Proceedings of the Sixteenth International Conference on
Computational Structures Technology
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 14, Paper 2.3
Civil-Comp Press, Edinburgh, United Kingdom, 2026
ISSN: 2753-3239, doi: 10.4203/ccc.14.2.3

Investigating Heavy-Ball Method as a Synchronization-Free Alternative to Conjugate Gradients

D. Hrbáč[1], D. Horák[2,3] and J. Kružík[2,3]

[1] Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VSB - Technical University of Ostrava, Czechia
[2] Department of Applied Mathematics, Faculty of Electrical Engineering and Computer Science, VSB - Technical University of Ostrava, Czechia
[3] Institute of Geonics, Czech Academy of Sciences, Ostrava, Czechia

Abstract

This paper investigates the heavy ball method as a synchronisation-free iterative solver for symmetric positive definite linear systems arising in finite element computations. Unlike conjugate gradient methods, heavy ball avoids global dot products and therefore removes global reductions from the iteration. We compare heavy ball with classical conjugate gradient and pipelined conjugate gradient on sparse matrices from structural mechanics, and we examine the effect of spectral parameter estimation using software-based eigenvalue estimates and Gershgorin bounds. The results show that heavy ball consistently eliminates global synchronisation but typically requires more iterations than conjugate-gradient-based methods. Its practical efficiency depends strongly on the quality and cost of spectral estimates. These findings suggest that heavy ball is not a universal replacement for conjugate gradient, but a promising

alternative in communication-limited regimes where synchronisation dominates time to solution.

Keywords: heavy ball method, communication reduction, conjugate gradient, parallel computing, high-performance computing, iterative solvers, synchronisation-free

1 Introduction

The solution of large systems of linear equations is a fundamental computational task in scientific and engineering applications. In particular, finite element discretisations of partial differential equations lead to sparse linear systems that must be solved efficiently, especially in parallel computing environments where the cost of communication between processors often dominates computation time.

Classical iterative methods, such as the conjugate gradient method, require frequent global synchronisation points where processors must exchange information through dot products. This communication becomes a bottleneck as the number of processors increases, motivating research into communication-avoiding and communication-reducing algorithms [1,2]. Two prominent approaches include pipelined CG, which hides communication behind computation, and communication-avoiding methods that eliminate global reductions entirely.

The heavy ball method, originally proposed by Polyak for accelerating iterative methods [3], provides an alternative to conjugate gradient that avoids global dot products. It belongs to the family of polynomial acceleration schemes and can be viewed as a synchronisation-free method.However, the method requires spectral bounds for parameter selection, and its practical usefulness depends on the trade-off between removed synchronisation and increased iteration counts.

In this paper we investigate whether the heavy ball method can serve as a practical synchronisation-free alternative to conjugate gradient (CG) and pipelined conjugate gradient (PipeCG) for symmetric positive definite systems arising in finite element computations. We focus on the trade-off between removal of global reductions, increased iteration counts, and the cost and quality of spectral parameter estimation.

Contributions

This paper makes the following contributions:

1. We position the heavy ball method as a synchronisation-free polynomial solver for SPD systems in HPC environments.

2. We compare heavy ball with classical CG and pipelined CG on representative sparse matrices from structural mechanics.

3. We evaluate the practical impact of spectral parameter estimation by comparing SLEPc-based eigenvalue estimates with Gershgorin bounds.

4. We discuss when the elimination of global reductions can compensate for the increased iteration count of heavy ball.

2 Heavy Ball Method

We consider the linear system

$$Ax = b, \tag{1}$$

where A is a symmetric positive definite matrix, b is the right-hand side vector, and x_k denotes the kth approximation to the solution. The heavy ball iteration updates the solution via

$$x_{k+1} = x_k + \alpha(b - Ax_k) + \beta(x_k - x_{k-1}), \tag{2}$$

where α is the step size and β is the momentum parameter. Assuming that A is symmetric positive definite with eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$, the optimal parameters are

$$\alpha = \frac{4}{(\sqrt{\lambda_{\max}} + \sqrt{\lambda_{\min}})^2}, \qquad \beta = \left(\frac{\sqrt{\lambda_{\max}} - \sqrt{\lambda_{\min}}}{\sqrt{\lambda_{\max}} + \sqrt{\lambda_{\min}}} \right)^2. \tag{3}$$

Figure 1 shows pseudocode for the three solvers.

The iteration uses only matrix-vector products and vector updates, which are naturally parallel and avoid global reductions. Unlike CG, which requires two global reductions per iteration for dot products, heavy ball requires none.

The method requires estimates of the smallest and largest eigenvalues of A. In practice, these bounds can be obtained by general eigenvalue solvers or by inexpensive estimates such as Gershgorin bounds. In the experiments reported here, we use SLEPc-based estimates and Gershgorin estimates. While the accuracy of the bounds influences convergence, approximate estimates are often sufficient to obtain a stable and competitive method.

From a communication perspective, the key benefit is the elimination of global reductions. This makes heavy ball a strong candidate for extreme-scale systems where latency is the primary bottleneck. However, the price paid is typically higher iteration counts compared to CG.

3 Numerical Experiments

The numerical experiments evaluate the communication behaviour of iterative methods on matrices from structural mechanics. All experiments are conducted on dis-

CG:

Require: A (SPD), b, x_0, tol ϵ
$r \leftarrow b - Ax_0$, $p \leftarrow r$
while $\|r\| \geq \epsilon\|b\|$
  $s \leftarrow Ap$
  $\alpha \leftarrow r^T r / p^T s$
  $x \leftarrow x + \alpha p$
  $r_{new} \leftarrow r - \alpha s$
  $\beta \leftarrow r_{new}^T r_{new} / r^T r$
  $p \leftarrow r_{new} + \beta p$, $r \leftarrow r_{new}$
end while
Output x

PipeCG:

Require: A (SPD), b, tol ϵ
$x_0 \leftarrow 0$, $r \leftarrow b$, $u \leftarrow Ar$
$p \leftarrow r$, $s \leftarrow u$, $q \leftarrow As$
$\gamma \leftarrow r^T r$, $\delta \leftarrow r^T u$, $\alpha \leftarrow \gamma / \delta$
while $\|r\| \geq \epsilon\|b\|$
  $x \leftarrow x + \alpha p$, $r \leftarrow r - \alpha s$, $u \leftarrow u - \alpha q$
  $\gamma_{new} \leftarrow r^T r$, $\delta_{new} \leftarrow r^T u$
  if $\|r\| < \epsilon\|b\|$ **break**
  $\beta \leftarrow \gamma_{new} / \gamma$
  $\alpha_{new} \leftarrow \gamma_{new} / (\delta_{new} - \beta\gamma/\alpha)$
  $p \leftarrow r + \beta p$, $s \leftarrow u + \beta s$
  $w \leftarrow As$, $q \leftarrow w + \beta q$
end while
Output x

HB:

Require: A (SPD), b, x_0, α, β, tol ϵ
$x_1 \leftarrow x_0 + \alpha(b - Ax_0)$
while $\|b - Ax_k\| \geq \epsilon\|b\|$
  $x_{k+1} \leftarrow x_k + \alpha(b - Ax_k) + \beta(x_k - x_{k-1})$
end while
Output x_k

Figure 1: Algorithms: CG, PipeCG and heavy ball.

tributed memory systems using MPI (Message Passing Interface). The test set includes stiffness matrices with varying sparsity patterns and condition numbers, representative of challenges in mechanics where mesh irregularity and material heterogeneity lead to nonuniform sparsity and load imbalance [4].

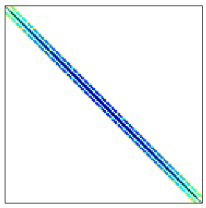
Experiments were performed on LUMI-C with 16 MPI ranks using PETSc (Portable, Extensible Toolkit for Scientific Computation) and SLEPc (Scalable Library for Eigenvalue Problem Computations) [5,8]. Three tolerance levels were tested: 10^{-1}, 10^{-2}, and 10^{-4}. Each reported value was computed from the available runs in three benchmark batches (a, b, c). Most combinations contain 30 runs per batch, but some contain fewer due to incomplete benchmark outputs.

The sparsity patterns of the test matrices are shown in Figure 2. These matrices represent typical challenges in finite element computations: irregular mesh connectivity, varying density, and heterogeneous material distributions.

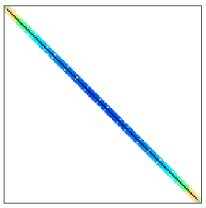
We compare three solvers: classical CG as the baseline, pipelined CG (PipeCG) as a communication-hiding variant, and heavy ball as a communication-avoiding variant. The key difference is the number of global reductions per iteration: CG requires two,

Table 1: Test matrices used in numerical experiments.

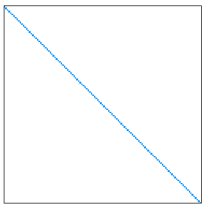
Matrix	Source / Domain	DOFs	nnz	Type
Boeing_crystm02	Boeing / crystal structure	4 870	218 168	SPD
Boeing_crystm03	Boeing / crystal structure	4 755	208 355	SPD
HB_bcsstm21	Boeing / structural mechanics	3 462	26 804	SPD
Pothen_bodyy4	Pothen / structural mechanics	3 345	99 472	SPD
MathWorks_Muu	MathWorks / 3D FEM	3 092	34 089	SPD



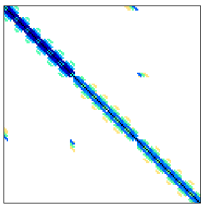
(a) Boeing_crystm02



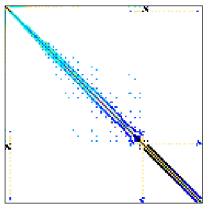
(b) Boeing_crystm03



(c) HB_bcsstm21



(d) MathWorks_Muu



(e) Pothen_bodyy4

Figure 2: Sparsity patterns of the test matrices.

Table 2: CG, PipeCG and heavy ball results at tolerance 10^{-1} on 1 node with 16 ranks.

	CG		PipeCG		HB	
Matrix	Iter.	Time [ms]	Iter.	Time [ms]	Iter.	Time [ms]
Boeing_crystm02	5	2.257	5	1.004	46	3.747
Boeing_crystm03	5	2.662	5	1.310	46	4.357
HB_bcsstm21	2	1.236	2	0.517	13	1.215
MathWorks_Muu	2	1.800	2	0.643	10	1.338
Pothen_bodyy4	16	3.126	16	2.066	46	3.062

Note: Mean over available runs.

Table 3: CG, PipeCG and heavy ball results at tolerance 10^{-2} on 1 node with 16 ranks.

	CG		PipeCG		HB	
Matrix	Iter.	Time [ms]	Iter.	Time [ms]	Iter.	Time [ms]
Boeing_crystm02	23	3.922	23	2.696	67	4.575
Boeing_crystm03	22	5.175	22	3.837	68	5.944
HB_bcsstm21	2	1.228	2	0.550	19	1.112
MathWorks_Muu	11	2.469	11	1.276	19	1.407
Pothen_bodyy4	47	5.475	47	4.233	87	5.165

Note: Mean over available runs.

PipeCG overlaps reductions with computation, and heavy ball requires none.

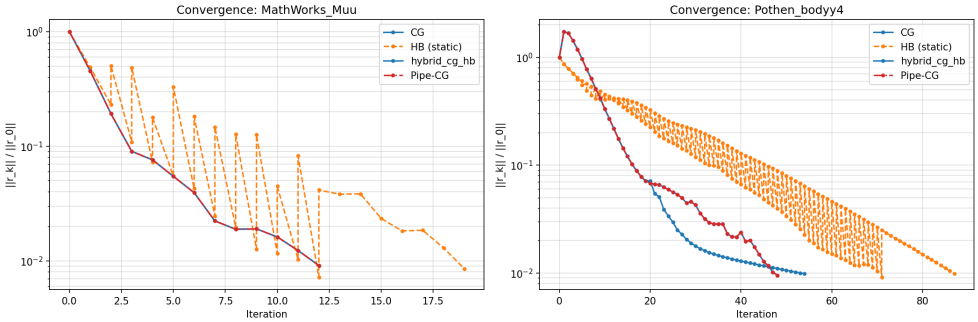
The results show that heavy ball consistently eliminates global reductions but requires more iterations than classical CG for the tested matrices at all three tolerance levels. PipeCG [6,7] maintains the same iteration counts as CG while reducing time-to-solution by hiding communication behind computation. At the looser tolerance (10^{-1}), the iteration count ratio between HB and CG is lower, but HB still does not outperform CG in time-to-solution for most matrices. At the stricter tolerance (10^{-4}), the higher HB iteration count becomes even more visible, although the method remains competitive on selected matrices.

The convergence plots in Figure 3 illustrate the different behaviour of the three methods. While CG and PipeCG show the typical convergence curves for conjugate

Table 4: CG, PipeCG and heavy ball results at tolerance 10^{-4} on 1 node with 16 ranks.

	CG		PipeCG		HB	
Matrix	Iter.	Time [ms]	Iter.	Time [ms]	Iter.	Time [ms]
Boeing_crystm02	53	6.743	53	5.073	107	6.850
Boeing_crystm03	51	8.665	51	7.096	109	8.800
HB_bcsstm21	2	1.251	2	0.551	31	1.507
MathWorks_Muu	28	3.663	28	2.387	38	2.298
Pothen_bodyy4	103	8.967	103	7.439	161	9.357

Note: Mean over available runs.



(a) MathWorks_Muu (b) Pothen_bodyy4

Figure 3: Convergence comparison of CG, PipeCG and HB at tolerance 10^{-2}.

Table 5: SLEPc vs Gershgorin: eigenvalue estimation and solve times at 10^{-2}.

Matrix	SLEPc			Gershgorin		
	Eigs time [s]	Solve time [s]	Iter.	Eigs time [s]	Solve time [s]	Iter.
HB_bcsstm21	0.159	0.0011	19	0.00018	0.0014	19
Pothen_bodyy4	0.329	0.0052	87	0.00039	0.0067	71

gradient-type methods, heavy ball exhibits a different convergence pattern due to its polynomial acceleration nature.

The method requires spectral bounds for parameter selection. We compare two approaches: SLEPc-based eigenvalue estimation and Gershgorin-based estimates.

The comparison indicates that the effect of spectral estimation on iteration counts is matrix-dependent. For HB_bcsstm21 both approaches lead to the same iteration count, while for Pothen_bodyy4 the Gershgorin-based estimate yields fewer iterations. However, SLEPc-based estimation introduces a substantially larger preprocessing overhead.

4 Discussion

Our experiments reveal several important observations about the practical use of heavy ball.

First, heavy ball consistently removes global reductions from the iteration, which is a clear advantage in communication-limited regimes. However, this advantage comes at the cost of increased iteration counts compared to CG-based methods. The time-to-solution of heavy ball is not uniformly superior to CG or PipeCG in the tested regime.

Second, the practical usefulness of heavy ball depends strongly on the availability of inexpensive and reliable spectral bounds. SLEPc-based estimates provide better

convergence but introduce significant overhead. Gershgorin estimates are essentially free but may yield suboptimal parameters and slower convergence.

Third, the performance is matrix-dependent. Some matrices show competitive behaviour of heavy ball, while others show that the increased iteration count cannot be compensated by the removal of synchronisation.

Fourth, our experiments are limited to a single-node regime with 16 MPI ranks. Strong scaling studies and larger problem sizes are needed to fully assess the potential of heavy ball at extreme scale.

Limitations

- Experiments in this paper are limited to a single-node regime (16 MPI ranks).
- Only symmetric positive definite matrices are considered.
- Heavy ball requires spectral bounds, which adds preprocessing cost.
- No full strong-scaling study is included here.

5 Conclusions

The heavy ball method offers a compelling synchronisation-free alternative to CG-based methods for SPD systems, because it eliminates global reductions entirely. However, our experiments also show that this advantage comes at the cost of increased iteration counts and sensitivity to spectral parameter selection. In the tested regime, heavy ball is competitive on selected matrices, but not uniformly superior in time to solution.

The practical usefulness of the method therefore depends not only on communication costs, but also on the availability of inexpensive and reliable spectral bounds. These results position heavy ball as a complementary method rather than a replacement for CG or PipeCG.

Future work will focus on stronger scaling experiments and on hybrid strategies that combine the fast initial convergence of PipeCG with the synchronisation-free iteration of heavy ball.

Acknowledgements

The authors acknowledge the financial support of the European Union under the REFRESH - Research Excellence For Region Sustainability and High-tech Industries project number CZ.10.03.01/00/22 003/0000048 via the Operational Programme Just Transition. This work was also supported by the the Czech Science Foundation grant 26-20457S.

References

[1] P. Ghysels, W. Vanroose, ”Hiding global synchronization latency in the preconditioned Conjugate Gradient algorithm”, Parallel Computing, 40, 224-238, 2014

[2] M.F. Hoemmen, ”Communication-avoiding Krylov subspace methods”, Technical Report UCB/EECS-2010-37, EECS Department, University of California, Berkeley, 2010

[3] B.E. Polyak, ”Some methods of speeding up the convergence of iteration methods”, USSR Computational Mathematics and Mathematical Physics, 4(5), 1-17, 1964

[4] K. Davis, Y. Hu, ”The University of Florida Sparse Matrix Collection”, ACM Transactions on Mathematical Software, 38(1), 1-25, 2011

[5] ”PETSc: Portable, Extensible Toolkit for Scientific Computation”, https://petsc.org/, 2026

[6] S. Cools, J. Cornelis, W. Vanroose, ”Numerically Stable Recurrence Relations for the Communication Hiding Pipelined Conjugate Gradient Method”, IEEE Transactions on Parallel and Distributed Systems, 30(11), 2507-2522, 2019

[7] S. Cools, J. Cornelis, P. Ghysels, W. Vanroose, ”Improving strong scaling of the Conjugate Gradient method for solving large linear systems using global reduction pipelining”, arXiv preprint arXiv:1905.06850, 2019

[8] ”SLEPc: Scalable Library for Eigenvalue Problem Computations”, https://slepc.upv.es/, 2026