Modular Testing in Sparse Matrix Solvers
Author: Beitong Tian   NetID: bt346

The output of the program:

********LAB2********

The result for testing Constructor module: 0
constructor test passed

The test result for testing matrix product: 6.47269e-08
matrix product test passed

Check if the matrix will convergence.
This matrix will convergence!

the tolerance is 1e-7
iteration time: 3
current mem use for b1: 40960
time for b1: 0.003674 s
residual for b1 : 8.92558e-07

iteration time: 3
current mem use for b2: 36864
time for b1: 0.003564 s
residual for b2 : 1.01347e-06

iteration time: 5
current mem use for b3: 40960
time for b1: 0.005797 s
residual for b3 : 1.24081e-07

the tolerance is 1e-10
iteration time: 4
current mem use for b1: 40960
time for b1: 0.004709 s
residual for b1 : 3.52298e-08

iteration time: 4
current mem use for b2: 40960
time for b1: 0.00485 s
residual for b2 : 4.02337e-08

iteration time: 7
current mem use for b3: 45056
time for b1: 0.008483 s
residual for b3 : 2.13235e-10

So, additional implementation achieves better results.
And the time can indicate the number of operations. The memory is almost the same because all the memory we used is on stack, it will be automatically recycled after the for loop and the function call.

Module and module test description:

In this lab, I implement a Jacobi solver in sparse matrix with row-compressed formats.
To achieve the Jacobi solver, I designed some matrix modules in sparse matrix formats.

1.Sparse matrix structure.

Description: In this structure, I define the basic sparse matrix elements such as row number, column number, row index, column index, non-zero elements value, total non-zero elements amount.
Test: The test of this module is combined with the next module.

2. Sparse matrix constructor.

Description: In this constructor. I initialize all the variables in the matrix structure. And there are two methods to construct the sparse matrix.
(1)Straightly read in each elements of the sparse matrix. For this lab, I used this method to read in the mat1 matrix.
(2)First read in the array, and then use the retrieveElement method to construct the Sparse matrix.
Test: For this lab, I used the first method and just read in the values supplied by others. But there are many temp matrix which is constructed by the method 2. So I have to test the method. I used the small matrix A to test this constructor. First I read in the sparse matrix. Then, I use the sparse matrix data structure to reconstruct the full matrix B. I calculate the ||A-B||2 to see if the sparse matrix I construct is right.
Result: The result is 0 which means the constructor is right. We can use this structure to represent the sparse matrix.

3. Sparse matrix Product.

Description: I design a method to achieve two sparse matrix product. The inputs are two sparse matrix and the output is the product result which is also a sparse matrix.
This function not only can do the matrix product vector, but also can do the matrix product matrix.

Test: I had two methods to test this function.
(1)I multiply the mat1 matrix with a vector which has 1.0 in all elements and then sum the result vector whose name is sum1. I also sum the matrix non-zero elements. And I named the result as sum2. I calculate the ||sum1-sum2|| to verify if the product result is right.
(2)I can also use Wilkinson testing method to test the product method. I used the small matrix and multiply them using the sparse matrix product. And I also used full matrix method to do the same process. I calculate the second norm of the difference of the two result vector to verify if the product is right.
Result: I choose method1 to do the test. The result is very small. So the product method is right.

4.residual calculation

Description: I use this method to calculate the normalized residual.

5. Jacobi

Description: I used this method to get the solution of the equation set. I didn't use the matrix product method. I calculated the value of each elements with the following equation.

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \ldots, n.$$

Test: I calculate the normalized residual to verify the method.

$$\varepsilon = \frac{\left\| b - Ax \right\|_2}{\left\| b \right\|_2}$$