Computer Assisted Mechanics and Engineering Sciences, 7: 363–376, 2000.

Numerical simulation of frontal polymerization

Thierry Dumont
Laboratoire d'Analyse Numérique, Université Lyon 1, CNRS, Bât. 101,
43 Bd. du 11 Novembre, 69621 Villeurbanne Cedex, France

(Received August 16, 1999)

Frontal polymerization has been studied for many years experimentally and theoretically. This technique can exhibit a phase change between a liquid monomer and a solid polymer and many studies, both theoretical and experimental, have been devoted to the stability of the front which separates the two phases in the presence of thermal convection. We present here a new technique for the numerical simulation of this process which takes account of the chemical reaction, the phase transition and the hydrodynamics; it is based on the method of characteristics and a fictitious domain method. These two methods are known, but the coupling of them and the application to this problem is new. We also present and discuss some results of simulations.

1. Introduction

Frontal polymerization is a technique for preparing polymers via a propagating thermal front which converts a liquid monomer to a polymer. A typical experiment is shown in Fig. 1: the monomer is put in a cylinder, and the polymerization is initiated near a boundary (by heating the bottom for example). The reaction is localized in space and it propagates along the cylinder. In front of the reaction zone there is a cold monomer and a hot polymer behind it. The polymer can be in a liquid or in a solid phase. We are interested here in the latter case: there is a phase transition and the liquid–solid interface propagates together with the reaction zone. An example of this process can be given by the frontal polymerization of methyl–methacrylate [14]. The stability of the reaction front is a fundamental concern, the key for industrial processes to get high quality polymers at the end of the process. Ground based experiments show that the ascending fronts are generally unstable, due to thermal convection. A stability analysis is fulfilled in [6]. Low gravity studies have been performed, but the effect of convection remains to be explored (see [7, 15] for the descriptions of the experiments). Numerous studies of combustion and polymerization fronts have shown that the detailed chemical kinetics is not usually very essential for the peculiarities of the front propagation [22]. In this work we study a simplified model where we assume that the chemical kinetics can be described by a one–step reaction. In the case of a radical polymerization for example, this assumption means that the concentration of the initiator is sufficiently large.

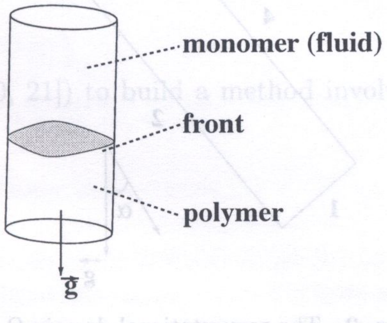

Fig. 1. Frontal polymerization

The coupling of the chemical reaction, phase transition and hydrodynamics makes the problem difficult to simulate numerically. There exist different numerical methods to take account of a phase change (see [17] for a review). They can be divided into two groups: the variable grid methods, where a mesh is adapted at each time step to the front, and the fixed grid methods. The second group seems to be better adapted to our problem where some equations need to be solved on the whole domain, and we use this approach.

2. Mathematical model

The model consists of the reaction–diffusion equations coupled with the Navier–Stokes equations for the fluid:

$$\begin{aligned} \partial_t T + (\vec{U}\cdot\nabla)\,T - \kappa\,\Delta T &= q\,W(C,T)\,, \\ \partial_t C + (\vec{U}\cdot\nabla)\,C - \varepsilon\,\Delta C &= W(C,T)\,, \end{aligned} \tag{1}$$

$$\left\{ \begin{aligned} \partial_t\vec{U} + (\vec{U}\cdot\nabla)\,\vec{U} - \nu\Delta\vec{U} + \frac{1}{\rho}\nabla P &= k_0\,\vec{g}\,(T-T_0)\,, \\ \nabla\cdot\vec{U} &= 0\,. \end{aligned} \right. \tag{2}$$

Here, T is the temperature, C is the depth of conversion ($0 \leq C \leq 1$) and $\vec{U}$ is the velocity of the fluid. κ is the thermal diffusivity, ε is a diffusion coefficient, ν is the kinematic viscosity of the fluid, ρ is the density, k_0 is the thermal expansion coefficient, q is the adiabatic heat release and $\vec{g}$ is the gravitational acceleration. For the Navier–Stokes equations, we have made the assumption that the density in all terms is constant, except for the buoyancy term (Boussinesq approximation).

The reaction rate $W(C,T)$ is given by the mass action law with the Arrhenius temperature dependence

$$W(C,T) = k\,(1-C)\,e^{-E/RT}\,. \tag{3}$$

Here, R is the ideal gas constant, E is the activation energy and k is the pre-exponential factor. Real experimental domains for this process are typically cylindrical tubes (see Fig. 1), but we restrict our simulations to a two dimensional rectangular domain Ω (see Fig. 2).

Equations (1) and (2) must be completed by initial and boundary conditions:

- At time $t=0$, we take $\vec{U}=0$, $T=T_0$ (a constant positive value), and $C=0$ everywhere in Ω (for C this means that we have only the monomer, everywhere in the domain).
- For the boundary conditions, we take $\vec{U}=0$ on all the boundary of Ω. We take the flux $\partial T/\partial n = 0$ on the "large" walls (2) and (4) (see Fig. 2), $T=T_0$ on the wall (3), and $T=T_1$ on the wall (1), with $T_1 > T_0$. We impose $\partial C/\partial n = 0$ on the boundary of Ω.

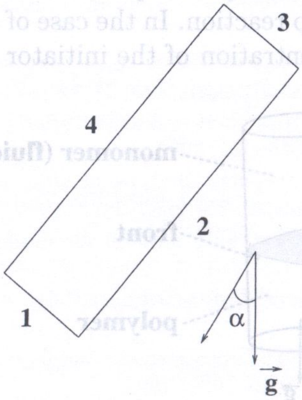

Fig. 2. The computational domain Ω

Other initial and boundary conditions can be taken as well, but these ones closely mimic the real experiments.

The phase transition between the liquid monomer and the solid polymer occurs when the conversion C exceeds a critical value C_0. We will put $C_0 = 0.5$. We should note that the choice of the critical value is not very essential because the reaction zone is narrow [19]. The evolution of this system (depending on the choice of the parameters and of the orientation of Ω) is the following: a reaction front propagates, starting from wall (1) (see Fig. 2), and the phase change occurs near this front: the domain Ω is splitted into two subdomains Ω_0 (the fluid phase), and Ω_1 (the solid phase). When the angle α is near 0, laboratory experiments show that the ascending front which separates Ω_0 and Ω_1 can be unstable; with $\alpha = \pi$ no convection occurs and, with our model, the descending front is stable.

3. Numerical method

Even without the phase change, the model (1), (2), (3) exhibits numerical difficulties:

1. this is a coupled system of nonlinear equations;
2. this system is convective;
3. we have the usual difficulties of the discretisation of the incompressible Navier–Stokes equations (i.e. we need a compatible choice of the discrete velocity and pressure fields, see [12]).

3.1. Spatial discretisation

- for the Navier–Stokes equations, we have chosen the classical Taylor–Hood finite element: P_2 (second order polynomial approximation) for the velocity and P_1 (first order polynomial approximation) for the pressure, on embedded grids. This finite element respects the well known Inf–Sup condition [12];
- for the temperature T and for the depth of conversion C, we use P_2 approximations on the same grid as the velocity.

These discretisations are of the second order (see [10, 12]).

3.2. Time discretisation

Having in mind that the phase change will complicate the problem, we must split our system of equations to be able to solve it in a reasonable computing time. Splitting of non-linear PDE is generally performed by some alternate direction method: for example, if we want to solve numerically the scalar equation

$$\frac{\partial v}{\partial t} = \Delta v + f(v),$$

it is classical (see for example [20, 21]) to build a method involving approximate integration over time steps of

$$\frac{\partial v}{\partial t} = f(v), \tag{4}$$

and

$$\frac{\partial v}{\partial t} = \Delta v. \tag{5}$$

More precisely, let $\delta t > 0$ be the time step of the discretisation and $t_n = n\,\delta t$. Let v_n be an approximation of v at time t_n; we can compute the approximation of v_{n+1} at time t_{n+1} in two sub-steps:

$$\frac{v_{n+1/2} - v_n}{\delta t} = f(v_{n+1/2})\,, \tag{6}$$

and

$$\frac{v_{n+1} - v_{n+1/2}}{\delta t} = \Delta v_{n+1}\,. \tag{7}$$

As the computations must be done only at the nodes of the finite element discretisation (or at quadrature points), we see that the first step can be performed by solving a finite set of independent algebraic equations. Note that:

- the two sub-steps (6), (7) are discretisations of (4) and (5) by the implicit Euler method. It is possible to use the explicit Euler method in one or both step, but the resulting scheme is less stable;
- a formal application of the Taylor formula shows that this scheme is of the first order.

Let us now consider a scalar advection–diffusion equation:

$$\frac{\partial v}{\partial t} + \nabla\cdot\left(\vec{\beta}(\vec{x},t)v\right) - \Delta v = f(\vec{x})\,, \qquad \nabla\cdot v = 0 \tag{8}$$

for which we modify the sub-step (6). We consider for any $\vec{x} \in \Omega$, the characteristic curve $\vec{X}_{\vec{x}}^{n+1}(t)$ issued from $\vec{x}$ at time t_{n+1} defined by the ordinary differential equation (integrated backward in time):

$$\begin{cases} \dfrac{\mathrm{d}\vec{X}_{\vec{x}}^{n+1}(s)}{\mathrm{d}t} = \vec{\beta}\left(\vec{X}_{\vec{x}}^{n+1}(s), s\right), \\ \vec{X}_{\vec{x}}^{n+1}(t_{n+1}) = \vec{x}\,. \end{cases} \tag{9}$$

The solution at time t_{n+1} of

$$\begin{cases} \dfrac{\partial v}{\partial t} + \nabla\cdot\left(\vec{\beta}(\vec{x},t)v\right) = f(\vec{x})\,, \\ v(\vec{x},t_n) = v_n(\vec{x})\,, \end{cases}$$

is exactly

$$v_{n+1/2} = v_n\left(\vec{X}_{\vec{x}}^{n+1}(t_n)\right) + \int_{t_n}^{t_{n+1}} f\left(\vec{X}_{\vec{x}}^{n+1}(s)\right)\mathrm{d}s\,. \tag{10}$$

For practical computation, Eq. (10) must be discretized. A classical choice is to approximate the characteristic curves defined by (9) by the Euler method, replacing the exact solution by

$$\tilde{X}_{\vec{x}}^{n+1}(t_n) = \vec{x} - \vec{\beta}(\vec{x}, t_{n+1})\,\delta t,$$

and the integral in (10) by $\delta t\ f(\vec{x})$. Then, (10) can be rewritten as

$$\frac{v_{n+1/2} - v_n\left(\tilde{X}_{\vec{x}}^{n+1}(t_n)\right)}{\delta t} = f(\vec{x})\,.$$

We have replaced the left side of (6) by a discretisation of the material derivative of v. This method, together with the diffusion step (7) is known as the method of characteristics. It is a first order method, which results in a stable discretisation of the convective terms [3].

Let us now return to our problem. As a result of personal numerical experiments, as well as of some benchmarks [16], it seems that a correct qualitative simulation of convection diffusion processes must involve a second order time discretisation (at least). A second order method of characteristics has been developed by Boukir, Maday and Métivet [5]. This method is based on the second order backward differentiation formula (BDF2) [11]. Let us recall that, for an ordinary differential equation like $\mathrm{d}v/\mathrm{d}t = f(v)$, the BDF2 formula defines v_{n+1} as a function of the previous known values v_n and v_{n-1} at time t_n and t_{n-1} as

$$\frac{3\,v_{n+1} - 4\,v_n + v_{n-1}}{2\,\delta t} = f(v_{n+1}) \,.$$

For the advection–diffusion equation, with the second order method of characteristics of Boukir, Maday and Métivet, v_{n+1} is given by

$$\frac{3\,v_{n+1} - 4\,w_1 + w_2}{2\,\delta t} - \Delta v_{n+1} = f_2(v_{n+1}) \,.$$

where

$$w_i = v_{n+1-i}\left(\tilde{X}^{n+1}_{\vec{x}}(t_{n+1-i})\right) + \int_{t_{n+1-i}}^{t_{n+1}} f_1\left(\tilde{X}^{n+1}_{\vec{x}}(s)\right)\,\mathrm{d}s \,. \tag{11}$$

Here, we have decomposed f in two parts: $f = f_1 + f_2$. This decomposition is arbitrary. As proven in [5] the characteristic curves in (11) can be approximated by the (first order) Euler method, but the integrals must be approximated by a second order method, like the trapezoid rule. For Navier–Stokes equations and for our problem, the convection field $\vec{\beta}$ is the unknown velocity $\vec{U}$ and we must replace it by a suitable approximation. The following is an adaptation of this method to the system (1), (2), (3).

Denote $\vec{U}^k$, T^k and C^k the approximations of $\vec{U}$, T and C at time $t_k = k\,\delta t$. We compute the approximations $\vec{U}^{n+1}$, T^{n+1} and C^{n+1} at time $t_{n+1} = (n+1)\,\delta t$ in four sub-steps:

First step: definition of a convection field and approximation of the characteristic curves

As explained above, we must define a convection field, as $\vec{U}^{n+1}$ is unknown. For this, let

$$\vec{U}^{n+1}_c = 2\,\vec{U}^n - \vec{U}^{n-1} \,,$$

which is a second order approximation of the velocity at time $t_{n+1} = (n+1)\,\delta t$. We approximate the characteristic curves $\vec{X}^{n+1}_{\vec{x}}(t)$ by:

$$\begin{cases} \dfrac{\mathrm{d}\vec{X}^{n+1}_{\vec{x}}}{\mathrm{d}t} &= \vec{U}^{n+1}_c\left(\vec{X}^{n+1}_{\vec{x}}(t)\right), \\ \vec{X}^{n+1}_{\vec{x}}(t_{n+1}) &= \vec{x}, \qquad t \in (t_{n-1}\,,\,t_{n+1}). \end{cases} \tag{12}$$

These differential equations need only to be computed at the quadrature nodes of the finite elements mesh. They are discretized to get approximate solutions at t_n and t_{n-1} ; as it is proved in [5], one can use a first order method like the explicit Euler formula (i.e. $\vec{X}^{n+1}_{\vec{x}}(t_{n+1-i}) = \vec{x} - i\,\delta t\,\vec{U}^{n+1}_c(\vec{x})$), without destroying the second order of the scheme. In practice, a good choice is to take a second order explicit formula (like the second order Runge–Kutta method) to reduce the numerical diffusion.

Second step: convection of temperature and concentration

Let us define:

$$f_c^{j,l}(C,T,\vec{x},t) = \frac{t-t_j}{t_l-t_j} W(C(\vec{x}),T(\vec{x})) + \frac{t_l-t}{t_l-t_j} W(C^j(\vec{x}),T^j(\vec{x})). \tag{13}$$

We compute $\bar{T}_1$, $\bar{T}_2$:

$$\bar{T}_1(\vec{x}) = T^n\left(\tilde{X}_{\vec{x}}^{n+1}(t_n)\right) + q\int_{t_n}^{t_{n+1}} f_c^{n,n+1}\left(\bar{C}_1,\bar{T}_1,\tilde{X}_{\vec{x}}^{n+1}(s),s\right)\,\mathrm{d}s, \tag{14}$$

$$\bar{T}_2(\vec{x}) = T^{n-1}\left(\tilde{X}_{\vec{x}}^{n+1}(t_{n-1})\right) + q\int_{t_{n-1}}^{t_{n+1}} f_c^{n-1,n+1}\left(\bar{C}_2,\bar{T}_2,\tilde{X}_{\vec{x}}^{n+1}(s),s\right)\,\mathrm{d}s, \tag{15}$$

and $\bar{C}_1$, $\bar{C}_2$:

$$\bar{C}_1(\vec{x}) = C^n\left(\tilde{X}_{\vec{x}}^{n+1}(t_n)\right) + \int_{t_n}^{t_{n+1}} f_c^{n,n+1}\left(\bar{C}_1,\bar{T}_1,\tilde{X}_{\vec{x}}^{n+1}(s),s\right)\,\mathrm{d}s\,, \tag{16}$$

$$\bar{C}_2(\vec{x}) = C^{n-1}\left(\tilde{X}_{\vec{x}}^{n+1}(t_{n-1})\right) + \int_{t_{n-1}}^{t_{n+1}} f_c^{n-1,n+1}\left(\bar{C}_2,\bar{T}_2,\tilde{X}_{\vec{x}}^{n+1}(s),s\right)\,\mathrm{d}s\,. \tag{17}$$

This is an adaptation of (11). The formula (13) for $f_c^{j,l}$ is not the same as in Boukir [4] where she used, for Navier–Stokes equations, a second order extrapolated formula which we can rewrite here like:

$$f_c(\vec{x}) = 2\,W(C^n(\vec{x}),\,T^n(\vec{x})) - W(C^{n-1}(\vec{x}),\,T^{n-1}(\vec{x})).$$

With this last choice of f_c and with the trapezoid rule, the resulting scheme for this step is explicit. This method is not surprisingly unstable in our problem. Proofs of consistency and order can be modified for our formula (13) from the original paper [4] in a straightforward way.

Our choice of $f_c^{j,l}$ and the trapezoid rule make the step implicit; we obtain algebraic systems of equations for $\bar{T}_1(\vec{x})$ and $\bar{C}_1(\vec{x})$ (and corresponding systemes for $\bar{T}_2(\vec{x})$ and $\bar{C}_2(\vec{x})$):

$$\begin{cases} \bar{T}_1(\vec{x}) = T^n(\bar{x}) + q\,(\delta t/2)\left[W(\bar{C}_1(\vec{x}),\bar{T}_1(\vec{x})) + H\right], \\ \bar{C}_1(\vec{x}) = C^n(\bar{x}) + (\delta t/2)\left[W(\bar{C}_1(\vec{x}),\bar{T}_1(\vec{x})) + H\right], \end{cases} \tag{18}$$

with $\bar{x} = \tilde{X}_{\vec{x}}^{n+1}(t_n)$. Here $H = W(C^n(\bar{x}),\,T^n(\bar{x}))$ is known. These algebraic problems are local: this is what provides the uncoupling of the system. They can be easily solved by Newton iterations for each node $\vec{x}$ of the finite element discretisation (or at quadrature nodes). Note that we can even eliminate one of the unknowns.

Third step: diffusion

We calculate T^{n+1} as a solution of

$$\frac{3\,T^{n+1} - 4\,\bar{T}_1 + \bar{T}_2}{2\,\delta t} - \kappa\,\Delta T^{n+1} = 0\,, \tag{19}$$

(with the corresponding boundary conditions), and C^{n+1} as a solution of

$$\frac{3\,C^{n+1} - 4\,\bar{C}_1 + \bar{C}_2}{2\,\delta t} - \varepsilon\,\Delta C^{n+1} = 0\,, \tag{20}$$

(with homogeneous Neuman boundary conditions).

Fourth step: resolution of a Stokes problem

We compute

$$\bar{U}_1(\vec{x}) = \vec{U}^n\Big(\tilde{X}^{n+1}_{\vec{x}}(t_n)\Big), \quad \bar{U}_2(\vec{x}) = \vec{U}^{n-1}\Big(\tilde{X}^{n+1}_{\vec{x}}(t_{n-1})\Big)$$

and finally $\vec{U}^{n+1}$, solution of the generalized Stokes problem:

$$\frac{3\,\vec{U}^{n+1} - 4\,\bar{U}_1 + \bar{U}_2}{2\,\delta t} - \nu\,\Delta\vec{U}^{n+1} + \frac{1}{\rho}\nabla P = k_0\,\vec{g}\,(T^{n+1} - T_0), \qquad \nabla\cdot\vec{U}^{n+1} = 0, \qquad \vec{U}^{n+1}_{\partial\Omega} = 0. \tag{21}$$

3.3. Simulation of the phase transition by a fictitious domain method

Let $\Omega_1 = \{\vec{x}\in\Omega;\ C(\vec{x}) \geq C_0\}$ be the solid polymerized zone and $\Omega_0 = \Omega\backslash\Omega_1$ the fluid domain. We want to impose the velocity of the fluid to be approximately zero in Ω_1. For this, we use a penalisation method.

Consider the generalized Stokes problem:

$$\zeta\vec{U} - \nabla\cdot\Big(\nu(\nabla\vec{U} + \nabla\vec{U}^t)\Big) + \nabla P = f; \quad \nabla\cdot\vec{U} = 0, \quad \vec{U}_{\partial\Omega} = 0. \tag{22}$$

Set ε a penalisation parameter ($0 < \varepsilon \ll 1$) and choose:

$$\zeta(\vec{x}) = \zeta_0, \qquad \nu(\vec{x}) = \nu_0, \qquad \forall x\in\Omega_0, \tag{23}$$

$$\zeta(\vec{x}) = \zeta_0/\varepsilon, \qquad \nu(\vec{x}) = \nu_0/\varepsilon, \qquad \forall x\in\Omega_1. \tag{24}$$

When ε is small enough, $\vec{U}$ will be close to zero in Ω_1. Trying to solve directly this problem would be practically impossible, due to the bad condition number of the linear system, but this can be done in a performing way by the fictitious domain method of Bakhvalov [1]. We describe it shortly.

Let

$$V = \{\vec{v}\in H^1_0(\Omega)^2;\quad \nabla\cdot\vec{v} = 0;\quad \vec{v}_{\partial\Omega} = 0\},$$

$$S_\varepsilon(\vec{v},\phi) = \int_\Omega (\zeta\vec{v}\phi + \nu(\nabla\vec{v} + \nabla\vec{v}^t)\nabla\phi)\,\mathrm{d}\Omega,$$

and S_1 be the corresponding formula for the constant coefficient problem (i.e. $\varepsilon = 1$). Set

$$W = \Big\{\vec{v}\in V;\quad \int_{\Omega_0} \zeta(\vec{v}-\vec{U})\vec{\phi} + \nu\left(\nabla(\vec{v}-\vec{U}) + \nabla(\vec{v}-\vec{U})^t\right)\nabla\vec{\phi} = 0, \quad \forall\vec{\phi}\in H^1(\Omega_0)^2, \quad \vec{\phi}_{\partial\Omega_0} = 0, \quad \nabla\cdot\vec{\phi} = 0\Big\}.$$

The solution U of (22), (23), (24) is the solution of the variational problem:

$$S_\varepsilon(\vec{U},\vec{\phi}) = \int_\Omega f\vec{\phi} \qquad \forall\vec{\phi}\in V. \tag{25}$$

In [1], it is proved that the linear operators corresponding to S_ε and S_1 are spectrally equivalent on W. As these two operators are positive definite on W we can solve (25) by the conjugate gradient method, using the constant coefficient generalized Stokes operator S_1 as preconditioner. If we take the starting vector of the iterations in W the iterations will remain in W (see [1]), and we will obtain a rate of convergence which depends only of the domain Ω_1 (and not of the mesh size, nor of ε).

3.4. The complete algorithm

The modification of our splitting method is obvious. We choose ε (as small as possible). Once we have computed T^{n+1} and C^{n+1}, we modify the last step to introduce the fictitious domain method: first we compute the set Ω_1, then after modification of the coefficients of (21) in Ω_1 like in (22), (23), (24), we solve this modified system, imposing $\vec{U}^{n+1} \simeq 0$ on Ω_1 :

Modified fourth step of 3.2:

1. compute the domain Ω_1 as a function of C^{n+1},
2. choose a starting vector in W (a solution of $S_1(\vec{v}, \vec{\phi}) = (f, \vec{\phi})$ by example),
3. solve the modified Stokes system by the gradient method preconditioned by S_1.

Note that we solve only constant coefficients generalized Stokes problems.

3.5. Implementation

To achieve a good performance (and also to simplify the implementation) we have used rectangular structured grids. This allows a very simple and very efficient implementation of the method of characteristics. We solve the Stokes problems by the conjugate gradient method on the pressure Schur complement, with a Poisson equation on the pressure as preconditioner [8]. Then all the linear systems that we need to solve are Poisson equations on the different grids, for which we have at our disposal fast algorithms like multigrids or the cyclic reduction method [18]. The algorithm is easy to implement and it seems to be efficient; but the computing time remains large (16 hours on an INTEL–PII 450 Mhz for the experiment shown on Fig. 4). We must remark that the fictitious domain method of Bakhvalov cannot be easily improved as it furnishes an optimal (spectrally equivalent) preconditioner. Improvements can be implemented for the Stokes solver (the inner loop!) embedded in the method, for which we have used a very classical preconditioned Uzawa method: multigrid methods on the Stokes problem could certainly achieve much better performances; an adaptive time step method could also improve largely this method. Another point which needs to be explored is the use of adaptive refinement for the meshes, which would certainly give a better description of the front and of the related phenomena in its vicinity.

We have limited our simulations to two dimensional problems because we lack an efficient three dimensional Stokes solver, but there is no theoretical limitation to dimension 2 for this method. Moreover the fictitious domain method, having an optimal rate of convergence (independent of the mesh size), should give good results in dimension 3.

We have tried other time discretisations: it seems interesting to use the fictitious domain method with Chorin–Teman and related schemes [9, 4, 13]: but all these methods impose very small time steps and the global iteration process could not compete with our scheme.

4. Numerical results

Let us recall that that we are looking for conditions where the front remains almost plane along the process: this is the interesting case for applications. Otherwise, the front is considered to be unstable. We show here two computations with different (albeit arbitrary) values of α.

- Figure 3 shows a case where the inclination of the tube is $\alpha = 3\pi/4$. The front is stable, descending, and propagates with a constant velocity. Laboratory experiments have been conducted with a good agreement with the numerical simulations [2]).

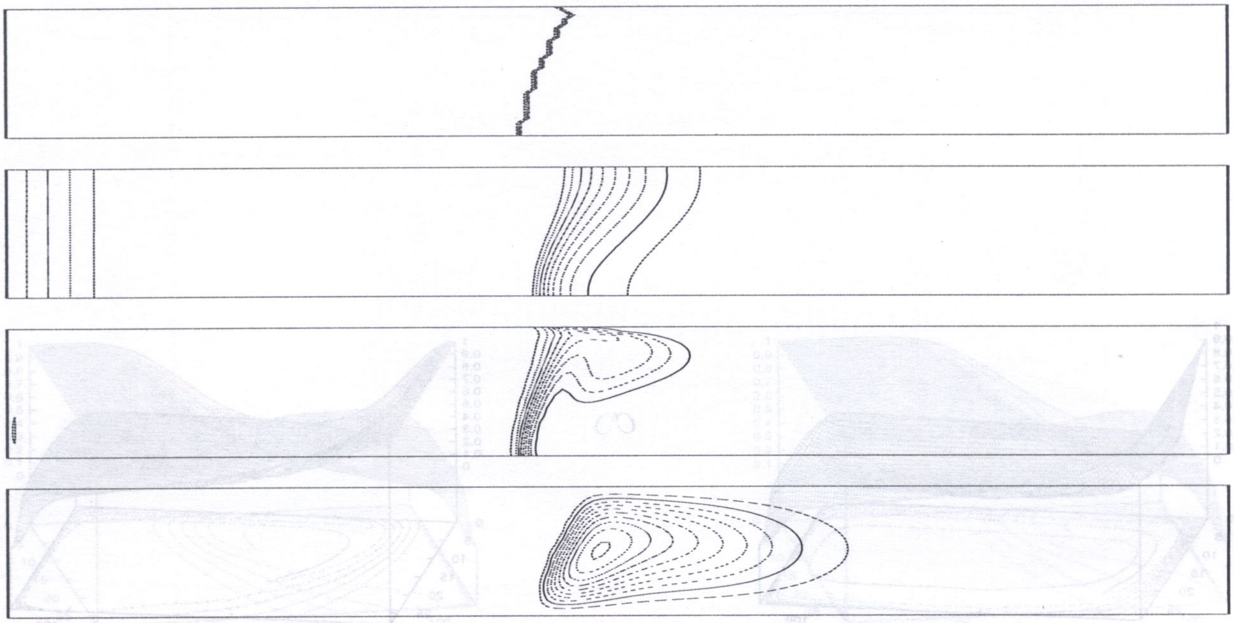

Fig. 3. From top to bottom: snapshot of the front, T, C and the stream function ψ (isovalues have been drawn by increment of 10 degrees K (T), 0.1 (C) and -0.01 (ψ))

- Figure 4 shows the instability of a vertical ascending front ($\alpha = \pi$). This is in agreement with the experiments of [7]. Figure 5 shows the corresponding stream function.

In two dimensional fluid simulations, the stream function ψ is often computed using the relation $\Delta\psi = -\operatorname{curl}\vec{U}$. Here we use the relation

$$\psi(\vec{x}) = \int_{l(\vec{x})} U_2\,\mathrm{d}x_1 - U_1\,\mathrm{d}x_2\,,$$

where $l(\vec{x})$ is a path from a fixed arbitrary point on the boundary to $\vec{x}$. Since $\nabla\cdot\vec{U} = 0$, the computed value is independent of the path $l(\vec{x})$. We can compute ψ everywhere in Ω by integrating the above expression on the edges of the finite element triangulation. This is a much faster method, the numerical cost being proportional to the number of nodes in the finite element discretisation.

The computations were performed on a grid with 32×120 nodes P_2. For the time steps we have not encountered stability problems, but too large steps result in a divergence of the Newton method in (18). Figure 6 shows the number of conjugate gradient iterations for the fictitious domain method along the whole computation of a polymerization process (with $\alpha = 10$ degrees). We can see that this number is moderate and does not depend too strongly of the size of Ω_1. In all the simulations, the choice of the penalisation parameter ε, as predicted by the theoretical study of the algorithm, was not critical for the convergence: choices between $\varepsilon = 10^{-5}$ and $\varepsilon = 10^{-20}$ do not affect the number of iterations. The following table gives the number of iterations for the solution of a problem in $\Omega = (0,1) \times (0,10)$ as a function of the number of points in the grid, (horizontally – number of discretisation points along the horizontal axis (0,1), and vertically – number of points along the vertical axis (0,10); the fixed domain is $(0,0.5) \times (0,10)$):

	10	20	40
50	5	7	14
100	4	5	10
200	3	4	5

We see that except for the case when the aspect ratio of the elements is very poor (that is to say the quotient h_x/h_y of the sizes of the elements along the 2 coordinates axis is far from 1), the number of iterations does not vary much.

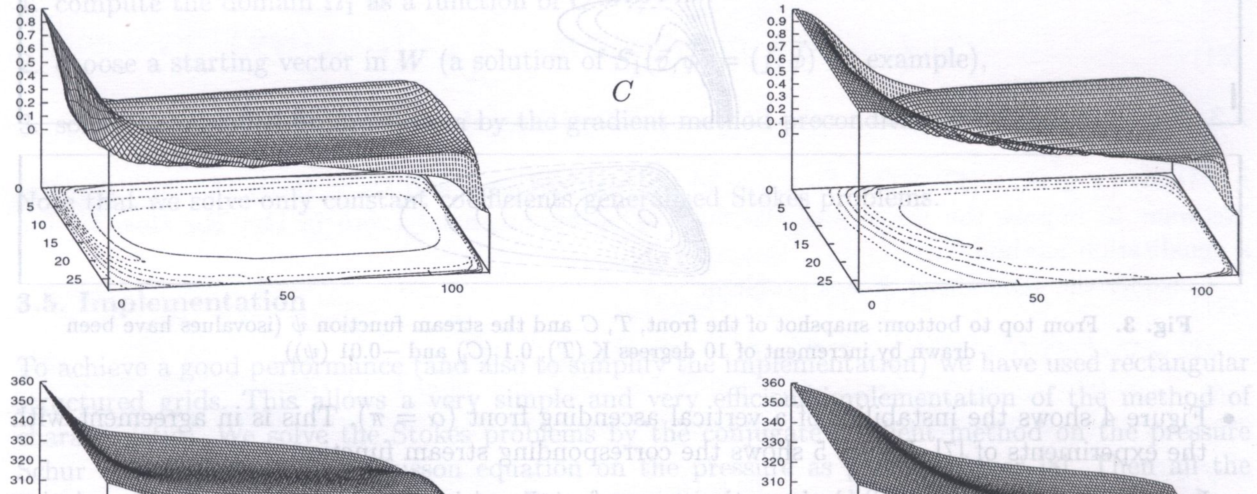

Fig. 4. Snapshots of C (first line), T (second line) and position of the front for different values of t *(continued in the next page)*

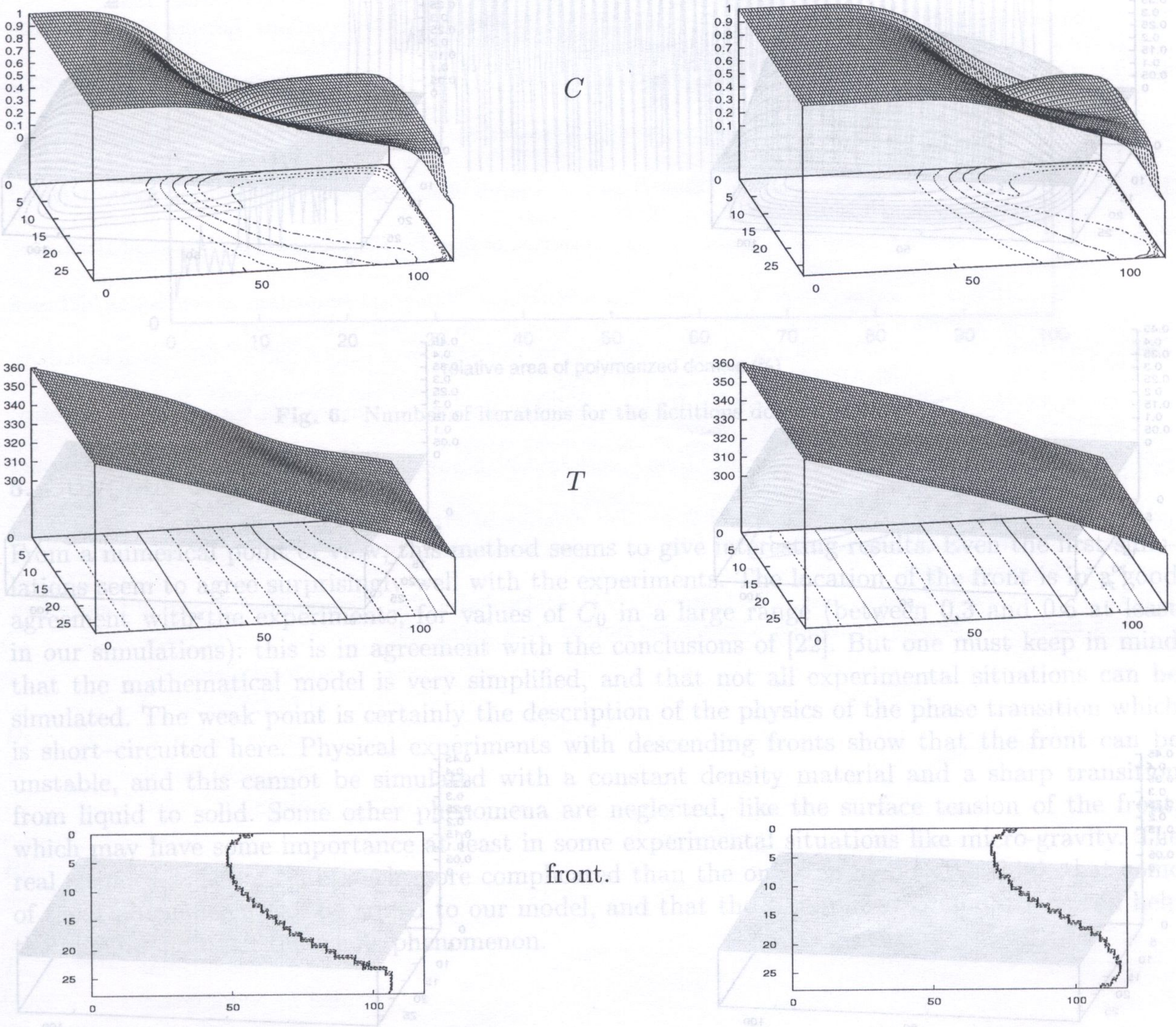

Fig. 4. *(continued)* Snapshots of C (first line), T (second line) and position of the front for different values of t

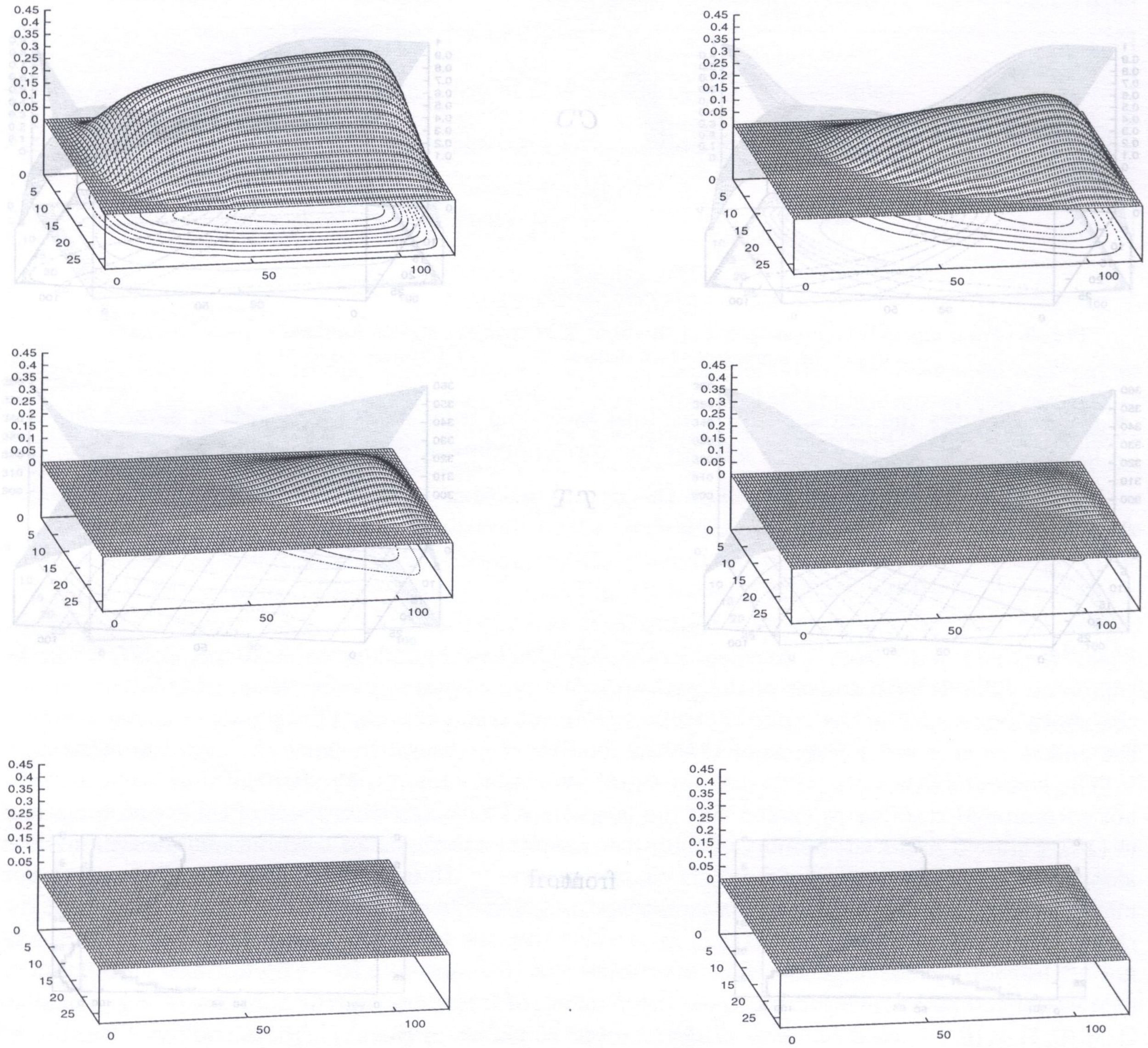

Fig. 5. Stream function

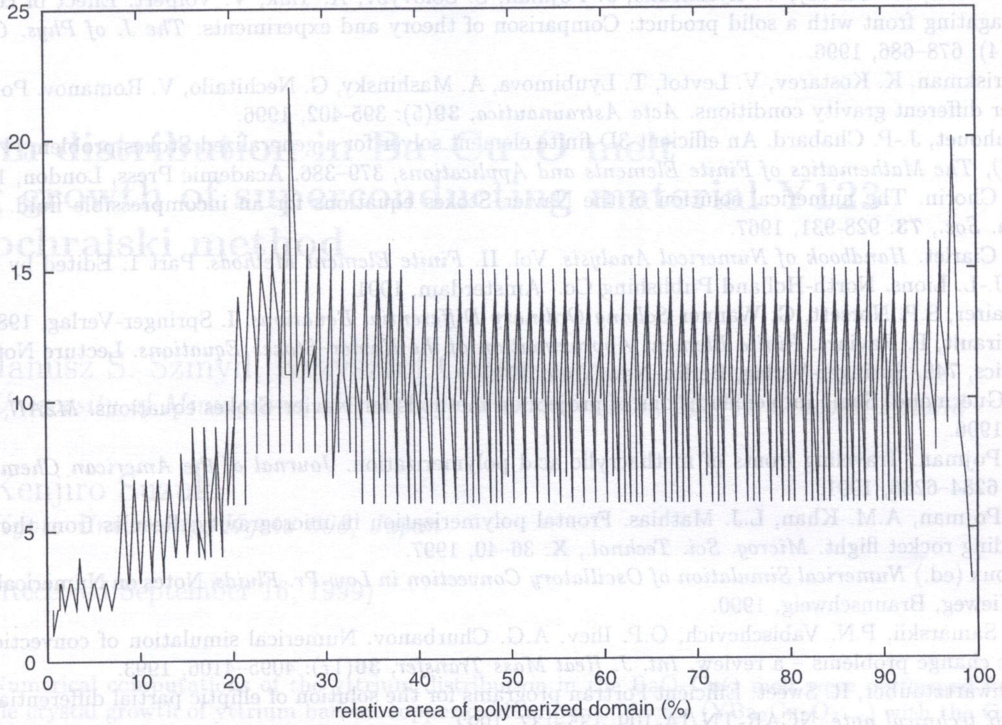

Fig. 6. Number of iterations for the fictitious domain method

5. Conclusion

From a numerical point of view, this method seems to give interesting results. Even the first simulations seem to agree surprisingly well with the experiments. The location of the front is in a good agreement with the experiments, for values of C_0 in a large range (between 0.3 and 0.6 at least in our simulations): this is in agreement with the conclusions of [22]. But one must keep in mind that the mathematical model is very simplified, and that not all experimental situations can be simulated. The weak point is certainly the description of the physics of the phase transition which is short–circuited here. Physical experiments with descending fronts show that the front can be unstable, and this cannot be simulated with a constant density material and a sharp transition from liquid to solid. Some other phenomena are neglected, like the surface tension of the front, which may have some importance at least in some experimental situations like micro-gravity. The real chemical reactions are much more complicated than the one step model. We think that some of these phenomena may be added to our model, and that these simplified computations can help the understanding of the whole phenomenon.

References

[1] S.N. Bakhvalov. Solution of the Stokes non stationary problems by the fictitious domain method. *Russ. J. Numer. Anal. Math. Modeling,* **10**(3): 163–172, 1995.

[2] M. Bazile, T. Dumont, H.A. Nichols, J.A. Pojman, V. Volpert. The effect of orientation on a propagating front with a solid product. Submitted to *Chaos*, 1999.

[3] M. Bercovier, O. Pironneau, V. Sastri. Finite elements and characteristics for some parabolic-hyperbolic problems. *Appl. Math. Modelling,* **7**(2): 89–96, 1983.

[4] K. Boukir. *Méthodes en temps d'ordre élevé par décomposition d'opérateurs. Application aux équations de Navier–Stokes.* Thesis, Université Paris 6, 1993.

[5] K. Boukir, Y. Maday, B. Métivet, E. Razafindrakoto. High-order characteristics/finite element method for the incompressible Navier–Stokes equations. *Internat. J. Numer. Methods Fluids.*, **25**(12): 1421–1454, 1997.

[6] G. Bowden, M. Garbey, V. Ilyashenko, J. Pojman, S. Solovyov, A. Taik, V. Volpert. Effect of convection on propagating front with a solid product: Comparison of theory and experiments. *The J. of Phys. Chemistry B*, **101**(4): 678–686, 1996.

[7] V. Briskman, K. Kostarev, V. Levtof, T. Lyubimova, A. Mashinsky, G. Nechitailo, V. Romanov. Polymerization under different gravity conditions. *Acta Astraunautica,* **39**(5): 395-402, 1996.

[8] J. Cahouet, J.-P. Chabard. An efficient 3D finite element solver for a generalized Stokes problem, *VI (Uxbridge, 1987), The Mathematics of Finite Elements and Applications,* 379–386. Academic Press, London, 1987.

[9] A.J. Chorin. The numerical solution of the Navier–Stokes equations for an incompressible fluid. *Bull. Amer. Math. Soc.*, **73**: 928-931, 1967.

[10] P.G. Ciarlet. *Handbook of Numerical Analysis.* Vol. II. *Finite Element Methods.* Part 1. Edited by P.G. Ciarlet and J.-L. Lions. North-Holland Publishing Co., Amsterdam, 1991.

[11] E. Hairer, S.P. Norsett, G. Wanner *Solving Ordinary Differential Equations*, I. Springer-Verlag, 1987.

[12] V. Girault, P. Raviart. *Finite Element Approximation of the Navier–Stokes Equations.* Lecture Notes in Mathematics, 749. Springer-Verlag, Berlin–New York, 1979.

[13] J.L. Guermond. Some implementations of projection methods for Navier–Stokes equations. *M2AN*, **30**(5): 637–667, 1996.

[14] J.A. Pojman. Traveling fronts of methacrylic acid polymerization. *Journal of the American Chemical Society.* **113**, 6284–6286, 1991.

[15] J.A. Pojman, A.M. Khan, L.J. Mathias. Frontal polymerization in microgravity: Results from the Conquest I sounding rocket flight. *Microg. Sci. Technol.*, **X**: 36–40, 1997.

[16] B. Roux (ed.) *Numerical Simulation of Oscillatory Convection in Low-Pr. Fluids.* Notes on Numerical Mechanics, 17. Vieweg, Braunschweig, 1990.

[17] A.A. Samarskii, P.N. Vabischevich, O.P. Iliev, A.G. Churbanov. Numerical simulation of convection/diffusion phase change problems – a review. *Int. J. Heat Mass Transfer*, **36**(17): 4095–4106, 1993.

[18] P. Schwartztauber, R. Sweet. Efficient Fortran programs for the solution of elliptic partial differential equations. *NCAR technical note*, NCAR-TN/IA-109, 135–137, 1992.

[19] S.E. Solovyov, V.A. Volpert, S. Davityan. Radially symetric flow of reacting liquids with changing viscosity. *SIAM J. Appl. Math.*, **50**(4): 907–914, 1993.

[20] R. Temam. *Numerical Analysis.* Translated from French by J.W. Nienhuys. D. Reidel Publishing Co., Dordrecht–Boston, 1973.

[21] N.N. Yanenko. *The Method of Fractional Steps. The Solution of Problems of Mathematical Physics in Several Variables.* Translated from Russian by T. Cheron. English translation edited by M. Holt. Springer-Verlag, New York–Heidelberg, 1973.

[22] Ya.B. Zeldovich, G.I. Barenblatt, V.B. Librovich, G.M. Makhviladze, *The Mathematical Theory of Combustion and Explosions.* English translation. Consultants Bureau, New York, 1985.