Efficient coarse correction for parallel time stepping in plaque growth simulations

Frei, Stefan; Heinlein, Alexander

EFFICIENT COARSE CORRECTION FOR PARALLEL TIME STEPPING IN PLAQUE GROWTH SIMULATIONS

STEFAN FREI[1] AND ALEXANDER HEINLEIN[2]

[1] Department of Mathematics and Statistics, University of Konstanz, 78457 Konstanz, Germany, stefan.frei@uni-konstanz.de

[2] Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628CD Delft, Netherlands, a.heinlein@tudelft.nl

Abstract. In order to make the numerical simulation of atherosclerotic plaque growth feasible, a temporal homogenization approach is employed. The resulting macro-scale problem for the plaque growth can be further accelerated by using parallel time integration schemes, such as the parareal algorithm. However, the parallel scalability is dominated by the computational cost of the coarse propagator. Therefore, in this paper, an interpolation-based coarse propagator, which uses growth values from previously computed micro-scale problems, is introduced. For a simple model problem, it is shown that this approach reduces both the computational work for a single parareal iteration as well as the required number of parareal iterations.

1 Introduction

We are concerned with the numerical simulation of atherogenesis, that is, the growth of plaque inside arteries. For instance, in case of rupture, this can lead to heart attacks or strokes, which are often fatal. While the plaque growth takes place over months to years, the wall shear stress, which is an important driving force for plaque growth, varies within the second of a heartbeat. Therefore, a fully time-resolved fluid-structure interaction (FSI) simulation of plaque growth can easily require $\mathcal{O}(10^9)$ time steps; this is computationally infeasible.

In order to make numerical simulations of arterial plaque growth feasible, a temporal homogenization approach [6] can be employed, decoupling the FSI and the plaque growth. In the resulting algorithm, a periodic FSI micro scale problem with fixed growth values has to be solved in each macro scale step of the plaque growth problem. The temporal micro-scale problem cannot be neglected. However, only few heartbeats have to be simulated to reach a periodic state; cf. [7, 5]. As a result, the computational cost is reduced significantly.

The time to solution can be further reduced by parallelization. Using parallelization in space typically results in a good parallel efficiency, however, even after reaching a strong scaling limit, the time to solution might remain high due to the large number of time steps. In order to parallelize further, a parallel time integration approach for the plaque growth problem has been introduced in the preprint [4], based on the parareal algorithm [8]. As usual in two level methods, the parallel scalability is limited by the cost of the coarse level, here, the coarse propagator. In

1

particular, when the classical parareal approach is applied to the plaque growth problem, a periodic micro scale FSI problem has to be solved in each time step of the coarse propagator. To speed up the coarse propagator, an approach which reuses growth values from previous micro-scale computations has been introduced in [4], leading to an increase in the number of required parareal iterations, but a reduction in the overall time to solution.

In this paper, we introduce an interpolation-based coarse propagator, which reduces the computational cost while being much more accurate compared to the re-use approach. The new interpolation approach uses growth values from all previous micro-scale computations for the interpolation and, thus, becomes more accurate with each parareal iteration. We test it on a simple two-dimensional model, for which it even speeds up the convergence of the parareal iteration, compared to the classical parareal algorithm.

2 Equations

We consider a time-dependent fluid-structure interaction system, where the fluid is modeled by the Navier–Stokes equations and the solid by the Saint Venant–Kirchhoff model. To account for the solid growth, we incorporate a multiplicative growth term into the deformation gradient. The growth term is motivated by typical plaque growth models, such as those in [12, 7, 11].

2.1 Fluid-structure interaction

We consider a partition of an overall domain $\Omega(t) = \mathcal{F}(t) \cup \Gamma(t) \cup \mathcal{S}(t)$ into a fluid part $\mathcal{F}(t)$, an interface $\Gamma(t)$ and a solid part $\mathcal{S}(t)$. The blood flow and its interaction with the surrounding vessel wall is modelled by the following non-stationary FSI system:

$$
\begin{aligned}
\rho_f(\partial_t \mathbf{v}_f + \mathbf{v}_f \cdot \nabla \mathbf{v}_f) - \operatorname{div} \boldsymbol{\sigma}_f &= 0, & \operatorname{div} \mathbf{v}_f &= 0 && \text{in } \mathcal{F}(t), \\
\rho_s \partial_t \hat{\mathbf{v}}_s - \operatorname{div}(\hat{\mathbf{F}}_e \hat{\boldsymbol{\Sigma}}_e) &= 0, & \partial_t \hat{\mathbf{u}}_s - \hat{\mathbf{v}}_s &= 0 && \text{in } \hat{S}, \\
\boldsymbol{\sigma}_f \vec{n}_f + \boldsymbol{\sigma}_s \vec{n}_s &= 0, & \mathbf{v}_f &= \mathbf{v}_s && \text{on } \Gamma(t).
\end{aligned}
\tag{1}
$$

Here, $\mathbf{v}_f$ and $\hat{\mathbf{v}}_s$ denote the fluid and solid velocities, respectively, and $\hat{\mathbf{u}}_s$ the solid displacement. Quantities with a "hat" are defined in Lagrangian coordinates, quantities without a "hat" in the current Eulerian coordinate framework. Two quantities $\hat{f}(\hat{x})$ and $f(x)$ correspond to each other by a $C^{1,1}$-diffeomorphism $\hat{\xi} : \hat{\Omega} \to \Omega(t)$ and the relation $\hat{f} = f \circ \hat{\xi}$. Below, we will also make use of the solid deformation gradient $\hat{F}_s = I + \hat{\nabla} \hat{u}_s$, which is the derivative of $\hat{\xi}$ in the solid part. The constants ρ_f and ρ_s are the densities of blood and vessel wall, and $\vec{n}_f$ and $\vec{n}_s$ are outward pointing normal vectors of the fluid and solid domains, respectively.

Furthermore, let $\boldsymbol{\sigma}_f = \rho_f \nu_f (\nabla \mathbf{v}_f + \nabla \mathbf{v}_f^T) - p_f I$ denote the Cauchy stress tensor of the fluid, where ν_f is the kinematic viscosity of blood. On the other hand, $\boldsymbol{\sigma}_s$ denotes the Cauchy stress tensor of the solid, which is related to the Piola-Kirchhoff stress $\hat{\boldsymbol{\Sigma}}_e$ (which is defined below) by the Piola transformation $\hat{\boldsymbol{\sigma}}_s = \hat{J}_e^{-1} \hat{\mathbf{F}}_e \hat{\boldsymbol{\Sigma}}_e \hat{\mathbf{F}}_e^T$.

A sketch of the computational domain, which will be used in the numerical examples below, is given in fig. 1. We split the outer boundary of Ω into a solid part Γ_s with homogeneous Dirichlet conditions for the solid displacement, a fluid part Γ_f^{in} with an inflow Dirichlet condition for the fluid velocity and an outflow part Γ_f^{out}, where a *do-nothing boundary condition* is imposed. In
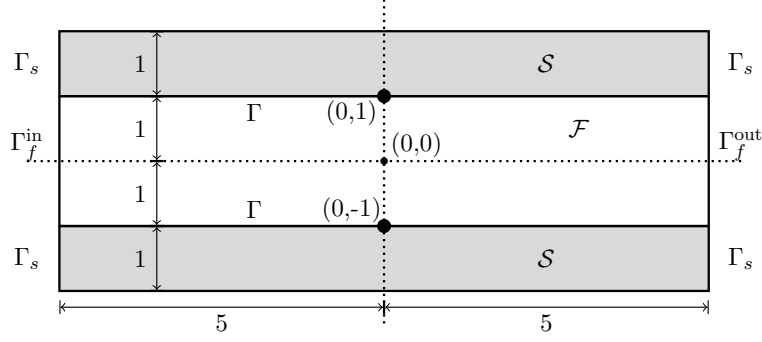
Figure 1: Sketch of the computational domain centered at the origin $(0,0)$; $\mathcal{F}$ and $\mathcal{S}$ are the fluid and solid parts, respectively, and the solid lines correspond to the fluid-solid interface Γ. The plaque growth is initiated at $(0, \pm 1)$.

particular, the boundary data is given by

$$\mathbf{v}_f = \mathbf{v}^{\mathrm{in}} \text{ on } \Gamma_f^{\mathrm{in}}, \quad \rho_f \nu_f (\vec{n}_f \cdot \nabla)\mathbf{v}_f - p_f \vec{n}_f = 0 \text{ on } \Gamma_f^{\mathrm{out}}, \quad \hat{\mathbf{u}}_s = 0 \text{ on } \hat{\Gamma}_s, \tag{2}$$

where $\mathbf{v}^{\mathrm{in}}$ is the inflow velocity on Γ_f^{in}.

2.2 A simple model for plaque growth

Developing a realistic model for the plaque growth at the vessel wall is a complex task that involves the interaction of many different molecules and species; see for example [11]. Furthermore, the plaque growth will also strongly depend on the geometry and material model of the arterial wall, as well as the interaction with the blood flow. In this contribution, we restrict ourselves to a simplified plaque growth model and instead focus on algorithmic aspects. More specifically, we consider a simple ODE-based model taken from [7, 5] that focuses on the influence of the concentration of foam cells c_s on the growth; the spatial distribution of the growth is prescribed in this model.

Therefore, let the spatial distribution of the plaque growth be fixed up to a constant factor, which corresponds to the foam cell concentration c_s. Moreover, we assume that the evolution of the foam cell concentration c_s can be described by a simple ODE, which depends on the wall shear stress $\boldsymbol{\sigma}_f^{WS}$ at the vessel wall:

$$\partial_t c_s = \gamma(\boldsymbol{\sigma}_f^{WS}, c_s) := \frac{\alpha}{(1 + c_s)} \left(1 + \frac{\|\boldsymbol{\sigma}_f^{WS}\|_{L^2(\Gamma)}^2}{\sigma_0^2} \right)^{-1}, \qquad \boldsymbol{\sigma}_f^{WS} := \left(I_d - \vec{n}_f \vec{n}_f^T \right) \boldsymbol{\sigma}_f \vec{n}_f. \tag{3}$$

The reference wall shear stress σ_0 and the scale separation parameter α are parameters of the growth model. For cardiovascular plaque growth, we typically have $\alpha = \mathcal{O}(10^{-7})\,\mathrm{s}^{-1}$; see the discussion in [6].

As mentioned earlier, we model the solid growth by a multiplicative splitting of the deformation gradient $\hat{\mathbf{F}}_s$. In particular, let $\hat{\mathbf{F}}_e$ be the elastic part and $\hat{\mathbf{F}}_g$ a growth function, such that

$$\hat{\mathbf{F}}_s = \hat{\mathbf{F}}_e \hat{\mathbf{F}}_g \quad \Leftrightarrow \quad \hat{\mathbf{F}}_e = \hat{\mathbf{F}}_s \hat{\mathbf{F}}_g^{-1} = [I + \hat{\nabla}\hat{\mathbf{u}}_s]\hat{\mathbf{F}}_g^{-1}; \tag{4}$$

3

cf. [10, 12, 7]. In the ODE model, we use the following growth function depending on c_s

$$\hat{g}(\hat{x}, \hat{y}, t) = 1 + c_s \exp\left(-\hat{x}^2\right)(2 - |\hat{y}|), \quad \hat{\mathbf{F}}_g(\hat{x}, \hat{y}, t) := \hat{g}(\hat{x}, \hat{y}, t)\, I. \tag{5}$$

Hence, the growth rate is influenced by the fluid-structure interaction only in form of the variable c_s. From eq. (5) and the fact that the computational domain is centered around the origin, we obtain that growth is concentrated at $(0, \pm 1)$; see fig. 1. It follows that

$$\hat{\mathbf{F}}_g = \hat{g}I \quad \Rightarrow \quad \hat{\mathbf{F}}_e := \hat{g}^{-1}\hat{\mathbf{F}}_s, \tag{6}$$

and the elastic Green–Lagrange strain is given by

$$\hat{\boldsymbol{E}}_e = \frac{1}{2}(\hat{\boldsymbol{F}}_e^T \hat{\boldsymbol{F}}_e - I) = \frac{1}{2}(\hat{g}^{-2}\hat{\boldsymbol{F}}_s^T \hat{\boldsymbol{F}}_s - I) \tag{7}$$

resulting in the Piola–Kirchhoff stresses (see [7] for details)

$$\hat{\mathbf{F}}_e\hat{\boldsymbol{\Sigma}}_e = 2\mu_s\hat{\boldsymbol{F}}_e\hat{\boldsymbol{E}}_e + \lambda_s \operatorname{tr}(\hat{\boldsymbol{E}}_e)\hat{\boldsymbol{F}}_e = 2\mu_s\hat{g}^{-1}\hat{\boldsymbol{F}}_s\hat{\boldsymbol{E}}_e + \lambda_s\hat{g}^{-1}\operatorname{tr}(\hat{\boldsymbol{E}}_e)\hat{\boldsymbol{F}}_s. \tag{8}$$

3 Numerical framework

To solve the FSI problem eq. (1) we use an Arbitrary Lagrangian Eulerian (ALE) approach; see, e.g., [9]. For a more detailed description of the numerical framework, we refer to [4].

3.1 Temporal two-scale approach

Even for the simplified two-dimensional configuration considered in this work, a resolution of the micro-scale dynamics with a scale of milliseconds to seconds is unfeasible over the complete time interval of interest $[0, T_{\text{end}}]$, with T_{end} being several months up to a year. For instance, when considering a relatively coarse micro-scale time step of $\delta\tau = 0.02s$, the number of time steps required to simulate a time frame of a whole year would be $365 \cdot 86\,400 \cdot \frac{1s}{\delta\tau} \approx 1.58 \cdot 10^9$, each step corresponding to the solution of a mechano-chemical FSI problem.

As a remedy, we apply the two-scale approach of Frei and Richter developed in [6], which can be seen as a variant of the Heterogeneous Multiscale Method (HMM) [3]. In particular, in [6], a periodic-in-time micro-scale problem is solved in each time step of the macro scale, for instance each day. The growth function $\gamma(\boldsymbol{\sigma}_f^{WS})$ is then averaged by integrating over one period of the heart beat, its average will be denoted by $\overline{\gamma}(\boldsymbol{\sigma}_f^{WS})$. This averaged growth function is applied to advance the foam cell concentration based on eq. (3).

Firstly, we divide the macro-scale time interval $[0, T_{\text{end}}]$ into N_l time steps of size δt

$$0 = t_0 < t_1 < ... < t_{N_l} = T, \qquad N_l = \frac{T_{\text{end}}}{\delta t}. \tag{9}$$

Since c_s varies significantly on the macro scale only, using $c_s(t_m)$ as a fixed value for the growth variable on the micro scale is a reasonable approximation in the time interval $[t_m, t_{m+1}]$. Then, one cycle of the pulsating blood flow problem (around 1 s) is to be resolved on the micro scale

$$0 = \tau_0 < \tau_1 < .... < \tau_{N_s} = 1s, \qquad N_s = \frac{1s}{\delta\tau} \tag{10}$$

to compute the averaged growth $\overline{\gamma}(\boldsymbol{\sigma}_f^{WS})$. It has been shown in [6] (for a simplified flow configuration) that this approach leads to a model error $\mathcal{O}(\epsilon)$ compared to a full resolution of the micro scale, where $\epsilon = \frac{1s}{T_{\text{end}}}$ denotes the ratio between macro and micro time scale and is in the range of $\mathcal{O}(10^{-7})$ for a typical cardiovascular plaque growth problem. From relations (9) and (10), we have $\delta\tau = \epsilon\delta t$. We note that, in the model problem formulated above, scale separation is induced by the parameter $\alpha = \mathcal{O}(\epsilon)$.

Within the two-scale algorithm, periodic micro-scale problems need to be solved. If suitable initial conditions for $\mathbf{w}^0 := (\mathbf{v}^0, \mathbf{u}^0)$ on the micro-scale are available, for example, from the micro scale problem from the previous macro time step, convergence to the periodic solution may be obtained by simulating a few cycles of the micro-scale problem, due to the dissipation of the flow problem; see [6, 5]. After each cycle, we can check numerically if the solution is sufficiently close to a periodic state. In particular, we apply a stopping criterion based on the computed averaged growth value:

$$|\overline{\gamma}(\boldsymbol{\sigma}_f^{WS,r}) - \overline{\gamma}(\boldsymbol{\sigma}_f^{WS,r-1})| < \epsilon_p,$$

where $r = 1, 2, \dots$ denotes the iteration index with respect to the number of cycles of the micro-scale problem. The two-scale algorithm is summarized in algorithm 1.

Algorithm 1: Two-scale algorithm

Set starting values $\mathbf{w}^{0,0} = (\mathbf{v}^{0,0}, \mathbf{u}^{0,0})$ and time step sizes $\delta t, \delta T$, $N_s = \frac{1}{\delta t}$, $N_l = \frac{T_{\text{end}}}{\delta T}$.

for $n = 1, 2, \dots, N_l$ **do**

    1.) **Micro problem**: Set $r \leftarrow 0$

        while $|\overline{\gamma}(\boldsymbol{\sigma}_f^{WS,r}) - \overline{\gamma}(\boldsymbol{\sigma}_f^{WS,r-1})| > \epsilon_p$ **do**

            1.a) Solve micro-scale problem (1) in $I_n = (t_n, t_n + 1s)$

$$\{\mathbf{w}^{r,0}, c_s^{n-1}\} \mapsto \{\mathbf{w}^{r,m}\}_{m=1}^{N_s}$$

            1.b) Compute the averaged growth function

$$\overline{\gamma}(\boldsymbol{\sigma}_f^{WS,r+1}) = \frac{1}{N_s} \sum_{m=1}^{N_s} \gamma(\boldsymbol{\sigma}_f^{WS,m}(\mathbf{v}^{r+1,m}), c_s^{n-1})$$

          and set $\mathbf{w}^{r+1,0} = \mathbf{w}^{r,N_s}$, $r \leftarrow r + 1$.

    2.) **Macro problem**: Update the foam cell concentration c_s^n by eq. (3).

We note that the macro problem in step 2.) is computationally very cheap since it consists of one time step of an ordinary differential equation. Before each macro step, however, a micro-problem needs to be solved in step 1.a) of algorithm 1. The solution of this micro problem is typically extremely expensive, as a time-dependent FSI problem needs to be solved. Considering a relatively coarse micro-scale discretization of $\delta t = 0.02\,\text{s}$, as the one used in section 6, 50 time steps are necessary to compute a single period of the heart beat. The simulation of two or more

cycles is typically necessary to obtain a near-periodic state, that fulfills the stopping criterion in 1); see [6, 5]. In a realistic scenario, each time step of the micro problem corresponds to the solution of a complex three-dimensional FSI problem, which already makes the solution of a single micro problem (i.e., ≥ 100 FSI time steps) very costly. We will thus in the following estimate the computational cost only by means of the number of micro problems that need to be solved. Even when a PDE of convection-diffusion-reaction type is considered in the plaque growth model instead, the cost of the micro problems dominates the overall computational work by far; see the discussion and the numerical results in [4].

4 Parallel time stepping

In the preprint [4], we have introduced parallel time stepping algorithms for the macro problem based on the parareal algorithm to further reduce the computational times. It turns out that the coarse propagator limits the maximum possible speed-up severely. A first algorithm to reduce the cost of the coarse propagator by re-using growth values from previous micro-scale computations has already been proposed in [4]; see also section 4.2. In this section, we will first review the developments in [4] and then propose a different approach to reduce the cost of the coarse propagator, based on an interpolation of the previously computed growth values.

4.1 The parareal algorithm

First, the time interval of interest $[0, T_{\text{end}}]$ on the macro scale is divided into P coarse sub-intervals $I_p = [T_{p-1}, T_p]$ of equal size. In the setting of parallelization in time, the sub-intervals are distributed among P parallel processes $p = 1, ..., P$:

$$0 = T_0 < T_1 < ... < T_P = T_{\text{end}}. \tag{11}$$

In order to define the parareal algorithm, suitable fine- and coarse problems need to be introduced. Following the work [4], we will apply the parareal algorithm only on the macro scale; hence, both the fine and the coarse scale of the parareal algorithm correspond to the macro scale of the homogenization approach. The fine problem advances the growth variable c_s from time T_p to T_{p+1} by applying the two-scale algorithm (algorithm 1) with a fine time step size δt (e.g. 0.3 days) on the corresponding fine time discretization of the interval $[T_p, T_{p+1}]$:

$$T_p = t_{p,0} < t_{p,1} < ... < t_{p,N_p} = T_{p+1}, \qquad N_p = \frac{T_{p+1} - T_p}{\delta t}, \qquad t_{p,q} := t_{p \cdot N_p + q} = t_{p,q-1} + \delta t.$$

We use capital letters T_p to denote the coarse discretization of $[0, T_{\text{end}}]$ into P parts corresponding to the processes and small letters t_i to denote the finer discretization on the parallel processes; the two discretizations yield the first level (fine problem) and second level (coarse problem) of the parareal algorithm. In the example studied in section 6, the time interval has length $T_{\text{end}} = 300$ days, δT is 30 days for $P = 10$, while δt is chosen as 0.3 days. On the micro scale, the times τ_i and time step size $\delta \tau$ are defined locally in $[t_i, t_i + 1s]$. In particular, we will use $\delta \tau = 0.02\,\text{s}$ here. Note that the micro scale influences the parareal algorithm only indirectly through the temporal two-scale approach.

On each process $p = 1, \ldots, P$, the fine propagator p consists of a time stepping procedure to

advance $c_s(T_p)$ to $c_s(T_{p+1})$ with the fine time step size δt. We abbreviate this computation by

$$c_s(T_{p+1}) = \mathcal{F}(c_s(T_p)).$$

In order to obtain parallel scalability, the coarse propagator needs to be much cheaper, since it advances the foam cell concentration globally on $[0, T_{\text{end}}]$ and thus introduces synchronization. A natural choice is to use the same two-scale algorithm as for the micro problems, but with a much larger time-step size $\delta T = T_{p+1} - T_p$. This means that in total P macro steps are required to solve the coarse problem on $[0, T_{\text{end}}]$, which includes the solution of P micro-scale FSI problems. We introduce the notation

$$c_s(T_{p+1}) = \mathcal{C}(c_s(T_p))$$

for one time step of this coarse propagator.

Then, given an iterate $\bar{c}_s^{(k)}$ for some $k \geq 0$, the parareal algorithm computes $\bar{c}_s^{k+1}$ by setting

$$\bar{c}_s^{(k+1)}(T_{p+1}) = \mathcal{C}(\bar{c}_s^{(k+1)}(T_p)) + \mathcal{F}(\bar{c}_s^{(k)}(T_p)) - \mathcal{C}(\bar{c}_s^{(k)}(T_p)) \quad \text{for } p = 0, ..., P-1. \quad (12)$$

As the fine-scale contributions depend only on the previous iterate $\bar{c}_s^{(k)}$ the contributions from each process $p = 1, \ldots, P$ can be computed in parallel.

Let us discuss the computational cost of the parareal algorithm in more detail. As mentioned above, the dominant part of the computational cost in the two-scale algorithm (algorithm 1) corresponds to the solution of the micro problems. The first term in eq. (12) requires the solution of one micro problem in each coarse time step $T_p \to T_{p+1}$. Within the fine-scale propagator (second term in eq. (12)) a maximum of $N_p = \lceil N_l/P \rceil$ time steps need to be computed per process, where $\lceil f \rceil$ denotes the next-biggest natural number to f ($\hat{=}$ `ceil(f)`). Each time step comes with the solution of one micro problem. The last term in eq. (12) has already been computed in the previous iteration (compare the first term on the right-hand side of the same equation) and therefore introduces no additional computational cost. Considering that all fine-scale problems ($p = 1, \ldots, P$) can be solved in parallel and assuming that each micro problem requires approximately the same time, the computing time for k_{par} iterations of the parareal algorithm on P processes corresponds to

$$\underbrace{k_{\text{par}} \cdot \lceil N_l/P \rceil}_{\text{fine level } (P \text{ parallel processes})} + \underbrace{(k_{\text{par}} + 1) \cdot P}_{\text{coarse level } (1 \text{ serial process})} \quad (13)$$

micro problems in a serial computation.

4.2 Re-Usage of Growth Values

Once the number of processes exceeds $P \geq \sqrt{N_l}$, the cost of the coarse-scale propagator gets dominant; cf. eq. (13). In [4], we propose an alternative coarse-scale propagator which avoids the solution of micro problems. Instead, we use the values $\bar{\gamma}(\boldsymbol{\sigma}_f^{WS}(t_{p,i}))$ computed within the fine-scale propagators. For this purpose, all values $\bar{\gamma}_{p \cdot N_p + i} := \bar{\gamma}(\boldsymbol{\sigma}_f^{WS}(t_{p,i}))$ computed on the fine scale on all processes $p = 1, ..., P$ are stored for all time steps $i = 1, ..., N_p$ and re-used within the coarse propagator.

Since no new micro problems need to be solved and approximations of the growth values $\overline{\gamma}_j$ are available for all fine time steps $j = 1, ..., N_l$, the coarse propagator can even be applied with the fine-scale time step size δt. The only additional cost is to advance the foam cell concentration by the ODE (3), which is clearly negligible compared to the solution of the micro problems in the fine-scale propagator. The only coarse-scale propagator, which can not be replaced, is the first one which is used for initialization. The total runtime for k_{par} iterations of the re-usage variant is

$$\underbrace{k_{\mathrm{par}} \cdot \lceil N_l/P \rceil}_{\text{fine level } (P \text{ parallel processes})} + \underbrace{P}_{\text{coarse level } (1 \text{ serial process})} \tag{14}$$

serial micro problems. Compared to eq. (13), this approach saves the computation of $k_{\mathrm{par}}P$ micro problems on the coarse scale; however, since it uses an approximation for the growth values, the number of parareal iterations k_{par} might be slightly higher. For details on the resulting algorithm, we refer to [4].

5 Interpolation Approach

The drawback of the re-use approach is that the growth values $\overline{\gamma}(\boldsymbol{\sigma}_f^{WS})(t_j)$ are computed based on the previous iterate of the foam cell concentration $c_s^{(k)}(t_j)$ instead of the parareal iterate $\overline{c}_s^{(k+1)}(t_j)$, which would be the most accurate approximation available in the coarse propagator. While we can expect that these are quite close after a few iterates, we have observed in [4] that this is not necessarily the case in the very first parareal iterate. Thus, typically a few more parareal iterations are required to reach a stopping criterion.

Here, we propose another approach, which may lead to better approximations of the growth values $\overline{\gamma}(\boldsymbol{\sigma}_f^{WS})(t_j)$. In particular, we store again the pairs of computed foam cell concentrations and growth values $(c_s^{(k)}(t_i), \overline{\gamma}(\boldsymbol{\sigma}_f^{WS}(t_i)))$ over all iterates k. In contrast to the re-usage variant, we interpolate linearly from the two values which are closest to $\overline{c}_s^{(k+1)}(t_j)$. For this purpose, we create an ordered list of values $\{c_s^i\}_{i=1}^N$ and their corresponding growth values $\{\gamma_{\mathrm{int}}^i\}_{i=1}^N$ *on the fly* during the parareal iterations and compute the linear interpolation from the values that fulfill $c_s^i < \overline{c}_s^{(k+1)}(t_j) < c_s^{i+1}$ when this is needed:

$$\overline{\gamma}_{\mathrm{int}}(\boldsymbol{\sigma}_f^{WS})(t_j) := \frac{c_s^{i+1} - \overline{c}_s^{(k+1)}(t_j)}{c_s^{i+1} - c_s^i}\gamma_{\mathrm{int}}^i + \frac{\overline{c}_s^{(k+1)}(t_j) - c_s^i}{c_s^{i+1} - c_s^i}\gamma_{\mathrm{int}}^{i+1}; \tag{15}$$

The resulting algorithm is given in algorithm 2. As in the re-usage variant no micro-problems need to be solved within the coarse-scale propagator. The total runtime corresponds thus again to the serial solution of $k_{\mathrm{par}} \cdot \lceil N_l/P \rceil + P$ micro problems and the optimal choice in terms of speed-up is again $P \approx \sqrt{k_{\mathrm{par}}N_l}$; see the discussion in Section 4.2.2 in [4]. However, compared to the re-usage algorithm, we expect that fewer iterations of the parareal algorithm k_{par} are required to fulfill the stopping criterion.

This approach can be regarded as a simple linear reduced order model for the micro problems. Since the interpolation data is updated during the parareal iterations, the model predictions may actually improve from iteration to iteration.

Algorithm 2: Parallel time stepping with interpolation of growth values

---

(I) **Initialization**: Compute $\left\{(\overline{c}_s^{(0)}(T_p), \overline{\mathbf{w}}^{(0)}(T_p)\right\}_{p=1}^P$ by means of algorithm 1 with a coarse macro time step $\delta t := T_{p+1} - T_p$ on the master process. Set $k \leftarrow 0$

(II) **while** $|c_s^{(k+1)}(T_{end}) - c_s^{(k)}(T_{end})| > \epsilon_{par}$ **do**

  (II.a) **Fine problem** on each process $p = 1, ..., P$:

   (i) Initialize $c_s^{(k+1)}(T_p) = \overline{c}_s^{(k)}(T_p)$, $\mathbf{w}^{(k+1)}(T_p) = \mathbf{w}^{(k)}(t_{p-1,N_p})$

   (ii) Compute $\left\{(c_s^{(k+1)}(t_{p,q}), \mathbf{w}^{(k+1)}(t_{p,q})\right\}_{q=1}^{N_p}$ by algorithm 1 with fine time step δt

   (iii) Add the pair $(c_s^{(k+1)}(t_{p,q}), \overline{\gamma}(\boldsymbol{\sigma}_f^{WS})(t_{p,q}))$ to the list of growth value pairs and order it by the value of its first component

  (II.b) **Coarse problem** on the master process:
  for $j = 0, ..., N_l - 1$ **do**

   (i) Compute the interpolated growth value $\overline{\gamma}_{\text{int}}(t_j)$ corresponding to $\overline{c}_s^{(k+1)}(t_j)$ by eq. (15)

   (ii) Advance $\overline{c}_s^{(k+1)}(t_{j+1}) = \overline{c}_s^{(k+1)}(t_j) + \delta t \, \overline{\gamma}_{\text{int}}(t_j)$.

---

6   Numerical Results

We show numerical results for a numerical example taken from [4]; see fig. 1 for a visualization of the configuration. The material parameters are chosen close to those of a realistic plaque growth problem; for details on the choice of the parameters, we refer to [4]. As an inflow boundary condition, a pulsating velocity inflow profile is prescribed on Γ_f^{in} .

For time discretization, we use a backward Euler method for the FSI problem and a forward Euler scheme for the ODE growth model. For spatial discretization, we use biquadratic (Q_2) equal-order finite elements for all variables and LPS stabilization [1] for the fluid problem. Our mesh, containing both fluid and solid, consists of 160 rectangular grid cells; this corresponds to relatively small FSI problem with a total of 3 157 degrees of freedom. The time step sizes are chosen as $\delta\tau = 0.02\,\text{s}$ and $\delta t = 0.3$ days (i.e., $N_l = 1\,000$); the tolerance for periodicity of the micro-scale problem is chosen as $\epsilon_p = 10^{-3}$. All the computational results have been obtained with the finite element library Gascoigne3d [2] using a fully monolithic approach for the FSI problem following Frei, Richter & Wick [7, 9]. All computations are performed in serial, and the parallel performance is discussed based on the computing time measured in terms of serial micro problems, as discussed in section 4. A visualization of the plaque growth at the end time T_{end} is shown in Figure 2.

In Tables 1, 2 and 3, we show results for the standard parareal algorithm as well as the re-usage and the interpolation variants, respectively. In all cases, the stopping criterion

$$|c_s^{(k+1)}(T_{\text{end}}) - c_s^{(k)}(T_{\text{end}})| < \epsilon_{\text{par}} = 10^{-3} \tag{16}$$

is used and numerical results for varying numbers of parallel processes P are provided.
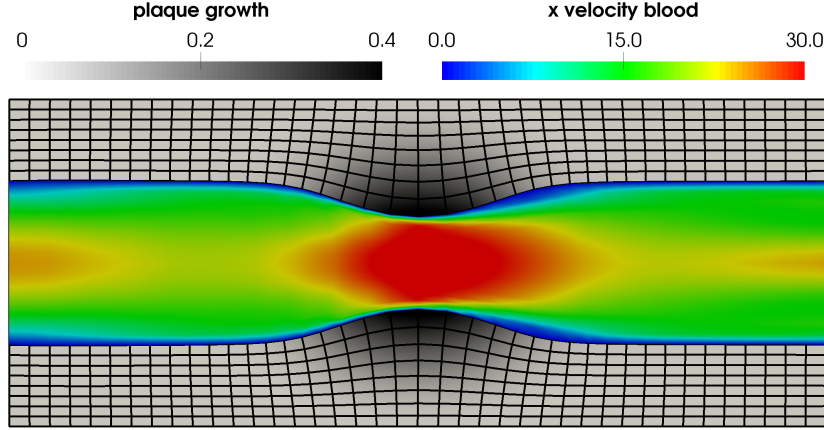
Figure 2: Visualization of the plaque growth at time $t = 300$ days. The horizontal velocity (in cm/s) and the vertical displacement (in cm) are shown on the deformed domain at micro time $\tau = 0.34\,\mathrm{s}$.

We observe that the standard parareal algorithm (table 1) needs 3-4 parareal iterations to fulfill the stopping criterion. The re-usage approach (table 2) needs 1-2 additional iterations, as a result of the approximation used in the coarse-scale propagator. Interestingly, for the interpolation variant, the values $c_s^{(1)}(T_{\mathrm{end}})$ on all processes agree with the reference computation in the first 8 digits already after only a single parareal iteration (the reference computation is performed with a serial time stepping). A second iterate is only computed, as the stopping criterion is based on a comparison between two subsequent iterates.

This might be surprising at first sight, as the interpolation in the coarse propagator is in principle also an approximation. On the other hand, the time step size $\delta t \ll \delta T$ is much smaller compared to the standard parareal coarse propagator. In fact, the interpolation between the values from the coarse initialisation (step (I) of algorithm 2) yields very accurate growth values in this example, such that the coarse propagator is both much cheaper and much more accurate compared to the coarse problem in the standard parareal algorithm. Of course, this might be partly attributed to the simple nature of the FSI and plaque growth problems. However, it shows that a reduced order model can be a powerful alternative to the standard parareal coarse propagator.

Due to the lower number of parareal iterations, the number of micro problems to be solved in the interpolation approach is significantly smaller compared to the other approaches for all processes. The minimal number of micro problems is attained for $P = 40$ and $P = 50$, where only 90 micro problems ($2 \cdot 25 + 40$ resp. $2 \cdot 20 + 50$) need to be solved. This corresponds to an expected speed up of 11.1 compared to a serial computation. As discussed above, the optimal number of processes is $P \approx \sqrt{k_{\mathrm{par}} N_l} \approx 44.72$. If one would stop the computation already after the first iterate, which is already accurate up to 8 digits, the computational cost would reduce further to 65 (for $P = 40$) resp. 70 (for $P = 50$) micro problems; however, this would require the derivation of an error estimator instead of the stopping criterion eq. (16).

k	$P = 10$	$P = 20$	$P = 30$	$P = 40$	$P = 50$	ref. (serial)
1	$2.21 \cdot 10^{-2}$	$1.19 \cdot 10^{-2}$	$8.12 \cdot 10^{-3}$	$5.83 \cdot 10^{-3}$	$5.46 \cdot 10^{-3}$	0.63831273
2	$2.24 \cdot 10^{-3}$	$5.63 \cdot 10^{-4}$	$2.62 \cdot 10^{-4}$	$1.37 \cdot 10^{-4}$	$8.61 \cdot 10^{-5}$	-
3	$1.42 \cdot 10^{-4}$	$2.02 \cdot 10^{-5}$	$6.53 \cdot 10^{-6}$	$2.32 \cdot 10^{-6}$	$8.30 \cdot 10^{-7}$	-
4	$5.76 \cdot 10^{-6}$	-	-	-	-	-
# mp	450	230	**222**	235	260	1 000
speedup	2.2	4.3	**4.5**	4.3	3.8	1.0
efficiency	**22 %**	**22 %**	15 %	11 %	8 %	100 %

Table 1: Errors $|c_s^{(k)}(T_{\text{end}}) - c_s^*(T_{\text{end}})|$ for $P = 10, \dots, 50$ for the standard parareal algorithm and a reference computation c_s^*. The time measure in terms of the number of serial micro problems (# mp) as well as speedup and efficiency compared to the reference computation (right column) are shown; best numbers marked in **bold face**.

k	$P = 10$	$P = 20$	$P = 30$	$P = 40$	$P = 50$	$P = 60$	$P = 70$	ref.
1	$2.56 \cdot 10^{-2}$	$1.10 \cdot 10^{-2}$	$7.48 \cdot 10^{-3}$	$5.46 \cdot 10^{-3}$	$4.23 \cdot 10^{-3}$	$3.63 \cdot 10^{-3}$	$3.12 \cdot 10^{-3}$	0.63831273
2	$7.78 \cdot 10^{-3}$	$4.48 \cdot 10^{-3}$	$3.16 \cdot 10^{-3}$	$2.36 \cdot 10^{-3}$	$1.99 \cdot 10^{-3}$	$1.61 \cdot 10^{-3}$	$1.41 \cdot 10^{-3}$	-
3	$1.73 \cdot 10^{-3}$	$1.20 \cdot 10^{-3}$	$9.02 \cdot 10^{-4}$	$6.92 \cdot 10^{-4}$	$6.34 \cdot 10^{-4}$	$5.14 \cdot 10^{-4}$	$4.32 \cdot 10^{-4}$	-
4	$2.32 \cdot 10^{-4}$	$2.28 \cdot 10^{-4}$	$1.87 \cdot 10^{-4}$	$1.31 \cdot 10^{-4}$	$1.26 \cdot 10^{-4}$	$1.10 \cdot 10^{-4}$	$9.98 \cdot 10^{-5}$	-
5	$2.19 \cdot 10^{-5}$	$4.71 \cdot 10^{-5}$	$3.05 \cdot 10^{-5}$	-	-	-	-	-
# mp	510	270	200	140	130	**128**	130	1 000
speedup	2.0	3.7	5.0	7.1	7.7	**7.8**	7.7	1.0
eff.	**20 %**	19 %	17 %	18 %	15 %	13 %	11 %	100 %

Table 2: Errors $|c_s^{(k)}(T_{\text{end}}) - c_s^*(T_{\text{end}})|$ for $P = 10, \dots, 70$ for the re-usage variant, best numbers marked in **bold face**.

The efficiency, defined as ratio between speedup and number of processes, is also significantly improved compared to the previous approaches. For $P = 10$, we obtain a parallel efficiency of 48 %, and for $P = 40$, where a speedup of ≈ 11 is obtained, the efficiency is still 28 %. For the other two approaches the corresponding efficiencies are 20% resp. 22% and 11% resp. 18%.

7 Conclusion

We have introduced an efficient coarse propagator for a scalar plaque growth model based on an interpolation of previously computed growth values. For the simple model example considered in this work, the modified parareal algorithm converges extremely fast against a reference value. Obviously, the performance needs to be confirmed in more complex examples in future.

A realistic plaque growth model typically consists of a system of partial-differential equations. In this case, an interpolation of the map $c_s \to \overline{\gamma}(c_s)$ is not straight-forward, as both c_s and $\overline{\gamma}(c_s)$ are spatially distributed quantities. In this case, more sophisticated reduced order models or neural networks may be used to approximate this map. This will be the subject of future research.

k	$P = 10$	$P = 20$	$P = 30$	$P = 40$	$P = 50$	$P = 60$	ref. ($P = 1\,000$)
1	$8.3 \cdot 10^{-9}$	$3.0 \cdot 10^{-8}$	$2.5 \cdot 10^{-8}$	$9.9 \cdot 10^{-9}$	$3.6 \cdot 10^{-9}$	$5.2 \cdot 10^{-9}$	0.6383127317
2	$3.0 \cdot 10^{-10}$	$4.3 \cdot 10^{-9}$	$1.6 \cdot 10^{-9}$	$8.0 \cdot 10^{-10}$	$1.2 \cdot 10^{-9}$	$1.7 \cdot 10^{-9}$	-
# mp	210	120	98	**90**	**90**	94	1 000
speedup	4.8	8.3	10.2	**11.1**	**11.1**	10.6	1.0
efficiency	**48 %**	42 %	34 %	28 %	22 %	18 %	100 %

Table 3: Errors $|\bar{c}_s^{(k)}(T_{\text{end}}) - c_s^*(T_{\text{end}})|$ for $P = 10, \ldots, 60$ for the interpolation approach; best numbers marked in **bold face**.

REFERENCES

[1] R. Becker and M. Braack. A finite element pressure gradient stabilization for the Stokes equations based on local projections. *Calcolo*, 38(4):173–199, 2001.

[2] R. Becker, M. Braack, D. Meidner, T. Richter, and B. Vexler. The finite element toolkit Gascoigne3d. http://www.gascoigne.de.

[3] W. E and B. Engquist. The heterogenous multiscale method. *Comm. Math. Sci.*, 1(1):87–132, 2003.

[4] S Frei and A Heinlein. Towards parallel time-stepping for the numerical simulation of atherosclerotic plaque growth. *arXiv e-print: https://arxiv.org/abs/2203.06526*, 2022.

[5] S Frei, A Heinlein, and T Richter. On temporal homogenization in the numerical simulation of atherosclerotic plaque growth. *PAMM*, 21(1):e202100055, 2021.

[6] S. Frei and T. Richter. Efficient approximation of flow problems with multiple scales in time. *SIAM J Multiscale Model Simul*, 18(2):942–969, 2020.

[7] S. Frei, T. Richter, and T. Wick. Long-term simulation of large deformation, mechano-chemical fluid-structure interactions in ALE and fully Eulerian coordinates. *J Comput Phys*, 321:874 – 891, 2016.

[8] J.-L. Lions, Y. Maday, and G. Turinici. A "parareal" in time discretization of PDE's. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 332:661–668, 2001.

[9] T. Richter. *Finite Elements for Fluid-Structure Interactions. Models, Analysis and Finite Elements.*, volume 118 of *Lecture Notes in Comput Sci and Eng*. Springer, 2017.

[10] E.K. Rodriguez, A. Hoger, and A. D. McCulloch. Stress-dependent finite growth in soft elastic tissues. *J. Biomech.*, 4:455–467, 1994.

[11] T. Silva, W. Jäger, M. Neuss-Radu, and A. Sequeira. Modeling of the early stage of atherosclerosis with emphasis on the regulation of the endothelial permeability. *J Theor Biol*, 496:110229, 2020.

[12] Y. Yang, W. Jäger, M. Neuss-Radu, and T. Richter. Mathematical modeling and simulation of the evolution of plaques in blood vessels. *J Math Biol*, pages 1–24, 2014.