DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:27:12.

```

3500      SUBROUTINE KOWDV
3510      C
3520      C determination of the coefficients for the discretised, transformed
3525      C vorticity diffusion equation.
3530      C
3540      REAL DEXI,DEET
3550      $INCLUDE`STRIP/BIEB/COMMON`
3560      DEXI=(XMAX-XMIN)/(M-1)
3570      DEET=YMAX/(N-1)
3580      DO 100 I=1,L
3590          AO(I)=DT*ZOT6(I)/(4*DEXI*DEET)
3600          BO(I)=DT*(-ZOT3(I)/(2*DEET)+ZOT5(I)/(DEET**2))
3610          CO(I)=-AO(I)
3620          DO(I)=DT*(-ZOT2(I)/(2*DEXI)+ZOT4(I)/(DEXI**2))
3630          EO(I)=-1.+DT*(-2*ZOT4(I)/(DEXI**2)-2*ZOT5(I)/(DEET**2))
3640          FO(I)=DT*(ZOT2(I)/(2*DEXI)+ZOT4(I)/(DEXI**2))
3650          GO(I)=-AO(I)
3660          HO(I)=DT*(ZOT3(I)/(2*DEET)+ZOT5(I)/(DEET**2))
3670          IO(I)=AO(I)
3680          JO(I)=-OMO(I)-DT*ZOT1(I)
3690      100 CONTINUE
3700      END
3710      C
  
```

STRIP/BIEB/RVWOME  
 DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:30:52.

```

2000      SUBROUTINE RVWOME
2010      C
2020      C this subroutine calculates the effect of the boundary conditions on
2030      C the coefficients in the differential scheme of the vorticity-diffu-
2040      C sion equation.
2050      C
2060      REAL DEXI,DEET,Q(10000),A
2070      $INCLUDE`STRIP/BIEB/COMMON`
2080      DEXI=(XMAX-XMIN)/(M-1)
2090      DEET=YMAX/(N-1)
2100      C
2110      C boundary conditions for the lower boundary y=0: mirror
2120      C
2130      DO 100 I=1,M
2140          GO(I)=GO(I)-AO(I)
2150          HO(I)=HO(I)-BO(I)
2160          IO(I)=IO(I)-CO(I)
2170      100 CONTINUE
2180      C
  
```

```

2190      C right boundary: X=XMAX, vorticity equal to zero.
2200      C
2210      DO 300 J=1,N
2220          K=J*M
2230          EO(K)=1.
2240          AO(K)=0.
2250          BO(K)=0.
2260          CO(K)=0.
2270          DO(K)=0.
2280          FO(K)=0.
2290          JO(K)=0.
2300      300 CONTINUE
2310      C
2320      C upper boundary: Y=YMAX, vorticity equal zero
2330      C
2340      DO 400 I=M,1,-1
2350          K=I+(N-1)*M
2360          EO(K)=1.
2370          AO(K)=0.
2380          BO(K)=0.
2390          CO(K)=0.
2400          DO(K)=0.
2410          FO(K)=0.
2420          JO(K)=0.
2430      400 CONTINUE
2440      C
2450      C left boundary: X=XMIN, vorticity equal zero
2460      C
2470      DO 500 J=N,1,-1
2480          K=(J-1)*M+1
2490          EO(K)=1.
2500          IO(K)=0.
2510          FO(K)=0.
2520          CO(K)=0.
2530          HO(K)=0.
2540          BO(K)=0.
2550          JO(K)=0.
2560      C
2570      C setting coefficients for points outside grid zero
2580      C
2590          AO(K)=0.
2600          DO(K)=0.
2610          GO(K)=0.
2620      500 CONTINUE
2630      C
2640      DO 510 I=1,M
2650          K=I
2660          AO(K)=0.
2670          BO(K)=0.
2680          CO(K)=0.
2690          DO(K)=0.
2700          FO(K)=0.
2710          JO(K)=0.
2720          HO(K)=0.
2730          IO(K)=0.
2740          BO(K)=0.
2750          CO(K)=0.
2760          DO(K)=0.
2770          FO(K)=0.
2780          JO(K)=0.
2790          HO(K)=0.
2800          IO(K)=0.
2810          BO(K)=0.
2820          CO(K)=0.
2830          DO(K)=0.
2840          FO(K)=0.
2850          JO(K)=0.
2860          HO(K)=0.
2870          IO(K)=0.
2880          BO(K)=0.
2890          CO(K)=0.
2900          DO(K)=0.
2910          FO(K)=0.
2920          JO(K)=0.
2930          HO(K)=0.
2940          IO(K)=0.
2950          BO(K)=0.
2960          CO(K)=0.
2970          DO(K)=0.
2980          FO(K)=0.
2990          JO(K)=0.
3000          HO(K)=0.
3010          IO(K)=0.
3020          BO(K)=0.
3030          CO(K)=0.
3040          DO(K)=0.
3050          FO(K)=0.
3060          JO(K)=0.
3070          HO(K)=0.
3080          IO(K)=0.
3090          BO(K)=0.
3100          CO(K)=0.
3110          DO(K)=0.
3120          FO(K)=0.
3130          JO(K)=0.
3140          HO(K)=0.
3150          IO(K)=0.
3160          BO(K)=0.
3170          CO(K)=0.
3180          DO(K)=0.
3190          FO(K)=0.
3200          JO(K)=0.
3210          HO(K)=0.
3220          IO(K)=0.
3230          BO(K)=0.
3240          CO(K)=0.
3250          DO(K)=0.
3260          FO(K)=0.
3270          JO(K)=0.
3280          HO(K)=0.
3290          IO(K)=0.
3300          BO(K)=0.
3310          CO(K)=0.
3320          DO(K)=0.
3330          FO(K)=0.
3340          JO(K)=0.
3350          HO(K)=0.
3360          IO(K)=0.
3370          BO(K)=0.
3380          CO(K)=0.
3390          DO(K)=0.
3400          FO(K)=0.
3410          JO(K)=0.
3420          HO(K)=0.
3430          IO(K)=0.
3440          BO(K)=0.
3450          CO(K)=0.
3460          DO(K)=0.
3470          FO(K)=0.
3480          JO(K)=0.
3490          HO(K)=0.
3500          IO(K)=0.
3510          BO(K)=0.
3520          CO(K)=0.
3530          DO(K)=0.
3540          FO(K)=0.
3550          JO(K)=0.
3560          HO(K)=0.
3570          IO(K)=0.
3580          BO(K)=0.
3590          CO(K)=0.
3600          DO(K)=0.
3610          FO(K)=0.
3620          JO(K)=0.
3630          HO(K)=0.
3640          IO(K)=0.
3650          BO(K)=0.
3660          CO(K)=0.
3670          DO(K)=0.
3680          FO(K)=0.
3690          JO(K)=0.
3700          HO(K)=0.
3710          IO(K)=0.
3720          BO(K)=0.
3730          CO(K)=0.
3740          DO(K)=0.
3750          FO(K)=0.
3760          JO(K)=0.
3770          HO(K)=0.
3780          IO(K)=0.
3790          BO(K)=0.
3800          CO(K)=0.
3810          DO(K)=0.
3820          FO(K)=0.
3830          JO(K)=0.
3840          HO(K)=0.
3850          IO(K)=0.
3860          BO(K)=0.
3870          CO(K)=0.
3880          DO(K)=0.
3890          FO(K)=0.
3900          JO(K)=0.
3910          HO(K)=0.
3920          IO(K)=0.
3930          BO(K)=0.
3940          CO(K)=0.
3950          DO(K)=0.
3960          FO(K)=0.
3970          JO(K)=0.
3980          HO(K)=0.
3990          IO(K)=0.
4000          BO(K)=0.
4010          CO(K)=0.
4020          DO(K)=0.
4030          FO(K)=0.
4040          JO(K)=0.
4050          HO(K)=0.
4060          IO(K)=0.
4070          BO(K)=0.
4080          CO(K)=0.
4090          DO(K)=0.
4100          FO(K)=0.
4110          JO(K)=0.
4120          HO(K)=0.
4130          IO(K)=0.
4140          BO(K)=0.
4150          CO(K)=0.
4160          DO(K)=0.
4170          FO(K)=0.
4180          JO(K)=0.
4190          HO(K)=0.
4200          IO(K)=0.
4210          BO(K)=0.
4220          CO(K)=0.
4230          DO(K)=0.
4240          FO(K)=0.
4250          JO(K)=0.
4260          HO(K)=0.
4270          IO(K)=0.
4280          BO(K)=0.
4290          CO(K)=0.
4300          DO(K)=0.
4310          FO(K)=0.
4320          JO(K)=0.
4330          HO(K)=0.
4340          IO(K)=0.
4350          BO(K)=0.
4360          CO(K)=0.
4370          DO(K)=0.
4380          FO(K)=0.
4390          JO(K)=0.
4400          HO(K)=0.
4410          IO(K)=0.
4420          BO(K)=0.
4430          CO(K)=0.
4440          DO(K)=0.
4450          FO(K)=0.
4460          JO(K)=0.
4470          HO(K)=0.
4480          IO(K)=0.
4490          BO(K)=0.
4500          CO(K)=0.
4510          DO(K)=0.
4520          FO(K)=0.
4530          JO(K)=0.
4540          HO(K)=0.
4550          IO(K)=0.
4560          BO(K)=0.
4570          CO(K)=0.
4580          DO(K)=0.
4590          FO(K)=0.
4600          JO(K)=0.
4610          HO(K)=0.
4620          IO(K)=0.
4630          BO(K)=0.
4640          CO(K)=0.
4650          DO(K)=0.
4660          FO(K)=0.
4670          JO(K)=0.
4680          HO(K)=0.
4690          IO(K)=0.
4700          BO(K)=0.
4710          CO(K)=0.
4720          DO(K)=0.
4730          FO(K)=0.
4740          JO(K)=0.
4750          HO(K)=0.
4760          IO(K)=0.
4770          BO(K)=0.
4780          CO(K)=0.
4790          DO(K)=0.
4800          FO(K)=0.
4810          JO(K)=0.
4820          HO(K)=0.
4830          IO(K)=0.
4840          BO(K)=0.
4850          CO(K)=0.
4860          DO(K)=0.
4870          FO(K)=0.
4880          JO(K)=0.
4890          HO(K)=0.
4900          IO(K)=0.
4910          BO(K)=0.
4920          CO(K)=0.
4930          DO(K)=0.
4940          FO(K)=0.
4950          JO(K)=0.
4960          HO(K)=0.
4970          IO(K)=0.
4980          BO(K)=0.
4990          CO(K)=0.
5000          DO(K)=0.
5010          FO(K)=0.
5020          JO(K)=0.
5030          HO(K)=0.
5040          IO(K)=0.
5050          BO(K)=0.
5060          CO(K)=0.
5070          DO(K)=0.
5080          FO(K)=0.
5090          JO(K)=0.
5100          HO(K)=0.
5110          IO(K)=0.
5120          BO(K)=0.
5130          CO(K)=0.
5140          DO(K)=0.
5150          FO(K)=0.
5160          JO(K)=0.
5170          HO(K)=0.
5180          IO(K)=0.
5190          BO(K)=0.
5200          CO(K)=0.
5210          DO(K)=0.
5220          FO(K)=0.
5230          JO(K)=0.
5240          HO(K)=0.
5250          IO(K)=0.
5260          BO(K)=0.
5270          CO(K)=0.
5280          DO(K)=0.
5290          FO(K)=0.
5300          JO(K)=0.
5310          HO(K)=0.
5320          IO(K)=0.
5330          BO(K)=0.
5340          CO(K)=0.
5350          DO(K)=0.
5360          FO(K)=0.
5370          JO(K)=0.
5380          HO(K)=0.
5390          IO(K)=0.
5400          BO(K)=0.
5410          CO(K)=0.
5420          DO(K)=0.
5430          FO(K)=0.
5440          JO(K)=0.
5450          HO(K)=0.
5460          IO(K)=0.
5470          BO(K)=0.
5480          CO(K)=0.
5490          DO(K)=0.
5500          FO(K)=0.
5510          JO(K)=0.
5520          HO(K)=0.
5530          IO(K)=0.
5540          BO(K)=0.
5550          CO(K)=0.
5560          DO(K)=0.
5570          FO(K)=0.
5580          JO(K)=0.
5590          HO(K)=0.
5600          IO(K)=0.
5610          BO(K)=0.
5620          CO(K)=0.
5630          DO(K)=0.
5640          FO(K)=0.
5650          JO(K)=0.
5660          HO(K)=0.
5670          IO(K)=0.
5680          BO(K)=0.
5690          CO(K)=0.
5700          DO(K)=0.
5710          FO(K)=0.
5720          JO(K)=0.
5730          HO(K)=0.
5740          IO(K)=0.
5750          BO(K)=0.
5760          CO(K)=0.
5770          DO(K)=0.
5780          FO(K)=0.
5790          JO(K)=0.
5800          HO(K)=0.
5810          IO(K)=0.
5820          BO(K)=0.
5830          CO(K)=0.
5840          DO(K)=0.
5850          FO(K)=0.
5860          JO(K)=0.
5870          HO(K)=0.
5880          IO(K)=0.
5890          BO(K)=0.
5900          CO(K)=0.
5910          DO(K)=0.
5920          FO(K)=0.
5930          JO(K)=0.
5940          HO(K)=0.
5950          IO(K)=0.
5960          BO(K)=0.
5970          CO(K)=0.
5980          DO(K)=0.
5990          FO(K)=0.
6000          JO(K)=0.
6010          HO(K)=0.
6020          IO(K)=0.
6030          BO(K)=0.
6040          CO(K)=0.
6050          DO(K)=0.
6060          FO(K)=0.
6070          JO(K)=0.
6080          HO(K)=0.
6090          IO(K)=0.
6100          BO(K)=0.
6110          CO(K)=0.
6120          DO(K)=0.
6130          FO(K)=0.
6140          JO(K)=0.
6150          HO(K)=0.
6160          IO(K)=0.
6170          BO(K)=0.
6180          CO(K)=0.
6190          DO(K)=0.
6200          FO(K)=0.
6210          JO(K)=0.
6220          HO(K)=0.
6230          IO(K)=0.
6240          BO(K)=0.
6250          CO(K)=0.
6260          DO(K)=0.
6270          FO(K)=0.
6280          JO(K)=0.
6290          HO(K)=0.
6300          IO(K)=0.
6310          BO(K)=0.
6320          CO(K)=0.
6330          DO(K)=0.
6340          FO(K)=0.
6350          JO(K)=0.
6360          HO(K)=0.
6370          IO(K)=0.
6380          BO(K)=0.
6390          CO(K)=0.
6400          DO(K)=0.
6410          FO(K)=0.
6420          JO(K)=0.
6430          HO(K)=0.
6440          IO(K)=0.
6450          BO(K)=0.
6460          CO(K)=0.
6470          DO(K)=0.
6480          FO(K)=0.
6490          JO(K)=0.
6500          HO(K)=0.
6510          IO(K)=0.
6520          BO(K)=0.
6530          CO(K)=0.
6540          DO(K)=0.
6550          FO(K)=0.
6560          JO(K)=0.
6570          HO(K)=0.
6580          IO(K)=0.
6590          BO(K)=0.
6600          CO(K)=0.
6610          DO(K)=0.
6620          FO(K)=0.
6630          JO(K)=0.
6640          HO(K)=0.
6650          IO(K)=0.
6660          BO(K)=0.
6670          CO(K)=0.
6680          DO(K)=0.
6690          FO(K)=0.
6700          JO(K)=0.
6710          HO(K)=0.
6720          IO(K)=0.
6730          BO(K)=0.
6740          CO(K)=0.
6750          DO(K)=0.
6760          FO(K)=0.
6770          JO(K)=0.
6780          HO(K)=0.
6790          IO(K)=0.
6800          BO(K)=0.
6810          CO(K)=0.
6820          DO(K)=0.
6830          FO(K)=0.
6840          JO(K)=0.
6850          HO(K)=0.
6860          IO(K)=0.
6870          BO(K)=0.
6880          CO(K)=0.
6890          DO(K)=0.
6900          FO(K)=0.
6910          JO(K)=0.
6920          HO(K)=0.
6930          IO(K)=0.
6940          BO(K)=0.
6950          CO(K)=0.
6960          DO(K)=0.
6970          FO(K)=0.
6980          JO(K)=0.
6990          HO(K)=0.
7000          IO(K)=0.
7010          BO(K)=0.
7020          CO(K)=0.
7030          DO(K)=0.
7040          FO(K)=0.
7050          JO(K)=0.
7060          HO(K)=0.
7070          IO(K)=0.
7080          BO(K)=0.
7090          CO(K)=0.
7100          DO(K)=0.
7110          FO(K)=0.
7120          JO(K)=0.
7130          HO(K)=0.
7140          IO(K)=0.
7150          BO(K)=0.
7160          CO(K)=0.
7170          DO(K)=0.
7180          FO(K)=0.
7190          JO(K)=0.
7200          HO(K)=0.
7210          IO(K)=0.
7220          BO(K)=0.
7230          CO(K)=0.
7240          DO(K)=0.
7250          FO(K)=0.
7260          JO(K)=0.
7270          HO(K)=0.
7280          IO(K)=0.
7290          BO(K)=0.
7300          CO(K)=0.
7310          DO(K)=0.
7320          FO(K)=0.
7330          JO(K)=0.
7340          HO(K)=0.
7350          IO(K)=0.
7360          BO(K)=0.
7370          CO(K)=0.
7380          DO(K)=0.
7390          FO(K)=0.
7400          JO(K)=0.
7410          HO(K)=0.
7420          IO(K)=0.
7430          BO(K)=0.
7440          CO(K)=0.
7450          DO(K)=0.
7460          FO(K)=0.
7470          JO(K)=0.
7480          HO(K)=0.
7490          IO(K)=0.
7500          BO(K)=0.
7510          CO(K)=0.
7520          DO(K)=0.
7530          FO(K)=0.
7540          JO(K)=0.
7550          HO(K)=0.
7560          IO(K)=0.
7570          BO(K)=0.
7580          CO(K)=0.
7590          DO(K)=0.
7600          FO(K)=0.
7610          JO(K)=0.
7620          HO(K)=0.
7630          IO(K)=0.
7640          BO(K)=0.
7650          CO(K)=0.
7660          DO(K)=0.
7670          FO(K)=0.
7680          JO(K)=0.
7690          HO(K)=0.
7700          IO(K)=0.
7710          BO(K)=0.
7720          CO(K)=0.
7730          DO(K)=0.
7740          FO(K)=0.
7750          JO(K)=0.
7760          HO(K)=0.
7770          IO(K)=0.
7780          BO(K)=0.
7790          CO(K)=0.
7800          DO(K)=0.
7810          FO(K)=0.
7820          JO(K)=0.
7830          HO(K)=0.
7840          IO(K)=0.
7850          BO(K)=0.
7860          CO(K)=0.
7870          DO(K)=0.
7880          FO(K)=0.
7890          JO(K)=0.
7900          HO(K)=0.
7910          IO(K)=0.
7920          BO(K)=0.
7930          CO(K)=0.
7940          DO(K)=0.
7950          FO(K)=0.
7960          JO(K)=0.
7970          HO(K)=0.
7980          IO(K)=0.
7990          BO(K)=0.
8000          CO(K)=0.
  
```

STRIP/BIEB/WERVLY

DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:34:24.

```

2690      DO 520 J=1,N
2700          K=J*M
2710          CO(K)=0.
2720          FO(K)=0.
2730      520      IO(K)=0.
2740          DO 530 I=M,1,-1
2750              K=(N-1)*M+I
2760              GO(K)=0.
2770              HO(K)=0.
2780      530      IO(K)=0.
2790      C
2800      C Vorticity on lower, left and upper boundary equal zero: coefficients
2810      C for these points at rows next to boundaries equal zero.
2820      C
2830          DO 550 I=1,M
2840              K=I+M
2850              AO(K)=0.
2860              BO(K)=0.
2870      550      CO(K)=0.
2880          DO 560 J=1,N
2890              K=(J-1)*M+2
2900              AO(K)=0.
2910              DO(K)=0.
2920      560      GO(K)=0.
2930          DO 570 I=1,M
2940              K=(N-2)*M+I
2950              GO(K)=0.
2960              HO(K)=0.
2970      570      IO(K)=0.
2980          DO 580 J=1,N
2990              K=J*M-1
3000              CO(K)=0.
3010              FO(K)=0.
3020              IO(K)=0.
3030      580      CONTINUE
3040          END
    
```

STRIP/BIEB/MSTIMP

DATE & TIME PRINTED: WEDNESDAY, JANUARY 27, 1988 @ 12:08:38.

```

4000      SUBROUTINE MSTIMP(M,N,OM,AA,BA,CA,DA,EA,FA,GA,HA,IA,JA,A,B,C,
4010      *      D,E,F,G,H,U,DE,VH,R,ALFA,EPOMME,EPOMMA,NITMAX,NITTOT)
4020      INTEGER M,N,NITMAX,NITTOT
4030      REAL OM(M*N),AA(M*N),BA(M*N),CA(M*N),DA(M*N),EA(M*N),FA(M*N),
4040      *      GA(M*N),HA(M*N),IA(M*N),JA(M*N),A(M*N),B(M*N),C(M*N),D(M*N),
4050      *      E(M*N),F(M*N),G(M*N),H(M*N),U(M*N),DE(M*N),R(M*N),ALFA,
4060      *      VH(M*N),EPOMME,EPOMMA
4070      IN LIBRARY MLIB
4080      END
    
```

81

```

6000      SUBROUTINE WERVLY(XTIMES,STOPCR,NOUTOF,EA,ER)
6010      C
6020      C WERVLY calculates a least-squares fit of a function FIE to the local
6030      C maxima of the vorticity distribution.
6040      C
6050      EXTERNAL RES
6060      INTEGER M,N,STOPCR,NPAR,NMAX
6070      REAL SOM,MAX,R(100),EA,ER,X1,X2,X3,X4,X5,D1,D2,D4,D5,F1,F2,F3,F4,
6080      *      F5,FAKS3,F2AKS3,A,B,OM(10000),X(10000),PAR(10),XTIMES,XM(100),
6090      *      YM(100),Y(10000)
6100      COMMON /XMYM/XM,YM,/FIELD/OM,/COORD/X,Y,/SIZE/M,N,/PARAM/NPAR,PAR
6110      NMAX=1
6120      DO 100 I=(M+1)/2,M
6130          MAX=0.
6140          SOM=0.
6150          DO 50 J=1,N
6160              K=(J-1)*M+I
6170              SOM=SOM+ABS(OM(K))
6180              IF (ABS(OM(K)).GE.MAX) THEN
6190                  MAX=ABS(OM(K))
6200                  KMAX=K
6210              END IF
6220      50      CONTINUE
6230      SOM=SOM/N
6240      IF (MAX.GE. XTIMES*SOM.AND.MAX.GT.1.E-20) THEN
6250          IF (KMAX.LE.2*M.OR.KMAX.GT.(N-2)*M) THEN
6260              YM(NMAX)=Y(KMAX)
6270          ELSE
6280              X1=Y(KMAX-2*M)
6290              X2=Y(KMAX-M)
6300              X3=Y(KMAX)
6310              X4=Y(KMAX+M)
6320              X5=Y(KMAX+2*M)
6330              D1=X1-X3
6340              D2=X2-X3
6350              D4=X4-X3
6360              D5=X5-X3
6370              F1=OM(KMAX-2*M)
6380              F2=OM(KMAX-M)
6390              F3=OM(KMAX)
6400              F4=OM(KMAX+M)
6410              F5=OM(KMAX+2*M)
6420              FAKS3=(F4*D2**2-F2*D4**2-F3*(D2**2-D4**2))/(D2*D4*(D2-D4))
6430              A=D1*D2*(D2**2-D1**2)
6440              B=D4*D5*(D5**2-D4**2)
6450              F2AKS3=B*(F1*D2**3-F2*D1**3)-A*(F4*D5**3-F5*D4**3)-
6460      *      F3*(B*(D2**3-D1**3)-A*(D5**3-D4**3))
    
```

A2. Programme-source

```

6470          YM(NMAX)=X3-FAKS 3*(.5*(B*(D2-D1)*(D1*D2)**2-
6480          *      A*(D5-D4)*(D4*D5)**2))/F2AKS 3
6490          END IF
6500          XM(NMAX)=X(KMAX)
6510          NMAX=NMAX+1
6520          END IF
6530          100 CONTINUE
6540          IF(NMAX.GE.2*NPAR)THEN
6550            NSTOP=STOPCR
6560          C
6570          C MARQUA calculates the set of parameters that minimises the residual
6580          C vector res, calculated in the function RES.
6590          C
6600          CALL MARQUA(NMAX,R,NPAR,PAR,RES,NSTOP,EA,ER,100,NOUTQF,86)
6610          END IF
6620          END
6630          C *****
6640          C
6650          SUBROUTINE RES(N,PAR,NMAX,R)
6660          C
6670          C RES calculates the residu between the curve-fit and the measured
6680          C values for the local maxima in the vorticity-distribution.
6690          C
6700          INTEGER NMAX,I,N
6710          REAL R(NMAX),PAR(N),XM(100),YM(100),F
6720          COMMON /XMYM/XM,YM,/PARAM/NPAR,/DIAM/D
6730          DO 100 I=1,NMAX
6740            IF(ABS(XM(I)).LT.1.E-2)THEN
6750              SIGMA=100.
6760            ELSE
6770              SIGMA=1./ABS(XM(I))
6780            END IF
6790            F=FIT(XM(I),PAR,D)
6800            R(I)=(F-YM(I))/SIGMA
6810          100 CONTINUE
6820          END
STRIP/BIEB/XYTOOM
DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:35:57.

5000          SUBROUTINE XYTOOM
5005          LOGICAL LFIT
5010          INTEGER NC,DELER1,DELER2,NPENAL,NITPEN
5020          REAL A,AOMRO(100),OMT,OMA(10000),HULP(100),DELT,A1,A2,DEY,
5030          *      XN(10000),YN(10000),DELTA(100),ALF1(100),ALF2(100),HIFX(100)
5040          SINCLUDEF STRIP/BIEB/COMMON
5050          DEXI=(XMAX-XMIN)/(M-1)
5060          DEET=YMAX/(N-1)
5070          C
5080          C This subroutine constructs a new grid, in which the grid points
5090          C are distributed proportional to the distribution of vorticity.
5100          C The two rows of gridpoints at the edges of the domain are held fixed
5110          C
5120          C for fitting of new x-coordinates first summation of vorticity
5130          C per column.(X=constant)
5140          C
5150          MH=(M+1)/2
5160          DO 20 I=1,M
5170            AOMRO(I)=0.
5180            DO 10 J=1,N-1
5190              K=(J-1)*M+I
5200              AOMRO(I)=AOMRO(I)+OM(K)*OM(K)
5210          10 CONTINUE
5220            AOMRO(I)=SQRT(AOMRO(I))
5230          20 CONTINUE
5240          C
5250          C creating new AOMRO(I) in such a way that AOMRO only increases
5260          C with increasing I.
5270          C
5280          DO 30 I=2,M-1
5290            DELTA(I)=AOMRO(I)-AOMRO(I-1)
5300            IF(DELTA(I).LT.0.)DELTA(I)=-DELTA(I)
5310          30 CONTINUE
5320          DO 40 I=2,M-1
5330            AOMRO(I)=AOMRO(I-1)+DELTA(I)
5340          40 NITPEN=0
5350          41 CONTINUE
5360            IF(NITPEN.GT.NITPMX)GO TO 49
5370          C
5380          C Scaling of the values of AOMRO with A1 and A2
5390          C
5400          A1=(-XMIN-DEXI)/(AOMRO(MH)-AOMRO(2))
5410          A2=(XMAX-DEXI)/(AOMRO(M-1)-AOMRO(MH))
5420          DO 42 I=3,MH-1
5430            AOMRO(I)=DEXI+A1*(AOMRO(I)-AOMRO(2))
5440          42 DO 44 I=MH+1,M-2
5450            AOMRO(I)=-XMIN+A2*(AOMRO(I)-AOMRO(MH))
5460          AOMRO(MH)=-XMIN
5470          AOMRO(1)=0.
5480          AOMRO(2)=DEXI
5490          AOMRO(M-1)=XMAX-XMIN-DEXI
5500          AOMRO(M)=XMAX-XMIN
5510          C
5520          C Checking the differences between consequent AOMRO, minimum value
5530          C is XIXMIN.
5540          C
5550          NPENAL=0
5560          DO 47 I=3,M-1

```

```

5570      DELTA(I)=AOMRO(I)-AOMRO(I-1)
5580      DEX=X(I)-X(I-1)
5590      IF(DELTA(I)/DEX.LT.XIXMIN)THEN
5592        IF(DELTA(I).LT.1.E-20)THEN
5594          DELTA(I)=DEX*XIXMIN
5596        ELSE
5600          DELTA(I)=DEX*DEX*XIXMIN*XIXMIN/DELTA(I)
5602        END IF
5610      NPENAL=NPENAL+1
5620      END IF
5630      IF(DELTA(I)/DEX.GT.XIXMAX)THEN
5640        DELTA(I)=DEX*DEX*XIXMAX*XIXMAX/DELTA(I)
5650        NPENAL=NPENAL+1
5660      END IF
5670      47 CONTINUE
5680      DO 48 I=3,M-1
5690      48  AOMRO(I)=AOMRO(I-1)+DELTA(I)
5700      IF(NPENAL.GT.0)NITPEN=NITPEN+1
5710      IF(NPENAL.GT.0)GO TO 41
5720      49 CONTINUE
5730      C
5740      C A1 and A2 are the coupling factors between the vorticity in a row and
5750      C the value of KSI (the vertical grid lines are spread according to the
5760      C sum of vorticity in a row)
5770      C
5780      A1=(-XMIN-DEXT)/(AOMRO(MH)-AOMRO(2))
5790      A2=(XMAX-DEXI)/(AOMRO(M-1)-AOMRO(MH))
5800      C
5810      C calculation of the values of the new KSI belonging to the old grid
5820      C points
5830      C
5840      DO 50 I=3,MH-1
5850        AOMRO(I)=XMIN+DEXT+A1*(AOMRO(I)-AOMRO(2))
5860      50 CONTINUE
5870      DO 60 I=MH+1,M-2
5880        AOMRO(I)=A2*(AOMRO(I)-AOMRO(MH))
5890      60 CONTINUE
5900      AOMRO(1)=XMIN
5910      AOMRO(2)=XMIN+DEXT
5920      AOMRO(M-1)=XMAX-DEXI
5930      AOMRO(M)=XMAX
5940      AOMRO(MH)=0.
5950      C
5960      C Now determination of X-coordinates of new gridpoints,using linear
5970      C interpolation of the new KSI-coordinates
5980      C The new X-coordinates are put in the array XN
5990      C
6000      XN(1)=XMIN
6010      XN(2)=XMIN/2.
6020      XN(MH)=0.
6030      XN(M-1)=XMAX/2.
6040      XN(M)=XMAX
6050      DO 120 I=3,MH-1
6060      DO 120 J=3,MH
6070        IF(AOMRO(J-1).LE.KSI(I).AND.AOMRO(J).GT.KSI(I))THEN
6080          XN(I)=X(J-1)+(KSI(I)-AOMRO(J-1))*(X(J)-X(J-1))/(AOMRO(J)-
6090          * AOMRO(J-1))
6100        END IF
6110      120 CONTINUE
6120      DO 130 I=MH+1,M-2
6130      DO 130 J=MH+1,M-1
6140        IF(AOMRO(J-1).LE.KSI(I).AND.AOMRO(J).GT.KSI(I))THEN
6150          XN(I)=X(J-1)+(KSI(I)-AOMRO(J-1))*(X(J)-X(J-1))/(AOMRO(J)-
6160          * AOMRO(J-1))
6170        END IF
6180      130 CONTINUE
6190      DO 140 J=2,N
6200      DO 140 I=1,M
6210        K=(J-1)*M+I
6220      140  XN(I)=XN(I)
6230      C
6240      C Changing old and new X-coordinates, Y-coordinates stay the same
6250      C
6260      DO 145 J=1,N
6270      DO 145 I=1,M
6280        K=(J-1)*M+I
6290        XO(K)=X(K)
6300      145  X(K)=XN(K)
6310      C
6320      C Now the the values of OM are interpolated for the new X-coordinates
6330      C
6340      C
6350      CALL SHFTOX
6360      C
6370      C For the gridpoints with altered X-coordinate,
6380      C the same procedure is followed for the coupling between vorticity
6390      C and new Y-coordinates.
6400      C
6410      DO 65 I=1,M
6420      65  YM(I)=FIE(X(I),PAR,D)
6430      DO 80 I=1,M
6440        OMA(I)=0.
6450      DO 70 J=2,N-2
6460        K=(J-1)*M+I
6470        DELT=ABS(OM(K))-ABS(OM(K-M))
6480        IF(DELT.LT.0.)DELT=-DELT
6490      70  OMA(K)=OMA(K-M)+DELT+ETYMIN*.5*(Y(K+M)-Y(K-M))
6500        K=(N-2)*M+I
6510        OMA(K)=ABS(OM(K))-ABS(OM(K-M))
6520        IF(OMA(K).LT.0.)OMA(K)=-OMA(K)
6530        OMA(K)=OMA(K-M)+OMA(K)+ETYMIN*(Y(K)-Y(K-M))
6540      80 CONTINUE

```

```

6550 C
6560 C the values of OMA in NCPL neighbouring columns are coupled and put
6570 C in the array H1
6580 C
6590 DO 85 I=1,M
6600 85 H1(I)=0.
6610 NITPEN=0
6615 LFIT=.TRUE.
6620 DO 90 J=2,N-1
6630 DO 90 I=1,M
6640 K=(J-1)*M+I
6650 H1(K)=OMA(K)
6660 DO 86 NC=1,NCPL
6670 IF(I.GT.NC)H1(K)=H1(K)+OMA(K-NC)/((NC+1)*(NC+1))
6680 IF(I.LT.M+1-NC)H1(K)=H1(K)+OMA(K+NC)/((NC+1)*(NC+1))
6690 86 CONTINUE
6700 90 CONTINUE
6710 IF(NITPEN.GT.NITPMY)LFIT=.FALSE.
6720 C
6730 C Determination of the coupling coefficients per column ALF1(I) and
6740 C ALF2(I)
6750 C
6760 DO 105 I=1,M
6770 DO 105 J=2,N-1
6780 K=(J-1)*M+I
6790 IF(Y(K-M).LE.YM(I).AND.Y(K).GT.YM(I))THEN
6800 H1FX(I)=H1(K-M)+(YM(I)-Y(K-M))*(H1(K)-H1(K-M))/(Y(K)-Y(K-M))
6810 END IF
6820 105 CONTINUE
6830 DO 100 I=1,M
6840 ALF1(I)=((NHETA-1)*DEET)/H1FX(I)
6850 100 ALF2(I)=((N-NHETA-1)*DEET)/(H1((N-2)*M+I)-H1FX(I))
6860 C
6870 C Calculation of the new ETA-coordinates of the old grid points.
6880 C
6890 DO 109 I=1,M
6900 H1(I)=0.
6910 109 H1((N-1)*M+I)=YMAX
6920 NPENAL=0
6930 DO 110 J=2,N-1
6940 DO 110 I=1,M
6950 K=(J-1)*M+I
6960 IF(H1(K).LE.H1FX(I))THEN
6970 H1(K)=ALF1(I)*H1(K)
6980 ELSE
6990 H1(K)=(NHETA-1)*DEET+ALF2(I)*(H1(K)-H1FX(I))
7000 END IF
7010 IF(.NOT.LFIT)GO TO 110
7015 H2(K)=H1(K)-H1(K-M)
7020 DEY=Y(K)-Y(K-M)

```

```

7040 IF(H2(K)/DEY.GT.ETymax)THEN
7050 H2(K)=DEY*DEY*ETymax*ETymax/H2(K)
7060 NPENAL=NPENAL+1
7070 END IF
7080 IF(H2(K)/DEY.LT.ETymin)THEN
7082 IF(H2(K).LT.I.E-20)THEN
7084 H2(K)=DEY*ETymin
7086 ELSE
7090 H2(K)=DEY*DEY*ETymin*ETymin/H2(K)
7092 END IF
7100 NPENAL=NPENAL+1
7110 END IF
7120 110 CONTINUE
7125 IF(.NOT.LFIT)GO TO 230
7130 IF(NPENAL.GT.0)THEN
7140 DO 220 J=2,N-1
7150 DO 220 I=1,M
7160 K=(J-1)*M+I
7170 220 H1(K)=H1(K-M)+H2(K)
7180 NITPEN=NITPEN+1
7190 END IF
7200 IF(NPENAL.GT.0)GO TO 90
7210 230 CONTINUE
7220 C
7230 C Determination of the Y-coordinates of the new gridpoints, using
7240 C linear interpolation of the ETA-coordinates of the new points
7250 C The new Y-coordinates are put in the array YN
7260 C
7265 IF(NHETA.GT.N-2)NHETA=N-2
7270 NH=(N+1)/2
7271 C OPEN(22,FILE='WEERBERICHT.',STATUS='NEW')
7272 C DO 149 J=1,N
7273 C 149 WRITE(22,*)J,H1((J-1)*M+I)
7280 DO 160 I=1,M
7290 YN(I)=0.
7300 YN((N-2)*M+I)=YMAX/2.
7310 YN((N-1)*M+I)=YMAX
7320 YN((NHETA-1)*M+I)=YM(I)
7330 DO 150 J=2,NHETA-1
7340 KJ=(J-1)*M+I
7350 DO 150 JI=2,N-1
7360 KI=(JI-1)*M+I
7370 IF(H1(KI-M).LE.ETA(KJ).AND.H1(KI).GT.ETA(KJ))THEN
7380 YN(KJ)=Y(KI-M)+(ETA(KJ)-H1(KI-M))*(Y(KI)-Y(KI-M))/
7390 * (H1(KI)-H1(KI-M))
7400 END IF
7410 150 CONTINUE
7420 DO 155 J=NHETA+1,N-2
7430 KJ=(J-1)*M+I
7440 DO 155 JI=2,N-1

```



```

7450      KI=(JI-1)*M+I
7460      IF(HI(KI-M).LE.ETA(KJ).AND.HI(KI).GT.ETA(KJ))THEN
7470          YN(KJ)=Y(KI-M)+(ETA(KJ)-HI(KI-M))*(Y(KI)-Y(KI-M))/
7480          *      (HI(KI)-HI(KI-M))
7490      END IF
7500      155 CONTINUE
7510      160 CONTINUE
7512      C DO 161 J=1,N
7513      C 161 WRITE(22,*)J,YN((J-1)*M+MH)
7514      C CLOSE(22,STATUS='CRUNCH')
7520      C
7530      C Changing old and new Y-coordinates.
7540      C
7550          DO 200 J=1,N
7560              DO 190 I=1,M
7570                  K=(J-1)*M+I
7580                  YQ(K)=Y(K)
7590                  Y(K)=YN(K)
7600      190 CONTINUE
7610      200 CONTINUE
7620      C
7630      C The values of the vorticity in the new gridpoints are interpolated
7640      C
7650          CALL SHFTOY
7660      C
7670      C The partial derivatives of KSI and ETA with respect to X and Y are
7680      C calculated in the subroutine DERIVA
7690      C
7700          CALL DERIVA
7701          IF(DIAGNO)THEN
7702              OPEN(11,TITLE='DIAGNO/XYTOOM.',STATUS='NEW')
7703              DO 999 J=0,N-1
7704          999 WRITE(11,*)EX(J*M+MH+1),EY(J*M+MH+1),EXX(J*M+MH+1),
7705          *      EYY(J*M+MH+1),EXY(J*M+MH+1)
7706          CLOSE(11,STATUS='CRUNCH')
7707      END IF
7710      1000 CONTINUE
7720      END
7730      C
7740      C*****
7750      C
7760          SUBROUTINE SHFTOX
7770          INTEGER NI,NK,NJ,K1,K2,K3,K4
7775          REAL D1,D2,YP,YQ,OP,OQ,XP1,YP1,XQ3,YQ3,X21,Y21,X43,Y43,D2P,D2Q
7780      SINCLUDE 'STRIP/BIEB/COMMON'
7790      C
7800      C This subroutine adapts the values of OM in the new gridpoints
7810      C X,Y to the values of OM in the old gridpoints XO,Y
7820      C notice: the line connecting points with equal X-index (i) is a
7830      C straight line. Only the x-coordinates have changed.

```

```

7840      C Furthermore the values of OM are zero on the boundaries.
7850      C HI is a help-array
7860      C First setting value for HI zero for each gridpoint.
7870      C
7880          DO 100 K=1,M*N
7890      100 HI(K)=0.
7970          DO 200 I=2,M-1
7980          DO 200 NI=2,M
8080          IF(XO(NI-1).LE.X(I).AND.XO(NI).GT.X(I))THEN
8081              DO 150 J=2,N-1
8082                  K=(J-1)*M+I
8083                  DO 150 NJ=2,N
8084                      NK=(NJ-1)*M+NI
8090                      D2=(X(K)-XO(NK-1))/(XO(NK)-XO(NK-1))
8100                      YP=Y(NK-M-1)+D2*(Y(NK-M)-Y(NK-M-1))
8110                      YQ=Y(NK-1)+D2*(Y(NK)-Y(NK-1))
8120                      IF(YP.LE.Y(K).AND.YQ.GT.Y(K))THEN
8180                          K1=NK-M-1
8190                          K2=NK-M
8200                          K3=NK-1
8210                          K4=NK
8211                          XP1=X(K)-XO(K1)
8212                          YP1=YP-Y(K1)
8213                          XQ3=X(K)-XO(K3)
8214                          YQ3=YQ-Y(K3)
8215                          X21=XO(K2)-XO(K1)
8216                          Y21=Y(K2)-Y(K1)
8217                          X43=XO(K4)-XO(K3)
8218                          Y43=Y(K4)-Y(K3)
8219                          D2P=SQR T((XP1*XP1+YP1*YP1)/(X21*X21+Y21*Y21))
8220                          D2Q=SQR T((XQ3*XQ3+YQ3*YQ3)/(X43*X43+Y43*Y43))
8229                          OP=OM(K1)+D2P*(OM(K2)-OM(K1))
8230                          OQ=OM(K3)+D2Q*(OM(K4)-OM(K3))
8240                          D1=(Y(K)-YP)/(YQ-YP)
8250                          H1(K)=OP+D1*(OQ-OP)
8251                      END IF
8252          150 CONTINUE
8253          END IF
8260      200 CONTINUE
8270          DO 300 J=2,N-1
8280          DO 300 I=2,M-1
8290              K=(J-1)*M+I
8300          300 OM(K)=HI(K)
8310          END
8314      C
8315      C*****
8316      C
8370          SUBROUTINE SHFTOY
8370          INTEGER NJ,NK
8377          REAL D1
8375

```

```

8780  $INCLUDE`STRIP/BIEB/COMMON`
8790  C
8800  C This subroutine adapts the values of OM in the new gridpoints
8810  C X,Y to the values of OM in the old gridpoints X,YO
8820  C notice: the line connecting points with equal X-index (i) is a
8830  C straight line. Only the y-coordinates have changed.
8840  C Furthermore the values of OM are zero on the boundaries.
8850  C HI is a help-array
8860  C First setting value for HI zero for each gridpoint.
8870  C
8880      DO 100 K=1,M*N
8890      i00  HI(K)=0.
8970      DO 200 I=2,M-1
8980          DO 200 J=2,N-1
8990              K=(J-1)*M+I
9000              DO 200 NJ=2,N
9010                  NK=(NJ-1)*M+I
9080                  IF(YO(NK-M).LE.Y(K).AND.YO(NK).GT.Y(K))THEN
9240                      D1=(Y(K)-YO(NK-M))/(YO(NK)-YO(NK-M))
9250                      HI(K)=OM(NK-M)+D1*(OM(NK)-OM(NK-M))
9251                  END IF
9260      200  CONTINUE
9270          DO 300 J=2,N-1
9280              DO 300 I=2,M-1
9290                  K=(J-1)*M+I
9300      300  OM(K)=HI(K)
9310      END

```

STRIP/BIEB/FIE

DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:48:03.

```

7000      REAL FUNCTION FIE(X,PAR,D)
7010  C this function provides the y-value of maximum vorticity for a given x
7020  C
7030      REAL X,PAR(10),D,X2,PI,PI2,SOM
7040      INTEGER I,NPAR
7050      COMMON /PARAM/NPAR
7060      PI=4.*ATAN(1.)
7070      X2=X*X
7080      PI2=PI*PI/4.
7090      SOM=0.
7100      DO i0 I=1,NPAR
7110      10  SOM=SOM+PAR(I)*X/(((2.*I-1)**2)*PI2+X2)
7120      FIE=D*(1.-2.*SOM)
7130      END
7140  C*****
7150  C

```

STRIP/BIEB/KOPOIS

DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:38:42.

```

3000      SUBROUTINE KOPOIS
3010  C
3020  C determination of the coefficients for the discretised, transformed
3025  C Poisson-PSI equation.
3030  C
3040      REAL DEXI,DEET
3050  $INCLUDE`STRIP/BIEB/COMMON`
3060      DEXI=(XMAX-XMIN)/(M-1)
3070      DEET=YMAX/(N-1)
3080      DO 100 I=1,L
3090          AP(I)=ZPT6(I)/(4*DEXI*DEET)
3100          BP(I)=-ZPT3(I)/(2*DEET)+ZPT5(I)/(DEET**2)
3110          CP(I)=-AP(I)
3120          DP(I)=-ZPT2(I)/(2*DEXI)+ZPT4(I)/(DEXI**2)
3130          EP(I)=-2*(ZPT4(I)/(DEXI**2)+ZPT5(I)/(DEET**2))
3140          FP(I)=ZPT2(I)/(2*DEXI)+ZPT4(I)/(DEXI**2)
3150          GP(I)=-AP(I)
3160          HP(I)=ZPT3(I)/(2*DEET)+ZPT5(I)/(DEET**2)
3170          IP(I)=AP(I)
3180          JP(I)=ZPT1(I)
3190      100  CONTINUE
3200      END

```

STRIP/BIEB/TRANIP

DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:37:20.

```

2500      SUBROUTINE TRANIP
2510  C
2520  C transformation of the coefficients of the Poisson-psi equation
2525  C into the computational plane.
2530  C
2540  $INCLUDE`STRIP/BIEB/COMMON`
2550      DO 100 I=1,L
2560          ZPT1(I)=ZP1(I)
2570          ZPT2(I)=KS IX(I)
2580          ZPT3(I)=EXX(I)+EYY(I)
2590          ZPT4(I)=KS IX(I)**2
2600          ZPT5(I)=EX(I)**2+EY(I)**2
2610          ZPT6(I)=2.*KS IX(I)*EX(I)
2620      100  CONTINUE
2630      END

```

STRIP/BIEB/RVWPSI

DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:39:58.

```

1000      SUBROUTINE RVWPSI
1010      C
1020      C this subroutine calculates the effect of the boundary conditions on
1030      C the coefficients in the differential scheme for the transformed
1040      C Poisson-PSI equation.
1050      C
1060      REAL Q(10000),DEXI,DEET,A,Z(100),Z1,Z2
1070      $INCLUDE 'STRIP/BIEB/COMMON'
1080      DEXI=(XMAX-XMIN)/(M-1)
1090      DEET=YMAX/(N-1)
1100      C
1110      C boundary conditions for the lower boundary y=0: mirror
1120      C
1130      DO 100 I=1,M
1140      Z1=(EY(I)/DEET)**2
1150      Z2=EYY(I)/(2.*DEET)
1160      Z(I)=2.*Z1/(Z1-Z2)
1170      100 Q(I)=(Z1+Z2)/(Z2-Z1)
1180      DO 200 I=1,M
1190      IF(I.GT.1)GP(I)=GP(I)+AP(I)*Q(I-1)
1200      HP(I)=HP(I)+BP(I)*Q(I)
1210      EP(I)=EP(I)+Z(I)*BP(I)
1220      IF(I.LT.M)IP(I)=IP(I)+Q(I+1)*CP(I)
1230      AP(I)=0.
1240      BP(I)=0.
1250      CP(I)=0.
1260      200 CONTINUE
1270      C
1280      C other boundaries: zero-valued velocities, streamfunction at boundary
1290      C and next to boundary equal zero.
1300      C
1310      DO 300 J=1,N
1320      DO 300 I=1,M
1330      K=(J-1)*M+I
1340      IF(.NOT.(J.GT.0.AND.J.LT.N-1.AND.I.GT.2.AND.I.LT.M-1))THEN
1350      AP(K)=0.
1360      BP(K)=0.
1370      CP(K)=0.
1380      DP(K)=0.
1390      EP(K)=1.
1400      FP(K)=0.
1410      GP(K)=0.
1420      HP(K)=0.
1430      IP(K)=0.
1440      JP(K)=0.
1450      END IF
1460      300 CONTINUE
1470      END

```

STRIP/BIEB/TRVELO

DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:41:30.

```

6000      SUBROUTINE TRVELO
6010      C
6020      C TRVELO determines the velocities in X- and Y-direction for each grid
6030      C point. The truncation error of the differential schemes used is of
6035      C order two.
6040      C
6050      REAL A,B,DEXI,DEET
6060      $INCLUDE 'STRIP/BIEB/COMMON'
6070      DEXI=(XMAX-XMIN)/(M-1)
6080      DEET=YMAX/(N-1)
6090      DO 100 I=1,M
6100      DO 100 J=1,N
6110      K=(J-1)*M+I
6120      IF(I.EQ.1)THEN
6130      A=(PSI(K+1)-PSI(K))/DEXI
6140      ELSE IF(I.EQ.M)THEN
6150      A=(-PSI(K-1)+PSI(K))/DEXI
6160      ELSE
6170      A=(PSI(K+1)-PSI(K-1))/(2*DEXI)
6180      END IF
6190      IF(J.EQ.1)THEN
6200      B=(PSI(K+M)-PSI(K))/DEET
6210      ELSE IF(J.EQ.N)THEN
6220      B=(-PSI(K-M)+PSI(K))/DEET
6230      ELSE
6240      B=(PSI(K+M)-PSI(K-M))/(2*DEET)
6250      END IF
6260      U(K)=EY(K)*B
6270      V(K)=-KSIX(K)*A-EX(K)*B
6280      100 CONTINUE
6290      END

```

STRIP/BIEB/REGGRI

DATE & TIME PRINTED: WEDNESDAY, MARCH 9, 1988 @ 20:42:49.

```

100      SUBROUTINE REGGRI
110      C
120      C setting up an orthogonal grid with the values of y at x=0 as a base.
130      C
140      INTEGER MH
150      $INCLUDE 'STRIP/BIEB/COMMON'
160      MH=(M+1)/2
170      DO 10 I=1,M*N
180      YO(I)=Y(I)
190      10 XO(I)=X(I)
200      DO 100 J=1,N
210      DO 70 I=1,MH-1

```

```

220      K=(J-1)*M+I
230      Y(K)=Y((J-1)*M+MH)
240      DO 100 I=MH+1,M
250      K=(J-1)*M+I
260      100 Y(K)=Y((J-1)*M+MH)
270      END
STRIP/BIER/COMMON
DATE & TIME PRINTED: MONDAY, FEBRUARY 29, 1988 @ 13:27:53.

10      C this block contains the variables that are used in all the sub-
11      C routines.
100      INTEGER M,N,NT,STOPCR,I,L,J,K,STOPCX,NPAR,NITO,NITP,NCPL,
104      * NITPMX,NITPMY,NOUTQF,NHETA
108      REAL XMIN,XMAX,YMAX,D,TBEGIN,TEIND,EPPOME,EPOMMA,EPIME,EPIMA,
112      * EA,ER,PAR(10),X(10000),Y(10000),U(10000),V(10000),OM(10000),
116      * XTIMES,OMO(10000),PSI(10000),KSI(10000),ETA(10000),AO(10000),
120      * BO(10000),CO(10000),DO(10000),EO(10000),FO(10000),GO(10000),
124      * HO(10000),IO(10000),JO(10000),AP(10000),BP(10000),CP(10000),
128      * DP(10000),EP(10000),FP(10000),GP(10000),HP(10000),IP(10000),
132      * JP(10000),
136      * ZO2(10000), ZO1(10000),ZO3(10000),ZO4(10000),ZO5(10000),
140      * ZOT1(10000),ZOT2(10000),ZOT3(10000),ZOT4(10000),ZOT5(10000),
144      * ZOT6(10000),XO(10000),YO(10000),
148      * H1(10000),H2(10000),H3(10000),H4(10000),H5(10000),H6(10000),
152      * H7(10000),H8(10000),H9(10000),H10(10000),H11(10000),H12(10000),
156      * F(10000),ZP1(10000),
160      * ZPT1(10000),ZPT2(10000),ZPT3(10000),ZPT4(10000),ZPT5(10000),
164      *ZPT6(10000),EX(10000),EXX(10000),EY(10000),EYY(10000),EXY(10000),
168      *KS IX(10000),KS IXX(10000),
172      * ALFAO,ALFAP,XM(100),YM(100),WIDTH,RHOO,NU,DT,DAKS,XIXMIN,
176      * XIXMAX,ETYMIN,ETymax,PRO
180      COMMON /CONF IG/XMIN,XMAX,XIXMIN,XIXMAX,ETYMIN,ETymax,NCPL,
184      * NITPMX,NITPMY,NHETA
188      COMMON /TIME/DT,/CPRO/PRO
192      COMMON /DIAM/D,WIDTH,DAKS
196      COMMON /XMYM/XM,YM
200      COMMON /DERIV/KS IX,KS IXX,EX,EXX,EY,EYY,EXY
204      COMMON /COORD/X,Y
208      COMMON /SIZE/M,N,L,RHOO,NU
212      COMMON /SPEED/U,V
216      COMMON /OLDOM/OMO
220      COMMON /STRMFU/PSI
224      COMMON /COOM/ZO1,ZO2,ZO3,ZO4,ZO5
228      COMMON /COPSI/ZP1
232      COMMON /FIELD/OM
236      COMMON /TRCOORD/KS I,ETA
240      COMMON /CODSOM/AO,BO,CO,DO,EO,FO,GO,HO,IO,JO
244      COMMON /CODSPS/AP,BP,CP,DP,EP,FP,GP,HP,IP,JP
248      COMMON /TRCOOM/ZOT1,ZOT2,ZOT3,ZOT4,ZOT5,ZOT6
252      COMMON /TRCOPS/ZPT1,ZPT2,ZPT3,ZPT4,ZPT5,ZPT6

```

88

```

256      COMMON /CALPAR/TBEGIN,TEIND,NT,EPOMME,EPOMMA,EPIME,EPIMA,
260      * XTIMES,STOPCR,EA,ER,ALFAO,ALFAP
264      COMMON /ITER/NITO,NITP
268      COMMON /OLDCOR/XO,YO
272      COMMON /PARAM/NPAR,PAR
276      SUBROUTINE MSTIMP(M,N,OM,AA,BA,CA,DA,EA,FA,GA,HA,IA,JA,A,B,C,
280      * D,E,F,G,H,U,DE,VH,R,ALFA,EPSMEA,EPSTMAX,NITMAX,NITTOT)
284      C
288      C THIS SUBROUTINE GIVES A SOLUTION TO A NINE POINT MOLECULE DIFFERENCE
292      C SCHEME BY APPLICATION OF THE MODIFIED STRONGLY IMPLICIT PROCEDURE,
296      C AS DESCRIBED IN : SCHNEIDER AND ZEDAN, NUMERICAL HEAT TRANSFER,
300      C VOL. 4, PP.1-19, 1981.
304      C
308      INTEGER M,N,I,J,K,L,NIT,NITMAX
312      REAL OM(M*N),AA(M*N),BA(M*N),CA(M*N),DA(M*N),EA(M*N),FA(M*N),
316      * GA(M*N),HA(M*N),IA(M*N),JA(M*N),A(M*N),B(M*N),C(M*N),D(M*N),
320      * E(M*N),F(M*N),G(M*N),H(M*N),U(M*N),DE(M*N),R(M*N),ALFA,
324      * VH(M*N),FI1,FI2,FI3,FI4,EPSMEA,EPSTMAX
328      REAL NORM,MAXN,DEK2
332      L=M*N
336      NIT=0
340      C
344      C CALCULATION OF THE COEFFICIENTS OF THE L- AND U-MATRICES
348      C
352      DO 50 J=1,N
356      DO 50 I=1,M
360      K=(J-1)*M+I
364      IF(I.EQ.1.OR.J.EQ.1)THEN
368      A(K)=0.
372      ELSE
376      A(K)=AA(K)
380      END IF
384      IF(J.EQ.1)THEN
388      B(K)=0.
392      ELSE
396      B(K)=BA(K)
400      IF(I.GT.1)B(K)=B(K)-A(K)*F(K-M-1)
404      IF(I.NE.M)B(K)=B(K)-ALFA*CA(K)*F(K-M+1)
408      B(K)=B(K)/(1.-ALFA*F(K-M)*F(K-M+1))
412      END IF
416      IF(I.EQ.M.OR.J.EQ.1)THEN
420      C(K)=0.
424      ELSE
428      C(K)=CA(K)-B(K)*F(K-M)
432      END IF
436      IF(I.EQ.1)THEN
440      D(K)=0.
444      ELSE
448      D(K)=DA(K)
452      IF(J.GT.1)D(K)=D(K)-A(K)*H(K-M-1)-B(K)*G(K-M)-
456      * 2.*ALFA*AK)*G(K-M-1)

```

```

4670      D(K)=D(K)/(1.+2.*ALFA*G(K-1))
4680      END IF
4690      IF(I.EQ.M.OR.J.EQ.1)THEN
4700        FI1=0.
4710      ELSE
4720        FI1=C(K)*F(K-M+1)
4730      END IF
4740      IF(I.EQ.1.OR.J.EQ.1)THEN
4750        FI2=0.
4760      ELSE
4770        FI2=A(K)*G(K-M-1)
4780      END IF
4790      IF(I.EQ.M.OR.J.EQ.1)THEN
4800        FI3=0.
4810      ELSE
4820        FI3=C(K)*U(K-M+1)
4830      END IF
4840      IF(I.EQ.1)THEN
4850        FI4=0.
4860      ELSE
4870        FI4=D(K)*G(K-1)
4880      END IF
4890      F(K)=EA(K)+ALFA*(FI1+FI2+FI3+2.*FI4)
4900      IF (I.GT.1.AND.J.GT.1)E(K)=E(K)-A(K)*U(K-M-1)
4910      IF(J.GT.1)E(K)=E(K)-B(K)*H(K-M)
4920      IF(I.LT.M.AND.J.GT.1)E(K)=E(K)-C(K)*G(K-M+1)
4930      IF(I.GT.1)E(K)=E(K)-D(K)*F(K-1)
4940      IF(I.EQ.M)THEN
4950        F(K)=0.
4960      ELSE
4970        F(K)=FA(K)-2.*ALFA*(FI1+FI3)
4980        IF(J.GT.1)F(K)=F(K)-B(K)*U(K-M)-C(K)*H(K-M+1)
4990      END IF
5000      F(K)=F(K)/E(K)
5010      IF(I.EQ.1.OR.J.EQ.N)THEN
5020        G(K)=0.
5030      ELSE
5040        G(K)=GA(K)
5050        IF(I.GT.1)G(K)=G(K)-D(K)*H(K-1)
5060        G(K)=G(K)/E(K)
5070      END IF
5080      IF(J.EQ.N)THEN
5090        H(K)=0.
5100      ELSE
5110        H(K)=HA(K)-ALFA*FI4
5120        IF(I.GT.1)H(K)=H(K)-D(K)*U(K-1)
5130        H(K)=H(K)/E(K)
5140      END IF
5150      IF(J.EQ.N.OR.I.EQ.M)THEN
5160        U(K)=0.
5170      ELSE
5180        U(K)=IA(K)/E(K)
5190      END IF

```

```

5200      50 CONTINUE
5210      C
5220      100 CONTINUE
5230      C BEGINNING OF THE LOOP
5240      C
5250        NIT=NIT+1
5260      C
5270      C CALCULATION OF THE RESIDUAL VECTOR Rj=JAj-TkjOMk
5280      C
5290        DO 200 K=1,M*N
5300          R(K)=JA(K)-EA(K)*OM(K)
5310          IF(K.GT.1) R(K)=R(K)-DA(K)*OM(K-1)
5320          IF(K.GT.M+1) R(K)=R(K)-AA(K)*OM(K-M-1)
5330          IF(K.LT.L-M+1)R(K)=R(K)-GA(K)*OM(K+M-1)
5340          IF(K.GT.M) R(K)=R(K)-BA(K)*OM(K-M)
5350          IF(K.GT.M-1) R(K)=R(K)-CA(K)*OM(K-M+1)
5360          IF(K.LT.L-1) R(K)=R(K)-FA(K)*OM(K+1)
5370          IF(K.LT.L-M-1)R(K)=R(K)-IA(K)*OM(K+M+1)
5380          IF(K.LT.L-M) R(K)=R(K)-HA(K)*OM(K+M)
5390      200 CONTINUE
5400      C
5410      C CALCULATION OF VH, THE RESULT OF LVH=R
5420      C
5430        VH(1)=R(1)/E(1)
5440        DO 300 K=2,L
5450          VH(K)=R(K)-D(K)*VH(K-1)
5460          IF(K.GT.M-1)VH(K)=VH(K)-C(K)*VH(K-M+1)
5470          IF(K.GT.M) VH(K)=VH(K)-B(K)*VH(K-M)
5480          IF(K.GT.M+1)VH(K)=VH(K)-A(K)*VH(K-M-1)
5490          VH(K)=VH(K)/E(K)
5500      300 CONTINUE.
5510      C
5520      C CALCULATION OF THE DIFFERENCE VECTOR DE BY U.DE=VH, AND OF THE
5530      C NEW SOLUTION VECTOR OM=OM+DE
5540      C
5550        NORM=0.
5560        MAXN=0.
5570        DE(L)=VH(L)
5580        OM(L)=OM(L)+DE(L)
5590        NORM=ABS(DE(L))
5600        MAXN=NORM
5610        DO 400 K=L-1,1,-1
5620          DE(K)=VH(K)-F(K)*DE(K+1)
5630          IF(K.LT.L-M+1)DE(K)=DE(K)-G(K)*DE(K+M-1)
5640          IF(K.LT.L-M) DE(K)=DE(K)-H(K)*DE(K+M)
5650          IF(K.LT.L-M-1)DE(K)=DE(K)-U(K)*DE(K+M+1)
5660          OM(K)=OM(K)+DE(K)
5670          DEK2=ABS(DE(K))
5680          IF(DEK2.GT.MAXN)MAXN=DEK2
5690          NORM=NORM+DEK2
5700      400 CONTINUE
5710        NORM=NORM/L
5720        IF(NORM.LE.EPSMEA.AND.MAXN.LE.EPSMAX)GO TO 500
5730        IF(NIT.GE.NITMAX)GO TO 500
5740        GO TO 100
5750      500 CONTINUE
5760        NITOT=NITOT+NIT
5770      END

```