유한요소 Sparse Matrix

Our problem is to solve the Dirichlet problem in the domain [-1 1]\*[-1 1] of an elliptic equation given by −∇ · p∇u = f in Ω where p=1+x+2 and the exact solution is u=x(1-x)y(1-y) by using sparse matrices.

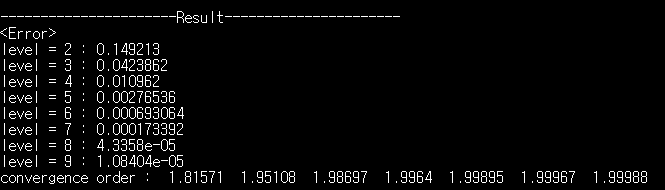
1. About the code

I made the header file t\_sparse which contains two class IVector\_double and IMatrix\_double. Then a sparse vector is expressed by IVector\_double class by saving nonzero data and the position of them. A sparse vector is expressed by IMatrix\_double by saving each vector as a sparse vector. I changed the code about Boundary Process because a lot of zero data could be saved in the sparse matrix mA, so it could lose the advantage of sparse matrices.

I tried both P1-conforming and P1-nonconforming.

1. P1-conforming

First, the errors and the convergence orders when the level is 2, 3… 9 are follow.

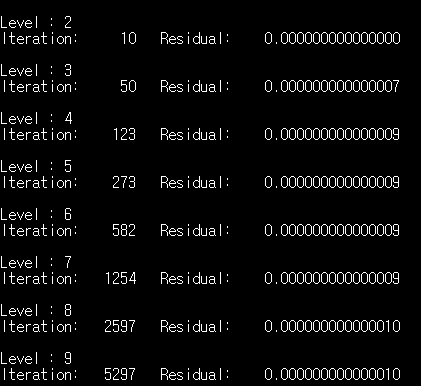


The result is the same to that of using full matrices for level6. However, when full matrices are used, since Visual Studio is for 32bit, the heap memory is at most 2GB. Thus for level7, the full matrix mA cannot be allocated dynamically to the heap memory and so I cannot test P1-conforming code for level7.

When sparse matrices are used, the size of mA is reduced. Moreover, as level increases to level+1, the size of mA just becomes double. Thus the code gets the efficiency of memory allocation.

Moreover, when CG are executed, mA is multiplied with a full vector. Thus if mA is a full matrix, the order is O() and if mA is a mA is a sparse matrix, then the order is O(n). Also, the number of iteration in CG increases about twice when the level increases by 1, which can be observed in the following result.

Thus the order of CG is O() for full matrices and is O() for sparse matrices.



Therefore, when we use sparse matrices, we get much faster speed for calculating errors.

1. P1-nonconforming

P1-nonconforming case has similar result with P1-conforming case.

