A First Finite Element Solver Shared By Neutron Diffusion, Heat Transfer And Mechanics

C. Patricot, G. Allaire, O. Fandeur

M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, Korea, April 16-20, 2017, on USB (2017)

A First Finite Element Solver Shared by Neutron Diffusion, Heat Transfer and Mechanics

Cyril Patricot,[a] Grégoire Allaire,[b] Olivier Fandeur[c]

[a]*CEA, DEN, DM2S, SERMA, F-91191 Gif-sur-Yvette, France*
[b]*CMAP, Ecole polytechnique, CNRS, Université Paris-Saclay, 91128 Palaiseau, France*
[c]*CEA, DEN, DM2S, SEMT, F-91191 Gif-sur-Yvette, France*
cyril.patricot@cea.fr, allaire@cmap.polytechnique.fr, olivier.fandeur@cea.fr

Abstract - *Nuclear reactor core simulations involve several physics, especially in accidental transients. Neutron transport is needed to compute the power distribution. Thermal-hydraulics drives the cooling of the core. Fuel mechanics and heat transfer describe fuel state (including temperature). Mechanics allows to take into account deformations of the core. A lot of works have been done on coupling techniques between these physics, but are usually based on separated codes or solvers. In this paper, we present an alternative approach: the development of a shared solver for the coupled physics. A multiphysics solver is proposed for a time-dependent coupling between neutron diffusion, heat transfer and linear mechanics. It is based on the finite element method and the Newton algorithm. A very simple application is given and shows the rightness of the developments and the relevance of the solver.*

I. INTRODUCTION

Nuclear reactor cores can be deformed by thermal expansion, irradiation effects or during particular accidental transients. These deformations are likely to impact neutron transport in all reactor types. Fast neutron reactors are particularly sensible to these effects, because of their thermal features (large temperature gradients, potentially strong temperature variations in case of accident) and the way neutrons evolve within the core (important role played by leakages in the neutron balance, weak fraction of delayed neutrons). As a consequence, in the context of the development of the fourth generation of nuclear reactors, methodologies to take into account core distortions in deterministic neutron transport codes have been developed.

These tools aim at providing linear feedback coefficients [1, 2], or to compute the neutron flux. In this last case, the mesh used can itself be deformed [3, 4] or not [5, 6] (a geometry projection method is then needed).

In the mean time, an important effort is done on multiphysics coupling techniques, mainly between neutron transport, thermohydraulics and fuel physics. The coupling between neutron transport and structural mechanics is rarely considered, simply because less needed. Nevertheless, generic lessons can be derived from this case.

In this paper, we present what we believe to be a first step toward a unified finite element solver for neutron transport and thermomechanics. Using the Newton algorithm, a complete jacobian matrix, including coupling terms between finite element discretisations, is written. A very simple application is given, on a simplified view of the Godiva experiment (which was already used in [7, 8] to study the neutron transport - thermomechanics coupling). Indeed, this kind of coupling techniques is rarely considered in nuclear reactor physics, but could lead to substantial improving of calculation time and robustness.

II. THE COUPLED EQUATIONS

1. Common notations

We consider a time-dependent problem, solved with an implicit time-scheme (Each physic impacts others at current time-step). Finite elements are used for the three coupled physics, that is to say neutron transport (diffusion), heat transfer and mechanics.

Heat transfer and neutron diffusion share the same elementary functions, noted ξ_i . Mechanics elementary functions are noted $\underline{\xi_i}$ and are vector quantities with the same dimension than space. In general, the number of underlinings indicates the dimension of tensors (one for vectors, two for matrices).

There is no need to give more information about the finite elements used. The methodology developed here is independent of the precise type of finite elements.

The non-deformed space domain is noted Ω_0 , the current one (deformed by mechanics) is noted Ω and the last computed one (from the last Newton iteration) Ω_p .

2. Neutron diffusion

We consider the time-dependent, multigroup neutron diffusion equations, coupled with delayed neutrons. A semi-analytical solving of the delayed neutron precursor equation leads to a linear system for the next time-step neutron flux Φ:

$$A_{ne}\Phi(t) = B_{ne}. \tag{1}$$

This linear system is generally solved with Gauss-Seidel iterations over energetic groups. For the sake of simplicity, we admit here that A_{ne} and B_{ne} are built with only three kind of matrices, K, R et M:

$$(K)^g_{ij} = \int_\Omega D^g \underline{\nabla}\xi_i . \underline{\nabla}\xi_j d\Omega, \tag{2a}$$

$$(R)_{ij} = \int_\Omega \sigma \xi_i \xi_j d\Omega, \tag{2b}$$

$$(M)_{ij} = \int_\Omega \xi_i \xi_j d\Omega. \tag{2c}$$

These matrices are impacted by mechanics through Ω, which may change, and through σ and D^g, which depend on isotopic concentrations and are therefore impacted by any expansion.

On the other hand, no Doppler effect is considered here, since it is negligible in the Godiva experiment. There is therefore no direct impact of temperature change on neutron diffusion.

3. Heat transfer

We consider the time-dependent heat transfer equation, which can be written as a linear system for the next time-step temperature $\mathcal{T}$:

$$A_{th}\mathcal{T}(t) = B_{th}, \tag{3}$$

with:

$$(A_{th})_{ij} = \frac{1}{\Delta t}\int_{\Omega_0} \rho C_p \xi_i \xi_j d\Omega_0 + \int_{\Omega_0} \lambda \nabla\xi_i \nabla\xi_j d\Omega_0, \tag{4a}$$

$$(B_{th})_i = \frac{1}{\Delta t}\int_{\Omega_0} \rho C_p \mathcal{T}(t-\Delta t)\xi_i d\Omega_0 + \int_{\Omega_0} P\xi_i d\Omega_0. \tag{4b}$$

Note that A_{th} and B_{th} are defined with Ω_0 instead of Ω. This very common simplification is known to have a very limited impact on results if distortion is small. As a consequence, heat transfer is not impacted by mechanics. However, neutron diffusion impacts heat transfer through the power P in (4b):

$$P = \sum_{g,j} K^g \xi_j \phi_j^g. \tag{5}$$

4. Mechanics

For the sake of simplicity, static linear elasticity is considered here. A time-dependent equation could be used without major difficulty, but it would make the equations uselessly complicated. The considered model can be written as a linear system for the next time-step displacement U:

$$A_{me}U(t) = B_{me}, \tag{6}$$

with:

$$(A_{me})_{ij} = \int_{\Omega_0} \left(\lambda_{me} Tr(\underline{\underline{b_i}}) Tr(\underline{\underline{b_j}}) + 2\mu Tr(\underline{\underline{b_i}} \cdot \underline{\underline{b_j}})\right) d\Omega_0, \tag{7a}$$

$$(B_{me})_i = \int_{\Omega_0} 3\kappa\alpha(\mathcal{T}(t) - \mathcal{T}(t=0)) Tr(\underline{\underline{b_i}}) d\Omega_0. \tag{7b}$$

Here again, the non-deformed space domain Ω_0 is used instead of Ω. As for heat transfer, this simplification is very common, and should not alter noticeably the results. Heat transfer impacts mechanics through the temperature $\mathcal{T}(t)$ in B_{me}. On the other hand, neutron diffusion has no direct feedback on mechanics.

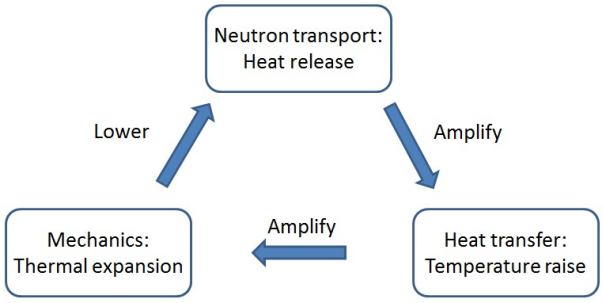

Fig. 1. Interactions between physics.

5. Summary of the coupling

The coupling is summarized in Figure 1.

III. COMMON SOLVING

1. The Newton algorithm

Let X be the concatenation of Φ, $\mathcal{T}$ and U. We define the global problem this way:

$$F(X(t)) = A(X(t))X(t) - B(X(t)) = 0, \tag{8}$$

with:

$$A(X) = \begin{pmatrix} A_{ne} & 0 & 0 \\ 0 & A_{th} & 0 \\ 0 & 0 & A_{me} \end{pmatrix} \text{ and } B(X) = \begin{pmatrix} B_{ne} \\ B_{th} \\ B_{me} \end{pmatrix}. \tag{9}$$

The equation (8) is not linear, as A and B depend on X. In order to solve it, we use the Newton algorithm.

The first step consists in defining the jacobian matrix J_X:

$$(J_X)_{ij} = \frac{\partial F_i}{\partial X_j} = \sum_k \frac{\partial A_{ik}}{\partial X_j} X_k + A_{ij} - \frac{\partial B_i}{\partial X_j}. \tag{10}$$

From a first guess $X^{(0)}$ of the solution, iterations are made:

- Computation of $J_{X^{(n-1)}}$ and of $F(X^{(n-1)})$,
- Solving of the linear system $J_{X^{(n-1)}}\delta X = -F(X^{(n-1)})$,
- Computation of the new approximation of X: $X^{(n)} = X^{(n-1)} + \delta X$.

Algorithm ends when $\|F(X^{(n)})\|$ is small enough.

2. Calculation of the jacobian matrix

A. Term from mechanics

Replacing $\mathcal{T}$, in (7b), by its decomposition on the finite element basis (the ξ_i) gives:

$$(B_{me})_i = \int_{\Omega_0} 3\kappa\alpha(\sum_k \mathcal{T}_k\xi_k - \mathcal{T}(t=0)) Tr(\underline{\underline{b_i}}) d\Omega_0.$$

One differentiates now this expression with respect to $\mathcal{T}$ components to get the result we were looking for (according to (10)):

$$\frac{\partial (B_{me})_i}{\partial \mathcal{T}_j} = \int_{\Omega_0} 3\kappa\alpha\xi_j Tr(\underline{\underline{b_i}}) d\Omega_0. \quad (11)$$

B. Term from heat transfer

In the same way, by replacing (5) in (4b) and by differentiating with respect to Φ components, one gets:

$$\frac{\partial (B_{th})_i}{\partial \phi_j^g} = \int_{\Omega_0} K^g \xi_j \xi_i d\Omega_0. \quad (12)$$

C. Terms from neutron diffusion

a. Change of variables

Neutron diffusion terms (2) are defined on the current space domain Ω, unlike heat transfer and mechanics terms. In the following, we use a change of variables to rewrite neutron diffusion terms on the last computed space domain Ω_p. Thanks to that rewriting, the impact of mechanics can be made explicit.

We introduce the following change of variables, between $\underline{y} \in \Omega_p$ and $\underline{x} \in \Omega$:

$$\underline{x} = \underline{y} + \underline{u'}(\underline{y}) = (\underline{\underline{Id}} + \underline{u'})(\underline{y}). \quad (13)$$

$\underline{u'}$ in (13) can be written (under the small displacement assumption) like:

$$\underline{u'} = \underline{u} - \underline{u}_p. \quad (14)$$

with $\underline{u}$ the displacement from Ω_0 to Ω and $\underline{u}_p$ the one from Ω_0 to Ω_p. $\underline{u}$ is the displacement computed by mechanics and $\underline{u}_p$ is assumed to be known (from previous Newton iteration).

Let f be a function of Ω. One notes $\bar{f}$ the corresponding function of Ω_p:

$$\bar{f}(\underline{y}) = f \circ (\underline{\underline{Id}} + \underline{u'})(\underline{y}) = f(\underline{x}).$$

The change of the elementary volume is: $d\Omega = |\det(\underline{\underline{\nabla}}(\underline{\underline{Id}} + \underline{u'}))| d\Omega_p$. In the following, we omit the absolute value. This determinant is always positive for a kinematically admissible displacement. We also assume that the considered displacements are small enough for that property to stay true, even with displacements interpolated by the Newton algorithm. As a consequence, one gets for all function f adequately regular:

$$\int_\Omega f \xi_i \xi_j d\Omega = \int_{\Omega_p} \bar{f} \bar{\xi}_i \bar{\xi}_j \det(\underline{\underline{Id}} + \underline{\underline{\nabla u'}}) d\Omega_p.$$

Unfortunately, this is not enough for matrices of the form (2b) because cross-sections are proportional to material densities. This means that we have, in order to maintain mass balance, the following equation ($\forall \Omega' \subset \Omega$ associated to Ω'_p by the displacement $\underline{u'}$), with σ_p a cross-section on Ω_p, and σ the corresponding one on Ω:

$$\int_{\Omega'} \sigma d\Omega' = \int_{\Omega'_p} \bar{\sigma} \det(\underline{\underline{Id}} + \underline{\underline{\nabla u'}}) d\Omega'_p = \int_{\Omega'_p} \sigma_p d\Omega'_p.$$

In other words:

$$\bar{\sigma} \det(\underline{\underline{Id}} + \underline{\underline{\nabla u'}}) = \sigma_p. \quad (15)$$

Therefore, volume of integration change and cross-section adjustments due to material expansions balance each other perfectly for matrices of the form (2b).

For matrices of the form (2a), the impact of the change of variables (13) on the gradients of elementary functions must be taken into account.

One gets, $\forall f, \underline{g}$ adequately regular:

$$\begin{aligned} \underline{\nabla}(f \circ \underline{g}) &= \left(\frac{\partial}{\partial x_i} (f \circ \underline{g}) \right)_i \\ &= \left(\sum_j \frac{\partial f}{\partial x_j} \circ \underline{g} \, \frac{\partial \underline{g}_j}{\partial x_i} \right)_i \\ &= (\underline{\underline{\nabla g}})^t (\underline{\nabla f}) \circ \underline{g}. \end{aligned}$$

As a consequence, if we note ∇_p the gradient defined on Ω_p, we get : $\underline{\nabla} \xi_i \circ (\underline{\underline{Id}} + \underline{u'}) = (\underline{\underline{Id}} + \underline{\underline{\nabla u'}})^{-t} \underline{\nabla}_p \bar{\xi}_i$.

Moreover, if . is the usual scalar product, we have $\forall \underline{\underline{A}}, \underline{X}, \underline{Y}$:

$$\underline{\underline{AX}}.\underline{\underline{AY}} = \underline{X}^t \underline{\underline{A}}^t \underline{\underline{AY}} = (\underline{\underline{A}}^t \underline{\underline{AX}}).\underline{Y}.$$

Finally, one gets $\forall f$ adequately regular:

$$\int_\Omega f \underline{\nabla} \xi_i . \underline{\nabla} \xi_j d\Omega =$$

$$\int_{\Omega_p} \bar{f} (\underline{\underline{Id}} + \underline{\underline{\nabla u'}})^{-1} (\underline{\underline{Id}} + \underline{\underline{\nabla u'}})^{-t} \underline{\nabla}_p \bar{\xi}_i . \underline{\nabla}_p \bar{\xi}_j \det(\underline{\underline{Id}} + \underline{\underline{\nabla u'}}) d\Omega_p. \quad (16)$$

In addition, diffusion coefficients (D^g in (2a)) are, in our case, inversely proportional to material densities. This means that (the demonstration is analogous to the one for (15)):

$$\bar{D^g} = \det(\underline{\underline{Id}} + \underline{\underline{\nabla u'}}) D_p^g.$$

Therefore, volume of integration change and diffusion coefficient adjustment due to material expansion lead to a squared determinant in (16) when f is a diffusion coefficient.

b. Calculation of partial derivatives

In order to calculate the partial derivatives needed in the jacobian matrix, we first write the first order approximation of previously calculated terms with respect to $\underline{\underline{\nabla u'}}$. We recall the following equations:

$$\det(\underline{\underline{Id}} + \underline{\underline{\nabla u'}}) = 1 + Tr(\underline{\underline{\nabla u'}}) + o(\underline{\underline{\nabla u'}})$$

$$(\underline{\underline{Id}} + \underline{\underline{\nabla u'}})^{-1} = \underline{\underline{Id}} - \underline{\underline{\nabla u'}} + o(\underline{\underline{\nabla u'}})$$

This leads to the following equations for all the terms needed in the neutron diffusion problem (2):

$$(K)^g_{ij} = \int_\Omega D^g \underline{\nabla}\underline{\xi}_i.\underline{\nabla}\underline{\xi}_j d\Omega \approx$$
$$\int_{\Omega_p} D^g_p \left(\underline{\underline{Id}}(1 + 2Tr(\underline{\underline{\nabla u'}})) - \underline{\underline{\nabla u'}} - \underline{\underline{\nabla u'}}^t\right) \underline{\nabla}_p\bar{\underline{\xi}}_i.\underline{\nabla}_p\bar{\underline{\xi}}_j d\Omega_p,$$

$$(R)_{ij} = \int_\Omega \sigma\xi_i\xi_j d\Omega = \int_{\Omega_p} \sigma_p\bar{\xi}_i\bar{\xi}_j d\Omega_p.$$

$$(M)_{ij} = \int_\Omega \xi_i\xi_j d\Omega \approx \int_{\Omega_p} \bar{\xi}_i\bar{\xi}_j(1 + Tr(\underline{\underline{\nabla u'}}))d\Omega_p.$$

Let $\underline{\underline{\epsilon'}}$ be the strain tensor : $\underline{\underline{\epsilon'}} = \frac{1}{2}\left(\underline{\underline{\nabla u'}} + \underline{\underline{\nabla u'}}^t\right)$, and $\underline{\underline{b_i}}$ the strain due to elementary displacements $\underline{\xi_i}$. Thanks to the linearity of the operator which links a strain to a displacement, one gets:

$$\underline{\underline{\epsilon'}} = \sum_i u'_i \underline{\underline{b_i}},$$

with u'_i the coordinates of the displacement $\underline{u'}$ on the basis made with the elementary displacements $\underline{\xi_i}$. In addition, from (14), one gets $u'_i = u_i - (u_i)_p$, where u_i are the coordinates of $\underline{u}$, and $(u_i)_p$ thoses of $\underline{u}_p$.

Finally if we replace $\underline{u'}$ by $\underline{\underline{\epsilon'}}$ in the previous expressions of K, R and M, we can get the following partial derivatives with respect to u_i, at $\underline{u}_p$:

$$\frac{\partial(K)^g_{ij}}{\partial u_k}(\underline{u}_p) = \int_{\Omega_p} 2D^g_p(Tr(\underline{\underline{b_k}})\underline{\underline{Id}} - \underline{\underline{b_k}})\underline{\nabla}_p\bar{\underline{\xi}}_i.\underline{\nabla}_p\bar{\underline{\xi}}_j d\Omega_p,$$

$$\frac{\partial(M)_{ij}}{\partial u_k}(\underline{u}_p) = \int_{\Omega_p} \bar{\xi}_i\bar{\xi}_j Tr(\underline{\underline{b_k}})d\Omega_p.$$

3. The jacobian matrix

Finally, if we replace the terms we have just calculated in (10), we get the jacobian matrix:

$$J_X = \begin{pmatrix} A_{ne} & 0 & C_{MN} \\ C_{NT} & A_{th} & 0 \\ 0 & C_{TM} & A_{me} \end{pmatrix}, \tag{17}$$

with

$$(C_{NT})^g_{ij} = -\int_{\Omega_0} K^g\xi_j\xi_i d\Omega_0, \tag{18a}$$

$$(C_{TM})_{ij} = -\int_{\Omega_0} 3\kappa\alpha\xi_j Tr(\underline{\underline{b_i}})d\Omega_0, \tag{18b}$$

$$(C_{MN})^{g'\to g}_{ij} =$$
$$\delta_{g,g'}\sum_k \left(\frac{1}{V^g\Delta t}\int_{\Omega_p} \bar{\xi}_i\bar{\xi}_k Tr(\underline{\underline{b_j}})d\Omega_p\left(\phi^g_k(t) - \phi^g_k(t-\Delta t)\right)\right.$$
$$\left. + \int_{\Omega_p} 2D^g_p(Tr(\underline{\underline{b_j}})\underline{\underline{Id}} - \underline{\underline{b_j}})\underline{\nabla}_p\bar{\underline{\xi}}_i.\underline{\nabla}_p\bar{\underline{\xi}}_k d\Omega_p\right). \tag{18c}$$

Note that C_{NT} and C_{TM} are constant. The impact of neutron diffusion on heat transfer, given by equations (4b) and (5), is exactly linear. This is therefore normal to find a constant bloc (C_{NT}) for this part of the jacobian. The same stands for the impact of heat transfer on mechanics (C_{TM}), given by (7b).

On the other hand, C_{MN} needs to be computed at each iteration, because of the integration over Ω_p and the presence of $\phi^g_k(t)$.

4. Approximations

We used two approximations in the jacobian for the following application. They are needed to simplify the computation of C_{MN}. The final result is not modified (the solved equation (8) is not modified), only the algorithm convergence speed is.

1. The first term in (18c) is replaced by:

$$\int_{\Omega_p} \bar{\xi}_i\bar{\xi}_k Tr(\underline{\underline{b_j}})d\Omega_p \approx \delta_{i,k}\int_{\Omega_p} \bar{\xi}_i Tr(\underline{\underline{b_j}})d\Omega_p. \tag{19}$$

2. Ω_p is replaced with Ω_0 in (18c).

In addition, we recall that the small displacement assumption was made to get (14).

IV. APPLICATION

CAST3M [9], a finite element code dedicated to structural mechanics, is used for this application.

We consider a simple numerical experiment with no physical meaning: an homogeneous rectangular in 2D space. Its properties (for the three physics) are choosen arbitrary in order to reinforce the coupling.

Elementary functions are first order polynomials. Null neutron flux and temperature boundary conditions are used, and global translations and rotations are eliminated.

At $t = 0$ the system is prompt-critical, and only one time-step is computed.

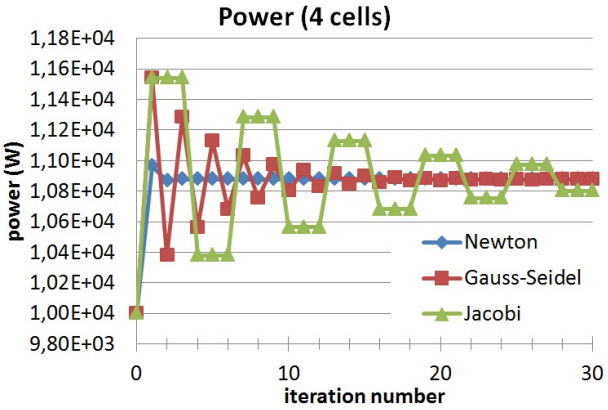

Fig. 2. Global power from different coupling techniques, with 4 cells.

Figures 2 and 3 give global power as function of iteration number, with 4 and 100 cells respectively, for different coupling techniques:

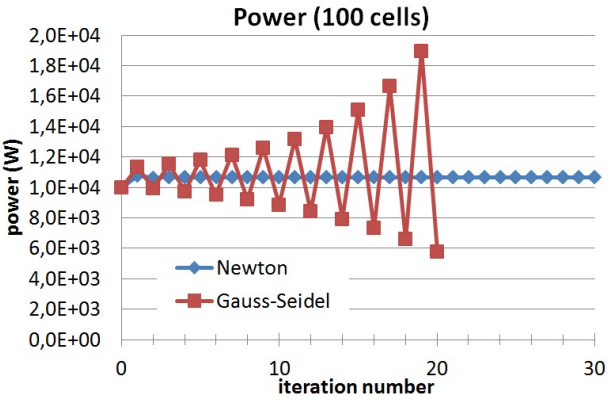

Fig. 3. Global power from different coupling techniques, with 100 cells.

- Newton: The algorithm presented in this paper;
- Gauss-Seidel: At each iteration, neutron diffusion is solved first, then heat transfer and finally mechanics;
- Jacobi: The three physics are solved at the same time (but independently), at each iteration.

One can see on Figures 2 and 3 that the Newton algorithm is faster than the others. Power computed by the Jacobi algorithm is modified only every three iterations because of the circularity of the coupling.

Discrepancy between current global power and converged one (after 50 Newton iterations), for the case 4 cells, is plotted in Figure 4. It confirms that the Newton algorithm is the fastest one by far. The ratio of minimal discrepancy to initial global power value is about 10^{-14}, the numerical noise level.

Note that the convergence of our "Newton" algorithm is linear (whereas actual Newton algorithm convergence should be quadratic). This is due to the approximations we made in the jacobian computation.

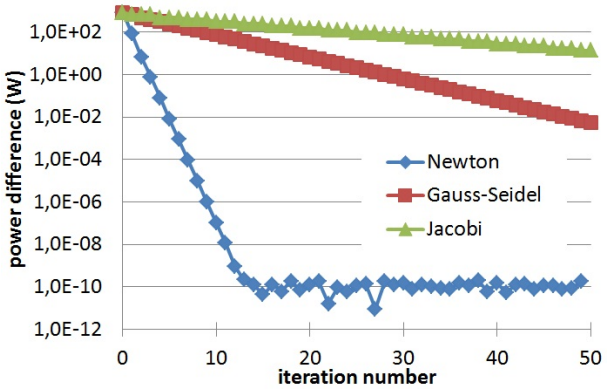

Fig. 4. Discrepancy with converge global power for different coupling techniques, with 4 cells.

V. CONCLUSION

We give in this paper the expression of the jacobian matrix for a coupling between finite element modeling of multigroup neutron diffusion, heat transfer and linear mechanics. This approach is not dependent on the application case considered or the finite element type used.

A simple application case shows that it does converge faster (in terms of iteration number) than common multisolver coupling techniques. A better robustness is also expected. It would be premature to conclude now on calculation time, work still needs to be done on the linear system solving.

The process was simplified here by the use of the same discretisation for every physics: use of the finite element method, with the same elementary functions, on the same mesh and with the same time-step. This is not required by the method. If discretisations differ, an additional step is needed, similar to a projection or a change of variables. It can be taken into account in the jacobian matrix.

Future work should be undertaken to remove the approximations we made in the jacobian calculation, in order to numericaly prove the exactness of our jacobian matrix.

We wish this work to inspire development of new innovative coupling solvers.

More elements on neutron transport in deformed cores and coupling techniques between neutron transport and mechanics can be found in [10].

NOMENCLATURE

α	thermal expansion coefficient
Δt	Time-step length
λ	Heat conductivity
$\lambda_{me}, \mu, \kappa$	Lamé parameters, $\kappa = \lambda_{me} + (2/3)\mu$
$\mathcal{T}$	Discretized form of the temperature
Ω	Current space domain
Ω_0	Non-deformed space domain
Ω_p	Previous Newton iteration space domain
Φ	Discretized form of the neutron flux
ϕ_j^g	Scalar component of Φ
ρC_p	Volumic heat capacity
σ	A macroscopic cross-section defined on Ω
σ_p	σ taking into account material expansion between Ω and Ω_p
$\underline{\underline{\epsilon'}}$	$= \frac{1}{2}\left(\underline{\underline{\nabla u'}} + \underline{\underline{\nabla u'^t}}\right)$ (strain tensor between Ω_p and Ω)
$\underline{\xi_i}$	Elementary function for mechanics
$\underline{\underline{b_j}}$	$= \frac{1}{2}\left(\underline{\underline{\nabla \xi_j}} + \underline{\underline{\nabla \xi_j}^t}\right)$ (elementary strain)
$\underline{u}$	Displacement from Ω_0 to Ω
$\underline{u'}$	Displacement from Ω_p to Ω

$\underline{u}_p$	Displacement from Ω_0 to Ω_p
ξ_i	Elementary function for heat transfer and neutron transport
A	Global matrix with A_{ne}, A_{th} and A_{me} on the diagonal
A_{me}	Matrix of the mechanics problem
A_{ne}	Matrix of the neutron diffusion problem
A_{th}	Matrix of the heat transfer problem
B	Concatenation of B_{ne}, B_{th} and B_{me}
B_{me}	Right hand side of the mechanics problem
B_{ne}	Right hand side of the neutron diffusion problem
B_{th}	Right hand side of the heat transfer problem
C_{MN}	Part of the jacobian taking into account the impact of mechancis on neutron diffusion
C_{NT}	Part of the jacobian taking into account the impact of neutron diffusion on heat transfer
C_{TM}	Part of the jacobian taking into account the impact of heat transfer on mechancis
D^g	Diffusion coefficient of group g
D_p^g	Diffusion coefficient of group g over Ω_p
F	$= AX - B$
K^g	Power production cross-section
P	Volumic power
U	Discretized form of the displacement
V^g	Speed of neutrons of group g
X	Concatenation of Φ, $\mathcal{T}$ and U

REFERENCES

1. M. REED, K. SMITH, and B. FORGET, “The "virtual density" theory of neutronics: a generic method for geometry distortion reactivity coefficients,” in “Proceedings of PHYSOR 2014,” Kyoto, Japan (Sep. 28 - Oct. 3 2014).
2. L. ANDRIOLO, A. RINEISKI, B. VEZZONI, F. GABRIELLI, X.-N. CHEN, W. MASCHEK, F. DELAGE, and E. MERLE-LUCOTTE, “An innovative methodology for evaluating core thermal expansion feedbacks in transient analyses,” in “Proceedings of ICAPP 2015,” Nice, France (May 03-06 2015).
3. C. FIORINA and K. MIKITYUK, “Application of the new GeN-Foam multi-physics solver to the European Sodium Fast Reactor and verification against available codes,” in “Proceedings of ICAPP 2015,” Nice, France (May 03-06 2015).
4. C. PATRICOT, O. FANDEUR, A.-M. BAUDRON, and D. BROC, “Neutronic calculation of deformed cores: development of a time-dependent diffusion solver in CAST3M, a mechanics dedicated finite elements code,” in “Proceedings of PHYSOR 2016,” Sun Valley, USA (May 01-05 2016).
5. M. GENTILI, B. FONTAINE, and G. RIMPAULT, “Deformed Core Reactivity Evaluation with Mesh Projection–Based Method,” Nuclear Technology, **192**, 1, 11–24 (2015).
6. C. PATRICOT, D. BROC, E. HOURCADE, and K. AMMAR, “APOLLO3 based method for 3D wraped cores calculations - Application to flowering tests of Phénix,” in “Proceedings of PHYSOR 2014,” Kyoto, Japan (Sep. 28 - Oct. 03 2014).
7. M. AUFIERO, C. FIORINA, A. LAUREAU, P. RUBIOLO, and V. VALTAVIRTA, “Serpent–OpenFOAM coupling in transient mode: simulation of a Godiva prompt critical burst,” in “Proceedings of M&C + SNA + MC 2015,” Nashville, USA (Apr. 19-23 2015).
8. C. PATRICOT, O. FANDEUR, and A.-M. BAUDRON, “Interests of the improved quasi-static method for multi-physics calculations illustrated on a neutronics-thermomechanics coupling,” in “Proceedings of PHYSOR 2016,” Sun Valley, USA (May 01-05 2016).
9. CAST3M, URL : http://www-cast3m.cea.fr/ (2015).
10. C. PATRICOT, Couplages multi-physiques: évaluation des impacts méthodologiques lors de simulations de couplages neutronique/thermique/mécanique., Ph.D. thesis, URL : https://pastel.archives-ouvertes.fr/tel-01355622, Université Paris-Saclay (2016).