Domain decomposition of the modified Born series approach for scaling up wave simulations

S. Mache[1,2], E. Marakis[3], M. Ktistakis[1], V. Sarantos[1] and I. M. Vellekoop[2]

[1] Rayfos Ltd, Winton House, Winton Square, Basingstoke, RG21 8EN, United Kingdom
[2] Biomedical Photonic Imaging Group, Faculty of Science and Technology, University of Twente, P.O. Box 217, Enschede, 7500 AE, The Netherlands
[3] Foundation for Research and Technology - Hellas, N. Plastira 100, Vassilika Vouton, Heraklion, Crete, GR - 700 13, Greece

Wave propagation simulations are vital in many fields ranging from optics to geophysics. Most popular solvers rely on finite difference or finite element discretisation of equations that govern wave propagation, but suffer from approximation errors and numerical dispersion, especially for large simulations. The modified Born series (MBS) method has overcome these limitations and is currently one of the fastest and most accurate solvers, outperforming competing approaches by orders of magnitude in speed. Here, we present a domain decomposition method that allows running the MBS on multiple parallel nodes or GPUs. Our approach extends the use of the MBS to problems that are too large to fit on a single system, while guaranteeing the monotonic convergence of the method and maintaining accuracy and memory efficiency. It is implemented such that the communication between the subdomains is truncated to a few points on the interface, minimising the memory transfer. Using this framework, we solved a Helmholtz problem of size 315×315×315 cubic wavelengths decomposed over 2 GPUs. Domain decomposition of the MBS enables arbitrarily large simulations, expanding the capabilities towards simulating wave scattering in complex structures in the cubic millimetre regime.

Introduction

Simulation of wave propagation in heterogeneous media is a challenging task [1]. Among the numerous available methods for solving wave equations [2], the modified Born series (MBS) approach is one of the fastest and most accurate [3]. MBS is an iterative method that monotonically converges to the solution for the Helmholtz or Maxwell equations [3]. Unlike the original Born series [4], which is known to diverge for heterogeneous media with strong scattering [5], the MBS is guaranteed to converge to the correct solution.

The MBS approach is grid-based, allowing for arbitrarily complex structures to be represented. It does not rely on finite difference approximations like the finite-difference time-domain (FDTD) [6] and pseudospectral time-domain (PSTD) [7] methods, and thus, does not suffer from numerical dispersion, which can lead to errors of 1% per propagated wavelength [7], [8], [8], [9], [10], [11]. Furthermore, MBS requires $3 \times 3 \times 3$ grid points per wavelength (λ) in 3D simulations, compared to the minimum $20 \times 20 \times 20$ grid points required by methods like FDTD to ensure numerical stability, a factor of 300 in terms of memory consumption. The MBS approach is also faster as it can propagate the field by several wavelengths in each iteration, compared to the typical 10-20 iterations per wavelength required by FDTD and PSTD [3].

However, the size of simulations in MBS was limited by the working memory of a single processor. Recently, Mache and Vellekoop [12], [13], introduced a domain decomposition of the MBS for the Helmholtz equation. The framework, while

maintaining the accuracy, memory efficiency and guaranteed convergence of the MBS, enables larger simulations across multiple PCs or graphics processing units (GPUs).

Methods

The modified [3] and original [4] Born series involve the key step of splitting the wave equation operator into wave propagation in a homogeneous medium and the scattering potential. The domain decomposition framework [12], [13] involves decomposing these operators over multiple subdomains. To enable communication between the subdomains, a correction was introduced to transfer the field at the edge of the subdomains. This correction is truncated to only few points at the subdomain boundaries to minimize the communication overhead. For details on the domain decomposition framework, we refer the reader to [12], [13].

Results

We use the open-source Python implementation on GitHub [14] to demonstrate and benchmark the domain decomposition framework through 3D simulations. Light propagation is simulated through a lens with refractive index $n = 1.5$ placed in a medium with $n = 1$. The lens is placed 50 pixels from the boundary in the x direction. A Gaussian beam source is placed at $x = 0$. We used absorbing boundaries of size 5λ [15] and truncated the edge corrections for communication between subdomains to 8 points at the edges of the subdomains.

The accuracy of the simulation is quantified by the relative error between the field obtained from domain decomposition and a single-domain simulation [3], [12], [16]. The iteration is stopped when the residual, the norm of the energy added in every iteration normalized by that of the preconditioned source, is below 10^{-6}. In the domain decomposition scenario, the residual over the full domain is given by the Euclidean norm over all subdomain residuals.

Figure 1 shows a simulation of size 100 cubic wavelengths (λ^3), which converges in 4425 iterations and takes 185 seconds. For reference, a recent 3D simulation of $100 \times 100 \times 46\ \lambda^3$ of a metalens structure, despite being less than half the size of the $100^3\lambda^3$ simulation, took a much longer 288 seconds using a hardware-accelerated FDTD solver [17].

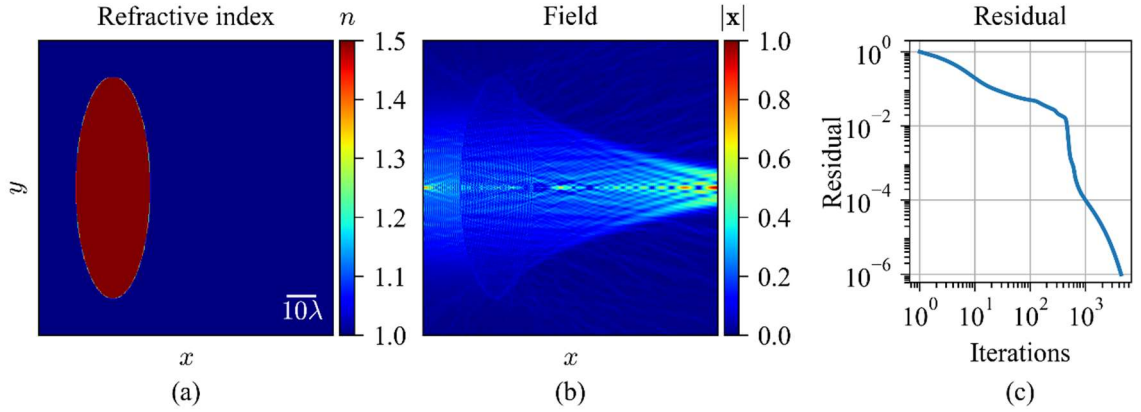

Figure 1: 3D simulation of size 100 cubic wavelengths (λ^3) with a lens in a medium with $n = 1$, a 2D slice of which is shown in (a), and that of the field in (b). (c) The residual as a function of of the iterations.

Figure 2 demonstrates a large 3D simulation enabled by the domain decomposition framework. The simulation is for a problem of size $3.1 \cdot 10^7 \lambda^3$ $(315 \times 315 \times 315)$ on a system with an Intel Xeon Silver 4216 CPU and 2 GPUs (both NVIDIA A40), decomposed into 2 domains along the x-direction. The simulation converged in 40096 iterations and took 7 hours (25188.74 seconds). Simulations of this size cannot fit on a single GPU without domain decomposition and need to be run on the CPU to make and comparisons but take more than $100 \times$ the time compared to domain decomposition-based GPU simulations [12], [13].

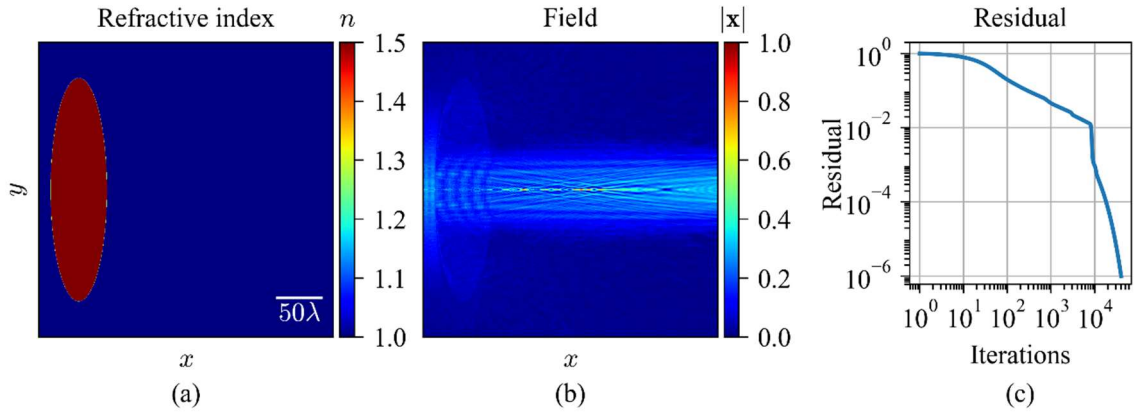

Figure 2: Large 3D simulation of size $315^3 \lambda^3$ with a lens with $n = 1.5$ in a medium with $n = 1$, a 2D slice of which is shown in (a). A 2D slice of the field is shown in (b). (c) The residual as a function of of the iterations.

A comparison of the simulations with and without domain decomposition [12], [13] showed that the number of iterations increased by $1.9 \times$ for the domain decomposition case, which, although large, is independent of the number of subdomains along an axis. The number of iterations increases only when a subdomain is added along a new axis [12], [13]. In addition to the $1.9 \times$ increase in the number of iterations, there is an overhead to computing the edge corrections and transferring them between GPUs to neighbouring subdomains in every iteration. Figure 3 shows that this overhead reduces as the simulation size increases, comparing the time for a 2-domain simulation with that of a simulation without domain decomposition for successively larger sizes.

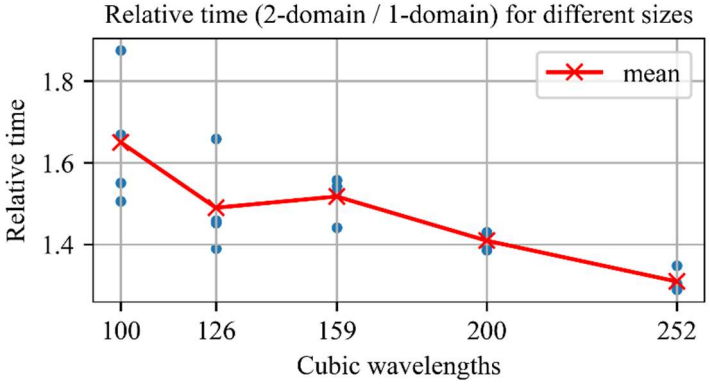

Figure 3: Relative time of 2-domain simulations vs. 1-domain simulations for different sizes, increasing by a factor of 2. The individual blue circles show different simulations, while the red x's show the mean.

Despite the overhead, the large 2-domain, 2-GPU simulations of size $315^3 \lambda^3$ are more than $100 \times$ faster than the 1-domain CPU simulations [12], [13]. Furthermore, domain

decomposition enabled a simulation $1.93\times$ larger with two GPUs than what was previously possible with a single GPU.

Conclusion

The domain decomposition of the modified Born series approach has removed the limitation of running simulations only as large as the working memory of a single processor, while retaining the advantages of speed, accuracy, monotonic convergence, and memory efficiency of MBS. We demonstrated and benchmarked its capabilities with a 3D simulation using two GPUs of an unprecedented size of $3.1 \cdot 10^7 \lambda^3$, $1.93\times$ the size that can be simulated on a single GPU without domain decomposition. The open-source Python implementation is available on GitHub [14].

References

[1] O. G. Ernst and M. J. Gander, "Why it is Difficult to Solve Helmholtz Problems with Classical Iterative Methods," in *Numerical Analysis of Multiscale Problems*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 325–363.

[2] J.-M. Jin, *Theory and computation of electromagnetic fields*. John Wiley & Sons, 2015.

[3] G. Osnabrugge, S. Leedumrongwatthanakun, and I. M. Vellekoop, "A convergent Born series for solving the inhomogeneous Helmholtz equation in arbitrarily large media," *J Comput Phys*, vol. 322, pp. 113–124, 2016, doi: 10.1016/j.jcp.2016.06.034.

[4] M. C. W. Van Rossum and Th. M. Nieuwenhuizen, "Multiple scattering of classical waves: Microscopy, mesoscopy, and diffusion," *Rev Mod Phys*, vol. 71, no. 1, pp. 313 – 371, 1999.

[5] R. E. Kleinman, G. F. Roach, and P. M. van den Berg, "Convergent Born series for large refractive indices," *J. Opt. Soc. Am. A*, vol. 7, no. 5, pp. 890–897, May 1990.

[6] K. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans Antennas Propag*, vol. 14, no. 3, pp. 302–307, 1966.

[7] Q. H. Liu, "The PSTD algorithm: A time-domain method requiring only two cells per wavelength," *Microw Opt Technol Lett*, vol. 15, no. 3, pp. 158–165, 1997.

[8] J.-M. Jin, "Concluding Remarks on Computational Electromagnetics," in *Theory and Computation of Electromagnetic Fields*, John Wiley & Sons, Ltd, 2010, ch. 12, pp. 533–558.

[9] D. Y. Heh and E. L. Tan, "Dispersion analysis of FDTD schemes for doubly lossy media," *Progress In Electromagnetics Research B*, 2009.

[10] F. Zheng and Z. Chen, "Numerical dispersion analysis of the unconditionally stable 3-D ADI-FDTD method," *IEEE Trans Microw Theory Tech*, vol. 49, no. 5, pp. 1006–1009, 2001.

[11] V. Dwarka and C. Vuik, "Pollution and accuracy of solutions of the Helmholtz equation: A novel perspective from the eigenvalues," *J Comput Appl Math*, vol. 395, p. 113549, 2021.

[12] S. Mache and I. M. Vellekoop, "Domain decomposition of the modified Born series approach for large-scale wave propagation simulations," 2024, [Online]. Available: https://arxiv.org/abs/2410.02395v1

[13] S. Mache and I. M. Vellekoop, "Domain decomposition of the modified Born series approach for large-scale wave propagation simulations," *submitted to Journal of Computational Physics*.

[14] S. Mache and I. M. Vellekoop, "Wavesim," 2024, v0.1.0-alpha. Accessed: Oct. 07, 2024. [Online]. Available: https://github.com/IvoVellekoop/wavesim_py

[15] G. Osnabrugge, M. Benedictus, and I. M. Vellekoop, "Ultra-thin boundary layer for high-accuracy simulations of light propagation," *Opt Express*, vol. 29, no. 2, pp. 1649–1658, 2021.

[16] T. Vettenburg and I. M. Vellekoop, "A universal matrix-free split preconditioner for the fixed-point iterative solution of non-symmetric linear systems," *arXiv preprint arXiv:2207.14222*, 2023.

[17] T. W. Hughes, M. Minkov, V. Liu, Z. Yu, and S. Fan, "A perspective on the pathway toward full wave simulation of large area metalenses," *Appl Phys Lett*, vol. 119, no. 15, p. 150502, Oct. 2021, doi: 10.1063/5.0071245/40471.