

Parameter optimization for restarted mixed precision iterative sparse solver

Computational techniques are continually advancing, facing new challenges that demand innovative solutions. One particularly intriguing challenge lies in utilizing mixed-precision computations for iterative methods within Krylov subspace. This approach promises to optimize computational efficiency without compromising accuracy.

A problem within this domain is to identify the optimal parameter that maximizes computational performance while ensuring robustness. This involves fine-tuning the parameter based on the inherent properties of the matrix when solving linear algebraic equations. Specifically, the focus is on employing the preconditioned conjugate gradient method with Jacobi preconditioning in a mixed-precision environment.

The computational workflow is the following: initially, the problem is tackled in single precision using the PETSc library (<https://petsc.org/release/>). Here, the matrix is read and solved, starting from a zero initial solution and refined up to a specified accuracy ε_1 . Subsequently, the solution is transitioned to double precision. Leveraging this refined solution as an initial solution, along with the matrix represented in double precision, the computation is further refined to achieve a higher accuracy, denoted as ε_2 .

The ultimate objective is to optimize the parameter ε_1 in such a manner that the overall computational time for solving the linear system of equations is minimized across a diverse range of Symmetric Positive Definite (SPD) matrices. These SPD matrices serve as representative benchmarks and can be sourced from the Sparse Suite collection (<https://sparse.tamu.edu/>). In case when solution is not provided, x it can be generated randomly with values between 1 and 2 and compute right hand side as $b = Ax$.

In essence, this problem encapsulates the balance between computational efficiency and numerical accuracy, aiming to push the boundaries of what is achievable in modern scientific computing.

The schematically proposed algorithm for solving the system of linear equations $Ax = b$ with parameters ε_1 and ε_2 can be outlined as follows:

1. Initialize the initial solution to zero, i.e. $\bar{x}_0=0$.
2. Convert matrix A and vector b to single precision, i.e., $\bar{A} = \text{float}(A)$, $\bar{b} = \text{float}(b)$. Matrix A is representative in float, so there isn't difference between coefficients \bar{A} and A .
3. Solve $\bar{A}\bar{x} = \bar{b}$ using the preconditioned conjugate gradient method with Jacobi preconditioning up to accuracy ε_1 , measuring the time as t_1 .
4. Convert the obtained solution \bar{x} to an initial solution in double precision $x_0 = \text{double}(\bar{x})$.
5. Solve $Ax = b$ using the preconditioned conjugate gradient method with Jacobi preconditioning up to accuracy ε_2 , measuring the time as t_2 .

Typically, the parameter defining the final accuracy is predetermined, let's say it is set to $\varepsilon_2 = 10^{-8}$. Our goal is to select the parameter ε_1 in such a way as to minimize the overall computation time. Alternatively, we could minimize the total number of iterations across both stages, considering that iterations at the first stage are significantly faster, approximately in 3-4 times. Also, if some extra

computations should be included to select ε_1 , computation cost should be added to estimation of execution time or some equivalent cost in iteration should be added in case if we compute in iterations.

The main question arising during the implementation of this scheme is what factors influence the choice of the optimal parameter. Our computational experience indicates that the optimal parameters can vary significantly for different types of matrices. Thus, the structure of the matrix may provide clues for this selection. However, we have not yet been able to explicitly identify this pattern.

As proposals for this challenge task, we would like to receive ideas that could help identify this pattern specifically and solve the problem either for a particular class of matrices or as a whole. We are interested in exploring this problem from various perspectives, including:

1. Which quantitative and qualitative properties of the matrix influence the choice of the optimal parameter? How computationally intensive is the calculation of these parameters, and do they have practical significance?
2. Are there some hidden or lesser-known properties of the matrix that could be leveraged to solve this problem?
3. What modern artificial intelligence methods, in general, and machine learning techniques, in particular, could assist in finding the optimal parameter?
4. Can formulation of problem be improved? Right now, we have a bit technical formulation but interesting in mathematical one.

We'd really value any insights, recommendations, or novel strategies you can bring to the table for this challenge. Your expertise and contributions can be decisive in strengthening our collaborative work and setting the stage for further exploration and beneficial partnerships in the future.