Parareal Method for Anisotropic Diffusion Denoising

Shan, Xiujie; van Gijzen, Martin B.

DOI
10.1007/978-3-031-30445-3_26

Publication date
2023

Document Version
Final published version

Published in
Parallel Processing and Applied Mathematics - 14th International Conference, PPAM 2022, Revised Selected Papers

Parareal Method for Anisotropic Diffusion Denoising

Xiujie Shan(✉) and Martin B. van Gijzen

Delft Institute of Applied Mathematics, Delft University of Technology, 2628 CD Delft, The Netherlands
xiujieshan@gmail.com

Abstract. This paper studies time-domain parallelisation using Parareal to speed up the computations of anisotropic diffusion filtering. We consider both explicit and implicit Euler based method for the propagation in time for Parareal. The Preconditioned Conjugate Gradient (PCG) method is used to solve the systems that arise in the implicit based method. The estimation of the iteration numbers of PCG allows us to predict the running time of Parareal calculation, which further guides us in the experimental stage. Parallelisation of the method is implemented using Coarray Fortran. We illustrate the experimental results on 3D low-field MRI images using up to 960 cores. The computational improvement in time is achieved.

Keywords: Image denoising · Parareal · Nonlinear Diffusion equation · High performance computing

1 Introduction

The nonlinear diffusion equation has many applications, and one of the most important is image denoising. The classical paper by Perona and Malik [18] proposed image denoising by considering denoising as a diffusion process, in which the diffusion parameter is chosen such that edges are preserved. Computationally this amounts to integrating time into a nonlinear diffusion equation. This process can be quite expensive. An option to speed up the computations is to use a parallel-in-time integration method.

Parallelisation in time was first considered by Nievergelt [16], to make full use of the potential of massively parallel computers. Parareal was proposed by Lions, Maday, and Turinici in 2001 [15]. The method was a real breakthrough, and it is now one of the most widely used parallel-in-time methods, particularly for the time discretization of partial differential evolution equations. A further concise version of the method, which is also commonly used now, was given in [3]. In [2], Bal and Maday gave the convergence analysis of Parareal for the heat equation and also the application to a nonlinear partial differential equation for pricing of an American put. Maday and others have been working on this topic continuously. Others developed a Parareal version for nonlinear PDEs [2] and a stable Parareal method for first- and second-order hyperbolic systems [5].

R. Wyrzykowski et al. (Eds.): PPAM 2022, LNCS 13827, pp. 313–322, 2023.
https://doi.org/10.1007/978-3-031-30445-3_26

Analysis of the Parareal algorithm has provided new insights into the relations with other algorithms, which has led to new parallel-in-time algorithms. In 2007, Gander and Vandewalle [12] analyzed the connection between the Parareal algorithm, space-time multigrid, and the multiple shooting methods. A historical review of Parareal can be found in [12] as well. Gander et al. further explored this topic in [11], and interpreted the Parareal algorithm as a multigrid method which led to the multigrid-reduction-in-time (MGRIT) algorithm. By considering Parareal a preconditioned iterative process, where the coarse time integration method acts as the preconditioner, the authors extended Parareal (two-level) to a multi-level method in 2014 [8]. Wu [24] proposed a parallel coarse grid correction diagonalization technique and analyzed the convergence rate of this method. Gander and Wu developed a diagonalization-based Parareal algorithm for dissipative and wave propagation problems. A new Parareal algorithm for ODEs with discontinuous source in time has been proposed [10]. By defining a smooth input according to the coarse discretization, the authors illustrate that the coarse propagator can capture a highly oscillatory or discontinuous source in time.

The research about domain decomposition methods for image denoising includes successive space correction methods, parallel space correction methods, etc. [4]. These methods aim to do parallel computing by decomposing the image (space domain) and they have been used to solve some convex minimization problems, and variational inequalities with the convex set constraint [7,9,25]. The Parareal algorithm to solve the anisotropic diffusion denoising belongs to a different category, in which the algorithm is designed to do the parallel calculation in the time domain. We consider both explicit time and implicit based integration methods for the coarse and fine grids. The implicit Euler based method requires solving an extensive nonlinear system at every time step. To linearise the equation, we compute the diffusion based on the solution of the previous time step. The resulting linear system is solved with preconditioned conjugate gradient (PCG) method. We refer to [22] for the details of this solver and the preconditioners. We will analyse the possible speed-up for both the explicit and implicit method by estimating the computing time per Parareal iteration, using the upper bounds on the number of CG iterations provided in [22]. The results are validated by numerical experiments using up to 960 cores of the DelftBlue supercomputer of the Delft University of Technology [6].

Our paper is organized as follows. Section 2 presents the denoising model and its numerical discretization. Section 3, discusses Parareal to integrate the equations in time. Section 4 investigates potential speedup of Parareal. Section 5 presents the numerical experiments. Section 6 makes some concluding remarks.

2 Model and Discretization

This section describes the anisotropic diffusion denoising model and its numerical discretization in space and time.

2.1 Diffusion Model

The idea of using a diffusion equation for image denoising was first considered by Koenderink in [1] by connecting the linear heat equation to the Gaussian filter. The nonlinear diffusion model we use was proposed by Peron and Malik in 1990 in [18] and is given by

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla \cdot (c(\|\nabla u\|_2)\nabla u), & \text{in} \quad \Omega \times (0,T), \\ \dfrac{\partial u}{\partial \boldsymbol{n}} = 0, & \text{on} \quad \partial\Omega \times (0,T), \\ u(0) = \hat{u}_0, & \text{in} \quad \Omega, \end{cases} \tag{1}$$

where $\Omega \subseteq \mathbb{R}^d$ for $d = 2,\ 3$. Choices for the diffusion coefficient are given by $c_1(\|\nabla u\|_2) = e^{-(\|\nabla u\|_2/K)^2}$, $c_2(\|\nabla u\|_2) = \frac{1}{1+\left(\frac{\|\nabla u\|_2}{K}\right)^2}$, where K is a damping parameter. The idea behind this nonlinear diffusion model is explained as follows. Since the edges of the image can be approximately estimated by $\|\nabla u\|_2$, diffusion coefficient $c(\|\nabla u\|_2)$ is also called as edge detector. $c(\|\nabla u\|_2) \to 0$ as $\|\nabla u\|_2 \to +\infty$, this means that in the neighbourhood of an edge (where $\|\nabla u\|_2$ is large), the diffusion coefficient is small, i.e., the diffusion is slow. Similarly, in a flat area, $c(\|\nabla u\|_2) \to 1$, when $\|\nabla u\|_2 \to 0$. This means that the nonlinear diffusion behaves like linear diffusion in a flat area, and noise is smoothed out quickly. Apart from image processing, this model also arises in other contexts, for example, faceted crystal growth [13] and continuum mechanics [14].

For the one dimensional space case, we consider $\Omega = [0,\ 1]$ and step size $h_x = \frac{1}{N_x}$, where N_x is the number of spatial grid points. $\frac{\partial}{\partial x}(c(|\frac{\partial u}{\partial x}|) \cdot \frac{\partial u}{\partial x})$ can be discretized as

$$\frac{\partial}{\partial x}(c(|\frac{\partial u}{\partial x}|) \cdot \frac{\partial u}{\partial x})_{x_i} \approx c_{i+\frac{1}{2}} \frac{(u_{i+1} - u_i)}{h_x^2} - c_{i-\frac{1}{2}} \frac{(u_i - u_{i-1})}{h_x^2},$$

where $c_{i\pm\frac{1}{2}} = \frac{c_{i\pm1} + c_i}{2}$. $c_i := c(|u_x|_i) = c(|\frac{u_{i+1} - u_{i-1}}{2h_x}|)$ for $0 \le i \le N_x - 1$. Because of the Neumann boundary conditions, we have that $u_{-1} = u_0$ and $u_{N_x-1} = u_{N_x}$. The discretization in space is given by

$$\frac{d\mathbf{u}}{dt} = \frac{1}{h_x^2} \begin{pmatrix} -c_{\frac{1}{2}} & c_{\frac{1}{2}} & & & & \\ c_{\frac{1}{2}} & -(c_{\frac{1}{2}} + c_{1+\frac{1}{2}}) & c_{1+\frac{1}{2}} & & & \\ & \ddots & \ddots & \ddots & & \\ & & c_{N-\frac{5}{2}} & -(c_{N-\frac{5}{2}} + c_{N-\frac{3}{2}}) & c_{N-\frac{3}{2}} & \\ & & & c_{N-\frac{3}{2}} & -c_{N-\frac{3}{2}} \end{pmatrix} \mathbf{u},$$

where $\mathbf{u} = (u_0, u_1, \ldots, u_{N-1})^T$. The higher-dimensional case can be discretised analogously.

2.2 Linearization and Time Discretization

After spatial discretisation we obtain a system of ordinary equations

$$\frac{d\mathbf{u}}{dt} = C(\mathbf{u})\mathbf{u}.$$

To integrate this system in time we consider the explicit and implicit Euler based method. Using explicit Euler method, we have for $m = 0, 1, 2, \ldots$

$$\frac{\mathbf{u}_{m+1} - \mathbf{u}_m}{\tau} = C(\mathbf{u}_m)\mathbf{u}_m. \tag{2}$$

We rewrite it into $\mathbf{u}^{m+1} = (\lambda(I + \tau C(\mathbf{u}^m)))\mathbf{u}^m$.

The implicit Euler based method follows $\frac{\mathbf{u}_{m+1}-\mathbf{u}_m}{\tau} = C(\mathbf{u}_{m+1})\mathbf{u}_{m+1}, m = 0, \ldots, N-1$. To linearise the right-hand side, we approximate $C(\mathbf{u}_{m+1})$ by $C(\mathbf{u}_m)$. With this modification the implicit Euler based method is given by

$$\left(\frac{1}{\tau}I - C(\mathbf{u}_m)\right)\mathbf{u}_{m+1} = \frac{1}{\tau}\mathbf{u}_m, \tag{3}$$

which is a linear system of the form

$$A\mathbf{u} = \tilde{\mathbf{f}}. \tag{4}$$

In every time iteration, we solve a linear system (4) by using the preconditioned Conjugate Gradient method (PCG) [20]. Since A is strongly diagonally dominant we use diagonal scaling (Jacobi preconditioner) as preconditioner, which means setting the preconditioned matrix M as the main diagonal elements of A.

3 Parareal Algorithm for the Anisotropic Diffusion Model

This section starts with explaining the idea of Parareal algorithm as a multiple shooting method [12] and giving the algorithm for solving model (1).

Divide the time interval $(0, T)$ into subintervals $I_n = (T_n, T_{n+1})$ of size ΔT, $n = 0, 1, \ldots P-1$. Then sub-interval I_n is decomposed further into smaller sub-interval with size δt. Now we consider solve the problem

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} = C(\mathbf{u})\mathbf{u}, & t \in (0, T), \\ u(0) = u_0. \end{cases} \tag{5}$$

The initial value problems on each coarse time intervals are given by

$$\begin{cases} \dfrac{\partial \mathbf{u}_n}{\partial t} = C(\mathbf{u}_n)\mathbf{u}_n, & t \in (T_n, T_{n+1}), \\ \mathbf{u}_n(T_n) = U_n \end{cases}$$

and the matching conditions are $U_0 = \mathbf{u}_0$ and $U_{n+1} = u_n(T_{n+1}, U_n)$.

These conditions compose a nonlinear system which we denote it by $\mathbf{F}(\mathbf{U}) = 0$, where $\mathbf{U} = (U_0, U_1, \ldots, U_P)^T$. Solving this system by Newton method leads to

$$\begin{cases} U_0^{k+1} = \mathbf{u}_0, \\ U_{n+1}^{k+1} = \mathbf{u}_n(T_{n+1}, U_n^k) + \dfrac{\partial \mathbf{u}_n}{\partial U_n}(T_{n+1}, U_n^k)(U_n^{k+1} - U_n^k), \end{cases}$$

where $k = 0,\ 1, \ldots$.

Approximating $\mathbf{u}_n(T_{n+1}, U_n^k)$ by the fine propagator $F(U_n^k)$ and $\frac{\partial \mathbf{u}_n}{\partial U_n}(T_{n+1}, U_n^k)(U_n^{k+1} - U_n^k)$ by the coarse propagators $G(U_n^{k+1}) - G(U_n^k)$, the recursion formula of the Parareal method is

$$\begin{cases} U_0^{k+1} = \mathbf{u}_0, \\ U_{n+1}^{k+1} = F(U_n^k) + G(U_n^{k+1}) - G(U_n^k), \end{cases}$$

For the explicit Euler scheme, we have $G(U_n) = (I + \Delta T \cdot C(U_n))U_n$ and $F(U_n) = (I + \delta t \cdot C(U_n))U_n$. For the implicit Euler based method, we have $G(U_n) = (I - \Delta T \cdot C(U_n))^{-1}U_n$ and $F(U_n) = ((I - \delta t \cdot C(U_n))^{-1})^{\frac{\Delta T}{\delta t}}U_n$.

One usual initial guess for U_{n+1}^0 is $G(U_n^0)$. As the iteration converges and $U_{n+1}^{k+1} - U_{n+1}^k \to 0$, the results from the coarse method $G(U_n^{k+1})$ and $G(U_n^k)$ will cancel out and Parareal will only reproduces the fine time solution. It has been proven in [12] that Parareal converges after a maximum of P iterations.

Algorithm 1. Parareal algorithm for solving the model

$U_0^0 \leftarrow \tilde{U}_0^0 \leftarrow \mathbf{u}_0$
for $n = 0$ to $P - 1$ **do**
  $\tilde{U}_{n+1}^0 \leftarrow G(\tilde{U}_n^0)$
  $U_{n+1}^0 \leftarrow \tilde{U}_{n+1}^0$
end for
$U_0^1 \leftarrow \mathbf{u}_0$
for $k = 0$ to $K_{max} - 1$ **do**
  for $n = 0$ to $P - 1$ **do** (parallel)
    $\hat{U}_{n+1}^k \leftarrow F(U_n^k)$
  end for
  for $n = 0$ to $P - 1$ **do**
    $\tilde{U}_{n+1}^{k+1} \leftarrow G(U_n^{k+1})$
    $U_{n+1}^{k+1} \leftarrow \hat{U}_{n+1}^k + \tilde{U}_{n+1}^{k+1} - \tilde{U}_{n+1}^k$ which equals to:
    $U_{n+1}^{k+1} \leftarrow F(U_n^k) + G(U_n^{k+1}) - G(U_n^k)$
  end for
  if $\frac{\|U_{n+1}^{k+1} - U_{n+1}^k\|_2}{\|\mathbf{u}_0\|_2} < \epsilon$ **then**
    BREAK
  end if
end for

4 Analysis of the Algorithm

The speed-up of Parareal has been estimated in [2]. The cost of the fine iteration method is proportional to $\frac{T}{\delta t}$. The computational cost of Parareal is proportional to $k(\frac{T}{\Delta T} + \frac{\Delta T}{\delta t})$. Fixing k, the cost of Parareal is optimal when $\frac{T}{\Delta T} = \frac{\Delta T}{\delta t}$. For the aim of comparison, keeping $\frac{T}{\delta t}$ fixed leads to $\Delta T = \sqrt{T\delta t}$. Therefore, the maximum gain in computational time is $S = \frac{1}{4}\sqrt{\frac{T}{\delta t}}$ for $k = 2$.

In our paper, for solving the problem with the implicit method, the iteration number of PCG can be bounded. Later we will use the bounds to estimate the algorithm's running time. Since $\max_i(|C_{i,i}|) \leq \frac{2d}{h^2}$, by using the Gershgorin's theorem, we have

$$\kappa_\tau \leq 1 + \frac{4d\tau}{h^2},$$

where τ is the time step and d is the space dimension. For the details we refer to [22]. The iteration number of PCG is given as

$$N_\tau = \ln\left(\frac{\epsilon}{2\sqrt{\kappa_\tau}}\right) / \ln\left(\frac{\sqrt{\kappa_\tau} - 1}{\sqrt{\kappa_\tau} + 1}\right). \tag{6}$$

4.1 Potential Speedup by Parareal (Ideal Case)

The total amount of calculation for the fine method with the time step δt is $\frac{T}{\delta t}$. By using the explicit Euler method as fine and coarse propagators, we have the speedup for Parareal method is:

$$S \approx \frac{\frac{T}{\delta_t}}{\left((k+1)\frac{T}{\Delta T} + k\frac{\Delta T}{\delta t}\right)},$$

where $k + 1$ is from the initialization of Parareal.

For implicit Euler based method with PCG as fine and coarse propagators, we have

$$S \approx \frac{\frac{T}{\delta_t} N_{\frac{T}{\delta_t}}}{\left((k+1)\frac{T}{\Delta T} N_{\frac{T}{\Delta_t}} + k\frac{\Delta T}{\delta t} N_{\frac{\Delta T}{\delta t}}\right)}, \tag{7}$$

where $N_{\{\cdot\}}$ stands for the PCG iterations for different time step. By fixing k, T and δ_t, we can search the maximal gain in time.

4.2 Potential Speedup by Parareal (with Communication Time)

Assuming that one communication time between fine and coarse is $T_{co} = T_{f2c} + T_{c2f}$, we have total communication time for Parareal is kT_{co}. By calculating the CPU time for one step time iteration T_f in the sequential case, we have the estimated CPU time for Parareal as $T_f\left((k+1)\frac{T}{\Delta T} + k\frac{\Delta T}{\delta t}\right)$. The total time for Parareal calculation for the explicit Euler in the super computer is then $T_{total} = T_f\left((k+1)\frac{T}{\Delta T} + k\frac{\Delta T}{\delta t}\right) + kT_{co}$. By denoting the CPU time for one PCG iteration as $T_{f_{cg}}$, we have $T_{total} = T_{f_{cg}}\left((k+1)\frac{T}{\Delta T} N_{\frac{T}{\Delta_t}} + k\frac{\Delta T}{\delta t} N_{\frac{\Delta T}{\delta t}}\right) + kT_{co}$ for the implicit Euler based method.

5 Experimental Results

This section will give the run time and image results tested by the denoising model. Image data and parameter choices for the Parareal algorithm are provided as well, including the CG iteration number, tolerance ϵ for Parareal, ΔT for coarse method, and δt for fine method.

The parallel implementation for the Parareal method is done in Fortran using Coarray Fortran (CAF). CAF follows the SPMD model. Each process (called image) has its private variables. Variables which have a so-called codimension are addressable from other images. We use CAF to implement the fine propagator in parallel on each coarse interval. We run the fine steps parallel in time indicating the number of cores.

The numerical tests have been performed on the DelftBlue supercomputer, which now has 228 Intel Xeon compute nodes with 48 cores each. We test a 3D melon image ($128 \times 128 \times 128$) scanned by low-field MRI machine [17] using model (choosing diffusion coefficient to be c_1 with $K = 20$) in Table 1 and Table 2. The fine time step is $1e^{-7}$ and the total number of time steps is 960, meaning that $T = 960 \times 1e^{-7}$. The relative tolerance for Parareal and PCG to converge is $1e^{-6}$.

Table 1 shows the results for explicit method. When the number of coarse steps equals 48, we have the lowest run time, with a speed-up of about two. The estimated time is obtained from the ideas described in Sect. 4.2 without communication time. This estimated time predicts well the optimal number of coarse steps for this example. The estimated times are consistently lower than the measured run times. This can be explained by the fact that we did not consider communication time in the estimated time.

The results for implicit Euler based method are tabulated in Table 2. The CG iterations we got from the experiments match the theoretical iterations in (6). We again observe that the estimated times predict well the optimal number of cores, which is 48, and again we see a speed-up of about a factor of two for the optimal number of cores. We do observe that, with for increasing number of cores, our estimation for the run time becomes too pessimistic. For this we do not have a satisfactory explanation yet (Fig. 1).

Table 1. Parareal times obtained for explicit Euler. ("Coarse" stands for coarse step $T/\Delta T$, "Parareal" the total Parareal iterations, "Elapsed time" the running time for the algorithm, "Total iterations" the fine and coarse iterations with Parareal iterations, "Estimated time" is calculated as T_{total}.)

Coarse	Parareal	Elapsed time	Total iterations	Estimated time
1	1	286	960	286.00
12	12	480	1116	332.48
48	4	127	320	95.33
96	3	136	414	123.34
192	3	243	783	233.27
480	2	449	1444	430.19
960	1	646	1921	572.30

Table 2. Parareal time obtained for implicit Euler based method. ("CG coarse" stands for the iterations of CG for one coarse propagation, "CG fine" iterations of CG for one fine propagation, "Total iterations" the CG iterations for coarse and fine propagations with Parareal iterations, "Estimated time" is calculated as T_{total}.)

Coarse	Parareal	CG coarse	CG fine	Elapsed time	Total iterations	Estimated time
1	1	2	2	1102	1920	1102
12	8	7	2	1413	2036	983
48	5	4	2	656	1352	578
96	4	3	2	690	1520	707
192	3	3	2	874	2334	1075
480	3	2	2	1882	3852	2211
960	1	2	2	1502	3842	2205

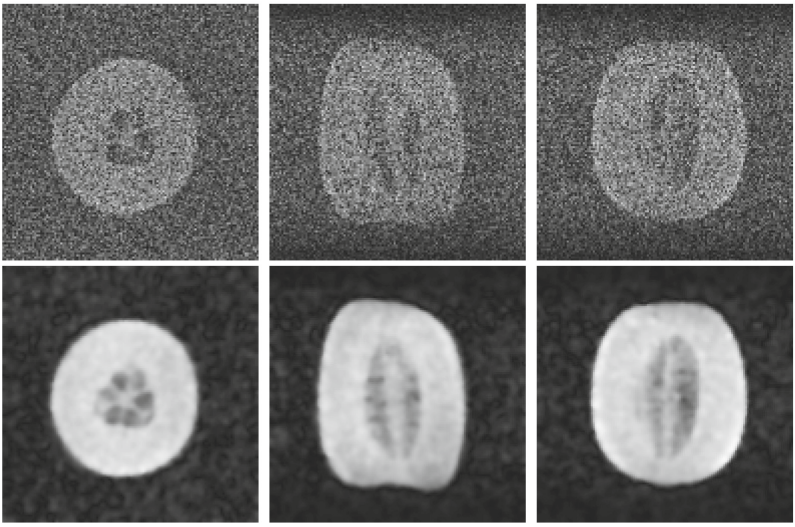


Fig. 1. Three slides from the 3D melon image, the model is with diffusion coefficient c_2 and $K = 15$. The total diffusion time is $2.4e^{-6}$.

6 Conclusions

In this paper, we have investigated the use of the Parareal method to speed up anisotropic diffusion filtering. The parallelisation in time can be done with only local modifications to the code, without the need to completely restructure the program. We have derived theoretical estimates for the run time that can be used to predict the optimal number of cores. A modest but useful speedup with a factor of two is obtained to denoise a 3D low-field MR image of a melon.

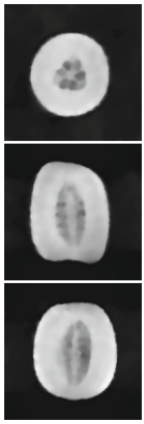
Acknowledgements. The authors thank the Leiden University Medical Center for providing the low-field MR image and the reviewers' comments to help improve the paper.

A Appendix

So far, we have focused on the classical denoising models as proposed in [18] by Perona and Malik. The total variational model proposed in [19] is a widely used alternative. In this appendix, we give numerical results for this technique. We use Parareal with the linearised implicit Euler method. As explained in [21], it is not possible to derive a useful upper bound on the number of CG iterations for the total variation model and make an a priori prediction for the optimal number of cores for this method. Solving the total variation model with gradient descent method equals solving (1) with $c(\|\nabla u\|_2) = \frac{1}{\|\nabla u\|_2}$. Following the idea of [23], we solve it with a lagged diffusivity fixed point iteration. For the numerical experiments, we use $\frac{1}{\|\nabla u\|+\epsilon}$ instead of $\frac{1}{\|\nabla u\|}$, where $\epsilon = 1e^{-5}$. One fine time step is $1.5e^{-6}$ and the total evaluation time is $7.2e^{-4}$ (Table 3).

Table 3. Parareal times obtained for implicit Euler. ("Coarse" stands for coarse step $T/\Delta T$, "Parareal" the total Parareal iterations, "Elapsed time" the running time for the algorithm.)

Coarse	Parareal	Elapsed time
1	1	2023.48
12	5	1580.85
48	2	1495.60
96	2	2717.97
240	1	2173.66
480	1	2263.87



References

1. The structure of images: biological cybernetics **50**, 363–370 (1984)
2. Bal, G., Maday, Y.: A "parareal" time discretization for non-linear PDE's with application to the pricing of an American put. In: Pavarino, L.F., Toselli, A. (eds.) Recent Developments in Domain Decomposition Methods, vol. 23, pp. 189–202. Springer, Berlin (2002). https://doi.org/10.1007/978-3-642-56118-4_12
3. Bal, G.: Parallelization in time of (stochastic) ordinary differential equations (2006)
4. Chang, H., Zhang, X., Xue-Cheng, T., Yang, D.: Domain decomposition methods for nonlocal total variation image restoration. J. Sci. Comput. **60**, 79–100 (2014). https://doi.org/10.1007/s10915-013-9786-9

5. Dai, X., Maday, Y.: Stable parareal in time method for first- and second-order hyperbolic systems. SIAM J. Sci. Comput. **35**(1), A52–A78 (2013)
6. Delft High Performance Computing Centre (DHPC): DelftBlue Supercomputer (Phase 1) (2022)
7. Duan, Y., Tai, X.C.: Domain decomposition methods with graph cuts algorithms for total variation minimization Adv. Comput. Math. **36**, 175–199 (2012). https://doi.org/10.1007/s10444-011-9213-4
8. Falgout, R.D., Friedhoff, S., Kolev, T.V., MacLachlan, S.P., Schroder, J.B.: Parallel time integration with multigrid. SIAM J. Sci. Comput. **36**(6), C635–C661 (2014)
9. Firsov, D., Lui, S.: Domain decomposition methods in image denoising using gaussian curvature. J. Comput. Appl. Math. **193**(2), 460–473 (2006)
10. Gander, M.J., Kulchytska-Ruchka, I., Niyonzima, I., Schöps, S.: A new parareal algorithm for problems with discontinuous sources. SIAM J. Sci. Comput. **41**(2), B375–B395 (2019)
11. Gander, M.J., Kwok, F., Zhang, H.: Multigrid interpretations of the Parareal algorithm leading to an overlapping variant and MGRIT. Compu. Vis. Sci. **19**, 59–74 (2018)
12. Gander, M.J., Vandewalle, S.: Analysis of the parareal time-parallel time-integration method. SIAM J. Sci. Comput. **29**(2), 556–578 (2007)
13. Kobayashi, R., Giga, Y.: Equations with singular diffusivity. J. Stat. Phys. **95**, 1187–1220 (1999). https://doi.org/10.1023/A:1004570921372
14. Kohn, R., Temam, R.: Dual spaces of stresses and strains, with applications to hencky plasticity. Appl. Math. Optim. **10**, 1–35 (1983). https://doi.org/10.1007/BF01448377
15. Lions, J.L., Maday, Y., Turinici, G.: Résolution d'edp par un schéma en temps «pararéel?». Comptes Rendus de l'Académie des Sciences - Series I - Mathematics **332**(7), 661–668 (2001)
16. Nievergelt, J.: Parallel methods for integrating ordinary differential equations. Commun. ACM **7**(12), 731–733 (1964)
17. O'Reilly, T., Teeuwisse, W., Webb, A.: Three-dimensional MRI in a homogenous 27.cm diameter bore Halbach array magnet. J. Magn. Reson. **307**, 106578 (2019)
18. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
19. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60**(1), 259–268 (1992)
20. Saad, Y.: Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics Philadelphia, PA, USA (2003)
21. Shan, X., van Gijzen, M.: Deflated preconditioned conjugate gradients for nonlinear diffusion image enhancement. In: Vermolen, F.J., Vuik, C. (eds.) Numerical Mathematics and Advanced Applications ENUMATH 2019. LNCSE, vol. 139, pp. 459–468. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-55874-1_45
22. Shan, X., van Gijzen, M.B.: Deflated preconditioned conjugate gradient methods for noise filtering of low-field MR images. J. Comput. Appl. Math. **400**, 113730 (2022)
23. Vogel, C., Oman, M.: Fast, robust total variation-based reconstruction of noisy, blurred images. IEEE Trans. Image Process. **7**(6), 813–824 (1998)
24. Wu, S.L.: Toward parallel coarse grid correction for the parareal algorithm. SIAM J. Sci. Comput. **40**(3), A1446–A1472 (2018)
25. Xu, J., Tai, X.C., Wang, L.L.: A two-level domain decomposition method for image restoration. Inverse Probl. Imaging **4**(3), 523–545 (2010)