Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXIV, págs. 1973-1987 (artículo completo)
Sebastián Giusti, Martín Pucheta y Mario Storti (Eds.)
Córdoba, 8-11 Noviembre 2016

AN IMPROVED FIXED-POINT ALGORITHM TO SOLVE THE LUBRICATION PROBLEM WITH CAVITATION

Hugo M. Checo[a], **Mohammed Jai**[a] **and Gustavo C. Buscaglia**[b]

[a] *Inst. de Ciências Matemáticas e de Computação, Univ. São Paulo, 13560-970 São Carlos, Brazil*

[b] *ICJ, INSA de Lyon (Pôle de Mathématiques), 69621 Villeurbanne, France*

Keywords: Multigrid, fixed-point algorithm, free boundary, Elrod-Adams model.

Abstract. The most widely adopted model to treat thin film (lubrication) problems with cavitation is the Elrod-Adams model. An efficient numerical solution is by no means trivial, as the classical Newton-Raphson method is not suitable for this problem. The best approach so far is to solve it with a Jacobi or Gauss-Seidel type algorithm, as done by Ausas (R. Ausas et al, *Journal of Tribology*, 131(3):031702 (2009)). However, it becomes computationally expensive for large problems, due to the elliptic problem being solved for the pressure. Multigrid methods arise like a natural choice to accelerate results, yet their implementation for the Elrod-Adams model is not straightforward. A mass-conserving multigrid algorithm was presented by Checo (Doctoral dissertation, Universidade de São Paulo (2016)), where it was shown that care must be taken when transferring the solution between multigrid meshes due to the cavitation boundary shift between them. Here we propose a fixed-point method to solve stationary flow in porous media, which outperforms our previous algorithm.

1 INTRODUCTION

Fluid-dynamic bearings are present in many key high-load, high-speed precision applications where other mechanisms like ball bearings would not perform as well due to noise and vibrations, or because is it even impossible to apply another solution. Most of these components, such as journal bearings, thrust bearings, connecting rods, ring/liner contacts in combustion engines among others, work under severe loading and velocity conditions. A complete separation of both bearing surfaces (*hydrodynamic* lubrication regime) is not assured, and partial solid-solid lubrication (*mixed* lubrication regime) or mainly solid-solid contact (*boundary* lubrication regime) takes place. Also due to the convergent-divergent geometry of the surfaces, the squeeze effect and the high velocities the fluid cavitates. Under these conditions the real topography of the surfaces is a determining factor in the bearing performance.

Surface roughness is usually treated in lubrication problems by means of stochastic models, averaged models or homogenization methods. The stochastic models (Tzeng and E. (1967); Christensen (1969)), compute expected values of pressure, friction force and load carrying capacity. The approach of averaged models (Patir and Cheng (1978, 1979)) is quite different, as in this case an average Reynolds equation is developed in terms of empirical flow factor functions based in a probability distribution for the surface roughness. Homogenization methods (Bayada et al. (1988, 1989); Jai (1995); Buscaglia et al. (2002); Buscaglia and Jai (2004)) assume periodicity of the surface and develop an average equation with coefficients computed from solutions of local problems. In this work, the uncertainties associated with such methods are avoided, thus leading to larger problems and the ensuing need of better numerical methods to alleviate the computational burden.

The Elrod-Adams model (Elrod and Adams, 1974) is the most acknowledged model to treat cavitation in hydrodynamic lubrication problems. As the solution obtained is discontinuous in one of its variables, Jacobian-based methods are unfitting. The most appropiate approach is to solve it with a relaxation scheme, as done by Ausas et al. (2009), however, it is expensive from a computational point of view. Fine meshes are required to capture the effect of the smaller scale features such as surface roughness and micro-textures and therefore the computational cost is high. As in the Elrod-Adams model an elliptic equation is solved in the pressurized parts of the domain, this makes the problem suitable for a multigrid implementation. An efficient and robust multigrid algorithm for dynamic lubrication problems with cavitation is here presented, and the difficulties of a *Full Approximation Scheme* solution (Fulton et al., 1986) are discussed.

This article is organized as follows: in Section 2 a general model for lubricated devices where cavitation phenomena takes place is presented. Details of the discretization are given in Section 3. In the same section fixed point single-grid and multigrid algorithms to solve the Elrod-Adams equations coupled with the bearing dynamics are presented. The methods based in Alt's (Alt, 1980) and Ausas (Ausas et al., 2009) algorithms are also introduced. A discussion of problems and performance issues arising from both methods is given in Section 4. Finally, conclusions are drawn in Section 5.

2 A MODEL FOR LUBRICATED CONTACTS

Let us pose the lubrication problem with cavitation in the most general form. The problem's domain is the rectangle $\Omega = (x_{1\ell}, x_{1r}) \times (0, w)$. We can set two surfaces moving with relative velocity $u(t)$ one against the other, with $u(t)$ in the x_1 direction. The reference frame is fixed on the upper surface. This is depicted in Figure 1. The upper surface, which bears a load $W(t)$, is given by the function $h_U(x_1, x_2)$ satisfying $\min_{(x_1,x_2)} h_U(x_1, x_2) = 0$, while the lower one is given by $h_L(x_1 - u\,t, x_2) \leq 0$, with the gap h between the surfaces being (partially) filled with a lubricating fluid.

Cavitation is common phenomena in lubricated devices due to the convergent-divergent geometries and the squeeze effect of the relative movement of the components. The Elrod-Adams model incorporates into a single formulation the Reynolds equation for the pressurized region and the Jacobsson-Floberg-Olsson boundary conditions for cavitation. In this model two fields are computed: $p = p(x_1, x_2, t)$ and $\theta = \theta(x_1, x_2, t)$, the hydrodynamic pressure and an auxiliary saturation-like variable, respectively, that (weakly) satisfy the equation

$$\nabla \cdot \left(\frac{h^3}{2} \nabla p \right) = \frac{u(t)}{2} \frac{\partial\, h\theta}{\partial x_1} + \frac{\partial\, h\theta}{\partial t} \tag{1}$$

under the complementarity conditions

$$\begin{cases} p \; > \; 0 & \Rightarrow \qquad \theta \; = \; 1 \\ \theta \; < \; 1 & \Rightarrow \qquad p \; = \; 0 \\ 0 \; \leq \; \theta \; \leq \; 1 & \end{cases} \tag{2}$$

which is here given in its non-dimensional form, assuming constant viscosity. Initial conditions for θ and p are provided. A flux of lubricant η is set upstream at the boundary of the computational domain. At $x_2 = 0$ and $x_2 = w$ proper boundary conditions are set to enforce periodicity in the x_2 direction.

The gap h changes not only due to the relative movement of the surfaces, but also due to the imbalance on the forces on the upper surface. For simplicity and without loss of generality we set one degree of freedom in the x_3 direction, and thus

$$h(x_1, x_2, t) = h_U(x_1, x_2) - h_L(x_1 - u\,t, x_2) + Z(t)\;, \tag{3}$$

where $Z(t) > 0$ parametrizes the position of the upper surface, that is, its distance to the $x_1 - x_2$ plane. Then, its dynamics are given by

$$m\frac{d^2 Z}{dt^2} = W(t) + W^h(t)\;, \tag{4}$$

where m is the mass associated to the upper surface, which in this case is allowed to move only in the x_3 direction. All quantities are assumed to be non-dimensional. Equation (4) is supplemented with initial conditions for $Z(t = 0) = z_0$ and $Z'(t = 0) = v_0$. The *hydrodynamic lift* $W^h(t)$ is given by

$$W^h(t) = \int_0^w \int_a^b p(x_1, x_2, t)\, dx_1\, dx_2\;, \tag{5}$$

where a and b are the starting and ending positions of the upper surface, e.g., $a = x_{1\ell}$ and $b = x_{1r}$ in Figure 1. Notice that equations (1) and (4) are coupled through the function $h(x_1, x_2, t)$, as the hydrodynamic pressure p depends on the solution of equation (1), which in time depends on $h(x_1, x_2, t)$.

The mathematical problem to be solved is to

"Find function $Z(t) > 0$ and fields $p(x_1, x_2, t)$ and $\theta(x_1, x_2, t)$ in $\Omega \times [0, T]$ solution of

$$\begin{cases} m\dfrac{d^2 Z}{dt^2} = W(t) + W^h(t) \qquad t \in [0, T] & (6) \\ Z(0) \; = \; z_0, & (7) \\ Z'(0) \; = \; v_0, & (8) \end{cases}$$

where $m > 0$, $T > 0$ are two given real numbers and $W(t)$ is a time-implicitly known function. The fields $p(x_1, x_2, t)$, $\theta(x_1, x_2, t)$, defined on $\Omega = (x_{1\ell}, x_{1r}) \times (0, w)$ and periodic/reflective in x_2 satisfy

$$\begin{cases} \nabla \cdot \left(\dfrac{h^3}{2} \nabla p \right) = \dfrac{u}{2} \dfrac{\partial h\theta}{\partial x_1} + \dfrac{\partial h\theta}{\partial t} \qquad \text{in } \Omega \times]0, T[& (9) \\ p > 0 \quad \Rightarrow \quad \theta = 1 & (10) \\ \theta < 1 \quad \Rightarrow \quad p = 0 & (11) \\ 0 \leq \theta \leq 1 & (12) \end{cases}$$

with boundary and initial conditions:

$$\begin{cases} uh\theta - h^3 \dfrac{\partial p}{\partial x_1} = \eta(x_2, t) \qquad \forall x_2 \in \mathit{\Gamma}_0 = \{x_{1\ell}\} \times [0, w] & (13) \\ p = 0 \qquad \forall x_2 \in \mathit{\Gamma}_1 = \{x_{1r}\} \times [0, w] & (14) \\ \theta(x_1, x_2, t = 0) = \theta_I(x_1, x_2), \qquad (x_1, x_2) \in \Omega & (15) \end{cases}$$

”.

Since Elrod & Adams published the model the mathematical study of the problem has been considered by several authors with increasing grade of complexity. Existence and uniqueness (under some assumptions) of the steady state problem have been treated in Bayada and Chambat (1984, 1986). The unsteady case was studied in El Alaoui (1986); Alvarez (1986) for journal and face seals boundary conditions. Proof of the existence of a solution of the transient problem for a journal bearing (that is, periodic boundary conditions in x_1) are given in El Alaoui Talibi and Jai (2012). Existence of a solution for the problem stated above was given by (Buscaglia et al., 2015).

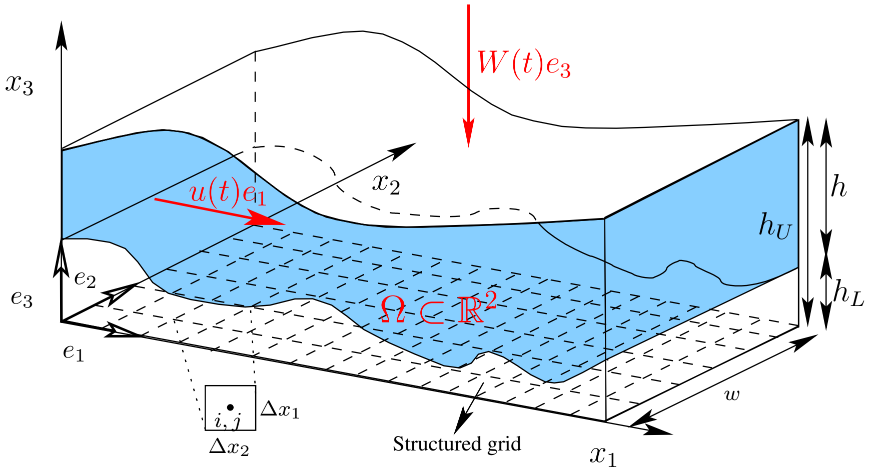


Figure 1: Upper and lower surfaces moving with relative velocity $u(t)$ along the x_1 direction.

3 NUMERICAL METHOD

3.1 Discretization

The domain Ω is divided in finite volumes in a structured grid, as shown in Figure 1. Pressure and saturation values are considered to be constant within each finite volume. A balance of fluxes in each cell i,j gives the following discrete equation for $p_{i,j}$ and $\theta_{i,j}$:

$$a_{i,j}\theta_{i,j} + b_{i,j}p_{i,j} - c_{i,j}(p,\theta) = l_{i,j} \ , \tag{16}$$

where

$$a_{i,j} = \begin{cases} (q_2 + u\Delta x_1)\, h_{ij}^{n+1} & \text{if } u \geq 0 \ , \\ (q_2 - u\Delta x_1)\, h_{ij}^{n+1} & \text{if } u < 0 \ , \end{cases} \tag{17}$$

$$b_{i,j} = s_{i-1,j}^{n+1} + s_{i+1,j}^{n+1} + q_1^2(s_{i,j-1}^{n+1} + s_{i,j+1}^{n+1}) \ , \tag{18}$$

$$c_{i,j}(p,\theta) = \begin{cases} s_{i-1,j}p_{i-1,j}^{n+1} + s_{i+1,j}p_{i+1,j}^{n+1} + q_1^2(s_{i,j-1}p_{i,j-1}^{n+1} & \\ +s_{i,j+1}p_{i,j+1}^{n+1}) + u\Delta x_1 h_{i-1,j}^{n+1}\theta_{i-1,j}^{n+1} & \text{if } u \geq 0 \ , \\ s_{i-1,j}p_{i-1,j}^{n+1} + s_{i+1,j}p_{i+1,j}^{n+1} + q_1^2(s_{i,j-1}p_{i,j-1}^{n+1} & \\ +s_{i,j+1}p_{i,j+1}^{n+1}) - u\Delta x_1 h_{i+1,j}^{n+1}\theta_{i+1,j}^{n+1} & \text{if } u < 0 \ , \end{cases} \tag{19}$$

$$s_{i+k,j} = \frac{h_{i+k,j}^3 + h_{i,j}^3}{2} \quad , s_{i,j+k} = \frac{h_{i,j}^3 + h_{i,j+k}^3}{2} \quad , \tag{20}$$

with $k = -1, 1$ and

$$l_{i,j} = q_2 h_{i,j}^n \theta_{i,j}^n \ . \tag{21}$$

where $q_1 = \Delta x_1/\Delta x_2$, $q_2 = 2\Delta x_1^2/\Delta t$.
Equation (4) is solved through a Newmark scheme

$$Z^{n+1} = Z^n + \Delta t \frac{dZ^n}{dt} + \frac{\Delta t^2}{2}\frac{d^2Z^{n+1}}{dt^2} \ , \tag{22}$$

$$\frac{dZ^{n+1}}{dt} = \frac{dZ^n}{dt} + \Delta t \frac{d^2Z^{n+1}}{dt^2} \ , \tag{23}$$

which leads to

$$Z^{n+1} = Z^n + \Delta t \frac{dZ^n}{dt} + \frac{\Delta t^2}{2m}\left(W(t^{n+1}) + W^h(t^{n+1})\right) \ , \tag{24}$$

$$\frac{dZ^{n+1}}{dt} = \frac{dZ^n}{dt} + \frac{\Delta t}{m}\left(W(t^{n+1}) + W^h(t^{n+1})\right) \ . \tag{25}$$

3.2 Fixed point algorithms for the lubrication problem

It is useful to introduce at this point the variable ω:

$$\omega_{i,j} = p_{i,j} + \theta_{i,j} \geq 0 \ . \tag{26}$$

It is easy to verify that $p_{i,j}$ and $\theta_{i,j}$ can be recovered uniquely from $\omega_{i,j} \geq 0$ with:

$$p_{i,j}(\omega_{i,j}) = \max\{\omega_{i,j} - 1, 0\} \ , \qquad \theta_{i,j}(\omega_{i,j}) = \min\{\omega_{i,j}, 1\} \ . \tag{27}$$

Given a certain ordering of the cell centers (i,j), for instance $i' = i + (j-1)N_1$, $i = 1, ..., N_1$, $j = 1, ..., N_2$, we can then write in vector form:

$$\boldsymbol{\omega} = \mathbf{p} + \boldsymbol{\theta} \ , \quad \boldsymbol{\omega} \in \mathbb{R}^{N = N_1 \times N_2} \ , \tag{28}$$

where $\boldsymbol{\omega}_{i'} = \omega_{i,j}$. At this point we can drop the prime superindexes on i'.

All the points i in the domain belong to a set that we will call I. The points in the Dirichlet part of the boundary belong to $I_1 \neq \{\varnothing\}$ and those in the reflection/periodic boundaries are in $I_2 \neq \{\varnothing\}$, both subsets of I such that $I_1 \cap I_2 = \{\varnothing\}$.

Thus for all points $i \in I \backslash I_1 \backslash I_2$

$$\boldsymbol{\mathcal{N}}_i(\boldsymbol{\omega}) = \mathbf{a}_i\boldsymbol{\theta}_i + \mathbf{b}_i\mathbf{p}_i - \mathbf{c}_i(\boldsymbol{\omega}) = \mathbf{l}_i \ , \tag{29}$$

must be satisfied. The flux η at the boundary Γ_0 is usually defined as a constant oil film thickness d_{oil} smaller than the gap, thus $\theta = d_{oil}/h(x_1, x_2, t) < 1$, $\forall (x_1 = x_{1\ell},\ x_2,\ t), (x_1 = x_{1r},\ x_2,\ t)$. Hence Dirichlet boundary conditions for $\boldsymbol{\omega}_i$ are set

$$\boldsymbol{\omega}_i = \overline{\boldsymbol{\omega}}_i = p_{\ell/r} + \frac{d_{oil,\ell/r}}{h_{\{1,N_1\},j}} \ , \tag{30}$$

for all $i \in I_1$. Subindexes ℓ and r refer to the left and right extremes of the domain. For points $i \in I_2$ we define an injective operator:

$$\mathcal{I} : I_2 \longrightarrow I \backslash I_2 \ , \tag{31}$$

that associates to every i in I_2 the corresponding j in $I \backslash I_2$ such that $j = \mathcal{I}(i)$ is the corresponding cell in the reflection/periodic boundary conditions.

Let us define the operator

$$(\mathbf{A}(\boldsymbol{\omega}))_i = \begin{cases} \mathcal{R}(\boldsymbol{\omega}_i) & i \in I \backslash I_1 \backslash I_2 \\ \overline{\boldsymbol{\omega}}_i & i \in I_1 \\ \boldsymbol{\omega}_{\mathcal{I}(i)} & i \in I_2 \end{cases} \ , \tag{32}$$

The operator $\mathcal{R}$ will be defined shortly and it is what differentiates the algorithms here presented. With appropriate properties on $\mathcal{R}$, $\mathbf{A}$ satisfies

$$\mathbf{A}(\boldsymbol{\omega})_i \geq 0 \quad \forall i \in I \ , \tag{33}$$

The operator $\mathcal{R}$, which is defined in section 3.2.2, depends on $\mathbf{a} = [\mathbf{a}_i]$, $\mathbf{b} = [\mathbf{b}_i]$, $\mathbf{c} = [\mathbf{c}_i]$ and $\mathbf{l} = [\mathbf{l}_i]$. Therefore, $\mathbf{A}$ depends on h^{n+1} and then on Z^{n+1}. Going back to equation 24, then we have

$$Z^{n+1} = Z^n + \Delta t \frac{dZ^n}{dt} + \frac{\Delta t^2}{2m}\left((W)^{n+1} + W^h(Z^{n+1})\right) \ . \tag{34}$$

The solution to the previous equation is the fixed point Z^{n+1}, however, it should not be solved through a fixed point method, as the right hand side of (34) is not guaranteed to be a contraction. It can be solved by Newton's method or a secant method instead, until a defined tolerance

$$|(Z^{n+1})^{q+1} - (Z^{n+1})^q| \leq \epsilon_Z \ , \tag{35}$$

is reached between two consecutive points $(Z^{n+1})^{q+1}$ and $(Z^{n+1})^q$ in the iteration.

It is to be noticed that at each step q, $(Z^{n+1})^q$ must be updated and hence the operator $\mathbf{A}$, in order to solve $\boldsymbol{\omega}$ to afterwards compute $W^h((Z^{n+1})^q)$. This leads us to the following algorithm:

Algorithm 1 Algorithm to solve problem (6)-(15)

```
Provide initial values, $Z^0 = z_0$, $(Z')^0 = v_0$, $\boldsymbol{\omega}_i^0 = (\theta_I)_i \; \forall i \in I$
$t = 0$, $n = 1$
while $t \leq T$ do
    $q = 0$, $Z^q = Z^{n-1}$, $k = 0$
    while $|Z^{q+1} - Z^q| > \epsilon_Z$ and $\|\mathbf{r}\| > \epsilon_r$ do
        Perform $K$ iterations of $\boldsymbol{\omega}^{k+K} = \mathbf{A}^K(\boldsymbol{\omega}^k)$ (Gauss-Seidel relaxations)
        Compute $W^h(t)$
        Obtain $Z^{q+1}$ by performing one iteration of the Newton or secant method on eq. (34)
        Update $h(x_1, x_2, t)^{q+1}$
        $q \; = \; q + 1$
    end while
    Compute $(Z')^n$ according to eq. (25)
    $Z^n = Z^q$
    $\boldsymbol{\omega}^n = \boldsymbol{\omega}^k$
    $n = n + 1$
end while
```

That is, for each time step n within each secant iteration q, K applications of the operator (32) are performed.

3.2.1 A multilevel algorithm

The operators of the previous sections were defined on a certain grid of size $(\Delta x_1, \Delta x_2)$. We will now establish a set of coarser meshes with sizes $(2^l \Delta x_1, 2^l \Delta x_2)$, with $l = 0, 1, 2, ..., M-1$. With a structured finest mesh, the definition of the coarser grids $l = 1, ..., M-1$ is straightforward. In each of these grids $\mathbf{a}^l$, $\mathbf{b}^l$ and $\mathbf{c}^l$ are defined as in equations (17), (18) and (19), however, $\mathbf{l}^l$ is defined as

$$\mathbf{l}^l = \begin{cases} q_2 \mathbf{h}_i^n \theta(\boldsymbol{\omega}_i^n) \quad \forall i \in I & \text{if } l = 0 \, , \\ (\mathcal{T}_{\mathbf{r}})_{l-1}^l \left(\mathbf{r}^{l-1}(\boldsymbol{\omega}^{l-1})\right) + \boldsymbol{\mathcal{N}}^l((\mathcal{T}_\omega)_{l-1}^l \boldsymbol{\omega}^{l-1}) & \text{otherwise} \end{cases} \, , \tag{36}$$

where $\mathbf{r}$ is the residue

$$\mathbf{r}^l(\boldsymbol{\omega}^l) = \mathbf{l}^l - \boldsymbol{\mathcal{N}}^l(\boldsymbol{\omega}^l) \quad \forall l \, , \tag{37}$$

$\mathcal{T}_{l-1}^l$ is the restriction from mesh $l-1$ to l, a linear operator which can be different for the residue ($\mathcal{T}_{\mathbf{r}}$) and the solution vector($\mathcal{T}_\omega$). For the former a *full weighting* operator is used and for the latter the *injection* operator (Fulton et al., 1986). Notice that in equation (32) the right hand side $\mathbf{l}$ is defined in the finest grid as in equation (21), but in the coarser meshes depends on a solution vector $\boldsymbol{\omega}^{l-1}$ on a immediately finer mesh $l-1$. Introducing the vectors

$$\boldsymbol{\nu}^d \in \mathbb{N}^{M-1} \, , \; \boldsymbol{\nu}^u \in \mathbb{N}^M \, , \tag{38}$$

and the restriction $\mathcal{T}_l^{l-1}$ from mesh $l-1$ to l, then the multigrid operator, here defined in a recursive manner, reads:

$$
\mathbf{M}(\boldsymbol{\omega}_d^{l=0}) = \begin{cases}
\boldsymbol{\omega}_u^{l=0}\ , \text{ where:} & \qquad\qquad\qquad\qquad\qquad\qquad (39a)\\
\boldsymbol{\omega}_u^{l} = \left(\mathbf{A}^{l}\right)^{\boldsymbol{\nu}_l^u} \left(\left(\boldsymbol{\omega}_u^{l}\right)^{(1)}\ , \mathbf{l}^l(\boldsymbol{\omega}_d^{l-1})\right)\ , & \text{with} \qquad (39b)\\
\left(\boldsymbol{\omega}_u^{l}\right)^{(1)} = \mathcal{T}_{l+1}^{l}\boldsymbol{\omega}_u^{l+1} + \boldsymbol{\omega}_d^{l} - \mathcal{T}_{l+1}^{l}\mathcal{T}_{l}^{l+1}\boldsymbol{\omega}_d^{l} & \text{for } l = 0, ..., M-2\ ; \qquad (39c)\\
\boldsymbol{\omega}_u^{l} = \boldsymbol{\omega}_d^{l} & \text{for } l = M-1\ ; \qquad (39d)\\
\boldsymbol{\omega}_d^{l} = \left(\mathbf{A}^{l}\right)^{\boldsymbol{\nu}_l^d} \left(\mathcal{T}_{l-1}^{l}\boldsymbol{\omega}_d^{l-1}\ , \mathbf{l}^l(\boldsymbol{\omega}_d^{l-1})\right) & \text{for } l = 0, ..., M-1\ ; \qquad (39e)
\end{cases}
$$

Here $\left(\mathbf{A}^{l}\right)^{\boldsymbol{\nu}_l^{u,d}}$ implies the application of the operator successively $\boldsymbol{\nu}_l^{u,d}$ times. The operation $\mathbf{M}(\boldsymbol{\omega})$ is called a V-cycle. This multigrid method is the *Full Approximation Scheme* (Fulton et al., 1986).

```
Algorithm 2 Multilevel algorithm to solve problem (6)-(15)
1: Replace A in line 6 of Algorithm 1 by M
```

REMARK 1 (An important remark on operator $\mathbf{M}$): Equation (39c) is the accretion of the high frequency components (last two terms in the right hand side) to the coarse mesh solution (first term). This does not guarantee $\left(\boldsymbol{\omega}_u^{l}\right)^{(1)} \geq 0$ for standard prolongation operators, that is, a vector $\left(\boldsymbol{\omega}_u^{l}\right)^{(1)}$ complying with the complementarity conditions.

3.2.2 Defining the operator $\mathcal{R}$

What remains to be defined is $\mathcal{R}$. The operator $\mathcal{R} = \mathcal{R}_A$ for the algorithm introduced in Ausas et al. (2009) can be written by first defining $\mathcal{R}_1$ and $\mathcal{R}_2$ as

$$
\mathcal{R}_1(\boldsymbol{\omega}_i) = \begin{cases}
\frac{\mathbf{l}_i+\mathbf{c}_i-\mathbf{a}_i}{\mathbf{b}_i} + 1 & \text{if } \boldsymbol{\omega}_i \geq 1 \text{ and } \mathbf{l}_i + \mathbf{c}_i - \mathbf{a}_i \geq 0\ ,\\
\theta_i(\boldsymbol{\omega}_i) & \text{if } \boldsymbol{\omega}_i \geq 1 \text{ and } \mathbf{l}_i + \mathbf{c}_i - \mathbf{a}_i < 0\ ,\\
\boldsymbol{\omega}_i & \text{if } \boldsymbol{\omega}_i < 1\ ,
\end{cases} \tag{40}
$$

and

$$
\mathcal{R}_2(\boldsymbol{\omega}_i) = \begin{cases}
\frac{\mathbf{l}_i+\mathbf{c}_i}{\mathbf{a}_i} & \text{if } \boldsymbol{\omega}_i < 1 \text{ and } \frac{\mathbf{l}_i+\mathbf{c}_i}{\mathbf{a}_i} < 1\ ,\\
1 & \text{if } \boldsymbol{\omega}_i < 1 \text{ and } \frac{\mathbf{l}_i+\mathbf{c}_i}{\mathbf{a}_i} \geq 1\ .
\end{cases} \tag{41}
$$

Then we define

$$
\mathcal{R}_A = \mathcal{R}_2 \circ \mathcal{R}_1\ . \tag{42}
$$

REMARK 2 (Remark on operator $\mathcal{R}_A$): The operator $\mathcal{R}_A$ takes a $\boldsymbol{\omega} \geq 0$ and if the finite volume is pressurized ($\boldsymbol{\omega} \geq 1.0$) it computes either pressure or saturation, as seen in equation (40). Then this temporary $\mathcal{R}_1(\boldsymbol{\omega})$ is corrected if $\mathcal{R}_1(\boldsymbol{\omega}) \leq 1.0$. The final outcome satisfies the complementarity conditions, however, for a single application of $\mathcal{R}_A$ it does not guarantee mass balance at each cell. It is to be noticed in equations (40), (41) and (42) that the outcome of applying $\mathcal{R}_A$ on a certain $\mathbf{w}_i < 1$, $i \in I\backslash I_1\backslash I_2$ is $\mathcal{R}_A(\mathbf{w}_i) = 1.0$ if $\frac{\mathbf{l}_i+\mathbf{c}_i}{\mathbf{a}_i} \geq 1$:

$$
\mathcal{R}_A(\mathbf{w}_i) = \mathcal{R}_2 \circ \mathcal{R}_1(\mathbf{w}_i) = \mathcal{R}_2(\mathbf{w}_i) = 1.0\ , \tag{43}
$$

while the answer given by operator $\mathcal{R}_{Alt}$ is

$$
\mathcal{R}_{Alt}(\mathbf{w}_i) = \frac{\mathbf{l}_i^l + \mathbf{c}_i^l - \mathbf{a}_i^l}{\mathbf{b}_i^l} + 1\ . \tag{44}
$$

This is most likely to be corrected in the next application of $\mathcal{R}_A$. In a single-grid algorithm this does not make a difference in the final solution, however, in a multi-grid algorithm it does, as the residue is higher in the cell i where the solution has not been correctly computed. Let us remember that the residue is used to define the problem in the coarser meshes, as set in equation (39e).

Alt (Alt, 1980) presented a general fixed point algorithm to solve stationary flow in porous media. This algorithm applies well to the lubrication problem with cavitation. The operator $\mathcal{R}$ as defined in Alt (1980) is

$$\mathcal{R}^l_{Alt}(\boldsymbol{\omega}_i) = \begin{cases} \frac{\mathbf{l}^l_i + \mathbf{c}^l_i - \mathbf{a}^l_i}{\mathbf{b}^l_i} + 1 & \text{if } \mathbf{l}_i + \mathbf{c}_i - \mathbf{a}_i \geq 0 \, , \\ \frac{\mathbf{l}^l_i + \mathbf{c}^l_i}{\mathbf{a}^l_i} & \text{otherwise} \, , \end{cases} \qquad \forall l \tag{45}$$

Notice that here mass-balance is satisfied at each cell i.

4 NUMERICAL TESTS

The computational domain is $\Omega = (0, 2) \times (0, 0.2)$. A sinusoidal traveling texture

$$h_L = A\left(1 - \sin\left(2\pi \frac{x_1 - u\,t}{\lambda}\right)\right) \, , \tag{46}$$

is set as the lower surface, with $A = 2$, $\lambda = 0.4$ and $u = -1$. The upper surface describing the shape of the pad is given by

$$h_U = \begin{cases} 32000 - \sqrt{32000^2 - (1000(x - 1.0))^2} & \text{if } 0.5 \leq x_1 \leq 1.5 \\ 500 & \text{otherwise} \, . \end{cases} \tag{47}$$

This geometry can be seen in the plane $x_2 = 0.1$ for $Z = 2$ in Figure 2. A constant load $W(t) = -8.3 \times 10^{-3}$ is set on the pad, which has mass m equal to $m = 1.67 \times 10^{-7}$. Although the domain in which the problem is solved is two-dimensional, the solution of this particular problem is one-dimensional. This is kept like this for the sake of clarity and simplicity in the presentation of results.

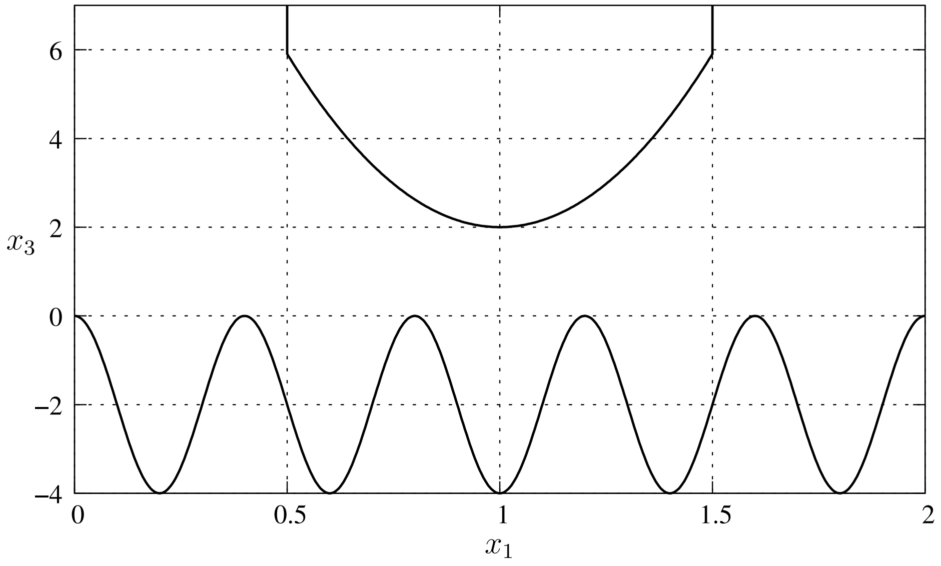


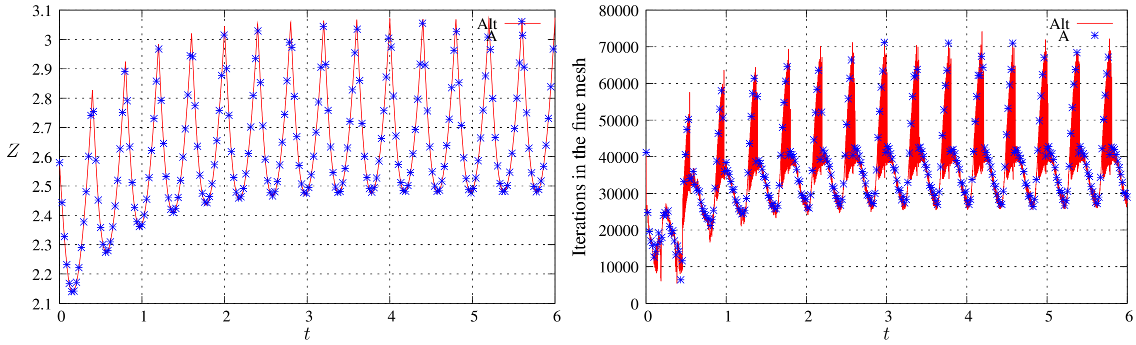
Figure 2: Section of the domain along the plane $x_2 = 0.1$.

Tolerances for the convergence of the hydrodynamic pressure p and the pad position Z are set to $\epsilon_r = 10^{-5}$ and $\epsilon_Z = 10^{-4}$ respectively. This values have been chosen so as to ensure convergence for a mesh made of 512×32 finite volumes. The time step Δt is defined to enforce a Courant number ($u\Delta t/\Delta x_1$) of 1.0 which for a final simulation time T=6.0 gave a total of 3072 time steps.

4.1 Single-level algorithm

In this section we solve the problem with algorithm 1, that is, only in the finest mesh (512×32 mesh) and with $K = 100$ in Algorithm 1, that is, Z is updated each 100 relaxations. Thus, we will distinguish between the implementation with the operator $\mathcal{R}_A$ and the implementation with the operator $\mathcal{R}_{Alt}$. In a single-level algorithm, that is, just performing Gauss-Seidel relaxations in the finest mesh, both $\mathcal{R}_A$ and $\mathcal{R}_{Alt}$ return the same solution. This can be seen in Figure 3(a): the differences in results in terms of dynamics $Z(t)$ are negligible for the chosen tolerances. In terms of pointwise quantities some small deviations can be seen. In Figure 4 both the fluid film ($h\theta$) and the hydrodynamic pressure are plotted at $t = 2.6953$. In the variable θ both results concur, although some small differences can be observed in the hydrodynamic pressure.

Even the computational burden is comparable for both operators $\mathcal{R}_A$ and $\mathcal{R}_{Alt}$. Figure 3(b) shows the number of Gauss-Seidel iterations.



(a) (b)

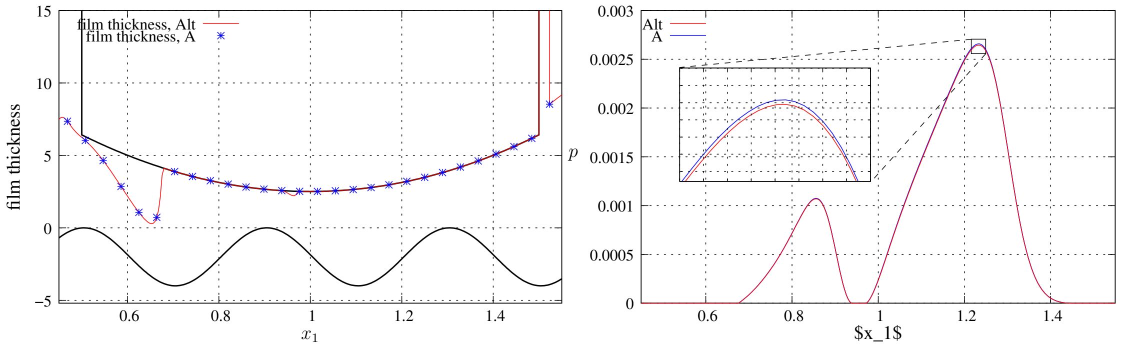
Figure 3: (a) Minimum clearance for Algorithm 1 using both operators. (b) Number of relaxation sweeps required to achieve tolerances ϵ_r and ϵ_Z. Results for $\mathcal{R}_A$ are shown at every tenth time step.

4.2 Multi-level algorithm

In the previous section it was made clear that no apparent advantage came from neither of both operators. The differences arise in multi-level algorithms, such as Algorithm 2. Here we will consider again the 512×32 finite volume mesh as the finest mesh, and 256×16, 128×8 and 64×4 as the coarser meshes, which will be identified as meshes l= 0, 1, 2 and 3 respectively.

4.2.1 With operator $\mathcal{R}_{Alt}$

The problem is solved with Algorithm 2 and operator $\mathcal{R}_{Alt}$ as presented in Section 3.2.1. For the residue $\mathbf{r}^l$ defined in each mesh l we set the same previous tolerances $\epsilon_r = 10^{-5}$ (for every multigrid level) and $\epsilon_Z = 10^{-4}$. What remains to be defined are the vectors $\boldsymbol{\nu}^u$ and $\boldsymbol{\nu}^d$. Here we choose $\boldsymbol{\nu}^d = \{1\ 1\ 1\}$ for every test. This amounts to perform just one Gauss-Seidel relaxation



(a) (b)

Figure 4: (a) Fluid film and (b) hydrodynamic pressure at $t = 2.6953$ for Algorithm 1 and both operators $\mathcal{R}_A$ and $\mathcal{R}_{Alt}$.

sweep in each mesh l=0,1,2 on the way down a V-cycle, which is the minimum required to define the problem to be solved in the immediately coarser mesh $l+1$, as seen in equation (36).

Results for nineteen $\boldsymbol{\nu}^u$ vectors are shown in Table 1. The fourth column is the total number of Gauss-Seidel relaxations performed in the fine mesh in the 3072 time steps of the simulations. This is a good measure of the computational effort only for Algorithm 1, but for Algorithm 2 the cost of the Gauss-Seidel iterations in the coarser meshes must be accounted for too. If we define a work unit (Wu) as one relaxation sweep in the finest mesh, then in the coarser meshes the cost of a Gauss-Seidel relaxation is 0.25 Wu, 0.0625 Wu and 0.015625 Wu respectively. Knowing the vectors $\boldsymbol{\nu}^d$ and $\boldsymbol{\nu}^u$ the cost in terms of work units can be computed for a v-cycle. This is shown in column 5. Then, the total computational work made in term of work units makes for a more fair comparison of the total computational burden. This is shown in column 6.

In the multigrid runs (Algorithm 2) a maximum number v-cycles is fixed, while for Algorithm 1 a number of Gauss-Seidel relaxations large enough to achieve convergence was provided. The seventh column presents the percentage of the 3072 time steps in which the solver failed to reduce the norm of the residue $\|\mathbf{r}\|$ to the specified tolerance ϵ_r.

The first line in Table 1 correspond to the usual number of relaxation sweeps $\boldsymbol{\nu}^u = \{1\;1\;1\;1\}$ taken in linear multigrid methods. The residue is not reduced to a value below the tolerance ϵ_r for 8.7% of the 3072 time steps. In those steps, the residue is not reduced to the tolerance not only because the maximum number of v-cycles is reached, but because of a problem in decreasing it. This is shown in Figure 5. In the top figure the residue is first reduced, then it slightly increases and then oscillates between two values higher than the required tolerance ϵ_r.

This can be explained in terms of Remark 1. Both the prolongation $\mathcal{T}^l_{l+1}$ operator (linear interpolation in this case) and the addition of the high frequency components tamper with the complementarity conditions, specially at the reformation boundaries. The bottom left figure (Fig. 5) shows in blue circles the position of the cells where the complementarity conditions have been tampered by the interpolation from the coarser meshes to the finest mesh. The residue $\mathbf{r}$ at each cell center is also shown. It can be noticed that at those points the residue is *orders* of magnitude higher than in the rest of the domain. After one Gauss-Seidel relaxation the residue is reduced and all cells in the domain satisfy the complementarity conditions as seen in the right figure, however, it is still high compared to the rest of the domain. This problem is repeated in the next v-cycle.

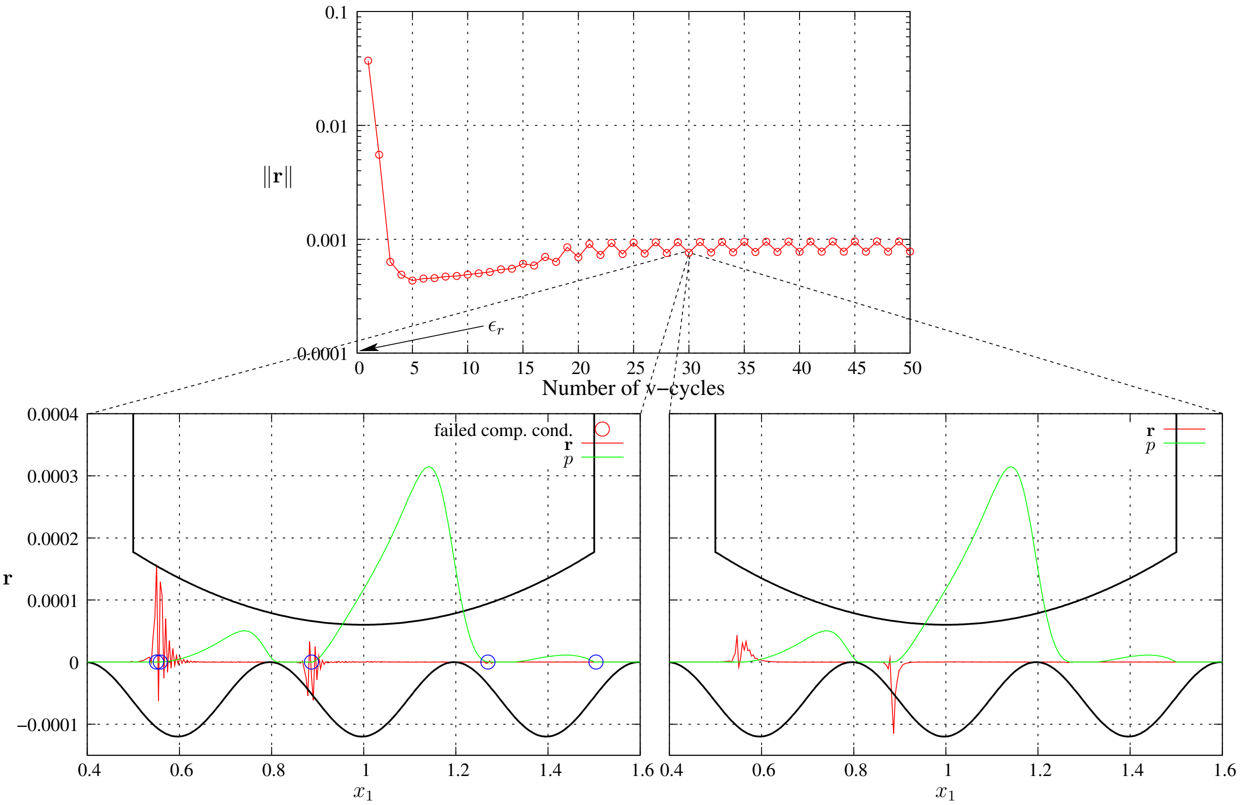


Figure 5: *Top*: Residue at the end of each v-cycle performed for the second time step of simulation 8 of Table 1. *Bottom*: Residue at every cell in the domain for v-cycle 30, *Left*: previous to relaxation and *Right*: after one Gauss-Seidel relaxation. In the left figure the solution from mesh l=1 was interpolated to mesh l=0 (finest mesh). The blue circles show the location of the cells where the complementarity conditions are not being met.

#	Algorithm	$\boldsymbol{\nu}^u$	iterations in fine mesh	Wu per v-cycle	Total Wu	$\|\mathbf{r}\| > \epsilon_r$?	Speedup
1	1	—	106,999,200	—	106,999,200	0.0 %	1.0
2	2	{ 1 1 1 1 }	1,222,974	2.64	1,614,325	8.7 %	66.3
3	2	{ 2 2 2 2 }	1,286,316	3.97	1,702,224	12.3 %	62.9
4	2	{ 4 4 4 4 }	1,176,505	6.64	1,561,222	6.6 %	68.5
5	2	{ 8 8 8 8 }	806,814	11.94	1,070,149	1.8 %	100.0
6	2	{ 16 16 16 16 }	909,211	22.56	1,206,576	1.0 %	88.7
7	2	{ 32 32 32 32 }	1,244,793	43.81	1,652,557	0.6 %	64.7
8	2	{ 1 1 1 40 }	2,243,288	3.25	3,645,343	69.7 %	29.4
9	2	{ 16 32 64 128 }	876,112	31.31	1,613,721	1.8 %	66.3
10	3	{ 1 1 1 1 }	1,401,801	2.64	1,849,426	0.0 %	57.9
11	3	{ 2 2 2 2 }	1,004,439	3.97	996,905	0.07 %	107.3
12	3	{ 4 4 4 4 }	877,792	6.64	726,922	0.0 %	147.2
13	3	{ 8 8 8 8 }	852,668	11.94	636,176	0.0 %	168.2
14	3	{ 16 16 16 16 }	1,142,736	22.56	805,628	0.0 %	132.8
15	3	{ 32 32 32 32 }	1,650,624	43.81	1,129,904	0.0 %	94.7
16	3	{ 1 2 4 8 }	604,427	3.19	963,305	0.16 %	111.1
17	3	{ 2 4 8 16 }	456,960	5.06	771,120	0.07 %	138.8
18	3	{ 4 8 16 32 }	284,616	8.81	501,636	0.0 %	213.3
19	3	{ 8 16 32 64 }	302,623	16.31	548,504	0.0 %	195.1

Table 1: Computational cost and efficiency for Algorithms 1, 2 and 3.

To circumvent this situation, the most forthright solution is increasing $\boldsymbol{\nu}_l^u$ at each multigrid level l. This is demonstrated in lines 2 to 7 of Table 1. From $\boldsymbol{\nu}_l^u$ = 1 to 32 $\forall$ l the percentage of time steps where the residue stalls is reduced from 8.7% to 0.6%, although with a increase in the computational cost. Merely increasing the number of iterations in the coarser meshes does not seem to bode well, as seen in the increase of column 7 for lines 8 and 9. These results suggest the following adaptive algorithm:

Algorithm 3 Adaptive multigrid algorithm to solve problem (6)-(15)

```
Provide initial values, $Z^0 = z_0$, $(Z')^0 = v_0$, $\boldsymbol{\omega}_i^0 = (\theta_I)_i \; \forall i \in I$
$t = 0$, $n = 1$
**while** $t \leq T$ **do**
    $q = 0$, $Z^q = Z^{n-1}$, $k = 0$
    $\|\mathbf{r}^{q=0}\| = \infty$, $\boldsymbol{\nu}^u = \boldsymbol{\nu}^u_{\text{default}}$
    **while** $|Z^{q+1} - Z^q| > \epsilon_Z$ and $\|\mathbf{r}\| > \epsilon_r$ **do**
        Perform $K$ iterations of $\boldsymbol{\omega}^{k+K} = \mathrm{M}^K(\boldsymbol{\omega}^k)$
        **if** $\|\mathbf{r}\| \geq \|\mathbf{r}^q\|$ **then**
            $\boldsymbol{\nu}^u \to \boldsymbol{\nu}^u + \boldsymbol{\nu}^u_{\text{default}}$
        **end if**
        $\|\mathbf{r}^q\| = \|\mathbf{r}\|$
        Compute $W^h(t)$
        Obtain $Z^{q+1}$ by performing one iteration of the Newton or secant method on eq. (34)
        Update $h(x_1, x_2, t)^{q+1}$
        $q = q + 1$
    **end while**
    Compute $(Z')^n$ according to eq. (25)
    $Z^n = Z^q$
    $\boldsymbol{\omega}^n = \boldsymbol{\omega}^k$
    $n = n + 1$
**end while**
```

Modifications from Algorithm 2 are shown in red. The idea behind it is to increase the number of relaxations $\boldsymbol{\nu}^u$ if the residue does not decrease. In Algorithm 3 it is done by adding a fixed number of relaxations $\boldsymbol{\nu}^u_{\text{default}}$, which is completely arbitrary, any other strategy could be devised to increase the number of relaxations when the residue is not decreased.

Lines 10 to 19 of Table 1 corresponds to this adaptive approach. The number of time steps where the residue fails to be reduced to the tolerance is much smaller or in most cases null than with Algorithm 2. From line 10 to 15, the workload increased (14.5%) only for $\boldsymbol{\nu}^u = \{1\;1\;1\;1\}$ compared with their Algorithm 2 counterparts.

Increasing the number of iterations in the coarser meshes in Algorithm 3 decreases considerably the number of v-cycles required, as seen in lines 16 to 19 of Table 1, thus lower computational times. For $\boldsymbol{\nu}^u = \{4\;8\;16\;32\}$ the tolerance was achieved in all time steps. The speedup, here defined as the quotient of the total work units required by the non-multigrid Algorithm 1 divided by the ones required by Algorithm 2 or 3, was around 200 for simulations 18 and 19 of Table 1.

4.2.2 With operator $\mathcal{R}_A$

In a single-level algorithm it has been shown that both $\mathcal{R}_{Alt}$ and $\mathcal{R}_A$ return the same results with comparable efficiency. However, several problems arise with the use of operator $\mathcal{R}_A$ in Algorithm 2 (and 3):

1. The problem with the prolongation operators as explained in Remark 1. This begets a much bigger problem with operator $\mathcal{R}_A$, since the former checks whether a cell is pressurized ($\mathbf{w}_i \geq 1.0$) or cavitated ($\mathbf{w}_i < 1.0$), rather than mass-balance as with $\mathcal{R}_{Alt}$, and thus requires a $\mathbf{w}$ satisfying $\mathbf{w}_i \geq 0$;

2. The problem explained in Remark 2. In a single-grid algorithm this does not make a difference in the final solution as shown in the numerical examples. However, in a multigrid algorithm it does, as the residue in a cell where the solution has not been correctly computed is high, and can lead to negative right hand sides in equation (36) for the problems defined in the coarser meshes.

Modified prolongation operators exploiting some heuristics to effectively reduce the residue with the algorithm originally proposed by Ausas are given in Checo's PhD thesis (Checo, 2016). However, the resulting algorithm is much more complicated than Algorithm 2 using operator $\mathcal{R}_{Alt}$, which allows within certain limitations (the need of increasing $\boldsymbol{\nu}^u$) to make use of a straightforward implementation of the *Full Approximation Scheme*.

5 CONCLUSIONS

In this work we have presented two relaxation operators to solve the discrete Elrod-Adams equations, the first one originally presented in Ausas et al. (2009) and another one based on Alt's Alt (1980) algorithm to solve multiphase stationary flow in porous media. The former is based on a cavitation check while the latter on mass-balance at each finite volume.

On a single-grid algorithm both operators showed similar performance, however, differences arise with multigrid algorithms. On a full approximation scheme the prolongation operators do not preserve the complementarity conditions, which has a different impact on both approaches. The Alt based operator proved to be more robust to this problem. A simple adaptive strategy based on the residue of the Elrod-Adams equation was devised, which proved efficient. Solving the problems emerging from a multigrid implementation using the other operator proved to be much more difficult, and not as efficient. A strategy to do so is presented in Checo (2016).

6 ACKNOWLEDGMENTS

The authors want to thank the financial support of this work provided by CAPES (grant 150283/2016-4).

REFERENCES

Alt H.W. Numerical solution of steady-state porous flow free boundary problems. *Numerische Mathematik*, 36(1):73–98, 1980. ISSN 0945-3245. doi:10.1007/BF01395990.

Alvarez S. *Problemas de frontera libre en teoría de lubricación*. Ph.D. thesis, Universidad Complutense de Madrid, Spain, 1986.

Ausas R., Jai M., and Buscaglia G. A Mass-Conserving Algorithm for Dynamical Lubrication Problems With Cavitation. *ASME Journal of Tribology*, 131:031702 (7 pages), 2009.

Bayada G. and Chambat M. Existence and uniqueness for a lubrication problem with non regular conditions on the free boundary. *Boll. U.M.I.*, 6(3b):543–557, 1984.

Bayada G. and Chambat M. Sur quelque modelisation de la zone de cavitation en lubrification hydrodynamique. *Journal de mécanique théorique et appliqué*, 5:703–729, 1986.

Bayada G., Chambat M., and Faure J. Some effects of the boundary roughness in a thin film. Boundary control and boundary variations. *Lecture Notes in Compu Sci.*, 100:96–115, 1988.

Bayada G., Chambat M., and Faure J. A Double Scale Analysis Approach of the Reynolds Roughness. Comments and Application to the Journal Bearing. *ASME Journal of Tribology*, 111:323–330, 1989.

Buscaglia G., Ciuperca I., and Jai M. Homogenization of the transient Reynolds equation . *Asymptotic Analysis*, 32:131–152, 2002.

Buscaglia G., El Alaoui Talibi M., and Jai M. Mass-conserving cavitation model for dynamical lubrication problems. Part I: Mathematical analysis. *Math. Comput. Simulation*, 2015.

Buscaglia G. and Jai M. Homogenization of the Generalized Reynolds Equation for Ultra-Thin Gas Films and Its Resolution by FEM . *ASME Journal of Tribology*, 126:547, 2004.

Checo H. *Models and methods for the direct simulation of rough and micropatterned surfaces*. Ph.D. thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2016.

Christensen H. Stochastic Models of Hydrodynamic Lubrication of Rough Surfaces. *Proc, Int. Mech. Eng*, 55:1013, 1969.

El Alaoui A. *Sur un problème à frontière libre en mécanique des films minces*. Ph.D. thesis, Univ. Claude Bernard Lyon I, Lyon, France, 1986.

El Alaoui Talibi M. and Jai M. Existence and numerical results of a transient lubrication problem with cavitation. *Journal of Mathematical Analysis and Applications*, 394:260–275, 2012.

Elrod H.G. and Adams M. A computer program for cavitation. Technical report 190. *1st LEEDS LYON Symposium on Cavitation and Related Phenomena in Lubrication, I.M.E.*, 103:354, 1974.

Fulton S.R., Ciesielsky P.E., and Schubert W.H. Multigrid methods for elliptic problems: a review. *Mon. Wea. Rev.*, 114:943–959, 1986.

Jai M. Homogenization and two-scale convergence of the compressible Reynolds lubrication equation modeling the flying characteristics of a rough magnetic head over a rough rigid-disk surface. *Math. Modeling and Numer. Anal.*, 29:199–233, 1995.

Patir N. and Cheng H. An Average Flow Model for Determining Effects of Three-Dimensional Roughness on Partial Hydrodynamic Lubrication. *Transactions of the ASME*, 100:12–17, 1978.

Patir N. and Cheng H. Application of Average Flow Model to Lubrication Between Rough Sliding Surfaces. *Transactions of the ASME*, 101:220–229, 1979.

Tzeng S. and E. S. Surface Roughness Effect on Slider Bearing Lubrication. *ASLE Transactions*, 10:334–338, 1967.