An Algorithm for the Resolution of Conduction-Convection Boundary Layers

R. Zanino, K. Lackner, W. Schneider

IPP 5/18 February 1988

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

8046 GARCHING BEI MÜNCHEN

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MÜNCHEN

An Algorithm for the Resolution of Conduction-Convection Boundary Layers

R. Zanino, K. Lackner, W. Schneider

IPP 5/18 February 1988

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

An Algorithm for the Resolution of Conduction-Convection Boundary Layers

R. Zanino, K. Lackner and W. Schneider

Max-Planck Institut für Plasmaphysik, EURATOM Association
D 8046 Garching, Federal Republic of Germany

ABSTRACT

A finite difference time-dependent algorithm for the steady-state solution of conduction-convection problems is discussed. The scheme is applied to a boundary layer problem; in the region where steep gradients occur a higher grid resolution is used. The accumulated mesh is mapped to an equidistant grid by a suitable space-transformation. A corresponding time-transformation has the potential to reduce the computational effort.

CONTENTS

1. Introduction

The numerical treatment of boundary layer problems, where in part of the region steep gradients can occur, usually requires an accumulation of gridpoints. This paper describes a method for achieving this and discusses a scheme to reduce the computing time for obtaining the steady state solution.

We adopt as a model the conduction-convection equation for A(y,t), $y \in [0,1]$,

$$\frac{\partial}{\partial t}\mathrm{A}(\mathrm{y},\mathrm{t}) = \frac{\partial}{\partial \mathrm{y}}(\kappa(\mathrm{A})\frac{\partial \mathrm{A}}{\partial \mathrm{y}}) - \mathrm{v}(\mathrm{y})\frac{\partial \mathrm{A}}{\partial \mathrm{y}} + \mathrm{S}(\mathrm{y}), \qquad (1)$$

with appropriate initial- and boundary-conditions. κ , v and S are given functions of the spatial variable y (or -in the nonlinear case- functions of y and A); all quantities are assumed to be dimensionless. In particular we are interested in the steady state solution of eq.(1) for the case $\mathrm{v} >> \kappa$; this limit is known as the *convection dominated* case.

The above equation (1) models a variety of physical problems; it constitutes a general Eulerian description of the diffusive transport of any fluid property A; in that case κ is the diffusion coefficient, v is the flow velocity field by which A is convected, and S is a source term.

Most previous studies of non-linear cases of eq.(1) have been restricted to nonlinearites in the first derivative term (e.g. v(A)=A leads to Burger's equation). We are however interested in the application of eq.(1) to the description of heat transport in a plasma; in that case A is the temperature, κ is the heat conductivity. Along field lines, in a classical (collision dominated) plasma κ is proportional to $A^{5/2}$; hence we will restrict ourselves in the following to this kind of power nonlinearities in the highest derivative term.

Numerical difficulties arise when convection dominates conduction, since then very steep gradients can originate. To qualitatively see why, we can compare in eq.(1) the convective term with the conductive one, by introducing the (mesh) *Péclet* number

$$\mathrm{Pe} = \frac{\mathrm{Convection}}{\mathrm{Conduction}} \sim \frac{\mathrm{v} * \Delta \mathrm{y}}{\kappa} \quad .$$

Even in the convection dominated case there nevertheless exists a small space scale $\epsilon \sim \frac{\kappa}{\mathrm{v}} \ll 1$ over which conduction is important. Physically, if $\mathrm{v} \gg \kappa > 0$, the left boundary condition is convected through almost all the domain, and the solution is affected by the right boundary condition only in the region $[1-\epsilon, 1]$. Depending also on the boundary conditions imposed to the problem, steeper gradients than elsewhere in $[0,1]$ can arise in that region, since the curvature of the solution is relatively large there.

Considering a finite difference approximation of equation (1), and restricting ourselves for the moment to the linear case, it appears as a result of many investigations, that a key role is played by the way in which the first derivative term is discretized. One may be faced in this respect with two different practical problems: either only a reasonable approximation of the hyperbolic part of the solution is required, then also Pe >> 1 is sufficient, or there is interest in the resolution of the boundary layer, so that Pe $\approx$ O(1) is required. Furthermore, it is clear that the steady state of eq.(1) may be found either by directly (or iteratively [1]) solving the steady state equivalent of eq.(1), or by reaching it, letting an initial condition evolve in time. In the latter case, either explicit or implicit time integration can be used. The direct solution and the implicit time-evolutionary approach both require a matrix inversion; they are equivalent as far as the following discussion is concerned.

When Pe >> 1 the (first order) upwind approximation of the first derivative term in eq.(1) is reported [2]to give a more accurate solution than the (second order) central difference for the direct approach; and, with the second order upwind a steady state is reached in a smaller number of iterations than with the central differences [3]. Also for the time-dependent approach one would prefer using upwind differences, since there is a weaker stability constraint [2]on the time step by using upwind than by using central differences.

When Pe $\approx$ 1, on the contrary, the situation reverses and it is advisable to use central differences: they give more accurate solutions than the (first order) upwind, do not give rise to spatial oscillations as in case of Pe >> 1, and, they have for the time-dependent approach a weaker stability constraint than the upwind [2]. Higher order upwind schemes are also used [3]. If a very high accuracy (say fourth order) is required, then so called compact implicit schemes [4]have been developed, which use a computational molecule smaller than the corresponding standard finite differences. We have chosen to use a weighted central/upwind scheme, the weight depending on the mesh Péclet number.

It is our purpose to show in the following how to determine a steady state solution of eq.(1) by using a finite difference method both in time and space, which be reasonably (second order) accurate, as to resolve the boundary layer, and fast.

1) Provided v(y) and S(y) do not vary on a scale-length comparable with the mesh interval (cases when this requirement is not satisfied are reported in [3]) a relatively coarse space discretization is sufficient over most of the domain, but the mesh must then be necessarily non-uniform in order to resolve a layer of thickness $O(\epsilon\ )$. A non-uniform grid is not only uncomfortable but usually also lowers the accuracy of the finite difference formulas. Therefore we choose a suitable space transformation, described in section 2.1, which maps equation (1) to a new equation in a transformed coordinate x that can be discretized uniformly.

2) If an explicit scheme is chosen for time integration, the maximum Δt is generally constrained - due to stability requirements [6]- by the minimum Δy; for instance

$$\Delta t_{\max} \sim \frac{\Delta y^2_{\min}}{\kappa_{\max}} \sim \frac{\epsilon^2}{\kappa_{\max}}$$

so that the explicit time integration of eq.(1) can become very expensive when $\frac{v}{\kappa} \sim \frac{1}{\epsilon} \gg 1$. In section 2.2 we therefore construct a suitable time transformation such that the numerical convergence is speeded up for both the explicit and implicit time integration scheme, provided that one be only interested in the steady state solution of eq.(1), and not in reproducing the transient.

In section 3 the numerical schemes and the boundary conditions will be discussed. After developing the method and checking the results against analytical steady state solutions for the linear form of eq.(1) the algorithm is applied to two different non-linear cases (section 4).

2. Space- and Time- Coordinate Transformation

2.1. Space Transformation

In order to resolve the steep gradients of A(y) arising in the vicinity of a point y_0, one has to accumulate gridpoints around y_0. For instance, in the convection dominated case of eq.(1) this occurs, for $v > 0$, near the right boundary ($y_0 \approx 1$); i.e. $\Delta y(y)$ has to be much smaller near $y_0 = 1$ than for the left part of the domain. To achieve this mesh accumulation, and still have the benefit of an equidistant grid for the numerical treatment, one can map the non-uniform mesh y to an equidistant mesh x, ($\Delta x = \text{const}(x)$), by

$$y = y(x)$$

For the transformation of eq.(1) into the new coordinate frame we express the derivatives according to

$$\frac{\partial}{\partial y} = F(x)\frac{\partial}{\partial x} \quad \text{where} \quad F(x) \equiv \frac{1}{\frac{\partial y}{\partial x}}.$$

The space-transformed eq.(1) then reads

$$\frac{\partial}{\partial t}A(x,t) = F\frac{\partial}{\partial x}(\kappa(A)F\frac{\partial A}{\partial x}) - v(x)F\frac{\partial A}{\partial x} + S(x). \tag{2}$$

In the examples discussed later, we have used the following spatial transformation

$$y(x) = \frac{1}{\sin(\frac{\varsigma\pi}{2})}\sin(\frac{\varsigma\pi}{2}x).$$

(A less general transformation (with $\varsigma = 1$) is used e.g. in the NAG routine D03PBF [6], devoted to the solution of problems similar to eq.(1).) With our choice of y(x) we get for the spatial mapping function F(x)

$$F(x) = \frac{\sin(\frac{\varsigma\pi}{2})}{(\frac{\varsigma\pi}{2})}\frac{1}{\cos(\frac{\varsigma\pi}{2}x)}.$$

The particular function y(x) we selected and the corresponding F(x) and $\Delta y(x)$ are plotted in fig.1 for some values of the parameter ς (fig.1a: ς = .25, fig.1b: $\varsigma = .999$).

A space coordinate transformation cannot obviously lead by itself alone to a faster algorithm. If one would solve (2) explicitely, the formal conservative requirement on the maximum Δt allowed would be for an Euler scheme

$$\Delta t \simeq \frac{\Delta x^2}{\kappa_{max} F_{max}^2}$$

which can as before result in an unreasonable constraint when $\frac{v}{\kappa} \gg 1$, i.e. when the layer to be resolved becomes very thin. In that case, as it can be seen in fig.1, $\varsigma \to 1$ is required, so that $F_{max} \sim \tan(\varsigma \frac{\pi}{2}) \to \infty$.

2.2. TIME TRANSFORMATION

When only the steady state solution of eq.(1) is required, a time-transformation $t = t(\tau)$ can be employed in order to relax the mentioned constraint on the timestep Δt. If we define a new "time" coordinate τ by means of

$$t = \frac{1}{G(x)}\tau$$

then the relationship between the corresponding time-derivative operators is

$$\frac{\partial}{\partial t} = G(x)\frac{\partial}{\partial \tau}$$

Applying this time-transformation to eq.(2) leads to the space- and time-transformed formulation of eq.(1):

$$G(x)\frac{\partial}{\partial \tau}A(x,\tau) = F\frac{\partial}{\partial x}(\kappa(A)F\frac{\partial A}{\partial x}) - v(x)F\frac{\partial A}{\partial x} + S(x), \qquad (3)$$

The formal stability condition arising from the diffusion term in case of an explicit difference scheme is

$$\Delta\tau_d \leq \frac{\Delta x^2}{(\frac{F^2}{G})\kappa}$$

If the space-dependent function G(x) is chosen to be

$$G(x) = F^2(x),$$

then the diffusive timestep limit is

$$\Delta\tau \leq \frac{\Delta x^2}{\kappa_{max}},$$

i.e. a factor of F^2_{max} larger than for the original problem eq.(1), or the space-transformed formulation eq.(2).

The time τ as a function of x is plotted also in fig.1 for two different values of ς; $\varsigma = .25$ (fig.1a), $\varsigma = .999$ (fig.1b).

Some remarks concerning the time transformation (3) are due:

- The artificial time τ depends on x ($\tau(x,t) = G(x) * t$), but the timestep $\Delta\tau$ doesn't ($\Delta\tau = \mathrm{const}(x)$) . If we imagine to take a picture of the system after M artificial time steps, i.e. at $\tau = M * \Delta\tau$, then, because $\Delta t = \Delta t(x) = \frac{1}{G(x)}\Delta\tau$, one has

$$t_1 = M * \Delta t(1) << t_0 = M * \Delta t(0).$$

The function Δt vs. x is shown in fig.1 for two different values of ς.

- The last relation makes clear that marching the system forward in τ, as we do in the numerical solution, formally corresponds to having the real time t flowing slower in the region of steep gradients, making it easier for the system to deal with the stronger transport and mixing there present. This kind of compensation in some sense evens out the different behaviour of the solution A for x far away or near to the right boundary, and ultimately explains why the combination of space- and time-transformation allows both good resolution and affordable "time"-stepsize.
- On the other hand it is also clear from the same relation that one cannot use this time-transformation technique when the details of the time transient are important, since it is not the true time transient that is being followed.

3. Methods of Solution and Boundary Conditions

For the solution of the previously described partial differential equations we have followed several approaches:

- An analytical solution for the stationary, linear case is derived in Appendix A, and has been used as a check of the numerical solution; the solution for a very particular non-linear case, to be used with the same purpose, is also reported in the same Appendix.
- As the basic numerical method of solution, a standard time dependent, Finite Difference (FD) scheme, described in Appendix B, has been used. The scheme is second order accurate in space but only first order in time, since we are only interested in the steady state solution. Both explicit and implicit time integration has been considered within the so called θ -scheme. In the non-linear cases a linearization procedure has been used, also described in Appendix B.

For the purpose of comparing efficiency and accuracy two other approaches have been considered:

- First and second order Lagrange, and, cubic Hermite Finite Element (FE) methods, described in Appendix C, to get a solution of the stationary equivalent of eq. (2) using a mesh accumulation according to the mapping function F(x).
- A Chebyshev Collocation (CC) method, discussed in Appendix D, to solve eq.(1); this method is particularly suited for problems where high resolution near the boundary is required, being the Chebyshev mesh "naturally" denser there.

The specific choice of boundary conditions can have an important role in determining the features of the solution. We have considered two types of boundary conditions; first

$$\text{BC I:} \qquad \frac{\partial}{\partial y}A(y=0)=0, \qquad A(y=1)=AR$$

have been chosen since they are typically used in the modeling of the scrape-off layer plasma; in that case the left boundary corresponds to the midplane of the tokamak, where a symmetry condition can be imposed, whereas the right boundary is practically set at the (limiter) divertor plate. However, these BCs lead to amplitudes of the steady state solution growing exponentially with $\frac{|v|}{\kappa}$ at the left boundary, so that the case of very thin boundary layers cannot be treated conveniently, if an explicit scheme is used.

Therefore in the following discussion we will use Dirichlet conditions:

$$\text{BC II:} \qquad A(y=0) = AL, \qquad A(y=1) = AR.$$

4. Results

A number of different cases has been studied in order to assess the usefulness of the method described above. The linear, constant coefficients cases are used to develop the algorithms and compare the results with the stationary analytical solutions. In all cases the Dirichlet boundary conditions

$$A(y=0) = AL = 2, \qquad A(y=1) = AR = 1$$

have been applied.

4.1. Linear Cases:

The steady state FD solutions A(y) and A(x) of eqs.(2) or (3) respectively, for v=1, S=1, $\kappa = .01$, are shown in fig.3 as functions of y and x. The representation of the solution in terms of x exhibits the smoother form of A(x) due to the spatial transformation. In fig.4 the solution obtained by the Chebyshev technique is shown for different numbers of points.

To check the accuracy and convergence of the different approaches, the solutions of eq.(3), corresponding to a different number of mesh points, are compared with the analytical solution $A_a(y)$:

$$\text{Err} := \max_{y\in[0,1]} \left\{ \frac{A(y) - A_a(y)}{A_a(y)} \right\}.$$

This relative deviation from the analytical solution is illustrated in fig.4 for the Chebyshev method. The convergence (i.e. the maxumum deviation from the analytical solution) of several FD schemes as $\Delta x \to 0$ is plotted in fig.5a ($\varsigma = 0.1$ equidistant grid) and fig.5b ($\varsigma = .999$ accumulated mesh). The dotted curves are obtained from first order upwind differences, the dashed curves are from central differences. For the weighted scheme (solid lines) a second order convergence is obtained for both cases $\varsigma = 0.1$ and $\varsigma = .999$.

The absolute improvement of accuracy due to the space-transformation is illustrated in fig.6 for several difference schemes: the broken lines are for approximately equidistant ($\varsigma = 0.1$), the solid lines are for $\varsigma = .999$.

Plots of Err vs. Δx are also shown in figs.7a (Lagrange Elements; $\varsigma = .100$ solid line; $\varsigma = .999$ dashed line; Lagrange FE of second order, $\varsigma = .999$ dotted line) and fig.7b

(Cubic Hermite Elements), for the solutions obtained using Lagrange and Cubic Hermite FE respectively. In fig.7c the results are plotted for the case in which the Chebyshev collocation method is applied. In this last case the remarkable property of exponential convergence is apparently verified.

4.2. Non-Linear Cases:

Two types of nonlinear conduction terms have been studied:

$$\text{(NL1)} \qquad \kappa(A)\frac{\partial}{\partial y}\left(\frac{\partial A}{\partial y}\right),$$

and

$$\text{(NL2)} \qquad \frac{\partial}{\partial y}\left(\kappa(A)\frac{\partial A}{\partial y}\right),$$

where $\kappa(A) = \kappa_0 A^{\alpha}; \kappa_0 = .001, \alpha = \frac{5}{2}$; both cases have been solved using the implicit scheme. The FD solutions of eq.(3) for these two cases are plotted in figs.8a,b (NL1) and figs.9a,b (NL2).

In order to check the improvements due to the time transformation, the number of time-steps necessary to reach the steady state solution is plotted vs. the (normalized) time-step size in figs.10 and 11, for some different cases:

In fig.10a the results for the case (NL1) are shown; the solid curves represent the results of the space-transformed eq.(2); the dashed curves demonstrate the gain in computing effort when the time-transformed scheme (eq.(3)) is employed. The same result still holds, with a smaller gain, for the case (NL2), as it can be seen from fig.11a.

When the linearization procedure is applied to the non-linear conduction term, the same conclusions of the non-linearized case remain valid. The corresponding results are shown in figs.10b and 11b.

5. Conclusions

The possibility to reduce the computational effort required to obtain the time asymptotic, steady state solution of a conduction- convection, boundary layer problem has been investigated. A space dependent transformation of the time coordinate is employed to this purpose.

- **For the linear problem a substantial advantage in terms of CPU time can be gained with the time transformation when an explicit difference scheme is used; not even the time transformed explicit scheme, however, reaches the effectiveness of either the standard or the time transformed implicit algorithm, and these two have the same performance.**
- **For the case when a nonlinear conductive term is considered, the time transformation significantly improves the performance of both the explicit and the implicit scheme, against their standard counterparts. This gain tends to grow as the non-linearity increases.**

A rather elegant procedure to accumulate meshpoints in regions of steep gradients by means of a spatial mapping to an equidistant grid has also been documented.

Appendix A: Analytical Solutions

The time-independent spatial differential equation for $A(y)$, $y \in [0,1]$,

$$\frac{d}{dy}(\kappa(A)\frac{dA}{dy}) - v(y)\frac{dA}{dy} + S(y) = 0$$

can be solved analytically, if κ, v, and, S are const(y). For the homogeneous part of this equation

$$\kappa A_{yy} - vA_y = 0$$

the Ansatz $A = e^{\lambda y}$ leads to the characteristic equation

$$(\kappa\lambda^2 - v\lambda)e^{\lambda y} = 0$$

i.e. the homogeneous solution is:

$$A_{homo}(y) = C_1 + C_2 e^{\lambda y}, \qquad \text{with} \qquad \lambda = \frac{v}{\kappa}.$$

A particular solution of the inhomogeneous equation can be found as:

$$A_{part}(y) = \sigma y, \qquad \text{with} \qquad \sigma = \frac{S}{v}.$$

Hence the general solution (type I) is:

$$A(y) = C_1 + C_2 e^{\lambda y} + \sigma y$$

BC I: $\frac{d}{dy}A(y=0) = 0, \qquad A(y=1) = AR;$

these boundary conditions lead to the following integration constants:

$$C_1 = AR + \frac{\sigma}{\lambda}e^{\lambda} - \sigma, \qquad C_2 = -\frac{\sigma}{\lambda}.$$

BC II: $A(y=0) = AL, \qquad A(y=1) = AR;$

for these boundary conditions the integration constants are:

$$C_1 = \frac{AR - \sigma - ALe^{\lambda}}{1 - e^{\lambda}}, \qquad C_2 = \frac{AL + \sigma - AR}{1 - e^{\lambda}}.$$

==

For the special case of v(y)= 0, i.e. $\kappa A_{yy} + S = 0$, one can integrate by quadrature and get the solution (type II):

$$A(y) = C_1 - C_2 y - \frac{S}{2\kappa} y^2.$$

BC I: $\frac{d}{dy} A(y=0) = 0, \qquad A(y=1) = AR;$

$$C_1 = AR + \frac{S}{2\kappa}, \qquad C_2 = 0.$$

BC II: $A(y=0) = AL, \qquad A(y=1) = AR;$

$$C_1 = AL, \qquad C_2 = AL - AR - \frac{S}{2\kappa}.$$

==

For the nonlinear equation

$$(\kappa(A) A_y)_y - v(y) A_y + S(y) = 0$$

where $\kappa(A)$ and v(y) may be arbitrary functions (e.g. $\kappa(A) = \kappa_0 A^\alpha$), a special analytical solution can be found by prescribing the source term S(y) as:

$$S(y) ::= -(\kappa(A) A_y)_y + v(y) A_y.$$

Hence the solution (type III) can be:

$$A(y) = A_0 + A_1 y + A_2 y^2,$$

where for BC II: $A(y=0) = AL, \qquad A(y=1) = AR;$ the integration constants are:

$$A_0 = AL, \qquad A_1 = AL - AR, \qquad A_2 = 2(AR - AL).$$

==

The time-independent form of (1), in the linear, variable coefficients case reads

$$(\kappa(y)A_y)_y - v(y)A_y + S(y) = 0.$$

This equation can be reduced to a linear, first order equation, and thus be analytically solved. Defining

$$p(y) \equiv \frac{v(y) - \kappa_y(y)}{\kappa(y)} \qquad \text{and} \qquad q(y) \equiv \frac{S(y)}{\kappa(y)}$$

the above ODE can be rewritten as

$$(A_y)_y - p(y)A_y + q(y) = 0.$$

The general solution is

$$A(y) = \int dy\{e^{\int p(y)dy}[-\int q(y)e^{-\int p(y)dy} + B_1]\} + B_2$$

with the integration constants B_1 and B_2 to be determined by the boundary conditions.

==

Appendix B: Finite Difference Approximation

For the solution of eq.(2)

$$\frac{\partial}{\partial t}A(x,t) = F\frac{\partial}{\partial x}(\kappa(A)F\frac{\partial A}{\partial x}) - v(x)F\frac{\partial A}{\partial x} + S(x)$$

by Finite Difference methods, some details are written down here:

The dependent variable A is defined at the center of cells j: A_j at meshpoints $X1_j$ (see fig. 2). For the time-coordinate (i.e. when advancing from time t by a timestep δt to $t + \delta t$) the following nomenclature is used:

$$\hat{t} \equiv t + \delta t$$

$$\hat{A}_j \equiv A_j(t + \delta t)$$

$$A_j \equiv A_j(t)$$

The time derivative is hence discretized as follows:

$$\frac{\partial}{\partial t}A \Longrightarrow \frac{\hat{A}_j - A_j}{\Delta t}$$

The time average $\tilde{A}_j$, governed by the impliciteness factor θ, is defined as:

$$\tilde{A}_j = \theta\hat{A}_j + (1-\theta)A_j$$

which is applied to all terms but the time-derivative. This time averaging procedure is performed before the spatial averaging and derivatives are processed. For $\theta = 0$ one gets the explicit, for $\theta = 1$ the fully implicit scheme.

In the spatial coordinate, where two staggered grids $[X1_j, X2_j]$ are used, the quantities S_j and v_j are defined at cell centers $X1_j$, whereas the quantity κ_j is more naturally defined at the cell boundaries $X2_j$ (see fig. 2). The mapping function F is defined both at cell centers $F1(X1_j)$, and at cell boundaries $F2(X2_j)$. The spatial average of A (i.e. A at $X2_j$) is usually:

$$\langle\tilde{A}_{j+1/2}\rangle := \frac{1}{2}(\tilde{A}_j + \tilde{A}_{j+1})$$

For the spatial derivatives the following standard second order scheme is applied:

$$\frac{\partial}{\partial x}A \Longrightarrow \frac{\langle\tilde{A}_{j+1/2}\rangle - \langle\tilde{A}_{j-1/2}\rangle}{\Delta x}$$

$$\frac{\partial^2}{\partial x^2} A \Longrightarrow \frac{\tilde{A}_{j+1} - 2\tilde{A}_j + \tilde{A}_{j-1}}{\Delta x^2}$$

In general the diffusion term $F\frac{\partial}{\partial x}(\kappa(A)F\frac{\partial A}{\partial x})$ is approximated as:

$$\frac{F1_j}{\Delta x}\Big[\kappa_j F2_j \frac{\tilde{A}_{j+1} - \tilde{A}_j}{\Delta x} - \kappa_{j-1} F2_{j-1} \frac{\tilde{A}_j - \tilde{A}_{j-1}}{\Delta x}\Big]$$

When $\kappa = \kappa(A)$, then κ_j could only be evaluated at the previous time t, i.e. $\kappa_j(A_j)$. A better approach is to apply the following linearization procedure: the above PDE can be written as

$$\frac{\partial A}{\partial t} = D(U_\mu)$$

where $U_\mu = \{A(x), \frac{\partial A}{\partial x}\}$. When advancing from $t \longrightarrow \hat{t} = t + \delta t$, one expands $D(U_\mu)$ into:

$$\hat{D} \approx D + \frac{\partial D}{\partial t}\Delta t = D + \frac{\partial D}{\partial U_\mu}\frac{\partial U_\mu}{\partial t}\Delta t$$

i.e.

$$\hat{D} \approx D + \frac{\partial D}{\partial U_\mu}(\hat{U}_\mu - U_\mu)$$

In order to get the time averaged expression, the θ -weighting has to be applied: $\tilde{D} = \theta\hat{D} + (1-\theta)D$, hence:

$$\tilde{D} \approx \theta\Big\{D + \frac{\partial D}{\partial U_\mu}(\hat{U}_\mu - U_\mu)\Big\} + (1-\theta)D$$

or

$$\tilde{D} \approx \theta\frac{\partial D}{\partial U_\mu}\hat{U}_\mu + \Big(D - \theta\frac{\partial D}{\partial U_\mu}U_\mu\Big)$$

For a conductive term of the form $\frac{\partial}{\partial x}(\kappa(A)\frac{\partial A}{\partial x})$, where $\kappa(A) = \kappa_0 A^\alpha$, this linearization technique leads to:

$$\frac{\partial}{\partial x}(\kappa_0\widetilde{A^\alpha}\frac{\partial}{\partial x}A) \approx \theta\alpha\frac{\partial}{\partial x}(\kappa_0 A^{\alpha-1}\hat{A}\frac{\partial}{\partial x}A) + \theta\frac{\partial}{\partial x}(\kappa_0 A^\alpha\frac{\partial}{\partial x}\hat{A})$$
$$+\{1-\theta(\alpha+1)\}\frac{\partial}{\partial x}(\kappa_0 A^\alpha\frac{\partial}{\partial x}A)$$

For the convection term $v\frac{\partial A}{\partial y}$, in case $\frac{|v|\Delta y}{\kappa} \gg 1$, usually an upwind difference approximation is applied. The standard upwind scheme is only first order accurate in space and therefore we use instead a weighted upwind method, described in the following.

We define

$$w_{j+1/2} \equiv \frac{1}{2}\frac{v_{j+1/2}\Delta y}{(|\,\kappa_{j+1/2}\,| + |\,v_{j+1/2}\Delta y\,|)}$$

and consequently

$$wU_{j+1/2} \equiv +(w_{j+1/2} + \frac{1}{2}),$$

$$wD_{j+1/2} \equiv -(w_{j+1/2} - \frac{1}{2}),$$

to calculate the space average as:

$$\langle \tilde{A}_{j+1/2} \rangle = wU_{j+1/2}\tilde{A}_j + wD_{j+1/2}\tilde{A}_{j+1}$$

This leads for $v > 0$ to the following extreme cases:

for $\frac{|v_{j+1/2}|\Delta y}{|\kappa_{j+1/2}|} \gg 1 \quad \Longrightarrow w_{j+1/2} \approx \frac{1}{2} \quad \Longrightarrow wU_{j+1/2} \approx 1, wD_{j+1/2} \approx 0;$

hence $\quad \langle \tilde{A}_{j+1/2} \rangle \approx \tilde{A}_j$ ...UPWIND $-$ differences

for $\frac{|v_{j+1/2}|\Delta y}{|\kappa_{j+1/2}|} \ll 1 \quad \Longrightarrow w_{j+1/2} \approx 0 \quad \Longrightarrow wU_{j+1/2} \approx \frac{1}{2}, wD_{j+1/2} \approx \frac{1}{2};$

hence $\quad \langle \tilde{A}_{j+1/2} \rangle \approx \frac{1}{2}(\tilde{A}_j + \tilde{A}_{j+1})$ ...CENTRAL $-$ differences

When the velocity v flows to the left $(v < 0)$, we get similarly:

for $\frac{|v_{j+1/2}|\Delta y}{|\kappa_{j+1/2}|} \ll 1 \quad \Longrightarrow w_{j+1/2} \approx 0 \quad \Longrightarrow wU_{j+1/2} \approx \frac{1}{2}, wD_{j+1/2} \approx \frac{1}{2};$

hence $\quad \langle \tilde{A}_{j+1/2} \rangle \approx \frac{1}{2}(\tilde{A}_j + \tilde{A}_{j+1})$ ...CENTRAL $-$ differences

for $\frac{|v_{j+1/2}|\Delta y}{|\kappa_{j+1/2}|} \gg 1 \quad \Longrightarrow w_{j+1/2} \approx -\frac{1}{2} \quad \Longrightarrow wU_{j+1/2} \approx 0, wD_{j+1/2} \approx 1;$

hence $\quad \langle \tilde{A}_{j+1/2} \rangle \approx \quad \tilde{A}_{j+1}$ ...DOWNWIND $-$ differences

For conduction dominated cases we thus recover the second order accurate, central difference scheme; whereas for cases where convection becomes more important, the described formalism tends towards upwind differences.

Appendix C: Finite Element Method

Part I: General

The treatment of the second order, two-point boundary value problem

$$\frac{d}{dy}(\kappa(A)\frac{dA}{dy}) - v(y)\frac{dA}{dy} + S(y) = 0 \tag{5.4}$$

for $A(y)$, $y \in [a,b] = G$, with Dirichlet boundary conditions (BC II)

$$A(y=a) = \gamma_a, \qquad A(y=b) = \gamma_b$$

in terms of Finite Elements requires the following steps:

---

(1) Multiplication of the above equation with a testfunction $T(y) \in \tilde{H}$ ($\tilde{H}$ a suitable class of test functions), and integration over G:

$$\int_G dy\Big(T\frac{d}{dy}(\kappa\frac{dA}{dy}) + Tv\frac{dA}{dy}\Big) + \int_G dy\Big(TS\Big) = 0. \tag{5.5}$$

Using the identity

$$T\frac{d}{dy}(\kappa\frac{dA}{dy}) = \frac{d}{dy}(T\kappa\frac{dA}{dy}) - \frac{dT}{dy}\kappa\frac{dA}{dy}$$

reduces the second order integrand to first order

$$\int_G dy(-\kappa\frac{dA}{dy}\frac{dT}{dy} + v\frac{dA}{dy}T) = -(\kappa\frac{dA}{dy}T)\Big|_a^b - \int_G dy(ST). \tag{5.6}$$

---

(2) Galerkin representation of a function $u(y)$:

$$u(y) = \sum_{\mu=1}^{N} u^\mu \Phi_\mu(y),$$

where the $\{\Phi_\mu(y)\}_{\mu=1,N}$ are the basis functions. Using this Galerkin representation for both the dependent function $A(y)$ and the testfunction $T(y)$, we get:

$$a^\mu t^\nu\Big\{\int_G dy(-\kappa\frac{d\Phi_\mu}{dy}\frac{d\Phi_\nu}{dy} + v\frac{d\Phi_\mu}{dy}\Phi_\nu)\Big\} = t^\nu\Big\{-(\kappa\frac{dA}{dy}\Phi_\nu)\Big|_a^b - \int_G dyS\Phi_\nu\Big\}. \tag{5.7}$$

(3) As this expression must hold for all test functions T, all its partial derivatives with respect to $\{t^\nu\}_{\nu=1,N}$ must vanish. Hence a solution of (4) under general boundary conditions is given by

$$A(y) = a^\nu \Phi_\nu(y) \tag{5.8}$$

where the $\{a^\nu\}_{\nu=1,N}$ are determined by the linear system of equations

$$\mathbf{M}_{\mu\nu} a^\nu = b_\mu \tag{5.9}$$

The solution of this system can be calculated by several algorithms, depending on the form of the matrix $\mathbf{M}_{\mu\nu}$, which is determined by the choice of the basis functions (see part II).

(4) For Dirichlet boundary conditions (BC II) the values of $A|\partial G$, (i.e. $A(a)$, $A(b)$) are given. For a subset of test functions $T \in \tilde{H}_0$ vanishing on ∂G

$$\tilde{H}_0 = \left\{T \in \tilde{H} : T|\partial G = 0\right\} \subset \tilde{H} \tag{5.10}$$

the first term on the RHS of (7) vanishes; hence:

$$a^\mu t^\nu \Big\{\int\limits_G dy(-\kappa \frac{d\Phi_\mu}{dy}\frac{d\Phi_\nu}{dy} + v\frac{d\Phi_\mu}{dy}\Phi_\nu)\Big\} = -t^\nu \Big\{\int\limits_G dy S \Phi_\nu\Big\}. \tag{5.11}$$

==

Part II:

Basically two different types of basis functions have been utilized in solving the above problem:

- Lagrange Elements of several order; the simplest ones being the linear elements; in addition higher order elements are used.
- Cubic Hermite Elements.

In case of linear Lagrange elements the matrix $\mathbf{M}$ in the linear system $\mathbf{M}_{\mu\nu} a^\nu = b_\mu$ is tridiagonal, as in case of the implicit finite difference (FD) approach. In contrast to the time-dependent FD algorithm the inversion of $\mathbf{M}$ yields the solution for the

FE case directly. In case however the original equation is nonlinear e.g. $\kappa = \kappa(A)$, the FE algorithm has to be iterated [7].

For the higher order Lagrange elements and the Cubic Hermite elements the matrix M is block-tridiagonal, or generally speaking a band-matrix. Both can be treated with a band matrix solver, or special techniques [7].

Appendix D

An alternative solution using Chebyshev spectral collocation in space and Gear's method (high order backwards differentiation) in time

Interpolation of a non-periodic function f at N equispaced points in (-1,1) by means of a trigonometric polynomial presents the Gibbs phenomenon at the boundaries, where overshooting occurs [8].

On the other hand, if one takes a Chebyshev grid on (-1,1)

$$x_j = \cos(\pi j/N) \qquad 0 \le j \le N, \tag{5.12}$$

and interpolates f with Chebyshev polynomials

$$T_k(x) = T_k(\cos\theta) = \cos(k\theta), \tag{5.13}$$

then the interpolation is spectrally accurate, i.e.

$$\forall x \in [-1,1] \qquad \max_{x\in[0,1]} \mid f(x) - \sum_{k=0}^{N} T_k(x) \mid \sim e^{-cN} \qquad c > 0$$

Chebyshev polynomials are very well suited to represent functions which have boundary layers. As one sees from (1), the distance d between the first two (or the last two) points of the mesh is

$$d = 1 - \cos(\pi/N) \sim \frac{\pi^2}{2N^2}$$

so that one can resolve, with N polynomials, layer thicknesses of order N^{-2}.

One can take advantage of the mentioned and other properties, to discretize space derivatives in PDE's.

Assume we want to compute df/dx at the points (1):

i) think of these values as 2π periodic data at

$$\theta_j = \pi j/N \qquad 0 \le j \le 2N;$$

ii) compute f_k via Fast Fourier Transform

$$f_k = \sum_{j=0}^{2N} e^{-ik\theta_j} f(\theta_j)$$

iii) compute $\frac{df}{d\theta}$ via inverse FFT

$$\frac{df}{d\theta}(\theta_j) = \frac{1}{2\pi}\sum_{k=-N+1}^{N-1}(ik)e^{ik\theta_j}f_k, \tag{5.14}$$

iv) change variables to get finally

$$\frac{df}{dx} = \frac{d\theta}{dx}\frac{df}{d\theta} = -\frac{1}{\sqrt{1-x^2}}\frac{df}{d\theta}.$$

The previous expression is not defined at the boundaries. An easy way out (not necessarily the best one) is to use l'Hopital rule and get

$$\frac{df}{dx}\Big|_{x=\pm1} = \frac{d^2f/d\theta^2}{d^2x/d\theta^2} = \mp\frac{d^2f}{d\theta^2}\Big|_{\substack{\theta=0\\ \theta=\pi}}$$

where the last derivative can be computed by an inversion similar to (3), premultiplying by $-k^2$ instead of ik.

The first step in the solution of the PDE is thus to "collocate" it at the mesh points (1).

In our case this means, considering for instance the second order term, that one computes $A_x(x_j)$ via the described procedure, then multiplies locally $[A(x_j)]^\alpha A_x(x_j)$, and then differentiates the result. The other terms are treated similarly.

Following the procedure indicated, one reduces the original PDE to a system of ODE's of the form

$$\frac{dA}{dt}(x_j) = \sum_{k=0}^{N} D_{jk}A(x_k) \qquad j = 0, .., N$$

Due to the global nature of the spectral approximation, the matrix D is full, in contrast with the banded matrices of finite differences. Implicit time integration is thus expensive, at least with exact, direct methods.

On the other hand some of the eigenvalues of the second order derivative matrix are $\sim N^4$; the corresponding stability limit on the time step for explicit (say Euler) integration could then be

$$\Delta t \leq O(N^{-4}).$$

However, numerical experiments have shown that the actual constraint on the case at hand is

$$\Delta t \leq O(N^{-2})$$

but nevertheless the solution by means of the NAG routine D02EAF, which utilizes Gear's method, proved to be an order of magnitude faster.

A word of caution must be added when collocation is applied to non-linear problems, since aliasing errors generally occur: only frequencies up to a certain threshold can be represented on the given grid. Higher frequencies, eventually generated by the non-linear operator during time evolution, are fed back to the lower ones that can be represented on the grid, introducing errors. These errors should however become of the order of the truncation error as N becomes large enough.

References:

[1] E.F.F. Botta and A.E.P. Veldman, Journal of Computational Physics **48**, 127–149 (1981).

[2] R. Peyret and T.D. Taylor,"Computational Methods for Fluid Flow", New York, NY; Springer 1983

[3] W. Shyy, Journal of Computational Physics **57**, 415–438 (1985).

[4] M. Ciment, S.M. Leventhal and B.C. Weinberg, Journal of Computational Physics **28**, 135–166 (1978).

[5] T.F. Chan, SIAM J. Numer. Anal. **21**, No. 2, April 1984.

[6] NAG Library,

[7] H.P. Zehrfeld et.al., Laboratory report on FE methods, in preparation

[8] D. Gottlieb, S.A. Orszag, "Numerical Analysis of Spectral Methods", SIAM (Philadelphia) 1977

Figure Captions:

Fig. 1a: Spatial mapping function and time transformation: $\varsigma = 0.250$.

Fig. 1b: Spatial mapping function and time transformation: $\varsigma = 0.999$.

Fig. 2: Spatial grids and location of variables.

Fig. 3: Finite difference solutions of the linear equation: A(y), A(x).

Fig. 4: Chebyshev solutions of the linear equation for several number of points and corresponding relative errors with respect to the analytical solution.

Fig. 5a: Convergence of several finite difference approximations: $\varsigma = 0$.

Fig. 5b: Convergence of several finite difference approximations: $\varsigma = 0.999$.

Fig. 6: Improvement due to space-transformation (zeta= .999; solid lines) compared with equidistant meshes ($\varsigma = .0$) for several difference schemes.

Fig. 7a: Convergence of Lagrange finite element approximations; the dotted line is calculated for Lagrange elements of second order and $\varsigma = .999$.

Fig. 7b: Convergence for cubic Hermite elements.

Fig. 7c: Convergence of spectral Chebyshev collocation approximation.

Fig. 8a: Finite difference solution of the non-linear equation NL1: A(y).

Fig. 8b: Finite difference solution of the non-linear equation NL1: A(x).

Fig. 9a: Finite difference solution of the non-linear equation NL2: A(y).

Fig. 9b: Finite difference solution of the non-linear equation NL2: A(x).

Fig. 10a: Computational cost for the implicit solution of the non-linear equation NL1 with standard (solid line) and time-transformed technique (dashed).

Fig. 10b: Computational cost for the implicit solution of the non-linear equation NL1 with standard and time-transformed technique; the non-linear term has been linearized.

Fig. 11a: Computational cost for the implicit solution of the non-linear equation NL2 with standard (solid line) and time-transformed technique (dashed).

Fig. 11b: Computational cost for the implicit solution of the non-linear equation NL2 with standard and time-transformed technique; the non-linear has been linearized.

fig. 1a

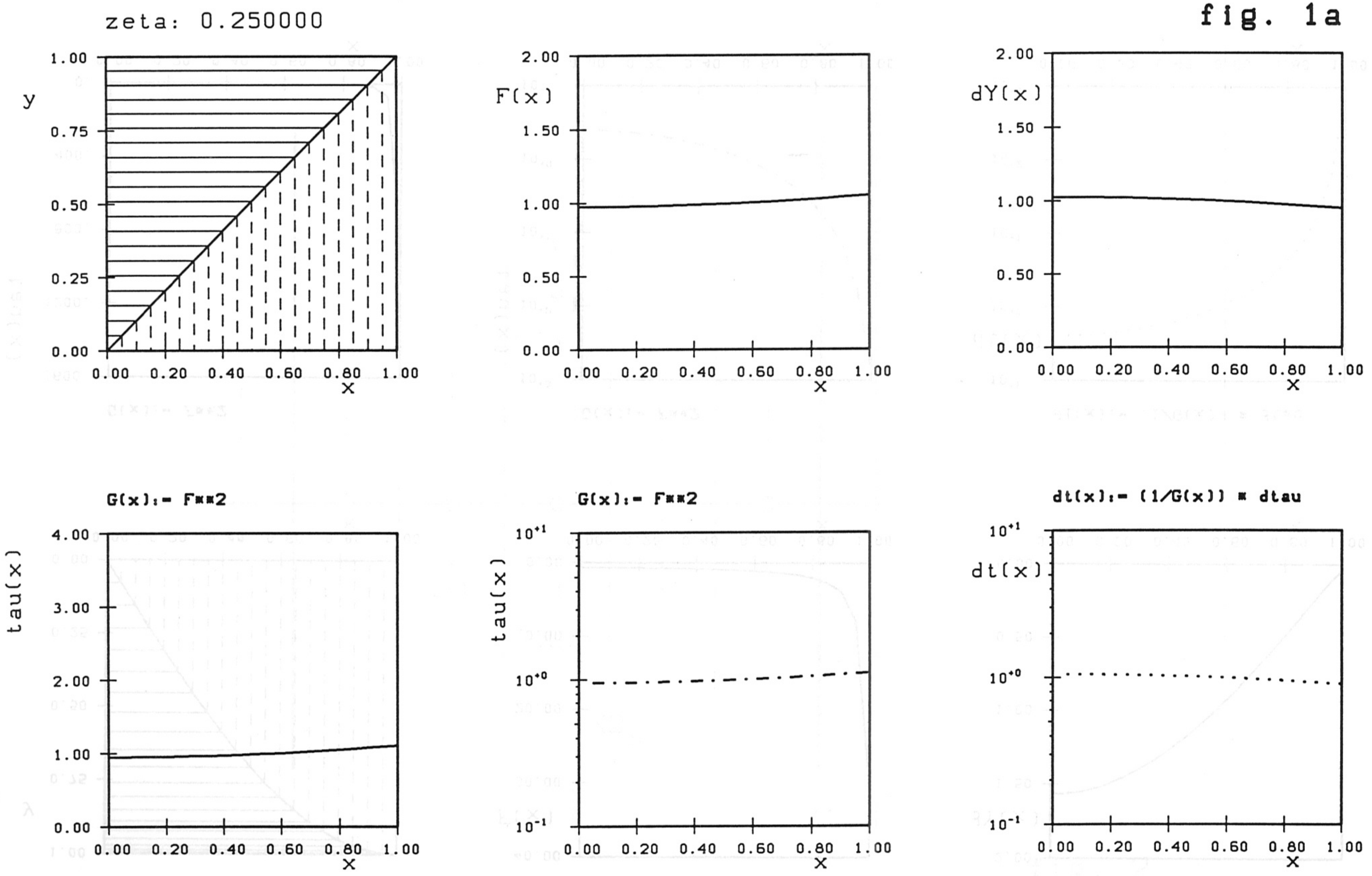
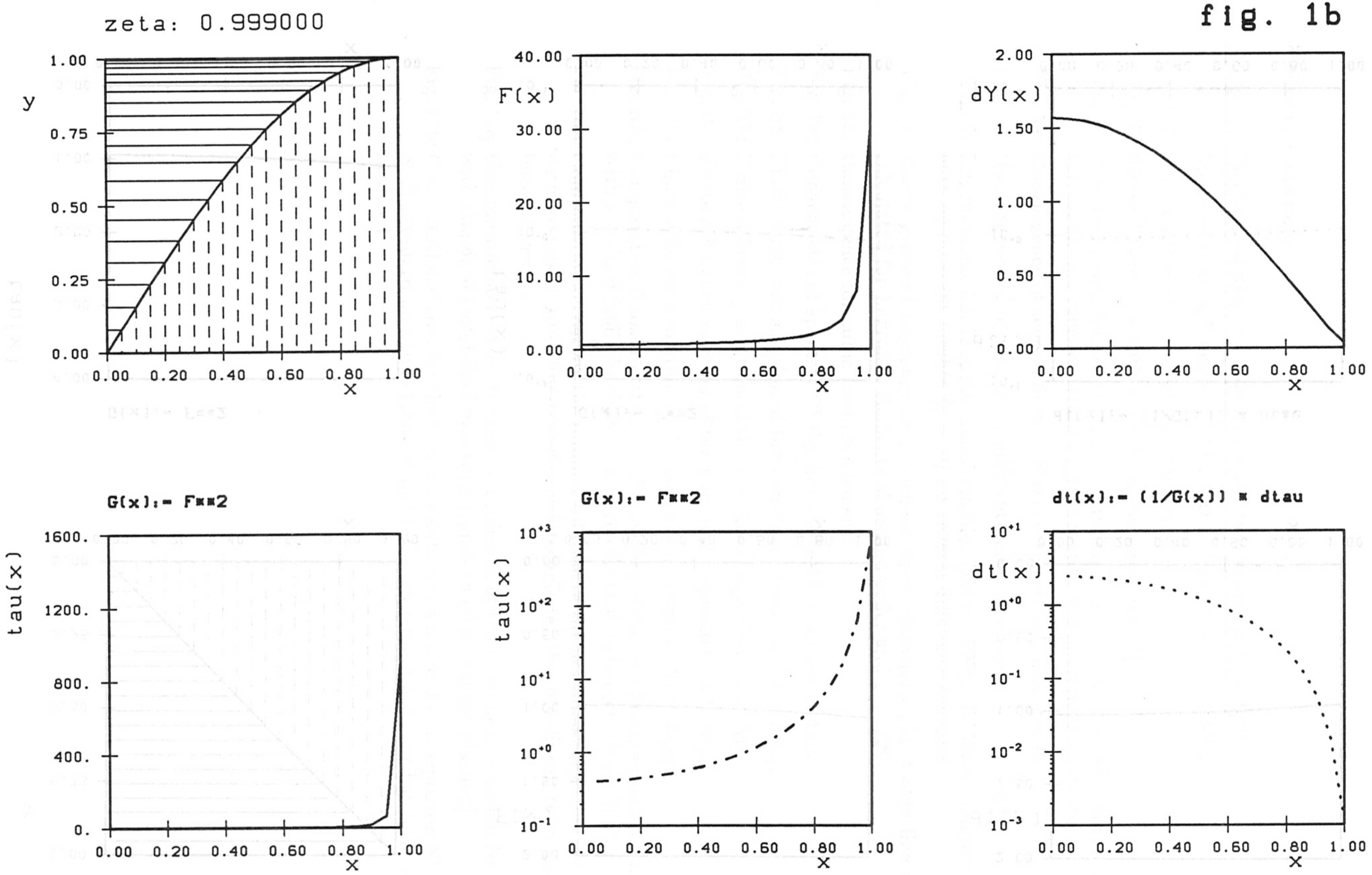

zeta: 0.250000
y
1.00
0.75
0.50
0.25
0.00
0.00 0.20 0.40 0.60 0.80 1.00
x
F(x)
2.00
1.50
1.00
0.50
0.00
0.00 0.20 0.40 0.60 0.80 1.00
x
dY(x)
2.00
1.50
1.00
0.50
0.00
0.00 0.20 0.40 0.60 0.80 1.00
x
G(x):= F**2
tau(x)
4.00
3.00
2.00
1.00
0.00
0.00 0.20 0.40 0.60 0.80 1.00
x
G(x):= F**2
tau(x)
10^{+1}
10^{+0}
10^{-1}
0.00 0.20 0.40 0.60 0.80 1.00
x
dt(x):= (1/G(x)) * dtau
dt(x)
10^{+1}
10^{+0}
10^{-1}
0.00 0.20 0.40 0.60 0.80 1.00
x

fig. 1b


zeta: 0.999000
y
1.00
0.75
0.50
0.25
0.00
0.00 0.20 0.40 0.60 0.80 1.00
x
F(x)
40.00
30.00
20.00
10.00
0.00
0.00 0.20 0.40 0.60 0.80 1.00
x
dY(x)
2.00
1.50
1.00
0.50
0.00
0.00 0.20 0.40 0.60 0.80 1.00
x
G(x):= F**2
tau(x)
1600.
1200.
800.
400.
0.
0.00 0.20 0.40 0.60 0.80 1.00
x
G(x):= F**2
tau(x)
10^+3
10^+2
10^+1
10^+0
10^-1
0.00 0.20 0.40 0.60 0.80 1.00
x
dt(x):= (1/G(x)) * dtau
dt(x)
10^+1
10^+0
10^-1
10^-2
10^-3
0.00 0.20 0.40 0.60 0.80 1.00
x

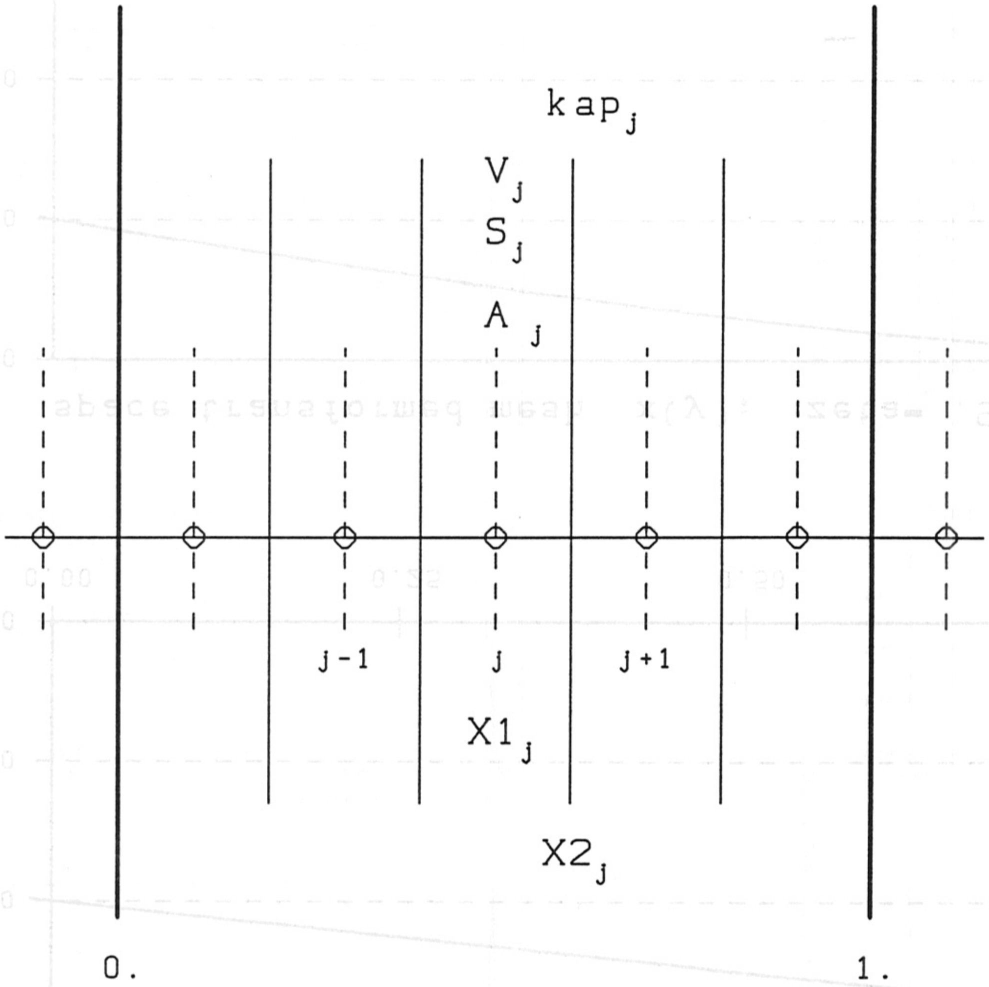
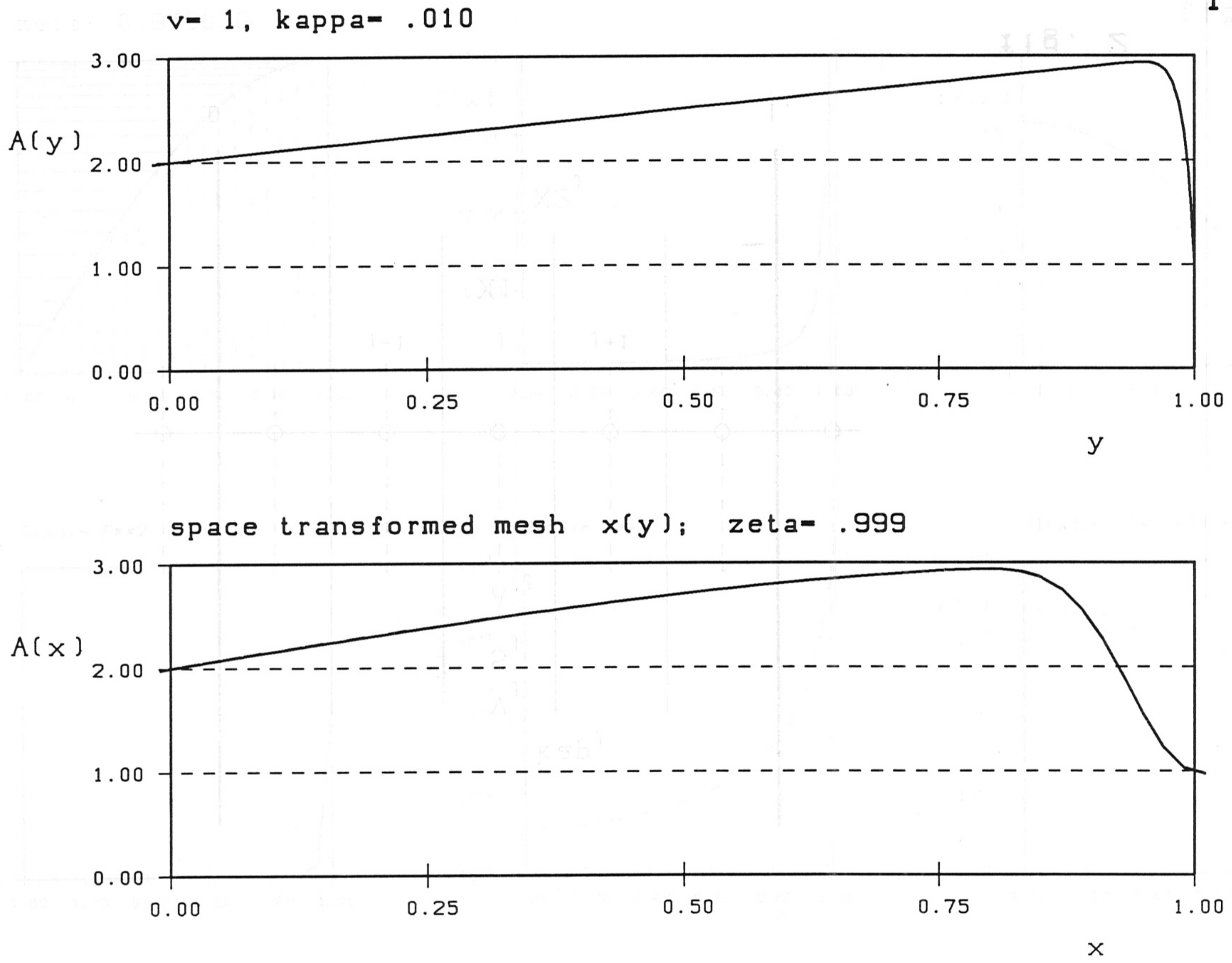


fig. 2

fig. 3



v= 1, kappa= .010

space transformed mesh x(y); zeta= .999

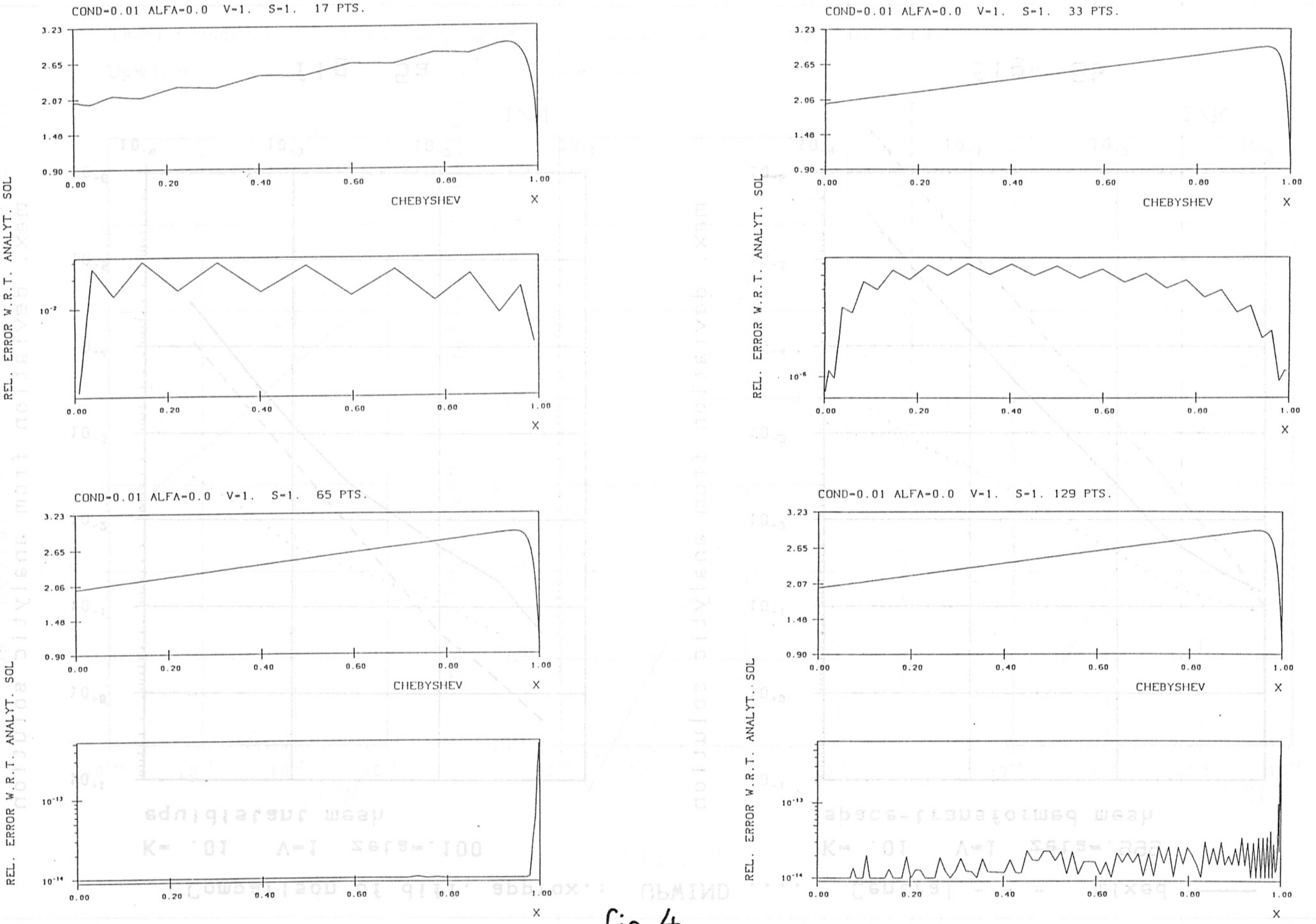

COND=0.01 ALFA=0.0 V=1. S=1. 17 PTS.
COND=0.01 ALFA=0.0 V=1. S=1. 33 PTS.
COND=0.01 ALFA=0.0 V=1. S=1. 65 PTS.
COND=0.01 ALFA=0.0 V=1. S=1. 129 PTS.
3.23
2.65
2.07
2.06
1.48
0.90
0.00
0.20
0.40
0.60
0.80
1.00
CHEBYSHEV
X
REL. ERROR W.R.T. ANALYT. SOL
10⁻²
10⁻⁶
10⁻¹³
10⁻¹⁴

fig. 4

Comparison of diff. approx.: UPWIND .... Central - - - mixed ——

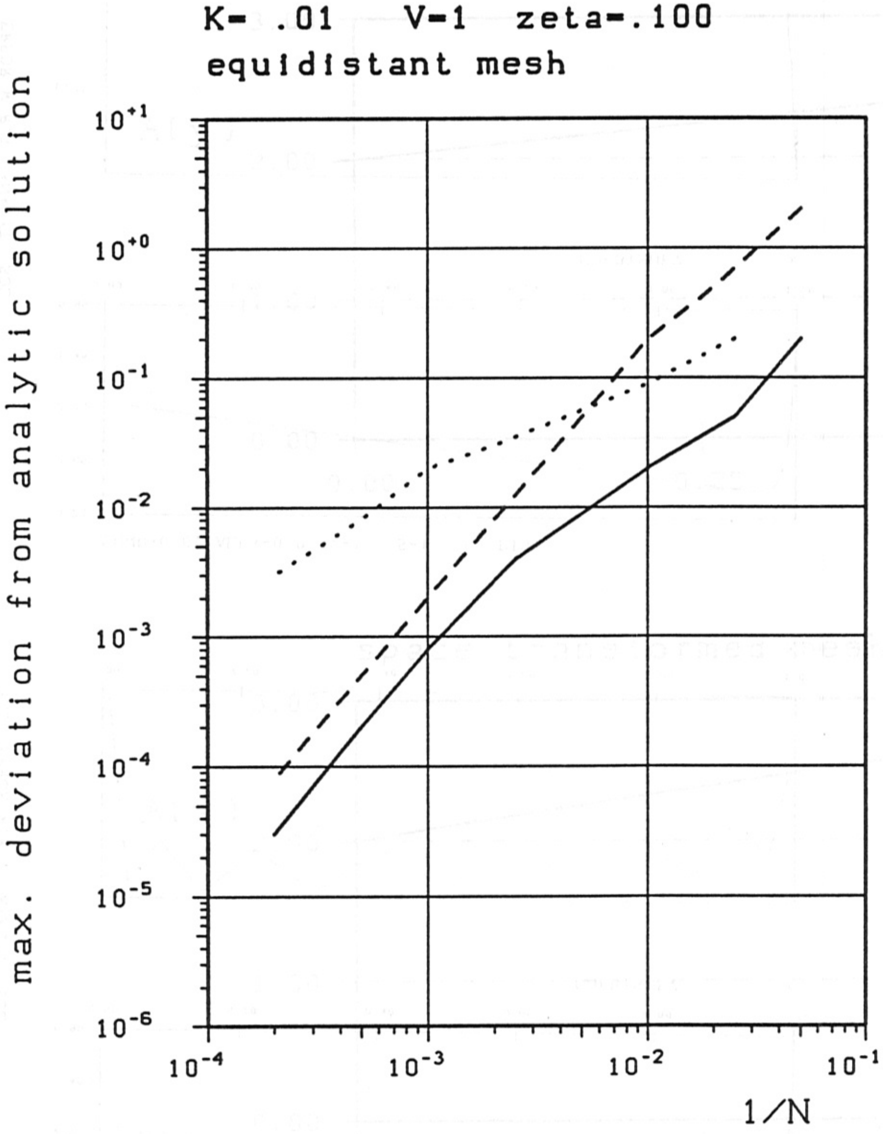


fig. 5a

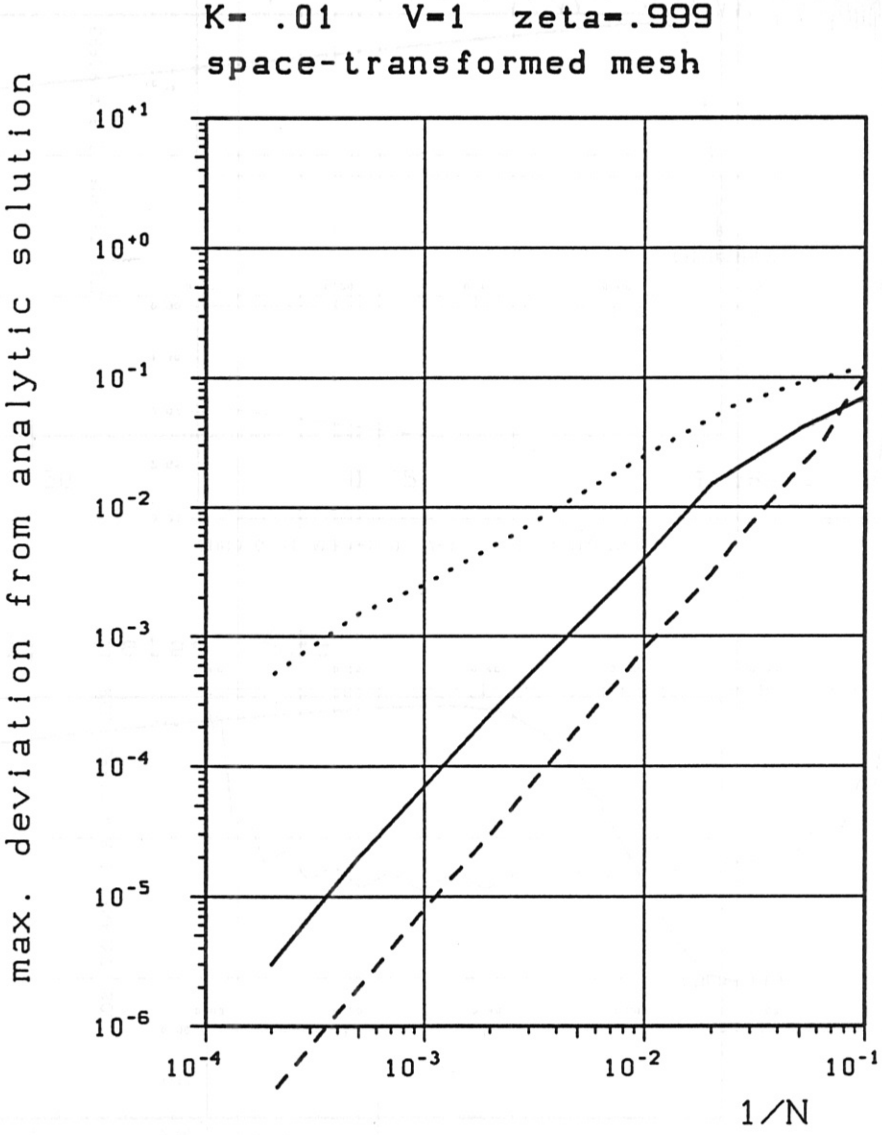


fig. 5b

Improvement of accuracy due to space-transformation zeta= .999 (k= .01 , V= 1)

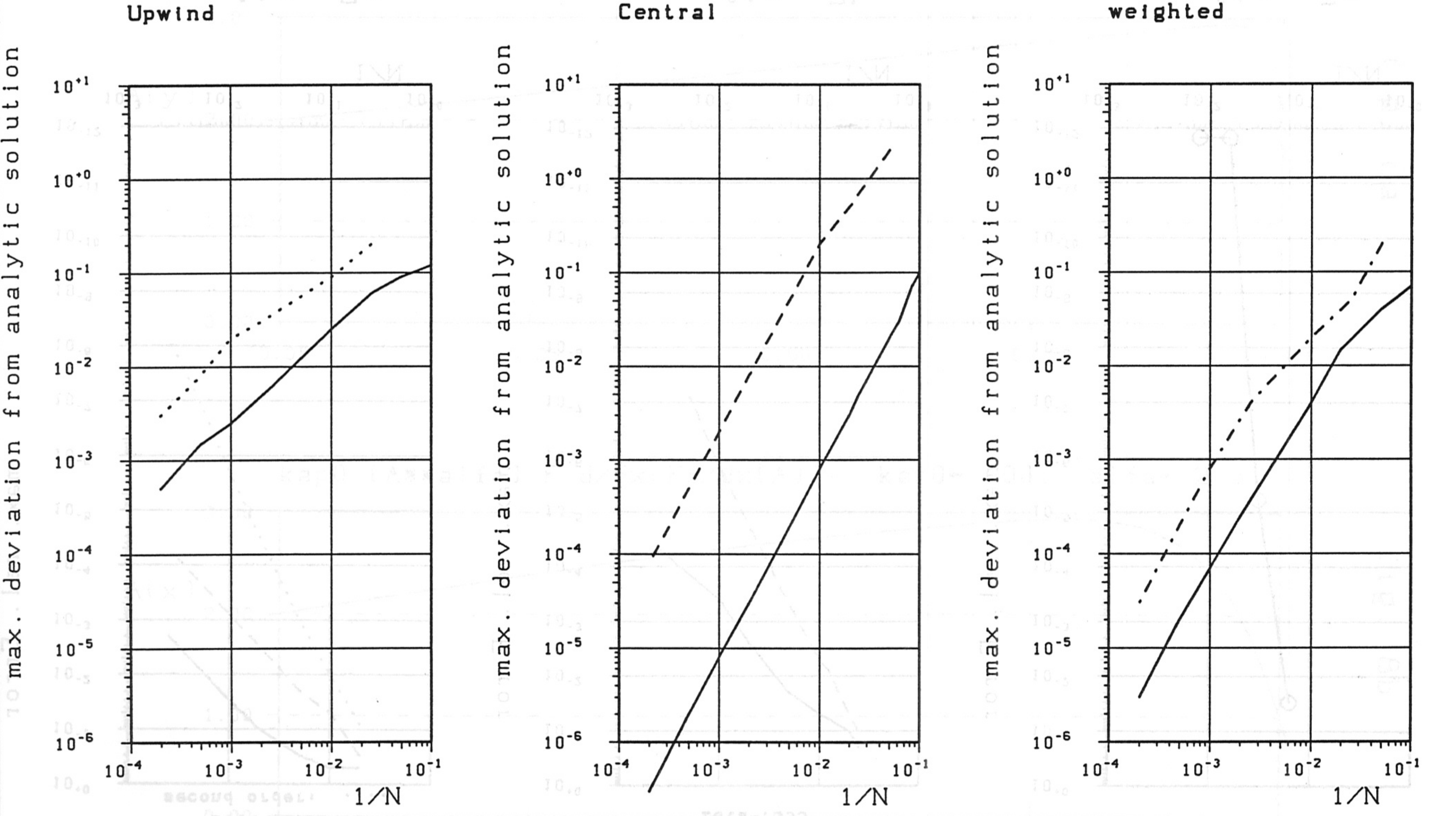


fig. 6

Comparison of FE-approaches and Chebyshev collocation; k=.010, v= 1

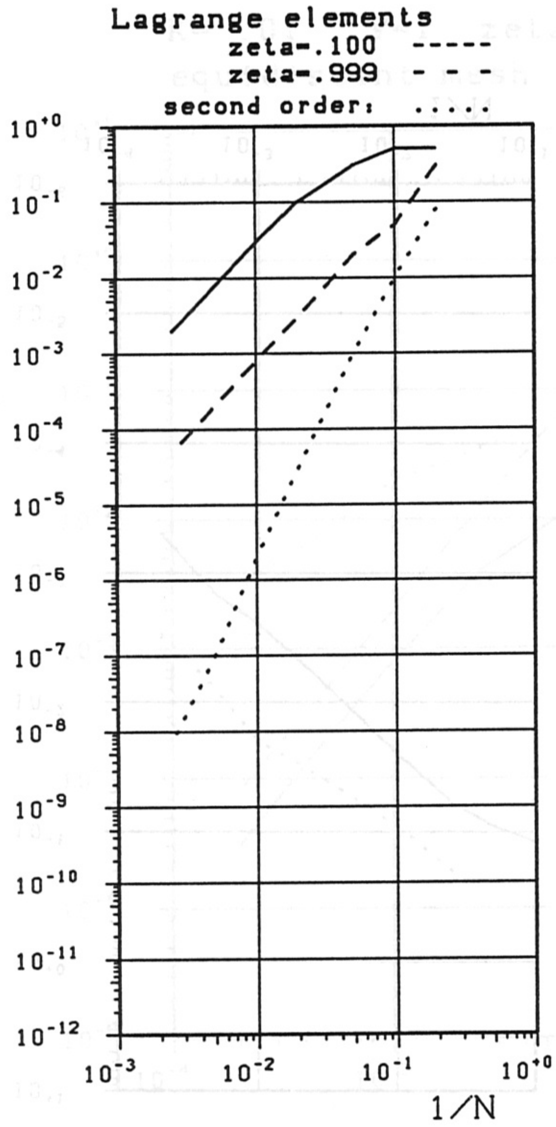


fig. 7a

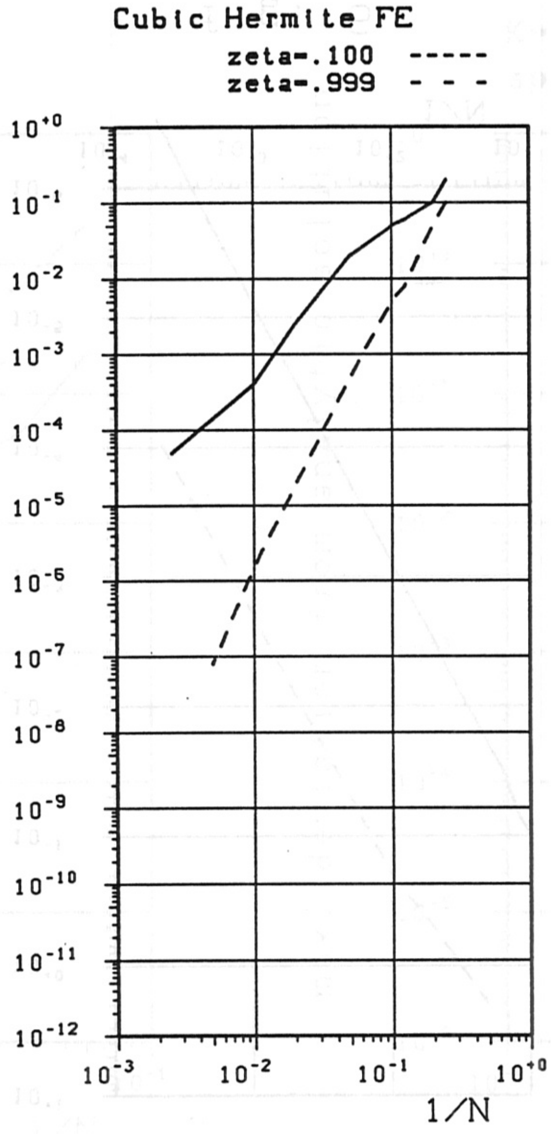


fig. 7b

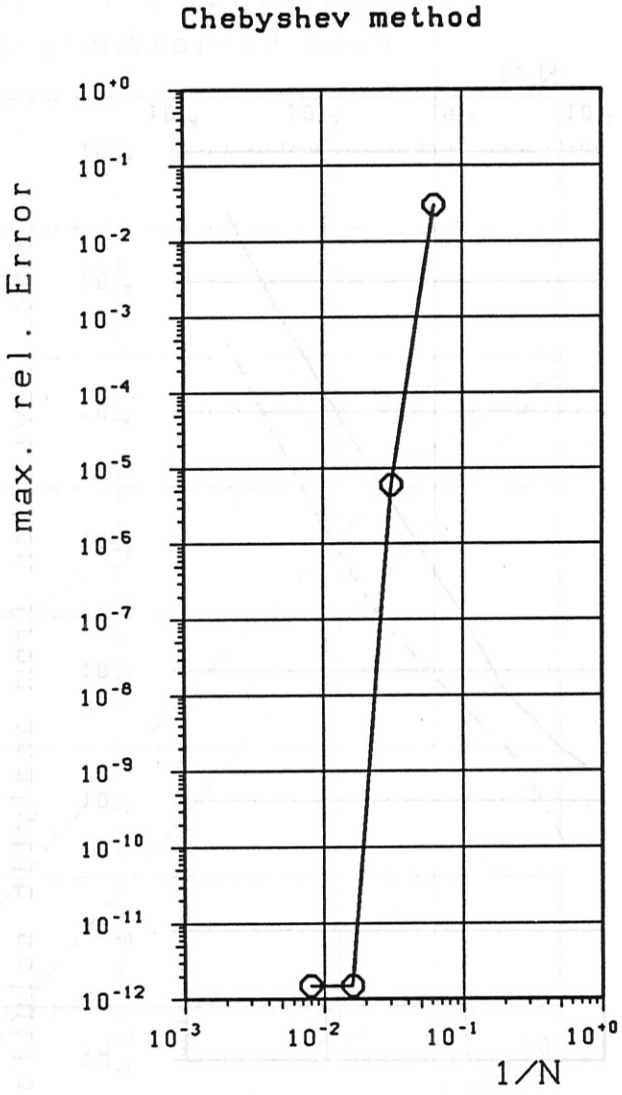


fig. 7c

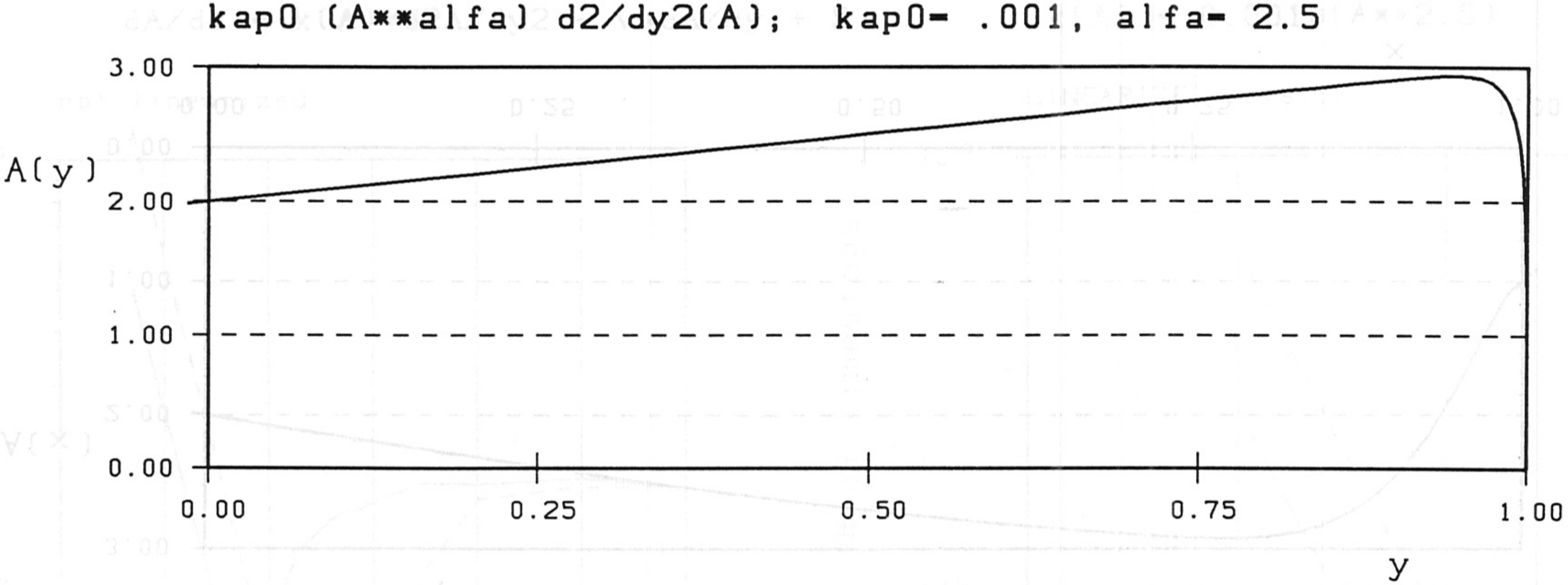

kap0 (A**alfa) d2/dy2(A); kap0= .001, alfa= 2.5
A(y)
3.00
2.00
1.00
0.00
0.00
0.25
0.50
0.75
1.00
y

fig. 8a

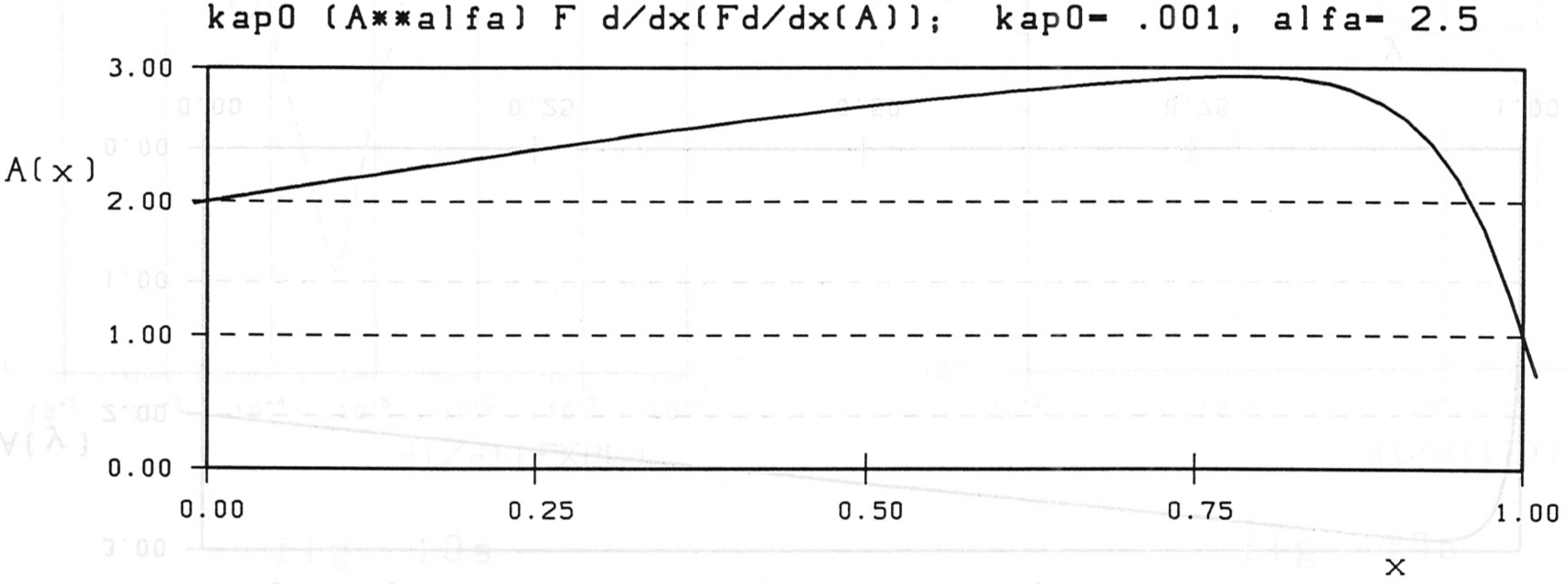

kap0 (A**alfa) F d/dx(Fd/dx(A)); kap0= .001, alfa= 2.5
A(x)
3.00
2.00
1.00
0.00
0.00
0.25
0.50
0.75
1.00
x

fig. 8b

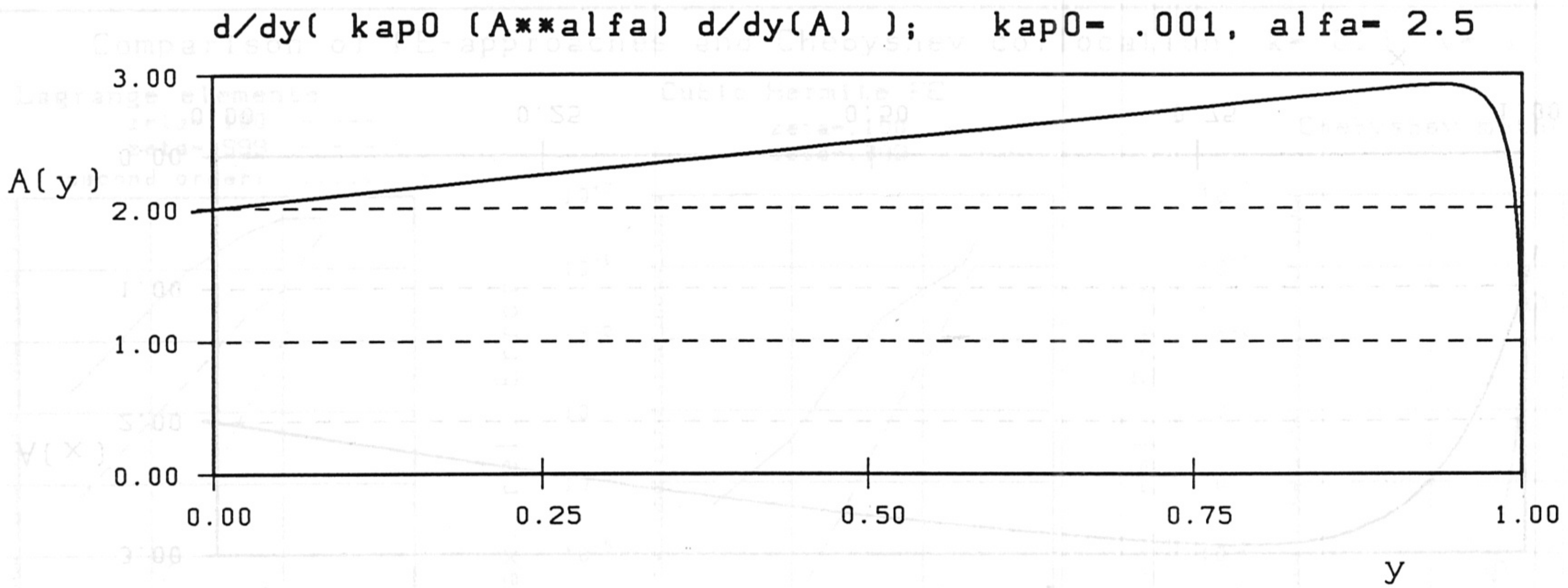

d/dy(kap0 (A**alfa) d/dy(A));   kap0= .001, alfa= 2.5
A(y)
3.00
2.00
1.00
0.00
0.00
0.25
0.50
0.75
1.00
y

fig. 9a

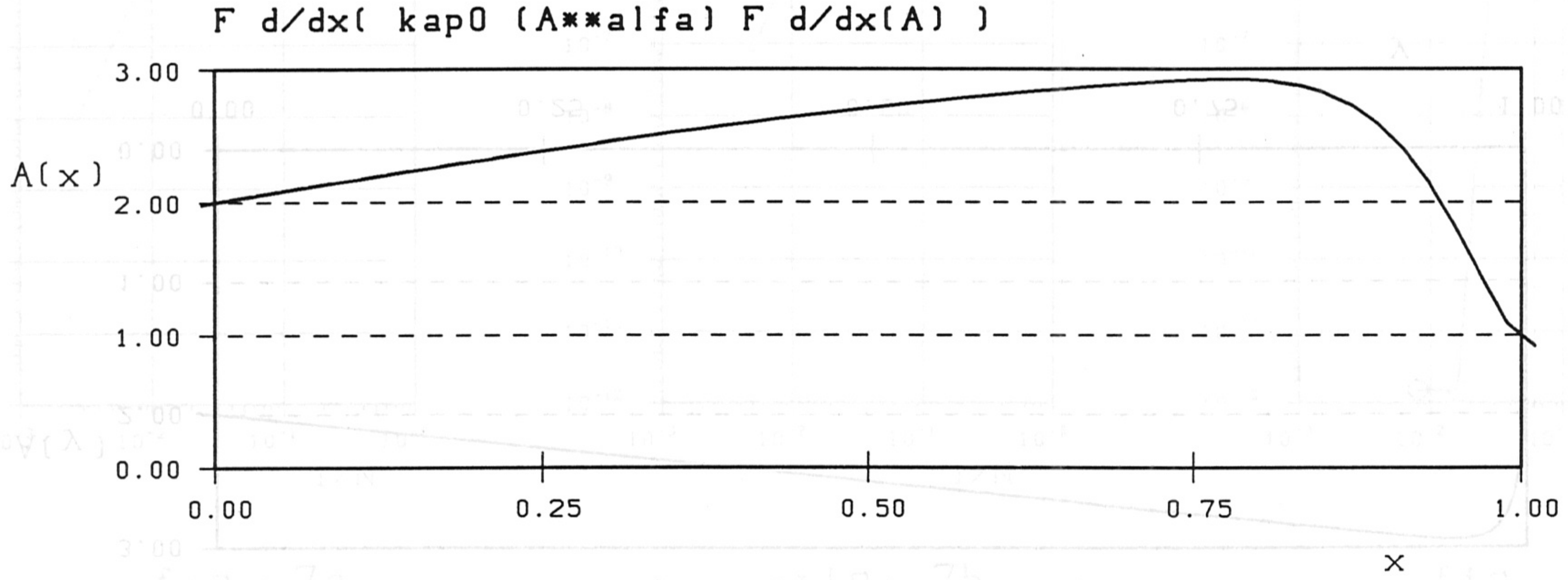

F d/dx(kap0 (A**alfa) F d/dx(A))
A(x)
3.00
2.00
1.00
0.00
0.00
0.25
0.50
0.75
1.00
x

fig. 9b

dA/dt = k(A)*d2A/dy2 - v*dA/dy + S   k(A) = 0.001*(A**2.5)

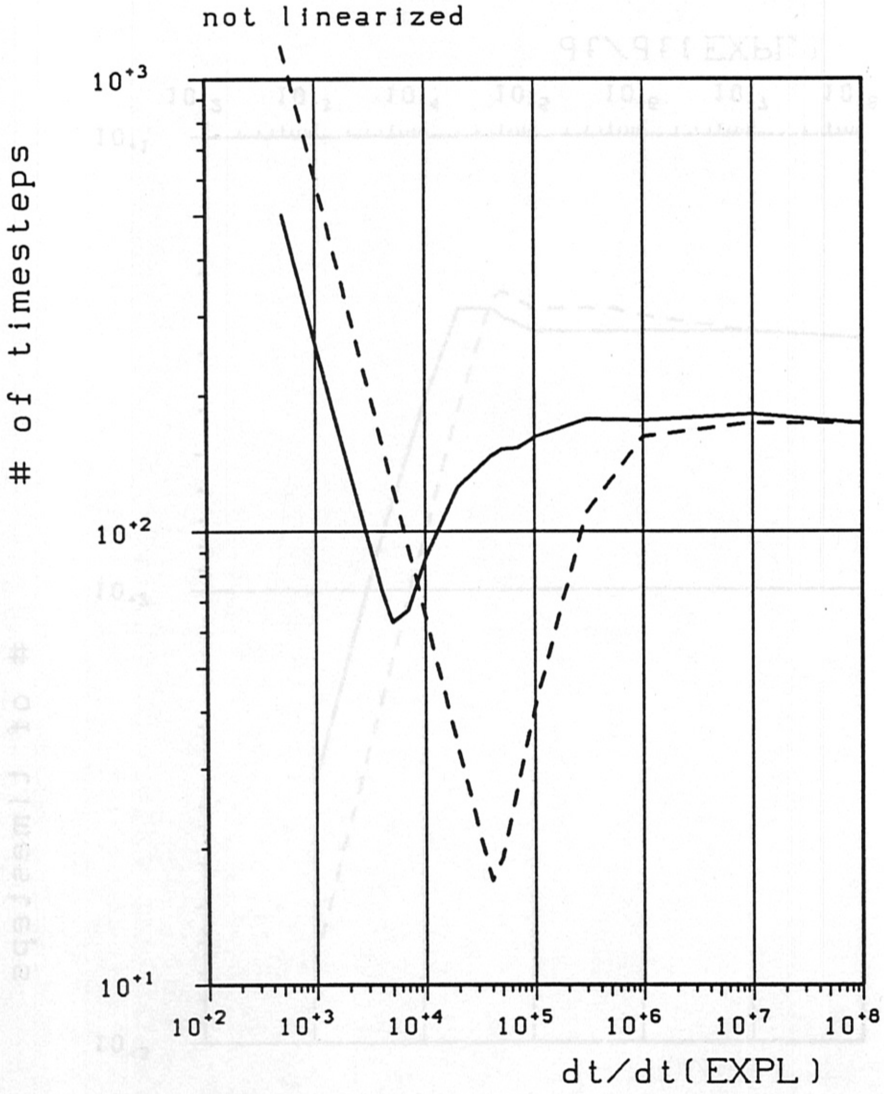


fig. 10a

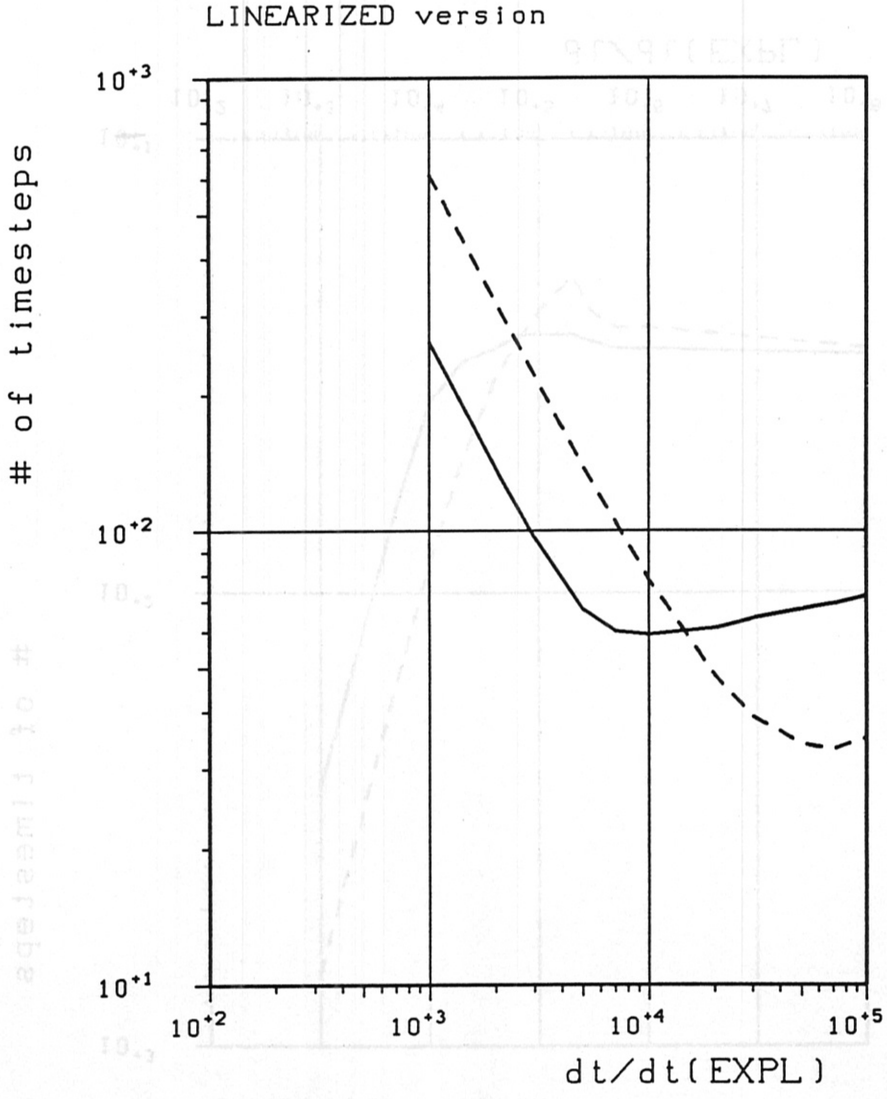


fig. 10b

dA/dt= d/dy(k(A)*dA/dy) - v*dA/dy + S;   k(A) = 0.001*(A**2.5)

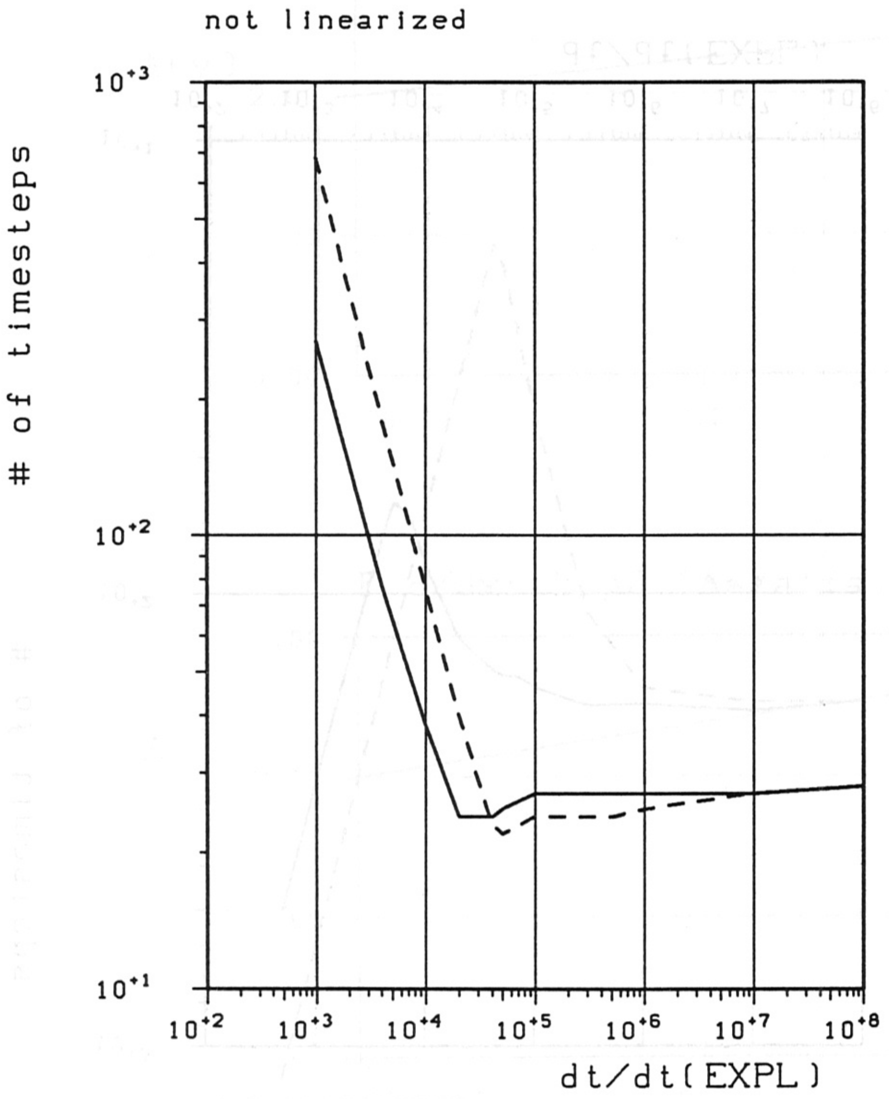

not linearized

fig. 11a

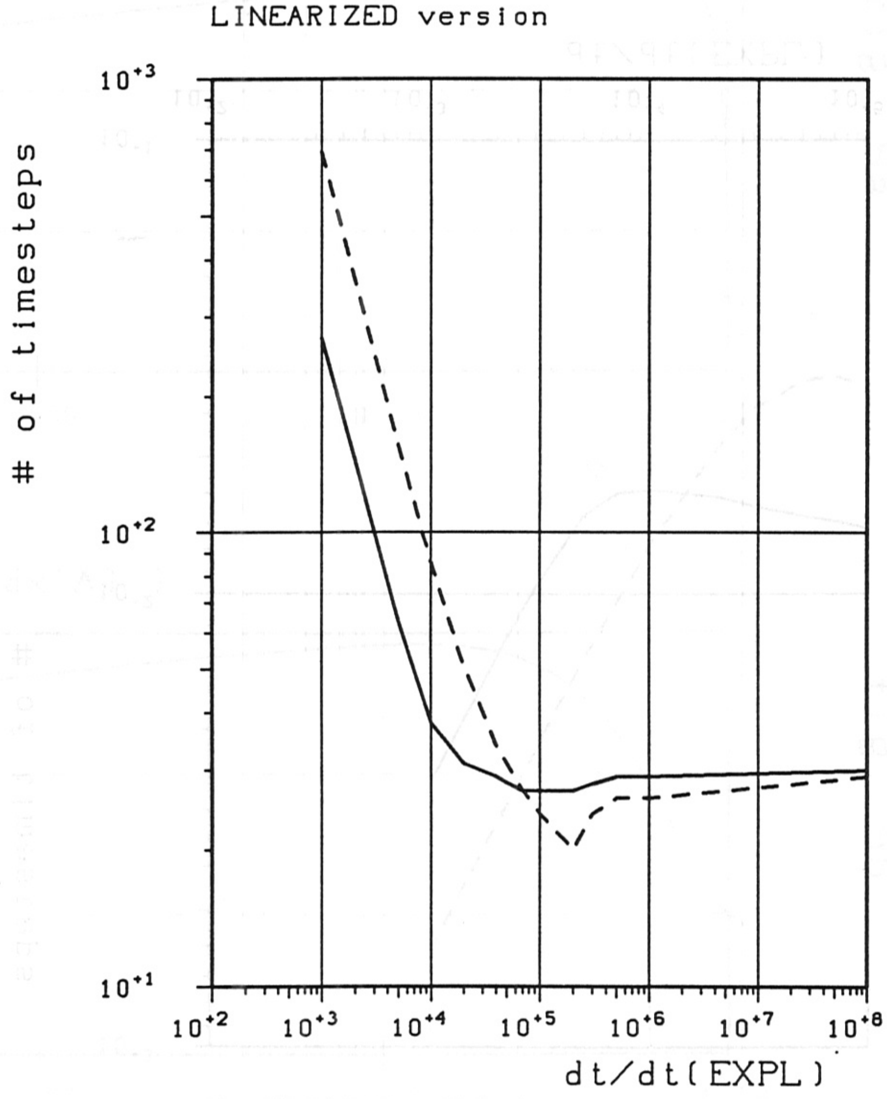

LINEARIZED version

fig. 11b