Numerical bifurcation analysis for three-dimensional fluid dynamics problems as new features in the multi-physical open-source software ELMER.

Y. Guevel[a,*], T. Allain[a], G. Girault[a,b], J.M. Cadou[a]

[a] *Laboratoire d'Ingénierie des Matériaux de Bretagne, Université de Bretagne Sud, Rue de Saint Maudé, Lorient, France*
[b] *Centre de recherche des Ecoles de Saint-Cyr Coëtquidan, Ecoles Militaires de Coëtquidan, Guer, France*

Abstract

New features for numerical bifurcation analysis in fluid dynamics are implemented in the open-source multi-physical software ELMER . Specific tools for continuation of three-dimensional steady flow solutions, detection of bifurcation points and branch switching are provided. The Asymptotic Numerical Method is used as nonlinear solver and also for its specific bifurcation analysis abilities. A full description of the methods and their implementation in ELMER is proposed. High performance non linear solvers are proposed for models involving million degrees of freedoms in conjunction with the use of MUMPS and OpenBlas libraries. Pitchfork bifurcations and limit points in internal flows are used as validation cases.

Keywords: Asymptotic Numerical Method, ELMER, Three-dimensional flow, Bifurcation analysis, Continuation, Branch switching, High Performance Computing

1. Introduction

Numerical bifurcation analysis of flow problems governed by the discretized Navier-Stokes equations with a large number of degrees of freedom is still a challenging task. A review of the existing numerical bifurcation methods in fluid dynamics is proposed in [25]. Toolbox exists for ordinary differential equations, such as MatCont [24] and AUTO-07p [27], hence with a small number of degrees of freedom [46]. A large scale toolbox for systems of discretized partial differential equations is LOCA (continuation and bifurcation analysis) [60] as part of the Trilinos framework [65].

An alternative to the well known incremental-iterative methods [26, 63, 71] is the Asymptotic Numerical Method (ANM) [21]. ANM is the association of a perturbation technique and a spatial discretization method, in our case the finite element method. It has been successfully applied in nonlinear solid mechanics [8, 22] or in fluid mechanics [41, 42, 51]. The main advantage of the ANM is its ability to determine analytical nonlinear solutions [16, 20] with computational CPU times lower than with the classical incremental-iterative method. On these nonlinear solutions, some bifurcation indicators have been introduced [15, 22, 40]. High performance computation (HPC) based on ANM for flow problems has been proposed in [50] as an in-house code.

HPC study for the Navier-Stokes equations exists as [54] based on the open-source project FEniCS [47] or in [36] with the open-source multi-physical software ELMER [23]. ELMER is developed by the CSC - IT Center for Science (CSC) in Finland. It includes models dedicated to fluid dynamics, structural mechanics, electromagnetics, heat transfer and acoustics, for example. These models are mainly described by partial differential equations which are solved using the Finite Element Method (FEM) [73].

Our objective is to propose numerical bifurcation methods based on ANM [22, 40, 51] for large scale flow problems as new features in ELMER . In this paper, implementation of those methods as new modules and User Defined Solver (UDS) are described. Moreover, HPC is performed in ELMER by the mean of the direct linear solver MUMPS [2] coupled with the multi-threaded linear algebra library OpenBlas [72]. We assume that the reader has some basic knowledge of how to write simple solvers for partial differential equations using the ELMER framework. More details on UDS programming are available in the ELMER documentation : "Basic Programming" [58], "Basic Programming with Elmer" [48] and "Partial Differential Equations" in the Elmer Tutorials [49].

We proposed an implementation of classic features of the ANM for steady flow solutions. Continuous analytic representation of nonlinear branches is performed using path-following technique [16, 20]. Then, power series analysis based on previous work [67] and applied to the ANM context [22, 51] allows steady bifurcation point detection.

*Corresponding author
Email address: yann.guevel@univ-ubs.fr (Y. Guevel)

Branch switching technique adapted to the Navier-Stokes equations [40] is also implemented. With the help of the ELMER code, those methods can be used for large scale flow simulations.

This paper is organized as follow. The numerical methods are described in the first part. Implementation of the ANM in ELMER is then detailed in the second part. Finally, validations are proposed with bifurcation analysis for three classic numerical cases including a three-dimensional internal flow problem.

2. Numerical methods

2.1. Governing equations

The steady Navier-Stokes equations for Newtonian and incompressible fluid in a domain (Ω) are :

$$
\begin{aligned}
-2\mu \, div\, \mathbf{D}(\mathbf{u}) + \mathbf{grad}\, p + \rho \mathbf{u} \cdot \mathbf{grad}\, \mathbf{u} &= \mathbf{0} \quad \text{in } (\Omega) && (1)\\
div\, \mathbf{u} &= 0 \quad \text{in } (\Omega) && (2)\\
\mathbf{u} &= \lambda \mathbf{u}_d \quad \text{on } (\Gamma) && (3)
\end{aligned}
$$

with $\mathbf{u}$ and p respectively the velocity and pressure unknowns, and λ a control parameter. A generic velocity profile $\mathbf{u}_d$ is prescribed on the boundary (Γ). $\mathbf{D}(\mathbf{u})$ is the symmetric part of the velocity gradient tensor. Both density ρ and kinematic viscosity μ are constant in the fluid domain. The Reynolds number is defined as $Re = \lambda \frac{\rho \mathbf{u}_d^{max} L}{\mu}$, with L a characteristic length of the model.

2.2. Finite Element Method in ELMER

ANM techniques are implemented as User Defined Solver based on the existing 'FlowSolver' module in ELMER [23]. As the (FEM) [28, 73] is used, short description of the spatial discretization steps is proposed. Discretized operators and a stabilized FEM formulation are presented here in order to make this paper self-contained.

2.2.1. Weak form

FEM relies on a weak integral form of the partial differential equations to be solved. In the context of the standard Galerkin FEM, test functions $\delta\mathbf{v}$ and δq, respectively for velocity and pressure, are chosen in the suitable functional spaces. Then multiplying the governing equations with those functions and integrating over the domain, one obtains the so-called weak form. An abusive operator notation is used in the context of the continuous problem for the sake of concision and clarity in order to introduce the numerical methods we implemented in ELMER . For the same reason, this notation will be used for the discrete operators.

The variational formulation of the nonlinear problem is written in the following operator form:

$$\mathrm{L}(\mathbf{U}) + \mathrm{Q}(\mathbf{U}, \mathbf{U}) = \lambda \mathbf{F} \quad \text{in} \quad \Omega \tag{4}$$

with $\mathbf{U} = \{\mathbf{u}, p\}$ a mixed vector. Dirichlet boundary condition Eq.(3) produces the right-hand side term $\lambda\mathbf{F}$, which is equivalent to a fictitious force [21, 73].

The linear operator L contains the weak form of the diffusive term, the pressure gradient and the continuity equation:

$$\mathrm{L}(\mathbf{U}) := \mathrm{L}(\mathbf{u}, p; \delta\mathbf{v}, \delta q) = \mathrm{A}(\mathbf{u}; \delta\mathbf{v}) - \mathrm{B}(\delta\mathbf{v}; p) + \mathrm{B}(\mathbf{u}; \delta q) \tag{5}$$

Where using the following definition for the continuity equation Eq.(2) as a bilinear form:

$$\mathrm{B}(\mathbf{u}; \delta q) = \int_\Omega div\, \mathbf{u}\ \delta q \, d\Omega \tag{6}$$

Integration by parts of the pressure gradient variational form is equal to $-\mathrm{B}(\delta\mathbf{v}; p)$.

Moreover, the weak form of the strain rate tensor in Eq.(1), is obtained using the Green-Gauss divergence theorem, and leads to:

$$\mathrm{A}(\mathbf{u}; \delta\mathbf{v}) = \int_\Omega 2\mu \mathbf{D}(\mathbf{u}) : \mathbf{grad}\, \delta\mathbf{v} \, d\Omega \tag{7}$$

Finally, the nonlinear operator Q contains the convective term:

$$\mathrm{Q}(\mathbf{U}, \mathbf{U}) := \mathrm{Q}(\mathbf{u}, \mathbf{u}; \delta\mathbf{v}) = \int_\Omega \rho \left((\mathbf{u} \cdot \mathbf{grad})\mathbf{u} \right) \cdot \delta\mathbf{v} \, d\Omega \tag{8}$$

2.2.2. Finite element discretization

Spatial discretization of the variational formulation is based on the classical Galerkin finite element approximation. The unknowns $\mathbf{u}$, p and the associated test functions $\delta\mathbf{v}, \delta q$ use the same basis functions $N_i(\mathbf{x})$ evaluated at Cartesian coordinates $\mathbf{x}$:

$$\mathbf{u}(\mathbf{x}) \approx \sum_{i=1}^{n} N_i(\mathbf{x})\overline{\mathbf{u}}_i = [N(\mathbf{x})]\{\overline{\mathbf{u}}\} \tag{9}$$

with $\overline{\mathbf{u}}_i$ the vectors of nodal values. The continuous problem is then transformed into a set of algebraic nonlinear problems to solve.

As an example, the quadratic operator Q is obtain with an assembly of local element contribution over the domain:

$$\mathrm{Q}(\mathbf{u}_a, \mathbf{u}_b) = \sum_E \mathrm{Q}^E(\mathbf{u}_a, \mathbf{u}_b) \tag{10}$$

with $\mathrm{Q}^E(\mathbf{u}_a, \mathbf{u}_b)$ the local quadratic operator for two fields $\mathbf{u}_a, \mathbf{u}_b$:

$$\mathrm{Q}^E(\mathbf{u}_a, \mathbf{u}_b) = \rho \int_{\Omega_E} (\mathbf{u}_a \cdot \mathbf{grad}\, \mathbf{u}_b)[N(\mathbf{x})]\, d\Omega \tag{11}$$

$$\approx \rho \sum_{k=1}^{NG} w_k \det(J_E(\xi_k))(\mathbf{u}_a \cdot \mathbf{grad}\, \mathbf{u}_b)[N(\xi_k)] \tag{12}$$

with Gauss quadrature to perform the integration over the element, ξ_k is the Gauss point coordinate and w_k the associated weight, $\det(J_E(\xi_k))$ is the Jacobian determinant of the mapping from global $\mathbf{x} \in \Omega$ to local coordinate $\xi \in \widehat{\Omega}_E$. The reader may found more details on this part in ELMER dedicated documentation [48, 49].

In this study, quadrangles or hexahedrons mixed finite elements with linear $\mathbb{Q}_1/\mathbb{Q}_1$ or quadratic $\mathbb{Q}_2/\mathbb{Q}_2$ interpolation are chosen. Those equal-order interpolation of mixed finite elements in velocity-pressure are known to create locking phenomenon in pressure or spurious mode [5]. This is the reason why different stabilization techniques are available in ELMER .

2.2.3. Stabilized finite element

We use the so-called 'Stabilized finite element' [33] as proposed in the ELMER 'Flow Solver' module. A nice review of this Petrov-Galerkin formulation is detailed in [34, 35]. The Galerkin formulation is modified by the addition of mesh-dependent terms in the test functions. The latter are modified as new weighted residuals of the differential equations. This technique makes it possible to circumvent the well-known Ladyzhenskaya-Babuska-Brezzi (LBB) or inf-sup condition, connected with the classical mixed finite elements [28, 73].

The diffusive operator is written in a semi-discrete form:

$$\mathrm{A}_{stab}(\mathbf{u}, p; \delta\mathbf{v}) = \mathrm{A}(\mathbf{u}; \delta\mathbf{v}) + \sum_E \int_{\Omega_E} R_E(\mathbf{u}, p)\ \tau\, P(\delta\mathbf{v})\ d\Omega \tag{13}$$

where $R_E(\mathbf{U})$ is the elementary Navier-Stokes residual operator and $P(\delta\mathbf{v})$ is a test function equation operator. Moreover the continuity equation is also modified:

$$\mathrm{B}_{stab}(\mathbf{u}; \delta\mathbf{v}, q) = \mathrm{B}(\mathbf{u}; \delta q) + \sum_E \int_{\Omega_E} \gamma\, \mathrm{d}iv\, \mathbf{u}\, \mathrm{d}iv\, \delta\mathbf{v}\, d\Omega \tag{14}$$

This stabilization uses the following parameters:

$$\gamma = \rho\|\mathbf{u}\|h_K Re_x \tag{15}$$

$$\tau = \frac{h_K}{2\rho\|\mathbf{u}\|} Re_x \tag{16}$$

$$Re_x = min\left(1, \frac{\rho m_K h_K \|\mathbf{u}\|}{4\mu}\right) \tag{17}$$

with h_K a characteristic length of the element and m_K the interpolation degree.

Steady bifurcation analysis using ANM with Petrov-Galerkin FEM stabilization has been studied in [12, 16]. The authors suggest to use constant stabilization parameters within a continuation step. Moreover the problem formulation stays quadratic which is convenient for the perturbation method canvas.

Others stabilization methods are available in the ELMER FlowSolve module and full description may be found in [28]. For example, residual-free bubbles method uses elements where the velocity approximation is augmented by using elementwise bubble functions. And hierarchical interpolation of the second-order elements rises the polynomial order of the velocity approximation.

2.3. Continuation with perturbation method

Pseudo-arc-length path-following technique [20] based on ANM is used to perform the continuation of the steady flow solutions. This has been already successfully implemented for two-dimensional flows in the case of steady incompressible Navier-Stokes equations in [15, 16, 22, 40] and for non-Newtonian fluids in [44]. Recently, three-dimensional flows study has been proposed in [51].

The ANM relies on linearization technique where the unknowns of the nonlinear problem are sought as power series:

$$\mathbf{X}(a) = \mathbf{X}_0 + \sum_{i=1}^{N} a^i \mathbf{X}_i \tag{18}$$

where $\mathbf{X}_0 = (\mathbf{U}_0, \lambda_0)$ is a known regular solution with the mixed vector notation $\mathbf{U}_0 = \{\mathbf{u}_0, p_0\}$, $a \in \mathbb{R}$ is a perturbation parameter and N is the truncation order of this polynomial representation.

Then the perturbation path parameter is chosen as :

$$a = \langle \mathbf{u} - \mathbf{u}_0, \mathbf{u}_1 \rangle + (\lambda - \lambda_0)\, \lambda_1 \tag{19}$$

Introducing Eq.(18) in the nonlinear system Eq.(4) and equating like powers of a, a set of linear algebraic systems is obtained. The same tangent operator is used for each of those linear systems and is defined as:

$$\mathrm{L}_t^0(\bullet) = \mathrm{L}(\bullet) + \mathrm{Q}(\mathbf{U}_0, \bullet) + \mathrm{Q}(\bullet, \mathbf{U}_0) \tag{20}$$

The first order reads:

$$\mathrm{L}_t^0(\mathbf{U}_1) = \lambda_1 \mathbf{F} \tag{21a}$$

$$\langle \mathbf{u}_1, \mathbf{u}_1 \rangle + \lambda_1^2 = 1 \tag{21b}$$

and the recurrence relation for $k \in [2, N]$ is:

$$\mathrm{L}_t^0(\mathbf{U}_k) = \lambda_k \mathbf{F} + \mathbf{FQ}_k \tag{22a}$$

$$\langle \mathbf{u}_k, \mathbf{u}_1 \rangle + \lambda_k \lambda_1 = 0 \tag{22b}$$

using the following definition:

$$\mathbf{FQ}_k = -\sum_{i=1}^{k-1} \mathbf{Q}(\mathbf{U}_i, \mathbf{U}_{k-i}) \tag{23}$$

Once the series $\{\mathbf{U}_i, \lambda_i\}_{i=1,N}$ are computed using Eq.(21)-(22), small part of the steady flow solution branch is approximated continuously with the polynomial representation Eq.(18).

The range of validity of this approximation is defined as :

$$a_{maxpoly} = \left(\eta \frac{\|\mathbf{u}_1\|}{\|\mathbf{u}_N\|} \right)^{1/(N-1)} \tag{24}$$

with η a user-defined tolerance. Other definitions may be used [20].

Moreover, use of Padé approximants is known to increase the range of validity of polynomial representations [31, 56] and has been used in fluid mechanics in [40, 42]. The main idea is to build the rational approximation from the polynomial one. Thus, the rational representation of the velocity unknown is written as:

$$\mathbf{X}_{Padé,N}(a) = \mathbf{X}_0 + \sum_{i=1}^{N-1} \frac{\Delta_{N-1-i}(a)}{\Delta_{N-1}(a)} a^i \mathbf{X}_i \tag{25}$$

The Δ_k are polynomials of degree k. The fractions in Eq.(25) all have the same denominator Δ_{N-1} in order to simplify the root finding of those polynomials. The range of validity for the rational representation is based on the same idea as Eq.(24). The gap between solutions of two consecutive orders is compared to a tolerance parameter by the following expression :

$$\eta_p = \frac{\|\mathbf{U}_{Padé,N}(a_{maxPadé}) - \mathbf{U}_{Padé,N-1}(a_{maxPadé})\|}{\|\mathbf{U}_{Padé,N}(a_{maxPadé})\|} \tag{26}$$

with η_p a required accuracy parameter. The $a_{maxPadé}$ is searched via dichotomy technique. This range of validity $[0, a_{maxPadé}]$ is most often greater than the range of the polynomial approximation. If not, this latter representation is used. It should be noted that not so much more CPU time is needed to perform this computation.

Finally, ANM needs two user parameters, the first one being the truncature order N and the second one is the tolerance parameter η. This latter governs the accuracy of the computed nonlinear solutions. At last, new regular

solution $\mathbf{X}_0 = \mathbf{X}(a_{max})$ is evaluated with either Eq.(18) or Eq.(25) and new continuation step might be performed. We recall that the approximation of the flow solution is continuous in a path-following continuation step. This makes it possible to evaluate steady flow solutions at an exact Reynolds number.

Usually, the ANM nonlinear prediction do not need correction at the end of a step. Nevertheless, for some values of the chosen parameter η the accuracy of the solution obtained with ANM cannot be satisfactory. It means that the computed residual is greater than a given accuracy. In such a case, very efficient and cheaper correctors can be used at the end of step computation to improve the quality of the ANM solution, see for example [14].

2.4. Steady bifurcation detection

Once steady non-linear solution branch has been computed, it is worth determining the critical Reynolds numbers for which a steady bifurcation appears. These singularities often characterize the loss of uniqueness of the flow solutions. In this study only simple symmetry breaking bifurcations of steady flow are considered. Nevertheless, the techniques might be applied for other kind of bifurcation analysis [8, 50].

In [67], the author shows that power series analysis may give information about singularity in its range of validity. Thus, ANM series analysis has been proposed for steady bifurcation detection in [22]. It has been applied to Newtonian incompressible fluid flows in [22, 51] and for non Newtonian fluids in [44]. Befor that, ANM classic indicators were used as steps accumulation or, more efficiently, the so-called 'bifurcation indicator' coupled with Padé approximants pôle detection of the rational representation Eq.(25) [1, 13, 15, 40, 41]. The authors demonstrate in [22] that the pseudo-arc-length continuation is perturbed in the neighborhood of a singularity. Geometric progression appears in the ANM series in the vicinity of a steady bifurcation point. This was a known fact as observed with steps accumulation as shortened range of validity of the polynomial representation, and apparition of Padé approximants pôles in the rational representation of the solution [15, 20, 31, 40]. In [22], the power series $\{\mathbf{X}_i\}$ computed in the singularity vicinity using Eq.(21)-(22) is linear combination of a geometric progression series and a nearly flawless ones :

$$\{\mathbf{X}_i\} = \{\widehat{\mathbf{X}}_i\}_{flawless} + \{\mathbf{X}_i\}_{geometric}, \quad \text{for } i = 1, N \tag{27}$$

Determining a steady bifurcation consists in finding geometric progression into the polynomial representation Eq.(18). This is numerically detected via collinearity condition and relative error test applied to the last terms of the series :

$$\sum_{p=N-3}^{N-2} \left(\frac{|\alpha_p|^{1/(N-p)}}{|\alpha_{n-1}|} - 1 \right)^2 < \epsilon_{gp1} \quad \text{, with} \quad \alpha_p = \frac{\left\langle \mathbf{X}_p, \mathbf{X}_N \right\rangle}{\langle \mathbf{X}_N, \mathbf{X}_N \rangle} \tag{28}$$

$$\sum_{p=N-3}^{N-1} \left(\|\mathbf{X}_p^{\perp}\| / \|\mathbf{X}_p\| \right) < \epsilon_{gp2} \quad \text{, with} \quad \mathbf{X}_p^{\perp} = \mathbf{X}_p - \alpha_p \mathbf{X}_N \tag{29}$$

The following parameters $\epsilon_{gp1} = 10^{-6}$ and $\epsilon_{gp2} = 10^{-3}$ gave satisfactory results for steady bifurcation detection in the present study. It should be noted that a large truncation order, around 30 in the present study, is required to accurately detect the geometric progression. Once the common ratio $1/\alpha_c = 1/\alpha_{N-1}$ and the associated scale factor $\alpha_c^N \mathbf{X}_N$ are determined, a nearly flawless enhanced series is computed :

$$\widehat{\mathbf{X}}_i = \mathbf{X}_i - \alpha_c^{N-i} \mathbf{X}_N, \quad \text{for } i = 1, N \tag{30}$$

In [22], it is demonstrated that α_c is exactly the arc-length distance to the critical point. Thus, the flow solution at this singular point is computed using the flawless enhanced series as:

$$\mathbf{X}_c = \mathbf{X}(a = \alpha_c) = \mathbf{X}_0 + \sum_{i=1}^{N-1} \alpha_c^i \widehat{\mathbf{X}}_i \tag{31}$$

This constitutes an easy method to detect accurately critical solution, because it is done while performing continuation steps with no additional computational effort.

2.5. Branch switching at simple bifurcation point

At simple steady bifurcation point, families of flow solution branches might be found using the classical ANM path-following technique by varying the ANM user parameters N and η [16]. It requires a lot of computations with no possible guess of which parameters lead to specific branch. In [40], the authors proposed a branch switching method for the Navier-Stokes equations resolution. Then, approximations of nonlinear post-bifurcated branches emanating from the singular solution at a simple bifurcation point are computed using an adapted ANM continuation. This branch-switching technique relies on classic bifurcation theory see eg [11, 26, 45, 63]. It has been successfully

coupled with ANM in [8, 9, 22, 69] in the solid mechanics framework. Because the FEM operators in fluid mechanics problems are unsymmetric, an adaptation is proposed in [40, 51]

First, the tangents of the two intersecting branches at a simple bifurcation point are determined. By choosing one of those tangents, it makes it possible to switch from a branch to another. Those non linear post-bifurcated branches are computed with a modified ANM path-following technique as presented in [40]. Hence, the non linear post-bifurcated branches are sought as a power series representation :

$$\mathbf{X}^{bi}(a) = \mathbf{X}_c + \sum_{j=1}^{N} a^j \mathbf{X}_j^{bi} \quad , \quad \text{for } i = 1, 2 \tag{32}$$

The starting point is the singular solution $\mathbf{X}_c$ computed as Eq.(31). Injecting the polynomial representation Eq.(32) in the nonlinear system Eq.(4) a set of linear algebraic systems is obtained equating like powers of a. The tangent operator assembled as in Eq.(20) with the solution at the bifurcation point, denoted L_t^c is singular. A special care is required to compute the post-bifurcated polynomial representation using this operator. Thus, the tangents at the bifurcation point are the first term of this representation.

At a simple bifurcation point, the tangent operator has the following properties:

$$\mathsf{Ker}\left(\mathrm{L}_t^c\right) = \{\mathbf{\Phi}\} \quad , \mathbf{\Phi} \in \Omega \, , \, \langle \mathbf{\Phi}, \mathbf{\Phi} \rangle = 1 \tag{33}$$

$$\mathsf{Ker}\left(\mathrm{L}_t^{c\top}\right) = \{\mathbf{\Psi}\} \quad , \mathbf{\Psi} \in \Omega \, , \, \langle \mathbf{\Psi}, \mathbf{\Phi} \rangle = 1 \tag{34}$$

with Φ the so-called bifurcation mode and the associated left bifurcation mode $\mathbf{\Psi}$. Moreover it exists a unique particular solution vector $\mathbf{W}$ such that:

$$\mathrm{L}_t^c(\mathbf{W}) = \mathbf{F} \quad , \mathbf{W} \in \Omega \, , \, < \mathbf{W}, \mathbf{\Phi} >= 0 \tag{35}$$

Different strategies are available to determine those vectors [45]. A way to compute those vectors is described in the following.

2.5.1. Tangents determination

At first order of the path parameter a the tangents $\mathbf{X}_1^{bi} = \begin{Bmatrix} \mathbf{U}_1^{bi} \\ \lambda_1^{bi} \end{Bmatrix}_{i=1,2}$ are solutions of the following linear system :

$$\mathrm{L}_t^c\left(\mathbf{U}_1^{bi}\right) = \lambda_1 \mathbf{F} \tag{36a}$$

$$\left\langle \mathbf{U}_1^{bi}, \mathbf{U}_1^{bi} \right\rangle + \left(\lambda_1^{bi}\right)^2 = 1 \tag{36b}$$

The singularity at the bifurcation point is treated classically with the help of Lyapunov-Schmidt reduction [39, 40, 45, 68]. To do so, the two tangents are written as linear combination of the bifurcation mode $\mathbf{\Phi} \in \mathsf{Ker}\left(\mathrm{L}_t^c\right)$ and the particular solution $\mathbf{W} \in \mathsf{Ker}^{\perp}\left(\mathrm{L}_t^c\right)$:

$$\mathbf{U}_1^{bi} = \lambda_1^{bi} \mathbf{W} + \eta_1^{bi} \mathbf{\Phi} \tag{37}$$

$\lambda_1^{bi}, \eta_1^{bi} \in \mathbb{R}$ being two scalars to be determined. To do so, we solve the linear system defined at the second order :

$$\mathrm{L}_t^c\left(\mathbf{U}_2^{bi}\right) = \lambda_2 \mathbf{F} - \mathbf{Q}\left(\mathbf{U}_1^{bi}, \mathbf{U}_1^{bi}\right) \tag{38a}$$

$$\left\langle \mathbf{U}_2^{bi}, \mathbf{U}_1^{bi} \right\rangle + \lambda_2^{bi} \lambda_1^{bi} = 0 \tag{38b}$$

Projecting Eq.(38a) on the left bifurcation mode Ψ Eq.(34), leads to the well known Algebraic Bifurcation Equation (ABE) :

$$\left\langle \mathbf{\Psi}, \mathrm{Q}\left(\mathbf{U}_1^{bi}, \mathbf{U}_1^{bi}\right) \right\rangle = 0 \tag{39}$$

Using the Lyapunov-Schmidt reduction Eq.(37), the fllowing quadratic ABE is obtained:

$$a_b \left(\lambda_1^{bi}\right)^2 + b_b \lambda_1^{bi} \eta_1^{bi} + c_b \left(\eta_1^{bi}\right)^2 = 0 \tag{40}$$

This quadratic equation may be easily solved using Eq.(36b), with the following coefficients :

$$a_b = \langle \mathbf{\Psi}, \mathrm{Q}\left(\mathbf{W}, \mathbf{W}\right) \rangle \tag{41a}$$

$$b_b = \left\langle \mathbf{\Psi}, \widetilde{\mathrm{Q}}\left(\mathbf{\Phi}, \mathbf{W}\right) \right\rangle \tag{41b}$$

$$c_b = \langle \mathbf{\Psi}, \mathrm{Q}\left(\mathbf{\Phi}, \mathbf{\Phi}\right) \rangle \tag{41c}$$

In Eq.(41b) the operator $\widetilde{\mathrm{Q}}(a,b)$ is defined according the following expression : $\widetilde{\mathrm{Q}}(a,b) = \mathrm{Q}(a,b) + \mathrm{Q}(b,a)$.

The general case with the restriction to $\eta_1^{bi} \neq 0$ and $a_b \neq 0$, gives:

$$t^{bi} = \frac{\lambda_1^{bi}}{\eta_1^{bi}} = \frac{-b_b + (-1)^i \sqrt{b_b^2 - 4a_b c_b}}{2a_b} \tag{42}$$

Using the norm condition Eq.(36b), and the property $\langle \mathbf{W}, \mathbf{\Phi} \rangle = 0$ one obtains:

$$\eta_1^{bi} = \frac{1}{\sqrt{t^{bi} \langle \mathbf{W}, \mathbf{W} \rangle + \langle \mathbf{\Phi}, \mathbf{\Phi} \rangle + t^{bi}}} \tag{43}$$

λ_1^{bi} is automatically deduced using Eq.(42).

Remark on symmetry breaking bifurcation case

Pitchfork bifurcation occurs at simple bifurcation point if flow symmetry is broken for one of the post-bifurcated family branch. Therefore, the coefficients defined in Eq.(41) are checked numerically. If $a_b = c_b = 0$ and $b_b \neq 0$, it implies that pitchfork bifurcation occurs at the detected critical point [70]. One is easily convinced that the condition $c_b = 0$ is equivalent to the fact that the bifurcation mode $\mathbf{\Phi}$ is solution of the ABE proposed in Eq.(39) and that it is one of the sought tangents [63, 70]. Moreover $a_b = 0$ implies that the other tangent is then collinear to the particular solution $\mathbf{W}$.

Finally in the case of pitchfork bifurcation, the two tangents are:

$$\mathbf{X}_1^{ba} = \begin{Bmatrix} \mathbf{U}_1^{ba} \\ \lambda_1^{ba} \end{Bmatrix} = \begin{Bmatrix} \mathbf{\Phi} \\ 0 \end{Bmatrix} \tag{44}$$

$$\mathbf{X}_1^{bs} = \begin{Bmatrix} \mathbf{U}_1^{bs} \\ \lambda_1^{bs} \end{Bmatrix} = \begin{Bmatrix} \lambda_1^{bs} \mathbf{W} \\ \lambda_1^{bs} \end{Bmatrix} \tag{45}$$

with :

$$\lambda_1^{bs} = \frac{1}{\sqrt{\langle \mathbf{W}, \mathbf{W} \rangle + 1}} \tag{46}$$

2.5.2. Specific vectors computation

In order to evaluate the coefficients in Eq.(41), or directly the pitchfork bifurcation tangents as Eq.(45)-(44), the specific vectors $\mathbf{\Phi}, \mathbf{W}, \mathbf{\Psi}$ are required.

In [15, 40, 69], the bifurcation mode Φ is computed using a classic ANM linear stability analysis technique known as the 'bifurcation indicator'. It should be noted that this requires some computational efforts. Then only $\mathbf{W}$ and Ψ remain to be computed using the definitions (34)-(35) by the mean of augmented systems presented in the following. This is done with no assumption of the symmetry breaking simple bifurcation case.

The recent development proposed in [22, 51] do not use the same Lyapunov-Schmidt reduction. Nevertheless, in the case of the symmetry breaking bifurcations we are able to extract $\mathbf{\Phi}, \mathbf{W}$ from the following vectors. The computation of one of the tangents at the bifurcation point is performed by differentiating the actual flawless enhanced power series Eq.(32), which is evaluated at the critical point:

$$\mathbf{X}_1^{b1} = \left. \frac{d\mathbf{X}(a)}{da} \right|_{a=\alpha_c} = \sum_{i=1}^{N-1} i \, \alpha_c^{i-1} \, \widehat{\mathbf{X}}_i \tag{47}$$

A Gram-Schmidt technique is used in order to extract the orthogonal vector from the last term of the serie :

$$\mathbf{X}_1^{b1\perp} = \mathbf{X}_N - \frac{\left\langle \mathbf{X}_N, \mathbf{X}_1^{b1} \right\rangle}{\left\langle \mathbf{X}_1^{b1}, \mathbf{X}_1^{b1} \right\rangle} \mathbf{X}_1^{b1} \tag{48}$$

In the case of symmetric branch the following equality holds : $\mathbf{\Phi} = \mathbf{X}_1^{b1\perp}$. Contrariwise, if the continuation is on branch of anti-symmetric solutions and returning to symmetric state, then the bifurcation mode is the tangent computed as $\mathbf{\Phi} = \mathbf{X}_1^{b1}$.

Moreover, the particular solution $\mathbf{W}$ might be computed using Eq.(47) or Eq.(48) because of the orthogonality relation. Instead, we use an augmented system that makes it possible to compute $\mathbf{W}$ and is reused in the following in order to circumvent the singularity of the tangent operator. This doesn't require more computational effort. The

vector $\mathbf{W}$ is solution of the linear equation in (35), and orthogonal to the bifurcation mode $\mathbf{\Phi}$ so that the augmented system to be solved is written:

$$\begin{bmatrix} \mathrm{L}_t^c & \mathbf{\Phi} \\ \mathbf{\Phi}^\top & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{W} \\ \kappa \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ 0 \end{Bmatrix} \tag{49}$$

κ acts as a Lagrangian multiplier, that would be as small as possible.

In the case of the Navier-Stokes equations the left mode remains to be computed using either an augmented system [40], or with iterative techniques such as bordering technique as presented in [22], or with an inverse power method [51]. Then, using the same consideration for the left bifurcation mode $\mathbf{\Psi}$ (see Eq.(34)), the augmented system is written:

$$\begin{bmatrix} \mathrm{L}_t^{c\top} & \mathbf{\Phi} \\ \mathbf{\Phi}^\top & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{\Psi} \\ \kappa \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ 1 \end{Bmatrix} \tag{50}$$

It is noticed that this augmented matrix is the transposed of the precedent in Eq.(49). Specific options might be used in linear solvers to reuse the already factorized matrix operator Eq.(49). In that way, only one factorization is required for the computation of the specific vectors and for every linear systems required for the post-bifurcated branches as described in the following.

2.5.3. Post-bifurcated branch continuation

Once the tangents are determined as solutions of Eq.(36), higher order of the power series Eq.(32) is determined for each of the post-bifurcated branches as solution of:

$$\mathrm{L}_t^c\left(\mathbf{U}_k^{bi}\right) = \lambda_k^{bi}\mathbf{F} - \sum_{j=1}^{k-1} \mathbf{Q}\left(\mathbf{U}_j^{bi}, \mathbf{U}_{k-j}^{bi}\right) \tag{51a}$$

$$\left\langle \mathbf{U}_k^{bi}, \mathbf{U}_1^{bi} \right\rangle + \lambda_k^{bi}\lambda_1^{bi} = 0 \tag{51b}$$

The following Lyapunov-Schmidt reduction [22, 40] is used:

$$\mathbf{U}_k^{bi} = \lambda_k^{bi}\mathbf{W} + \eta_k^{bi}\mathbf{\Phi} + \mathbf{V}_k^{bi} \tag{52}$$

Injecting Eq.(52) in Eq.(51), the new vector $\mathbf{V}_k^{bi}$ is solution of:

$$\mathrm{L}_t^c\left(\mathbf{V}_k^{bi}\right) = -\sum_{j=1}^{k-1} \mathbf{Q}\left(\mathbf{U}_j^{bi}, \mathbf{U}_{k-j}^{bi}\right) \tag{53a}$$

$$\left\langle \mathbf{V}_k^{bi}, \mathbf{\Phi} \right\rangle = 0 \tag{53b}$$

This system of equations is exactly the linear augmented system described in Eq.(49), with different right-hand-side vector. It makes it possible to compute $\mathbf{V}_k^{bi}$ in an efficient way because only a new RHS needs to be assembled.

The coefficients $\lambda_k^{bi}, \eta_k^{bi}$ are determined with the same procedure used for the tangents. Injecting the Lyapunov-Schmidt reduction in the order $k+1$ system Eq.(51a), projected on the left bifurcation mode $\mathbf{\Psi}$, and using the condition Eq.(51b), one obtains the following system to solve:

$$\begin{bmatrix} \left\langle \mathbf{\Psi}, \widetilde{\mathrm{Q}}\left(\mathbf{W}, \mathbf{U}_1^{bi}\right)\right\rangle & \left\langle \mathbf{\Psi}, \widetilde{\mathrm{Q}}\left(\mathbf{\Phi}, \mathbf{U}_1^{bi}\right)\right\rangle \\ \left\langle \mathbf{W}, \mathbf{U}_1^{bi}\right\rangle + \lambda_1^{bi} & \left\langle \mathbf{\Phi}, \mathbf{U}_1^{bi}\right\rangle \end{bmatrix} \begin{Bmatrix} \lambda_k^{bi} \\ \eta_k^{bi} \end{Bmatrix} = \begin{Bmatrix} -g_k \\ -\left\langle \mathbf{V}_k^{bi}, \mathbf{U}_1^{bi}\right\rangle \end{Bmatrix} \tag{54}$$

with the coefficient definition:

$$g_2 = \left\langle \mathbf{\Psi}, \mathrm{Q}\left(\mathbf{V}_2^{bi}, \mathbf{U}_1^{bi}\right)\right\rangle \tag{55}$$

and for k>2:

$$g_k = \left\langle \mathbf{\Psi}, \mathrm{Q}\left(\mathbf{V}_k^{bi}, \mathbf{U}_1^{bi}\right)\right\rangle + \sum_{j=2}^{k-1} \left\langle \mathbf{\Psi}, \mathbf{Q}\left(\mathbf{U}_j^{bi}, \mathbf{U}_{k-j+1}^{bi}\right)\right\rangle \tag{56}$$

When a complete power-series is obtained, the range of validity is evaluated. Then, for a chosen tangent, the associated post-bifurcated branches emanating from the bifurcation points are approximated using Eq.(32), with a_{max} or $-a_{max}$ in the two directions. This procedure is repeated at wish for each of the computed tangents. Each new starting point on the post-bifurcated branches is either saved for a restart procedure used directly for new continuation steps.

It should be noted that in the case of simple steady bifurcation, 4 branches are approximated using two tangents, and new regular solutions are saved for next restart.

Remark on symmetry breaking bifurcation case for high order terms

For the symmetric post-bifurcated branch, the tangent $\mathbf{X}_1^{bs}$ is orthogonal to the bifurcation mode, which leads to the following simplification:

$$\lambda_k^{bs} = -\frac{\left\langle \mathbf{V}_k^{bs}, \mathbf{W} \right\rangle}{\langle \mathbf{W}, \mathbf{W} \rangle + 1} \tag{57a}$$

$$\eta_k^{bs} = -\frac{g_k^{bs}}{b_b} \tag{57b}$$

Then, using the tangent of the antisymmetric branch $\mathbf{X}_1^{ba}$, which is exactly the bifurcation mode, the sought coefficients are:

$$\lambda_k^{ba} = -\frac{g_k^{ba}}{b_b} \tag{58a}$$

$$\eta_k^{ba} = 0 \tag{58b}$$

Which leads to the following expression of the high order terms:

$$\mathbf{U}_k^{ba} = \lambda_k^{ba}\mathbf{W} + \mathbf{V}_k^{ba} \tag{59}$$

In that particular case, each high order term of the series is orthogonal to the bifurcation mode according to the pseudo-arc length condition Eq.(51b).

3. ELMER User Defined Solver implementation

In this section, the code implementation of ANM techniques in a so-called ELMER User Defined Solver (UDS) is described. Continuation, steady bifurcation detection and branch switching techniques are implemented in a UDS. None of these features are natively available in ELMER . They rely on new modules for the ANM techniques, specific FEM algebraic operators, or direct linear solvers procedures. Those modules are convenient toolboxes for further ANM developments. Specific algorithms for the presented numerical method, developments of new modules, compilation procedure and data setting for models are described in the next subsections.

ELMER provides a way to solve partial differential equations by the mean of dynamicaly loaded library. Some are available in the original install repositories, but users might want to implement different operators or ways to solve specific problems. For instance, in UDS, one may implement the assembly of FEM operators, new kind of boundary conditions and even implement nonlinear solvers. The operating diagram in Fig.(1) describes the interaction of the UDS and the ELMER core. The ELMER Navier-Stokes "FlowSolver" module [59] is used as a canvas for our UDS.

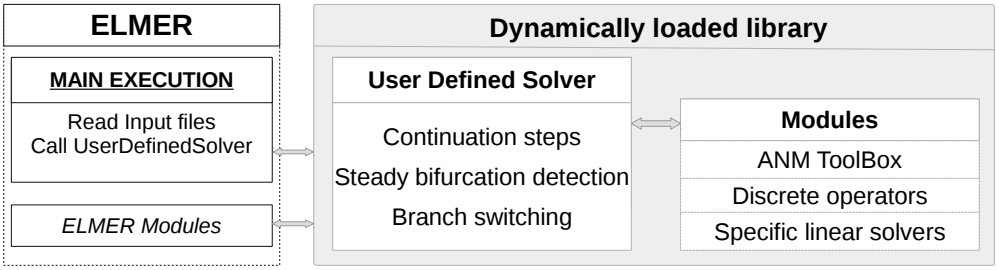


Figure 1: Operating diagram of a User Defined Solver interaction with ELMER solver core.

More details on UDS programming are available in the ELMER documentation. See for example "Basic Programming" in [58], "Basic Programming with Elmer" in [48] and "Partial Differential Equations" in the Elmer Tutorials [49].

3.1. ELMER Work-flow

A complete work-flow using ELMER from scratch is depicted here. First, a mesh is generated according to given geometry of the domain using Gmsh [37] for chosen types of FEM elements. Gmsh functionality of "physical groups" is used to simplify the body id's numbering and the boundary conditions identification in the ELMER Solver Input File (SIF). In order to use this mesh in our computation using ELMER , *ElmerGrid* is used to transform it into a set of 'ELMER format' mesh files. Then a 'Solver Input file' (SIF) is generated from template, or with the help of the

graphical user interface 'ElmerGUI'. In this file, every detail of the model are listed using the correct convention. The simulation is launched with the command 'ElmerSolver'. Those actions are reported as command line in Lst.(1) .

```
# Mesh generation
gmsh -3 GEO-domain.geo -o MSH-domain.msh
# Mesh transfer to ELMER Format
ElmerGrid 14 2 MSH-domain.msh -autoclean
# Launch simulation
ElmerSolver SIF-file.SIF
```

Listing 1: Mesh generation using command line

Finally, when the full simulation is completed, output files are available. In our case, VTK/VTU output files [62] are requested. The visualization is made with Paraview software [3].

3.2. UDS algorithms

A Pseudo arc-length continuation technique is used to compute path of steady flow solutions. When simple bifurcation is detected, every post-bifurcated branches are computed. Algorithms and implementation of the numerical methods are depicted here.

3.2.1. Continuation

The ANM continuation algorithm is detailed in [20, 21, 40]. We describe here, the implementation of the method in the ELMER UDS context. The ELMER "FlowSolver" module has been adapted to our method. In the flowchart diagram Fig.(2), the gray blocks represent ELMER parts, whereas blue blocks represent codes that were implemented in the UDS or in new modules.

Boundary conditions are prescribed by ELMER on the first "FlowSolution" $\mathbf{U}_0^{init}$ including the generic velocity profile $\mathbf{u}_d$. This $\mathbf{U}_0^{init}$ is saved in order to keep this generic profile. This technical solution makes it possible to apply the condition Eq.(3) whenever it is necessary.

The tangent operator L_t^0 is adapted from the ELMER modules to our specific case Eq.(20). In ELMER the assembly of FEM operators is performed directly in the UDS with the local contributions. Thus, a specific routine is created in order to assemble this operator using elementary contributions.

Then, the generic profile is applied to the actual FlowSolution such that the Dirichlet routine provided by ELMER produces the required generic RHS vector $\mathbf{F}$. Then ANM series is computed using the Alg.(1) as described in [20, 40].

Algorithm 1 Compute a serie for pseudo-arc-length continuation using ANM as in [20]

Require: L_t^0, $\mathbf{F}$
  Factorize L_t^0
  Solve $\mathrm{L}_t^0\mathbf{U}^* = \mathbf{F}$, with $\mathbf{U}^* = (\mathbf{u}^*, p^*)$
  Extract $\mathbf{U}_1, \lambda_1$ as:
  $\lambda_1 = 1/\sqrt{\langle \mathbf{u}^*, \mathbf{u}^* \rangle + 1}$
  $\mathbf{U}_1 = \lambda_1 \mathbf{U}^*$
  for k=2 **to** N **do**
    Assemble $\mathbf{FQ}_k$
    Solve $\mathrm{L}_t^0\mathbf{U}^* = \mathbf{FQ}_k$, with $\mathbf{U}^* = (\mathbf{u}^*, p^*)$
    Extract $\mathbf{U}_k, \lambda_k$ as:
    $\lambda_k = -\lambda_1 \frac{\langle \mathbf{u}^*, \mathbf{u}_1 \rangle}{\lambda_1^2 + \langle \mathbf{u}_1, \mathbf{u}_1 \rangle}$
    $\mathbf{U}_k = \mathbf{U}^* + \mathbf{U}_1 \frac{\lambda_k}{\lambda_1}$
  end for
  return $\{U_i, \lambda_i\}_{i=1,N}$

A steady bifurcation might be detected with the geometric progression criterion Eq.(28) and Eq.(29). If no critical point is detected, then a new continuation step is performed using the new regular solution evaluated as Eq.(18). Otherwise, the steady bifurcation procedure is used in order to accurately compute the singular solution and the post-bifurcated branches path.

The Fortran code listing Lst.(2) presents the classic ANM algorithm Alg.(1) . This part represents the block "Compute ANM series" in Fig.(2) . The tangent operator is factorized only for the first resolution. To do so an ELMER option "Linear System Refactorize" is modified on request. User defined specific routine "HMumps_SolveLinearSystem" is used as linear direct solver instead the ELMER SolveLinearSystem. Because MUMPS is not available for serial use of ELMER, this requires classic implementation of the serial MUMPS library in a dedicated module.

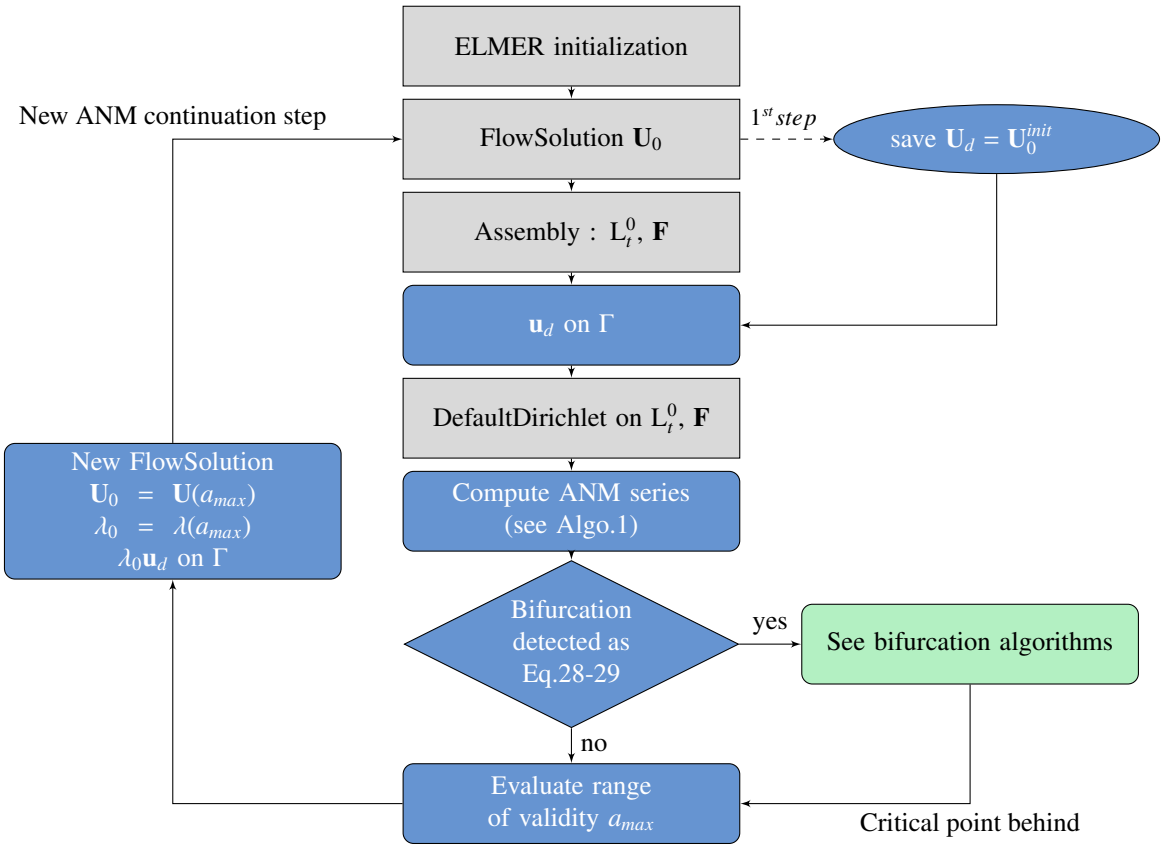


Figure 2: Continuation algorithm of steady state in ELMER as User Defined Solver. In gray, the functionalities provided by ELMER and used as such. In blue the user implementations. In green the bifurcation algorithm. Dashed line represent action performed only once.

```
!------ORDER 1
  ! Factorize A and SOLVE : A X_1 = F
  CALL ListAddLogical( Solver % Values, 'Linear System Refactorize', .TRUE. )
  CALL HMumps_SolveLinearSystem( StiffMatrix, UMan(:,1), ForceVector, Solver, MUMPSFICH)
  CALL ListAddLogical( Solver % Values, 'Linear System Refactorize', .FALSE. )
  ! Extract U_1 Lambda_1 from X_1
  CALL ANMExtractULambda( Lambda, UMan, 1, NoPressure,VMan)
!------LOOP ORDER > 1
  DO k=2,NORDRE
    ! ASSEMBLE : RHS at order k
    CALL ANMPFrhsFQ(FQMan, UMan, k, NSDOFs, FlowPerm, &
                    USAV, GradSAV, FQManTemp, &
                    Uelex, Ueley, Uelez, FlowSolution_Init )
    ! SOLVE : A Xk = FQk
    CALL HMumps_SolveLinearSystem( StiffMatrix, UMan(:,k), ForceVector, Solver, MUMPSFICH)
    ! Extract U_k Lambda_k from X_k
    CALL ANMExtractULambda(Lambda, UMan, k, NoPressure, VMan )
  ENDDO
```

Listing 2: Compute ANM power-series as presented in Alg.(1) [20]

3.2.2. Branch switching at simple steady bifurcation

Once steady bifurcation point is detected, it is worth looking for the post-bifurcated branches. The full scenario is described in Fig.(3) .

The sign of the ratio α_c gives information on the location of the bifurcation point along the branch. If this point is in the opposite direction of the actual continuation direction, then the flawless enhanced series is used to perform the last part of the continuation algorithm [22]. Otherwise, the solution at the critical parameter value is evaluated.

In this paper only the case of simple bifurcation is presented. The deficiency rank might be evaluated using the linear solver features. The branch switching algorithm requires the computations of the vectors $\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{W}$ using Eqs.(47)-(48)-(49)-(50). Then, the tangents $\mathbf{X}_1^{bs}$ and $\mathbf{X}_1^{ba}$ are available with Eqs.(44)-(45). For each of those latter, the high order terms of the corresponding series are computed. Once series defining each branch of the steady flow are computed new regular solutions are evaluated and restart files are created.

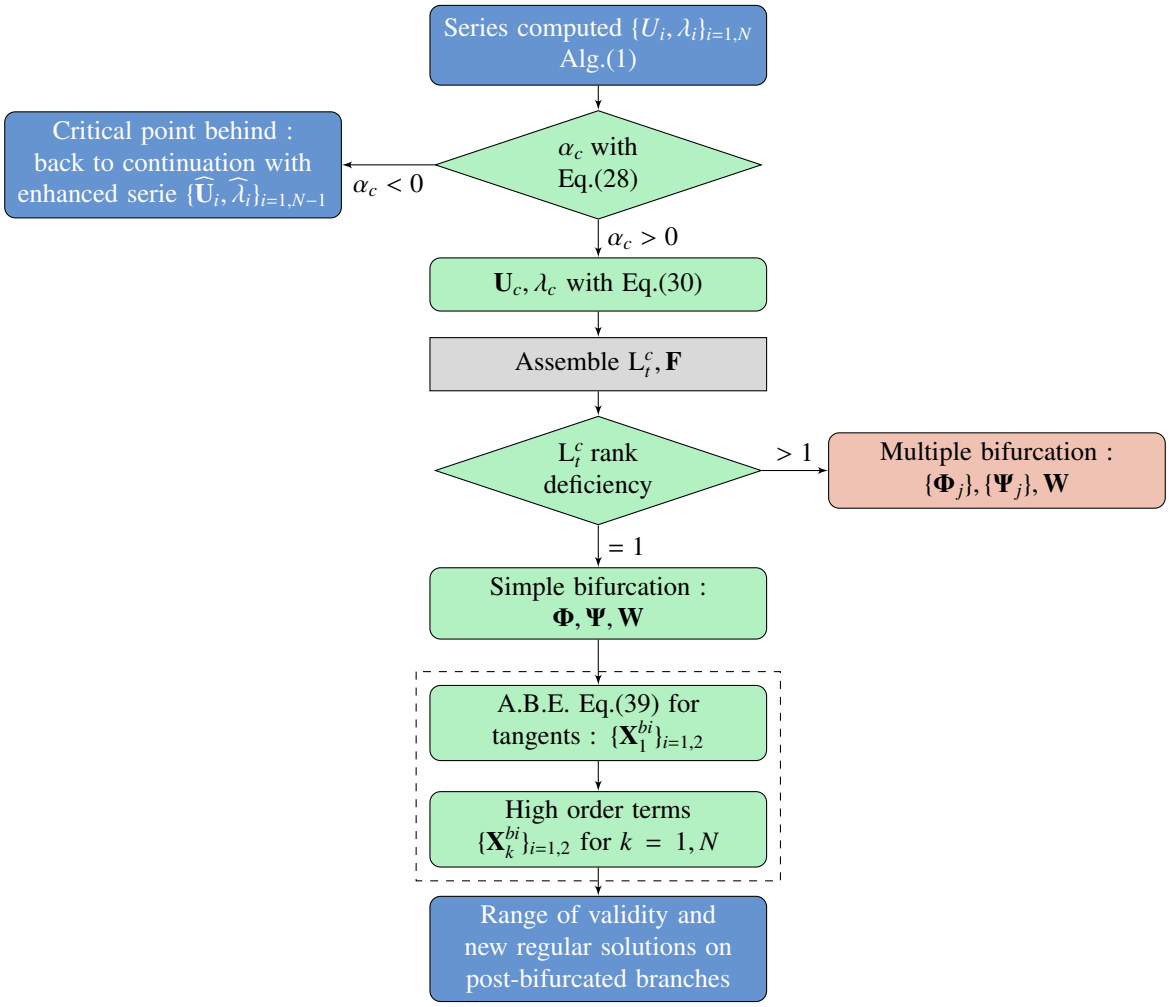


Figure 3: Case of steady bifurcation point detected using the geometric progression criterion. Branch switching is performed using ANM adapted continuation (dashed).

3.3. *New* ELMER *modules*

In order to organize all the routines and functions needed to perform such methods, new ELMER modules are created. We describe here only the main features of those modules.

3.3.1. *ANM ToolBox module*

Some ANM functionalities are grouped into a dedicated module:

- evaluation of the range of validity of series, computation of solutions using the polynomial representation
- Padé approximants: orthogonalization procedure, calculation of coefficients, search of associated poles and evaluation of the range of validity
- Specific Dirichlet procedure

```
MODULE ANMToolBox

  USE SolverUtils
  USE DefUtils

  IMPLICIT NONE

CONTAINS
!...
!... Subroutines
!... Functions
!...
END MODULE ANMToolBox
```

Listing 3: ANM dedicated ELMER module

Specific boundary condition routine

In our implementation, two kinds of boundary conditions exist, imposed velocity profile and no-slip wall law. We decided to create dedicated routine in order to impose specific values only on boundary nodes. The very first FlowSolution allows to keep the generic inlet velocity profile. The routine "CalCondLimCoeff" Lst.(4) is a modification of the Dirichlet ELMER routine. Each node of all boundary elements are concerned by this routine. It might be used at wish for either imposing the generic velocity profile on boundaries ('cond==0') or this latter multiplied by a scalar ('cond==1'). The last feature ('cond==2') ensures null values on boundaries for given vector, only on boundary nodes that are null on the initial solution.

```
  SUBROUTINE CalCondLimCoeff( Vector, Coeff, Cond, VectBCInit,
!...
! DOF LOOP
    DO DOF=1,x % DOFs
! ELEMENT LOOP
      DO i=1,Solver % Mesh % NumberOfBoundaryElements
        Element => GetBoundaryElement(i)
!...
! NODE LOOP
        DO l=1,n
          nb = x % Perm( gInd(l) )
          nb = Offset + x % DOFs * (nb-1) + DOF
          !--------------------------------
          ! CASE 0: put the given vector 'VectBCInit' on boundary of 'Vector'
          IF ( Cond == 0 ) THEN
            Vector(nb) = VectBCInit(nb)
          ! CASE 1: put the given vector multiply by a scalar on boundary
          ELSE IF ( Cond == 1 ) THEN
            IF(VectBCInit(nb).GT.ZERO)THEN
              Vector(nb) = Coeff * VectBCInit(nb)
            ELSE
              Vector(nb) = 0.0_dp
            ENDIF
          ! CASE 2: Impose No-Slip Wall Law
          ELSE IF ( Cond == 2 ) THEN
            !IF Value is 0 on given BC then put 0 on 'Vector'
            !Else don't modify 'Vector'
            IF (VectBCInit(nb).LT.ZERO) THEN
              Vector(nb) = 0.0_dp
            ENDIF
          END IF
          !--------------------------------
        END DO ! Loop NODE
      END DO ! Loop ELEMENT
    END DO ! Loop DOF
  END SUBROUTINE CalCondLimCoeff
```

Listing 4: Specific routine to impose values on boundary using boundary conditions created in the SIF file. Three case are propose for the same CalCondLimCoeff subroutine.

3.3.2. *Discrete operators module*

A module is dedicated to the discrete operators assembly required by the proposed methods.

Tangent operator

The tangent operator is assembled for given flow solution using the following ELMER keywords : 'convect', 'NewtonLinearization', 'Div Discretization' and 'stabilized'. The main element loop is given in Lst.(5) .

```
!------------ Lt(*)=L(*) + Q(*,U0) + Q(U0,*) ---------------
    SUBROUTINE OperatorsLtF( StiffMatrix, ForceVector, USOL,
! ....
        DO t = 1,GetNOFActive() ! LOOP OVER ELEMENTS
! ....
          U(1:nd) = USOL(NSDOFs*FlowPerm(Indexes(1:nd))-3)
! ....
          Density(1:n) = GetReal( Material,'Density' )
          Viscosity(1:n) = GetReal( Material,'Viscosity' )
! ....
          CALL ANMTangentOperatorCompose( MASS,STIFF,FORCE,U,V,W,....
! ....
          CALL DefaultUpdateEquations( STIFF, FORCE )
        END DO
        CALL DefaultFinishBulkAssembly()

        DO t = 1,GetNOFBoundaryElements()
! ....
          CALL DefaultUpdateEquations( STIFF, FORCE )
        END DO
        CALL DefaultFinishAssembly()
    END SUBROUTINE OperatorsLtF
```

Listing 5: Routine for the tangent operator and force vector.

3.3.3. *Non linear operators :* $\mathrm{Q}(\boldsymbol{A},\boldsymbol{B})$ *and* $\boldsymbol{FQ}_k$

The nonlinear convection operator $\mathbf{Q}(\mathbf{A},\mathbf{B})$ as defined in Eq.(8) is used for the ABE and the high order terms computation for the branch switching method. We do not describe the corresponding code for this operator because of the similarity with the nonlinear right hand side $\mathbf{FQ}_k$ Eq.(23). This latter is sum of nonlinear terms using previous series orders solutions. Problem of computational effort arises when calling the routine in the recurrence of linear system resolution. As depicted in Eq.(60), the tensor $\mathrm{Q}(\bullet,\mathbf{U}_1)$ and the vector $\mathrm{Q}(\mathbf{U}_1,\bullet)$ are evaluated on the first order, and then reused in the next orders sequence. This happens for every new solution computed and reused for the next RHS operators.

$$\mathrm{FQ}_2 = -\mathrm{Q}(\mathbf{U}_1,\mathbf{U}_1) \tag{60a}$$

$$\mathrm{FQ}_3 = -\mathrm{Q}\left(\mathbf{U}_2,\boxed{\mathbf{U}_1}\right) - \mathrm{Q}\left(\boxed{\mathbf{U}_1},\mathbf{U}_2\right) \tag{60b}$$

$$\mathrm{FQ}_4 = -\mathrm{Q}\left(\mathbf{U}_3,\boxed{\mathbf{U}_1}\right) - \mathrm{Q}\left(\boxed{\mathbf{U}_2},\boxed{\mathbf{U}_2}\right) - \mathrm{Q}\left(\boxed{\mathbf{U}_1},\mathbf{U}_3\right) \tag{60c}$$

$$\ldots$$

We propose to avoid the evaluation of terms that are already computed. For example, for a given order 'k' the velocity vector and velocity gradient terms are saved for reuse (boxes in Eq.(60)). Vectors and gradients are saved into specific Fortran arrays. The main routine is proposed in Lst.(6) and the local contribution with the saving optimization in Lst.(7) .

```
  SUBROUTINE ANMPFrhsFQ( FQMan, UMan, IO, NSDOFs, FlowPerm, &
!...
    DO t = 1 , GetNOFActive() ! ELEMENT LOOP
!...
      Uelex(1:n) = UMan( NSDOFs*FlowPerm(Indexes(1:nd))-3 , IO-1)
      Ueley(1:n) = UMan( NSDOFs*FlowPerm(Indexes(1:nd))-2 , IO-1)
      Uelez(1:n) = UMan( NSDOFs*FlowPerm(Indexes(1:nd))-1 , IO-1)
!...
      CALL ANMFQelemOPTI( FQManTemp, FQlocal, Density, &
                          Element, n, IO, ElementNodes, &
                          USAV, GradSAV, Element % ElementIndex , NSDOFs, &
                          Uelex, Ueley, Uelez )
      CALL UpdateGlobalForce( FQMan, FQManTemp, n, NSDOFs, FlowPerm(Indexes(1:n)), UElement=Element )
    END DO
    CALL CalCondLimCoeff( FQMan, 0.0_dp, 1,FlowSolution_Init ) ! SPECIFIC BOUNDARY CONDITION
END SUBROUTINE ANMPFrhsFQ
```

Listing 6: Routine for FQ_k assembly of local contributions at ANM order 'IO'.

```
SUBROUTINE ANMFQelemOPTI( FQManTemp, FQlocal, Nodalrho, &
                          Element, n, k , ElementNodes, &
                          USAV , GradSAV, numel , NSDOFs, &
                          Uelex, Ueley, Uelez )
!...
    DO t=1,N_Integ ! GAUSS INTEGRATION LOOP
!...
      ! SAVE ELEMENT INFORMATION FOR ACTUAL SERIES TERM : U and GradU
      ! U_{io-1}
      USAV( numel , t , k-1 , 1 ) = SUM( Basis(1:n) * Uelex(1:n) )
      USAV( numel , t , k-1 , 2 ) = SUM( Basis(1:n) * Ueley(1:n) )
      IF (dim > 2) USAV( numel , t , k-1 , 3) = SUM( Basis(1:n) * Uelez(1:n) )
      ! grad U_{io-1}
      DO j=1,3
        GradSAV( numel , t , k-1 , 1 , j ) = SUM( Uelex(1:n) * dBasisdx(1:n,j) )
        GradSAV( numel , t , k-1 , 2 , j ) = SUM( Ueley(1:n) * dBasisdx(1:n,j) )
        IF ( DIM > 2 ) GradSAV( numel , t , k-1 , 3 , j ) = SUM( Uelez(1:n) * dBasisdx(1:n,j) )
      END DO

      ! COMPUTE FQ_k FOR THE CURRENT ELEMENT
      FQelemtemp = 0.0_dp
      DO r=1, k-1 ! ANM CROSS ORDER SUM LOOP
        DO i = 1 ,dim
          DO j = 1 ,dim
            ! USE THE SAVED TERMS
            U = USAV ( numel , t , r , j )
            GradU = GradSAV( numel , t , k-r , i , j )
            FQelemtemp(i) = FQelemtemp(i) - U * GradU
          END DO
        END DO
      END DO ! END ANM CROSS ORDER SUM LOOP

      ! ASSEMBLY
      DO p = 1 , NBasis
        FQlocal => FQManTemp( c*(p-1)+1 : c*(p-1)+c )
        DO i = 1, c
          FQlocal(i) = FQlocal(i) + s * rho * FQelemtemp(i) * Basis(p)
        END DO
      END DO
    END DO ! END GAUSS INTEGRATION
END SUBROUTINE ANMFQelemOPTI
```

Listing 7: Routine for local contribution required by the FQ_k vector with term computation optimization. This is done for the 'Element' at order 'k' using velocity.

3.3.4. Augmented linear systems

In order to compute the left bifurcation mode $\mathbf{\Psi}$ using Eq.(50) and high order terms in the case of singular tangent operator Eq.(53), augmented system is used [40]. The ELMER Lagrangian multiplier canvas is adapted in order to create and solve the augmented systems. It should be noted that using MUMPS the system or the transpose one, might use the same factorization using 'ICNTL(9)' parameter. In Lst.(8) , the ELMER code of the creation and resolution of augmented system is presented.

```
! CREATE AUGMENTED SYSTEM
k = StiffMatrix % NumberOfRows
!> Mat( 1:k , k+1 ) <- BifMode(1:k)
DO j=1,k
  CALL AddToMatrixElement( CollectionMatrix, j , k+1 , BifMode(j) )
END DO
!> Mat( k+1 , 1:k ) <- BifMode(1:k)
DO j=k,1,-1 !reverse because allocation copy in background see List_EnlargeMatrix
  CALL AddToMatrixElement( CollectionMatrix, k+1 , j , BifMode(j) )
END DO
!> Mat( 1:k , 1:k ) <- Ltc(1:k,1:k)
DO i=k,1,-1
  DO j=StiffMatrix % Rows(i+1)-1,StiffMatrix % Rows(i),-1
    CALL AddToMatrixElement( CollectionMatrix, i, StiffMatrix % Cols(j), StiffMatrix % Values(j) )
  END DO
  !> RHS vector
  CollectionVector(i) = ForceVector(i)
END DO
!> Mat( k+1 , k+1 ) <- 0
CALL AddToMatrixElement( CollectionMatrix, k+1 , k+1 , 0.0_dp )

!> RHS last value
CollectionVector(k+1) = CondVal

! SOLVE AUGMENTED SYSTEM
!> Factorize if required by user and SOLVE
CALL HMumps_SolveLinearSystem( CollectionMatrix, CollectionSolution, CollectionVector, &
                               Solver , MUMPSFICH )
! Separate the solution from CollectionSolution
Solution = 0.0_dp
Solution(1:k) = CollectionSolution(1:k)
```

```
restv = CollectionSolution(k+1)
```

Listing 8: Assembly of augmented system. Adapted from ELMER (FETI) for Mumps unsymetric serial multi-threaded resolution.

3.3.5. Linear solver module : MUMPS

ELMER comes with the *UMFpack* linear direct solver. Despite several tests with the help of ELMER developers, we were not able to perform computations with memory greater thant 4GB. This is the reason why we used Mumps V5.0.0 [2] as the direct linear solver. Moreover, we compiled Mumps with the OpenBLAS2.14 linear algebra library [72] in order to get multi-threaded operations.

Then, ELMER has to be compiled with options in order to work with Mumps. This library is used in ELMER in order to solve domain decomposition problems. Thus, we implement a serial implementation of the linear solver. We do not describe this part in this paper as this is a classic implementation of MUMPS.

3.4. UDS *for ANM continuation*

We describe the User Defined Solver that perform the ANM continuation procedure in Lst.(9) . In order to use this UDS , it needs to be compiled. First the modules are prepared, then the UDS is linked and compiled with the required libraries. An example of compilation script is given in Lst.(10) .

```
#!/bin/bash
# Usage : ./script.sh UDS.f90
sep=" "
extsrc=".f90"
extobj=".o"
extlib=".so"

filetocompile=$1

module="MOD-HomeMadeDirectSolvers"
module+=" MOD-DiscreteOperators"
module+=" MOD-ELMERModResOut"
module+=" MOD-ANMToolBox"

MODULESf90=""
MODULESSO=""
for tab in $module
do
  MODULESf90+=$tab$extsrc$sep
  MODULESSO+=$tab$extobj$sep
done

elmerf90 -c $MODULESf90
elmerf90 $MODULESSO $filetocompile -o ${filetocompile%$extsrc}$extlib
```

Listing 10: UDS compilation with modules

3.4.1. SIF example file

An example of Solver Input File (SIF) is given in Lst.(11) . The velocity profile is prescribed on the inlet boundaries using User Defined Function [49].

```
! Solver Input File

! Repositories
$mesh="MSH_SE_3D_E3_Ai6"
$Res="ResultFolder"

! ANM Parameters
$ANMNBSTEPS=20
$ANMORDER=30
$ANMPREC=1e-20

! User Defined fonction. Here for the inlet profile
$UDFINLET="UDF_VelocProfile_RectDuct3D.so"

Header
  CHECK KEYWORDS Warn
  Mesh DB "." "$mesh$"
  Include Path ""
  Results Directory "$Res$"
End

Solver 1
  Equation = Navier-Stokes
  Variable = Flow Solution[Velocity:3 Pressure:1] ! 2D/3D
  Procedure = "UDS_ANMPathFollowing" "ANMPathFollowing" ! "UDS.so" "SubRoutineName"
  ANM Step = $ANMNBSTEPS$
  ANM Order = $ANMORDER$
  ANM Tolerance = $ANMPREC$
```

```
   SUBROUTINE ANMPathFollowing( Model,Solver,dt,TransientSimulation)
!------------------------
    USE NavierStokes
    USE Adaptive
    USE DefUtils
    USE FreeSurface
!------------------------
!   USER MODULES
    USE ANMToolBox
    USE DiscreteOperators
    USE HomeMadeSolvers
    USE ELMERModResOut
!------------------------
    IMPLICIT NONE
!...
    REAL(KIND=dp) , ALLOCATABLE :: UMan(:,:), Lambda(:)
!...
    ! PARAMETERS RED FROM SIF FILE USING SOLVER.KEYWORDS FILE DEFINITION
    ANMStep =    GetInteger( Solver%Values , 'ANM Step', arg )       ! NBSTEPs
    ANMOrd  =    GetInteger( Solver%Values , 'ANM Order', arg )      ! ANM ORDER
    ANMTol  = GetConstReal( Solver%Values , 'ANM Tolerance', arg ) ! PREC FOR Amax
!...
    ALLOCATE( UMan( StiffMatrix % NumberOfRows , ANMOrd ), Lambda( ANMOrd ),   &
!...
! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    FlowSolution_Init = FlowSolution  ! Saved for generic dirichlet
    ! Steady continuation step loop
    DO Step=1, ANMStep
      ! Assemble Lt
!...
       CALL CalCondLimCoeff( FlowSolution , 0._dp, 0, FlowSolution_Init )
       CALL DefaultDirichletBCs()
!...
      ! COMPUTE SERIES : {UMan} {Lambda}
!...
      ! Evaluate a RANGE OF VALIDITY
      amax = ANMseriesRoV( UMan, ANMOrd, ANMTol )
      ! NEW SOLUTION
      U0 = FlowSolution
      CALL ANMSolPol( U0, lambda0, UMan, Lambda, ANMOrd, amax ) ! X0<-X0+a*X1+a^2*X2+...
      ! Ensure Lambda0 x ud on boundary
      CALL CalCondLimCoeff( U0, lambda0, 1, FlowSolution_Init )
      FlowSolution = U0
      FlowSol % Values => FlowSolution
      ! OUTPUT FILES
      CALL ResultOutputSolver( Model,Solver,dt,Transient )
!...
    END DO ! END ANMStep loop
! - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
!...
  END SUBROUTINE ANMPathFollowing
```

Listing 9: UDS-PathFollowingOnly.f90 : extraction of code of the path-following UDS solver

```
  Optimize Bandwidth = True
  Stabilize = True
  Div Discretization = True
  Bubbles = False
  Linear System Solver = Direct
  Exec Solver = Always
End

Boundary Condition 1
  Name = "Inlet"
  Target Boundaries(1) = 1
! Velocity 1 = Variable Coordinate 2; Real MATC "1.0 - (tx(0)/5.0)^2" ! 2D parabola
  Velocity 1 = Variable Coordinate 2 ! 3D UDF
  procedure "$UDFINLET$" "VProfileRectDuct" ! 3D UDF
  Velocity 2 = 0.0
  Velocity 3 = 0.0
End

Boundary Condition 2
  Target Boundaries(2) = 3 4
  Name = "Walls"
  Velocity 1 = 0.0
  Velocity 2 = 0.0
  Velocity 3 = 0.0
End

Boundary Condition 3
  Target Boundaries(1) = 2
  Name = "Outlet"
  Pressure = 0.0
End
```

Listing 11: SIF example

4. Numerical validations

In order to validate the implementation of the ANM methods in ELMER , bifurcation analysis is performed for the case of symmetry breaking steady bifurcation. The critical Reynolds numbers are investigated for three well documented test cases : the two-dimensional sudden expansion, the sudden expanded part channel with a downstream contraction, and finally the three-dimensional sudden expansion. Results obtained with the ANM methods in ELMER are compared to the bifurcation informations of the literature. ANM continuation of steady flow solution is performed. Then the geometric progression criterion is checked during the continuation. Possibly, steady bifurcations are detected and critical Reynolds numbers are reported. All the computation presented in this article were performed on DELL Precision T7500 using TwoIntel(R) Xeon(R) X5677@3.47GHz (4-cores), with 96GB of memory.

4.1. Two dimensional cases

The two-dimensional test cases are depicted in Fig.(4) . Using the definition $E = H/h$, both cases have geometric aspect ratio $E = 3$. Moreover, using the definition $A = L/3h$ for the expansion contraction case, ratio $A = 8/3$ is chosen.

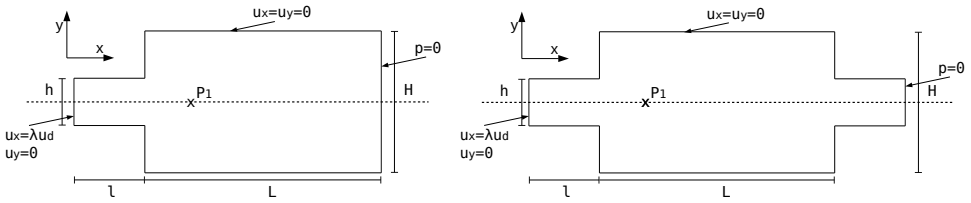


Figure 4: Channels with prescribed velocity profile $\mathbf{u}_d$ on the inlet, no-slip wall law and null pressure on outlet. Left : $h = 10$, $H = 3h$, $l = 3h$, $L = 30h$. Right : $h = 10, H = 30, l = l_{in} = l_{out} = 3h, L = 8h$. A velocity probe is placed near to the expansion on the symmetric axis.

4.1.1. Sudden expansion

The bifurcation analysis of this case is well documented either experimentally [4, 6, 17, 29, 32, 64] or numerically [7, 32, 61] and even using the ANM abilities in [1, 40, 41, 51]. We report in Tab.(1) , the spatial discretization informations. The memory space and factorization CPU time (LU) associated with the tangent operator are reported.

Spatial discretization convergence study is performed using two kinds of mixed finite elements Q4/4C and Q9/9C [73]. In [36] the authors conclude that the Q8/8C element type can't give good results with the ELMER 'FlowSolver'. Hence, validation cases using this element type are not performed in this study. The critical Reynolds numbers detected using ELMER with the ANM are depicted in Fig.(5) . The behavior is classic in FEM, the quadratic element type gives faster convergence results with less degrees of freedom (DoF). Nevertheless, both element types give satisfactory critical values compared to some critical values extracted from the literature (see Tab.(2)).

MESH	EL	Inlet	body	Nodes	DoF	HBW	Mem.	LU
Q4-24k	Q4/4C	16 x 48	48 x 480	24385	73155	84	160 Mo	0.5s
Q4-96k	Q4/4C	32 x 96	96 x 960	96385	289155	164	780 Mo	3s
Q4-383k	Q4/4C	64 x 192	192 x 1920	383233	1149699	324	3800 Mo	20s
Q9-24k	Q9/9C	8 x 24	24 x 240	24385	73155	170	175 Mo	0.5s
Q9-96k	Q9/9C	16 x 48	48 x 480	96385	289155	330	820 Mo	3s
Q9-383k	Q9/9C	32 x 96	96 x 960	383233	1149699	650	3900 Mo	20s

Table 1: Spatial discretization. HalfBandWitdh (HBW), Memory and CPU factorization times required by MUMPS and OpenBlas with 8 threads.

Référence	Re_c	
Fearn et al. 1990 [32]	80.85	BA
Battaglia et al. 1997 [7]	79.5	BA
Schreck and Schäfer 2000 [61]	81.45	DNS
Allery et al. 2004 [1]	80.5	ANM BA (I)
Guevel et al. 2011 [41]	80.46	ANM BA (I+H)
Medale and Cochelin 2015 [51]	82.08	ANM BA (GP)

Table 2: Critical Reynolds numbers for the first steady pitchfork bifurcation in a sudden expansion of ratio $E = 3$. BA stands for bifurcation analysis, DNS for direct numerical simulation, (I) for bifurcation indicator, (H) for homotopy, and (GP) for the geometric progression analysis.

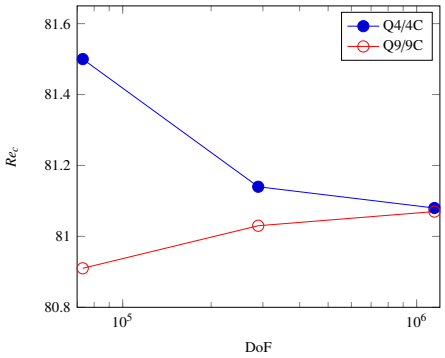


Figure 5: Critical Rynolds numbers plot against number of degrees of freedom (DoF) for two kind of finite element in ELMER . ANM parameters $N = 30, \eta = 1e-14$.

4.1.2. Sudden expanded part

This case is studied experimentally and numerically in [40, 52, 53, 55]. In [40], bifurcation analysis is performed using ANM. Ratio $A = 8/3$ leads to twos kind of singularity. Two primary pitchfork bifurcations are detected on the fundamental branch, and limit point exists on the post-bifurcated branches. The associated critical Reynolds numbers are reported in Tab.(3) , and bifurcation diagram is proposed in Fig.(6) . In the diagram, the ANM steps using the Lyapunov-Schmidt reduction are plot in red. Using ANM in ELMER , we are able to accurately detect the steady bifurcation point and the limit point. The ANM parameters are $N = 30$,$\eta = 10^{-30}$.

	Re_{cb1}	Re_{cb2}	Re_{clp}
Mizushima et al. 1996 [52]	41.1	X	111.0
Guevel et al. 2011 [40]	41.8	106.0	112.0
study	41.7	106.5	111.2

Table 3: Comparison of the critical Reynolds numbers from the literature for the two bifurcation $B1$, $B2$ and the limit point LP Fig.(6) .

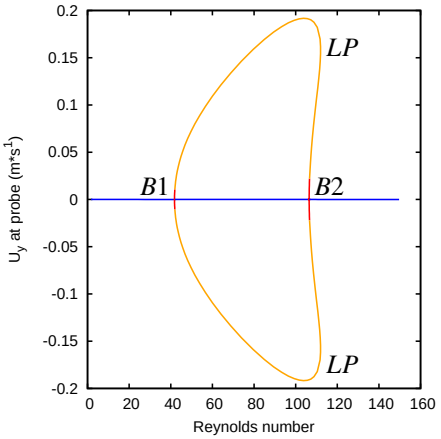


Figure 6: Bifurcation diagram for the sudden expansion part with $E = 3$ A=8/3.

4.2. Three-dimensional sudden expansion

This test case is well documented in [4, 6, 29, 32]. Bifurcation analysis using the aspect ratio $E = 3$ are proposed numerically in [18, 19, 61, 66] and more recently in [51] using ANM features. The fluid domain and the boundary conditions are reported in Fig.(7) . Established velocity profile is prescribed on the inlet. The length L is chosen long enough in order to capture the steady bifurcation phenomenon.

Features of mesh with different geometric ratios $A_i = W/h$ are presented in Tab.(4) . Linear interpolation H8/8C finite element are used. Those cases allow direct comparisons with critical values from the literature.

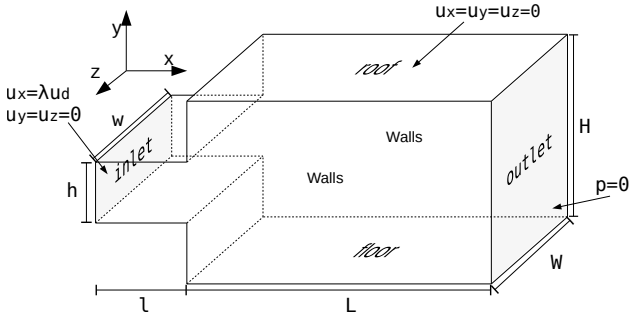


Figure 7: Sudden expansion. $h = 10, H = 3h, l = 3h, L = 30h, W = A_i h$

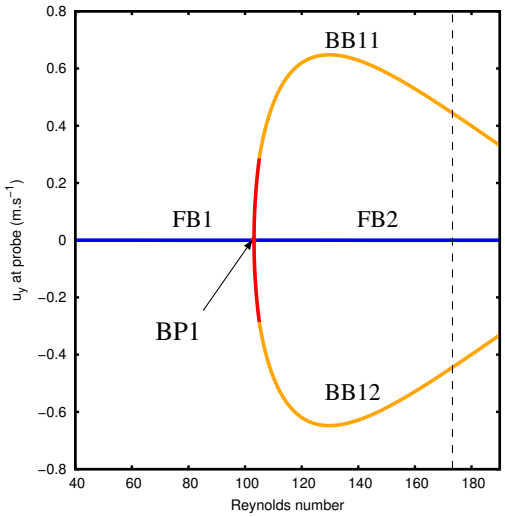
The critical Reynolds reported in the literature Tab.(8a) are correctly confirmed in ELMER with the ANM. Illustration of the three-dimensional flow is depicted in Tab.(8c) . Two Reynolds number are considered, the critical value and higher value where three solutions exists Fig.(8b) . The streamlines of the flow solutions allow us to identify recirculation zones and vortex lines. First, the steady flow solution shows similarity with two-dimensional sudden expansion solutions [41].

Mesh	Inlet	Body	Nodes	DoF	HBW	Mem.	LU
H8-*Ai*3	8x24x24	24x24x240	156025	624100	2162	17Go	550s (v4)
H8-*Ai*3.5	8x28x24	24x28x240	180989	723956	2220	22Go	860s (v4)
H8-*Ai*6	8x48x24	24x48x240	305809	1223236	2667	48Go	3000s (v4)
H8-*A*8	8x70x24	24x70x240	486989	1947956	3785	89Go	1500s
H8-*Ai*10	8x80x24	24x80x240	505521	2022084	4267	96Go	1800s

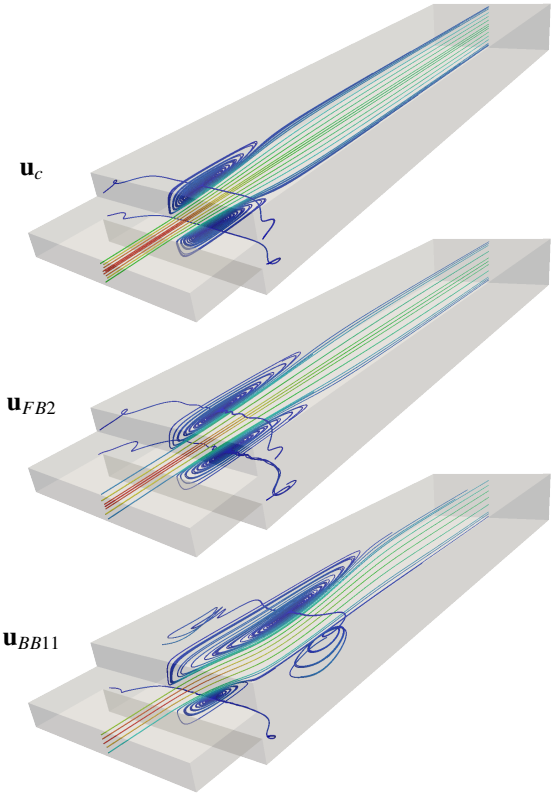
Table 4: Meshes for the case $E = 3, l = 3h, L = 30h$ and 5 ratios $A_i = W/h$. Half-Bandiwdth (HBW), memory space and CPU factorisation time required by MUMPS are given. (v4) stands for mono threaded computation using the Mumps V4 library, otherwise the Mumps V5 and OpenBlas libraries are used on 8 threads.

A_i	ref	Re_c	Re_c study
3	Tsui and Wang 2008 [66]	171	169
3.5	Chiang et al. 2000 [18]	145	149
6	Schreck and Schäfer 2000 [61]	113	112
10	Médale and Cochelin 2015 [51]	98	98

(a) Critical Reynolds numbers for the first primary steady bifurcation using 4 different values of ratio $A_i = W/h$



(b) Bifurcation diagram for $A_i = 8$. FB denote a fundamental branch, BP1 the first primary bifurcation point and BB11, BB12 the post-critical un-symmetric nonlinear branches.



(c) Streamlines for three-dimensional steady flows, $A_i = 8$.

Figure 8: Bifurcation analysis for the three-dimensional sudden expansion with rectangular inlet.

In order to validate the topology of the flow, some characteristics of the topology are determined using skin friction lines [30]. We depict skin friction lines on the floor, roof and lateral wall in Fig.(9) for un-symmetric flow solution. The kinds of singularity and their positions are in good agreement with the pattern depicted in [18].

5. Conclusions

Numerical bifurcation analysis are proposed in the multiphysical software ELMER as new features for the Navier-Stokes equations. Continuation of steady flow solutions, bifurcation detection and branch switching techniques are implemented using the Asymptotic Numerical Method. Power series analysis makes it possible to accurately detect bifurcation points. Specific continuation technique is used in the case of the path following of post-critical branch of steady flow solutions.

Those techniques are implemented as new modules and User Defined Solver in ELMER . The algorithm and the dedicated programming code of new features are described. Large scale problems are now possibly studied, either only for steady flow continuation, or for a detailed bifurcation analysis.

Validation of this implementation is performed on internal flows test cases. The critical Reynolds numbers for bifurcation and limit points are in good agreement with the literature. Three-dimensional sudden expansions critical values and flow topologies are correctly reproduced. Implementation of MUMPS as serial direct linear solver using the multi-thread features of OpenBlas, allows us to perform three-dimensional flow simulation up to 2 Million degrees

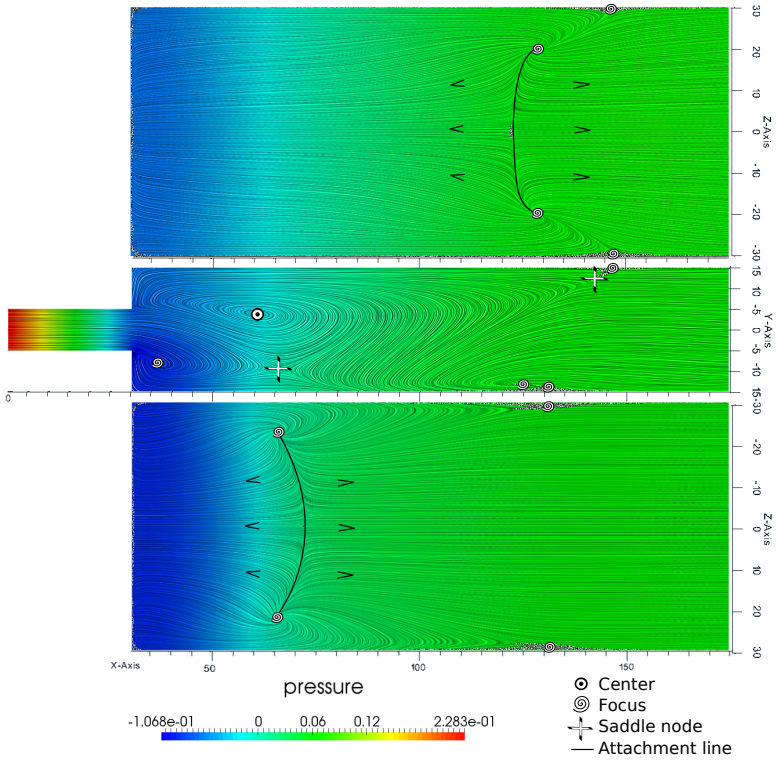


Figure 9: Skin friction for the post-critical BB12 branch of the sudden expansion with $A_i = 6$ at $Re = 135$. Made with ParaView with Line Integral Convolution.

of freedom in the present study. The stabilized finite element method used in ELMER does not perturb the bifurcation phenomenon.

We now have generic tool for large scale bifurcation analysis for the three-dimensional Navier-Stokes equations. Eigenvalue solver might be used in ELMER in order to determine the flow solution's stability. More techniques based on ANM may now be implemented, like parametric analysis of steady bifurcation using homotopy [41], or Non-Newtonian fluids [44], Hopf bifurcation detection [10, 15, 38, 43], transient nonlinear solvers [42, 57].

[1] Allery, C., Cadou, J. M., HamdouniAziz, & Razafindralandy, D. (2004). Application of the asymptotic numerical method to the coanda effect study. *Revue Européenne des Éléments*, *13*(1-2), 57–77.

[2] Amestoy, P. R., Duff, I. S., L'Excellent, J.-Y., & Koster, J. (2001). A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, *23*(1), 15–41.

[3] Ayachit, U. (2015). *The ParaView Guide: A Parallel Visualization Application* (4.3 ed.). Kitware, Incorporated.

[4] Baloch, A., Townsend, P., & Webster, M. (1995). On two- and three-dimensional expansion flows. *Computers & Fluids*, *24*(8), 863 – 882.

[5] Bathe, K. (2006). *Finite Element Procedures*. Prentice Hall.

[6] Battaglia, F. & Papadopoulos, G. (2005). Bifurcation characteristics of flows in rectangular sudden expansion channels. *Journal of Fluids Engineering*, *128*(4), 671–679.

[7] Battaglia, F., Tavener, S. J., Kulkarni, A. K., & Merkle, C. L. (1997). Bifurcation of low reynolds number flows in symmetric channels. *AIAA journal*, *35*(1), 99–105.

[8] Bekhoucha, F., Rechak, F., Duigou, L., & Cadou, J. M. (2015). Branch switching at hopf bifurcation analysis via asymptotic numerical method: Application to nonlinear free vibrations of rotating beams. *Communications in Nonlinear Science and Numerical Simulation*, *22*(1–3), 716 – 730.

[9] Boutyour, E. H., Zahrouni, H., Potier-Ferry, M., & Boudi, M. (2004). Bifurcation points and bifurcated branches by an asymptotic numerical method and padé approximants. *International Journal for Numerical Methods in Engineering*, *60*(12), 1987–2012.

[10] Brezillon, A., Girault, G., & Cadou, J. M. (2010). A numerical algorithm coupling a bifurcating indicator and a direct method for the computation of Hopf bifurcation points in fluid mechanics. *Computers & Fluids*, *39*(7), 1226 – 1240.

[11] Brezzi, F., Rappaz, J., & Raviart, P.-A. (1982). Finite dimensional approximation of nonlinear problems. part iii: Simple bifurcation points. *Numerische Mathematik*, *38*(1), 1–30.

[12] Cadou, J. M. (1997). *Méthode Asymptotique Numérique pour le calcul des branches solutions et des instabilités dans les fluides et pour les problèmes d'intéraction fluide-structure.* PhD thesis, Université de Metz.

[13] Cadou, J. M., Guevel, Y., & Girault, G. (2012). Numerical tools for the stability analysis of 2d flows: application to the two-and four-sided lid-driven cavity. *Fluid Dynamics Research*, *44*(3), 031403.

[14] Cadou, J. M. & Potier-Ferry, M. (2010). A solver combining reduced basis and convergence acceleration with applications to non-linear elasticity. *International Journal for Numerical Methods in Biomedical Engineering*, *26*(12), 1604–1617.

[15] Cadou, J. M., Potier-Ferry, M., & Cochelin, B. (2006). A numerical method for the computation of bifurcation points in fluid mechanics. *European Journal of Mechanics - B/Fluids*, *25*(2), 234 – 254.

[16] Cadou, J. M., Potier-Ferry, M., Cochelin, B., & Damil, N. (2001). ANM for stationary Navier–Stokes equations and with Petrov–Galerkin formulation. *International Journal for Numerical Methods in Engineering*, *50*(4), 825–845.

[17] Cherdron, W., Durst, F., & Whitelaw, J. H. (1978). Asymmetric flows and instabilities in symmetric ducts with sudden expansions. *Journal of Fluid Mechanics*, *84*, 13–31.
[18] Chiang, T., Sheu, T. W., & Wang, S. (2000). Side wall effects on the structure of laminar flow over a plane-symmetric sudden expansion. *Computers & Fluids*, *29*(5), 467 – 492.
[19] Chiang, T. P., Sheu, T. W., Hwang, R. R., & Sau, A. (2002). Spanwise bifurcation in plane-symmetric sudden-expansion flows. *Phys. Rev. E*, *65*(1), 016306.
[20] Cochelin, B. (1994). A path-following technique via an asymptotic-numerical method. *Computers and Structures*, *53*(5), 1181 – 1192.
[21] Cochelin, B., Damil, N., & Potier-Ferry, M. (2007). *Méthode asymptotique numérique*. Collection Méthodes numériques. Hermès Science publications.
[22] Cochelin, B. & Medale, M. (2013). Power series analysis as a major breakthrough to improve the efficiency of Asymptotic Numerical Method in the vicinity of bifurcations. *Journal of Computational Physics*, *236*, 594 – 607.
[23] CSC - IT Center for Science (1995). ELMER - Finite Element Solver for Multiphysical Problems. *www.csc.fi/elmer*.
[24] Dhooge, A., Govaerts, W., & Kuznetsov, Y. A. (2003). Matcont: a matlab package for numerical bifurcation analysis of odes. *ACM Transactions on Mathematical Software (TOMS)*, *29*(2), 141–164.
[25] Dijkstra, H. A., Wubs, F. W., Cliffe, A. K., Doedel, E., Dragomirescu, I. F., Eckhardt, B., Gelfgat, A. Y., Hazel, A. L., Lucarini, V., Salinger, A. G., Phipps, E. T., Sanchez-Umbria, J., Schuttelaars, H., Tuckerman, L. S., & Thiele, U. (2014). Numerical bifurcation methods and their application to fluid dynamics: Analysis beyond simulation. *Communications in Computational Physics*, *15*, 1–45.
[26] Doedel, E., Keller, H. B., & Kernevez, J. P. (1991). Numerical analysis and control of bifurcation problems (i): Bifurcation in finite dimensions. *International Journal of Bifurcation and Chaos*, *01*(03), 493–520.
[27] Doedel, E. J. & Wang, X. (1994). *AUTO 94, Software for Continuation and Bifurcation Problems in Ordinary Differential Equations*. Caltech, Dept of Applied Mathematics.
[28] Donea, J. & Huerta, A. (2003). *Finite Element Methods for Flow Problems*. Finite Element Methods for Flow Problems. John Wiley & Sons.
[29] Durst, F., Melling, A., & Whitelaw, J. H. (1974). Low reynolds number flow over a plane symmetric sudden expansion. *Journal of Fluid Mechanics*, *64*, 111–128.
[30] Délery, J. (2013). *Skin Friction Lines Pattern and Critical Points*, (pp. 1–26). John Wiley & Sons, Inc.
[31] Elhage-Hussein, A., Potier-Ferry, M., & Damil, N. (2000). A numerical continuation method based on Padé approximants. *International Journal of Solids and Structures*, *37*(46–47), 6981 – 7001.
[32] Fearn, R. M., Mullin, T., & Cliffe, K. A. (1990). Nonlinear flow phenomena in a symmetric sudden expansion. *Journal of Fluid Mechanics*, *211*, 595–608.
[33] Franca, L. P. & Frey, S. L. (1992). Stabilized finite element methods: Ii. the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, *99*(2–3), 209 – 233.
[34] Franca, L. P., Hauke, G., & Masud, A. (2006). Revisiting stabilized finite element methods for the advective–diffusive equation. *Computer Methods in Applied Mechanics and Engineering*, *195*(13–16), 1560 – 1572. A Tribute to Thomas J.R. Hughes on the Occasion of his 60th Birthday.
[35] Franca, L. P., Hughes, T. J. R., & Stenberg, R. (1993). Stabilized finite element methods. In M. D. Gunzburger & R. A. Nicolaides (Eds.), *Incompressible Computational Fluid Dynamics* (pp. 87–108). Cambridge University Press. Cambridge Books Online.
[36] Gagliardini, O., Zwinger, T., Gillet-Chaulet, F., Durand, G., Favier, L., de Fleurian, B., Greve, R., Malinen, M., Martín, C., Råback, P., Ruokolainen, J., Sacchettini, M., Schäfer, M., Seddik, H., & Thies, J. (2013). Capabilities and performance of elmer/ice, a new-generation ice sheet model. *Geoscientific Model Development*, *6*(4), 1299–1318.
[37] Geuzaine, C. & Remacle, J.-F. (2009). Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, *79*(11), 1309–1331.
[38] Girault, G., Guevel, Y., & Cadou, J. M. (2012). An algorithm for the computation of multiple Hopf bifurcation points based on Padé approximants. *International Journal for Numerical Methods in Fluids*, *68*(9), 1189–1206.
[39] Golubitsky, M. & Schaeffer, D. G. (1984). *Singularities and Groups in Bifurcation Theory, vol. I*. New York: Springer-Verlag.
[40] Guevel, Y., Boutyour, H., & Cadou, J. M. (2011). Automatic detection and branch switching methods for steady bifurcation in fluid mechanics. *Journal of Computational Physics*, *230*(9), 3614 – 3629.
[41] Guevel, Y., Girault, G., & Cadou, J. M. (2014). Parametric analysis of steady bifurcations in 2d incompressible viscous flow with high order algorithm. *Computers & Fluids*, *100*, 185 – 195.
[42] Guevel, Y., Girault, G., & Cadou, J. M. (2015). Numerical comparisons of high-order nonlinear solvers for the transient navier–stokes equations based on homotopy and perturbation techniques. *Journal of Computational and Applied Mathematics*, *289*, 356 – 370. Sixth International Conference on Advanced Computational Methods in Engineering (ACOMEN 2014).
[43] Heyman, J., Girault, G., Guevel, Y., Allery, C., Hamdouni, A., & Cadou, J. M. (2013). Computation of Hopf bifurcations coupling reduced order models and the Asymptotic Numerical Method. *Computers & Fluids*, *76*, 73 – 85.
[44] Jawadi, A., Boutyour, H., & Cadou, J. M. (2013). Asymptotic Numerical Method for steady flow of power-law fluids. *Journal of Non-Newtonian Fluid Mechanics*, *202*, 22 – 31.
[45] Keller, H. B. (1977). Numerical solution of bifurcation and nonlinear eigenvalue problems. In Rabinowitz, P. H. (Ed.), *Applications of Bifurcation Theory*, (pp. 359–384). Academic Press.
[46] Kłosiewicz, P., Broeckhove, J., & Vanroose, W. (2009). Using pseudo-arclength continuation to trace the resonances of the schrödinger equation. *Computer Physics Communications*, *180*(4), 545 – 548. Special issue based on the Conference on Computational Physics 2008CCP 2008.
[47] Logg, A., Ølgaard, K., Rognes, M., Wells, G., Jansson, J., Kirby, R., Knepley, M., Lindbo, D., & Skavhaug, O. (2011). The FEniCS project. A continually updated technical report. http://fenicsproject.org.
[48] Lyly, M. (2010). *Basic Programming with Elmer*. CSC – IT Center for Science.
[49] Lyly, M., Malinen, M., & Råback, P. (2014). *Elmer Programmer's Tutorial*. CSC – IT Center for Science. ElmerV7.
[50] Medale, M. & Cochelin, B. (2009). A parallel computer implementation of the asymptotic numerical method to study thermal convection instabilities. *Journal of Computational Physics*, *228*(22), 8249 – 8262.
[51] Medale, M. & Cochelin, B. (2015). High performance computations of steady-state bifurcations in 3D incompressible fluid flows by Asymptotic Numerical Method. *Journal of Computational Physics*, *299*, 581 – 596.
[52] Mizushima, J., Okamoto, H., & Yamaguchi, H. (1996). Stability of flow in a channel with a suddenly expanded part. *Physics of Fluids*, *8*(11), 2933–2942.
[53] Mizushima, J. & Shiotani, Y. (2000). Structural instability of the bifurcation diagram for two-dimensional flow in a channel wilh a sudden expansion. *J. Fluid. Mech.*, *420*, 131.
[54] Mortensen, M. & Valen-Sendstad, K. (2015). Oasis: A high-level/high-performance open source navier?stokes solver. *Computer Physics Communications*, *188*(0), 177 – 188.

[55] Mullin, T., Shipton, S., & Tavener, S. J. (2003). Flow in a symmetric channel with an expanded section. *Fluid Dynamics Research*, *33*(5-6), 433.

[56] Najah, A., Cochelin, B., Damil, N., & Potier-Ferry, M. (1998). A critical review of asymptotic numerical methods. *Archives of Computational Methods in Engineering*, *5*(1), 31–50.

[57] Razafindralandy, D. & Hamdouni, A. (2013). Time integration algorithm based on divergent series resummation, for ordinary and partial differential equations. *Journal of Computational Physics*, *236*, 56 – 73.

[58] Ruokolainen, J., Malinen, M., Råback, P., Zwinger, T., Pursula, A., & Byckling, M. (2014). *Elmer Models Manual*. CSC – IT Center for Science. ElmerV7.

[59] Råback, P., Malinen, M., Ruokolainen, J., Pursula, A., & Zwinger, T. (2015). *Elmer Models Manual*. CSC – IT Center for Science. ElmerV7.

[60] Salinger, A. G., Bou-Rabee, N. M., Pawlowski, R. P., Wilkes, E. D., Burroughs, E. A., Lehoucq, R. B., & Romero, L. A. (2002). LOCA 1.0 Library of Continuation Algorithms: Theory and Implementation Manual. Technical Report SAND2002-0396, Sandia National Laboratory.

[61] Schreck, E. & Schäfer, M. (2000). Numerical study of bifurcation in three-dimensional sudden channel expansions. *Computers & Fluids*, *29*(5), 583 – 593.

[62] Schroeder, W., Martin, K., & Lorensen, B. (2006). *The Visualization Toolkit* (4th ed. ed.). Kitware, Incorporated.

[63] Seydel, R. (2009). *Practical Bifurcation and Stability Analysis*. Interdisciplinary Applied Mathematics. Springer.

[64] Shapira, M., Degani, D., & Weihs, D. (1990). Stability and existence of multiple solutions for viscous flow in suddenly enlarged channels. *Computers & Fluids*, *18*(3), 239 – 258.

[65] Trilinos. Trilinos.

[66] Tsui, Y.-Y. & Wang, H.-W. (2008). Side-wall effects on the bifurcation of the flow through a sudden expansion. *International Journal for Numerical Methods in Fluids*, *56*(2), 167–184.

[67] Van Dyke, M. (1974). Analusis and improvement of pertubation series. *The Quarterly Journal of Mechanics and Applied Mathematics*, *27*(4), 423–450.

[68] Vanderbauwhede, A. (2011). Lyapunov–schmidt method for dynamical systems. In R. A. Meyers (Ed.), *Mathematics of Complexity and Dynamical Systems* (pp. 937–952). Springer New York.

[69] Vannucci, P., Cochelin, B., Damil, N., & Potier-Ferry, M. (1998). An asymptotic-numerical method to compute bifurcating branches. *International Journal for Numerical Methods in Engineering*, *41*(8), 1365–1389.

[70] Werner, B. & Spence, A. (1984). The computation of symmetry-breaking bifurcation points. *Journal on Numerical Analysis*, *21*(2), 388–399.

[71] Winters, K. (1991). Bifurcation and stability: a computational approach. *Computer Physics Communications*, *65*(1–3), 299 – 309.

[72] Zhang Xianyi, Wang Qian, Z. Y. (2012). Model-driven level 3 blas performance optimization on loongson 3a processor. In *2012 IEEE 18th International Conference on Parallel and Distributed Systems (ICPADS),*.

[73] Zienkiewicz, O. & Taylor, R. (2000). *The Finite Element Method, Fluid Dynamics*. The Finite Element Method. Wiley.