

# Simple application of the PoPe method

## Work in progress...

T. Cartier-Michaud, ...

October 2017

### Contents

## 1 PoPe method in a nutshell : decomposition onto a relevant basis plus an error

The Projection on Proper element (PoPe **cite**) method is a method allowing to recover the equations that have generated a set of data. In terms of verification, using output of any simulations to verify a code is fundamentally different from the Method of Manufactured Solutions (MMS **cite**) where the code is verified with respect to an “artificial” solution as it is difficult to obtain a target analytical solution representative of typical simulations. Often the analytical solution chosen is smooth in space and time with respect to the discretisation, it is based on simplified geometries or boundary conditions and it might not use every operators of the model : this verification is partial. Using PoPe onto a set of data produced by a code, if one searches for the full set of equations implemented in the code considered, one does code verification, if one searches for a reduced set of equations compare to the set used in the first place by the code, one does model reduction. PoPe can also be understood as a procedure of a posteriori error checking. In this thesis, it has been used as a post treatment developed independently of the code as a Matlab/Octave library.

## 2 Practical implementation of PoPe method

To explain the method, we use the following density equation eq.(1) as an example :

$$\partial_t N = \vec{\nabla} \cdot (D_N \vec{\nabla}_\perp N) - \vec{\nabla} \cdot (N u_\parallel^{ion} \vec{b}) - \vec{\nabla} \cdot (N \vec{u}_{E \times B}) - \vec{\nabla} \cdot (N \vec{u}_{\vec{\nabla} B}^{ion}) + S_N \quad (1)$$

Equation we immediately rewrite in a more compact form, using “*th*” subscript for “theoretical”,  $\{O^i\}_{th}$  the five operators of eq.(1) for  $i \in [1, 5]$  associated to five theoretical weights  $w_{th}^i$ , also using Einstein’s convention on

indices :

$$\{\partial_t N\}_{th} = w_{th}^i \{O^i\}_{th} \quad (2)$$

$$w_{th} = [+1, -1, -1, -1, +1] \quad (3)$$

What we can easily do is :

1. Measurements of  $\{\partial_t N\}_{ef}$ , the effective (“*ef*” subscript) time derivative computed with finite differences from outputs of a given code. In the present case, finite differences of order four compare to the order one time integration scheme in TOKAM3X.

$$\{\partial_t N\}_{ef}(t) = \sum_{j=-2}^2 c(j) N(t + j\Delta t) + \mathcal{O}(\Delta t^4) \quad (4)$$

$$c(-2 : 2) = [+1, -8, 0, +8, -1]/(12\Delta t) \quad (5)$$

2. Computations of each operator in eq.(1), labeled  $\{O\}_{ol}^i$  for  $i \in [1, 5]$ , using highly accurate off-line (“*ol*” subscript) post-treatments. TOKAM3X using second order method in space, the offline estimation uses method of order 2 to 8.

$$\{O^i\}_{ol} \simeq \{O^i\}_{th} \quad (6)$$

$$\|\{O^i\}_{th} - \{O^i\}_{ol}\| \ll \|\{O^i\}_{th} - \{O^i\}_{ef}\| \quad (7)$$

3. Linear projections of  $\{\partial_t N\}_{ef}$  onto  $\{O\}_{ol}^i$  and thus recover effective weights  $w_{ef}^i$  and an effective residual  $\epsilon_{ef}$  lineary independent of  $\{O\}_{ol}^i$ .

$$\{\partial_t N\}_{ef} = w_{ef}^i \{O\}_{ol}^i + \epsilon_{ef} \quad (8)$$

This projection is simply done by solving the following linear systems eq.(9) based on the least mean square approach. The residual is then recovered by explicitly computing eq.(10).

$$A^t A w_{ef} = A^t \{\partial_t N\}_{ef} \quad (9)$$

$$\{\partial_t N\}_{ef} - A w_{ef} = \epsilon_{ef} \quad (10)$$

$$A(p, i) = \{O(p)^i\}_{ol} \quad (11)$$

The matrix  $A$  is of size  $P \times I$ . Each column of the matrix  $A$  is the evaluation of the  $i^{th}$  operator from the set of  $I$  operators present in the tested equation. Those evaluations are performed for  $P$  points labeled by the indice  $p$ .  $P$  is define by the discretisation used to solve the equation

with the code we want to verify in the first place. Usually  $P \gg I$  which justifies the mean least square approach to obtain the best estimation of  $w_{ef}$ . For TOKAM3X, typically, we have  $P = N_r \times N_\theta \times N_\phi \times N_t \gg 10^6$  for a unique simulation. Nevertheless, to introduce a time dependence in  $w_{ef}$ , we usually use  $N_t$  sets of  $N_r \times N_\theta \times N_\phi \geq 10^4$  points.

This leads to the interpretation of the effective time derivative of a given code ( $\{\partial_t N\}_{ef}$ ) as a sum of weighted (by  $w_{ef}^i$ ) operators ( $\{O\}_{ol}^i$ ) plus a residual ( $\epsilon_{ef}$ ) which has no linear dependency toward the operators of the equations. Ideally,  $w_{ef} = w_{th}$  and  $\epsilon_{ef} = 0$ . This decomposition is relevant as weights such as  $w_{ef}^i$  define the nature of the equation so they control the behavior of the system (theoretically and numerically). A simple error such as the used of a diffusion coefficient two times higher than the one theoretically wanted would immediately be seen with  $w_{ef}^1 = 2w_{th}^1$ . The control of weights would be absolute if  $\{O^i\}_{ef}$ , operators effectively implemented in the code, were “exact”. As we discretise solutions over a finite number of degree of freedom, each having a finite accuracy, operators  $\{O^i\}_{ef}$  usually differ from theoretical expression of operators  $\{O^i\}_{th}$ . The theoretical expression of operators being not accessible usually, we do not compare  $\{O^i\}_{ef}$  to  $\{O^i\}_{th}$  but rather to  $\{O^i\}_{ol}$ , a set of operators computed off-line using a greater accuracy than for  $\{O^i\}_{ef}$ , so  $\{O^i\}_{ol}$  evaluations are closer to  $\{O^i\}_{th}$  than  $\{O^i\}_{ef}$  are close to  $\{O^i\}_{th}$  as define eq.(7). This point is important to be able to associate the residual  $\epsilon_{ef}$  to an error in the code and not an error in the verification process. This off-line greater accuracy is easily obtained as for off-line computations we don’t have to take care of any stability with respect to time integration. Also, verification is usually performed on about 1% of the data generated by a code. For example 1 time step out of 100 over time is sufficient as generally the time step is define to insure small variations of the solution between consecutive iterations. Verifying each time step is still possible but it would conduct to verify many systems very close to each other. Finally recommendation concerns the localisation in space and time of the data used for the verification process. One equation is seen as a law that has to be verified anywhere and at any time with respect to the state of the system at those exact location and time. Thus, we have to carefully handle variables used to recompute the equation in order to be sure not to mix a variable  $a$  evaluated at  $(x, t)$  with a variable  $b$  evaluated at  $(x + \Delta x, t + \Delta t)$ , unless the analytical system clearly states it is the way to do it, that the equation contains a time delay and an action over a distance.

### 3 Aim of this package : a test bed for PoPe

This package aims to offer a test case of the PoPe method in order to quickly learn how to use PoPe and to test new ideas. It is written in Matlab/Octave on a simple advection - diffusion problem using two parameters  $(c, D)$  and an optional source  $S$  :

$$\partial_t f = c \partial_x f + D \partial_x^2 f + S \quad (12)$$

#### 3.1 How to use this package

Just run "scan\_void"...

#### 3.2 Open issues

### 4 Analysis of PoPe's results

As a preliminary study, we can introduce a more general definition of the effective residual  $\epsilon_{ef}$  which, using  $\{O^i\}_{ol} = \{O^i\}_{th}$  for simplification, is the total residual  $\epsilon$  defined as :

$$\{\partial_t N\}_{ef} - \{\partial_t N\}_{th} = \epsilon \quad (13)$$

$$= (w_{ef}^i - w_{th}^i) \{O\}_{ol}^i + \epsilon_{ef} \quad (14)$$

This total residual contains a part linearly dependent on the operators as seen eq.(??). It allows to clearly identify  $\delta^i = w_{ef}^i - w_{th}^i$  the error on weights of the equations. Nevertheless, using only eq.(??) we do not need to solve any linear system to obtain  $\epsilon$ .

In this section, first we estimated the quality of a simulation studying  $\epsilon$  which contains the entire information on the discrepancy between the theoretical evolution of the system and the effective evolution. Then a statistical study of  $w_{ef}$  and  $\epsilon_{ef}$  is performed in order to give a more physical interpretation of the results...

### References