Federal University of Espírito Santo
Technical Report - REL-LCAD-03/2012

Globalization and forcing term effects on the Inexact Newton-Krylov method for solving nonlinear equations emanating from the stabilized finite element simulation of free-surface problems

Lucia Catabriga, José Camata, Alvaro Coutinho and Norminda Bodart

November 9, 2012

Abstract

In this work we evaluate the performance of the Inexact Newton-Krylov method to solve the nonlinear equations arising from the SUPG/PSPG stabilized finite element formulation of transient incompressible fluid flows considering several forcing terms. We consider the free surface flow on a dam break as a benchmark problem for Navier-Stokes equations. A simple backtracking scheme also contributes to improve convergence and to reduce computer time, without compromising accuracy for all forcing terms strategies studied in the present work.

Chapter 1

Introduction

The discretization of the incompressible Navier-Stokes equations leads us to a system of nonlinear equations due the presence of convective terms in the momentum equation. Newton-type schemes are often the most used strategy to solve this system of nonlinear equations at each time step because they can provide superlinear and even quadratic convergence rapidly from any sufficiently good initial guess [7, 21]. The Newton-type algorithm requires the solution of linear systems at each iteration and exact solutions can be too expensive if the number of unknowns is large. However, the computational effort to find exact solutions for the linearized systems may not be justified when the nonlinear iterates are far from the solution. The use of an iterative Krylov method to solve these linear systems is the strategy generally adopted.

When iterative Krylov methods are used to solve the linearized system of the Newton-type scheme, the resulting methods are known as Inexact Newton-Krylov methods (INK). They have been used to reduce the computational effort related to nonlinearities in many problems of computational fluid dynamics, offering a compromise between the accuracy and the amount of effort spent per iteration. The tolerance to which the linearized system is solved, known as the forcing term, plays an instrumental role in the numerical performance of this method. INK success depends mainly on three factors: (*i*) quality of initial Newton step, (*ii*) robustness of Jacobian evaluation and (*iii*) proper forcing term choice [21].

A naive selection criterion for the forcing term can lead the method to successive oversolving stages. Many works have explored practical mechanisms to adaptively choose the forcing term [7, 24, 10, 1, 17]. They often set up their approaches based on the reduction in the Euclidean norm of the nonlinear residual. The main objective is to find out what level of accuracy is required to preserve the rapid convergence of the Newton's method.

One of the first works about this theme is by [7]. They proposed what now we call the Inexact Newton method, a slight, but very insightful and useful variation of the Newton's method, including a mathematical demonstration on how a suitable sequence of the forcing term can provide a fast convergence. An earlier forcing term is described in [24] and has been used in several works, e.g., [26, 6, 5].

Another important consideration described by [10] and also discussed by [21] is oversolving, that is, a choice too small of the forcing term may, at times, increase the number of inner iterations without guaranteeing significant reduction in the residual norm. [10] introduced two choices for the forcing term, considering oversolving control, and provided convergence results for both choices.

[1] described a new strategy to choose forcing terms, that consider the ratio of actual reduction to predicted reduction of the residual norm. With the new forcing terms, the Inexact Newton method is locally Q-superlinearly convergent. [17] also described another way to choose the forcing term, that takes into account the number of times the residual norm is evaluated. For a set of simple experiments, the forcing term described seems to be superior to the other choices (e.g., the ones introduced by [10]), because it is faster in terms of number of outer iterations, number of function evaluations and CPU time.

To enhance the robustness of these methods, they can be augmented with a suitable globalization strategy, i.e., auxiliary procedures that increase the convergence to the solution when good initial approximate solutions are not available. There are two major categories of globalizations: backtracking methods and trust-region methods [21]. Both methods have strong theoretical support, that can be seen in [8, 9]. The backtracking technique usually shortens steps as necessary to ensure adequate decrease in the residual norm of the nonlinear system. On the other hand, the trust-region methods are techniques where a step is ideally chosen to minimize the residual norm of a local linear model defined from the nonlinear system within a specified "trust region" [25]. Many works studies globalization procedures in Newton-Krylov methods to solve nonlinear systems arising from the solution of fully coupled large scale CFD problems [3, 25, 32, 12].

In this work we evaluate the performance of the Inexact Newton-Krylov scheme to solve the nonlinear equations arising from the SUPG/PSPG finite element formulation of transient incompressible fluid flows [27, 13, 14] with the backtracking global strategy and considering several forcing terms proposals.

The present paper is organized as follows. Section 2 presents the governing equations and the SUPG/PSPG finite element formulation. Section 3 describes the Inexact Newton-type schemes including forcing terms calculations, the backtracking global strategy and one illustrative simple example in one-dimension. Three-dimensional test problems are presented in Section 4 and the paper ends with a summary of our main conclusions.

Chapter 2

Governing Equations and Finite Element Formulation

Given a spatial domain $\Omega \subset R^3$, bounded by a piecewise regular surface and a time interval $[0, t_f]$, the motion of an incompressible, viscous and Newtonian fluid in this region is governed by the following equations:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \sigma = \mathbf{f} \quad \text{on } \Omega \times (0, t_f) \tag{2.1}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \quad \text{on } \Omega \times (0, t) \tag{2.2}$$

where $\mathbf{u}$ is the velocity field, ρ is the fluid density, $\mathbf{f}$ is a vector body force acting on the fluid and σ is the stress tensor, that for Newtonian fluids is given by

$$\sigma(p, \mathbf{u}) = -p\mathbf{I} + 2\mu\epsilon(\mathbf{u}) \tag{2.3}$$

with $\mathbf{I}$ the identity tensor, μ the viscosity and $\epsilon(\mathbf{u})$ the strain rate tensor, defined as $\epsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla^T \mathbf{u})$.

The Navier-Stokes system formed by Eqs. (2.1), (2.2) and (2.3) consisting of the momentum, continuity and constitutive equations, along with appropriate boundary and initial conditions contains all information necessary to model incompressible fluid flows, including turbulent effects.

Let us assume that we have some suitably defined finite-dimensional trial solution and test function spaces for velocity and pressure, $S_{\mathbf{u}^h}$, $V_{\mathbf{u}^h}$, S_p^h and $V_p^h = S_p^h$. The finite element formulation of Equations (2.1) and (2.2) using SUPG and PSPG stabilizations for incompressible fluid flows can be written (see [31]) as follows: find $\mathbf{u_h} \in S_{\mathbf{u}^h}$ and $p_h \in V_{\mathbf{u}^h}$ such that $\forall \mathbf{w_h} \in V_{\mathbf{u}^h}$ and $\forall q_h \in V_p^h$

$$
\begin{aligned}
\int_\Omega \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}_h}{\partial t} + (\mathbf{u}^h \cdot \nabla)\mathbf{u}^h - \mathbf{f} \right) d\Omega + \int_\Omega \epsilon(\mathbf{w}^h) {:} \sigma(p^h, \mathbf{u}^h) d\Omega \\
- \int_\Gamma \mathbf{w}^h \cdot \mathbf{h} d\Gamma + \int_\Omega q^h \nabla \cdot \mathbf{u}^h d\Omega \\
+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \frac{1}{\rho} \left(\tau_{SUPG} \rho \mathbf{u}^h \cdot \nabla \mathbf{w}^h + \tau_{PSPG} \nabla q^h \right) \\
\cdot \left[\rho \left(\frac{\partial \mathbf{u}_h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) - \nabla \cdot \sigma(p^h, \mathbf{u}^h) - \rho \mathbf{f} \right] d\Omega \\
+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_{LSIC} \nabla \cdot \mathbf{w}^h \rho \nabla \cdot \mathbf{u}^h d\Omega = \mathbf{0}
\end{aligned}
\tag{2.4}
$$

In the above equation the first four integrals on the left-hand side represent terms that appear in the Galerkin formulation of problem (2.1)–(2.3), while the remaining integral expressions represent the additional terms which arise in the stabilized finite element formulation. Note that the stabilization terms are evaluated as the sum of element-wise integral expressions, where n_{el} is the number of elements in the mesh. The first summation corresponds to the SUPG term and the second to the PSPG term. Stabilization prevents spurious node-to-node oscillations in the velocity and pressure fields, respectively. We have evaluated the SUPG and PSPG stabilization parameters according to [30]. In Equation (2.4), the last summation is the least-squares incompressibility constraint (LSIC) term, added to prevent oscillations in high Reynolds number flows, as defined in [28]. The discretization of Equation (2.4) leads us to a nonlinear ordinary differential system of equations that can be written as

$$
\begin{aligned}
\mathbf{M}\dot{\mathbf{u}} + \mathbf{M}_\delta(\mathbf{u})\dot{\mathbf{u}} + \mathbf{N}(\mathbf{u}) + \mathbf{N}_\delta(\mathbf{u}) + \mathbf{K}\mathbf{u} - (\mathbf{G} + \mathbf{G}_\delta)\mathbf{p} = \mathbf{f}_\mathbf{u} \\
\mathbf{M}_\varphi(\mathbf{u})\dot{\mathbf{u}} + \mathbf{G}^T\mathbf{u} + \mathbf{N}_\varphi(\mathbf{u}) + \mathbf{G}_\varphi \mathbf{p} = \mathbf{f}_\mathbf{p}
\end{aligned}
\tag{2.5}
$$

where $\mathbf{u}$ is the vector of unknown nodal values of $\mathbf{u}_h$ and $\mathbf{p}$ is the vector of unknown nodal values of p_h. The non-linear vectors $\mathbf{M}_\delta(\mathbf{u})$, $\mathbf{M}_\varphi(\mathbf{u})$, $\mathbf{N}(\mathbf{u})$, $\mathbf{N}_\delta(\mathbf{u})$ and $\mathbf{N}_\varphi(\mathbf{u})$, the matrices $\mathbf{M}$, $\mathbf{K}$, $\mathbf{G}$, $\mathbf{G}_\delta$ and $\mathbf{G}_\varphi$ emanate, respectively, from the temporal, convective, viscous and pressure terms. The vector $\mathbf{f}_\mathbf{u}$ includes boundary conditions, the vector $\mathbf{f}$ and stabilization terms, $\mathbf{f}_\mathbf{p}$ also includes boundary conditions and stabilization terms. The subscripts δ and φ identify the SUPG and PSPG contributions respectively. The discretization in time of Equation (2.5) using the predictor multicorrector finite difference scheme – described by [15] – leads us to a nonlinear system of equations to be solved at each time step that can be written as,

$$\mathbf{F}(\mathbf{x}) = \mathbf{0} \tag{2.6}$$

where $\mathbf{x} = (\mathbf{u}, p)$ is denoted by a vector of nodal variables comprising both nodal velocities and pressures. We consider the approximate Jacobian form described by [27] and used in [13]. This numerically approximated Jacobian is based on Taylor's expansions of the nonlinear terms in Eq. (2.5), and presents an alternative and simple way to implement the approximate tangent matrix employed by Inexact Newton-type methods to solve the nonlinear system (2.6).

We are also interested in solving free-surface problems using an interface capturing method. In this kind of problem, the interface can be represented by a scalar function which delimits the regions filled with the fluids involved. In other words, the interface position is implicitly captured in a scalar marking function value and the interface evolution is determined by the additional cost of solving an advection equation for the marker. In this work, we consider the volume-of-fluid (VOF) method, that was developed by [19] for finite differences and applied for finite elements for instance, in [29] and [11]. This method is an interface capturing technique, that consider a scalar marking function $\phi = \phi(\mathbf{x}, t)$ defined over the computational domain. In air-water flows, this marking function can be used to define the regions occupied by air, by water or a fraction of these two fluids, that can be used to compute the free surface position and some physical properties of partially filled elements. The fluid density ρ and viscosity μ, in a fixed position, can be computed as

$$\begin{aligned} \rho &= \phi\rho_a + (1-\phi)\rho_w \\ \mu &= \phi\mu_a + (1-\phi)\mu_w \end{aligned} \tag{2.7}$$

where the sub-indexes a and w stands, respectively, for air and water. The marking function $\phi = \phi(\mathbf{x}, t)$, along the time interval $[0, t_f]$ in an open, bounded spatial domain $\Omega \subset R^3$, considering a given continuous velocity field $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$, is governed by the following hyperbolic differential equation

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \;\; \text{on} \;\; \Omega \times [0, t_f] \tag{2.8}$$

One more time, let us assume that we have a suitably standard test and weight finite element spaces, respectively, Z^h and Q^h. The finite element formulation of Eq. (2.8) using SUPG stabilization enriched with the nonlinear discontinuity-capturing term can be written as follows: find $\phi_h \in Z^h$ such that $\forall \psi_h \in Q^h$

$$\begin{aligned}
&\int_\Omega \psi_h \left(\frac{\partial \phi_h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi_h + \phi_h \nabla \cdot \mathbf{u}^h \right) d\Omega \\
+\sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_{SUPG_M} \mathbf{u}^h \cdot \nabla \psi_h &\left(\frac{\partial \phi_h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi_h + \phi_h \nabla \cdot \mathbf{u}^h \right) d\Omega \\
&\sum_{e=1}^{n_{el}} \int_{\Omega^e} \delta_M \nabla \psi_h \cdot \nabla \phi_h d\Omega = 0
\end{aligned} \tag{2.9}$$

where $\mathbf{u}^h$ is given by the solution of the nonlinear system (2.6) for each time step. The evaluation of τ_{SUPG_M} and δ_M stabilization terms follows the definitions described in [4, 16]. The discretization of the Eq. (2.9) leads us to an ordinary differential equation system in time, that is solved by the well known implicit predictor-multicorrector algorithm [20]. Following [11], we consider a global mass conservation algorithm to enforce that the mass of the species involved are correctly represented as the solution evolves. The computational effort to solve the marking function is limited to a narrow band around the free surface by a parallel dynamic-deactivation (PDD) scheme. Details for all formulation can be found in [11].

Chapter 3

Inexact Newton-Krylov Method

The nonlinear system (2.6) can be solved by Newton's method. It is an iterative method for nonlinear equations that approximate the function $\mathbf{F}$ at a given point $\mathbf{x} = (x_1, x_3, \dots, x_N)^t$ by a linear function. The Jacobian matrix $\mathbf{J}$ represents the variation of the function $\mathbf{F}$ with respect of $\mathbf{x}$. Each iteration of the Newton's method is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{s}^k, \tag{3.1}$$

where $\mathbf{s}^k$ is calculated by the solution of the linear system:

$$\mathbf{J}(\mathbf{x}^k)\mathbf{s}^k = -\mathbf{F}(\mathbf{x}^k). \tag{3.2}$$

We may terminate the iteration when the relative nonlinear residual $\|\mathbf{F}(\mathbf{x}^k)\|_2/\|\mathbf{F}(\mathbf{x}^0)\|_2$ is small, i.e. when $\|\mathbf{F}(\mathbf{x}^k)\|_2 < \tau_{NL} = \tau_{res}\|\mathbf{F}(\mathbf{x}^0)\|_2$, for a given tolerance τ_{res} .

When an iterative method is used to solve the system (3.2) the Newton's method is known as the Inexact Newton method, that is especially well suited for large-scale problems and have been used very successfully in many applications. The main idea is to give up precision in name of performance. In this context, it is introduced the concept of forcing term, expressed by η_k, that represent the tolerance of the inner iterative linear method on nonlinear iteration k. Several works have explored practical mechanisms to adaptively choose the forcing term. They often set up their approaches based on the reduction in the Euclidean norm of the nonlinear residual. In Section 3.1 we discuss some choices for the forcing terms. We also can improve the updated solution $\mathbf{x}^{k+1}$ on Eq. (3.1), if we consider $\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda\mathbf{s}^k$, where $\lambda > 0$ is a suitable parameter. Section 3.2 shows a procedure to calculate this parameter, that shortens steps as necessary to ensure adequate decrease in the residual norm of the nonlinear system. This strategy, known as backtracking, is widely used [21].

3.1 Forcing terms definitions

Let η_0 be a given maximum initial tolerance for the Inexact Newton method. The well known forcing term, introduced by [24], is defined by

$$\eta_k^{PP} = \min\{\eta_0, \left(\frac{\|\mathbf{F}(\mathbf{x}^k)\|_2}{\|\mathbf{F}(\mathbf{x}^{k-1})\|_2}\right)^{\lambda}\} \tag{3.3}$$

with the parameter $\lambda = 2$.

[10] introduced a new forcing term that was also described by [21]. Considering

$$\eta_k^{A} = \gamma \left(\frac{\|\mathbf{F}(\mathbf{x}^k)\|_2}{\|\mathbf{F}(\mathbf{x}^{k-1})\|_2}\right)^{\alpha} \tag{3.4}$$

where γ and α are given parameters, the forcing term can be evaluated as

$$\eta_k^{EW*} = \begin{cases} \eta_0 & k = 0 \\ \min\{\eta_0, \eta_k^A\} & k > 0, \end{cases} \tag{3.5}$$

It may happen that η_k^{EW*} is small for one or more iterations while $\mathbf{x}^k$ is still far from the solution. A method of safeguarding against this possibility was suggested by [10] to avoid volatile decreases in η_k^{EW*}. The idea is that if η_{k-1}^{EW*} is sufficiently large we do not let η_k^{EW*} decrease by much more than a factor of η_{k-1}^{EW*}, that is

$$\eta_k^{EW*} = \begin{cases} \eta_0 & k = 0, \\ \min\{\eta_0, \eta_k^A\} & k > 0, \gamma(\eta_{k-1}^{EW*})^{\alpha} < \beta, \\ \min\{\eta_0, \max\{\eta_k^A, \gamma(\eta_{k-1}^{EW*})^{\alpha}\}\} & k > 0, \gamma(\eta_{k-1}^{EW*})^{\alpha} > \beta. \end{cases} \tag{3.6}$$

In this work we consider two forcing terms defined by Eq. (3.6). One described by [21]—named here as η_k^{EWK}— considers $\alpha = 2$, $\gamma = 0.9$ and $\beta = 0.1$, and another defined in the PETSc Library [2]—named here as η_k^{EWC}— considers $\alpha = \dfrac{1+\sqrt{5}}{2}$, $\gamma = 1.0$ and $\beta = 0$. It is important to note that, both parameters in EWC and EWK were previously defined as an option by [10].

Recently, [17] defined a new choice of the forcing term that depended on both the change in $\|\mathbf{F}(\mathbf{x}^k)\|$ and the computational cost invested during the kth nonlinear iteration, including inner iterations. They defined the cost ($\texttt{price}_k$) as the number of iterations performed by the linear solver ($\texttt{iter}_k$) plus the number of function evaluations ($\texttt{feval}_k$), that is, $\texttt{price}_k = \texttt{iter}_k + \texttt{feval}_k$. Note that both $\texttt{iter}_k$ and $\texttt{feval}_k$ represent the total number of inner iterations and the total number of function evaluations performed during the first k nonlinear iterations. They also studied a way to control the angle of the decrease on $\|\mathbf{F}\|$ as follows. If θ_k is the slope coefficient on the iteration k of $\|\mathbf{F}\|$, $cos(\theta_k)$ is a good measure for the tradeoff between convergence and computational costs and can be described as the ratio:

$$cos(\theta_k) = \frac{b_k}{\sqrt{a_k^2 + b_k^2}} \tag{3.7}$$

where

$$\begin{aligned} a_k &= (log_{10}\|\mathbf{F}(\mathbf{x}^k)\|_2 - log_{10}\|\mathbf{F}(\mathbf{x}^{k-1})\|_2 \\ b_k &= log_{10}(\texttt{price}_k - \texttt{price}_{k-1}) \end{aligned} \tag{3.8}$$

Note that $\theta \in (-\pi/2, \pi/2)$. If $cos(\theta_k)$ is close to -1 the process is doing fine and a stricter forcing term may be tried. If $cos(\theta_k)$ is close to zero, the iterations are either too costly or are getting nowhere (oversolving) and the forcing term has to be relaxed. If $cos(\theta_k)$ is positive, $\|\mathbf{F}(\mathbf{x}^k)\|$ has actually increased and a drastic action is necessary. Let us consider

$$\eta_k^A = \left(\frac{1}{k+1}\right)^{\nu} [cos(\theta_k)]^2 \left(\frac{\|\mathbf{F}(\mathbf{x}^k)\|_2}{\|\mathbf{F}(\mathbf{x}^{k-1})\|_2}\right) \tag{3.9}$$

where $\nu \in (1, 2]$ and we adopted $\nu = 1.1$. Here, this forcing term is named GLT and is defined by

$$\eta_k^{GLT} = \begin{cases} \eta_0 & k = 0 \\ \min\{\eta_0, \eta_k^A\} & k > 0, \end{cases} \tag{3.10}$$

As described by [21], there is a chance that the final iterate will reduce $\|\mathbf{F}(\mathbf{x}^k)\|$ far beyond the desired level and that the cost of the solution of the linear equation for the last step will be higher than is really needed. This oversolving in the final step can be controlled by comparing the norm of the current nonlinear residual to the nonlinear norm at which the iteration would terminate (τ_{NL}) and bounding η_k^* by a constant multiple of $\tau_{NL}/\|\mathbf{F}(\mathbf{x}^k)\|$, that is

$$\eta_k^* = \min\{\eta_0, \max\{\eta_k^*, \rho \quad \tau_{NL}/\|\mathbf{F}(\mathbf{x}^k)\|_2\}\} \tag{3.11}$$

We consider in our experiments $\rho = 0.5$ and the super index $*$ can be PP, EWK, EWC and GLT. In this work, we use those four schemes for choosing adaptively tolerances for the inner iterative method in each Inexact Newton iteration.

3.2 The Backtracking Strategy

The backtracking strategy idea is quite simple and can be defined as follows. Given a descent direction $\mathbf{s}^k$, we take a step in that direction that yields acceptable $\mathbf{x}^{k+1}$, that is:

(i) calculate a descent direction $\mathbf{s}^k$;

(ii) set $\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda \mathbf{s}^k$ for some $\lambda > 0$ that makes $\mathbf{x}^{k+1}$ an acceptable next iterate.

The backtracking strategy is also referred as a line search strategy, that is, a procedure to choose λ in (ii). Until the mid 1960s the prevailing belief was that λ should be chosen to solve the one-dimensional minimization problem accurately. After that, careful computational tests has led to a complete turnaround [8]. In this work we evaluate the performance of the Inexact Newton-Krylov scheme with backtracking as in [21] to solve the nonlinear equations (2.6). We employ the Armijo rule, that proposes to apply only enough reduction, such that, the following condition is satisfied:

$$\|\mathbf{F}(\mathbf{x}^k)\|_2 < (1 - \alpha\lambda)\|\mathbf{F}(\mathbf{x}^{k-1})\|_2 \tag{3.12}$$

where $\alpha \in (0, 1)$. Here we adopted $\alpha = 10^{-4}$ according to [21]. The parameter λ is obtained in order to produced a reduction such that $\sigma_0\lambda_{old} \leq \lambda_{new} \leq \sigma_1\lambda_{old}$, where $0 < \sigma_0 < \sigma_1 < 1$.

Algorithm 1: Inexact Newton-Krylov Backtracking Method - INKB

```
1  Set $\eta_0$;
2  $k = 0$;
3  $\tau_{NL} = \tau_{res}\|\mathbf{F}(\mathbf{x}^k)\|_2$;
4  while $\|\mathbf{F}(\mathbf{x}^k)\|_2 > \tau_{NL}$ do
5      Compute $\mathbf{J}(\mathbf{x}^k)$;
6      Solve $\mathbf{J}(\mathbf{x}^k)\mathbf{s} = -\mathbf{F}(\mathbf{x})$ by a Krylov method with tolerance $\eta_k$ ;
7      Set $\lambda_1 = 1$;
8      Compute $\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda_1\mathbf{s}$;
9      $i = 0$;
10     $\mathbf{x}^t = \mathbf{x}^{k+1}$;
11     while $\|\mathbf{F}(\mathbf{x}^t)\|_2 > (1 - \alpha\lambda_i)\|\mathbf{F}(\mathbf{x}^{k-1})\|_2$ and $i \leq nbt$ do
12         Choose $\lambda_{i+1}$;
13         Update $\mathbf{x}^t = \mathbf{x}^k + \lambda_{i+1}\mathbf{s}$;
14         $i = i + 1$;
15     endw
16     if $i < nbt$ then
17         Update $\mathbf{x}^{k+1} = \mathbf{x}^t$;
18     else
19         backtracking loop rejected ;
20     endif
21     $k = k + 1$;
22     Select $\eta_k$;
23 endw
```

Algorithm 1 shows the Inexact Newton backtracking algorithm implemented. The backtracking globalization steps are in lines 7 until 15, where *nbt* is the maximum number of backtracking steps considered. In this work, we consider a simple line search scheme of reduction of the steplength (λ), that is described in [21]. For each backtracking step i, we can consider a scalar function as:

$$f(\lambda) = \|\mathbf{F}(\mathbf{x}^k + \lambda\mathbf{s}_k)\|_2^2. \tag{3.13}$$

This function defines a polynomial and the minimum of that polynomial is the next steplength. The values of $f(0)$, $f'(0)$ and the value of f at the current value of λ ($f(\lambda_i)$) is used to construct a 2nd degree interpolating polynomial for f. The polynomial $p(\lambda)$ is

$$p(\lambda) = f(0) + f'(0)\lambda + c\lambda^2 \tag{3.14}$$

where $f(0) = \|\mathbf{F}(\mathbf{x}_k)\|_2^2$ is known and $f'(0)$ can be computed as

$$f'(0) = 2(\mathbf{J}(\mathbf{x}^k)^T\mathbf{s}^k)^T\mathbf{F}(\mathbf{x}^k) = 2\mathbf{F}(\mathbf{x}^k)^T(\mathbf{J}(\mathbf{x}^k)\mathbf{s}^k) \tag{3.15}$$

and

$$c = \frac{f(\lambda_i) - f(0) - f'(0)\lambda_i}{\lambda_i^2} \tag{3.16}$$

The minimum of the polynomial (3.14) is

$$\lambda_{min} = \frac{-f'(0)}{2c} > 0 \tag{3.17}$$

Since we are considering Krylov methods to solve the linear system (3.2), the evaluation of $\mathbf{J}(\mathbf{x}^k)\mathbf{s}^k$ on (3.17) is done by examination of the final residual on the linear solver (line 6 on Algorithm 1). Thus, the next steplength λ_{i+1} can be calculated by

$$\lambda_{i+1} = \begin{cases} \sigma_0\lambda_i & \text{if } \lambda_{min} < \sigma_0\lambda_i, \\ \sigma_1\lambda_i & \text{if } \lambda_{min} > \sigma_1\lambda_i, \\ \lambda_{min} & \text{otherwise} \end{cases} \tag{3.18}$$

where the parameter σ_0 guarantees that, the reduction step does not reach values close to zero, thus becoming useless and σ_1 is an upper limit value. The typical values for σ_0 and σ_1 are, respectively, 0.1 and 0.5, as suggested in [21].

Chapter 4

Numerical Experiments

This section presents a three-dimensional benchmark problem for Navier-Stokes equations: the free surface flow on a dam break to compare the forcing terms strategies considering INK and INKB algorithms. The experiments were run on the SGI Altix ICE 8200 installed at the High Performance Computing Center at COPPE/UFRJ.

The collapse of liquid column is a well known problem, widely employed to validate free-surface codes based on interface capturing methods since there are experimental results available (see [23], for details) and several simulation results obtained with different numerical methods [18, 22]. The problem consists of a liquid column initially sustained by a dam which is suddenly removed. The liquid falls under the influence of gravity ($g = 9.81\ m/s^2$), acting vertically, and flows downward until hitting the opposite wall. The model, schematically depicted in Fig. 4.1, is a box with dimensions $4a \times a \times 2.4a$, where a is a parameter, assumed here to be equal to $0.146\ m$, following [22]. The liquid column has dimensions $a \times a \times 2a$. Slip boundary conditions are used on the walls. This set of boundary conditions makes this problem basically two-dimensional until the liquid mass hits the opposite wall, where three-dimensional effects begin to appear. The density of liquid is $\rho_w = 1,000\ kg/m^3$ and the dynamic viscosity $\mu_w = 0.01\ kg/(ms)$. The density of the air is assumed to be $\rho_a = 1\ kg/m^3$ and the dynamic viscosity $\mu_a = 0.0001\ kg/(ms)$.

The accuracy of the solution is accessed in terms of volume loss of the filled region with respect to the initial data. As described in [11], considering the volume of the region completely filled with water, the relative amount of fluid lost/gained at a given time step can be calculated by

$$\tilde{V} = 100 \times \frac{V^{t_n} - V^{t_0}}{V^{t_0}} \tag{4.1}$$

where t_0 and t_n represent the initial and current time steps, respectively, and $\tilde{V}$ is the relative volume fluctuation.

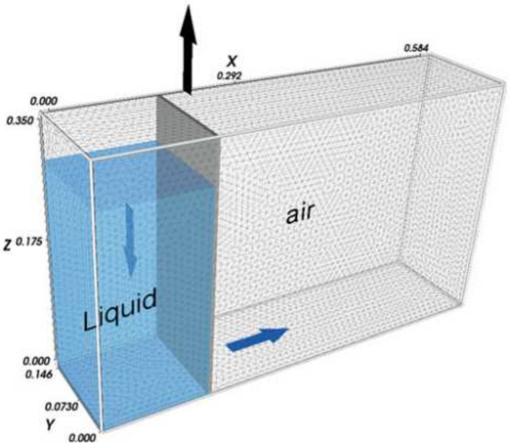


Figure 4.1: Geometric for the collapse of a liquid column – The free surface flow on a dam break

In our experiments we consider the final time equal to 0.3 seconds, a fixed time-step $\Delta t = 0.01$, the edge-based nodal-block diagonal preconditioned GMRES solver with 35 vectors to restart (GMRES(35)), the maximum number of Newton iterations equal to 20, the maximum number ob backtracking iteration equal to 5 and the Newton tolerance 10^{-4} for the Navier-Stokes equations. The VOF solution employs an implicit predictor-multicorrector scheme, with a fixed edge-based diagonal preconditioned GMRES(25) tolerance of 10^{-3}. The multicorrection iterations for the marker function solution are halted after a relative residual decrease of 3 orders of magnitude. The unstructured mesh was built with $46,766$ nodes, $251,807$ linear tetrahedra and consequently $306,597$ edges.

The position of the leading edge of the collapsed water column is shown in Fig. 4.2, where it is plotted the dimensionless leading edge position (x/a) versus the dimensionless time $(t(2g/a)^{1/2})$ for fixed tolerance 10^{-3}, forcing terms EWK, GLT, and PP, considering INK and INKB algorithms together with the results by [18, 23]. As one can see, our results are in good agreement with the experimental and numerical reference data. Moreover, in order to estimate how the forcing terms influence the volume conservation we present in Fig. 4.3 the transient volume loss for all forcing terms, which in all cases volume losses are smaller than 1%.

Figures 4.4 and 4.5 show the liquid configuration at $t = 0.1$, $t = 0.2$ and $t = 0.3$ seconds, respectively, for fixed tolerance 10^{-3} and the forcing term

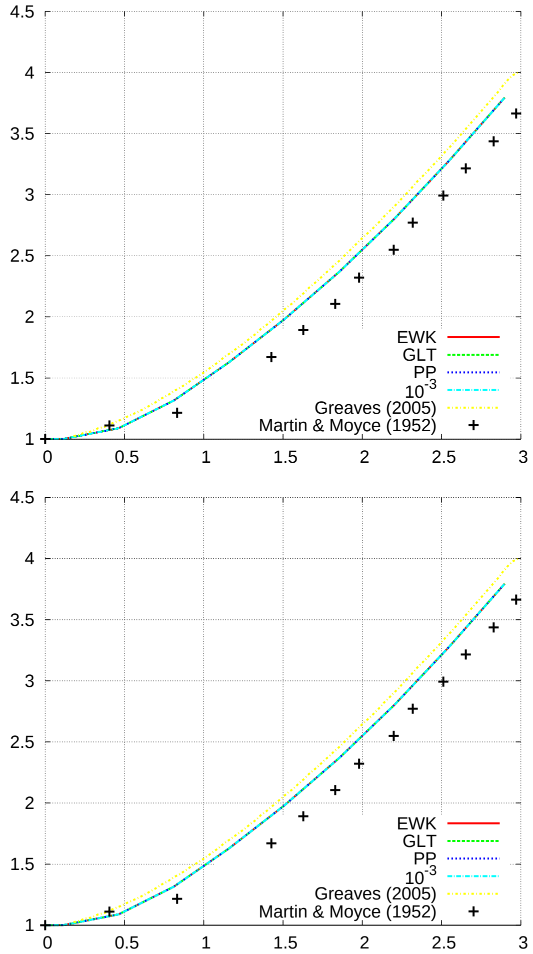


Figure 4.2: Leading Edge position – The free surface flow on a dam break

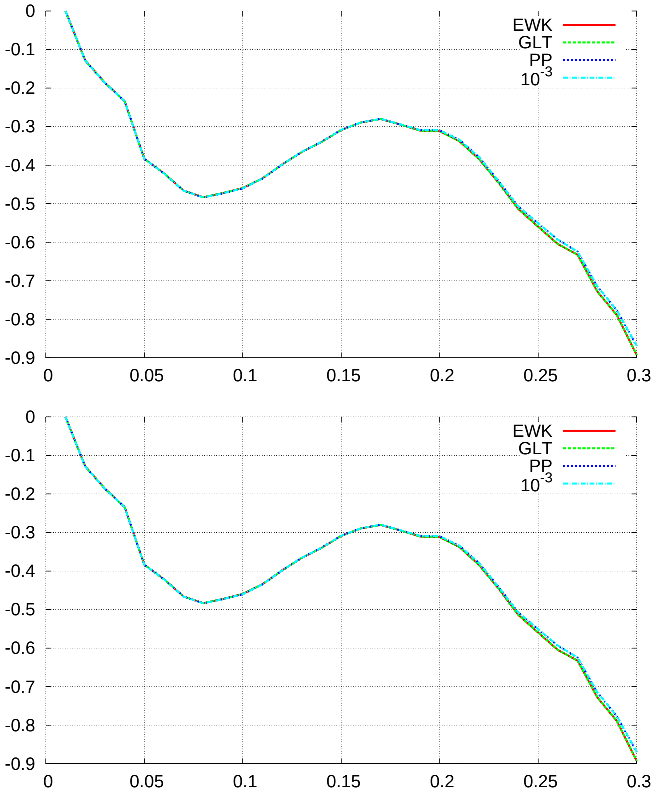
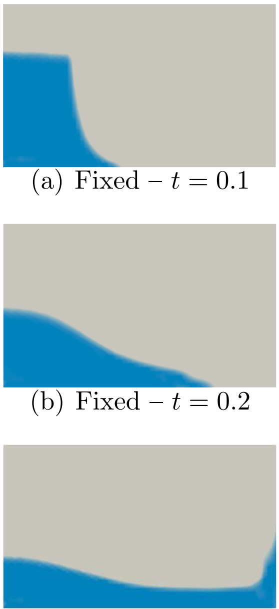


Figure 4.3: Volume loss (%) – The free surface flow on a dam break

EWK, considering INK and INKB algorithms. In general, the results are in good agreement with the result presented by [22, 11].



(a) Fixed – $t = 0.1$

(b) Fixed – $t = 0.2$

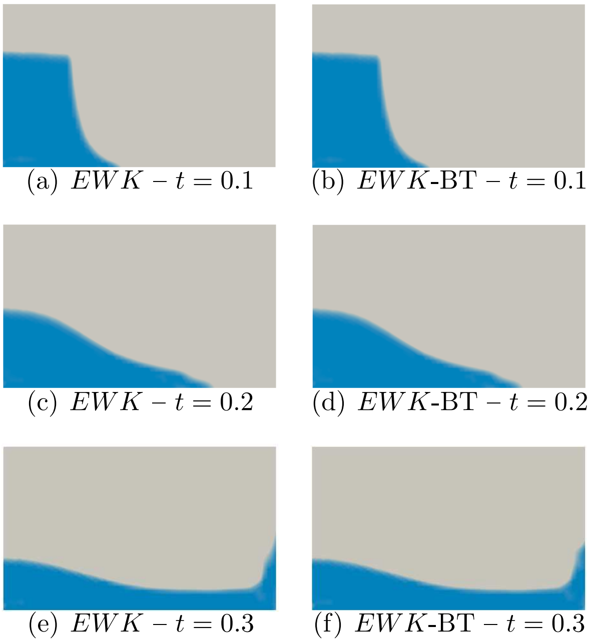
(c) Fixed – $t = 0.3$

Figure 4.4: Fixed tolerance 10^{-3} – Air-liquid configurations – The free surface flow on a dam break.

Table 4.1 shows the relative CPU times, where column INK shows the CPU time for the Inexact Newton-Krylov algorithm and column INKB shows the CPU time for the Inexact Newton-Krylov Backtracking algorithm, both considering different forcing terms and a fixed tolerance of 10^{-3}. The relative CPU time are smaller for all forcing terms in the INKB algorithm compared with the in INK algorithm. In general the forcing terms can reduce the CPU time more than 66% for INK algorithm and around 60% for INKB algorithm.

Table 4.1: Relative CPU time – The free surface flow on a dam break.

Forcing Term	INK	INKB
EWK	0.67	0.59
GLT	0.88	0.76
PP	0.66	0.61
Fixed 10^{-3}	1.00	



(a) $EWK - t = 0.1$

(b) EWK-BT $- t = 0.1$

(c) $EWK - t = 0.2$

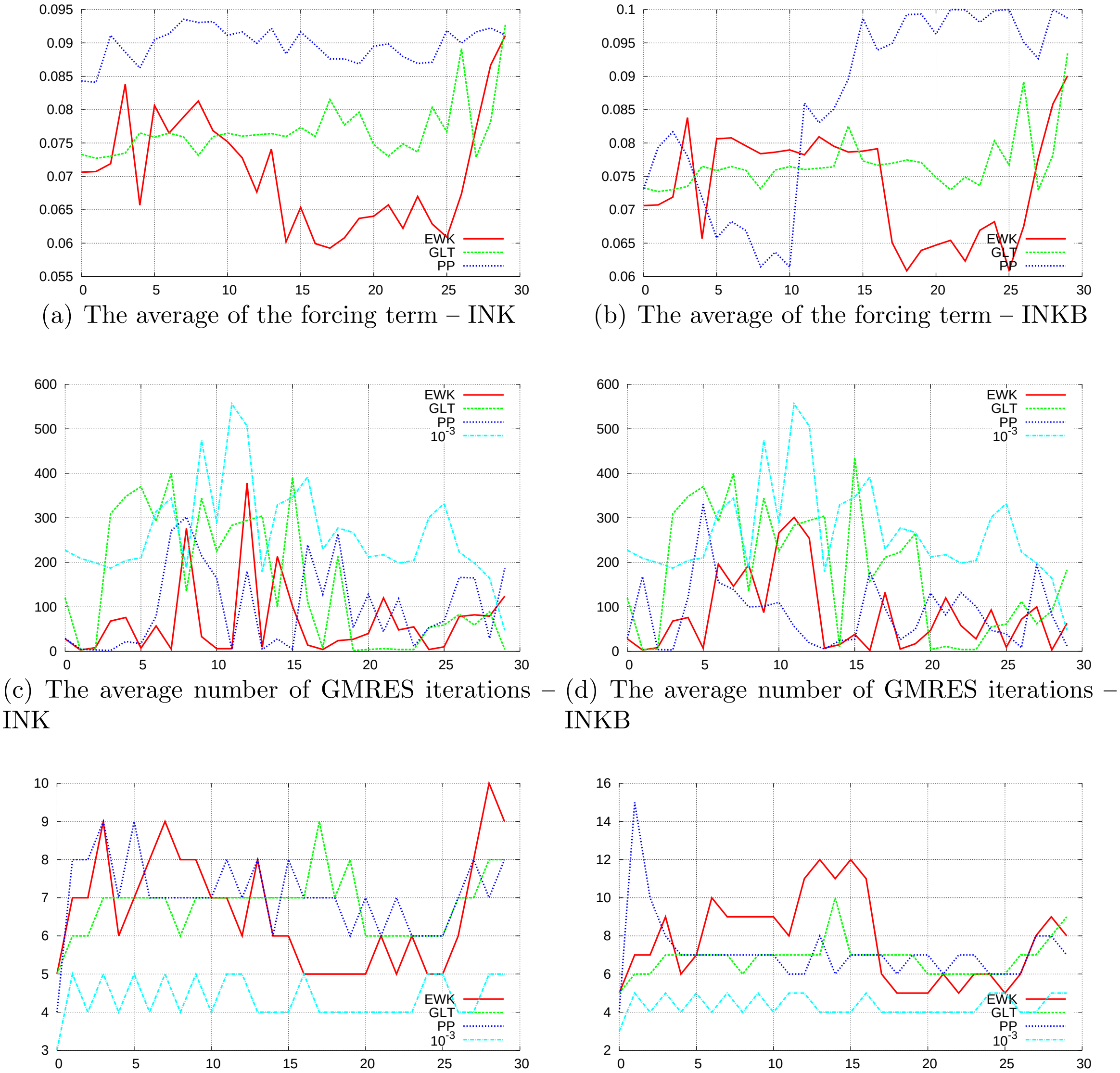
(d) EWK-BT $- t = 0.2$

(e) $EWK - t = 0.3$

(f) EWK-BT $- t = 0.3$

Figure 4.5: EWK – Air-liquid configurations – The free surface flow on a dam break.

Figure 4.6 presents the forcing terms average time history for 30 time steps – $\Delta t = 0.01$ and the final time equal 0.3 seconds. As we can see in Fig. 4.6(a), the PP average forcing term is larger than the others for the INK algorithm, whereas it is closer to the limit for the majority of time steps for INKB algorithm. The average number of GMRES iterations are larger for the fixed tolerance solution for both algorithms, INK and INKB (Figs. 4.6(c) and 4.6(d)). Figures Figs. 4.6(e) and 4.6(f) show that the number of Newton iterations is smaller for the fixed tolerance solution for both algorithms.



(a) The average of the forcing term – INK

(b) The average of the forcing term – INKB

(c) The average number of GMRES iterations – INK

(d) The average number of GMRES iterations – INKB

(e) The number of Newton iterations – INK

(f) The number of Newton iterations – INKB

Figure 4.6: The forcing term behaviour for 30 time steps simulations – The free surface flow on a dam break.

Chapter 5

Concluding Remarks

In this work we evaluated the performance of the Inexact Newton-Krylov method to solve the nonlinear equations arising from the SUPG/PSPG finite element formulation of transient incompressible fluid flows considering several forcing terms proposals and a backtracking strategy based on Armijo rule to improve convergence. We consider the free surface flow on a dam break as a 3D benchmark problem for Navier-Stokes equations. Our experiments have shown that a suitable choice of the forcing term can lead to an accurate and efficient numerical solution using considerably less CPU time. Furthermore, when backtracking is switched on for all forcing terms the CPU times are reduced.

Acknowledgment

This work was partly supported by CNPq/MCT, FAPES and FAPERJ. We are indebted to Dr. Renato Elias from the Federal University of Rio de Janeiro, by his invaluable help in the implementation and understanding of EdgeCFD software. We also would like to thank NACAD/UFRJ team, for their support and many fruitful discussions throughout the course of this work.

Bibliography

[1] Heng-Bin An, Ze-Yao Mo, and Xing-Ping Liu. A choice of forcing terms in inexact Newton method. *Journal of Computational and Applied Mathematics*, 200:47–60, 2007.

[2] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.1, Argonne National Laboratory, 2010.

[3] S. Bellavia and S. Berrone. Globalization strategies for Newton-Krylov methods for stabilized FEM discretization of Navier-Stokes equations. *Journal of Computational Physics*, 226:2317–2340, 2007.

[4] A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[5] L. Catabriga, A. M. P. Valli, B. Z. Melotti, L. M. Pessoa, and A.L.G.A. Coutinho. Performance of LCD iterative method in the finite element and finite difference solution of convection-diffusion equations. *Communications in Numerical Methods in Engineering*, 22:643–656, 2006.

[6] A.L.G.A. Coutinho, M.A.D. Martins, J. L. D. Alves, L. Landau, and A. Moraes. Edge-based finite element techniques for nonlinear solid mechanics problems. *International Journal for Numerical Methods in Engineering*, 50:2053–2068, 2001.

[7] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.

[8] J.E. Dennis Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, 1983.

[9] S.C. Eisenstat and H.F. Walker. Globally convergent Inexact Newton methods. *SIAM J.Optim.*, 4:393–422, 1994.

[10] S.C. Eisenstat and H.F. Walker. Choosing the forcing terms in an Inexact Newton method. *SIAM J. Sci. Comput*, 17:16–32, 1996.

[11] R.N. Elias and A.L.G.A. Coutinho. Stabilized edge-based finite element simulation of free-surface flows. *Int. J. Numer. Meth. Fluids*, 54:965–993, 2007.

[12] R.N. Elias, A.L.G.A. Coutinho, and M.A.D.A. Martins. Inexact-Newton-type methods for non-linear problems arising from the SUPG/PSPG solution of steady incompressible Navier-Stokes equations. *J. Braz. Soc. Mech. Sci. & Eng.*, 26:330–339, 2004.

[13] R.N. Elias, A.L.G.A. Coutinho, and M.A.D.A. Martins. Inexact Newton-type methods for the solution of steady incompressible viscoplastic flows with the SUPG/PSPG finite element formulation. *Comput. Methods Appl. Mech. Engrg.*, 195:3145–3167, 2006.

[14] R.N. Elias, M.A.D.A. Martins, and A.L.G.A. Coutinho. Parallel edge-based solution of viscoplastic flows with the supg/pspg formulation. *Computational Mechanics*, 38:365–381, 2006.

[15] L.P. Franca and S.L. Frey. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 99:209–233, 1992.

[16] A.C.R.N. Galeão and E.G. Do Carmo. A consistent approximate Speed-up Upwind Petrov-Galerkin method for convection-dominated. *Comp. Meth. Appl. Mech. Eng.*, 10:83–95, 1988.

[17] M. Gomes-Ruggiero, V. Lopes, and Julia Toledo-Benavides. A globally convergent inexact Newton method with a new choice for the forcing term. *Annals of Operations Research*, 157:193–205, 2008. 10.1007/s10479-007-0196-y.

[18] Deborah M. Greaves. Simulation of viscous water column collapse using adapting hierarchical grids. *Int. J. Numer. Meth. Fluids*, 50:693–711, 2005.

[19] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[20] T.J.R Hughes and T.E. Tezduyar. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. *Computer Methods in Applied Mechanics and Engineering*, 45:217–284, 1984.

[21] C.T. Kelley. *Iterative Methods for Linear and Nonlinear Equations.* 1995.

[22] S. Koshizuka, H. Tamako, and Y. Oka. A particle method for incompressible viscous flow with fluid fragmentation. *J. Computational Fluid Dynamics*, 4:29–46, 1995.

[23] J.C. Martin and W.J. Moyce. Part IV. An experimental study of the collapse of liquid columns on a rigid horizantal plane. *Philosophical Transactions of the Royal Society A*, 244:312–324, 1952.

[24] M. Papadrakakis and V. Balopoulos. Improved quasi-Newton methods for large non-linear problems. *J. Eng. Mech., ASCE*, 117:1201–1219, 1991.

[25] R.P. Pawlowski, J.N. Shadid, J.P. Simonis, and H.F. Walker. Globalization techniques for Newton-Krylov methods and applications to the fully coupled solution of the Navier-Stokes equations. *SIAM Review*, 48:700–721, 2006.

[26] John N. Shadid, Ray S. Tuminaro, and Homer F. Walker. An Inexact Newton method for fully coupled solution of the Navier-Stokes equations with heat and mass transport. *Journal of Computational Physics*, 137:155–185, 1997.

[27] T.E. Tezduyar. Finite elements in fluids: Lecture notes of the short course on finite elements in fluids. *Computational Mechanics Division*, , vol. 99–77, Japan Society of Mechanical Engineers, Tokyo, Japan, 1999.

[28] T.E. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8:83–130, 2001.

[29] T.E. Tezduyar. Interface-tracking and interface-capturing techniques for finite element computation of moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 195:2983–3000, 2006.

[30] T.E. Tezduyar, S. Mittal, and R. Shih. Time-accurate incompressible flow computations with quadrilateral velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 87:363–384, 1991.

[31] T.E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190:411–430, 2000.

[32] R.S. Tuminaro, F. Walker, and J.N. Shadid. On backtracking failure in Newton-GMRES methods with a demonstration for the Navier-Stokes equations. *Journal of Computational Physics*, 180:549–558, 2002.