Journal of Computational Physics 228 (2009) 8249–8262

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A parallel computer implementation of the Asymptotic Numerical Method to study thermal convection instabilities

Marc Medale [a,*], Bruno Cochelin [b]

[a] IUSTI, UMR 6595 CNRS-Université Aix-Marseille, 5 rue Enrico Fermi, Technopole de Château-Gombert, 13453 Marseille Cedex 13, France
[b] LMA, UPR 7051 CNRS, 38 rue Frédéric Joliot Curie, Technopole de Château-Gombert, 13451 Marseille Cedex 20, France

ARTICLE INFO

Article history:
Received 17 April 2009
Received in revised form 24 July 2009
Accepted 29 July 2009
Available online 6 August 2009

PACS:
02.70.Dh
47.10.Ad
47.20.Ky
47.11.−j
47.20.Bp
47.20.Dr

Keywords:
Path-following or continuation methods
Direct solver for large scale algebraic systems
Parallel computer implementation
Rayleigh–Bénard–Marangoni instability

ABSTRACT

We have developed a numerical model to efficiently compute steady-state combined buoyancy and thermocapillary convection solutions. It features a parallel computer implementation of an Asymptotic Numerical Method to perform steady-state path-following and locate bifurcation points in problems involving large size algebraic systems, up to few million degrees of freedom. The model has been first validated on a problem for which a reference solution exists and then is used to analyse the influence of the container size and shape on Rayleigh–Bénard–Marangoni convection and its related cellular pattern.

1. Introduction

Thermal convection acts around us at various places and scales, but also in very different ways as in geophysical cases (atmosphere and ocean movements, earth mantle convection, etc.) or in many industrial processes where fluids are involved. However, the particular case we are interested in occurs when several non-miscible fluids are present so thermal gradient may induce natural convection driven by both buoyancy in the bulk and surface-tension along the interface [43]. In this framework our goal is to develop a computational tool that can help us to better understand how to either enhance or decrease natural convection, depending on the application. Moreover, as far as instabilities are concerned, the numerical model should also be able to determine the range of the control parameter for which the solution changes radically, i.e. stable or unstable, steady or unsteady.

The tool we need sets in the non-linear computation field [49], for which path-following algorithms also designated continuation methods aims at computing branches of steady state solutions for a range of the control parameter. Most of them

* Corresponding author. Tel.: +33 491 106 914; fax: +33 491 106 969.
E-mail address: Marc.Medale@polytech.univ-mrs.fr (M. Medale).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.032

are of predictor-corrector type [2] and consist in starting from a known point on the solution curve, seeking the next one at a certain distance apart of it in a predictor step, then as this guessed point is generally no longer on the curve, iteratively correcting the initial prediction in a corrector step. However, along with these predictor-corrector steps a parameterization strategy [39] and a step-length control [35] should complete the continuation method to trace pointwise the whole branch while looping over the steps. Finally, a robust algorithm should be able to adaptively choose the step-length according to the user-required accuracy and also to detect bifurcations [29]. The basic tasks involved in the latter issue generally consist in three stages: firstly detect the bifurcation, then calculate the bifurcation point accurately, and finally switch possibly branches if another branch intersects the previous one [49].

In the path-following process the major part of the computational time is dedicated to either the predictor step (for high-order predictor) or to the corrector one when the predicted solution is far from the solution branch. In popular path-following algorithms the predictor step is carried out by a tangent (or first order Euler) approximation, whereas the corrector step is performed by Newton's method or one of its variants [35,49]. On the other hand, high-order predictors have more recently been used in continuation algorithms based on the Asymptotic Numerical Method (ANM), which consists in a combination of a high-order perturbation method, a discretization technique and a parameterization strategy [24,6,21,19]. This method is a general and efficient non-linear solution technique [22], which has been successfully applied in solid and structural mechanics [20,55,33], but few attempts have also been made in hydrodynamics problems [16,1,14].

Nevertheless, in all the previously quoted ANM applications the discretized problem resulted in small-to-moderate size algebraic systems (few tens of thousands of degrees of freedom). However, in many Computational Fluid Dynamics problems and more particularly in fluid flow or thermal instability problems, spatial resolution is of first concern to accurately mimic the physics leading to large size algebraic systems to be solved. So, for this class of problems the computational efficiency of large scale non-linear solver becomes a very challenging issue. Therefore, in the present paper we present the numerical model we have developed to efficiently compute steady-state thermocapillary convection solutions and locate bifurcation points between them. It features a parallel computer implementation of the ANM able to deal with thermal instability problems involving algebraic system up to few millions of degrees of freedom. The numerical model has been used to analyse the influence of the container size and shape on the onset of convection and its associated cellular pattern. For that purpose we have considered regular polygon base containers, from conductive (linear) to strongly supercritical (highly non-linear convection) steady states.

The paper is organaized as follows. The Rayleigh–Bénard–Marangoni instability is introduced in Section 2 and the governing equations resulting from the one layer model are recalled. Then the numerical model is described in Section 3 and detailed validations and comparisons to a classical predictor-corrector algorithm are provided in Section 4. Then, the ANM algorithm has been intensively used to study the influence of container shape and size on the RBM instability problem. These results in polygonal containers are reported in Section 5, they demonstrate the model capabilities in confined thermal-convection instabilities. Finally, some conclusions are drawn.

2. The Rayleigh–Bénard–Marangoni instability

The physical problem under consideration concerns a horizontal liquid layer heated from below and cooled from above across a thin air layer, as sketched in Fig. 1(a). Beyond a critical vertical temperature difference, surface-tension-gradient and buoyancy are both responsible for convection to occur, the so-called Rayleigh–Bénard–Marangoni convection or instability. However, in the shallow layer case the first mechanism is predominant and leads to the surface-tension-driven convection, referred to as Bénard–Marangoni convection. The main feature of the latter is, for a fluid layer of large horizontal extension, just above the threshold the fluid flow is well organized into a cellular hexagonal pattern, as shown in Fig. 1(b), first observed by Bénard [10] more than a century ago.

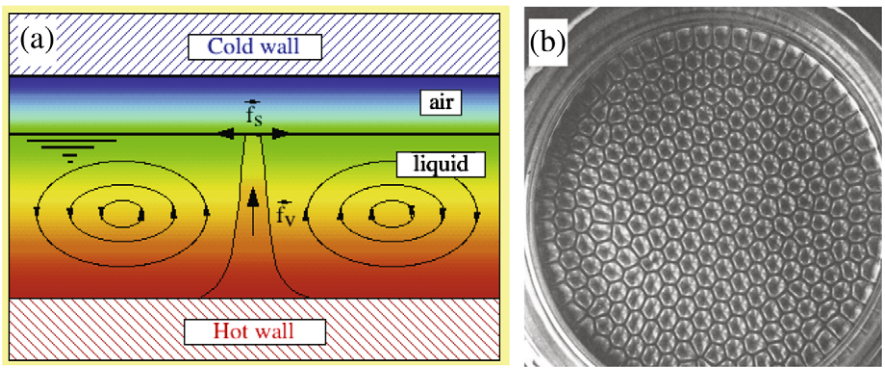


Fig. 1. Rayleigh–Bénard–Marangoni convection. (a) Schematic cross section of problem setup; (b) Top view of the liquid layer exhibiting the famous hexagonal multi cellular pattern, from [37].

2.1. The one layer model

Real world problems are often out of reach from deep scientific insight, so judiciously simplified model problems can help to approximate them or catch their intrinsic features [30,34,41]. For the thermocapillary convection problem considered here, physical simplificating assumptions lead to the one layer model [51,43]. This model takes advantage of that surface-tension only dominates buoyancy in ground experiments if the fluid layer thickness is of order of a millimetre or less. So in this case the liquid–air interface is assumed to be flat, horizontal and of constant depth. Furthermore, the air layer located just above the interface has a thickness of the same order than the liquid layer one, so the heat transfer across that thin air layer remains dominated by conduction. Finally the liquid layer is bounded from below by an isothermal hot wall and surrounded by adiabatic lateral ones. Taking into account all these assumptions results in a simplified model restricted to the liquid layer, referred to as the one layer model, which has been successfully compared to experiments in the above conditions [32,42].

In this setting, depending mainly on both the distance from the threshold, thermal boundary conditions and liquid Prandtl number, numerous transitions could occur from hexagonal to other polygonal patterns (pentagonal, square, or mixed). Many of such steady state solutions have been observed or predicted by several authors [45,11,32,48]. Moreover, the container size and shape strongly affect most of the flow characteristics, which are very different from the infinite layer case owing to the imposed geometrical restraints. Convincing experiments illustrated this result in hexagonal vessels [18], circular and square ones [38], encouraging others to reproduce them. Computations were first performed in square and rectangular enclosures [28,47,25], for Prandtl number of unity in the former work and 10^4 in the latter. Later on, Biswal and Rao [12] complemented the existing results in small rectangular vessels and also considered prescribed heat flux at the lower boundary. Moving on to circular configurations, Zaman and Narayanan [54] and Dauby et al. [26] predicted the convective pattern at the threshold. Dynamical regimes (periodic, chaotic, etc.) have also been studied but are beyond the scope of the present paper, so the interested reader is invited to refer to dedicated literature [36,13,23,43].

2.2. Governing equations

According to the one layer model, the problem is governed by the coupled incompressible Navier–Stokes and energy equations in the liquid layer, complemented with appropriate boundary conditions at the walls and free surface. Introducing dimensionless variables of space, velocity ($\vec{V} = \{u, v, w\}$), pressure (P) and temperature (θ) into the conservation equations of mass, momentum and energy results in the set of steady state equations in the open domain denoted Ω, restricted to the liquid layer:

$$\vec{\nabla} \cdot \vec{V} = 0 \tag{1}$$

$$\frac{1}{Pr}(\vec{V} \cdot \nabla \vec{V}) = -\vec{\nabla} P + \Delta \vec{V} + Ra\theta \vec{e}_z \tag{2}$$

$$\vec{V} \cdot \vec{\nabla} \theta = \Delta \theta \tag{3}$$

On the one hand the fluid flow boundary conditions are the classical no-slip condition at solid walls (Eq. (4)) and the Marangoni condition at the free surface (Eq. (5)):

$$\vec{V} = \vec{0} \quad \text{on } \Sigma_V \tag{4}$$

$$\vec{V} \cdot \vec{n} = 0; \quad \frac{\partial u}{\partial z} = Ma \frac{\partial \theta}{\partial x} \quad \text{and} \quad \frac{\partial v}{\partial z} = Ma \frac{\partial \theta}{\partial y} \quad \text{on } \Sigma \tag{5}$$

On the other hand the associated heat transfer boundary conditions are a prescribed uniform temperature at the lower heated wall (Eq. (6)), an adiabatic condition at all the lateral walls (Eq. (7)) and finally a heat exchange condition at the free surface, accounting for the conductive heat transfer across the modelled air layer (Eq. (8)):

$$\theta = \bar{\theta} \quad \text{on } \Sigma_\theta \tag{6}$$

$$\frac{\partial \theta}{\partial n} = 0 \quad \text{on } \Sigma_{lat} \tag{7}$$

$$\frac{\partial \theta}{\partial n} = -Bi\,\theta \quad \text{on } \Sigma \tag{8}$$

The related physical and geometrical parameters have been gathered into four dimensionless numbers, namely the Biot number $\left(Bi = \frac{d_l}{k_l}\frac{k_a}{d_a}\right)$, the Marangoni number $\left(Ma = \frac{\gamma d_l \Delta T}{\mu \kappa}\right)$, the Rayleigh number $\left(Ra = \frac{\beta g d_l^3 \Delta T}{\nu \kappa}\right)$ and the Prandtl number $\left(Pr = \frac{\nu}{\kappa}\right)$. They involve eleven physical quantities, $d_l, d_a, k_l, k_a, g, \beta, \gamma, \kappa, \nu, \mu, \Delta T$, representing liquid and air layer thicknesses, liquid and air thermal conductivity coefficients, coefficient of gravitational acceleration, temperature coefficients of density and surface-tension, thermal diffusivity, liquid kinematic and dynamic viscosities, average temperature difference across the liquid layer, respectively.

Moreover, in finite size containers the aspect ratio (denoted Γ) is the geometrical parameter relevant to the liquid layer confinement and should be related to the container shape to entirely describe the problem geometry. It is defined as the ratio of a characteristic horizontal length to the liquid layer depth ($\Gamma = \frac{\sqrt{A}}{d_l}$, where A designates the free surface area).

3. Numerical model

The numerical model we have developed has been devised to efficiently compute steady-state Rayleigh–Bénard–Marangoni solutions and locate bifurcations between them. It has been carefully designed to perform three dimensional thermal instability problem computations in reasonable elapsed time and rational computational efficiency. To achieve that goal, it is of first concern to fulfill compatibility conditions between the problem formulation, the selected solution algorithms or strategies and finally the computer architecture and programming environment. So, the three main components of our numerical model are addressed now, namely the finite element formulation, the continuation algorithm based on the Asymptotic Numerical Method and finally the parallel computer implementation.

3.1. Finite element formulation

According to the standard Weighted Residual Method [27,17], a mixed formulation can be built to cope with the fully coupled steady-state problem governed by Eqs. (1)–(8). Its associated weak integral form reads:

$$\begin{aligned}
&\int_\Omega \delta p\left(\nabla\cdot\vec{V}-\frac{p}{\lambda}\right)d\upsilon = 0\\
&\int_\Omega \left(\frac{1}{Pr}(\vec{V}\cdot\nabla\vec{V})\delta\vec{V}+\nabla\vec{V}:\nabla\delta\vec{V}\right)d\upsilon - \int_\Omega p\nabla\cdot\delta\vec{V}\,d\upsilon - \int_\Omega Ra\,\theta\,\delta\vec{V}\cdot\vec{e}_z d\upsilon + \int_\Sigma Ma\,\delta\vec{V}\cdot\nabla_s\theta\,ds = 0\\
&\int_\Omega (\delta\theta\vec{V}\cdot\nabla\theta+\nabla\theta\cdot\nabla\delta\theta) + \int_\Sigma Bi\,\theta\,\delta\theta ds = 0
\end{aligned} \tag{9}$$

where $\delta\vec{V}, \delta p$ and $\delta\theta$ are the weighting functions associated with the velocity vector, the pressure and the temperature fields, respectively.

Following the standard Bubnov–Galerkin finite element method, the spatial discretization of these integral forms is performed with a 27-node iso-parametrical hexahedral finite element for the domain integrals (Ω) associated with velocity and temperature unknowns, whereas the boundary integrals over the free surface (Σ) are discretized in space with a 9-node iso-parametrical quadrilateral finite element. These approximations ensure a piecewise quadratic C_0 approximation for the velocity and temperature fields. Moreover, we have used a piece-wise linear but discontinuous approximation of the pressure field that enables us to eliminate the discretized pressure unknowns at the element level in a static condensation stage, thanks to a penalisation parameter λ taken to 10^9 in the present computations. This finite element discretization satisfies the compatibility condition between the velocity and pressure approximation fields for incompressible flows ($inf-sup$ or $L.B.B.$ condition).

3.2. The Asymptotic Numerical Method

The continuation procedure we have implemented is based on an Asymptotic Numerical Method [22]. It consists in a high order predictor built on a perturbation method, associated with a pseudo arc-length parameterization and a corrector of Newton–Raphson type. The latter is nevertheless often useless when one makes use of short step lengths or high order expansion series since in this case the predictor gives an accurate representation of the branch inside the range of utility of the perturbation series.

3.2.1. Perturbation method

Asymptotic expansions have been used to solve a broad variety of regular perturbation problems in fluid mechanics, whether or not viscosity and compressibility are included [53]. Nevertheless, in the early years preceding a widespread computer usage only few series terms, usually less than ten, were calculated by hand after a life of labor. Since researchers delegated this fastidious task to computers, series of arbitrary high order have been substituted to the low order ones [52] contributing to enhanced accuracy.

The high order predictor of our path-following algorithm is obtained by a regular perturbation technique. Starting from a known and regular solution on the curve $(U_0, \Delta T_0)$, where $\vec{U}$ gathers velocity, pressure and temperature unknowns ($\vec{U} = \{\vec{V}, P, \theta\}$), then expanding up to a prescribed truncation order n both the unknown vector and the scalar control parameter ΔT (driving the Marangoni and Rayleigh numbers) with respect to a path parameter s, it results the following power series expansion:

$$\begin{aligned}
\vec{U}(s) &= \vec{U}_0 + \vec{U}_1 s + \vec{U}_2 s^2 + \vec{U}_3 s^3 + \cdots + \vec{U}_n s^n\\
\Delta T(s) &= \Delta T_0 + \Delta T_1 s + \Delta T_2 s^2 + \Delta T_3 s^3 + \cdots + \Delta T_n s^n
\end{aligned} \tag{10}$$

Then, the parameterization equation aims at providing an extra relationship for the path parameter in order to keep a well-posed problem. Various choices are possible, but in the present work it is the pseudo arc-length relationship of [35] that has been implemented:

$$s = \langle \vec{U} - \vec{U}_0, \vec{U}_1 \rangle + (\Delta T - \Delta T_0)\Delta T_1 \tag{11}$$

where $\langle \vec{U}, \vec{V} \rangle$ and $(\vec{U}_1, \Delta T_1)$ denote the scalar product and tangent vector associated with the augmented system, respectively.

Let us now insert the polynomial representation Eq. (10) into both the weak integral form Eq. (9) and the parameterization Eq. (11), equating then like powers of s, it results a set of recurrent linear equations for unknowns $\left(\vec{U}_i, \Delta T_i\right)$ at every order i $(1 \leqslant i \leqslant n)$ of the asymptotic expansion:

$$\begin{array}{ll} \text{Order 1} & \text{Order } p,\ 2 \leqslant p \leqslant n \\ \left\{ \begin{array}{l} L_t(\vec{U}_1) = 0 \\ \vec{U}_1 = \Delta T_1 \vec{U}_D \quad \text{on} \quad \partial\Omega_D \\ \langle \vec{U}_1, \vec{U}_1 \rangle + \Delta T_1^2 = 1 \end{array} \right. & \left\{ \begin{array}{l} L_t(\vec{U}_p) = -\sum_{q=1}^{p-1} Q(\vec{U}_q, \vec{U}_{p-q}) \\ \vec{U}_p = \Delta T_p \vec{U}_D \quad \text{on} \quad \partial\Omega_D \\ \langle \vec{U}_p, \vec{U}_1 \rangle + \Delta T_p \Delta T_1 = 0 \end{array} \right. \end{array} \tag{12}$$

where L_t is the tangent operator obtained by taking the first variation of the weak integral form Eq. (9), Q is the quadratic operator associated with the convective terms $(\delta \vec{U}(\vec{V} \cdot \nabla \vec{U})$ in Eq. (9)), and $\vec{U}_D$ is the vector restricted to non-zero Dirichlet boundary conditions (imposed on $\partial\Omega_D = \Sigma_\theta$).

3.2.2. Finite element discretization

Let us first introduce into Eq. (12) the spatial discretization described in Section 3.1, then transpose and adapt to three-dimensional fully coupled velocity–pressure–temperature 27-node-hexahedral finite elements the discretization and pressure condensation procedures presented in [16,1] for isothermal and incompressible Navier–Stokes fluid flow problems, one obtains the set of discrete linear problems to be solved at every order i $(1 \leqslant i \leqslant n)$:

$$\begin{array}{ll} \text{Order 1} & \text{Order } p,\ 2 \leqslant p \leqslant n \\ \left\{ \begin{array}{l} [K_t(\vec{U}_0)]\{\vec{U}_1\} = \Delta T_1 \{\vec{F}\} \\ \langle \vec{U}_1 \rangle \{\vec{U}_1\} + \Delta T_1^2 = 1 \end{array} \right. & \left\{ \begin{array}{l} [K_t(\vec{U}_0)]\{\vec{U}_p\} = \Delta T_p \{\vec{F}\} - \sum_{q=1}^{p-1} \{Q(\vec{U}_q, \vec{U}_{p-q})\} \\ \langle \vec{U}_p \rangle \{\vec{U}_1\} + \Delta T_p \Delta T_1 = 0 \end{array} \right. \end{array} \tag{13}$$

where $[K_t(\vec{U}_0)]$ is the discrete tangent matrix, $\{\vec{U}_i\}$ is the discrete unknown vector to solve for at every order to get the power series expansion Eq. (10), and finally $\{F\}$ is the discrete vector resulting from the conversion of the applied Dirichlet boundary conditions into corresponding discretized loads.

Therefore one has to solve a set of linear algebraic systems that have the same tangent matrix but different right hand sides. So, the tangent matrix can be factored only once per predictor step if a direct solver is used, as in the present case.

3.2.3. Continuation technique

The solution of the linear algebraic systems Eq. (13) at every order provides the power series expansion Eq. (10) in the vicinity of the known starting solution on the curve $(U_0, \Delta T_0)$. However, as this polynomial representation has a limited range of utility related to the radius of convergence of the power series, one has to define the maximum path parameter value corresponding to an acceptable departure from the solution curve within the ANM predictor step. For that purpose we make use of a reliable criterion proposed by [19] to evaluate the maximum path parameter value s_{max}. It is built on the ratio between the first term of the series to the last one, times a user prescribed constant δ:

$$s_{max} = \left(\delta \frac{\| \vec{U}_1 \|}{\| \vec{U}_n \|} \right)^{\frac{1}{n-1}} \tag{14}$$

Moreover, as the ANM predictor step could produce a guess solution slightly departing from the solution curve, we have implemented the classical Newton–Raphson algorithm as a corrector step. However, in the present work we choose δ values in Eq. (14) such that the corrector step is most of the time unused.

Finally, the ANM path following method describes one part of the solution curve within each step. Therefore, the whole curve solution is obtained step-by-step, taking either $(\vec{U}(s_{max}), \Delta T(s_{max}))$ as the starting point of the next portion of the curve if no corrector step is required [19] or the solution from the corrector step if any, then looping over this sequence until the user-defined control parameter range has been spanned.

This ANM path-following algorithm represents a standard basis from which substantial improvements could be included in both the ANM predictor step or the corrector one. For example, substituting the power series by rational fractions (Padé approximants) in the ANM predictor increases the step-length for a comparable accuracy [21,33]. Moving on to the corrector step, homotopy and projection techniques can reduce significantly its cost [40,15]. Although we are aware of the numerous improvements that can be done from the present basic ANM algorithm, in the present paper we have chosen to emphasize our contribution on the solution of large size problems, up to few millions of unknowns. Therefore, to address such issue efficient computer implementation becomes of first concern.

3.3. Parallel computer implementation

Not only to have an advantageous trade-off between computational performances and ease to develop, but also to get some long term operating on up to date supercomputers, our finite element path-following code has been developed in the PETSc framework [7–9]. As a major benefit this Portable Extensible Toolkit for Scientific Computations library has been carefully designed to provide users with parallel objects of high level of abstraction, along with numerous external package interfaces among which efficient direct parallel solvers.

At every predictor step of our ANM path-following algorithm the set of linearised algebraic systems is solved repeatedly at every order of the power series expansion with the Multifrontal Massively Parallel sparse direct Solver [3–5]. Indeed, we first tried to solve these algebraic systems with parallel iterative algorithms (GMRES, BCGS, etc.) associated with several preconditioners (SSOR, ASM, etc.) provided in the PETSc library or more specialised ones (EUCLID and PILUT) from the HYPRE library [31]. But we never succeeded to get residuals as low as roughly few thousand times that obtained with the MUMPS parallel direct solver for comparable CPU times. The reason is related to the fact that the larger the condition number of an algebraic system (highest to smallest eigenvalue ratio) the smaller the convergence rate of an iterative solver. Unfortunately, in the present algebraic systems arising from the discretization of the fully coupled and penalized mixed weak integral form Eq. (9), the condition number is proportional to the penalization coefficient, whose huge value ($\lambda = 10^9$) enables us to carefully satisfy the incompressibility constraint in the fluid flow. So, in the present extremely ill-conditioned algebraic systems the MUMPS parallel direct solver provides much more accurate results than widespread iterative ones at comparable CPU costs, provided one can afford a huge amount of required computer memory. Moreover, a very welcome feature to compute the power series expansion at every ANM predictor step is once the factorisation step has been performed multiple solves can be achieved with different right hand sides for no significant extra cost.

Finally this high-level library-oriented implementation enables us to efficiently solve algebraic systems up to few millions of unknowns with a direct parallel solver running high performance computers and taking advantage of massively parallel processing.

4. Validations and computational capabilities

Substantial validations have been conducted, especially in the RBM convection field in which theoretical, experimental and numerical results are available for assessment. Quantitative comparisons have been performed in the infinite layer configuration for which Pearson [46] determined the threshold value and critical wavenumber in the surface-tension-driven convection case, Sparrow et al. [50] determined these critical quantities when only buoyancy acts and finally Nield [44] gathered the two previous cases considering combined surface-tension-driven and buoyancy convection. On the other hand confined configurations are more appropriate to laboratory experiments, so comparisons can also be made in those cases provided well controlled conditions have been used, e.g. [38].

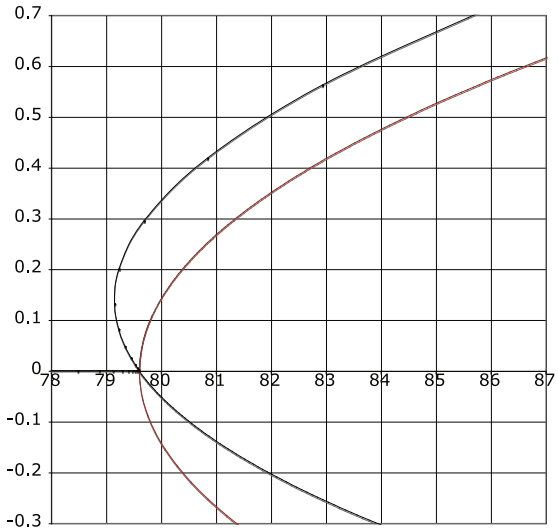
4.1. Validation on surface-tension-driven convection

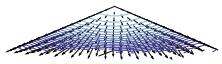
Let us consider the idealized case of an infinite Prandlt number liquid layer of infinite horizontal extension subjected to pure surface-tension-driven convection. In this particular case Pearson [46] first determined the onset of convection (critical Marangoni number $Ma_{c0} \approx 80$) and critical wave number ($k_c \approx 2$) in the limit of vanishing Biot number ($Bi = 0$). Few years later Nield [44] provided slightly more accurate results than Pearson's ones for this case ($Ma_{c0} = 79.607$ and critical wave number $k_c = 1.993$).

So, based on this theoretical background we have chosen this configuration as the first test problem to validate our numerical model. However to perform computations at a reasonable cost in this infinite layer configuration, we have set the computational domain to match one wavelength in each of the horizontal directions ($lx/d_l = 4\pi/k_c$, $ly/d_l = 4\pi/k_c\sqrt{3}, lz/d_l = 1$, with $k_c = 1.9929$) applying accordingly symmetry boundary conditions over the vertical limit planes of the computational domain.

As a natural choice in the present thermal convection problem we have set the continuation parameter as the mean temperature drop across the liquid layer (ΔT), which therefore directly transforms into the dimensionless Marangoni number (Ma). Then, we have performed continuation computations with our ANM algorithm from a purely conductive (motionless) base state associated with the following set of control parameters $Ma = 70$ and $Bi = 0$, with an asymptotic expansion limited to thirty terms ($n = 30$ in Eq. (10)) and a tolerance criteria $\delta = 10^{-12}$ in Eq. (14).

As one of the interesting outputs of the continuation computations we have plotted in Fig. 2(a) the magnitude of the signed mean fluid flow velocity over the computational domain versus the Marangoni number for mesh $M2$ defined in Table 1. This signed mean velocity is defined as: $V_{mean} = \frac{sign(w(O))}{Vol}\left(\int_{Vol}(u^2 + v^2 + w^2)dv\right)^{1/2}$, where $sign(w(O))$ designates the sign of the vertical velocity component at the center of the computational domain. Three branches appears in the plot of Fig. 2(a), they intersect at the threshold point designated A. Each branch is associated with one specific pattern: B_1 to the motionless purely conductive base state (linear thermal field in the vertical direction), B_2 to the perfectly hexagonal pattern (represented in Fig. 2(b)) and B_3 to a roll pattern (represented in Fig. 2(c)), respectively. The part of the B_2 branch which has a positive mean velocity corresponds to an upward velocity at the center of the hexagonal cells (Fig. 2(b)), meanwhile the





velocity is reversed in the cells along the negative mean velocity part of the plot. A comparable analysis stands also for the roll pattern associated to the B_3 branch.

In this infinite layer configuration, for high or infinite Prandtl number liquid and vanishing Biot number, the bifurcation from the base state to the hexagonal pattern is transcritical at the onset of convection (threshold point A in Fig. 2(a)). Moreover it is subcritical since it exists a turning point (point B in Fig. 2(a)) at a lower value than the onset of convection. On the other hand the bifurcation from the base state to the roll pattern is a pitchfork at threshold. Both type of bifurcations are perfectly recovered by our ANM computations.

Moreover a quantitative validation has been also performed comparing the computed threshold value on three progressively refined meshes to the theoretical value established by Nield [44]. The details of the meshes considered for the spatial convergence analysis are reported in Table 1, meanwhile the computed threshold values on these three meshes are reported in Table 2, together with their relative distance to the reference theoretical value. It is noteworthy that even on the coarse mesh $(M1)$ the critical Marangoni number is quite accurately computed, meanwhile on the finest one $(M3)$, which is four times finer than the former in every directions, one gets a two orders of magnitude more accurate solution than on the coarser one. Their relative departure from the theoretical reference value are less than $5 \times 10^{-4}\%$ and $3 \times 10^{-6}\%$, respectively, see Table 2.

Furthermore this spatial convergence study enables us to assess the order of convergence of our numerical model and it can also provide us with an extrapoled value for an hypothetical infinitely refined mesh using a Richardson extrapolation. This extrapolated critical Marangoni number is also provided in Table 2 along with its relative distance to the theoretical value, the former being evaluated in the following way. Let us denote $Ma_c(h_i)$ the critical Marangoni number computed

on mesh i, whose mesh size is h_i. Their Richardson extrapolation for an infinitely refined mesh reads for three progressively refined meshes such that $\frac{h_1}{h_2} = \frac{h_2}{h_3}$:

$$\begin{aligned} Ma_c(h \to 0) &= Ma_c(h_i) + Ch_i^{\alpha} + O\left(h_i^{\alpha+1}\right) \\ \alpha &= \frac{\ln\left(\frac{Ma_c(h_1) - Ma_c(h_2)}{Ma_c(h_2) - Ma_c(h_3)}\right)}{\ln\left(\frac{h_1}{h_2}\right)} \\ C &= \frac{Ma_c(h_2) - Ma_c(h_3)}{h_3^{\alpha} - h_2^{\alpha}} \end{aligned} \tag{15}$$

where α designates the convergence order of this expansion. This Richardson extrapolation procedure can only be considered meaningful provided the function to be extrapolated is smooth enough and the considered mesh sizes are small enough such that the discrete solutions are in the convergence radius of the asymptotic expansion. If so, the $O\left(h_i^{\alpha+1}\right)$ term in Eq. (15) is of negligible importance with respect to the two others r.h.s. terms and can be therefore dropped. Furthermore, in the present case we have used Eq. (15) in a multidimensional framework, setting $h_i = max(\Delta x_i, \Delta y_i, \Delta z_i)$ and constant $\frac{\Delta x_i}{\Delta z_i} = \frac{\Delta y_i}{\Delta z_i}$ ratios, for $i = 1, 3$.

In the present computations one gets a convergence order $\alpha = 2.989$, this means that halving the mesh size from one mesh to the next one results in an error roughly eight times smaller, see Table 2. So, halving once more time the mesh size would probably enable us to recover the theoretical third order convergence rate associated with the quadratic finite element approximation used in our model ($\alpha \approx p + 1, p$ being the degree of the finite element polynomial approximation) but at a prohibitive cost. Nevertheless, one can notice that the three meshes considered in this spatial convergence analysis produce a value of the extrapolated critical Marangoni number whose relative error from the theoretical reference value is less than $2 \times 10^{-6}\%$, see Table 2.

4.2. ANM computational performance versus classical Newton–Raphson based algorithm

We have checked in the previous section that the present numerical model is able to accurately reproduce the physics of the problem we are interested in modelling. So the next step is to see how our ANM algorithm performs from the computational point of view in comparison with the classical continuation algorithm based on a first-order (Euler) predictor associated with a Newton–Raphson corrector and a pseudo arc-length parameterization, as introduced by Keller [35], which will be designated ENR-ALP in short.

We have performed such comparisons on the infinite layer test case of the previous section for the mesh $M2$ of Table 1, descending the B_2 branch from $Ma = 90$ and positive mean velocity to its symmetrical point with respect to the turning point of the branch (point B in Fig. 2(a)). The ANM continuation computations have been performed using a truncation order of the power series limited to $n = 30$, a tolerance criteria $\delta = 10^{-12}$ in the predictor step and no correction stage. On the other hand, the ENR-ALP algorithm has been used with a tolerance on the L_2 norm of the residual $\varepsilon = 5 \times 10^{-6}$ and a pseudo-arc length $\Delta s = 1$. Both computations have been performed on an IBM-SP6 computer using 32 cores for this test case. The specific behaviour of these two algorithms are clearly illustrated in Fig. 3, where the signed mean velocity over the computational domain is plotted versus the continuation parameter, i.e. the Marangoni number (bold dots and squares stand for step-ends of the ANM and ENR-ALP continuation algorithms, respectively). It turns out that the ANM algorithm outperforms the standard ENR-ALP algorithm in self-adapting its step-length along the branch path-following. This intrinsic feature is very effective in locating accurately the bifurcation point, but as the step-length decreases in a geometric series way in the bifurcation vicinity (see Fig. 3(b)), it takes many steps to go through the bifurcation point and keeps going the path-following on this branch. After the bifurcation point, the ANM step-lengths increase again and a slight deviation from the ENR-ALP reference results can be observed until the end of the path-following in the lower part of the branch. Concerning the ENR-ALP algorithm, its step-length is almost constant all along the path-following except in the vicinity of the turning point where it is roughly halved compared to elsewhere (see Fig. 3(b)).

Furthermore, quantitative results of this comparison are presented in Table 3, considering both CPU times and accuracy of the results expressed by the L_2 norm of the residual at the end of the continuation. It is noteworthy that descending along the B_2 branch from $Ma = 90$, the ANM algorithm (without any correction step) requires much less global matrix factorisations (35 versus 201) than the classical ENR-ALP algorithm. So, for the present class of moderate to large size algebraic systems to factorize (420,000 *dof* for mesh $M2$), the AMN algorithm requires about five times less CPU-time on 32 cores than the ENR-ALP algorithm. On the other hand, without any correction step the accuracy of the ANM algorithm degenerates a little bit in the course of the continuation (particularly after going through the bifurcation point), so that the L_2 norm of the residual at the end of the computation is four times greater than that obtained with the ENR-ALP algorithm. There are two ways to prevent from this accuracy degeneration, one can either reduce the tolerance criteria δ or introduce a correction step when the residual norm becomes greater than the required tolerance ε. The former shortens the step size so a higher number of continuation steps are involved, whereas the latter does not change the step length but introduce a few additional global matrix factorizations along the continuation. Concerning the ENR-ALP algorithm, increasing the step length (Δs) reduces the number of steps but increases the number of Newton–Raphson iterations at every correction step. Conversely, reducing

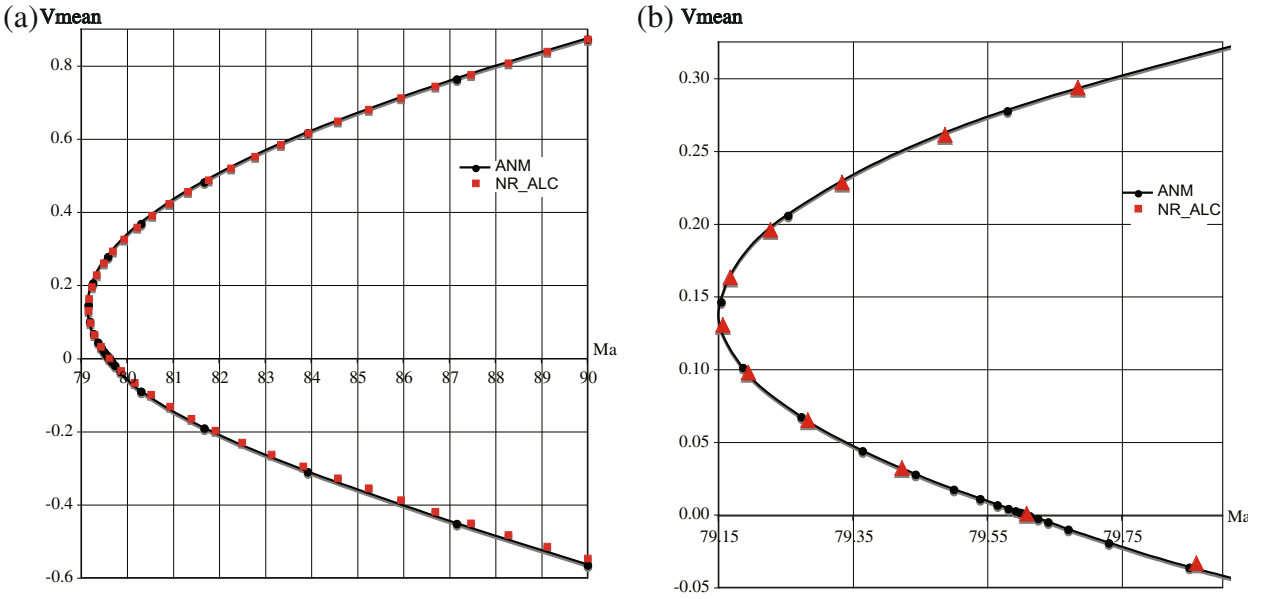


Fig. 3. ANM and ENR-ALP continuation algorithm comparisons on surface-tension-driven convection in a liquid layer of infinite horizontal extension following the B_2 branch. (a) Signed mean velocity versus Marangoni number for the two algorithms. (b) Close-up in the vicinity of the threshold.

Table 3
Comparison of some computational characteristics of the ANM versus classical Newton–Raphson based algorithm on surface-tension-driven convection, for path-following descending the B_2 branch with mesh $M2$.

Computational characteristics	AMN algorithm ($n = 30, \delta = 10^{-12}$)	Classical N–R based algorithm ($\Delta s = 1, \varepsilon = 5 \times 10^{-6}$)
Total # continuation steps	35	44
Total # global matrix factorisations	35	201
Average # factorisation/continuation step	1	4.6
$\|\|Residual\|\|_{L_2}$ at end of continuation	1.62×10^{-5}	4.87×10^{-6}
Total CPU-time in s (32 cores IBM SP6)	3346	15,980

the step-length reduces the number of correction iterations but increases the number of steps. So, with the ENR-ALP algorithm the total number of global matrix factorisations does not change significantly when tuning the step-length parameter. A more sophisticated self-adaption of the step-length in the course of the continuation would be preferred.

4.3. Capabilities of our ANM implementation for large size algebraic systems

As we are interested in dealing with algebraic systems of quite large size (up to few millions of unknowns) with a direct solver owing to the poor conditioning of the fully-coupled incompressible fluid flow and heat transfer problem, the parallel implementation has to be analysed in details. In this respect, we have reported in Table 4 the CPU times related to one ANM continuation step of the surface-tension-driven convection tests of the previous section, for the three discrete problems associated with the meshes of Table 1. Thanks to the good parallel efficiency of the PETSc implementation [7] the global matrix and right-hand-side assembly times scale quite linearly versus the number of degrees of freedom, provided a suitable

Table 4
Characteristic CPU times (in seconds) to perform one ANM continuation step (order $n = 50$), obtained on various numbers of IBM-SP6 cores, for the three discrete problems associated to the meshes considered in Table 1.

Mesh #	$M1$ (56,916 *dof*)			$M2$ (418,946 *dof*)			$M3$ (3,210,372 *dof*)		
# Cores	2	4	8	16	32	64	96	128	160
K_t assembly	4.427	2.226	1.134	4.572	2.470	1.385	4.721	3.712	3.569
LU factorisation	11.86	7.197	5.836	116.6	78.63	87.86	1147	903.1	866.9
Last r.h.s assembly	2.963	1.484	0.756	3.079	1.539	0.798	3.199	2.519	2.418
Algebraic system solving	0.131	0.089	0.084	0.974	0.822	1.609	7.269	5.725	5.496
Complete ANM continuation step	95.90	50.45	29.72	203.5	135.4	166.9	1589	1327	1288

amount of operations per computing element (core on the IBM-SP6) is maintained. In the present case, the successively refined meshes are designed to have a constant ratio of eight between their respective total node numbers (so do their associated *dof*), therefore we have applied the same ratio of eight to the minimum number of cores requested for the three problem sizes. This welcome linear scaling behaviour has been observed in all the global matrix and *r.h.s.* assembly computations reported in Table 4, but this nice scaling is somehow lost for the factorisation step of such large matrix sizes.

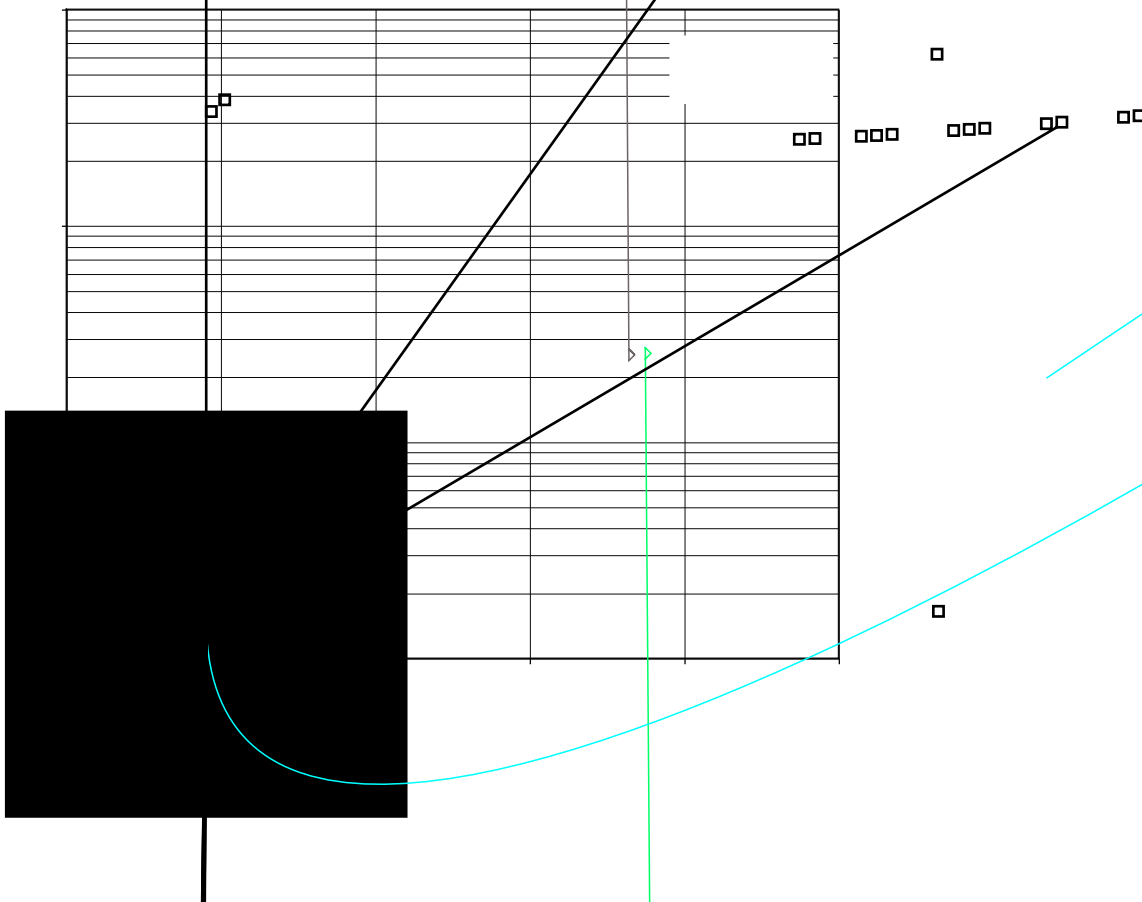
The global tangent matrix is factored once per continuation step with the Multifrontal Massively Parallel sparse direct Solver (MUMPS [5], which is interfaced with the PETSc library), then n linear solves are performed (one at every order of the power series expansion) with a right hand side specific to each order. It is noteworthy that for the large size algebraic systems considered in the present computations the factorisation stage requires a huge amount of both CPU-time and RAM. Therefore, the minimum number of requested cores has been selected such that the maximum RAM per core never exceeds 3.2 Go, which is the upper limit of the RAM/core available on the IBM-SP6 supercomputer we have access to. So, it follows from this limitation that the factorisation step represents roughly 10%, 60% and 70% of the total CPU time per continuation step with a power series expansion truncated at order $n = 50$, for the algebraic systems of size 56,916, 418,946 and 3,210,372 *dof*, respectively. This points out that the size of the algebraic system associated with the continuation problem significantly affects the relative importance of the CPU-time related to the matrix factorisation step and the expansion series computation one.

This dependency is clearly illustrated in Fig. 4, where the total CPU time to perform one ANM step is plotted versus the truncation order n of the power series expansion, for the three discrete problems associated to the meshes considered in Table 1. In order to perform a fair comparison analysis, the number of requested cores respects the problem size ratio, so that every core deals with approximately the same amount of operations. As the CPU times span more than two order of magnitude, a logarithmic scale has been used for the ordinates, so the slopes should be analysed correspondingly.

It turns out that the bigger the algebraic system size, the smaller the relative time to perform a high order power series expansion with respect to the factorisation step duration. This means that to reach a global computational efficiency of the ANM continuation algorithm one should relate the truncation order of the power series expansion to the algebraic system size. Indeed, increasing the truncation order extends the step length and consequently reduces the number of continuation steps for a comparable step-end accuracy.

5. Influence of container shape and size on RBM convection pattern

Once numerous validations have been performed we have run our numerical model to study the influence of the container shape and size on the main fluid flow characteristics, such as the threshold value, its associated pattern and several



flow patterns in the supercritical range (beyond the onset of convection). In the present section we are interested in confined RBM convection, which can be characterised by two geometrical quantities (the container shape and liquid layer aspect ratio Γ) along with the distance from the threshold. The latter is defined by $\epsilon = \frac{Ma}{Ma_{c0}} + \frac{Ra}{Ra_{c0}} - 1$, where Ma_{c0} and Ra_{c0} designate critical (at threshold) Marangoni and Rayleigh numbers, respectively, both depending on the container shape and liquid layer aspect ratio.

Moreover in the present confined RBM convection the cellular pattern results from two kinds of independent constraints: the first one comes from the cellular convection in the bulk, which sets in as a rather regular tiling of the plane, and the second one is induced by surrounding lateral walls. Hence, when the pattern can reconcile these two types of constraints, it can result in a quasi-perfect cellular pattern, whereas otherwise defects can be observed. Therefore, we have investigated the influence of the container shape and size for a silicone oil layer ($Pr = 900$ at 25 °C) topped by a thin air layer ($Bi = 0.1$). We have considered basic container shapes (regular polygons from 3 to 6 wall sides), of various sizes (Γ ranging from 2 up to 25). As the threshold depends on both the container shape and liquid layer aspect ratio, we have used our ANM path-following algorithm to get the steady-state solution in each configuration (for given container shape and Γ) at the same prescribed distance from the threshold $\epsilon = 0.25$, with a thermal Bond number set to $Bo = \frac{Ra}{Ma} = 10$ in all cases.

The meshes associated to each particular shape (triangular, pentagonal and hexagonal), are made up of eight identical layers in the vertical direction of H27 finite elements and comparable spatial resolution in the horizontal directions. The details of the meshes used in these computations are reported in Table 5.

Starting in each case from a purely conductive regime ($\epsilon = -0.1$) we have performed steady-state path following computations with our ANM algorithm until reaching the target solution at $\epsilon = 0.25$. This procedure resulted in a different number of continuation steps depending on the size and shape of the container, typically from 70 steps in the cases where only one bifurcation (the threshold of convection) has been encountered and up to 165 steps when a secondary bifurcation also exists in the spanned interval ($-0.1 \leqslant \epsilon \leqslant 0.25$). So the CPU time required for the computations in one configuration (defined by a given aspect ratio and container shape) ranged from 28,000 to 66,000 s on 48 IBM-SP6 cores.

We present in Figs. 5–7 top views of steady-state thermal fields at the free surface of regular triangular, pentagonal and hexagonal containers for various aspect ratios Γ. For the prescribed distance from the threshold the patterns evolve progressively for increasing aspect ratio, from only one cell (inheriting the container shape), to several wall-cells, then a single central-core-cell (intrinsic cell) surrounded with wall-cells, and finally a central zone made up of intrinsic-cells surrounded with wall-cells. To go further in the analysis, one can define the container shape compatibility (or incompatibility) to a regular convection pattern as the ability (or inability) for a given container to induce a quasi-perfect steady-state cellular pattern. According to this definition, the equilateral-triangle and the regular hexagonal shape containers can conform to the basic RBM hexagonal pattern, provided an extra size compatibility condition is fulfilled: perimeter and diameter should match

Table 5
Meshes used to study the influence of the container shape and size on the steady-state RBM convection patterns.

Domain shape	H27 finite elements (horizontal plane x in the depth)	Mesh nodes (horizontal plane x in the depth)	Degrees of freedom (*dof*)
Triangular	41,496 (5187 × 8)	321,487 (18,911 × 17)	1,285,948
Pentagonal	41,640 (5205 × 8)	323,527 (19,031 × 17)	1,294,108
Hexagonal	41,496 (5187 × 8)	321,487 (18,911 × 17)	1,285,948

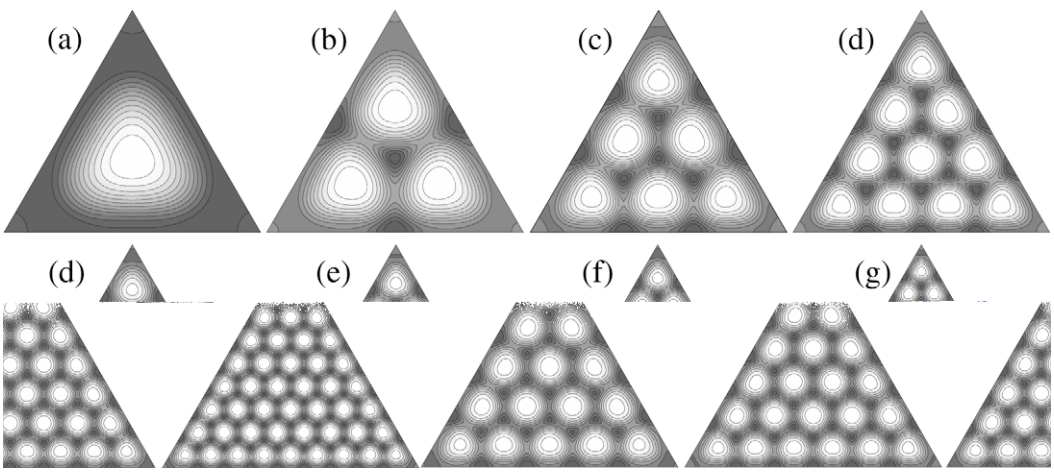


Fig. 5. Free surface thermal field for RBM convection in triangular containers ($\epsilon = 0.25, Bo = 10, Pr = 900$ and $Bi = 0.1$). (a) $\Gamma = 3.0$; (b) $\Gamma = 6.1$; (c) $\Gamma = 8.4$; (d) $\Gamma = 10.7$; (e) $\Gamma = 12.9$; (f) $\Gamma = 15.2$; (g) $\Gamma = 17.5$; and (h) $\Gamma = 22.0$.

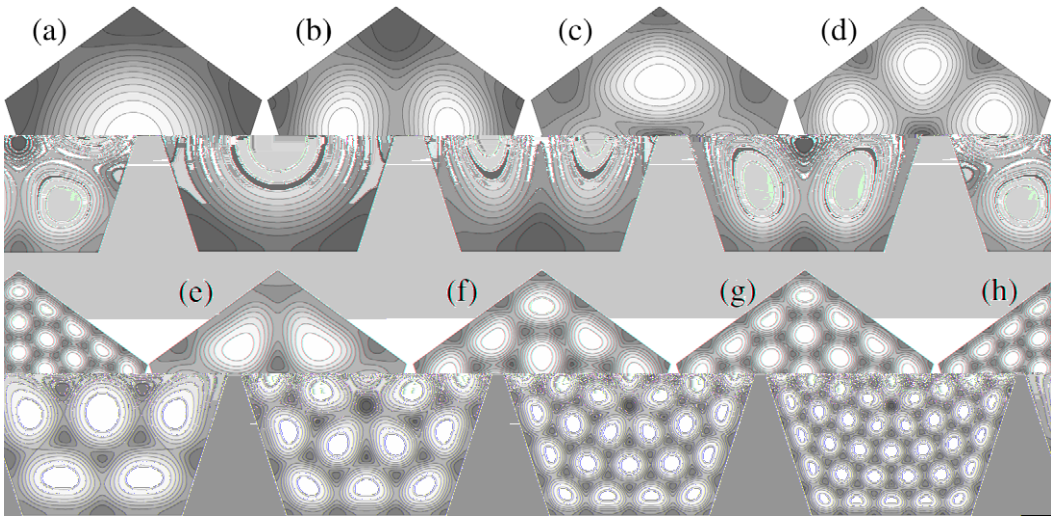


Fig. 6. Free surface thermal field for RBM convection in pentagonal containers ($\epsilon = 0.25, Bo = 10, Pr = 900$ and $Bi = 0.1$). (a) $\Gamma = 3.0$; (b) $\Gamma = 4.6$; (c) $\Gamma = 6.2$; (d) $\Gamma = 7.7$; (e) $\Gamma = 9.25$; (f) $\Gamma = 13.4$; (g) $\Gamma = 18.5$; and (h) $\Gamma = 23.6$.

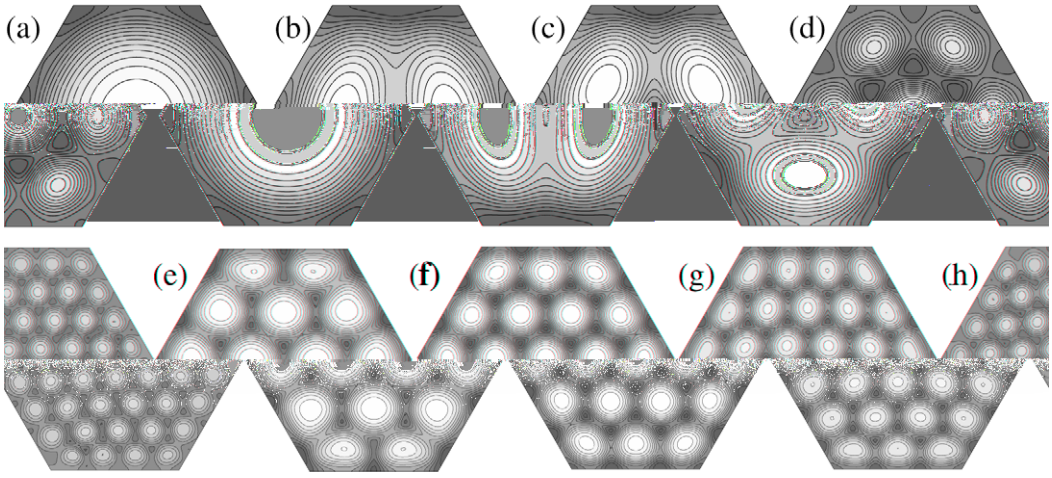


Fig. 7. Free surface thermal field for RBM convection in hexagonal containers ($\epsilon = 0.25, Bo = 10, Pr = 900$ and $Bi = 0.1$). (a) $\Gamma = 3.0$; (b) $\Gamma = 5.0$; (c) $\Gamma = 6.0$; (d) $\Gamma = 10.4$; (e) $\Gamma = 14.0$; (f) $\Gamma = 17.5$; (g) $\Gamma = 21.0$; and (h) $\Gamma = 27.75$.

an integer times the intrinsic cell size. In this case the resulting patterns are such that the core cells have very close characteristics compared to those that could exist in the theoretical layer of infinite horizontal extension. On the other hand, regular pentagonal containers are the archetype of incompatible vessels with respect of the compatibility condition considered.

6. Conclusion

We have presented in this paper a parallel computer implementation of the Asymptotic Numerical Method to perform continuation along branches of steady state thermal convective instability problems. This computational model features the ability to deal efficiently with large size algebraic systems, up to few millions of degrees of freedom, thanks to an implementation in the PETSc framework, interfaced with the MUMPS library.

Its accuracy and efficiency to locate bifurcation points and follow branches of steady state solutions has been analysed in the case of surface-tension-driven convection for a liquid layer of infinite horizontal extension. Indeed, in this particular case a reliable value for the threshold (onset of convection in the liquid layer) has been provided by linear stability analyses and has been therefore used as reference value for the validation of our computations. Moreover a spatial convergence analysis confirmed that the convergence order is very close to the optimal one (2.989 instead of 3), so that one can be confident on the Richardson extrapolation value of the threshold from the three meshes considered.

After this quantitative validation, we have also addressed the computational efficiency of the ANM algorithm through a comparison with a classical continuation algorithm based on a first order predictor associated with a Newton–Raphson corrector and a pseudo arc-length parameterization. The two path-following algorithms have tracked the same branch, which crosses two other branches at the threshold. The ANM algorithm performed the continuation five times faster than the classical one, with a power series expansion truncated at order 30 and no correction step. Nevertheless, with these characteristics, the residual at the end of the path-following is roughly four times greater with the ANM algorithm than with the classical predictor-corrector one. The parallel implementation has also been extensively tested in the case of large size algebraic systems. Indeed, in this case the major part of the total CPU time is consumed to perform the factorisation stage of a very large and ill-conditioned algebraic system. However, the multifrontal massively parallel direct solver (MUMPS) enabled us to efficiently deal with algebraic systems up to 3.2 millions equations on an IBM-SP6 supercomputer.

Therefore, in the present class of problems in which one has to deal with large size and ill-conditioned algebric systems one can legitimately assert that the larger the size of the algebraic system to solve, the more efficient the ANM algorithm will be with respect to the classical Newton–Raphson based continuation algorithms, since the former requires about four times less factorization stages than the latter. Therefore, the ANM is believed to become a continuation method of choice in the computational fluid mechanics field where high spatial resolutions are often required to access to the most interesting hydrodynamical or thermal-convection instability problems.

Acknowledgment

The first author thanks CNRS for providing substantial computing resources on its IBM-SP6 supercomputer at IDRIS (Orsay, France).

References

[1] C. Allery, J.-M. Cadou, A. Hamdouni, D. Razafindralandy, Application of the asymptotic numerical method to the Coanda effect study, Rev. Eur. Elém. Finis 13 (2004) 57–77.
[2] E. Allgower, K. Georg, Numerical Continuation Methods, Springer-Verlag, New York, 1990.
[3] P.R. Amestoy, I.S. Duff, J. Koster, J.-Y. L'Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM J. Matrix Anal. Appl. 23 (1) (2001) 15–41.
[4] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, Multifrontal parallel distributed symmetric and unsymmetric solvers, Comput. Meth. Appl. Mech. Eng. 184 (2000) 501–520.
[5] P.R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, Parallel Computing 32 (2) (2006) 136–156.
[6] L. Azrar, B. Cochelin, N. Damil, M. Potier-Ferry, An asymptotic numerical method to compute the post-buckling behaviour of elastic plates and shells, Int. J. Numer. Meth. Eng. 36 (1993) 1251–1277.
[7] S. Balay, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc users manual. Technical Report ANL-95/11 – Revision 2.1.5, Argonne National Laboratory, 2004.
[8] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Web page, 2001. <http://www.mcs.anl.gov/petsc>.
[9] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A.M. Bruaset, H.P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.
[10] H. Bénard, Les tourbillons cellulaires dans une nappe liquide transportant de la chaleur par convection en régime permanent, Ann. Chim. Phys. 23 (1901) 62–144.
[11] M. Bestehorn, Square patterns in Bénard–Marangoni convection, Phys. Rev. Lett. 76 (1996) 46–49.
[12] P.C. Biswal, A.R. Rao, Thermal instability in a three-dimensional rigid container with prescribed heat flux at lower boundary, Int. J. Eng. Sci. 39 (2001) 1315–1325.
[13] J. Bragard, M.G. Velarde, Review article: Bénard convection flows, J. Non-Equilib. Thermodyn. 22 (1997) 1–19.
[14] J.M. Cadou, B. Cochelin, M. Potier-Ferry, A numerical method to compute bifurcation points in fluid mechanics, Eur. J. Mech. B: Fluids 25 (2006) 234–254.
[15] J.-M. Cadou, N. Damil, M. Potier-Ferry, B. Braikat, Projection techniques to improve high-order iterative corrector, Finite Elem. Anal. Des. 41 (2004) 285–309.
[16] J.-M. Cadou, M. Potier-Ferry, B. Cochelin, N. Damil, Asymptotic numerical method for stationary Navier–Stokes equations and with Petrov–Galerkin formulation, Int. J. Numer. Meth. Eng. 50 (2001) 825–845.
[17] G. Carey, J.T. Oden, Finite Elements. Vol. 6: Fluids Mechanics. The Texas Finite Element Series, Prentice-Hall, Englewood Cliffs, NJ, 1986.
[18] P. Cerisier, C. Pérez-Garcia, C. Jamond, J. Pantaloni, Wavelength selection in Bénard–Marangoni convection, Phys. Rev. A 35 (1987) 1949–1952.
[19] B. Cochelin, A path following technique via an asymptotic numerical method, Comput. Struct. 53 (1994) 1181–1192.
[20] B. Cochelin, N. Damil, M. Potier-Ferry, The asymptotic numerical method: an efficient perturbation technique for non-linear structural mechanics, Rev. Eur. Elém. Finis 5 (1994) 415–442.
[21] B. Cochelin, N. Damil, M. Potier-Ferry, Asymptotic numerical method and Padé approximants for non-linear elastic structures, Int. J. Numer. Meth. Eng. 37 (1994) 1187–1213.
[22] B. Cochelin, N. Damil, M. Potier-Ferry, Méthode Asymptotique Numérique, Hermès–Lavoisier, 2007.
[23] P. Colinet, J.C. Legros, M.G. Velarde, Non-inear Dynamics of Surface-tension-instabilities, Wiley-VCH, 2001.
[24] N. Damil, M. Potier-Ferry, A new method to compute perturbed bifurcations: application to the buckling of imperfect elastic structures, Int. J. Eng. Sci. 28 (1990) 943–957.
[25] P.C. Dauby, G. Lebon, Bénard–Marangoni instability in rigid rectangular containers, J. Fluid Mech. 329 (1996) 25–64.
[26] P.C. Dauby, G. Lebon, E. Bouhy, Linear Bénard–Marangoni instability in rigid circular containers, Phys. Rev. E 56 (1997) 520–530.
[27] G. Dhatt, G. Touzot, Une présentation de la méthode des éléments finis, Maloine et Les Presses de L'Université Laval, 1981.
[28] H.A. Dijkstra, Surface-tension driven cellular patterns in three-dimensional boxes – the formation of hexagonal patterns, Microgravity Sci. Technol. 8 (1995) 155–162.
[29] E. Doedle, H.B. Keller, J. Kernevez, Numerical analysis and control of bifurcation problems (i) bifurcation in finite dimensions, Int. J. Bifurcation Chaos 3 (1991) 493–520.
[30] P.G. Draizin, W.H. Reid, Hydrodynamic Stability, Cambridge Press, 1981.

[31] R.D. Falgout, U.M. Yang, Hypre: a library of high performance preconditioners, Lect. Notes Comput. Sci. (2002) 632–641.
[32] K. Eckert, M. Bestehorn, A. Thess, Square cells in surface-tension-driven Bénard convection: experiment and theory, J. Fluid Mech. 356 (1998) 155–197.
[33] A. Elhage-Hussein, M. Potier-Ferry, N. Damil, A numerical continuation method based on Padé approximants, Int. J. Solids Struct. 37 (2000) 6981–7001.
[34] C. Godrèche, P. Manneville, Hydrodynamics and non-linear instabilities, Collection Aléa, CEA Saclay, 1998.
[35] H.B. Keller, Numerical Solution of Bifurcation and Non-linear Eigenvalue Problems, Academic Press, New York, 1977.
[36] E.L. Koschmieder, Bénard Cells and Taylor Vortices, Cambridge University Press, 1993.
[37] E.L. Koschmieder, S.G. Pallas, Heat transfer through a shallow, horizontal convecting fluid layer, Int. J. Heat Mass Transfer 17 (991) (1974) 1002.
[38] E.L. Koschmieder, S.A. Prahl, Surface-tension-driven Bénard convection in small containers, J. Fluid Mech. 215 (1990) 571–583.
[39] E.B. Kusnetsov, Optimal parametrization in numerical construction of a curve, J. Franklin Inst. 334 (2007) 658–671.
[40] H. Lahmam, J. Cadou, H. Zahrouni, N. Damil, M. Potier-Ferry, High-order predictor-corrector algorithms, Int. J. Numer. Meth. Eng. 55 (2002) 685–704.
[41] P. Manneville, Structures dissipatives, Chaos et turbulences, Collection Aléa, CEA Collection Aléa, CEA Saclay, 1991.
[42] M. Medale, P. Cerisier, Numerical simulation of Bénard–Marangoni convection in small aspect ratio containers, Numer. Heat Trans. A 42 (2002) 55–72.
[43] A.A. Nepomnyashchy, M.G. Velarde, P. Colinet, Interfacial Phenomena and Convection, Chapman and Hall, 2002.
[44] D.A. Nield, Surface-tension and buoyancy effects in cellular convection, J. Fluid Mech. 19 (1964) 341–352.
[45] K. Nitschke, A. Thess, Secondary instability in surface-tension-driven Bénard convection, Phys. Rev. E 52 (1995) R5772–R5775.
[46] J.R. Pearson, On convection cells induced by surface-tension, J. Fluid Mech. 4 (1958) 489–500.
[47] V. Regnier, P.C. Dauby, P. Parmentier, G. Lebon, Square cells in gravitational and capillary thermoconvection, Phys. Rev. E 55 (1997) 6860–6865.
[48] M.F. Schatz, S.J. Vanhook, W.D. Mccormick, J.B. Swift, H.L. Swinney, Time-independent square patterns in surface-tension-driven convection, Phys. Fluids 11 (1999) 2577–2582.
[49] R. Seydel, Non-linear computation, J. Franklin Inst. 334B (1997) 1015–1047.
[50] E.M. Sparrow, R.J. Goldstein, V.K. Jonsson, Thermal instability in a horizontal fluid layer: effect of boundary conditions and non-linear temperature profile, J. Fluid Mech. 19 (1964) 513–528.
[51] A. Thess, S.A. Orszag, Surface-tension-driven Bénard convection at infinite Prandtl number, J. Fluid Mech. 283 (1995) 201–230.
[52] M. Van Dyke, Computer extension of perturbation series in fluid mechanics, SIAM J. Appl. Math. 28 (1975) 720–734.
[53] M. Van Dyke, Perturbation Methods in Fluid Mechanics, The Parabolic Press, Stanford, California, 1975.
[54] A.A. Zaman, R. Narayanan, Interfacial and buoyancy-driven convection: the effect of geometry and comparison with experiments, J. Colloid Interf. Sci. 179 (1996) 151–162.
[55] H. Zahrouni, B. Cochelin, M. Potier-Ferry, Computing finite rotations of shells by an asymptotic numerical method, Comput. Meth. Appl. Mech. Eng. 175 (1999) 71–85.