

## PARALLEL IN TIME ALGORITHMS FOR NONLINEAR ITERATIVE METHODS

MUSTAFA GAJA<sup>1,2</sup> AND OLGA GORYNINA<sup>3</sup>

**Abstract.** In this paper we investigate the feasibility of applying the Parareal algorithm [5, 6] for quasi-static nonlinear structural analysis problems. We describe how this proposal has been realized and present some preliminary numerical results of applying this algorithm to a beam undergoing nonlinear deflection with a contact boundary condition. Further numerical experiments are needed to provide an evidence for the efficiency of the method.

**Résumé.** Dans cet article on cherche à déterminer si l'algorithme de Parareel [5, 6] est applicable pour des structures quasi-statiques non linéaires. On décrit comment cette approche a été réalisée puis on présente quelques résultats numériques préliminaires de notre algorithme pour un problème poutre non linéaire dont une condition au limite modélise un contact. D'autres expériences numériques sont nécessaires afin de prouver l'efficacité de la méthode.

### INTRODUCTION

The simulation of complex nonlinear structures via the finite element method is generally based on quasi-static incremental loading procedure leading to successive nonlinear problems solved by Newton-like methods [7, 9] which themselves entail solving multiple ill-conditioned large linear systems. For complex structures, the total number of steps may be very large and, since the procedure is recursive, the computational cost is very large. Reducing the time for such simulations is of great interest for numerical engineering in industry. A recently developed parallel in time method for time dependent problems is the Parareal algorithm [5, 6] that allows the parallelization of the computation in the temporal domain making use of HPC facilities.

The objective of this work is to present a proposal for the application of the Parareal algorithm for quasi-static nonlinear processes in order to perform these quasi-static steps in parallel by analogy with the Parareal method for time-dependent systems of ordinary or partial differential equations. In this work we describe how this proposal has been realized on a common example of a beam undergoing nonlinear elastic deformation with additional boundary nonlinearity. Numerical results demonstrate the possibility of using the Parareal algorithm on elementary quasi-static problems, although more conclusive numerical tests are needed to provide a substantial evidence for the efficiency of the approach.

**Plan of the paper.** We present the model problem, its spacial discretization and linearization in Section 1. Basic idea and precise description of the Parareal algorithm for the case of quasi-static nonlinear problems are laid out in Section 2. Numerical experiments for several test cases are reported in Section 3. We discuss some reoccurring themes and questions that faced us during this work in Section 4. Finally, we draw some concluding remarks in Section 5.

---

<sup>1</sup> Max Planck Institute for Plasma Physics

<sup>2</sup> Technische Universitt Mnchen

<sup>3</sup> Laboratoire de mathématiques de Besançon, University of Franche-Comté

### 1. NONLINEAR STRUCTURES AND INCREMENTAL LOADING

We treat the boundary value problem for the equations of nonlinear elasticity [1,8] as a minimization problem. For a domain  $\Omega = [0, 1] \times [0, 1] \subset \mathbb{R}^2$  we want to find the displacement field  $(u_1, u_2) : \Omega \rightarrow \mathbb{R}^2$  that minimizes the functional of total potential energy  $J(u_1, u_2)$ :

$$\min_{(u_1, u_2) \in V} J(u_1, u_2), \tag{1}$$

where  $V$  is a suitable function space that satisfies some Dirichlet boundary conditions:

$$(u_1, u_2)(x_1, x_2) = (\bar{u}_1, \bar{u}_2), \quad (x_1, x_2) \in \Gamma_v, \tag{2}$$

where  $\bar{u}_1$  and  $\bar{u}_2$  are some given constants,  $\Gamma_v$  denotes the part of the boundary on which the displacements are given. The total potential energy is given by the following formula:

$$J(u_1, u_2) = \int_{\Omega} f(F(u_1, u_2)) - \int_{\Gamma_P} P_{\alpha} u_2, \tag{3}$$

where  $F(u_1, u_2) = A(E[u_1, u_2], E[u_1, u_2])$ ,  $A(X, Y)$  is a bilinear symmetric positive form with respect to matrices  $X$  and  $Y$ ,  $f$  is a given  $C^2$  function corresponding to hyperelastic constitutive law,  $P_{\alpha}$  is some external force which we apply on the boundary surface  $\Gamma_P \subset \partial\Omega$  (for the sake of simplicity we suppose that the force act in the  $x_2$ -direction) and

$$E[u_1, u_2] = (E_{ij})_{i=1,2,j=1,2}, \quad E_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_i} \frac{\partial u_k}{\partial x_j} \right), \tag{4}$$

is the Green-Saint-Venant strain tensor.

The finite element method is the standard modeling approach to simulate and analyze the behavior of solids [2]. Let us approximate the space  $V$  by the finite element space  $V_h$ . We thus introduce a regular mesh  $\mathcal{T}_h$  on  $\Omega$ , a triangle  $K$  of the mesh  $\mathcal{T}_h$  and a standard  $P1$  finite element space on it  $V_h \subset V$ :

$$V_h = \{v_h \in C(\bar{\Omega}) : v_h|_K \in \mathbb{P}_1 \quad \forall K \in \mathcal{T}_h\},$$

Thus we obtain the finite element problem of finding the field  $(u_h, v_h)$  that satisfy (2) and

$$\min_{(u_h, v_h) \in (V_h, V_h)} J(u_h, v_h). \tag{5}$$

The solution of the large strain nonlinear problem will require an iterative process and Newton's method forms the basis of practical schemes [9]. We start with an initial guess  $(u_0, v_0)$  which is reasonably close to the true correct solution. Let us suppose that we already know the solution  $(u_{n-1}, v_{n-1})$  from the step  $n - 1$  then we attempt to correct this guess to bring it closer to the proper solution by setting

$$u_n = u_{n-1} + du, \quad v_n = v_{n-1} + dv,$$

where the correction term  $(du, dv)$  is given by

$$D^2 J(u_{n-1}, v_{n-1})((w, s), (du, dv)) = DJ(u_{n-1}, v_{n-1})(w, s) \quad \forall (w, s) \in (V_h, V_h), \tag{6}$$

where  $DJ(u, v)$  and  $D^2 J(u, v)$  are the Jacobian and the Hessian of  $J(u, v)$  respectively. Then we check if the magnitude of the correction  $(du, dv)$  is small enough, and if not, we go to the next iteration  $n + 1$  and compute the new correction term  $(\tilde{d}u, \tilde{d}v)$ .

Loads	$N_I$	$N_{IC}$
1,2,3	no convergence	...
4	4	11
5	4	9
6	4	8
7	3	29
8	3	11

TABLE 1. Convergence of Newton's method depending on the number of loading steps. Cantilever rubber beam loaded at the upper bound with a uniformly distributed vertical force  $F = 200 Pa$  with a contact surface, mesh  $60 \times 20$  points,  $N_I$  - average number of Newton iterations,  $N_{IC}$  - number of Newton iterations at the loading step when the contact appears for the first time.

### 1.1. Incremental loading

There is no guarantee that Newton's method will converge. It will converge quadratically to the exact solution only if the initial guess is sufficiently close to the correct solution. A common way to avoid this problem is to apply the load in a series of increments instead of all at once [7, 9]. For example, we would like to solve our problem for some certain boundary force  $F$ , but numerical test shows us that Newton's method does not converge. We can divide the force into 2 loads: at the beginning we solve our problem with force  $F/2$  and after we solve the problem once more but with force  $F$  using the solution at the end of the preceding increment as the initial guess. The quality of this guess could be improved by reducing the increment once more. In general, small load increments are essential for a better accuracy of the solution. In Table 1 we show the case of a cantilever rubber beam loaded with a uniformly distributed force  $F = 200 Pa$  and a contact surface. In the case of applying the force all at once, we can't achieve the convergence of Newton's method. The situation is persistent when using 2 and 3 loading steps, whereas we achieve convergence when we start using 4 loading steps and so on. Moreover, this example demonstrates that the number of incremental loads can affect the number of iterations required by Newton's method to achieve the convergence. Indeed, when the force is divided into 7 loading steps we only need 3 Newton iterations to achieve the same tolerance like in the case of 4 loading steps with 4 Newton iterations. This example shows that the incremental loading procedure is required in the case when the initial guess for Newton's method is too far from the exact solution. Note that with incremental loads we obtain essentially a quasi-static nonlinear problem instead of a static one; from now on we have a so-called "pseudo-time" direction which is in fact an additional dimension where it is possible to apply the idea of Parareal method. But before discussing this topic we will introduce a boundary nonlinearity to our problem.

### 1.2. Boundary nonlinearity

The incremental loading procedure allows us to insert a different form of nonlinearity in our model: a contact problem [4]. Let us consider the cantilever beam:

$$u_1(0, x_2) = 0, u_2(0, x_2) = 0, \quad x_1 \in [0, 1], x_2 \in [0, 1],$$

under a vertical load  $P_\alpha$  and in one moment the beam touches a solid surface  $X_2 = X_{sf}$  with border  $\Gamma_S$  where  $X_{sf}$  and  $(X_1, X_2)$  are the Eulerian coordinates. We model the appearance of the contact surface with the following algorithm: at every incremental loading step we look for the nodes of our mesh at the border of the beam and check if the vertical displacements of those nodes are equal to or greater than the vertical coordinate of the surface  $X_{sf}$  (See Figure 1 a), in what follows we call them violated nodes [3]. On the next incremental loading step for every violated node we set the sliding contact boundary condition (a vertex of the finite element mesh can slide over the surface, but cannot move away from it), in other words we take the vertical displacement

equal to the displacement corresponding to the contact surface with Dirichlet boundary conditions (See Figure 1 b):

$$u_2(x_1, 0) = X_{sf}, \quad \forall x_1 \in \Gamma_S.$$

The number of incremental loadings is very important for detecting the contact surface. From Table 1 we see that for the cantilever beam configuration loaded with an uniformly distributed force we achieve the maximum number of the Newton iterations in the load when the contact appears, because we are converging in presence of such strong nonlinear effect like a contact. Thus the accuracy of the solution depends on the nonlinearity involved in the model and the number of taken load increments.

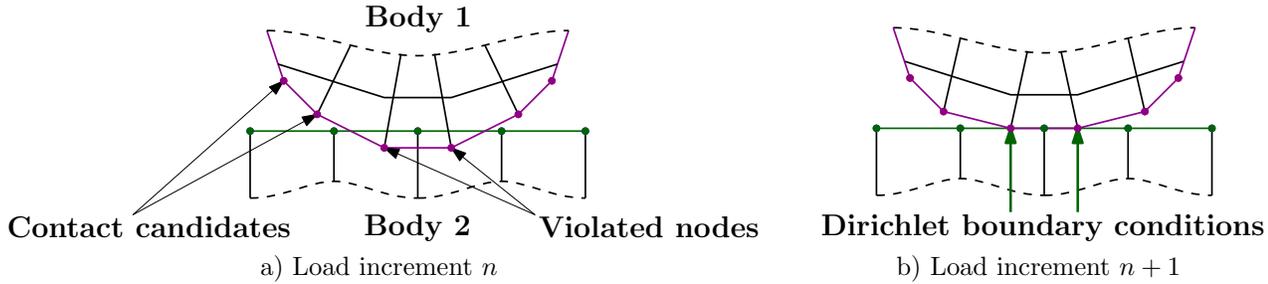


FIGURE 1. Contact surface modulation: we detect the violated nodes at displacements corresponding to load increment  $n$ , then we impose Dirichlet boundary conditions for the violated nodes before the start of load increment  $n + 1$ .

## 2. PARAREAL METHOD

### 2.1. Parareal method for ODEs

In order to understand the mechanism of the parallelization in time we will briefly explain the basic idea of the parareal method with a simple first order ordinary differential equation [5]:

$$u'(t) = -au(t) \text{ on } [0, T], \quad u(0) = u_0, \tag{7}$$

where  $a$  is a constant. Let us introduce the temporal mesh on the time interval  $[0, T]$ :

$$0 = T^0 < T^1 < \dots < T^N = T,$$

with time steps  $\Delta T = T/N$ . The Parareal algorithm utilizes a coarse solver  $G$  to quickly step through time by computing relatively cheap approximate solutions for all time intervals of interest, and then simultaneously refines all of these approximate solutions using an accurate fine solver  $F$ . Let  $G_{\Delta T}(u_{n-1}, T^{n-1})$  for all  $n = 1, \dots, N$  be a rough approximation of  $u(T^n)$  with an initial condition  $u(T^{n-1}) = u_{n-1}$ ,  $F_{\Delta T}(u_{n-1}, T^{n-1})$  is a more accurate approximation of the solution  $u(T^n)$  with an initial condition  $u(T^{n-1}) = u_{n-1}$ . Then starting with a coarse approximation  $U_n^0$  at the time points  $T^1, T^2, \dots, T^N$ , Parareal performs for  $k = 0, 1, \dots$  the correction iteration:

$$U_{n+1}^{k+1} = F_{\Delta T}(U_n^k, T^n) + G_{\Delta T}(U_{n+1}^{k+1}, T^n) - G_{\Delta T}(U_n^k, T^n). \tag{8}$$

The application of the fine solver to each time interval  $[T^n, T^{n+1}]$  is independent of the other time intervals, and thus parallelizable. From (8) we see that the refined solutions are then fed back to the coarse solver, and the iterative cycle continues until all time intervals are converged. The dependencies between the time intervals are carried through the coarse solver which involves stepping sequentially through time. This, of course, represents a sequential process, but it leads to a more rapid solution, assuming that the coarse solver is much faster than the fine solver, and hence, computationally negligible.

In the simple numerical example for equation (7) we set  $T = 10$  and perform 5 coarse steps each parareal iteration, thus the coarse solution is obtained by the implicit Euler scheme with step  $\Delta T = 2$ . After that, we introduce a finer mesh with a smaller time step  $\Delta t = 0.02$  on each interval  $[T^n, T^{n-1}]$ , thus we have 100 fine time steps on each interval  $[T^n, T^{n-1}]$ . We use the same implicit Euler scheme as a fine solver.

We present the error in log scale between the parareal solution and the analytical solution of equation (7) for each of the 5 parareal iterations in Figure 2. In order to show the convergence of the parareal method at each parareal iteration we also present the error between the solution of Euler's method with 500 time steps and the analytical solution of equation (7).

The solution of the first coarse time step is processed by the fine solver using the exact initial state  $u(0)$ . For every further parareal iteration, the converged accurate solution is computed with initial conditions from the previous iteration, thus the second iteration will give us convergence at least for the first two steps, the third iteration for the first three steps, and so on. Thus the algorithm is guaranteed to converge in at most  $K = N$  iterations (see Figure 2), though in practice it is possible to achieve much faster convergence with judicious choice of the coarse propagator. The computational complexity of the parareal algorithm is given by the following formula:

$$KNC_C + (K - 1)N \frac{\Delta T}{\Delta t} C_{\mathcal{F}}, \quad (9)$$

where  $C_C$  is the cost to perform the coarse propagator over one time step,  $C_{\mathcal{F}}$  is the complexity of one time step for the fine propagator,  $K$  is the number of iterations required to achieve the required convergence of the parareal algorithm and  $N$  is the number of coarse steps (and the number of processors as well).

Looking at the Parareal algorithm from a different perspective, it is reasonable to presume that the idea of parallelizing the time domain for time-dependent problems can be used for the case of quasi-static problems in order to reduce the computation time.

## 2.2. Parareal method for nonlinear structures

In what follows we shall propose a "time" parallel iterative method for solving the minimization problem (5). At first, we introduce  $NM$  load increment steps and assume that we compute the coarse solution at points  $0, M, 2M, \dots, NM$  and the fine solution at points  $0, 1, 2, \dots, NM$ . Thus we introduce a pseudo-time parameter  $\tilde{t} \in [0, \tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_{NM}]$  which is used to describe our algorithm via an analogy with a time-dependent problem. For the sake of simplicity we assume that all increments are the same and denote points for the coarse solution as  $[0, \tilde{T}_1, \tilde{T}_2, \dots, \tilde{T}_N]$ , in other words  $\tilde{t}_M = \tilde{T}_1, \tilde{t}_{2M} = \tilde{T}_2, \dots, \tilde{t}_{NM} = \tilde{T}_N$ . At every point  $[0, \tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_{NM}]$  we introduce the exact solution of problem (5)  $(u(\tilde{t}_n), v(\tilde{t}_n))$  for  $n \in [0, 1, \dots, NM]$ . We want to find approximate solution  $(U_n^k, V_n^k) = (U^k(\tilde{t}_n), V^k(\tilde{t}_n))$  where index  $k$  is the number of parareal iterations. Like in the case of a partial differential equation, let  $G(U_{nM}^k, V_{nM}^k)$  for all  $n = 0, \dots, N - 1$  be a rough approximation of  $(u(\tilde{T}_{n+1}), v(\tilde{T}_{n+1}))$  which we compute starting from initial guess  $(U_{nM}^k, V_{nM}^k)$  by using the coarse propagator that is described below. Let  $F(U_{nM}^k, V_{nM}^k)$  be a more accurate approximation of the solution  $(u(\tilde{T}_{n+1}), v(\tilde{T}_{n+1}))$  which we compute starting from displacements  $(U_{nM}^k, V_{nM}^k)$  by using the fine propagator that is described below.

We start from the initial guess  $(U_0^0, V_0^0) = (u_0, v_0)$  for Newton's method at the first increment step and obtain an initial coarse solution via  $(U_{nM}^0, V_{nM}^0) = G(U_{(n-1)M}^0, V_{(n-1)M}^0)$  for  $n \in [1, \dots, N]$ . That is we obtained Parareal solution of iteration  $k = 0$ . Hereafter on every Parareal iteration  $k \geq 1$  we determine our numerical solution by the formula:

$$(U_{nM}^k, V_{nM}^k) = G(U_{(n-1)M}^k, V_{(n-1)M}^k) + F(U_{(n-1)M}^{k-1}, V_{(n-1)M}^{k-1}) - G(U_{(n-1)M}^{k-1}, V_{(n-1)M}^{k-1}), \quad (10)$$

### The coarse propagator

Once the solution is known at iteration  $k - 1$ , the coarse problem at iteration  $k + 1$  is Newton's method for  $N$  incremental loads. For all coarse increment steps  $n = 1, \dots, N$  starting with the initial guess for Newton's method as displacements from the previous parareal iteration  $G^0(U_{nM}^k, V_{nM}^k) = (U_{nM}^{k-1}, V_{nM}^{k-1})$ , we compute the

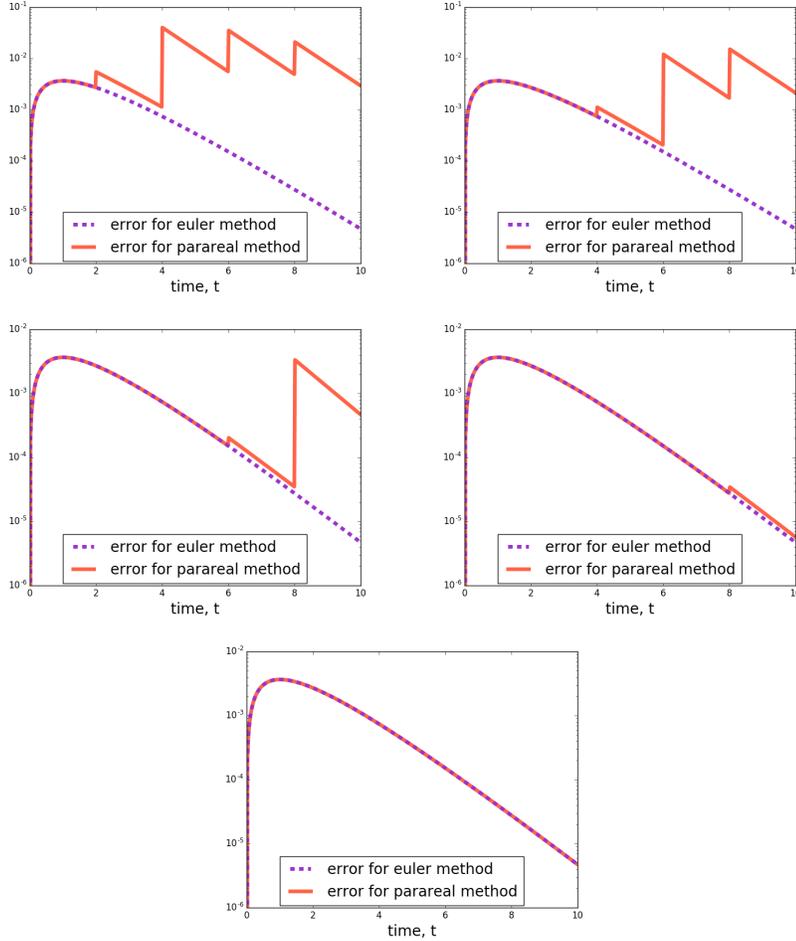


FIGURE 2. The error between analytical solution of equation (7) and parareal solution (8) during 5 iterations of parareal method.

coarse solution at the Newton iteration  $j$  as:

$$G^j(U_{nM}^k, V_{nM}^k) = G^{j-1}(U_{nM}^k, V_{nM}^k) + (du_n^j, dv_n^j), \tag{11}$$

where the correction term  $(du_n^j, dv_n^j)$  by analogy with (6) follows from

$$D^2 J(G^{j-1}(U_{nM}^k, V_{nM}^k))((w, s), (du_n^j, dv_n^j)) = DJ(G^{j-1}(U_{nM}^k, V_{nM}^k))(w, s) \quad \forall (w, s) \in (V_h, V_h). \tag{12}$$

Let's assume that at the Newton iteration number  $j^*$  we achieved the required accuracy for approximate solution, then the coarse propagator is  $G(U_{nM}^k, V_{nM}^k) = G^{j^*}(U_{nM}^k, V_{nM}^k)$ .

**The fine propagator**

We use a procedure similar to (11) and (12) to obtain the fine solution  $F(U_{nM}^k, V_{nM}^k)$  on parareal iteration  $k$ . Inside every "coarse" pseudo-time interval  $[\tilde{T}_n, \tilde{T}_{n+1}]$  we introduce a local fine solver on fine mesh  $\tilde{F}(U_{nM+m}^k, V_{nM+m}^k)$  to emphasize the fact that this solver propagates the solution on fine increment steps. We compute the solution with the same Newton's method for  $M$  incremental loads using the displacements from the previous Parareal iteration as an initial guess for the first incremental load  $\tilde{F}^0(U_{nM}^k, V_{nM}^k) =$

Set $N$	Number of coarse steps
Set $M$	Number of fine steps inside every coarse step
Set $K$	Number of parareal iterations
Set $(U_0^0, V_0^0) = (u_0, v_0)$	
For $n = 1$ to $N$	
Compute $(U_n^0, V_n^0) = G(U_{n-1}^0, V_{n-1}^0)$	Initial coarse propagation
End for	
For $k = 1$ to $K$ do	Iterative loop
For $n = 0$ to $N - 1$ do	Parallelizable part
For $m = 0$ to $M - 1$ do	
Compute $\tilde{F}(U_{nM+m}^{k-1}, V_{nM+m}^{k-1})$	Local fine propagator
End for	
Compute $F(U_{nM}^{k-1}, V_{nM}^{k-1})$	Fine propagation
End for	
For $n = 0$ to $N - 1$ do	Sequential part
Compute $G(U_n^{k-1}, V_n^{k-1})$	Coarse propagation
Compute $(U_n^k, V_n^k)$ from (10)	Correction
End for	
End for	

TABLE 2. Parareal algorithm for nonlinear quasi-static structure.

$\tilde{F}(U_{nM}^{k-1}, V_{nM}^{k-1})$ . For all the next fine increment steps  $m = 1, \dots, M - 1$  we use displacements from the previous increment step as an initial guess for Newton's method, in other words, we start by setting the initial guess as  $\tilde{F}^0(U_{nM+m}^k, V_{nM+m}^k) = \tilde{F}(U_{nM+m-1}^k, V_{nM+m-1}^k)$ . If we already know the fine solution  $\tilde{F}^{j-1}(U_{nM+m}^k, V_{nM+m}^k)$  of the Newton iteration  $j - 1$  then the solution of iteration  $j$  is:

$$\tilde{F}^j(U_{nM+m}^k, V_{nM+m}^k) = \tilde{F}^{j-1}(U_{nM+m}^k, V_{nM+m}^k) + (du_m^j, dv_m^j), \quad (13)$$

where the correction term  $(du_m^j, dv_m^j)$  like in the case of the coarse solver (12) is defined by

$$D^2 J \left( \tilde{F}^{j-1}(U_{nM+m}^k, V_{nM+m}^k) \right) ((w, s), (du_m^j, dv_m^j)) = DJ \left( \tilde{F}^{j-1}(U_{nM+m}^k, V_{nM+m}^k) \right) (w, s) \quad \forall (w, s) \in (V_h, V_h). \quad (14)$$

Let's imagine that at the Newton iteration number  $j^*$  we achieve the required tolerance then we determine the fine propagator as  $\tilde{F}(U_{nM+m}^k, V_{nM+m}^k) = \tilde{F}^{j^*}(U_{nM+m}^k, V_{nM+m}^k)$  for all fine increment steps  $m = 1, \dots, M - 1$ . Thus we determine the fine propagator at coarse time points as  $F(U_{nM}^k, V_{nM}^k) = \tilde{F}(U_{(n+1)M-1}^k, V_{(n+1)M-1}^k)$ .

Our algorithm is summarized in Table 2.

### 3. A NUMERICAL STUDY OF THE PARAREAL ALGORITHM FOR A NONLINEAR BEAM

We now turn to numerical examples for our algorithm. What we report here are preliminary results that have to be extended to more complex cases. We run our experiments for four different cases with different boundary conditions: a beam anchored at one or two ends, with and without a contact; see Figure 3. However, here we report numerical results only for the case of a one-end fixed beam with a contact, see Figure 3 d, since

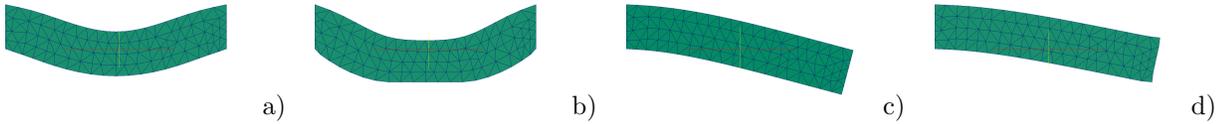


FIGURE 3. Beam configuration: a) beam fixed at two ends without a contact, b) beam fixed at two ends with a contact, c) beam fixed at one end without a contact, d) beam fixed at one end with a contact.

Parareal iteration	$\ \mathbf{u}\ _{L^\infty}$	$\ u\ _{L^2}$	$\ v\ _{L^2}$	<i>AbsError</i>
direct method	0.027394	0.0480169	0.161934	
1	0.0278855	0.0512589	0.168256	4.9e-4
2	0.0280801	0.0521499	0.170242	6.8e-4
3	0.0265591	0.0443302	0.15321	8.3e-4
4	0.0265591	0.0443302	0.15321	8.3e-4
5	0.0265599	0.0443482	0.152963	8.4e-4

TABLE 3. Nonlinear beam with one fixed end and a contact. Convergence for fixed mesh  $60 \times 20$ ,  $F = 120 Pa$ , 6 coarse steps, 6 fine steps. The absolute error is calculated between the parareal solution and the direct method in  $L^\infty$  norm .

it is an interesting configuration in the sense of the degree of nonlinearity involved. All the results reported hereafter correspond to a vertically uniform distributed external load applied to upper bound of a rubber beam. In the numerical experiments, FreeFem++ is used for the finite element formulation and the linear algebra implementation, the fine solver is parallelized using MPI.

The first experiment considered here is the Parareal algorithm parallelized on 4 cores, 6 coarse steps and 6 fine steps are used, see Table 3. In order to measure the quality of the solution we are looking for  $L^\infty$  and  $L^2$  norms of displacements. The first line of Table 3 corresponds to displacements obtained by applying 36 loading steps for Newton’s method. We observe that we have a convergence from the first Parareal iteration, which is reasonable since our numerical example is performed with a sufficiently fine mesh. Indeed, the coarse solver converges and it means that 6 loading steps are enough to reach an accurate solution under the final load and since our process is quasi-static we do not care about the solution at every "time" point. We would like to underline that our aim here is to demonstrate the possibility of using Parareal algorithm on elementary quasi-static problems in order to inspire ourselves for further more realistic and strongly nonlinear models.

The problem formulation for the second experiment is the same as that of the previous experiment except that here we have used a different space mesh size for the coarse and fine propagation. Due to the sequential nature of the coarse propagator we would like it to be as fast and cheap as possible. Since the coarse propagator will be corrected with the fine propagator we can use rough space mesh coarse steps. Results with different coarse and fine mesh sizes are summarized in Table 4. First line of Table 4 corresponds to displacements obtained with loading steps for direct Newton’s method. Note that as in the previous example we have a convergence from the first Parareal iteration.

#### 4. DISCUSSION

The results presented in Section 3 support the proposal for the feasibility of applying the parareal method for nonlinear structural analysis problems. That said, there is a crucial point that needs to be clarified; the pseudo time stepping is only there in order to ensure the convergence of Newton’s method. It is thus more like a stability problem: once we have a converged result then it is the good one. Thus a legitimate question is why

Coarse mesh	Fine mesh	Parareal iteration	Number of loads	$\ \mathbf{u}\ _{L^\infty}$	$\ u\ _{L^2}$	$\ v\ _{L^2}$	<i>Abs Error</i>
$60 \times 20$	$60 \times 20$	direct method	20	0.0278	0.0473	0.1603	
$60 \times 20$	$90 \times 30$	1	12	0.0272	0.0469	0.1596	6e-4
$60 \times 20$	$90 \times 30$	2	12	0.0272	0.0469	0.1596	6e-4
$60 \times 20$	$60 \times 20$	1	12	0.0269	0.0462	0.1579	9e-4
$60 \times 20$	$60 \times 20$	2	12	0.0271	0.0469	0.1595	7e-4

TABLE 4. Nonlinear beam with one fixed end and a contact, different mesh sizes for coarse and fine propagation,  $F = 100 Pa$ , 12 coarse steps, 12 fine steps. The absolute error is calculated between the parareal solution and the direct method in  $L^\infty$  norm .

both with several parareal iterations in the first place? For instance, one can think of taking as an initial guess the coarse solution and then apply a fine solver in order to achieve an accurate solution at the last loading step. This line of thought is interesting, but one would be losing the guaranteed theoretical convergence proof laid out in theory [5], as we are unaware of a theory supporting the aforementioned line of thought.

We don't present results comparing the wall clock time for the regular sequential loading versus the parareal implementation because for the purposes of this work, we had to write files as an output for the coarse and fine solvers and in turn, read them as an initial guess, calculating and propagating the differences and so forth. This process of reading and writing files is naturally computationally expensive, in comparison to the direct sequential case within FreeFem++ as the data points are passed directly from one loading step to another. This comes as a result of not finding a way to go around passing the data points directly without writing them in FreeFem++ for the parareal case between the fine and coarse solver, and we chose to focus on tackling the question of feasibility of the idea. Further implementations would naturally require optimization to avoid this computational bottle-neck. Finally, an interesting recurring question is about the required number of parareal iterations. We think that this point would depend on the model and boundary conditions at hand. Not only, but also on the requirement posed by the application and what is the required level of convergence.

## 5. CONCLUDING REMARKS

We presented some preliminary numerical results as a proposal to show the feasibility of applying the Parareal algorithm for nonlinear structural analysis problems. The subject of the forthcoming work is to perform more extensive numerical tests to provide a more comprehensive evidence for the feasibility study and then to extend this algorithm to industrial 3D problems and more realistic models using the software package NUMEA provided by Hutchinson.

## 6. ACKNOWLEDGEMENTS

This research was funded by Total. We thank François-Xavier Roux for supervision and assistance in the project, and Yvon Maday for useful comments. We would also like to show our gratitude to the Research and Development team of Hutchinson company, and we thank Maxime Pallud and Alexandros Markopoulos for their help and support for laying the foundation for the upcoming prospective work and giving us access to their software NUMEA.

## REFERENCES

- [1] S.S. Antman. Nonlinear Problems of Elasticity, Springer, 2005.
- [2] D. Braess. Finite elements: Theory, fast solvers, and applications in solid mechanics. Cambridge University Press, 2007.
- [3] N.-H. Kim. Introduction to nonlinear finite element analysis. Springer Science & Business Media, 2014.
- [4] W. Lacarbonara. Nonlinear structural mechanics: theory, dynamical phenomena and modeling. Springer Science & Business Media, 2013.

- [5] J.L. Lions, Y. Maday & G. Turinici. A "parareal" in time discretization of PDEs. *Comptes Rendus de l'Academie des Sciences Series I Mathematics*, 332(7):661-668, 2001.
- [6] Y. Maday & G. Turinici. The parareal in time iterative solver: a further direction to parallel implementation. In T. J. Barth, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, T. Schlick, R. Kornhuber, R. Hoppe, J. Periaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 441-448. Springer Berlin Heidelberg, 2005.
- [7] R. W. Ogden. *Non-linear elastic deformations*. Courier Corporation, 1997
- [8] J.J. Stoker. *Nonlinear elasticity*, Gordon and Breach, 1968.
- [9] O. C. Zienkiewicz & R. L. Taylor. *The finite element method: solid mechanics*. Butterworth-heinemann, T. 2., 2000.