Parallel Iterative method for Ax=b

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathrm{P}^{-1}\mathbf{r}^{(k)}$$

To solve $\mathrm{A}\mathbf{x} = \mathbf{b}$ in parallel using an iterative scheme it is normally necessary to have designed the matrix-times-vector operation in parallel, and the preconditioner solve as well.

Objectives (you may skip some)

1- using *gc.hpp* and the matrix class in Example/src/Matrix (a full matrix with openMP to parallelize the product with a vector), implement a conjugate gradient algorithm with no preconditioner (the preconditioner is a class with a method solve that represent the identity). Create a sdp matrix and test the code.

2 – Replace the Matrix calss with the Pmatrix class in Examples/Paralla/MPI/Pmatrix (look at the local README.md file) , that partition the matrix in an MPI environment and see if it works correctly (see note below).

3- Replicate the Pmatrix code, but new using as base matrix no more the full one I have created in Matrix.hpp, but a sparse matrix of the Eigen library (or write your own!). Get a sdp sparse matrix from Matrix Market https://math.nist.gov/MatrixMarket/ and try the system.

More advanced stuff (may lead to a exam project)

4- Extend to other iterative schemes, like GMRES-BGCStab
5- Implement a parallel preconditioner based on approximate inverse [5]

Note:
The test the code with the full matrix you can generate a random matrix (you already have a function member of the Matrix class for that). Then you generate the right hand side by doing $\mathbf{b} = \mathrm{A}\mathbf{1}$, where $\mathbf{1}$ is a vector of all ones (of course you may also use any vector different from 0). Then you know the solution of the system $\mathrm{A}\mathbf{x} = \mathbf{b}$, and you can use it to verify the correctness of your code.

[1] templatesForIterativeMethods.pdf
[2] LinearSystem_short.pdf
[3] https://eigen.tuxfamily.org/dox/group__SparseQuickRefPage.html

[4] https://eigen.tuxfamily.org/dox/group__TutorialSparse.html#TutorialSparse_SubMatrices
[5] ApproximateInversePreconditioner.pdf