C.E.A.
Etablissement : C.E. de SACLAY
Direction des Réacteurs Nucléaires
Département de Mécanique & de Technologie

Date : LE 7 OCTOBRE 1994
Réf. : DMT/SERMA LETR/94/109

CEA-DMT

95 000583
CIRST

DEMANDE D'AUTORISATION
~~DE COMMUNICATION~~ OU DE PUBLICATION

PUBLICATION - TITRE : CONVERGENCE ACCELERATION OF TWO-PHASE FLOW CALCULATIONS IN FLICA 4, A THERMAL HYDRAULIC 3D COMPUTER CODE

Auteur (s) : I.TOUMI
Nature de la Publication :
Editeurs (CEA, AIEA, OCDE,...) : JPSJ26IN, FR9503250

COMMUNICATION - TITRE :
Auteur (s) :

Titre de la Conférence : ANS International Conf on Mathematics, ~~Reactor~~ Physics and Computational [illegible]

Lieu et Date : PORTLAND 30 Avril 4 Mai 95

ORGANISATEURS : ANS

ARRIVEE - CIRST
07 FEV.95 | 000965
Circ. : | Cit : NC

Date de remise des textes :
Commentaires :

Ce texte a-t-il été déjà publié ? : NON
Si OUI : Référence de la publication antérieure :
Cette publication contient-elle, à votre avis, des informations brevetables OUI (NON)

DATE DE LA DEMANDE : 7/10/94

Chef de Laboratoire : P. RAYMOND

Chef de Service : J.B. THOMAS

DEMANDE D'AVIS (éventuellement) :

a) - C.P.I./Saclay — OUI NON
b) - Chargé de Mission pour les affaires industrielles — OUI NON
c) - SYFRA — OUI NON
d) - Partenaires concernés (EDF, FRA, Dpts de la DRN) : (joindre photocopie) — Date de la Demande :
e) - Autres unités opérationnelles : (joindre photocopie) — Date de la Demande :
f) - Autres avis demandés par le Chef du D.M.T. :

Avis du : (le cas échéant)

Date :
les avis sont à envoyer au DMT/DIR

Date d' Arrivée au D.M.T. — Décision D.M.T. n° 215 — Date 14-10-94

Le Chef du Département

P.J.: Un texte complet -
Copie : DMT/DOC
~~DMT/EA~~ + Fuselta
Nota : Copie autorisation + Résumé + Texte à I.N.S.T.N./MIST/CIRST.

CONVERGENCE ACCELERATION OF TWO-PHASE FLOW CALCULATIONS IN FLICA-4. A THERMAL-HYDRAULIC 3D COMPUTER CODE.

Imad TOUMI
Commissariat à l'Energie Atomique
DMT/SERMA. Bat 470
CE Saclay.91191 Gif-sur-Yvette (France)
Fax : (331) 69 08 23 81
e-mail: toum@soleil.serma.cea.fr

Abstract

Time requirements for 3D two-phase flow steady state calculations are generally long. Usually, numerical methods for steady state problems are iterative methods consisting in time-like methods that are marched to a steady state. Based on the eigenvalue spectrum of the iteration matrix for various flow configuration, two convergence acceleration techniques are discussed: overrelaxation and eigenvalue annihilation. This methods were applied to accelerate the convergence of three dimensional steady state two-phase flow calculations within the **FLICA-4** computer code. These acceleration methods are easy to implement and no extra computer memory is required. Successful results are presented for various test problems and a saving of 30 to 50 % in CPU time have been achieved.

1 Introduction

For design and safety studies of nuclear reactors, the need for faster and more efficient numerical method for two-phase flow computations becomes more pronounced. The **FLICA-4** computer code is devoted to three dimensional thermal-hydraulic steady state and transients analysis of nuclear reactor cores. The **FLICA-4** numerical method consists in a finite volume method based on an approximate Riemann solver for the discretization of inviscid flux terms and on a centered scheme for diffusive terms. For advancing in time, a linearized implicit integrating step is used. The resulting method for steady state computations consists in a time marching algorithm which can be written as an iterative method

$$x_{n+1} = Mx_n + b$$

From the spectral analysis done in this paper, it is noticed that the rate of convergence of this iterative method is governed by the spectral radius of the matrix M. Our purpose here is to point out a simple and easy technique which can is some cases produce a significant speedup in the computation of the steady state.

There are various ways of improving the rate of convergence of an iterative method. Two of them are overrelaxation methods [2] and eigenvalue annihilation [4]. Roughly speaking, convergence acceleration is a transformation T of previous iterate solutions $x_n, x_{n-1}, \cdots, x_{n-k}$ that produces $T(x_n, x_{n-1}, \cdots, x_{n-k})$ which is a better approximation than x_{n+1} to the solution. Overrelaxation methods consist in choosing a transformation

$$T(x_n, x_{n-1}) = x_{n-1} + \omega(x_n - x_{n-1})$$

and a parameter ω that minimizes the spectral radius of the iteration matrix. The idea underlying the explicit eigenvalue annihilation is that in the later stages of

the marching algorithm, the iteration process may behave linearly. If so, one can estimate the dominant eigenvalue of the iteration matrix and annihilate the error in the direction of the eigenvector corresponding to this eigenvalue by a special transformation.

This paper presents an application of the Richardson acceleration method and the explicit eigenvalue annihilation for two-phase flow calculations within the FLICA-4 computer code.

2 Governing Equations

The two-phase flow model of the FLICA-4 computer code consists in a mixture mass conservation equation, a mixture momentum balance equation, a mixture energy balance equation and a mass phase equation which allows the calculation of subcooled boiling flows.

$$\begin{aligned}
&\partial_t \left(\sum_{k=v,l} \rho_k \alpha_k \right) + \nabla \left(\sum_{k=v,l} \rho_k \alpha_k \mathbf{u}_k \right) = 0 \\
&\partial_t \left(\sum_{k=v,l} \rho_k \alpha_k \mathbf{u}_k \right) + \nabla \left(\sum_{k=v,l} \left(\rho_k \alpha_k \mathbf{u}_k \otimes \mathbf{u}_k + \alpha_k \Pi_k \right) \right) \\
&\qquad = \sum_{k=v,l} \left(\rho_k \alpha_k \vec{g} + \tau_p^k \right) \\
&\partial_t \left(\sum_{k=v,l} \rho_k \alpha_k \left(e_k + \frac{\mathbf{u}_k^2}{2} \right) \right) + \nabla \left(\sum_{k=v,l} \left(\rho_k \alpha_k \mathbf{u}_k \left(e_k + \frac{\mathbf{u}_k^2}{2} \right) \mathbf{u}_k \right. \right. \\
&\qquad \left. \left. - \alpha_k \Pi_k \cdot \mathbf{u}_k - \alpha_k \mathbf{q}_k \right) \right) = \mathcal{Q}_{tot} + \sum_{k=v,l} \rho_k \alpha_k \mathbf{u}_k \cdot \vec{g} \\
&\partial_t \left(\rho_k \alpha_k \right) + \nabla \left(\rho_k \alpha_k \mathbf{u}_k - \alpha_k \mathbf{M}_k \right) = \Gamma_k
\end{aligned} \tag{1}$$

In Equation (1) the subscript k refers to the vapor phase ($k = v$) or the liquid phase ($k = l$). The nomenclature is as follows: α is the void volume fraction ($\alpha_v + \alpha_l = 1$), ρ is the phasic density, $\mathbf{u}$ is the phasic velocity, e is the phasic internal energy and h is the phasic enthalpy. $\mathbf{M}_k$ is a phase mass diffusion term due to two-phase flow turbulence. Π_k contains the viscous stress tensor and turbulence effects modelling, and $\mathbf{q}_k$ represents fluid heat conduction and energy turbulence diffusion.

In the right hand side of Equation (1), Γ represents the interphase mass exchange, τ_p^k represents the wall drag force, $\mathcal{Q}_{tot}$ is the wall heat flux and $\vec{g}$ is the gravity. The two phases are assumed to be at the same pressure.

3 Numerical Method

3.1 Finite Volume Method

Considering an integral form of the partial differential equations on a computational domain Ω, a finite volume method leads to the following approximation of the set of balance equations:

$$\frac{Vol\left(\Omega_i\right)}{\Delta t} \left(U_i^n - U_i^{n-1} \right) + \sum_j \left(F_{nv} \left(U_i^n, U_j^n \right) + F_v \left(U_i^n, U_j^n \right) \right) = S \left(U_i^n \right) \tag{2}$$

where U_i^n denotes the conservative variables, $S(U_i^n)$ is the vector of source terms, and $F_{nv}(U_i^n, U_j^n)$ and $F_v(U_i^n, U_j^n)$ denote respectively the non viscous and the viscous contributions to the flux on the cell Ω_i in direction of the neighbor Ω_j. The evaluation of the viscous contributions requires a knowledge of the first derivatives of intensive variables as velocities, temperature and phase concentration. They are obtained by a centered difference scheme in which the derivatives are represented by piecewise constant functions over the computational domain.
Here, we describe the basic points of the numerical method, which are

- the use of an approximate Riemann solver to descretize the inviscid fluxes between two cells Ω_i and Ω_j
- the implicit integrating step to advance in time.

3.2 Approximate Riemann Solvers

The inviscid flux in the normal direction to the cell interface $\partial\Omega_{ij}$ is given by

$$F_{nv}(U_i^n, U_j^n) = \int\int_{\partial\Omega_{ij}} (F(U) \cdot \vec{n})\, d\sigma \tag{3}$$

From an original idea proposed by Godunov [3] for gas dynamic calculations, an efficient way to define upwind numerical fluxes is to solve local one dimensional Riemann problems at cell interfaces

$$\begin{gathered} \partial_t U + \partial_x F(U) = 0 \\ U(x,0) = U_i^n \quad (x < x_{ij}) \qquad U(x,0) = U_j^n \quad (x > x_{ij}) \end{gathered} \tag{4}$$

where $F(U)$ is the inviscid flux in the x-direction normal to the cell interface $\partial\Omega_{ij}$ located at $x = x_{ij}$.

In practice solving the above nonlinear Riemann problem may be difficult and time consuming because it typically requires some iterations for nonlinear equations. This suggests to use approximate Riemann solvers to build more efficient Godunov type numerical methods. One of the most popular approximate Riemann solver, currently in use in aerodynamics problems, is due to Roe [5]. To solve the nonlinear Riemann problem (4) for hyperbolic systems of conservation laws, Roe introduces a local linearization

$$\begin{gathered} \partial_t U + A(U_i^n, U_j^n)\partial_x U = 0 \\ U(x,0) = U_i^n \quad (x < x_{ij}) \qquad U(x,0) = U_j^n \quad (x > x_{ij}) \end{gathered} \tag{5}$$

where the matrix $A(U_i^n, U_j^n)$, known as the Roe averaged matrix, is some Jacobian matrix depending on the initial states U_i^n and U_j^n. The Roe averaged matrix $A(U_i^n, U_j^n)$ is constructed to have the following properties

$$F(U_j^n) - F(U_i^n) = A(U_i^n, U_j^n)(U_j^n - U_i^n)$$

In Reference [5], Roe gives a method to construct the matrix for Euler equations with perfect gases.

However, Roe's method does not apply directly to thermal-hydraulic model due to two equations of state and unequal velocity fields. To overcome these difficulties, we use a weak formulation of Roe's approximate Riemann solver [8], which has been applied to construct a Roe averaged Jacobian matrix for the Euler equations with arbitrary equations of state. This weak formulation has also been used to build approximate Riemann solvers for an homogeneous two-phase flow model [8], and

for two-fluid models [10]. [7]

Once the Roe averaged matrix has been constructed, Roe's numerical flux between cells Ω_i and Ω_j is given

$$F(U_i^n, U_j^n) = \frac{1}{2}\left(F(U_i^n) + F(U_j^n) - \left|A(U_i^n, U_j^n)\right|(U_j^n - U_i^n)\right) \tag{6}$$

A full description of this numerical flux and Roe averaged matrix for the FLICA-4 thermal-hydraulic model is given in [9].

3.3 Implicit Linearized Scheme

The matrix $A(U_i^n, U_j^n)$ can be split into a positive and a negative part:

$$A(U_i^n, U_j^n) = A(U_i^n, U_j^n)^+ - A(U_i^n, U_j^n)^- \tag{7}$$

with

$$\begin{aligned} A(U_i^n, U_j^n)^+ &= R_{ij}^{-1}\left(|\Lambda_{ij}| + \Lambda_{ij}\right) R_{ij} \\ A(U_i^n, U_j^n)^- &= R_{ij}^{-1}\left(|\Lambda_{ij}| - \Lambda_{ij}\right) R_{ij} \end{aligned} \tag{8}$$

Λ_{ij} is a diagonal matrix containing the eigenvalues of $A(U_i^n, U_j^n)$ and R_{ij}^{-1} and R_{ij} are matrices which contain the right and left eigenvectors of the matrix $A(U_i^n, U_j^n)$. It is to be noted that both $A(U_i^n, U_j^n)^+$ and $A(U_i^n, U_j^n)^-$ have non-negative eigenvalues. The inviscid flux (6) can be written in either of the two equivalent form:

$$\begin{aligned} F_{nv}(U_i^n, U_j^n) &= F(U_i^n) - A(U_i^n, U_j^n)^- \cdot (U_j^n - U_i^n) \\ &= F(U_j^n) + A(U_i^n, U_j^n)^+ \cdot (U_j^n - U_i^n) \end{aligned} \tag{9}$$

Using the first expression for the flux through the interface $\partial\Omega_{ij}$ and the second for the flux through the opposite interface, leads to a flux balance equation of the form:

$$\frac{(U_i^n - U_i^{n-1})}{\Delta t} + \left(\sum_k C_k^n\right) U_i^n - \sum_k (C_k^n U_k^n) = S(U_i^n) \tag{10}$$

where the sum is over all neighbors Ω_k of Ω_i. The set of resulting equations is both conservative and positive because all the matrix coefficients involved have non-negative eigenvalues. In order to get U_i^n for all control volumes, one formally needs to solve a set of nonlinear algebraic equations iteratively. The iterations needed to solve this system of nonlinear equations, at each time step, can make the ability to take larger time steps useless.

4 Convergence Acceleration of Steady-State Calculations

4.1 Steady-State Algorithm

For steady-state calculations, to obtain an efficient implicit algorithm, we linearize the nonlinear system (10) and use new unknowns

$$\delta U_i = U_i^n - U_i^{n-1} \tag{11}$$

Then, substituting (11) into (10), and using a first order approximation for the source terms

$$S(U_i^n) = S(U_i^{n-1}) + \frac{\partial S}{\partial U}(U_i^{n-1})\,\delta U_i + \mathcal{O}(\Delta t^2) \tag{12}$$

we get the following linear system

$$\begin{gathered}\left(\mathrm{I} + \Delta t \sum_k C_k^{n-1} - \Delta t \frac{\partial S}{\partial U}(U_i^{n-1})\right)\delta U_i - \Delta t \sum_k C_k^{n-1}\delta U_k = \\ \Delta t S(U_i^{n-1}) - \Delta t \sum_k F_{nv}(U_i^{n-1}, U_k^{n-1})\end{gathered} \tag{13}$$

Once the boundary conditions and the dissipation terms have been added to the linear system (13), it condenses to

$$\mathrm{M}(U^n)\cdot\delta U^n = R(U^n)$$

where $\mathrm{M}(U^n)$ is the matrix containing the implicit operator, $R(U^n)$ is the RHS with source terms and explicit fluxes, and $U^n = (U_i^n)_{i=1,N}$ is the solution vector containing the conservative variables in each control volume Ω_i (N is the number of cells).

The right hand side of (13) contains a conservative discretization of the balance equations of the two-phase flow model. It follows that the steady-state solution obtained when the time variation of conservative variables leads to zero, is consistent with the conservation form and independent of the time step used in the iterations. The linear system (13) can be solved by very efficient methods. The **FLICA-4** computer code provides both a direct method (LDU factorization) and several iterative methods: a conjugate gradient squared (CGS) method with partial preconditionning and the Generalized Minimal Residual (GMRES) algorithm [1].

For steady-state calculations, this implicit scheme allows to take large time steps to reach the steady solution after a pseudo-transient computation. Since variations of the Roe averaged matrix coefficients between two time steps lead to second order variation for the solution, it turns out that it is not necessary to update the matrix of the linear system (13) at each time step to achieve a good convergence towards the same steady solution. Consequently it is possible to significantly reduce CPU time by saving the preconditionned form of the matrix and using it for several time steps.

4.2 Spectrum of the Implicit Operator

Inversion of the matrix $\mathrm{M}(U^n)$ leads to the solution U^{n+1}:

$$U^{n+1} = U^n + \delta U^n = U^n + \mathrm{M}(U^n)^{-1} R(U^n) \tag{14}$$

Equation (14) can be written as an operator equation

$$U^{n+1} = \mathcal{L}(U^n) \quad with \quad \mathcal{L}(U^n) = U^n + \mathrm{M}(U^n)^{-1} R(U^n)$$

The rate of convergence of this iterative algorithm is governed by the largest eigenvalue(s) of the operator $\mathcal{L}$. A study of its spectrum and convergence acceleration will be carried out in order to accelerate convergence of the steady-state algorithm.

The implicit operator $\mathcal{L}$ is a nonlinear transformation of the conservative variables, and it is necessary to know its Jacobian in order to calculate exactly its eigenvalues. Since it is difficult to calculate the Jacobian $\mathcal{A} = \frac{\partial \mathcal{L}}{\partial U}$, an efficient iterative method must be adopted to obtain good approximations of the eigenvalues of $\mathcal{L}$. Arnoldi's algorithm, described below, can be used to reveal a subset of the dominant eigenvalues of $\mathcal{L}$ [6].

Let's start by an arbitrary vector q_1 and define $\hat{q}_1 = \frac{q_1}{\|q_1\|}$

For $k = 1$ to m. do

$$\begin{aligned} q_{k+1} &= \mathcal{A}\hat{q}_k - \sum_{j=1}^{k} c_{jk}\hat{q}_j \quad with \quad c_{jk} = < q_j, \mathcal{A}\hat{q}_k > \\ \hat{q}_{k+1} &= \frac{q_{k+1}}{\|q_{k+1}\|} \end{aligned} \tag{15}$$

Next k

where $< x, y >$ is the inner product of the vectors x and y, and $\|.\|$ is the L_2-norm. In Equation (15) the product $\mathcal{A}\hat{q}_k$ can be calculated using only the operator $\mathcal{L}$ and a Frechet derivative

$$\mathcal{A}\hat{q}_k = \frac{1}{2\epsilon}(\mathcal{L}(U + \epsilon\hat{q}_k) - \mathcal{L}(U - \epsilon\hat{q}_k)) + \mathcal{O}(\epsilon^2)$$

The eigenvalues of the matrix $C = (c_{ij})$ are approximations of some eigenvalues of the Jacobian matrix $\mathcal{A}$. In practice the largest eigenvalues obtained by Arnoldi's method are good approximations to the actual ones.
Another way to obtain approximations of the eigenvales is to linearize the nonlinear operator $\mathcal{L}$ and to calculate exactly its eigenvalues by standard eigenvalues software.

Figures 1. and 3. are plots of typical spectrum for 1D and 3D problems. As expected in the construction of the implicit operator, we remark that all the eigenvalues are positive and that the spectral radius is less than 1.

4.3 Richardson Relaxation Algorithm

Various convergence acceleration techniques have been successfully applied to aerodynamic computations. In this section, application of a Richardson's overrelaxation method to two-phase flow computations is described.
The matrix $\mathrm{M}(U^n)$ is symbolically denoted by

$$\mathrm{M}(U^n) = \mathrm{I} + \Delta t \sum_k C_k^n - \Delta t \frac{\partial S}{\partial U} = \mathrm{I} + \mathrm{G}$$

we assume also that in the RHS of (13) the fluxes and the source terms are both homogeneous of degree one with respect to the conservative variables such that:

$$R(U^n) = \frac{\partial R}{\partial U} U^n = -\mathrm{G}U^n$$

Thus, Equation (13) can be written as

$$(\mathrm{I} + \mathrm{G})\,\delta U^n = -\mathrm{G}U^n$$

The solution U^{n+1} is given by

$$\begin{aligned} U^{n+1} &= U^n - (\mathrm{I} + \mathrm{G})^{-1}\,\mathrm{G}U^n \\ &= (\mathrm{I} + \mathrm{G})^{-1}\,U^n \end{aligned} \tag{16}$$

Let U^* be the exact steady solution and define the error vector e^n at time $n\Delta t$ by $e^n = U^n - U^*$. Then it follows from (16) that

$$e^{n+1} = (\mathrm{I} + \mathrm{G})^{-1}\,e^n = \mathrm{M}^{-1}e^n \tag{17}$$

The residual $\delta U^n = U^{n+1} - U^n$ satisfy the same equation

$$\delta U^{n+1} = \mathrm{M}^{-1}\delta U^n \tag{18}$$

These equations show that the errors and the residual propagate according to the powers of the matrix M^{-1}. Consequently the spectral radius of the matrix M^{-1} controls the convergence of the steady-state algorithm. Let ω be a relaxation factor, such that

$$\begin{aligned} U^{n+1} &= U^n + \omega \delta U^n \\ &= U^n - \omega (\mathrm{I} + \mathrm{G})^{-1} \mathrm{G} U^n \\ &= \left(\mathrm{I} - \omega (\mathrm{I} + \mathrm{G})^{-1} \mathrm{G} \right) U^n \end{aligned} \tag{19}$$

defines a new relaxed algorithm. We call this a Richardson relaxation step with parameter ω. Now errors obey the equation

$$\begin{aligned} e^{n+1} &= \left(\mathrm{I} - \omega (\mathrm{I} + \mathrm{G})^{-1} \mathrm{G} \right) e^n \\ &= \left((1 - \omega) \mathrm{I} + \omega \mathrm{M}^{-1} \right) e^n \\ &= \mathrm{M}_\omega^{-1} e^n \end{aligned}$$

If λ_ω denotes the eigenvalues of the new iteration scheme and λ denotes the eigenvalues corresponding to the original one, then they are related by

$$\lambda_\omega = (1 - \omega) + \omega \lambda$$

The transformation that maps λ to λ_ω is linear. Thus the left and right eigenvalues of M^{-1} are mapped into the left and right eigenvalues of M_ω^{-1}. Moreover we can show the following result.

Let λ_{min} and λ_{max} be the smallest and the largest eigenvalues of M^{-1}. Assume λ_{min} and λ_{max} real. Then

$$\omega_0 = \frac{2}{2 - \lambda_{min} - \lambda_{max}}$$

is the optimal relaxation factor, that is the spectral radius of $\mathrm{M}_{\omega_0}^{-1}$ is minimum.

In order to apply the relaxation method with $\omega = \omega_0$, the eigenvalues λ_{min} and λ_{max} must be know. Assume the eigenvalues of M^{-1} ordered $|\lambda_1| = |\lambda_{max}| \geq |\lambda_2| \geq \cdots \geq |\lambda_N| = |\lambda_{min}|$. Therefore, from (18), the residual can be written in the form

$$\delta U^n = \lambda_1^n m_1 + \lambda_2^n m_2 + \cdots$$

where $(m_i)_{i=1,N}$ depend on the eigenvector of M^{-1} and the initial vector. Since the eigenvalues are ordered, as the number of iterations increases the number of eigenvalues that significantly contribute to the residual decreases. Then, in the limit $n \to \infty$, the residual behaves like

$$\delta U^n \approx \lambda_{max}^n m_1 \tag{20}$$

Figure 3. illustrates this type of log-linear convergence as the number of iterations becomes large. This log-linear limit behavior, coresponding to the dominant eigenvalue is a fairly common convergence model for both linear and nonlinear iteration schemes. From (20) the value of λ_{max}^n can be approximated by the ratio of the residual at two consecutive iterations, that is

$$\lambda_{max}^n \approx \frac{||\delta U^{n+1}||}{||\delta U^n||} \tag{21}$$

The smallest eigenvalue is close to zero for an implicit algorithm based on Roe's scheme. Then, λ_{min}^n is simply set to zero, since there is no easy way by which to approximate λ_{min}^n.

4.4 Eigenvalue Annihilation

In the later steps of the marching algorithm for steady-state calculations, the iterations may behave linearly. If so, one can estimate the dominant eigenvalue of the Jacobian matrix and annihilate the dominant eigenvector component of the error by a special iteration step.

Suppose now that $(\mathbf{I}+\mathbf{G})^{-1}$ is diagonalizable with real eigenvalues λ_i and corresponding eigenvectors r_i. Take $(|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_N|)$ and $e^0 = \sum_{i=1}^{N} \gamma_i r_i$. Then from (17)

$$\begin{aligned} e^n &= \sum_{i=1}^{N} \gamma_i \lambda_i^n r_i \\ &= \lambda_1^n \left(\gamma_1 r_1 + \sum_{i=2}^{N} \gamma_i \left(\frac{\lambda_i}{\lambda_1}\right)^n r_i \right) \end{aligned}$$

If $|\lambda_1 > |\lambda_2|$, the slowest decreasing component of the error is the component in the direction of r_1. To annihilate this component we consider a Richardson step with a parameter ω. Then we have

$$\begin{aligned} e^{n+1} &= \left((1-\omega)\mathbf{I} + \omega \mathbf{M}^{-1}\right) e^n \\ &= \sum_{i=1}^{N} \gamma_i \lambda_i^n \left((1-\omega) + \omega\lambda_i\right) r_i \\ &= \mathbf{M}_\omega^{-1} e^n \end{aligned}$$

If we choose $\omega = \frac{1}{1-\lambda_1}$, the component of e^{n+1} in the direction of r_1 is zero. Thus the relaxed solution (19) converges according to the second largest eigenvalue and not $\lambda_1 = \lambda_{max}$, which can be considerably faster. However, the rate of convergence of the method will eventually revert to λ_1 due to nonlinear effects.

For 3D problems where there are a lot of eigenvalues close to 1, this method does not perform well. In this case, we can use a group annihilation method where the error asociated with a number of eigenvalues is corrected at the same time. For example, if there is a dominant pair of complex conjugate eigenvalues, say $\lambda_1 = \bar{\lambda}_2$ then two Richardson steps are needed. The algorithm can be written as follow [2]:

1. $U^{n+1} = U^n + \frac{|\omega_1|^2}{2Re(\omega_1)} \delta U^n$
2. $\delta U^{n+1} = -(\mathbf{I}+\mathbf{G})^{-1} \mathbf{G} U^{n+1}$
3. $\delta U^{n+2} = U^n + 2Re(\omega_1)\delta U^{n+1}$

where $\omega_1 = \frac{1}{1-\lambda_1}$ and $Re(\omega_1)$ is the real part of ω_1.

5 Preliminary Results

5.1 1D Tests

In this section we will consider a one dimensional two-phase flow in a channel. The boundary conditions are an imposed mass flow rate at the inlet and a constant pressure at the outlet. The 1D problem is small enough that all the eigenvalues of the linearized operator can be calculated by a standard eigenvalue software.

Figure 1. shows the eigenvalues spectrum. The dominant eigenvalue is real and equal to $\lambda_{max} = 0.956$. The smallest eigenvalue is close to zero. First, we ran the implicit algorithm for 300 time steps without any relaxation. The resulting L_2-norm of the residual is shown by the solid line in Figure 2. Next, the Richardson acceleration algorithm was employed, with λ_{max} estimated by (21). The results are shown by the dotted line in Figure 2.

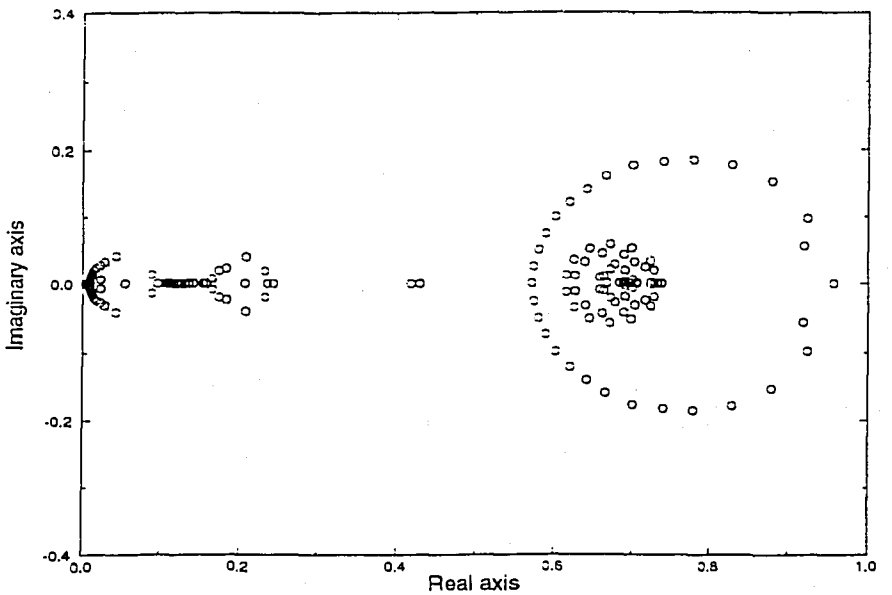

Figure 1: eigenvalues spectrum for a 1D problem

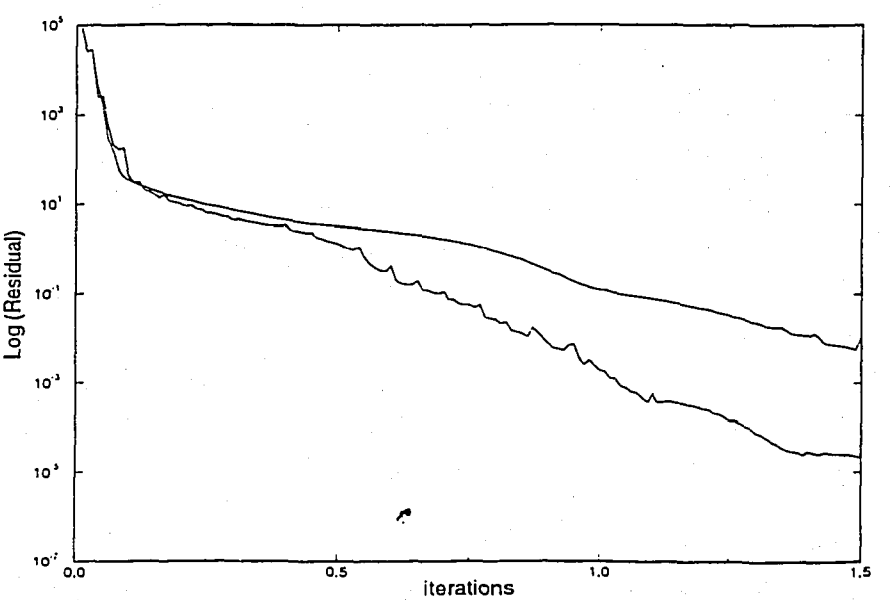

Figure 2: History of residuals, 1D problem

5.2 1/8 PWR Core steady state calculations

The second example deals with a steady state calclations of 1/8 PWR core (26 assemblies). The mesh is composed of 806 elementary cells. The steady state is computed with the algorithm previously described. Figure 3. shows the eigenvalues spectrum of the linearized operator. The dominant eigenvalue is real and equal to $\lambda_{max} = 0.996$. Figure 4. compares the history of residuals for the steady-state algorithm without relaxation and the overrelaxation method.

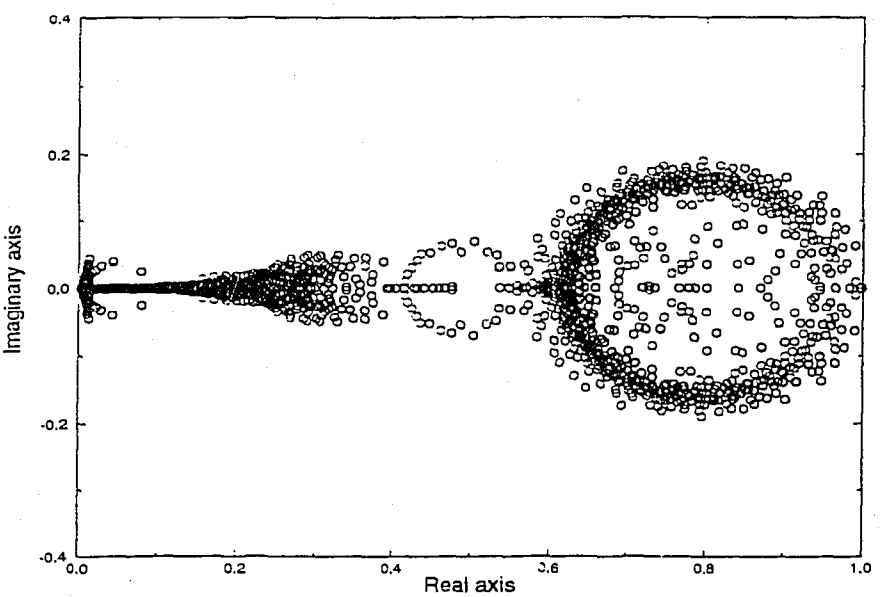

Figure 3: eigenvalues spectrum for a 3D problem

References

[1] ALLAIRE, G. Flica-4 version 1.0 : manuel de référence des modules de résolution des systèmes linéaires.. Rapport DMT 92/536, CEA, 1992.

[2] DAVID A. SMITH, W. F. F., AND SIDI, A. Extrapolation methods for vector sequences. *SIAM Review 29*, 2 (June 1987), 199–233.

[3] GODUNOV, S. K. *Math Sb. 47* (1959), 217.

[4] JESPERSEN, D. C., AND BUNING, P. G. Accelerating an iterative process by explicit annihilation. *SIAM J. Sci. Stat. Comput. 6*, 3 (July 1985), 639–651.

[5] ROE, P. L. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics 43* (1981), 357–372.

[6] SAAD, Y. Variations on Arnoldi's method for computing eigenvalues of large unsymmetric matrices. *Linear Algebra and Its Applications 23* (1980), 269–295.

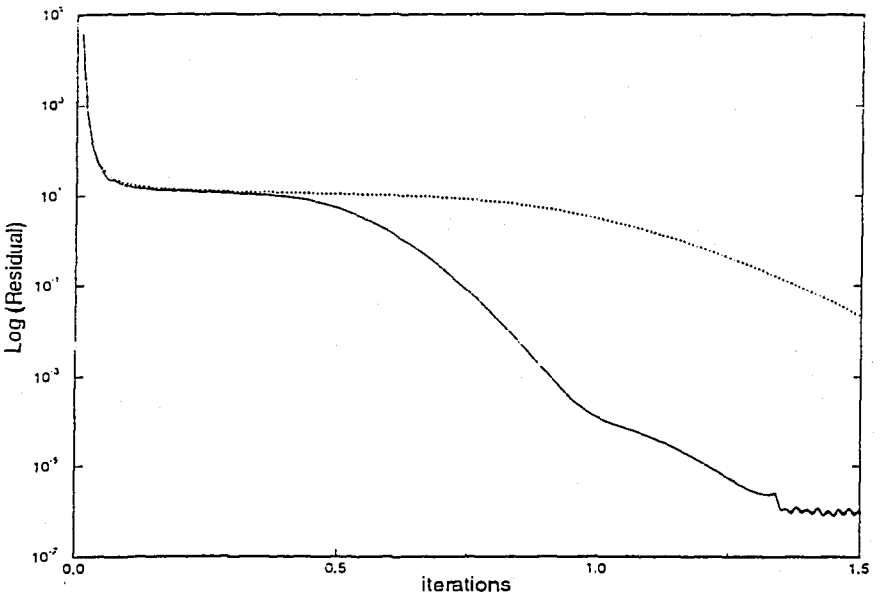

Figure 4: History of residuals. 1/8 PWR core claculation

[7] TOUMI, I. An upwind numerical method for a six equation two-fluid model. Sub. for pub. to Nuclear Science and Engineering.

[8] TOUMI, I. A weak formulation of Roe's approximate riemann solver. *Journal of Computational Physics 102* (1992), 360–373.

[9] TOUMI, I. Flica-4 version 1.0 : manuel de référence de la méthode numérique. Rapport DMT 93/439, SERMA 93/1533, CEA, 1993.

[10] TOUMI, I., AND KUMBARO, A. A linearized approximate riemann solver for a two-fluid model. Sub. for pub. to J. Comput. Phys.