INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 13, 371–399 (1991)

ANALYSIS OF PLANE AND AXISYMMETRIC FLOWS OF INCOMPRESSIBLE FLUIDS WITH THE STREAM TUBE METHOD: NUMERICAL SIMULATION BY TRUST-REGION OPTIMIZATION ALGORITHM

J. R. CLERMONT AND M. E. DE LA LANDE

Institut de Mécanique de Grenoble (IMG), BP 53X, F-38401 Grenoble Cedex, France

AND

T. PHAM DINH AND A. YASSINE

Equipe Modélisation et Optimisation, Lab. TIM 3, IMAG, BP 53X, F-38401 Grenoble Cedex, France

SUMMARY

New concepts for the study of incompressible plane or axisymmetric flows are analysed by the stream tube method. Flows without eddies and pure vortex flows are considered in a transformed domain where the mapped streamlines are rectilinear or circular. The transformation between the physical domain and the computational domain is an unknown of the problem. In order to solve the non-linear set of relevant equations, we present a new algorithm based on a trust region technique which is effective for non-convex optimization problems. Experimental results show that the new algorithm is more robust compared to the Newton–Raphson method.

KEY WORDS Incompressible fluids Streamlines Streamfunction Vortex flows Trust-region (TR) algorithm Newton–Raphson method

1. INTRODUCTION: METHODS OF FLOW ANALYSIS

Numerical simulation in fluid mechanics has undergone major developments in recent years in relation to numerous fundamental problems and applications which arise in the analysis of complex flows occurring with various types of fluids (see e.g. Reference 1). There are few analytical solutions for flow problems and researchers in this field wish to model flows by using either perturbation techniques on a close solution to their problem or direct computational methods. Thus, given a flow domain, the authors aim to define a suitable mathematical form in such a way that the set of equations and boundary conditions developed to describe the whole system can be successfully discretized and solved numerically. Velocity and pressure are the classic unknowns to be considered.

The mathematical models in this paper are based on conservation of mass and momentum in the isothermal case and the constitutive relations for the fluid in question. The incompressibility equation and Navier–Stokes equations are obtained for incompressible Newtonian fluids, and more complex dynamic equations are written for other constitutive equations,[1] e.g. models derived from viscoelasticity or plasticity. At the present time, important work has led to

0271–2091/91/130371–29$14.50

Received January 1990
Revised May 1990

interesting results in mathematical analysis, including existence, regularity, development of singularities[2,3] and change of type[4] in relation to the relevant equations. In the computational field, various flow simulations are performed involving finite difference[5] or finite element[6] techniques. The finite element methods are more widely investigated because they enable easy adaptation of a non-regular mesh to a complex domain of flow. However, in that context, grid generation methods can be used successfully as conformal mappings (see e.g. References 7 and 8), transfinite interpolation, multisurface and two-boundary techniques.[9] These methods allow the physical flow domain to be transformed into a computational domain, using a known transformation, in order to obtain a regular mesh in the transformed domain where velocity computation is related to inviscid fluids[10] or Newtonian fluids.[7,8]

Some years ago we introduced the stream tube method[11,12] for incompressible fluids. This can also be classified as a geometrical method since the physical domain D is mapped into a computational domain D_1. However, in contrast to grid generation methods, the unknowns of the problem are the pressure and the transformation between the domain D and its transformed domain D_1.

In the present paper, only steady flows will be considered, without loss of generality for unsteady situations. Before recalling the main assumptions already formulated for plane or axisymmetric non-circulatory flows,[12] we propose first a classification of flows which may be dealt with by stream tube analysis:

(1) plane or axisymmetric flows without vortices
(2) purely circulatory flows or vortex flows.

In the following we will examine these two classes of flows and their relevant equations. By performing the transformation of the physical domain, we have in mind two objectives:

(i) The computational domain is more simple than the original flow domain
(ii) The transformed lines of the physical streamlines are defined as straight lines or circles.

2. THE STREAM TUBE METHOD IN PLANE AND AXISYMMETRIC NON-CIRCULATORY FLOWS

As pointed out earlier, the elements of the stream tube method were initially proposed for plane and axisymmetric flows without eddies[11,12] in relation to the incompressibility condition

$$\operatorname{div} \mathbf{V} = 0. \tag{1}$$

We consider two possible expressions for the velocity vector $\mathbf{V}$:

$$\mathbf{V} = u(x, z)\boldsymbol{\varepsilon}_1 + w(x, z)\boldsymbol{\varepsilon}_2 \tag{2}$$

for plane flows, in Cartesian co-ordinates $x=1$, $y=2$, $z=3$;

$$\mathbf{V} = v_r(r, z)\mathbf{e}'_r + v_z(r, z)\mathbf{e}'_z \tag{3}$$

for axisymmetric flows, using a cylindrical co-ordinate system $r=1$, $\theta=2$, $z=3$.

In both cases the velocity components can be expressed in terms of streamfunctions $K(x, z)$ or $H(r, z)$ such that

$$u = \frac{\partial K}{\partial z}(x, z), \qquad w = -\frac{\partial K}{\partial x}(x, z), \tag{4}$$

$$v_r = \frac{\partial H}{\partial z}(r, z), \qquad v_z = -\frac{1}{r}\frac{\partial H}{\partial r}(r, z). \tag{5}$$

Therefore, by definition of the streamfunctions,[13] K or H is a constant on a streamline (L) of the physical domain.

Without loss of generality for extension to the planar case, we will consider axisymmetric flows in the present case. The main assumptions related to the method are the following:

(i) There exists a section z_0 of radius r_0 where the kinematics is known.
(ii) The physical domain D, shown in Figure 1, is simply connected with regard to the streamlines: the flow does not involve vortices.

From assumption (i) the streamfunction H is known and may be expressed in terms of the known velocity profile at z_0, as

$$r \in [0, r_0]: \quad H(r, z_0) = -\int_0^r v(\xi, z_0)\xi \, d\xi. \tag{6}$$

We denote by x^i the cylindrical co-ordinates in D:

$$x^1 = r, \qquad x^2 = \theta, \qquad x^3 = z; \tag{7}$$

and ξ^j is the system of variables defined as

$$\xi^1 = R, \qquad \xi^2 = \Theta, \qquad \xi^3 = Z \tag{8}$$

in a domain D_1.

We can thus define a one-to-one transformation $\xi^j(x^i)$ between a physical domain D and a transformed domain D_1 such that the transformed streamline ($\mathscr{L}$) from the original streamline (L) in D_1 is rectilinear and parallel to the axis of symmetry of D (Figure 1). The transformation

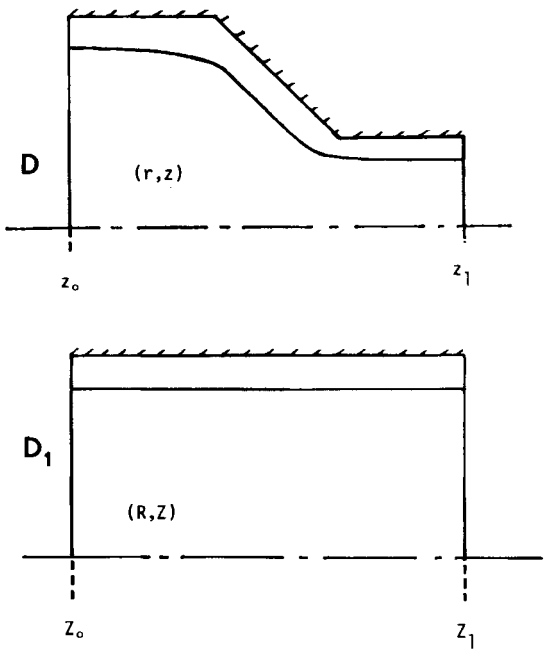

Figure 1. The stream tube method: transformation of a physical domain D into D_1

$\xi^j(x^i)$ is expressed by the equations

$$r=f(R, Z), \qquad \theta=\Theta, \qquad z=Z, \tag{9}$$

with the initial conditions

$$r=f(r, z_0)=R, \qquad \theta=\Theta, \qquad z_0=Z_0. \tag{10}$$

As shown in Figure 1, the transformed domain D_1 is a cylinder of radius r_0 of generants parallel to the axis of symmetry of D (and D_1).

Since the transformed streamlines are parallel straight lines, a transformed streamfunction H^* may be defined as

$$H^*(R)=H(r, z_0), \qquad r=R. \tag{11}$$

The transformed streamfunction H^* is constant and known on every streamline ($\mathscr{L}_0$) of the mapped domain D_1. The relevant system of variables has similarities with the Protean system co-ordinates Ψ^i as introduced and used by Duda and Vrentas[14] and Adachi.[15] In this system one co-ordinate is the streamfunction $H(r, z)$ such that

$$\Psi^1=H(r, z), \qquad \Psi^2=\theta, \qquad \Psi^3=\zeta. \tag{12}$$

A recent work of Papanastasiou *et al.*[16] has involved this Protean system in the computation of flows of fluids with memory, notably for the evaluation of kinematic tensors.

The derivatives in terms of r and z are written as

$$\frac{\partial}{\partial r}=\frac{1}{f'_R}\frac{\partial}{\partial R}, \qquad \frac{\partial}{\partial z}=\frac{-f'_Z}{f'_R}\frac{\partial}{\partial R}+\frac{\partial}{\partial Z}. \tag{13}$$

The Jacobian of the transformation is

$$J^*=\frac{\partial(r, \theta, z)}{\partial(R, \Theta, Z)}=f'_R. \tag{14}$$

J^* is non-singular in the absence of secondary flows. Upon this condition, the velocity components v_r and v_z become, from (12),

$$v_r=\frac{-f'_Z}{ff'_R}H^{*\prime}(R), \qquad v_z=\frac{-1}{ff'_R}H^{*\prime}(R). \tag{15}$$

From equation (15) the incompressibility condition or continuity equation (1) is readily verified. This formulation allows us to consider only the momentum conservation equation[13]

$$\rho\frac{\mathrm{D}\mathbf{V}}{\mathrm{D}t}=\rho\mathbf{F}_v+\mathrm{div}\,(\sigma), \tag{16}$$

where $\mathbf{F}_v$ denotes the body force vector, ρ is the fluid density and $\mathrm{D}/\mathrm{D}t$ is the material derivative. σ, the total stress tensor, is often broken into a separate isotropic component $-p\mathbf{I}$, where p is the pressure and $\mathbf{I}$ the identity tensor, and a rheological tensor component $\mathbf{T}$:

$$\sigma=-p\mathbf{I}+\mathbf{T}. \tag{17}$$

Using the velocities in terms of the new variables R and Z (equation (15)) and the explicit expression of $\mathbf{T}$, the momentum equations can be written as a non-linear set of differential equations

$$F_1(f, p)=0, \qquad F_2(f, p)=0. \tag{18}$$

These equations involve derivative terms of the unknowns f and p to be determined numerically.

3. PROBLEM FORMULATION FOR VORTEX FLOWS

These flows can be encountered either separately (Figure 2) or as subdomains in more general flow situations (secondary flows) as shown in Figures 3(a) and 3(b). It is now necessary to present the assumptions and analysis for an incompressible circulatory plane flow in a physical domain D. The velocity vector is given by (2). Assuming that secondary eddies as shown in Figure 4 are avoided, there exists only one stagnation point in the flow domain. Using the Cartesian co-ordinates

$$x^1 = x, \qquad x^2 = y, \qquad x^3 = z \tag{19}$$

and new variables defined as

$$\xi^1 = R, \qquad \xi^2 = y, \qquad \xi^3 = \phi, \tag{20}$$

it is possible to define a one-to-one transformation $\xi^j(x^i)$ between a physical domain D (involving one stagnation point C in the 'central' region) (Figure 5) by writing the following equations:

$$x = a + R\lambda(R, \phi)\sin\phi, \qquad y = Y, \qquad z = b + R\lambda(R, \phi)\cos\phi. \tag{21}$$

The function $\lambda(R, \phi)$ is unknown. The variables (R, ϕ) are related to the transformed domain D_1 of Figure 5. The triplet (a, y, b) involves the co-ordinates of the transformed point C_1 of the single stagnation point C of the physical domain. The point C_1 is the centre of concentric circles $(\mathscr{L})$ which are mapped streamlines of the lines (L) of the physical domain D. The transformed function K^* of the streamfunction $K(x, z)$ is constant along the lines $(\mathscr{L})$ and consequently depends only on the variable R. It should be pointed out that in the description of plane vortex flows there exists *a priori* no particular upstream section as featured in non-circulatory plane or axisymmetric flows. Hence the Jacobian J^* of the transformation $x^i(\zeta^j)$ is given by

$$J^* = \left|\frac{\partial(x, y, z)}{\partial(R, Y, \phi)}\right| = R\lambda(\lambda + R\lambda'_R). \tag{22}$$

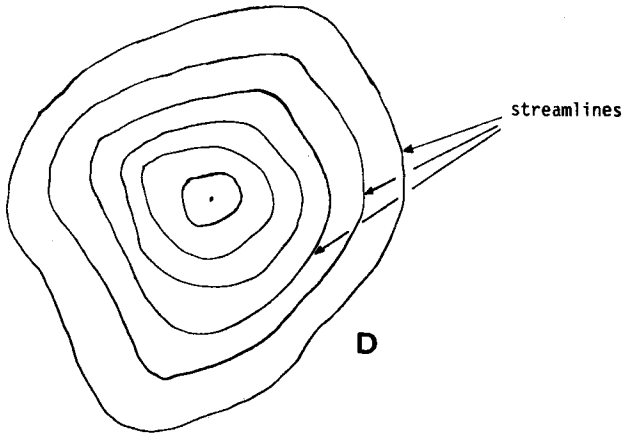

Figure 2. Domain of a purely circulating flow (vortex flow)

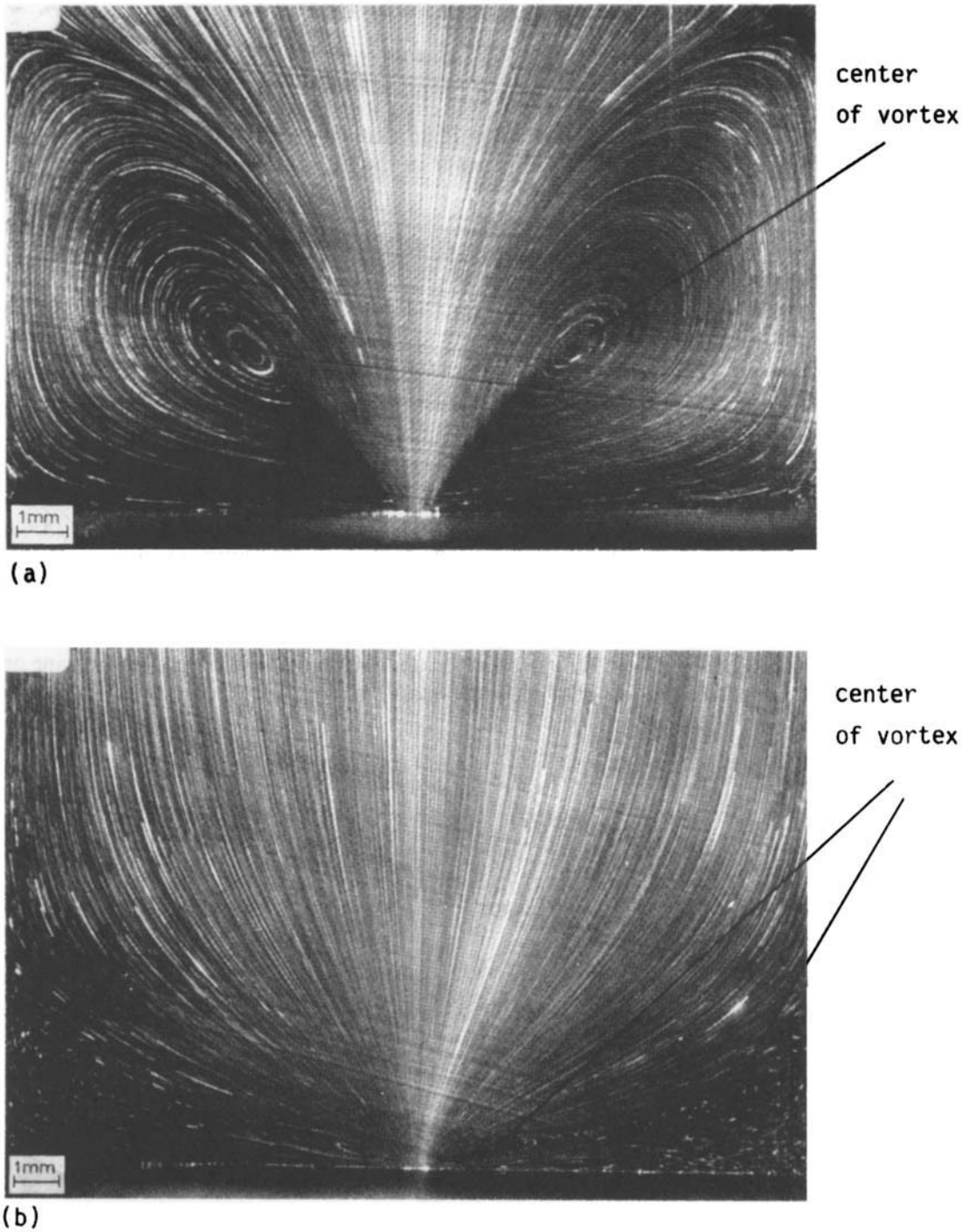

Figure 3. (a) Secondary flows of a polymer through an orifice. The mean velocity is 0·4 $m\,s^{-1}$ (the orifice diameter is 0·5 mm). Courtesy of Prof. J. M. Piau, Institut de Mécanique de Grenoble; (b) Secondary flows of a polymer involving two vortices. The mean velocity is 1·8 $m\,s^{-1}$ (the orifice diameter is 0·5 mm). Courtesy of Prof. J. M. Piau, Institut de Mécanique de Grenoble

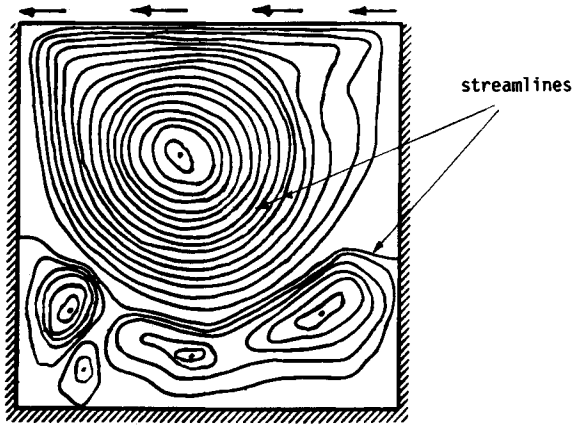

Figure 4. Multiple eddies (flow in a cavity domain)

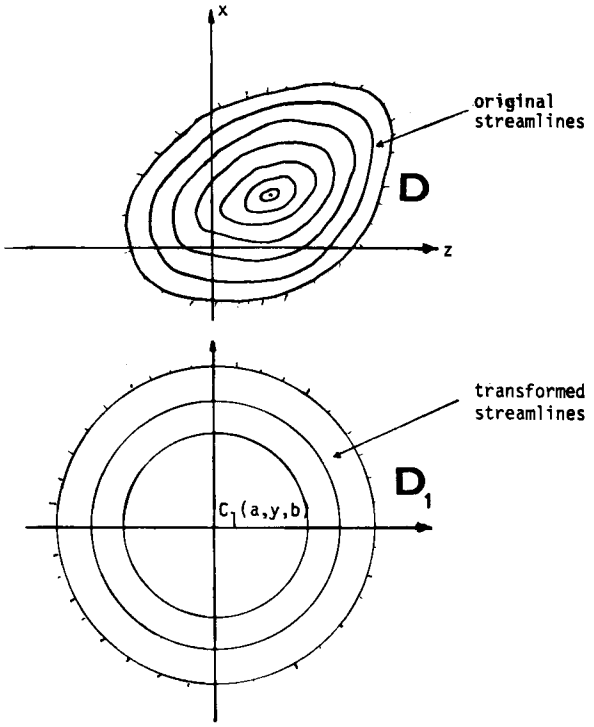

Figure 5. Transformation of the physical domain D into D_1 for a circulating flow

From (21) the derivative operators in terms of x and z may be written for $R \neq 0$, $\lambda \neq 0$ as follows:

$$\frac{\partial}{\partial x} = \frac{1}{R\lambda(\lambda + R\lambda'_R)} \left(R(\lambda'_\phi \cos\phi - \lambda \sin\phi) \frac{\partial}{\partial R} + (\lambda \cos\phi + R\lambda'_R \cos\phi) \frac{\partial}{\partial \phi} \right), \tag{23}$$

$$\frac{\partial}{\partial z} = \frac{1}{R\lambda(\lambda + R\lambda'_R)} \left(R(\lambda'_\phi \sin\phi + \lambda \cos\phi) \frac{\partial}{\partial R} - (\lambda \sin\phi + R\lambda'_R \sin\phi) \frac{\partial}{\partial \phi} \right). \tag{24}$$

The expressions of the two velocity components u and w thus become

$$u=\frac{\lambda'_\phi \sin\phi-\lambda\sin\phi}{\lambda(\lambda+R\lambda'_R)}\,K^{*\prime}(R), \tag{25}$$

$$w=\frac{\lambda'_\phi \sin\phi+\lambda\cos\phi}{\lambda(\lambda+R\lambda'_R)}\,K^{*\prime}(R). \tag{26}$$

From (25) and (26), mass conservation is readily verified. As in the former case, the use of (25) and (26) and the choice of a Newtonian of non-Newtonian constitutive equation in the dynamic equations (16) lead to a non-linear set of differential equations of unknowns λ and p:

$$G_1(\lambda, p)=0, \qquad G_2(\lambda, p)=0. \tag{27}$$

The spatial variables are R and ϕ.

Unlike the computational domain involving rectilinear streamlines, the streamfunction $K(x, z)$ and its transform $K^*(R)$ are unknowns and must be computed with the unknown transformation $\lambda(R, \phi)$ and the pressure $p(R, \phi)$.

4. BOUNDARY CONDITIONS IN THE STREAM TUBE METHOD

In the present analysis the boundary conditions are related to the unknown transformation f or λ and the pressure p. It can be easily understood that since the computational domain is simpler than the original flow domain D, the resulting equations (18) and (27) which involve the unknown functions f or λ and p are more complicated than those written with the original variables of the physical domain. Moreover, the method involves the description of flow by streamlines. Slight changes of streamlines may lead to considerable modification of velocities and stresses induced by the computation of kinematic gradients (such as the rate-of-deformation tensors and the Cauchy–Green tensors which are used in the expression of constitutive equations[17]). Approximate numerical solutions of the equations must be obtained by means of efficient and reliable computational methods. Before dealing with this point, it is necessary to consider the boundary conditions to be used with the non-linear equations (18) and (27).

4.1. Non-circulatory flows

In this case the transformed streamlines are rectilinear and the flow domain D is generally located between two fully developed regions. The duct is assumed to be of constant diameter upstream and downstream of sections z_0 and z_1 respectively. The entrance and exit Poiseuille velocity profiles can be computed analytically (in the Newtonian case for instance) or numerically for more complicated constitutive equations. In such situations Clermont and de la Lande[18] have shown that the computation of flow in D_1 is possible for a simply connected geometry by considering successive stream bands limited by two transformed streamlines, from the wall to the 'central' region of the flow, provided the action of the complementary domain is considered (Figure 6). In axisymmetric flows the forces exerted on the removed part are considered by writing the following appropriate equation:

$$\int_{\partial B_{ic}} \boldsymbol{\sigma}(f, p)\cdot\mathbf{n}(f)\,\mathrm{d}s=0, \tag{28}$$

where ∂B_{ic} denotes the closed surface which limits the complementary domain and $\mathbf{n}$ is the outer normal vector to the surface. The total stress tensor $\boldsymbol{\sigma}$ involves the pressure p (equation (17)) and

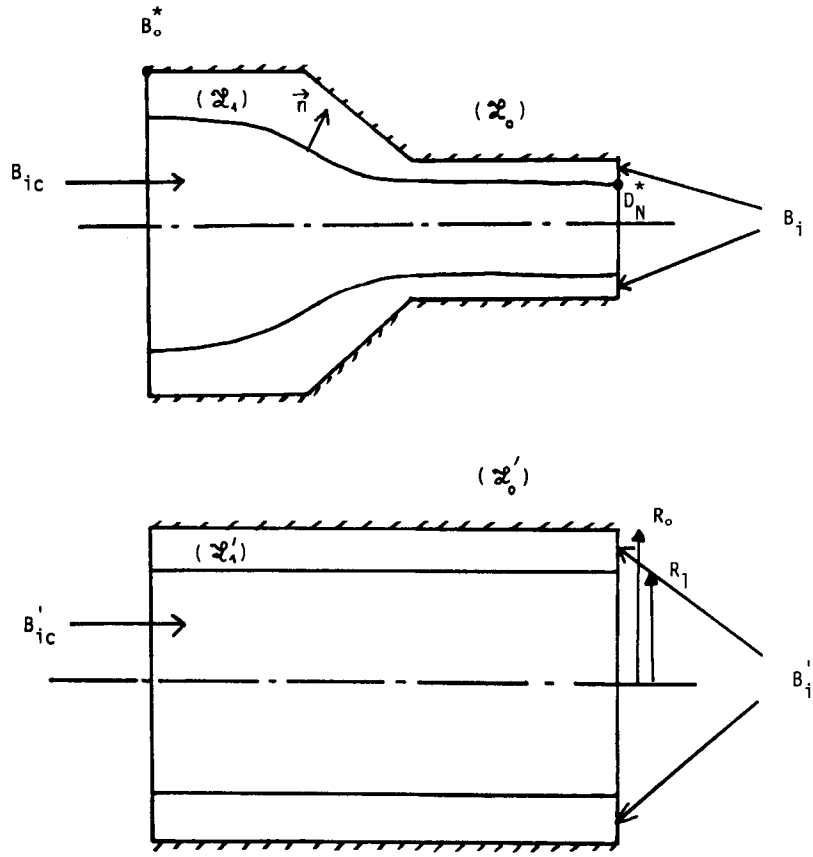

Figure 6. Stream tube B_i and its complementary domain B_{ic} in D. Transformed subdomains B_i' and B_{ic}' in D_1

the derivatives of the velocity. The components α_r and α_z of the unit normal vector $\mathbf{n}$ are

$$\alpha_r = \frac{1}{\sqrt{(1+f_z'^2)}}, \qquad \alpha_z = \frac{-f_z'}{\sqrt{(1+f_z'^2)}}. \tag{29}$$

Knowing the entrance and exit velocity profiles, a single value ξ_1 corresponds to a value ξ_0 at z_0 at section z_1. The function f is known at the wall and must verify the integrodifferential equation (28) for an unknown line (L_i) which limits the subdomain B_i. The pressure distribution at the upstream section z_0 is known provided a pressure value is assigned at the wall point $B(r_0, z_0)$.

The equations of the problem involve the dynamic equations (18) and the boundary equation (28). Stream tube analysis enables us to compute the flow field in a simply connected geometry by considering subdomains of the entire domain, thus allowing us to limit the storage area and CPU time.

In the case of doubly connected geometries as shown, for example, in Figure 7, the use of the boundary condition (28) for the complementary domain B_{ic} implies the computation of unknown stresses at the inner wall S_1. Thus the flow field cannot be determined by computation of successive stream bands in D_1. Consideration of the entire domain D_1 may be an alternative solution.

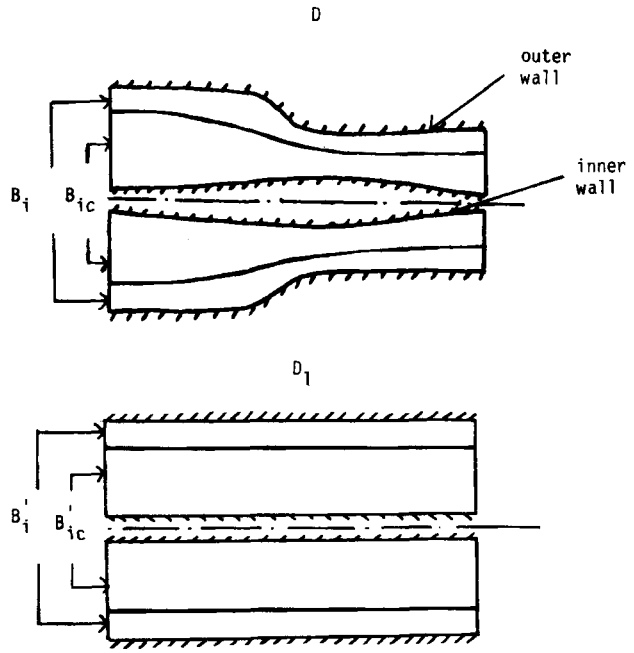

Figure 7. Doubly connected geometry

4.2. Circulatory flows

The transformed domain D_1 of Figure 5 involves circular streamlines described with co-ordinates R and Z. No additional forces are to be involved when considering a peripheral 'stream ring' limited by a known streamline ($\mathscr{L}$) and an unknown streamline ($\mathscr{L}_i$). An equation of the type (27) is not necessary and only the dynamic equations are to be written. The flow can be computed in successive stream rings from the wall to the stagnation point.

The case of a doubly connected flow domain is illustrated in Figure 8. The geometry is that of a journal bearing. The fluid is contained between the two cylinders. The inner one rotates at a constant velocity ω. When considering a stream ring in the transformed domain D_1, the rotation of the inner wall produces an action to be involved in the equations and the former procedure cannot be applied.

5. APPLICATION TO NON-CIRCULATING FLOWS: THE NEWTONIAN CASE

In the following we investigate the flow of a Newtonian fluid in a converging geometry (Figure 1). The cone angle α of the convergent is small enough to avoid recirculations.

5.1. Dynamic and boundary equations for a Newtonian fluid

We consider the case of a Newtonian constitutive equation

$$\mathbf{T}=2\mu\mathscr{D}, \tag{30}$$

where $\mathscr{D}$ denotes the rate-of-deformation tensor and μ is the viscosity.

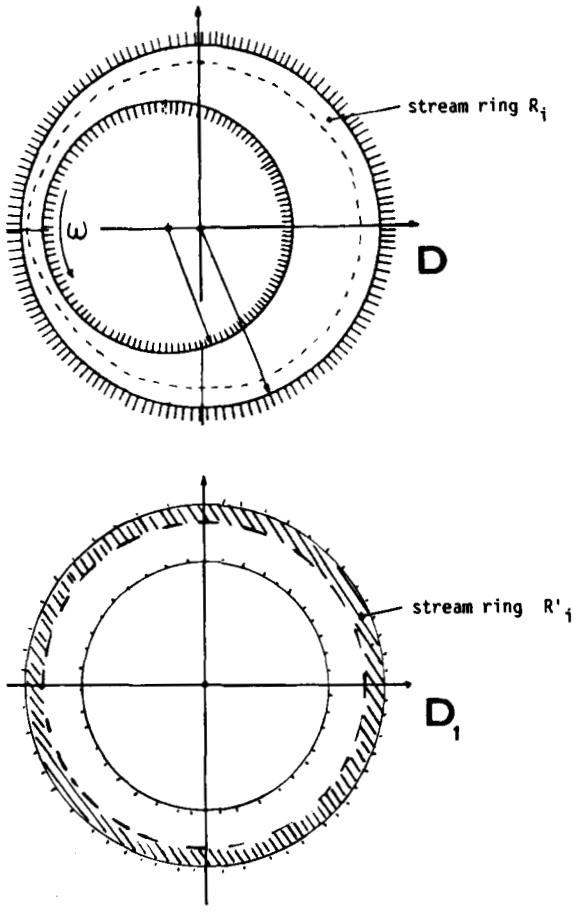

Figure 8. Mapping of the flow domain of a journal bearing

The components of the tensor $\mathscr{D}$ may be expressed from (13) and (15) in terms of the variables (R, Z) or from (23)–(26) in terms of (R, ϕ). Let us consider a non-circulating flow in the axisymmetric converging geometry shown in Figure 1. Two Poiseuille flows are considered at z_0 and z_1 corresponding to the following equations for the velocity:

$$v_r(r, z_0)=0, \qquad v_z(r, z_0)=\alpha_0(r_0^2-r^2), \qquad v_r(r, z_1)=0, \qquad v_z(r, z_1)=\alpha_1(r_1^2-r^2), \tag{31}$$

where the relation between α_0 and α_1 is

$$\alpha_0 r_0^4=\alpha_1 r_1^4. \tag{32}$$

From equations (12) the dynamic equations (15) in cylindrical co-ordinates are expressed as

$$0=\frac{-1}{f'_R}\frac{\partial p}{\partial R}+\frac{1}{f'_R}\frac{\partial T^{11}}{\partial R}+\frac{T^{11}-T^{22}}{f}-\frac{f'_Z}{f'_R}\frac{\partial T^{13}}{\partial R}+\frac{\partial T^{13}}{\partial Z}, \tag{33}$$

$$0=\frac{f'_Z}{f'_R}\frac{\partial p}{\partial R}-\frac{\partial p}{\partial Z}+\frac{1}{f'_R}\frac{\partial T^{13}}{\partial R}+\frac{T^{13}}{f}-\frac{f'_Z}{f'_R}\frac{\partial T^{33}}{\partial R}+\frac{\partial T^{33}}{\partial Z}. \tag{34}$$

The superscripts 1 and 3 which are used for the tensor components are related to the cylindrical co-ordinates r and z respectively. The detailed expressions of (33) and (34) in terms of f and p have been given in a previous paper.[12] Only the general form is presented here as

$$-f^2 f_R'^4 \frac{\partial \Pi}{\partial R} + A_1 H^{*\prime}(R) + A_2 H^{*\prime\prime}(R) + A_3 H^{*\prime\prime\prime}(R) = 0, \tag{35}$$

$$f^3 f_R'^4 f_Z' \frac{\partial \Pi}{\partial R} - f^3 f_R'^5 \frac{\partial \Pi}{\partial Z} + B_1 H^{*\prime}(R) + B_2 H^{*\prime\prime}(R) + B_3 H^{*\prime\prime\prime}(R) = 0, \tag{36}$$

in terms of the variables f and Π, with

$$\Pi = p/\mu. \tag{37}$$

The derivatives $H^{*\prime}$, $H^{*\prime\prime}$ and $H^{*\prime\prime\prime}$ are known quantities and A_1, A_2, A_3, B_1, B_2 and B_3 denote functions of derivatives of f.

The streamfunction H^* is computed from equations (4) and (10). Hence

$$H^*(R) = \frac{\alpha_0}{4}(R^4 - 2R_0^2 R^2). \tag{38}$$

The non-linear system of dynamical equations is to be used together with the boundary condition (28) and boundary values related to f and Π.

Owing to the symmetry of the flow, the vector equation (28) is reduced to

$$\left(\int_{\partial B_{ic}} \boldsymbol{\sigma}(f, p) \cdot \mathbf{n}(f) \, \mathrm{d}s \right) \mathbf{e}_z' = 0. \tag{39}$$

We now consider the particular points B_0^* and D_N^* in the physical domain D of Figure 6. From (38) we obtain for a Newtonian fluid

$$\begin{aligned} &-2 \int_{z_0}^{z_1} (\Pi f f_Z')(R_1, \zeta) \, \mathrm{d}\zeta \\ &-2H^*(R_1) \int_{z_0}^{z_1} \left(\frac{f_R'^2 + f f_{R^2}'' + f f_Z'^2 f_{R^2}'' - f f_R'^2 f_{Z^2}''}{f f_R'^3}(R_1, \zeta) - \frac{f_Z'^2 + 1}{f_R'^2}(R_1, \zeta) H^{*\prime\prime}(R_1) \right) \mathrm{d}\zeta \\ &+ \Pi(D_N) R_1^2 \left(\frac{r_1}{r_0} \right)^2 - \Pi(B_0) R_1^2 = 0. \end{aligned} \tag{40}$$

This integrodifferential equation involves unknowns f and Π on the streamline $\mathscr{L}_1$ which is to be determined. Concerning the unknowns of the problem, it should be pointed out that: the function f is known at the wall and at sections z_0 and z_1; the pressure p is given only at section z_0.

Now we wish to compute the flow field by considering successive stream bands in the mapped domain D_1.

In this paper we will only present the computational analysis and the results for the first band limited by the transformed wall $\mathscr{L}_0'$ and the first line $\mathscr{L}_1'$, the transformed curve of which is to be computed (see Figure 6).

It has been shown[12, 18] that the difficulties have mainly concerned the computation of f and p in the first band B_0', particularly owing to the fact that the pressure p is unknown at the wall $\mathscr{L}_0'$. The procedure can be readily extended to the successive stream bands in D_1.

Two points are to be investigated: the discretization of the unknowns on a stream band in D_1; the computational method to be applied for the numerical determination of the unknowns f and p.

5.2. Discretization; computational methods

The peripheral stream band B'_0 in D_1 involves the transformed wall $\mathscr{L}'_0$ and the line $\mathscr{L}'_1$ related to the unknown streamline $\mathscr{L}_1$ of D. We consider N elements $A^iB^iC^iD^i$ in B'_0, from $Z_0=z_0$ to $Z_1=z_1$, as shown in Figure 9. On each element (i) of B'_0 the unknowns f and p at a point M are assumed to be expressed in terms of second-order polynomials of R and Z as

$$f_M=f_{B^i}+(R_M-R_{B^i})x_1^i+(Z_M-Z_{B^i})x_2^i+(R_M-R_{B^i})^2x_3^i+(Z_M-Z_{B^i})^2x_4^i$$
$$+(R_M-R_{B^i})(Z_M-Z_{B^i})x_5^i, \tag{41}$$

$$p_M=p_{B^i}+(R_M-R_{B^i})x_6^i+(Z_M-Z_{B^i})x_7^i+(R_M-R_{B^i})^2x_8^i+(Z_M-Z_{B^i})^2x_9^i$$
$$+(R_M-R_{B^i})(Z_M-Z_{B^i})x_{10}^i, \tag{42}$$

where (R_M, Z_M) and (R_{B^i}, Z_{B^i}) are co-ordinates of points M and B^i respectively in the element (i). The superscript i indicates an element (i), for which 10 unknowns are involved. Additional compatibility equations are written to complete the set of differential equations (35), (36) and boundary condition (40).

We then obtain a closed non-linear system of $10N$ equations of the form

$$\phi(x)=0, \tag{43}$$

where $n=10N$ and $\phi(x)=(\phi_1(x), \ldots, \phi_n(x))$ is twice continuously differentiable from $\mathbb{R}^n$ to $\mathbb{R}^n$. Here the Jacobian $J(x)$ of $\phi(x)$ is non-singular. It is clear that the non-linear equation (43) is equivalent to the following non-linear least squares problem:

$$0=\min\left\{\psi(x)=\tfrac{1}{2}\sum_{i=1}^{n}\phi_i(x)^2 : x\in\mathbb{R}^n\right\}. \tag{44}$$

By simple calculus we obtain

$$\nabla\psi(x)=\sum_{i=1}^{n}\phi_i(x)\nabla\phi_i(x)=J(x)^{\mathrm{T}}\phi(x), \tag{45}$$

$$\nabla^2\psi(x)=\sum_{i=1}^{n}\nabla\phi_i(x)\nabla\phi_i(x)^{\mathrm{T}}+\sum_{i=1}^{n}\phi_i(x)\nabla^2\phi_i(x)=J(x)^{\mathrm{T}}J(x)+\sum_{i=1}^{n}\phi_i(x)\nabla^2\phi_i(x). \tag{46}$$

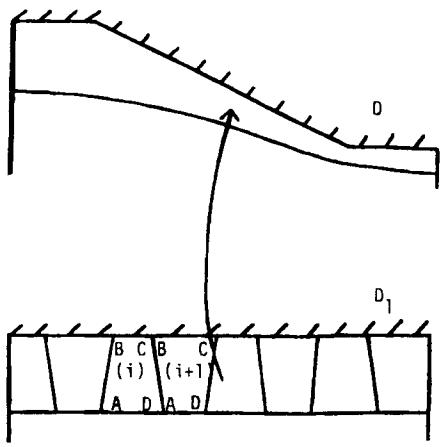

Figure 9. Elements in the computational domain

The Newton–Raphson method was applied first to the dynamic equations. A limit of convergence was obtained for a cone angle of the convergent $\alpha^* = 29°3$. This procedure has led to the definition of an optimal mesh in the computational domain. The elements are trapezoidal and their optimal shape, shown in Figure 9, is consistent with the current results obtained in grid generation techniques. The same grid is retained for the further numerical experiments.

In order to obtain solutions beyond the angle α^* given by the Newton algorithm, we now consider the computational problem of determining the unknowns f and p by using the trust-region (TR) optimization algorithm.

6. TRUST-REGION METHOD IN UNCONSTRAINED MINIMIZATION

Let us consider the unconstrained problem (P):

$$(\mathrm{P}): \quad \min\{\psi(x): x \in \mathbb{R}^n\}, \tag{47}$$

where ψ is a function from $\mathbb{R}^n$ to $\mathbb{R}$. In this section we wish to describe and analyse a technique related to the solution of (47).

The approach we shall present is well known.[19–22] It is appropriately called a 'model trust-region approach': the step to a new iterate is obtained by minimizing a local quadratic model to the objective function over a restricted spherical region centred about the current iterate. The diameter of this region is expanded and contracted in a controlled way based upon how well the local model predicts the behaviour of the objective function. It will be seen that it is possible to control the iteration in such a way that convergence is forced from any starting value assuming reasonable conditions on the objective function.

Some very strong convergence properties for this method will be given. It will be shown that one can expect (but not ensure) that the iteration will converge to a point which satisfies the second-order necessary conditions for a minimum.

The trust-region method computes the sequence of iterates by solving at each step a quadratic problem (P_k) with a Euclidean norm constraint:

$$(\mathrm{P}_k): \quad \min\{q_k(d): \|d\| \leq \delta_k\}, \tag{48}$$

where q_k is a quadratic approximation of the variation of ψ at x_k defined by

$$q_k(d) = \langle g_k, d\rangle + \tfrac{1}{2}\langle H_k d, d\rangle, \tag{49}$$

g_k is the gradient of the function ψ at x_k and H_k is the Hessian of ψ at x_k or an approximation of it, which therefore will be called 'quasi-Hessian'. The strictly positive number δ_k is called the 'trust radius'.

In trust-region algorithms it is necessary to compute the gradient of the objective function. The use of first-order information leads in general to the first-order stationary point. By incorporating the second-order information $H_k = \nabla^2\psi(x_k)$, these algorithms may satisfy the second-order necessary conditions for the problem (P) of (47). In this case the trust-region method can be regarded as a modified Newton method applied to finding a zero gradient of the objective function. The procedure can be described as follows.

Computing the direction d_k, the solution of (P_k), we can easily check the quality of the local approximation $q_k(d)$ and hence take the appropriate decision.

(a) If the approximation is satisfactory, then the solution of (P_k) yields a new iterate; in addition, the trust radius is increased.

(b) In the opposite case the iterate is unchanged; in addition, the trust radius is decreased until q_k yields a satisfactory approximation inside the trust region (which necessarily occurs for small radii since the gradient is supposed to be exactly known and the first-order term in q_k becomes dominant if $\|g_k\| \neq 0$).

The quality of the approximation is generally examined through the following quantity:

$$r_k = \frac{\psi(x_k) - \psi(x_k + d_k)}{q_k(0) - q_k(d_k)}. \tag{50}$$

Here r_k is called the 'quality coefficient' and represents the ratio of the actual reduction of ψ when moving from x_k to $x_k + d_k$ and the predicted reduction according to the quadratic approximation. Thus in a trust-region algorithm the main source of computation effort, apart from the function evaluation required, is the work on a problem of the form (P_k) of (48) to determine the step from the current iterate. A practical aspect of this procedure is given in the following example illustrated in Figure 10.

Let x_0 be the initial iterate of the sequence $\{x_k\}$ and δ_0 be the corresponding trust-region radius of the sphere (or $\mathbb{R}^n$) of centre x_0 (Figure 10).

Step 1. The algorithm leads to an admissible approximation $x_0 + d^*$ (by using the coefficient r_k of (50)). Then a new iterate $x_1 \neq x_0$ is updated and the trust radius δ_1 may be increased:

$$\delta_1 > \delta_0.$$

Step 2. The approximation $x_1 + d^*$ is not satisfactory and the iterate remains unchanged:

$$x_2 = x_1.$$

Then the trust radius δ_2 of the sphere of centre x_2 is reduced:

$$\delta_2 < \delta_1.$$

The next steps k of the algorithm lead to determination of the position of x_k and the corresponding radius δ_k until convergence is obtained.

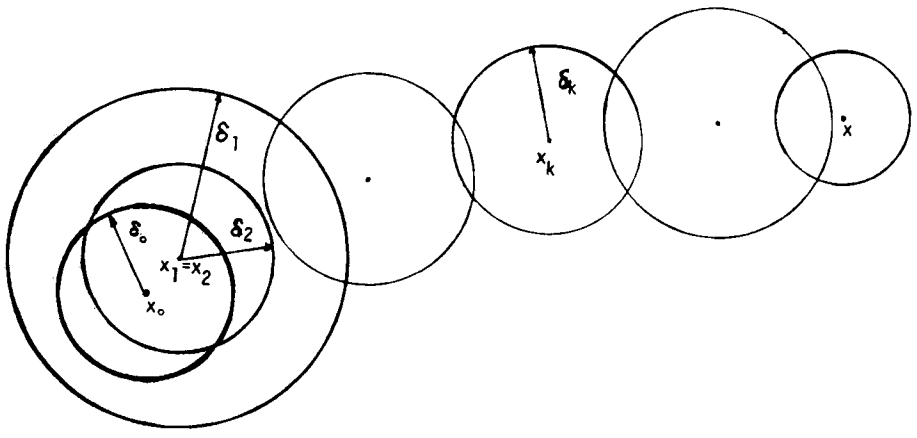

Figure 10. An illustrative example for the trust-region method

It should be pointed out that trust-region algorithms differ in their strategies for approximately solving (P_k) (see Section 6.1).

6.1. Algorithms for problems (P) and (LP)

Bearing in mind the general procedure for solving the problem (48), we shall be applying the following algorithm.

ALG1

1. Let $x_0 \in \mathbb{R}^n$, $\delta_0 > 0$ and H_0 be given.
2. For $k = 0, 1, 2, \ldots$
   (a) Compute $g_k = \nabla\psi(x_k)$; if $g_k = 0$, x_k will be taken as an optimal solution. Stop! (see Section 6.3).
   (b) Determine a solution d_k to problem (P_k) and compute the quality coefficient r_k via (46).
   (c) Update the iterate.
   (d) Update the trust radius and the quasi-Hessian and go to step 2.

Points (a), (b) and (c) of the above abstract algorithm are discussed in Appendix I and are closely related to the work of Gay,[23] Sorensen,[20] Moré[19] and Dennis and Schnabel.[24]

The local problem (LP) involved by ALG1 is reduced to that of minimizing a quadratic form inside a sphere:

$$\text{(LP)}: \quad \min\{\langle g, d\rangle + \tfrac{1}{2}\langle d, Hd\rangle : \|d\| \leq \delta\}, \tag{51}$$

where g is a n-vector, H is a symmetric matrix and δ is a positive number.

Since the constraint set $\{d \in \mathbb{R}^n : \|d\| \leq \delta\}$ is compact, the problem (LP) has a solution. If in addition H is positive definite, then there is unicity of solution to (LP).

A complete discussion of the theoretical aspects of the problem (LP) of (51) and the nature of the computational difficulties that may be encountered are presented in Appendix II. Some of the results are known (see e.g. References 20–24). A statement and proof of these results that is more complete and better suited to this presentation are given.

Two algorithms related to the local problem (LP) are detailed in Appendix III. The first is the Hebden algorithm and is generally applied when H is positive semidefinite. The second algorithm uses curvature information and is carried out to deal with 'hard cases' (see 2(i) and 3(ii) in Lemma 2 of Appendix II). In this algorithm the computation of the smallest eigenvalue λ_1 of H and of an eigenvector of H corresponding to λ_1 is required.

6.2. Convergence tests

We can use the following termination criteria in trust-region algorithms.

First-order necessary condition test. If $\nabla\psi(x_k) = 0$, then stop.

Second-order sufficient condition. When the first-order necessary condition is satisfied, we can stop the algorithm or proceed to verify the second-order sufficient condition as follows.

(i) If H_k is not positive definite, then the algorithm continues (see Lemma 3 of Appendix II for the solution of (LP) in the case where $g = 0$).
(ii) If $\nabla^2\psi(x_k)$ is positive definite, then stop. Otherwise $H_k = \nabla^2\psi(x_k)$ and we restart the algorithm from step (d) (of ALG1).

The convergence results of the TR method are given in Appendix IV.

6.3. *Adapted TR algorithm*

In this subsection we will present the practical TR-based algorithm applied in our experiments.

ALG2

0. Initialization:
   $x^{(0)}$, the initial iterate
   $\delta_0 > 0$, initial trust radius
   $\varepsilon > 0$, 'zero' for general tests
   $\varepsilon_g > 0$, zero for $\| g \|$
   $\varepsilon_\psi > 0$, zero for ψ
   $k = 0$, iteration counter.
1. Calculating $\psi_k = \psi(x^{(k)})$, $g_k = \nabla\psi(x^{(k)})$ and H_k, the Gauss–Newton approximation of $\nabla^2\psi(x^k)$ ($H_k = J(x^{(k)})^T J(x^{(k)})$ (see (46)), which is positive definite in our problem).
2. If $\| g_k \| \leq \varepsilon_g$ or $\psi_k \leq \varepsilon_\psi$, then $x^{(k)}$ is out solution; stop.
3. Calculating d_k:
   solve $H_k d = -g_k$
   if ($\| d \| \leq \delta_k$) then $d_k = d$
   else using Hebden algorithm to find a $\mu > 0$ so that the solution of $(H_k + \mu I)d = -g_k$ satisfies $| \| d \| - \delta_k | \leq \varepsilon$, then $d_k = d$.
4. Calculating δ_{k+1} and x_{k+1}:
   let $r_k = [\Psi(x_k) - \Psi(x_k + d_k)]/[q_k(0) - q_k(d_k)]$
   if $r_k \geq 0{\cdot}25$, then
   begin
   $x^{(k+1)} = x^{(k)} + d_k$
   if $r_k \geq 0{\cdot}75$, then $\delta_{k+1} = 2\delta_k$
   else $\delta_{k+1} = \delta_k$
   $k = k + 1$ and return to step 1
   end
   else (if $r_k < 0{\cdot}25$)
   $\delta_k = \delta_k/2$ and return to step 3.

7. COMPUTATION IN CONVERGING GEOMETRIES: RESULTS AND DISCUSSION

In order to test and validate the general behaviour of the proposed mathematical model, various numerical experiments were performed. The TR algorithm was first applied to a Poiseuille flow problem. The transformation is the identity and the pressure variation is linear versus z (or Z). The derivatives of the streamfunction H^* are computed from (38).

A DN 4000 Apollo workstation was used in all our tests. Double-precision variables were considered. The first aim was to compare the TR technique and the Newton–Raphson method in the Poiseuille flow problem from the points of view of robustness and accuracy.

According to the numerical results, the accuracy of the two methods was found to be equivalent. In order to test the efficiency of the algorithms, we started from a perturbation of 5% for the unknowns $\{x_n^i\}$, $n = 1, \ldots, N$, related to the elements of the peripheral stream band B_0. A divergence of the Newton–Raphson method was observed while the TR algorithm still converged to the analytical Poiseuille solution.

For converging geometries of angle $\alpha > 0$ we adopted as computational domain the peripheral stream band B_0 of the mapped domain D_1 (Figure 9). Given a converging angle α, the flow field

was computed using as initial estimate x_0, the solution vector x that had been computed for an angle $\alpha' < \alpha$, such that

$$a = \alpha - \alpha' > 0. \tag{52}$$

Using the Newton–Raphson procedure, the distance a was always chosen to be smaller than 5°. This distance had to be strongly reduced for angles α close to $\alpha^* = 29°3$, which is the limit angle of convergence for the method.

In the context of the TR algorithm the numerical results were determined for distances a of the order of 10°. For $\alpha \leq \alpha^*$ the numerical data were found to be identical to those given by the Newton–Raphson method. Moreover, convergence was still obtained for angles greater than α^*.

In Figure 11 we give, versus z, the evolution of the computed streamline ($\mathscr{L}_1$) of the peripheral stream band B_0, the dimensionless pressure p/p_0 and the dimensionless axial component of the

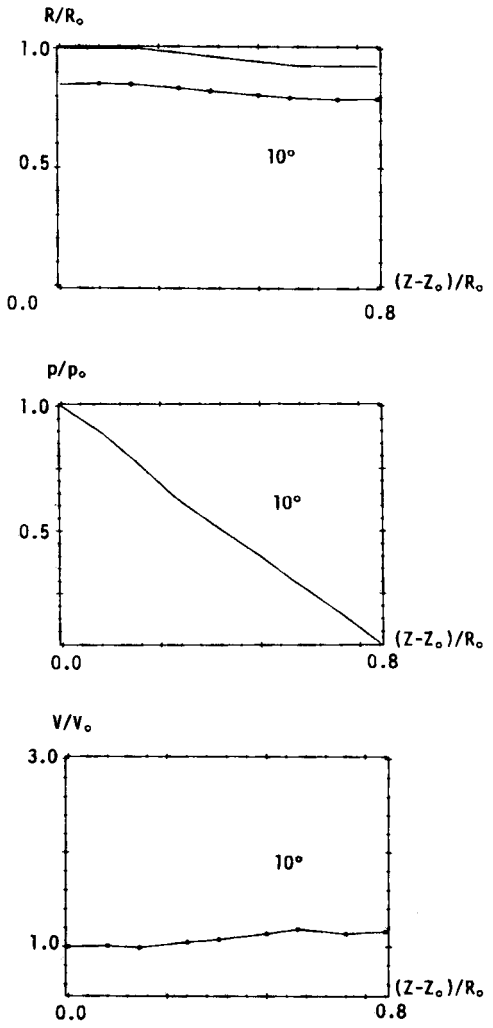

Figure 11. Computed streamline, dimensionless pressure p/p_0 and velocity v/v_0 on the streamline versus the dimensionless abscissa $(z-z_0)/R_0$ for an axisymmetric converging flow of angle $\alpha = 10°$ (Newton–Raphson and TR algorithm)

velocity v/v_0 on the streamline, where v_0 denotes the velocity at $z=z_0$. The cone angle is 10°. The numerical results of the two methods are found to be identical.

Figure 12 shows the computed streamline of B_0, the pressure distribution and the velocity on the streamline for a cone angle $\alpha=45°$ determined with the TR method.

In Figure 13 are shown the evolutions of the computed streamline ($\mathscr{L}_1$) and the corresponding pressure, versus a dimensionless axial distance, for $\alpha=40°$, 50° and 55°.

A divergence of the TR algorithm is found for a cone angle $\alpha=60°$. This failure of the computational method applied to a converging flow situation is to be analysed from both theoretical and physical points of view. As outlined in Section 2, one main assumption for the

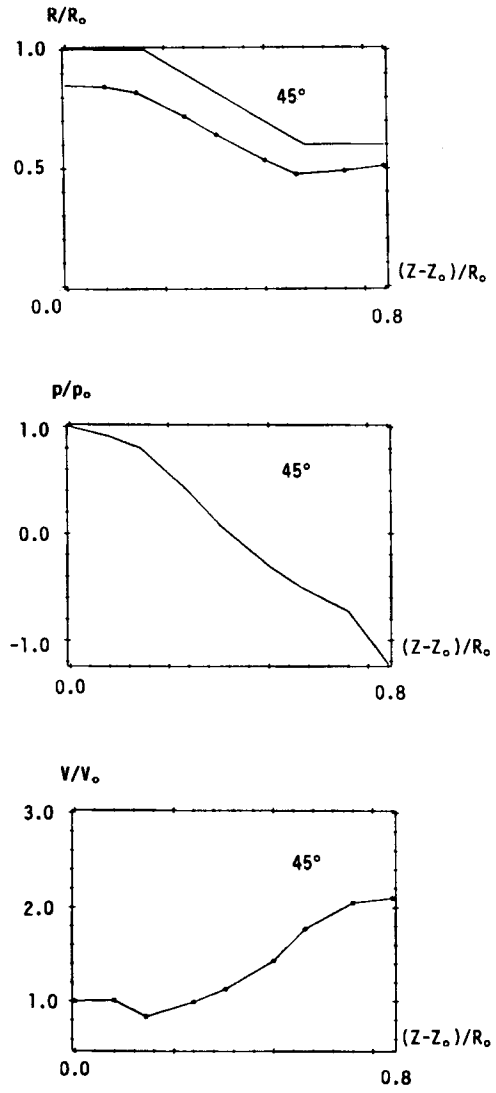

Figure 12. Computed streamline, dimensionless pressure p/p_0 and velocity v/v_0 on the streamline versus the dimensionless abscissa $(z-z_0)/R_0$ for an axisymmetric converging flow of angle $\alpha=45°$ computed with the TR algorithm

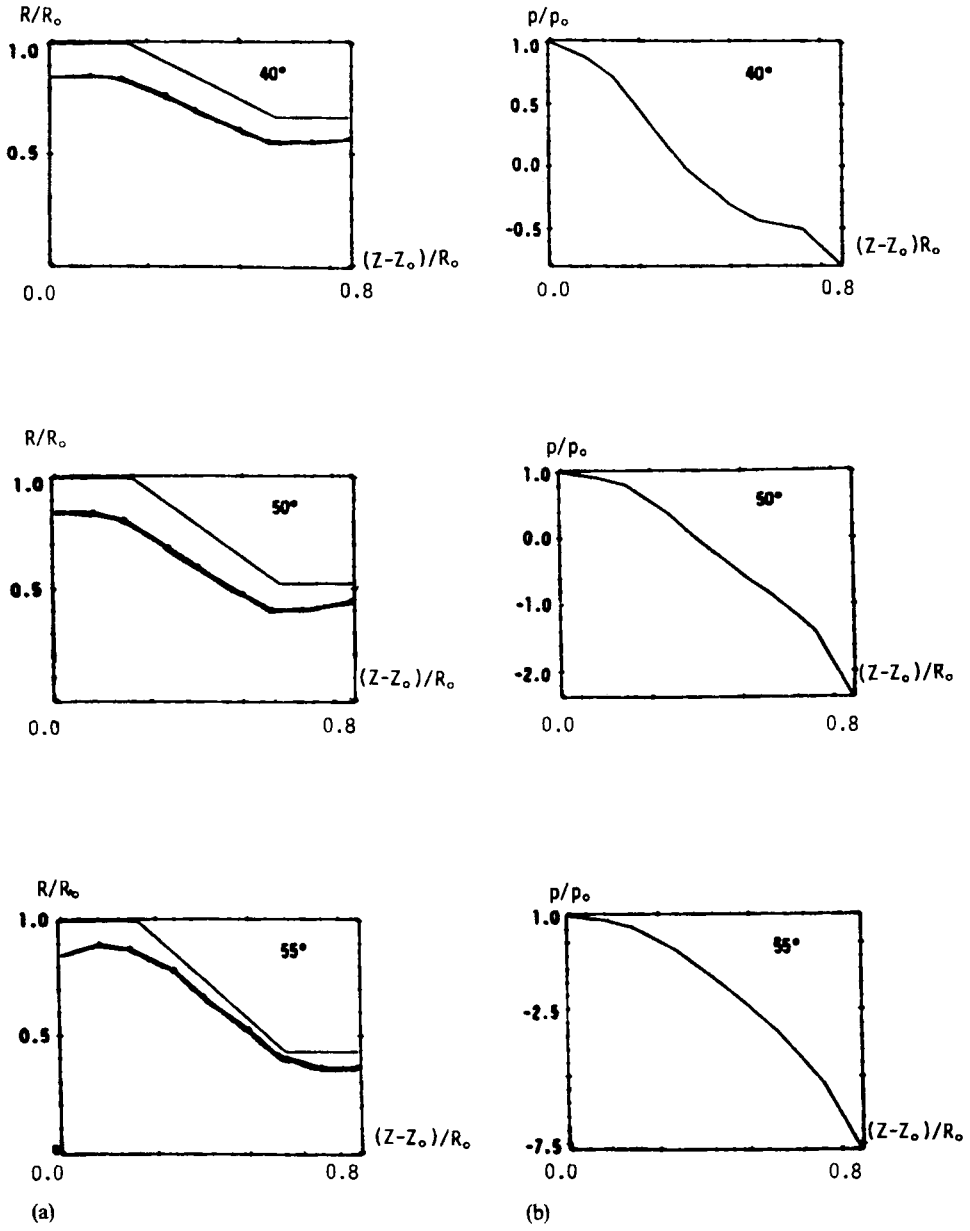

Figure 13. Evolutions of the (a) computed streamline and (b) pressure for different angles of the converging geometry

applicability of the stream tube method concerns the non-existence of secondary flows; otherwise the Jacobian of the transformation f'_R (equation (14)) vanishes and a circulatory flow can exist. An interesting point was to test the robustness of the TR algorithm in relation to the appearance of secondary flows. According to general experimental data, flows in converging geometries usually exhibit vortices for angles $\alpha \geq 60°$ (given a contraction ratio of the upstream and downstream diameters). We wanted to compare our numerical results with those given using the fluid flow

analysis software package POLYFLOW of Professor Crochet (a Fortran finite element program for calculating viscous flow; Université Catholique de Louvain, Louvain-La-Neuve, Belgium) in the same flow conditions as those of the TR procedure. The POLYFLOW code indicated a limit angle $\alpha = 61°$, which is very close to the result ($\alpha = 60°$) given by the TR algorithm.

8. CONCLUSIONS

In this paper we have presented a new method for the analysis of incompressible plane or axisymmetric flows. The stream tube method enables us to consider non-circulatory flows as well as pure vortex flows by using a mapped domain of computation where the mapped streamlines are parallel straight lines or concentric circles. Under certain assumptions this method permits us to compute the flow field by considering successive stream bands or rings in the computational domain, as done here for axisymmetric converging flows. Such a procedure leads to a reduction in computing time and storage area. The trust-region algorithm was used in order to determine the unknowns related to non-linear equations in the case of axisymmetric converging flows. The limit angle α for convergence indicated by the TR method is directly connected to the appearance of secondary flows. This interesting result gives consistency to our analysis and to the efficiency of the TR method.

Work will be continued towards the computation of flows of fluids obeying other constitutive equations such as integral models and the consideration of purely circulating flows such as the journal bearing geometry.

APPENDIX I

The computing TR (trust-region) iteration must involve the following steps.

I.1. Update of the trust radius

The trust radius is updated according to the following rule.
Let $0 < \mu < \eta < 1$ and $0 < \gamma_1 < \gamma_2 < 1 < \gamma_3$ be specified constants.

1. If $r_k \leq \mu$, then $\delta_{k+1} = \Delta \in [\gamma_1 \delta_k, \gamma_2 \delta_k]$.
2. If $r_k \leq \eta$, then $\delta_{k+1} \in [\gamma_2 \delta_k, \delta_k]$; else $\delta_{k+1} \in [\delta_k, \gamma_3 \delta_k]$.

I.2. Update of the iterate

In the trust-region method the updating of the iterate is usually governed by a parameter s such that $0 < s \leq 0{\cdot}25$, which must be kept constant throughout the iteration. The rule is the following.

1. If $r_k < s$, then $x_{k+1} = x_k$.
2. If $r_k \geq s$, then $x_{k+1} = x_k + d_k$.

Note that a significant decrease of the function is demanded to allow the algorithm to move from x_k to $x_k + d_k$.

I.3. Update of the quasi-Hessian H_k

In this subsection we propose two algorithms for updating H_k.
Suppose $\gamma_k = \nabla\psi(x_{k+1}) - \nabla\psi(x_k)$, $s_k = x_{k+1} - x_k$ and H_0 has been initialized as a symmetric matrix.

(*i*) *Formula of rank 1*

$$H_{k+1}=H_k+\alpha_k u_k u_k^{\mathrm{T}}, \tag{53}$$

where

$$\alpha_k=-1/s_k^{\mathrm{T}}(H_k s_k-\gamma_k), \tag{54}$$

$$u_k=H_k s_k-\gamma_k. \tag{55}$$

(*ii*) *Formula of rank 2*

$$H_{k+1}=H_k+\alpha_k u_k u_k^{\mathrm{T}}+\beta_k v_k v_k^{\mathrm{T}}, \tag{56}$$

where

$$\alpha_k=1/s_k^{\mathrm{T}}\gamma_k, \qquad u_k=\gamma_k, \tag{57}$$

$$\beta_k=-1/s_k^{T}H_k s_k, \qquad v_k=H_k s_k. \tag{58}$$

It has been proved[25–27] that if ψ is quadratic and $A=\nabla^2\psi(x)$ is positive definite, then H_k calculated by the above formula converges to $\nabla^2\psi(x^*)$.

APPENDIX II

Here our purpose is to give a complete discussion of the theoretical aspects of problem (LP) of (51):

$$\min\{\langle g, d\rangle+\tfrac{1}{2}\langle d, Hd\rangle: \|d\|\leq\delta\}. \tag{LP}$$

We begin with the following main result (which is the basis of the trust-region algorithm) due to Gay[23] and Sorensen[20] in a more simplified form.

Theorem 1

d^* is a solution to (LP) if and only if there is $\mu\geq 0$ such that

(i) $H+\mu I$ is positive semidefinite
(ii) $(H+\mu I)d^*=-g$
(iii) $\|d^*\|\leq\delta$ and $\mu(\|d^*\|-\delta)=0$.

Such a μ is unique.

Proof. It is clear that (i) and (ii) are the Kuhn–Tucker conditions which are necessary for optimality in (LP) (see e.g. Reference 28). Moreover, if A is itself positive semidefinite, then (LP) is a convex optimization problem. In this case (i) is always verified and Theorem 1 is known. Now let d^* satisfy (i) and (iii). By simple calculus using (ii) we obtain

$$q(d)=g^{\mathrm{T}}d+\tfrac{1}{2}d^{\mathrm{T}}Ad=q(d^*)-(\mu/2)[d^{\mathrm{T}}d-d^{*\mathrm{T}}d^*]+\tfrac{1}{2}(d-d^*)^{\mathrm{T}}(H+\mu I)(d-d^*). \tag{59}$$

Necessity. Let d^* be a solution to (LP); then (1) implies

$$(d-d^*)^{\mathrm{T}}(H+\mu I)(d-d^*)\geq\mu[d^{\mathrm{T}}d-d^{*\mathrm{T}}d^*] \quad \text{for } d \text{ such that } \|d\|\leq\delta. \tag{60}$$

If $d^*=0$, then $g=0$ and we have $\min\{\frac{1}{2}d^{\mathrm{T}}Hd: \|d\|\leq\delta\}=0$. In other words, H is positive semidefinite. If follows that $H+\mu I$ is positive semidefinite for all $\mu\geq 0$.

Now assume that d^* is non-null; then we have

$$(d-d^*)^{\mathrm{T}}(H+\mu I)(d-d^*)\geq 0 \quad \forall d, \qquad d^{\mathrm{T}}d=d^{*\mathrm{T}}d^*. \tag{61}$$

This is equivalent to

$$p^{\mathrm{T}}(H+\mu I)p\geq 0 \quad \forall p, \qquad p^{\mathrm{T}}d^*\neq 0. \tag{62}$$

Using the continuity of the quadratic form, we can conclude that $p^{\mathrm{T}}(H+\mu I)p\geq 0$, $\forall p$.

Sufficiency. Let d^* verify the three conditions in Theorem 1. If $\|d^*\|\leq\delta$, then (iii) implies that $\mu=0$ and d^* is a solution to (LP) according to (59). If $\|d^*\|=\delta$, then (59) implies that d^* is a solution to (LP) because $d^{\mathrm{T}}d\leq d^{*\mathrm{T}}d^*$ for every d such that $\|d\|\leq\delta$ and $H+\mu I$ is positive semidefinite.

The uniqueness of μ will be proved in Lemma 3.

As an immediate consequence of Theorem 1 we obtain the following (unusual) complete characterization of the solution to (LP).

The solution of (LP) is straightforward if (LP) has no solution on the boundary of $\{d\in\mathbb{R}^n\colon \|d\|\leq\delta\}$. In fact (LP) has no solution d with $\|d\|=\delta$ if and only if H is positive definite and $\|H^{-1}g\|<\delta$. The non-negative scalar μ in Theorem 1 is called the Lagrange multiplier associated with the constraint $\langle d, d\rangle\leq\delta^2$.

It is worth noting that (LP) represents an interesting case of a non-convex optimization problem whose complete characterization of solution can be pointed out.

Now let us define the function ϕ on $\{\mu\in\mathbb{R}\colon H+\mu I \text{ non-singular}\}$ by

$$\phi(\mu)=\|d(\mu)\|, \tag{63}$$

where $d(\mu)$ is a solution of $(H+\mu I)d=-g$.

Denote by $\lambda_1\leq\lambda_2\leq\ldots\leq\lambda_n$ the eigenvalues of H and by $u_1, u_2, \ldots, u_n$ their corresponding eigenvectors. We denote $\mathcal{N}(H-\lambda_1 I)$ the null space of $H-\lambda_1 I$ (or the eigenspace relative to λ_1) and $J_1=\{i\colon \lambda_i=\lambda_1\}$.

It is clear that the solution of problem (LP) is closely related to the non-linear equation $\phi(\mu)=\delta$ for μ in $]-\lambda_1, +\infty[$. More precisely, this is the case where (LP) has a solution on the boundary of its constraint set and there is $\mu>\max(0, -\lambda_1)$ in Theorem 1. In this case the Hebden algorithm (which is presented in Appendix III) is known to be reliable and efficient for solving $\phi(\mu)-\delta=0$.

The so-called 'hard case' corresponds to the situation where the coefficient μ in Theorem 1 must be equal to $-\lambda_1$.

Let us consider now the solution of the equation

$$(H+\mu I)d=-g \qquad \text{for } \mu\geq-\lambda_1, \tag{64}$$

using the eigensystem of H. We find that if $\mu>-\lambda_1$, then the solution to (64) is defined by

$$u_i^{\mathrm{T}}d=\frac{-u_i^{\mathrm{T}}g}{\lambda_i+\mu} \qquad \text{for } i=1,\ldots,n. \tag{65}$$

Therefore

$$\phi(\mu)=\|d(\mu)\|=\left(\sum_{i=1}^{n}\frac{(u_i^{\mathrm{T}}g)^2}{(\lambda_i+\mu)^2}\right)^{1/2}. \tag{66}$$

The function $\phi(\mu)$ is positive and strictly decreasing in $]-\lambda_1, +\infty[$, with $\lim_{\mu\to\infty}\phi(\mu)=0$; then the equation $\phi(\mu)=\delta$ has at most one solution in this interval.

Lemma 1

1. If g is perpendicular to $\mathcal{N}(H-\lambda_1 I)$, then problem (64) with $\mu \geq -\lambda_1$ always has a solution.
2. Problem (3) with $\mu = -\lambda_1$ has a solution if and only if g is perpendicular to $\mathcal{N}(H-\lambda_1 I)$. Moreover the solution set to problem (64) with $\mu = -\lambda_1$ is $d^* + \mathcal{N}(H+\mu I)$, where d^* is defined by

$$u_i^T d^* = \begin{cases} 0 & \text{if } i \in J_1, \\ -u_i^T g/(\lambda_i+\mu) & \text{otherwise.} \end{cases} \tag{67}$$

Proof. It suffices to prove statement 2. If $\mu = -\lambda_1$, then the range of $H+\mu I$ is $\mathrm{Lin}\{u_i : i \notin J_1\}$ which is equal to $\mathcal{N}(H-\lambda_1 I)^\perp$. Statement 2 is then immediate.

It can be verified that d^* is a solution to (64) with $\mu = -\lambda_1$. It follows that the solution set of problem (64) is $d^* + \mathcal{N}(H-\lambda_1 I)$. Because $d^* \in \mathcal{N}(H-\lambda_1 I)^\perp$, d^* is a solution of minimum norm to problem (64) with $\mu = -\lambda_1$:

$$\| d^* \| = \min\{ \| d \| : d \in d^* + \mathcal{N}(H-\lambda_1 I) \}.$$

In terms of the pseudo-inverse [29,30] we can write $d^* = -(H-\lambda_1 I)^+ g$.

The following results (which are a consequence of the previous ones) are helpful in understanding the algorithm described for solving (LP).

Lemma 2

Assume that the matrix H is non-null.

(1) If the vector g is different from zero, then the function $\phi(\mu)$ is positive and strictly decreasing in $]-\lambda_1, +\infty[$, with $\lim_{\mu \to \infty} \phi(\mu) = 0$.

(2) If g is not perpendicular to $\mathcal{N}(H-\lambda_1 I)$, then

$$\lim_{\substack{\mu \to \lambda_1 \\ \mu > -\lambda_1}} \phi(\mu) = +\infty. \tag{68}$$

(3) If g is perpendicular to $\mathcal{N}(H-\lambda_1 I)$, then

$$\phi(\mu) = \left(\sum_{i \notin J_1} \frac{(u_i^T g)^2}{(\lambda_i+\mu)^2} \right)^{1/2}, \tag{69}$$

$$\lim_{\substack{\mu \to \lambda_1 \\ \mu > -\lambda_1}} \phi(\mu) = \left(\sum_{i \notin J_1} \frac{(u_i^T g)^2}{(\lambda_i-\lambda_1)^2} \right)^{1/2} = \| (H-\lambda_1 I)^+ g \|. \tag{70}$$

In this case $\phi(\mu)$ can be continuously prolonged at $-\lambda_1$ by setting

$$\phi(-\lambda_1) = \| (H-\lambda_1 I)^+ g \|. \tag{71}$$

Proof. This is immediate from the expression of $\phi(\mu)$ in (66) and Lemma 1.

Remark

In the construction of an algorithm for solving (LP), Lemma 2 will be used as follows (see the most negative curvature method in Appendix III).

(1) For any fixed number $\delta>0$, if we cannot find an $\varepsilon>0$ sufficiently small (by the dichotomy technique, for example) such that $\phi(-\lambda_1+\varepsilon)\geq\delta$, then g is perpendicular to $\mathcal{N}(H-\lambda_1 I)$.
(2) If g is perpendicular to $\mathcal{N}(H-\lambda_1 I)$, then for ε' sufficiently small, $d(-\lambda_1+\varepsilon')$ and $\phi(-\lambda_1+\varepsilon')$ represent acceptable approximations of $-(H-\lambda_1 I)^+ b$ and $\|(H-\lambda_1 I)^+ g\|$ respectively.

In this way we avoid the expensive computation of $(H-\lambda_1 I)+b$ in the hard case.

Lemma 3

Suppose that $g\neq 0$ in (LP) (see Lemma 4 for the case $g=0$).

1. If $\lambda_1>0$ (i.e. H is positive definite), then
   (i) if $\|H^{-1}g\|\leq\delta$, then $d=-H^{-1}g$ is the only solution to (LP) ($\mu=0$ in Theorem 1 and $\phi(\mu)-\delta=\|d(\mu)\|-\delta\leq 0$)
   (ii) otherwise $\|H^{-1}g\|>\delta$ and $\phi(0)-\delta=\|d(0)\|-\delta>0$, therefore there is a unique solution $\mu>0$ to $\phi(\mu)=\delta$.
2. If $\lambda_1=0$ (i.e. H is only positive semidefinite), then
   (i) if $\|(H-\lambda_1 I)^+ g\|\leq\delta$ and $(H-\lambda_1 I)(H-\lambda_1 I)^+ g=g$ (this equality occurs if and only if g is perpendicular to $\mathcal{N}(H-\lambda_1 I)$ according to Lemma 1), then
   $$\{d\in\mathbb{R}^n\colon d=-(H-\lambda_1 I)^+ g+u,\quad u\in\mathcal{N}(H-\lambda_1 I)$$
   and $\|d\|^2=\|(H-\lambda_1 I)^+ g\|^2+\|u\|^2\leq\delta^2\}$ is the solution set to problem (LP) and $\mu=0$ is the only non-negative number which satisfies Theorem 1 (see Lemma 2).
   (ii) otherwise $\|(H-\lambda_1 I)^+ g\|>\delta$ or $(H-\lambda_1 I)(H-\lambda_1 I)^+ g\neq g$, then $\phi(\mu)-\delta=0$ admits the unique solution μ^* in $]-\lambda_1, +\infty[$ and μ^* is the only non-negative number which satisfies Theorem 1 (see Lemma 2).
3. If $\lambda_1<0$, then
   (i) if $\|(H-\lambda_1 I)^+ g\|\leq\delta$ and $(H-\lambda_1 I)(H-\lambda_1 I)^+ g=g$ (this equality occurs if and only if g is perpendicular to $\mathcal{N}(H-\lambda_1 I)$ according to Lemma 1), then
   $$\{d\in\mathbb{R}^n\colon d=-(H-\lambda_1 I)^+ g+u,\quad u\in\mathcal{N}(H-\lambda_1 I)$$
   and $\|d\|^2=\|(H-\lambda_1 I)^+ g\|^2+\|u\|^2=\delta^2\}$ is the solution set to problem (LP) and $\mu^*=-\lambda_1$ is the only non-negative number which satisfies Theorem 1 (see Lemma 2).
   (ii) otherwise $\|(H-\lambda_1 I)^+ g\|>\delta$ or $(H-\lambda_1 I)(H-\lambda_1 I)^+ g\neq g$, then $\phi(\mu)-\delta=0$ admits the unique solution μ^* in $]-\lambda_1, +\infty[$ and μ^* is the only non-negative number which satisfies Theorem 1 (see Lemma 2).
4. If $\lambda_n\leq 0$ (i.e. the quadratic function $q(d)$ is concave), then (LP) has only one solution on the boundary of $\{d\colon\|d\|\leq\delta\}$ unless g and H are null.[31]
   (i) if $\lambda_1=0$, then $H=0$. In this case the solution set of (LP) is
   $$\{-(g/\|g\|)\delta\}\quad\text{if } g\neq 0,\qquad \{d\colon\|d\|\leq\delta\}\quad\text{if } g=0\quad\text{(see also Lemma 4).}$$
   (ii) if $\lambda_1<0$, then we apply the same results as in statement 3.

The situation where $g=0$ in (LP) is closely related to the variational spectral theory.[29]

Lemma 4

If $g=0$ in problem (LP), then the solution set of (LP) is

$$\begin{aligned} &\{0\} && \text{if } \lambda_1>0,\\ &\{d\in\mathcal{N}(H-\lambda_1 I):\|d\|\le\delta\} && \text{if } \lambda_1=0,\\ &\{d\in\mathcal{N}(H-\lambda_1 I):\|d\|=\delta\} && \text{if } \lambda_1<0. \end{aligned}$$

Proof. We can use the results above or those concerning variational properties of the spectral theory.[29]

APPENDIX III: ALGORITHMS FOR SOLVING THE LOCAL QUADRATIC PROBLEM (LP)

III.1. Hebden algorithm

Let us describe now this algorithm for the solution of problem (LP) in the case where the non-linear equation $\phi(\mu)-\delta=0$, with

$$\phi(\mu)=\left[\sum_{i=1}^{n}\left(\frac{u_i^{\mathrm{T}}g}{\lambda_i+\mu}\right)^2\right]^{1/2}, \tag{72}$$

admits a unique root in $]-\lambda_1, +\infty[$.

Reinsch[32] and Hebden[33] observed independently that in order to solve equation (64) already presented in Appendix II, i.e.

$$(H+\mu I)d=-g \quad \text{for } \mu\ge-\lambda_1,$$

the fact which should be taken into account is that the function $\phi^2(\mu)$ is a rational function in μ with second-order poles on a subset of the negatives of the eigenvalues of H.

The Newton method, which is based on a local linear approximation to $\phi(\mu)$, is then not likely to be the best method for solving (64) because the rational structure of $\phi^2(\mu)$ is ignored. Instead, an iteration for solving (64) can be derived based upon a local rational approximation to ϕ. The iteration is obtained by requiring $\phi^*(\mu)=\gamma/(\alpha+\mu)$ to satisfy $\phi^*(\mu)=\phi(\mu)$ and $\phi^{*\prime}(\mu)=\phi'(\mu)$, where we take μ as the current approximation to the root μ^*. This approximation is then improved by solving for a $\hat{\mu}$ that satisfies $\phi^*(\hat{\mu})=\delta$. The resulting iteration is

$$\mu_{k+1}=\mu_k+\left(\frac{\phi(\mu_k)}{\phi'(\mu_k)}\right)^*\frac{\delta-\phi(\mu_k)}{\delta}. \tag{73}$$

In fact, Hebden's algorithm can be viewed as Newton's algorithm applied to the equation

$$\Omega(\mu)=\frac{1}{\delta}-\frac{1}{\phi(\mu)}=0 \quad \text{for } \mu\in]-\lambda_1, +\infty[. \tag{74}$$

The local rate of convergence of this iteration is quadratic, but the most important feature of (73) is that usually the number of iterations required to produce an acceptable approximation of μ^* is very small because the iteration is based upon the rational structure of ϕ^2. Iteration (73) can be implemented without explicit knowledge of the eigensystem of H. This important observation, which is due to Hebden,[33] makes it possible to implement (73) merely by solving a linear system with $H+\mu I$ as coefficient matrix. This is easy to see since

$$\phi(\mu)=\|d(\mu)\|, \qquad \phi'(\mu)=-[1/\phi(\mu)]d(\mu)^{\mathrm{T}}(H+\mu I)^{-1}d(\mu),$$

where $(H+\mu I)d(\mu)=-g$.

Therefore the Hebden algorithm can be described as follows.
Let $\mu_0 \geq 0$ with $H+\mu_0 I$ positive definite and $\phi(\mu_0) > \delta$.

0. $k=0, 1, 2, \ldots$ until convergence.
1. Factor $H+\mu_k I = R^T R$.
2. Solve $R^T R p = -g$.
3. Solve $R^T q = p$.
4. Let

$$\mu_{k+1} = \mu_k + \left(\frac{\|p\|}{\|q\|}\right)^2 \frac{\|p\|-\delta}{\delta}.$$

$k=k+1$ and return to step 0.

In this algorithm $R^T R$ is the Cholesky factorization of $H+\mu I$ with R upper triangular. The function Ω is convex and strictly decreasing on $]-\lambda_1, +\infty[$. This fact was discovered by Reinsch[32] and follows from the expression of $\phi(\mu)$. It implies that Newton's method started from $\mu_0 \in]-\lambda_1, +\infty[$ with $\Omega(\mu_0) > 0$ (i.e. $\phi(\mu_0)-\delta > 0$) produces a monotonically increasing sequence converging to the solution of $\phi(\mu)-\delta=0$.

III.2. The most negative curvature method

The first algorithm stated above for solving the problem (LP) is very efficient when the matrix H is positive definite. If H is indefinite or only positive semidefinite, it would be preferable to use the following strategy due to Shultz *et al.*[34,35] for the solution of problem (LP) in the hard case ($\mu = -\lambda_1$ in Theorem 1 and Lemma 2 of Appendix II).

1. Taking $\alpha \in]-\lambda_1, c \max(|\lambda_1|, \lambda_n)]$, where $c > 1$ is a constant which depends on each problem.
2. Solving $p = -(H+\alpha I)^{-1} g$.
3. If $\|p\| > \delta$, then apply the Hebden method to $H := H+\alpha I$ in order to find a direction d.
4. If $\|p\| = \delta$, then $d = p$.
5. If $\|p\| < \delta$, then $d = p + \xi u_1$, where u_1 is an eigenvector of H corresponding to λ_1 and ξ is chosen such that $\|d\| = \delta$ and $\text{sign}(\xi) = \text{sign}(u_1^T p)$.

Compared to the Hebden algorithm, the only modification introduced by the method of negative curvature is at step 5, where we should proceed as if $\mu = -\lambda_1$ in Theorem 1 and Lemma 2.

The direction d obtained by this method gives as good a decrease of the quadratic model as a direction of sufficient negative curvature and satisfies the practical Conditions 1 and 2 that will be given in Appendix IV.

APPENDIX IV: CONVERGENCE RESULTS OF TR METHOD

We recall here some well-known convergence results of the TR algorithm.[19,20,24]

Theorem 2

If the function ψ is differentiable and bounded below on $\mathbb{R}^n$, and if $\nabla\psi$ is uniformly continuous, then

$$\lim_{k \to +\infty} \|\nabla\psi(x_k)\| = 0.$$

Theorem 3

If the function ψ is twice continuously differentiable and bounded below on $\mathbb{R}^n$, and if $\nabla^2\psi$ is bounded on the level set $\{x\in\mathbb{R}^n: \psi(x)\leq\psi(x_0)\}$, then trust-region algorithms with $H_k=\nabla^2\psi(x_k)$ possess the following properties.

1. $\lim_{k\to+\infty}\|\nabla\psi(x_k)\|=0$.
2. If $\{x_k\}$ is bounded, then there is a limit point x^* with $\nabla^2\psi(x^*)$ positive semidefinite.
3. If x^* is an isolated limit point of $\{x_k\}$, then $\nabla^2\psi(x^*)$ is positive semidefinite.
4. If $\nabla^2\psi(x^*)$ is non-singular for some limit point x^* of $\{x_k\}$, then

   (a) $\nabla^2\psi(x^*)$ is positive definite
   (b) $\lim x_k=x^*$ and there exists an $\varepsilon>0$ and integer K such that $\delta_k>\varepsilon$ for all $k>K$
   (c) The convergence is superlinear.

Besides the superlinear convergence, the zero gradient and positive definiteness of the Hessian are also very important properties of the algorithm which encourage us to adapt the algorithm to our problem. Although in practice we use an approximation instead of computing the Hessian during the optimization, these properties are still present.

In the above two theorems we have an implicit assumption that every step in the TR algorithm can be exactly carried out. For example, d_k is supposed to be exactly solved. Apparently, this cannot always be true in practice. What conditions should every calculated direction satisfy in order to preserve the above properties?

Before giving the convergence conditions, we define an additional notation: let $d(g, H, \delta)$ stand for a calculated direction (depending on g, H and δ),

$$\text{pred}(g, H, \delta)=-\langle g, d(g, H, \delta)\rangle-\tfrac{1}{2}\langle d(g, H, \delta), Hd(g, H, \delta)\rangle.$$

Our conditions that a step selection strategy may satisfy are as follows.

Condition 1

There exist $c_1, s_1>0$ so that $\forall g\in\mathbb{R}^n$, $\forall H\in\mathbb{R}^{n\times n}$ symmetric and $\forall\delta>0$,

$$\text{pred}(g, H, \delta)\geq c_1\|g\|\min(\delta, s_1\|g\|/\|H\|).$$

Condition 2

There exists $c_2>0$ so that $\forall g\in\mathbb{R}^n$, $\forall H\in\mathbb{R}^{n\times n}$ symmetric and $\forall d>0$,

$$\text{pred}(g, H, \delta)\geq c_2[-\lambda_1(H)]\delta^2.$$

Here λ_1 is the smallest eigenvalue of H.

Condition 3

If the matrix H is positive definite and $\|H^{-1}g\|\leq\delta$, then $d(g, H, \delta)=-H^{-1}g$.

Theorem 4[34,35]

Let ψ: $\mathbb{R}^n\mapsto\mathbb{R}$ be twice continuously differentiable and bounded below, and let $H(x)=\nabla^2\psi(x)$ satisfy $\|H(x)\|\leq\beta_1$ for all $x\in\mathbb{R}^n$. Suppose that a practical TR algorithm is applied to $\psi(x)$, starting from $x_0\in\mathbb{R}^n$, generating a sequence $\{x_k\}$, $x_k\in\mathbb{R}^n$, $k=1, 2, \ldots$.

1. If $d(g, H, \delta)$ satisfies Condition 1 and $\|H_k\|\leq\beta_2$ for all k, then $\nabla\psi(x_k)$ converges to zero (first-order stationary point convergence).

2. If $d(g, H, \delta)$ satisfies Conditions 1 and 3, $H_k = H(x_k)$ for all k, $H(x)$ is Lipschitz continuous with constant L and x^* is a limit point of $\{x_k\}$ with $H(x^*)$ positive definite, then x_k converges quadratically to x^*.
3. If $d(g, H, \delta)$ satisfies Conditions 1 and 2, $H_k = H(x_k)$ for all k, $H(x)$ is uniformly continuous and $\{x_k\}$ converges to x^*, then $H(x^*)$ is positive semidefinite (second-order stationary point convergence, with Statement 1).

REFERENCES

1. J. R. Pearson and S. Richardson, *Computational Analysis of Polymer Processing*, Applied Science, London and New York, 1983.
2. R. Temam, *Navier–Stokes Equations*, North-Holland, Amsterdam, 1979.
3. C. M. Dafermos and J. A. Nohel, 'Energy methods for non linear hyperbolic Volterra integro differential equations', *Commun. PDE*, **4**, 219–278 (1979).
4. D. D. Joseph, M. Renardy and J. C. Saut, 'Hyperbolicity and change of type in the flow of viscoelastic fluid', *Arch. Rat. Mech. Anal.*, **87**, 213–251 (1985).
5. P. J. Roache, *Computational Fluid Dynamics*, Hermosa, Albuquerque, NM, 1972.
6. T. J. Chung, *Finite Element Analysis in Fluid Dynamics*, McGraw-Hill, New York, 1978.
7. J. S. Lee and Y. C. Fung, 'Flow in locally constricted tubes at low Reynolds numbers', *J. Appl. Mech.*, 9–16 (1970).
8. M. E. Ryan and A. Dutta, 'A finite difference simulation of extrudate swell', *Proc. 2nd World Congr. of Chemical Engineering, Vol. 6*, Montreal, 1981, pp. 277–281.
9. R. E. Smith, 'Algebraic grid generation', *Appl. Math. Comput.*, **10/11**, 137–170 (1982).
10. J. F. Thompson, F. C. Thames and C. W. Mastin, 'Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies', *J. Comput. Phys.*, **15**, 299–319 (1974).
11. J. R. Clermont, 'Sur la modélisation numérique d'écoulements plans et méridiens de fluides non newtoniens incompressibles', *C. R. Acad. Sci. Paris (Sér. II.I)*, **297**, (1983).
12. J. R. Clermont and M. E. de la Lande, 'A method for the simulation of plane or axisymmetric flows of incompressible fluids based on the concept of the stream function', *Eng. Comput.*, **3**, 339–347 (1986).
13. G. K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, Cambridge, 1967.
14. J. L. Duda and J. S. Vrentas, 'Fluid mechanics of laminar liquid jets', *Chem. Eng. Sci.*, **22**, 855, (1967).
15. K. Adachi, 'Calculation of strain histories in Protean coordinate systems', *Rheol. Acta*, **22**, 326 (1983).
16. A. C. Papanastasiou, L. E. Scriven and C. W. Macosco, 'A finite element method for liquid with memory', *J. Non-Newtonian Fluid Mech.*, **22**, 271 (1986).
17. C. Truesdell and W. Noll, 'The non linear theories of mechanics', *Handb. Phys. III*, **3**, (1965).
18. J. R. Clermont and M. E. de la Lande, 'A computational method for flows of incompressible fluids based on a non-conformal transformation of the physical domain', *Mech. Res Commun.*, **13**, 239–246 (1986).
19. J. J. Moré, 'Recent developments in algorithm and software for trust region methods', *Mathematical Programming, The State of the Art*, Springer, Berlin, 1983, pp. 258–287.
20. D. C. Sorensen, 'Newton's method with a model trust region modification', *SIAM J. Numer. Anal.*, **19**, 409–426 (1982).
21. Pham Dinh Tao, S. Wang and A. Yassine, 'Training multi-layered neural network with a Trust-Region based algorithm', *Math. Modell. Numer. Anal.*, **24** (4), 523–553 (1990).
22. A. Yassine, 'Etudes adaptatives et comparatives de certains algorithmes en optimisation. Implémentations effectives et applications', *Doctoral Thesis*, Applied Mathematics Department, Joseph Fourier University, 1989.
23. D. M. Gay, 'Computing optimal constrained steps', *SIAM J. Sci. Stat. Comput.*, **2**, 186–197 (1981).
24. J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
25. M. Minoux, *Programmation Mathématique, Tome 1*, Dunod, Paris, 1983.
26. P. E. Gill, W. Murray and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.
27. R. Fletcher, *Practical Methods of Optimization, Vol. 1*, Wiley, New York, 1980.
28. D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1972.
29. P. Lancaster, *Theory of Matrix*, Academic Press, New York and London, 1969.
30. G. W. Stewart, *Introduction to Matrix Computation*, Academic Press, New York, 1973.
31. R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
32. H. C. Reinsch, 'Smoothing by spline functions II', *Numer. Math.*, **16**, 451–454 (1971).
33. M. D. Hebden, 'An algorithm for minimization using exact second derivatives', *Report TP515*, Atomic Energy Research Establishment, Harwell, 1973.
34. G. A. Shultz, R. B. Schnabel and R. H. Byrd, 'A family of trust region-based algorithms for unconstrained minimization with strong global convergence properties', *SIAM J. Numer. Anal.*, **22**, 47–67 (1985).
35. G. A. Shultz, R. B. Schnabel and R. H. Byrd, 'Approximate solution of the trust-region problem by minimization over two-dimensional subspaces', *Math. Program.*, **40**, 247–263 (1988).