**COMP3046 Course Project Report**

2018-05-10

16227018 GROVER Karan Yogesh

15251276 ZHANG, Chengxin

1. **Pseudocodes** 
   1. Serial solution

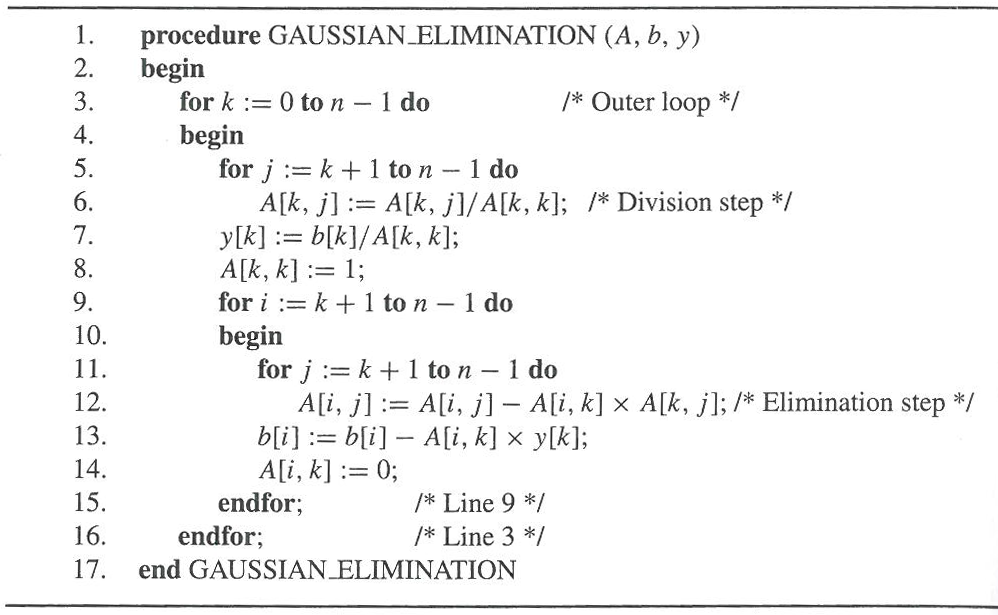
For the Serial solution, we just reference for the lecture slides pseudocodes code.

The Gaussian Elimination algorithm converts ***A****x*=*b* to ***U****x*=*y*

Input: ***A***, *b*

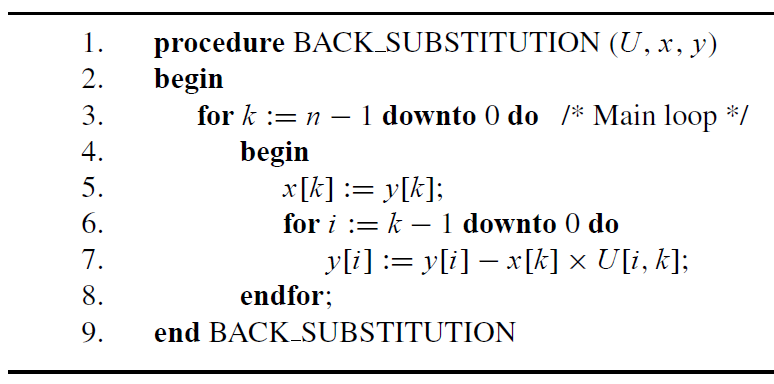
Output: ***U***, *y*

***U*** is stored in the upper-triangular locations of ***A***.



Sequential back-substitution algorithm for solving an upper-triangular system of equations ***U****x*=*y*

* + ***U*** is a unit upper-triangular matrix (i.e., ***U***[*k*,*k*] = 1)
  + It takes *n*2/2 multiplications and subtractions



* 1. Parallel solution

1. Procedure Gaussian\_emilination\_omp
2. int i, j, k
3. double temp;
4. //gauss elimination parts
5. for k := 0 to n-1 do /\* Main Loop \*/
6. begin
7. //part 1
8. do omp parallel operation
9. for j := k + 1 to n - 1 do
10. begin
11. A[i,j] := A[i,j] - A[i,k] \* A[k,j];
12. **ender for;**
13. b[i] = b[i] - A[i,k] \* y[k];
14. A[i,k] := 1;
15. **ender for;**
17. //part 2
18. do omp parallel operation
19. for i := k + 1 to n – 1
20. begin
21. for j := k + 1 to n-1
22. A[i, j] = A[i, j] - A[i, k] \* A[k, j];
23. **ender for;**
24. b[i] = b[i] - A[i, k] \* y[k];
25. A[i, k] = 0;
26. **ender for;**
27. //Back-substitution parts
28. # pragma omp parallel for num\_threads(thread\_count) \
29. default(none) private(i) shared(A, x, y, n, k)
30. for k := n – 1 downto 0;
31. begin
32. x[k] = y[k];
33. for i := k – 1 down to 0
34. Begin
35. y[i] = y[i] - x[k] \* A[i , k];
36. **ender for;**
37. **ender for;**

1. **Experimental environment** 
   1. CPU: Intel(R) Core(TM) i7-4770K CPU @ 3.5GHz 3.5GHz
   2. RAM: 8.00GB WITH 7.90GB usable
   3. GPU: NVIDIA GeForce GT 740
   4. Operating system: 64-bit Windows 7 Enterprise
   5. Software and IDE

Visual Studio 2013;

Developer Command Prompt for VS2013;

1. **Compilation**

To compile the program, put files in a single folder, including “*main.c*”, “*gaussian.h*”, ” *gaussian.c* ”. Then open Developer Command Prompt for VS2013, change the director to your single folder. Then type “**cl –openmp gaussian.c main.c**” in the Prompt, then type “*main.exe*” to run.

1. **Experimental results and speedup analysis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. **Explanations on the experimental results**

The openMP increase the performance.

1. **Contribution**
   1. Coding:

ZHANG, Chengxin: Gaussian elimination and back substitution;

Serial solution for Single thread;

Verification

Karan: Parallel solution with OpenMP

* 1. Report:

ZHANG, Chengxin: Report writing and formatting

*Some remarks for the original working distribution: Just because Karan has an example on the May 10 evening, thus he has not enough time to finish his parts of the reports.*