COMP-40730 HPC

Report for Assignment 1

Author:	Paula Dwan
Due date:	13-July-2014
Lecturer:	Alexey Lastovetsky
Subject:	COMP-40730 High Performance Computing
College:	University College Dublin

CONTENTS

Exercise

Write C programs implementing the following three algorithms of multiplication of two n×n dense matrices:

1) Straightforward non-blocked IJK algorithm.
2) Blocked IJK algorithm using square b×b blocks.
3) Blocked KIJ algorithm using square b×b blocks.

Experiment with the programs and build/plot:

1) The dependence of the execution time of each program on the matrix size n and the block size b .
2) The speedup of the Blocked algorithms over the non-blocked one as a function of the matrix size and the block size.
3) Compare the fastest program with the BLAS/ATLAS routine **dgemm** implementing the same operation.

Explain the results.

Variants of the assignment:

1) Multiplication of matrix blocks in the implementation of the Blocked IJK algorithm:
   a. manually written
   b. BLAS calls
   c. ATLAS calls
2) Multiplication of matrix blocks in the implementation of the Blocked KIJ algorithm:
   a. manually written
   b. BLAS calls
   c. ATLAS calls
3) Comparison with BLAS/ATLAS routine;
   a. BLAS
   b. ATLAS

Observations

I had started working on the three different implementations and decided to complete each in order to learn better the theory behind the assignments. I also used three separate programs to complete each section. At some times of the day, UCD servers are faster than others, this may explain some discrepancies in the results.

Overview of Computations Obtained and How

Assignment 1 basically involved (for me) writing three programs :

1. **A1-Sijk-1D.c**

   Implementation of a straight forward matrix nxn multiplication using both a manual (dot product calculation) and BLAS application.

   Algorithm

   no additional variable

```
for (i=0; i<rows; i++)
{
    for (j=0; j<cols; j++)
    {
        for (k=0; k<rows; k++)
        {
            C [i][j] += (A [i][k]) * (B [k][j]);
        }
    }
}
```

   This was calculated using a temporary variable to improve cache memory access and reduce access and thus improve performance.

   Algorithm

   using additional variable

```
for (i=0; i<rows; i++)
{
    for (j=0; j<cols; j++)
    {
        double sum = 0.0;
        for (k=0; k<rows; k++)
        {
            sum+= (A [i][k]) * (B [k][j]);
        }
        C [i][j] = sum;
    }
}
```

2. **A1-Bijk-1D.c**

   Blocked IJK : This was calculated using manual algorithm blocked IJK and DGEMM. Also matrix |C| was evaluated using a straight-forward IJK/KIJ algorithm in order to best compare the time taken.

   So I had two types of Blocked KIJ : manual and DGEMM as well as the simplified straight-forward calculation. Blocked IJK was implemented using simplified algorithm of three inner loops and adding block size to each.

   Algorithm

   blocked IJK

```
for(bi = 0; bi < nx; bi += nb)
{
  for(bj = 0; bj < nx; bj += nb)
  {
    for(bk = 0; bk < nx; bk += nb)
    {
      for(ni = 0; ni < nb; ni++)
      {
        for(nj = 0; nj < nb; nj++)
        {
          for(nk = 0; nk < nb; nk++)
          {
            C[ (bi+ni)*nx+bj+nj ] += A[ (bi+ni)*nx+bk+nk] *
                B[(bk+nk)*nx+bj+nj ];
          }
        }
      }
    }
  }
}
```

Straightforward IJK which was included for comparison

Algorithm
Straight-forward IJK

```
for (ni=0; ni<nx; ni++)
{
  for (nj=0; nj<nx; nj++)
  {
    for (nk=0; nk<(n/b); nk++)
     {
      C [ni*nx+nj]+= A [ni*nx+nk] * B [nk*nx+nj];
    }
  }
}
```

3. **A1-Bkij-1D.c**

   Blocked KIJ : This was calculated using manual algorithm blocked IJK and DGEMM. Also matrix |C| was evaluated using a straight-forward IJK/KIJ algorithm in order to best compare the time taken.

   So I had two types of Blocked KIJ : manual and DGEMM as well as the simplified straight-forward calculation. Blocked IJK was implemented using simplified algorithm of three inner loops and adding block size to each..

Algorithm
blocked KIJ

```
for(bk = 0; bk < nx; bk += nb)
{
  for(bi = 0; bi < nx; bi += nb)
  {
    for(bj = 0; bj < nx; bj += nb)
    {
      for(nk = 0; nk < nb; nk++)
      {
        for(ni = 0; ni < nb; ni++)
        {
          for(nj = 0; nj < nb; nj++)
          {
            C[ (bk+nk)*nx+bi+ni] += A[ (bk+nk)*nx+bi+nj] *
                B[ (bi+nj)*nx+bi+ni];
          }
        }
      }
    }
  }
}
```

Simplified Blocked KIJ – included for comparison

Algorithm
Straight-forward KIJ

```
for (nk=0; nk<nx; nj++)
{
  for (ni=0; ni<nx; ni++)
  {
    for (nj=0; nj<nx; nj++)
    {
      C [ni*nx+nj] += B [ni*nx+nk] * B[nk*nx+nj];
    }
  }
}
```

For each .c program, |C| matrix was calculated manually using blocked IJK or blocked as well as the straight-forward equivalent and DGEMM (cblas by defaut otherwise the use could decide to re-build and execute using cblas or atlas.

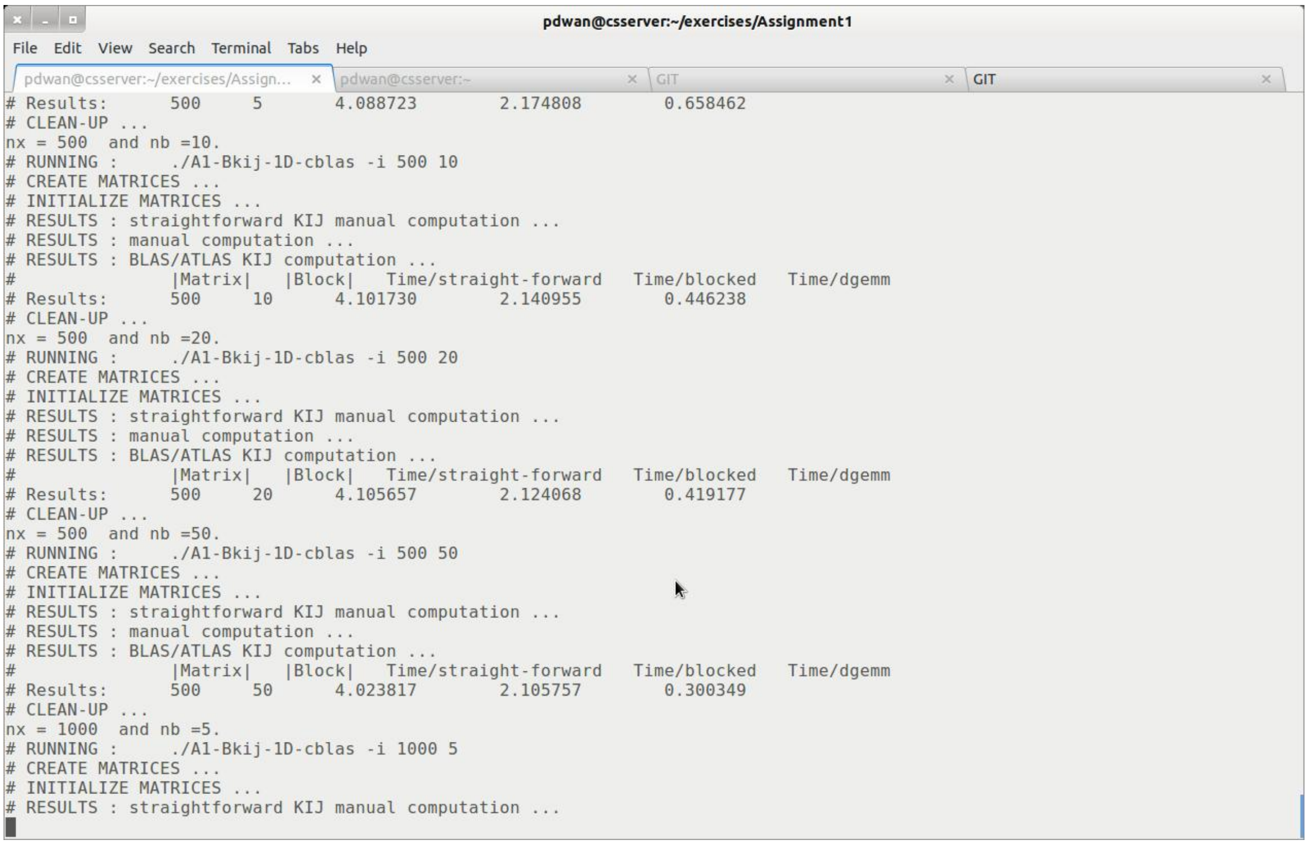
Thus results were obtained for each and for DGEMM using the same source |A| and |B|. The same matrix computation was implemented using cblas / atlas. Time taken to calculate |C| was noted and graphed.

Multiple implementation of each .c program was enabled using ***./runAssignment1.sh,*** for example :

```
$ ./runAssignment1.sh -a -i -v
```

This runs all three algorithm c programs using cblas for incremental column values, using predefined settings for matrix and block sizes for each implementation.

Sample CLI output follows :


pdwan@csserver:~/exercises/Assignment1
File Edit View Search Terminal Tabs Help
pdwan@csserver:~/exercises/Assign...
pdwan@csserver:~
GIT
GIT

```
# Results:      500     5       4.088723        2.174808        0.658462
# CLEAN-UP ...
nx = 500  and nb =10.
# RUNNING :     ./A1-Bkij-1D-cblas -i 500 10
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : straightforward KIJ manual computation ...
# RESULTS : manual computation ...
# RESULTS : BLAS/ATLAS KIJ computation ...
#               |Matrix|   |Block|   Time/straight-forward   Time/blocked   Time/dgemm
# Results:      500     10      4.101730        2.140955        0.446238
# CLEAN-UP ...
nx = 500  and nb =20.
# RUNNING :     ./A1-Bkij-1D-cblas -i 500 20
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : straightforward KIJ manual computation ...
# RESULTS : manual computation ...
# RESULTS : BLAS/ATLAS KIJ computation ...
#               |Matrix|   |Block|   Time/straight-forward   Time/blocked   Time/dgemm
# Results:      500     20      4.105657        2.124068        0.419177
# CLEAN-UP ...
nx = 500  and nb =50.
# RUNNING :     ./A1-Bkij-1D-cblas -i 500 50
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : straightforward KIJ manual computation ...
# RESULTS : manual computation ...
# RESULTS : BLAS/ATLAS KIJ computation ...
#               |Matrix|   |Block|   Time/straight-forward   Time/blocked   Time/dgemm
# Results:      500     50      4.023817        2.105757        0.300349
# CLEAN-UP ...
nx = 1000  and nb =5.
# RUNNING :     ./A1-Bkij-1D-cblas -i 1000 5
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : straightforward KIJ manual computation ...
```

Single implementation was completed using **A1-<algorithm variant>-1D.c.**

This is compiled using *gcc* for atlas and cblas, as follows :

```
gcc  -I/home/cs/khasanov/libs/CBLAS/src A1-Sijk-1D.c -o A1-Sijk-1D-cblas
/home/cs/khasanov/libs/cblas_LINUX.a  /usr/lib/libblas.a -lgfortran

gcc -o A1-Sijk-1D-atlas A1-Sijk-1D.c -I/home/cs/khasanov/libs/ATLAS/include/
-L/home/cs/khasanov/libs/ATLAS/lib/Linux_UNKNOWNSSE2_4/ -lcblas -latlas -lm -O3
```

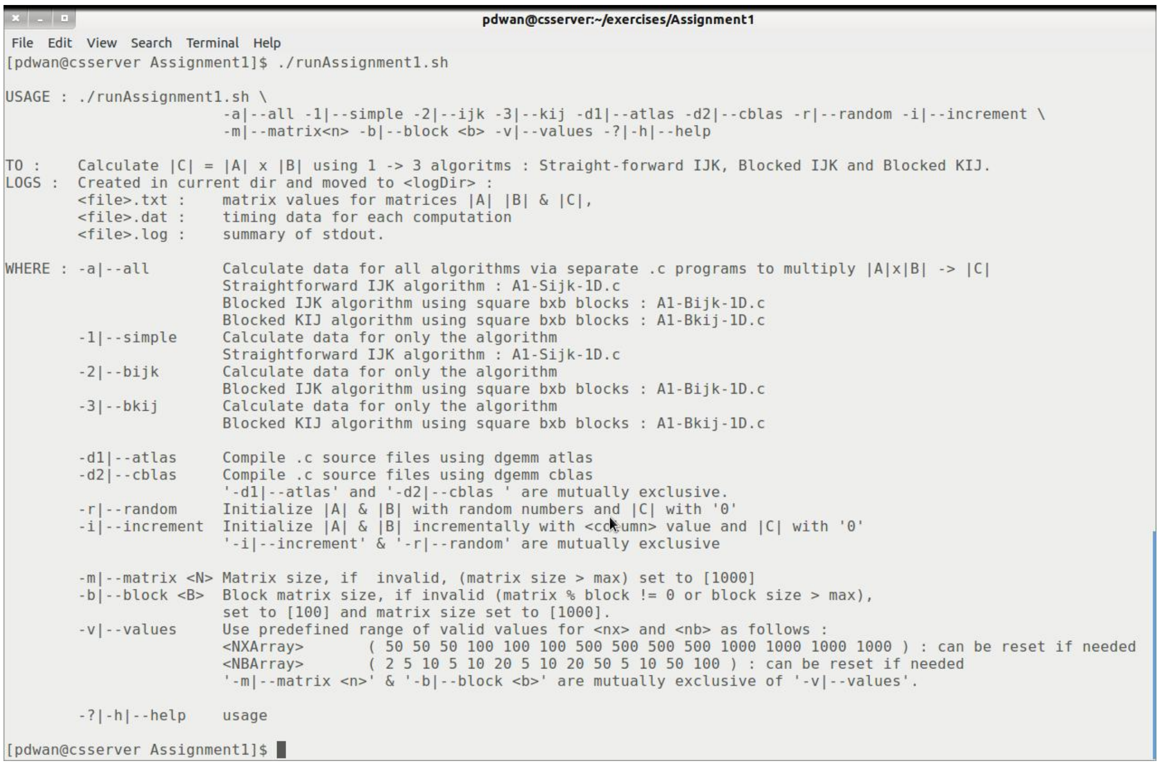
```
gcc  -I/home/cs/khasanov/libs/CBLAS/src A1-Bijk-1D.c -o A1-Bijk-1D-cblas
/home/cs/khasanov/libs/cblas_LINUX.a  /usr/lib/libblas.a -lgfortran

gcc -o A1-Bijk-1D-atlas A1-Bijk-1D.c -I/home/cs/khasanov/libs/ATLAS/include/
-L/home/cs/khasanov/libs/ATLAS/lib/Linux_UNKNOWNSSE2_4/ -lcblas -latlas -lm -O3
```

```
gcc  -I/home/cs/khasanov/libs/CBLAS/src A1-Bkij-1D.c -o A1-Bkij-1D-cblas
/home/cs/khasanov/libs/cblas_LINUX.a  /usr/lib/libblas.a -lgfortran

gcc -o A1-Bkij-1D-atlas A1-Bkij-1D.c -I/home/cs/khasanov/libs/ATLAS/include/
-L/home/cs/khasanov/libs/ATLAS/lib/Linux_UNKNOWNSSE2_4/ -lcblas -latlas -lm -O3
```

Assignment Execution

Each program was executed multiple times standalone or using the script *./runAssignment1.sh* to obtain as wide a range of time taken to calculate |C| using each algorithm.

This has multiple options and the syntax and usage follows :

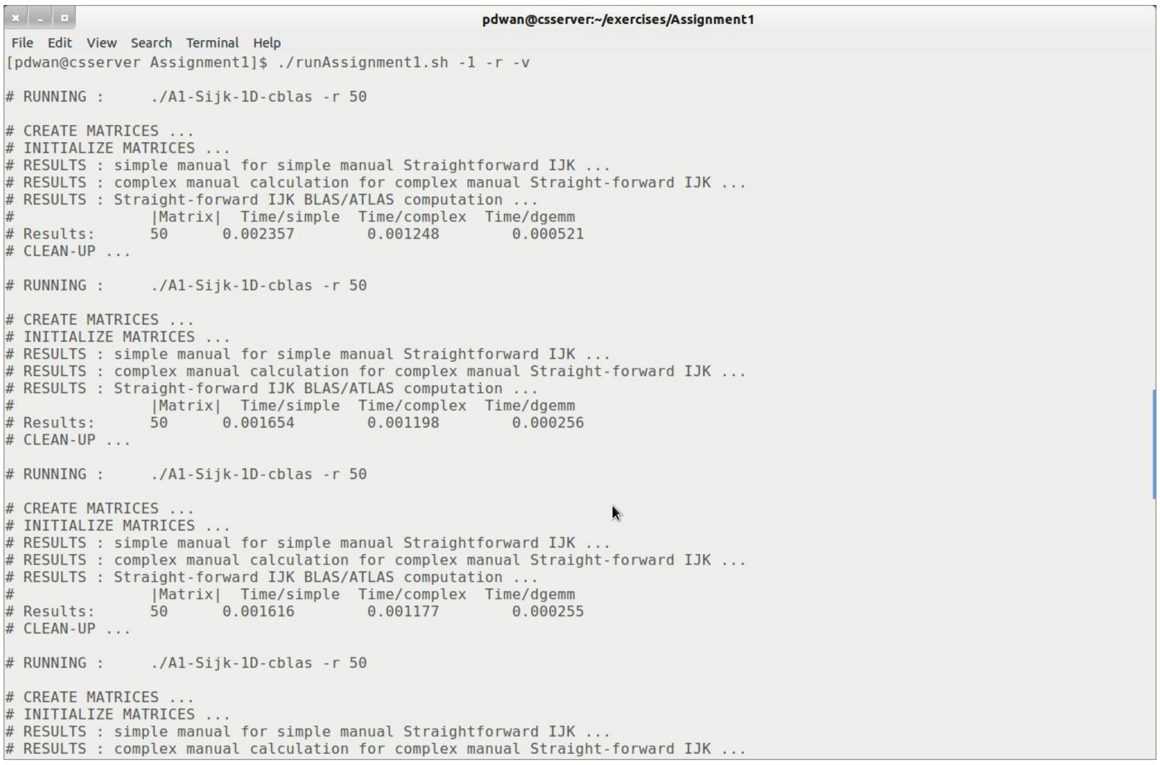

pdwan@csserver:~/exercises/Assignment1
File Edit View Search Terminal Help

```
[pdwan@csserver Assignment1]$ ./runAssignment1.sh

USAGE : ./runAssignment1.sh \
                        -a|--all -1|--simple -2|--ijk -3|--kij -d1|--atlas -d2|--cblas -r|--random -i|--increment \
                        -m|--matrix<n> -b|--block <b> -v|--values -?|-h|--help

TO :    Calculate |C| = |A| x |B| using 1 -> 3 algoritms : Straight-forward IJK, Blocked IJK and Blocked KIJ.
LOGS :  Created in current dir and moved to <logDir> :
        <file>.txt :    matrix values for matrices |A| |B| & |C|,
        <file>.dat :    timing data for each computation
        <file>.log :    summary of stdout.

WHERE : -a|--all        Calculate data for all algorithms via separate .c programs to multiply |A|x|B| -> |C|
                        Straightforward IJK algorithm : A1-Sijk-1D.c
                        Blocked IJK algorithm using square bxb blocks : A1-Bijk-1D.c
                        Blocked KIJ algorithm using square bxb blocks : A1-Bkij-1D.c
        -1|--simple     Calculate data for only the algorithm
                        Straightforward IJK algorithm : A1-Sijk-1D.c
        -2|--bijk       Calculate data for only the algorithm
                        Blocked IJK algorithm using square bxb blocks : A1-Bijk-1D.c
        -3|--bkij       Calculate data for only the algorithm
                        Blocked KIJ algorithm using square bxb blocks : A1-Bkij-1D.c

        -d1|--atlas     Compile .c source files using dgemm atlas
        -d2|--cblas     Compile .c source files using dgemm cblas
                        '-d1|--atlas' and '-d2|--cblas ' are mutually exclusive.
        -r|--random     Initialize |A| & |B| with random numbers and |C| with '0'
        -i|--increment  Initialize |A| & |B| incrementally with <column> value and |C| with '0'
                        '-i|--increment' & '-r|--random' are mutually exclusive

        -m|--matrix <N> Matrix size, if  invalid, (matrix size > max) set to [1000]
        -b|--block <B>  Block matrix size, if invalid (matrix % block != 0 or block size > max),
                        set to [100] and matrix size set to [1000].
        -v|--values     Use predefined range of valid values for <nx> and <nb> as follows :
                        <NXArray>       ( 50 50 50 100 100 100 500 500 500 500 1000 1000 1000 1000 ) : can be reset if needed
                        <NBArray>       ( 2 5 10 5 10 20 5 10 20 50 5 10 50 100 ) : can be reset if needed
                        '-m|--matrix <n>' & '-b|--block <b>' are mutually exclusive of '-v|--values'.

        -?|-h|--help    usage

[pdwan@csserver Assignment1]$ 
```

Execute this script in the home directory of Assignment 1.

Sample execution follows for :

`$ ./runAssignment.sh -1 -a -v`


pdwan@csserver:~/exercises/Assignment1
File Edit View Search Terminal Help

```
[pdwan@csserver Assignment1]$ ./runAssignment1.sh -1 -r -v

# RUNNING :     ./A1-Sijk-1D-cblas -r 50

# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual for simple manual Straightforward IJK ...
# RESULTS : complex manual calculation for complex manual Straight-forward IJK ...
# RESULTS : Straight-forward IJK BLAS/ATLAS computation ...
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      50      0.002357        0.001248        0.000521
# CLEAN-UP ...

# RUNNING :     ./A1-Sijk-1D-cblas -r 50

# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual for simple manual Straightforward IJK ...
# RESULTS : complex manual calculation for complex manual Straight-forward IJK ...
# RESULTS : Straight-forward IJK BLAS/ATLAS computation ...
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      50      0.001654        0.001198        0.000256
# CLEAN-UP ...

# RUNNING :     ./A1-Sijk-1D-cblas -r 50

# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual for simple manual Straightforward IJK ...
# RESULTS : complex manual calculation for complex manual Straight-forward IJK ...
# RESULTS : Straight-forward IJK BLAS/ATLAS computation ...
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      50      0.001616        0.001177        0.000255
# CLEAN-UP ...

# RUNNING :     ./A1-Sijk-1D-cblas -r 50

# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual for simple manual Straightforward IJK ...
# RESULTS : complex manual calculation for complex manual Straight-forward IJK ...
```

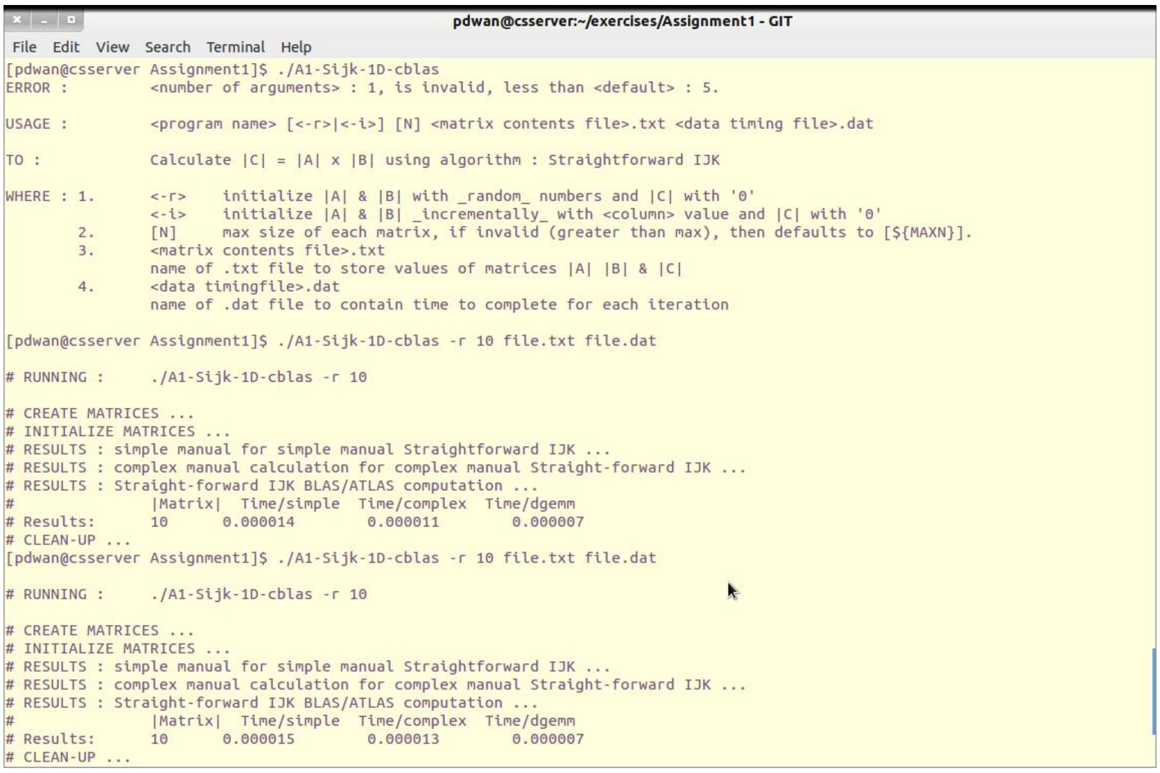
Note: Please retain the overall directory structure when unzipping.

Note that the script `./runAssignment1.sh` allows two types of implementation

- Multiple iteration : use the switch <-v|--values>, when a predefined range applies for [N] : matrix size and [T] : number of threads applicable.
- Single iteration : use the switch <-m|--matrix> [N] where the user specifies the values for [N] : matrix size and [T] : number of threads applicable.

Running A1-Sijk-1D-<atlas | cblas> : *standalone*

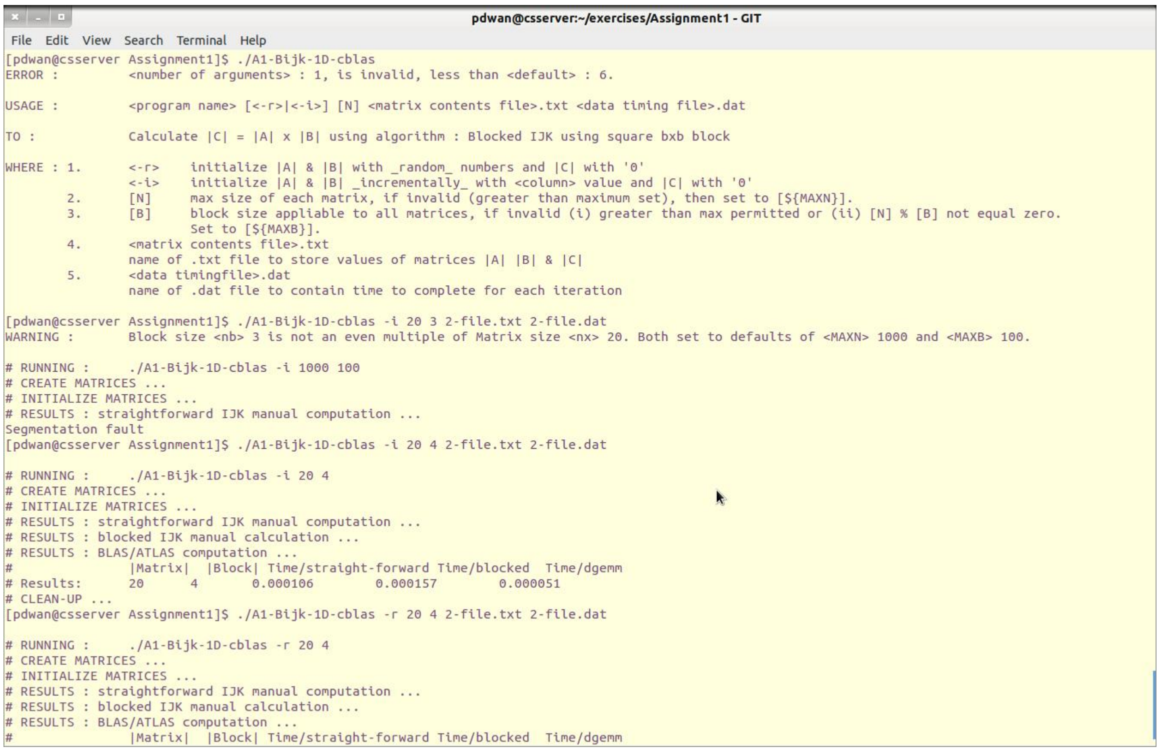
The compiled .c program may also be run standalone. Usage and sample execution follows :



```
[pdwan@csserver Assignment1]$ ./A1-Sijk-1D-cblas
ERROR :         <number of arguments> : 1, is invalid, less than <default> : 5.

USAGE :         <program name> [<-r>|<-i>] [N] <matrix contents file>.txt <data timing file>.dat

TO :            Calculate |C| = |A| x |B| using algorithm : Straightforward IJK

WHERE : 1.      <-r>    initialize |A| & |B| with _random_ numbers and |C| with '0'
                <-i>    initialize |A| & |B| _incrementally_ with <column> value and |C| with '0'
        2.      [N]     max size of each matrix, if invalid (greater than max), then defaults to [${MAXN}].
        3.      <matrix contents file>.txt
                name of .txt file to store values of matrices |A| |B| & |C|
        4.      <data timingfile>.dat
                name of .dat file to contain time to complete for each iteration

[pdwan@csserver Assignment1]$ ./A1-Sijk-1D-cblas -r 10 file.txt file.dat

# RUNNING :     ./A1-Sijk-1D-cblas -r 10

# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual for simple manual Straightforward IJK ...
# RESULTS : complex manual calculation for complex manual Straight-forward IJK ...
# RESULTS : Straight-forward IJK BLAS/ATLAS computation ...
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      10      0.000014        0.000011        0.000007
# CLEAN-UP ...
[pdwan@csserver Assignment1]$ ./A1-Sijk-1D-cblas -r 10 file.txt file.dat

# RUNNING :     ./A1-Sijk-1D-cblas -r 10

# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual for simple manual Straightforward IJK ...
# RESULTS : complex manual calculation for complex manual Straight-forward IJK ...
# RESULTS : Straight-forward IJK BLAS/ATLAS computation ...
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      10      0.000015        0.000013        0.000007
# CLEAN-UP ...
```

Running A1-Bijk-1D-<atlas | cblas>: *standalone*

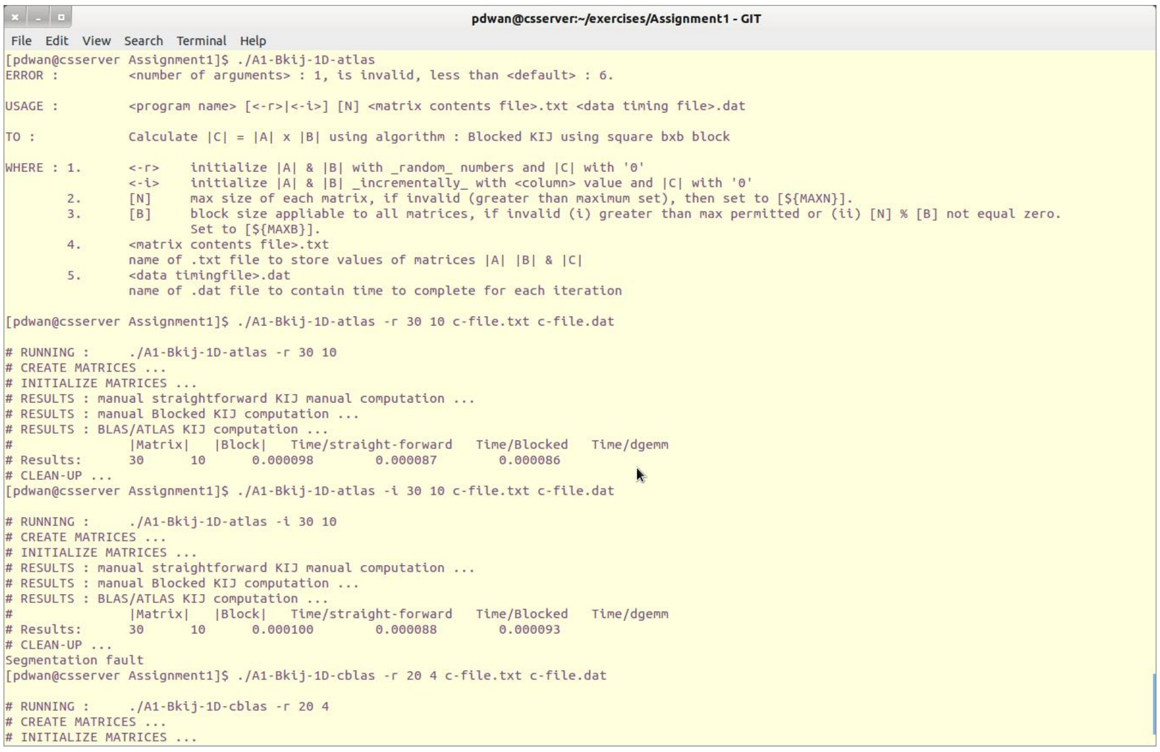
The compiled .c program may also be run standalone. Usage and sample execution follows :


pdwan@csserver:~/exercises/Assignment1 - GIT
File Edit View Search Terminal Help

```
[pdwan@csserver Assignment1]$ ./A1-Bijk-1D-cblas
ERROR :         <number of arguments> : 1, is invalid, less than <default> : 6.

USAGE :         <program name> [<-r>|<-i>] [N] <matrix contents file>.txt <data timing file>.dat

TO :            Calculate |C| = |A| x |B| using algorithm : Blocked IJK using square bxb block

WHERE : 1.      <-r>    initialize |A| & |B| with _random_ numbers and |C| with '0'
                <-i>    initialize |A| & |B| _incrementally_ with <column> value and |C| with '0'
        2.      [N]     max size of each matrix, if invalid (greater than maximum set), then set to [${MAXN}].
        3.      [B]     block size appliable to all matrices, if invalid (i) greater than max permitted or (ii) [N] % [B] not equal zero.
                        Set to [${MAXB}].
        4.      <matrix contents file>.txt
                name of .txt file to store values of matrices |A| |B| & |C|
        5.      <data timingfile>.dat
                name of .dat file to contain time to complete for each iteration

[pdwan@csserver Assignment1]$ ./A1-Bijk-1D-cblas -i 20 3 2-file.txt 2-file.dat
WARNING :       Block size <nb> 3 is not an even multiple of Matrix size <nx> 20. Both set to defaults of <MAXN> 1000 and <MAXB> 100.

# RUNNING :     ./A1-Bijk-1D-cblas -i 1000 100
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : straightforward IJK manual computation ...
Segmentation fault
[pdwan@csserver Assignment1]$ ./A1-Bijk-1D-cblas -i 20 4 2-file.txt 2-file.dat

# RUNNING :     ./A1-Bijk-1D-cblas -i 20 4
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : straightforward IJK manual computation ...
# RESULTS : blocked IJK manual calculation ...
# RESULTS : BLAS/ATLAS computation ...
#               |Matrix|  |Block| Time/straight-forward Time/blocked  Time/dgemm
# Results:      20      4       0.000106        0.000157        0.000051
# CLEAN-UP ...
[pdwan@csserver Assignment1]$ ./A1-Bijk-1D-cblas -r 20 4 2-file.txt 2-file.dat

# RUNNING :     ./A1-Bijk-1D-cblas -r 20 4
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : straightforward IJK manual computation ...
# RESULTS : blocked IJK manual calculation ...
# RESULTS : BLAS/ATLAS computation ...
#               |Matrix|  |Block| Time/straight-forward Time/blocked  Time/dgemm
```

Running A1-Bkij-1D-<atlas | cblas>: *standalone*

The compiled .c program may also be run standalone. Usage and sample execution follows :


pdwan@csserver:~/exercises/Assignment1 - GIT
File Edit View Search Terminal Help

```
[pdwan@csserver Assignment1]$ ./A1-Bkij-1D-atlas
ERROR :         <number of arguments> : 1, is invalid, less than <default> : 6.

USAGE :         <program name> [<-r>|<-i>] [N] <matrix contents file>.txt <data timing file>.dat

TO :            Calculate |C| = |A| x |B| using algorithm : Blocked KIJ using square bxb block

WHERE : 1.      <-r>    initialize |A| & |B| with _random_ numbers and |C| with '0'
                <-i>    initialize |A| & |B| _incrementally_ with <column> value and |C| with '0'
        2.      [N]     max size of each matrix, if invalid (greater than maximum set), then set to [${MAXN}].
        3.      [B]     block size appliable to all matrices, if invalid (i) greater than max permitted or (ii) [N] % [B] not equal zero.
                        Set to [${MAXB}].
        4.      <matrix contents file>.txt
                name of .txt file to store values of matrices |A| |B| & |C|
        5.      <data timingfile>.dat
                name of .dat file to contain time to complete for each iteration

[pdwan@csserver Assignment1]$ ./A1-Bkij-1D-atlas -r 30 10 c-file.txt c-file.dat

# RUNNING :     ./A1-Bkij-1D-atlas -r 30 10
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : manual straightforward KIJ manual computation ...
# RESULTS : manual Blocked KIJ computation ...
# RESULTS : BLAS/ATLAS KIJ computation ...
#               |Matrix|   |Block|   Time/straight-forward   Time/Blocked   Time/dgemm
# Results:      30      10      0.000098        0.000087        0.000086
# CLEAN-UP ...
[pdwan@csserver Assignment1]$ ./A1-Bkij-1D-atlas -i 30 10 c-file.txt c-file.dat

# RUNNING :     ./A1-Bkij-1D-atlas -i 30 10
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : manual straightforward KIJ manual computation ...
# RESULTS : manual Blocked KIJ computation ...
# RESULTS : BLAS/ATLAS KIJ computation ...
#               |Matrix|   |Block|   Time/straight-forward   Time/Blocked   Time/dgemm
# Results:      30      10      0.000100        0.000088        0.000093
# CLEAN-UP ...
Segmentation fault
[pdwan@csserver Assignment1]$ ./A1-Bkij-1D-cblas -r 20 4 c-file.txt c-file.dat

# RUNNING :     ./A1-Bkij-1D-cblas -r 20 4
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
```

Log Files Obtained

Data text files suitable containing the values of the computation used for matrices |A| and |B| and the results stored in |C| are saved in the appropriate log files. File naming convention via the script is :

<data log file name>	`Values-<time>-<A1-Sijk-1D->-<iteration>.txt`
example:	`Values-20140715.170928-A1-Bijk-1D-0.txt` `Values-20140715.171707-A1-Bkij-1D-10.txt`

Single iteration also applies where the user enters arbitrary, valid values for matrix size and does not use the scripts and the other required parameters. Each new matrices |A| and |B| and the results in |C| were saved to the data file, thus simple validation using *LibreOffice Calc*.

A summary file containing processing time for each computation (manual and BLAS) for is also saved. This is in a format suitable for us with GNUplot.

<timing log file name>	`Data-<time>-<A1-Sijk-1D->.dat`
example:	`Data-20140715.171337-A1-Bijk-1D.dat` `Data-20140715.172124-A1-Bkij-1D.dat`

I did not save a separate .dat file for each run of the script for each algorithm. Instead each .dat file contains the time taken for each matrix size (and block size, if applicable) for the preset range of values. RunAssignment1.sh may be updated with more if needed but the following are those in use at the moment.

```
# Matrix size - range :
declare -a NXArray=( 50 50 50 50 50 50 100 100 100 100 100 100 )
# Equivalent Block size - range :
declare -a NBArray=( 10 10 10 10 10 10 20 20 20 20 20 20 )

# Matrix size - range :
declare -a NXArray=( 50 50 50 100 100 100 500 500 500 500 1000 1000 1000 1000 )
# Equivalent Block size - range :
declare -a NBArray=( 2 5 10 5 10 20 5 10 20 50 5 10 50 100 )
```

For compilation using the script, a suffix of ***-atlas*** indicates compilation for atlas and a suffix of ***-cblas*** indicates that the c program was compiled via cblas.

Finally a log file containing a listing of each algorithm used for that iteration.

After each run, all .log, .txt, .dat and .bup files are copied to the directory *logDir/.*

If ***./A1-Sijk-1D-cblas*** or any of the other algorithm files is used without the script then .dat and .txt files may be named whatever the user wishes and no .log file applies.

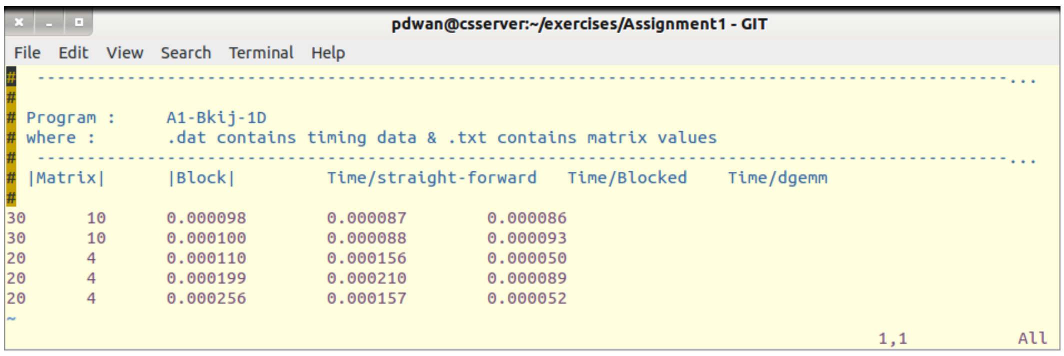
I wished to keep each .c program as clean as possible and so all production setup was completed in the script for each assignment. Thus file creation and validation for each iteration was completed before the .c program was even called. Simple validation of the arguments passed to each .c program is also completed if ran standalone.

I also spot-checked the results as practical. Results spot-check are detailed in Appendix I – Validate Results, while sample results are listed in Appendix II – summary of time taken using predefined values.

GNUplot Execution

I followed the same structure for each .dat file as produced, an example follows :

Sample .dat file



pdwan@csserver:~/exercises/Assignment1 - GIT
File Edit View Search Terminal Help

```
# ---------------------------------------------------------------------------------...
#
# Program :     A1-Bkij-1D
# where :       .dat contains timing data & .txt contains matrix values
# ---------------------------------------------------------------------------------...
# |Matrix|      |Block|         Time/straight-forward   Time/Blocked    Time/dgemm
#
30      10      0.000098        0.000087        0.000086
30      10      0.000100        0.000088        0.000093
20      4       0.000110        0.000156        0.000050
20      4       0.000199        0.000210        0.000089
20      4       0.000256        0.000157        0.000052
~
                                                                  1,1           All
```

If wished, the .txt file contains the matrices |A| and |B| used to calculate |C| and the type of computation applicable and the time taken to complete. The .dat file is just a summary of the matrix and block sizes (when the later is applicable) as well as time taken for each type of computation.

The contents of each .dat was then presented in graphical format using GNUplot, comparing times taken for manual and for BLAS/ATLAS computations.

Sample GNUplot program execution

```
# To execute, launch GNUplot and run :
# gnuplot> load <filename.gp>
# making sure that the data file name used is updated if needed.
# ------------------------------------------------------------------------

# Paula Dwan : Assignment 1
reset
set xtic auto
set ytic auto
set size 1,1
set grid
set key outside
#
set title 'Blocked KIJ Comparison : Block size -v- Time taken'
set ylabel 'Time taken / seconds'
set xlabel 'Block size : BxB'
set xrange [0:1100]
set yrange [0:34]
set xtics (100,200,300,400,500,600,700,800,900,1000,1100)
set ytics
(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28
,29,30,31,32,33,34)
set origin 0,0
set key outside
plot 'logDir/Data-Bkij.dat' u 1:3 t 'manual simple' w l lw 0.8 lc rgb
'blue', 'logDir/Data-Bkij.dat' u 1:4 t 'manual complex' w l lw 0.8 lc rgb
'black', 'logDir/Data-Bkij.dat' u 1:5 t 'dgemm' w l lw 0.8 lc rgb 'red'
#
pause -1
```

Thankfully for Linux (Ubuntu) – I could install and run GNUplot locally.

Screen shots of each were taken and added to the section Summary Results.

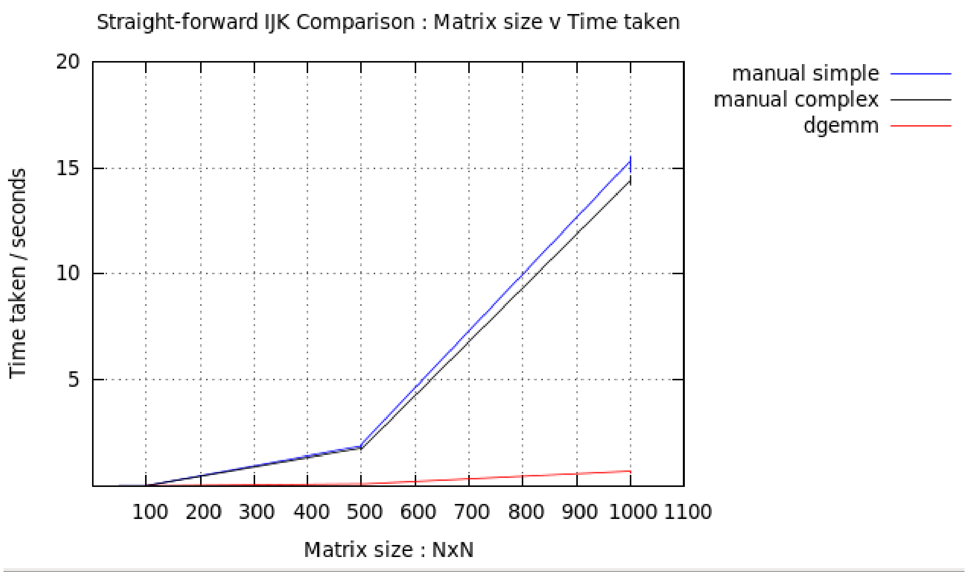
SUMMARY RESULTS :

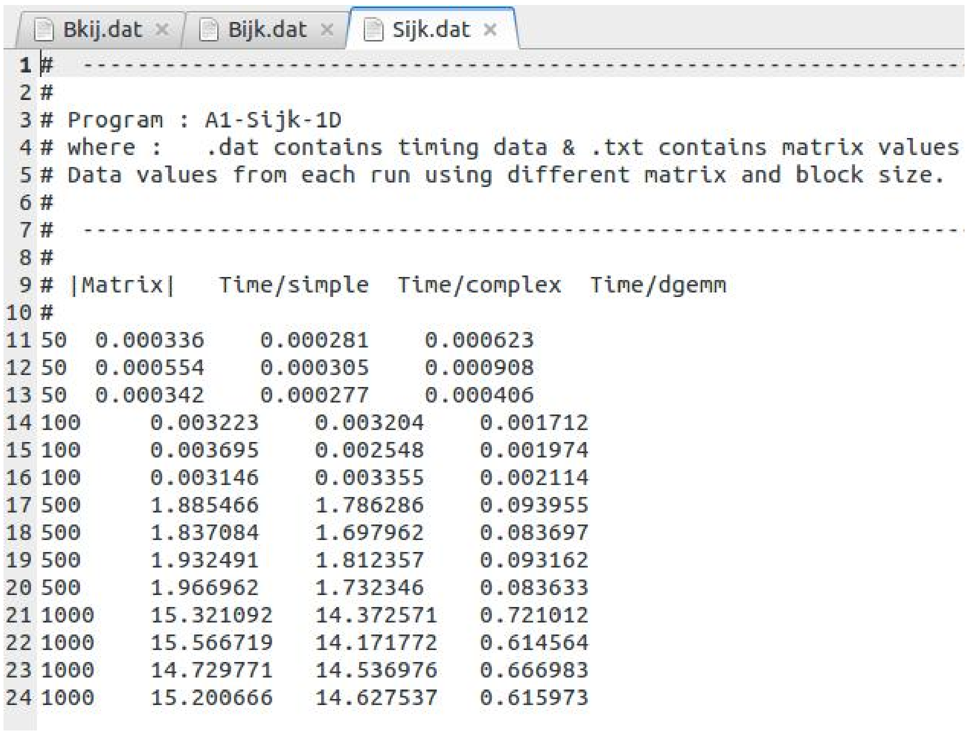
Build/plot:	1)	The dependence of the execution time of each program on the matrix size n and the block size b .
	2)	The speedup of the Blocked algorithms over the non-blocked one as a function of the matrix size and the block size.
	3)	Compare the fastest program with the BLAS/ATLAS routine **dgemm** implementing the same operation.

GNUPLOT GRAPHS – ATLAS USING MATRIX SIZES
(50 50 50 100 100 100 500 500 500 500 1000 1000 1000 1000)

$ ./RUNASSIGNMENT1.SH -A <-R|-I> -V -D1

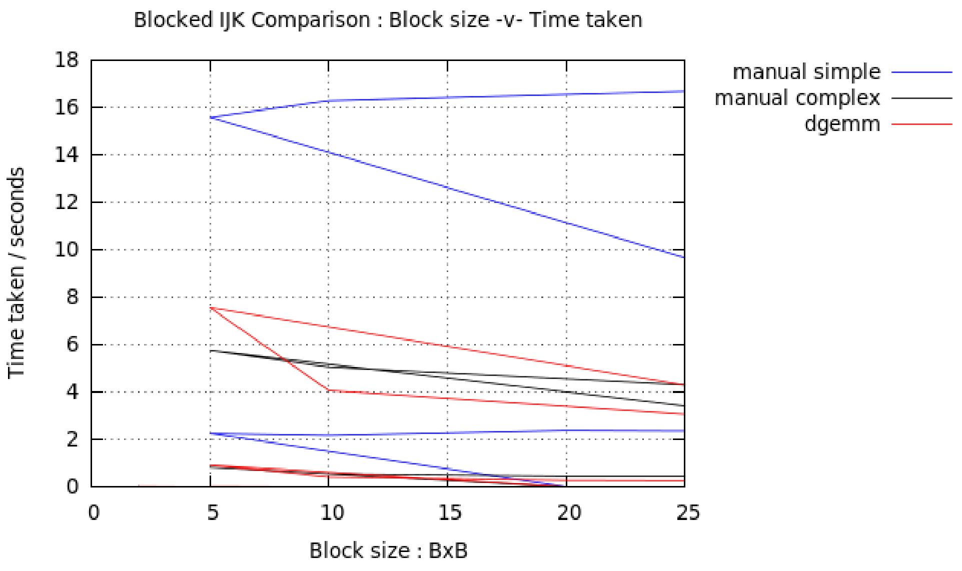
Simple IJK

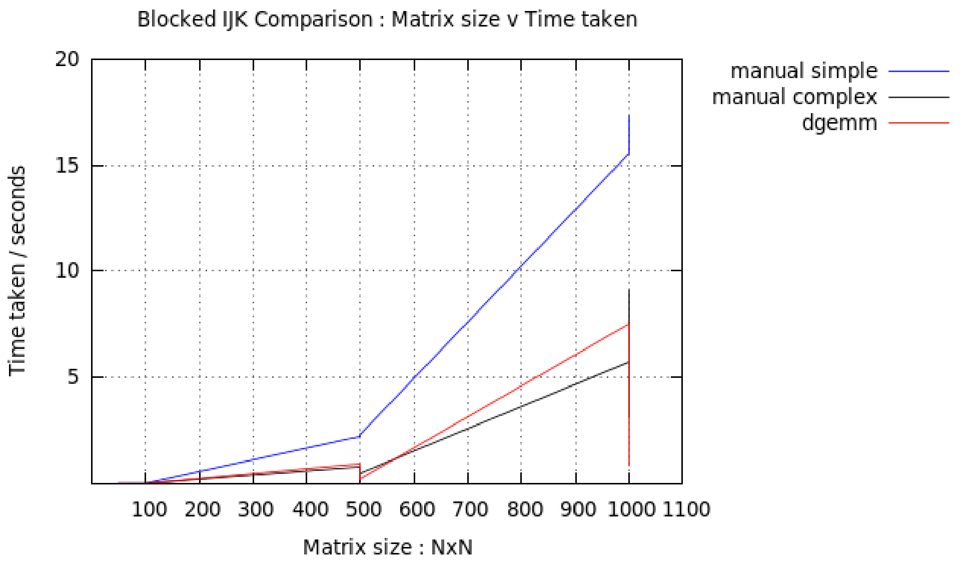


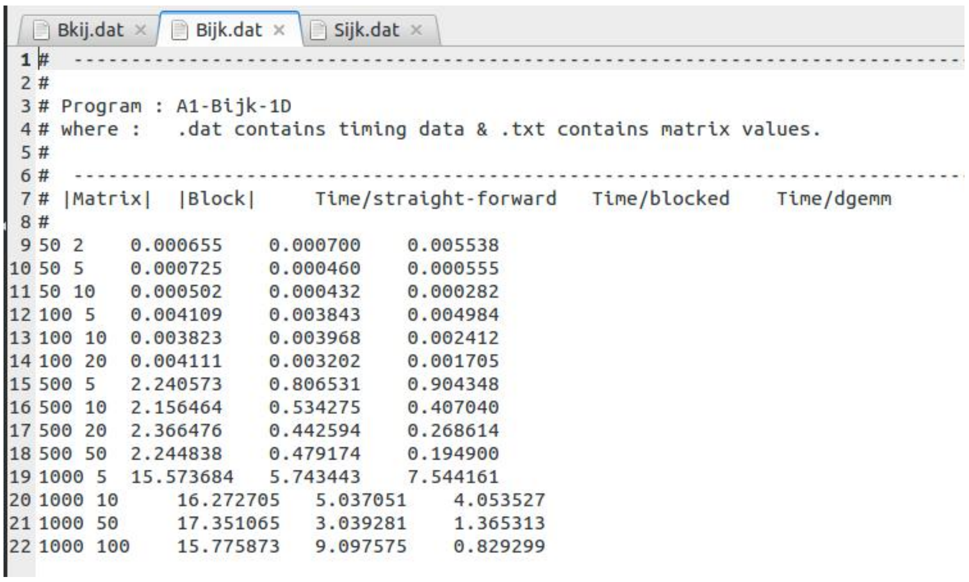


```
# ------------------------------------------------------------------
#
# Program : A1-Sijk-1D
# where :    .dat contains timing data & .txt contains matrix values
# Data values from each run using different matrix and block size.
#
# ------------------------------------------------------------------
#
# |Matrix|   Time/simple  Time/complex  Time/dgemm
#
50   0.000336    0.000281    0.000623
50   0.000554    0.000305    0.000908
50   0.000342    0.000277    0.000406
100      0.003223    0.003204    0.001712
100      0.003695    0.002548    0.001974
100      0.003146    0.003355    0.002114
500      1.885466    1.786286    0.093955
500      1.837084    1.697962    0.083697
500      1.932491    1.812357    0.093162
500      1.966962    1.732346    0.083633
1000     15.321092   14.372571   0.721012
1000     15.566719   14.171772   0.614564
1000     14.729771   14.536976   0.666983
1000     15.200666   14.627537   0.615973
```

Blocked IJK

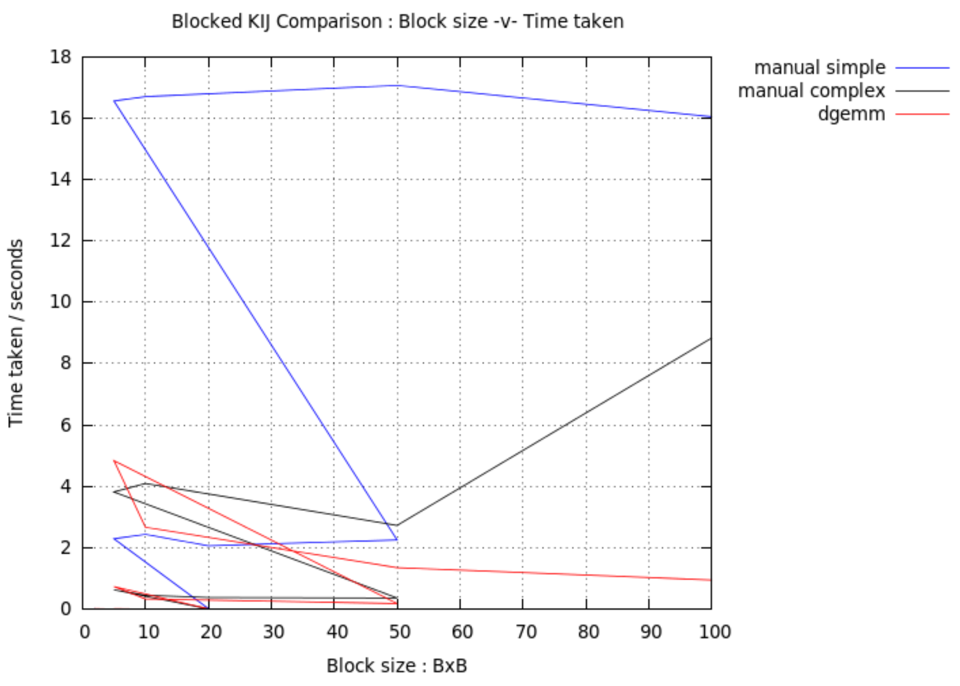


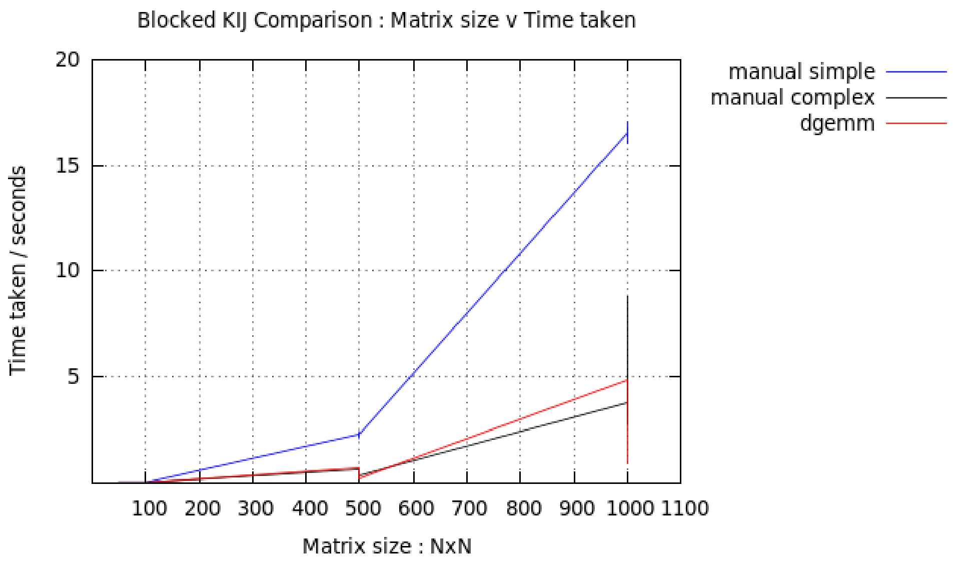


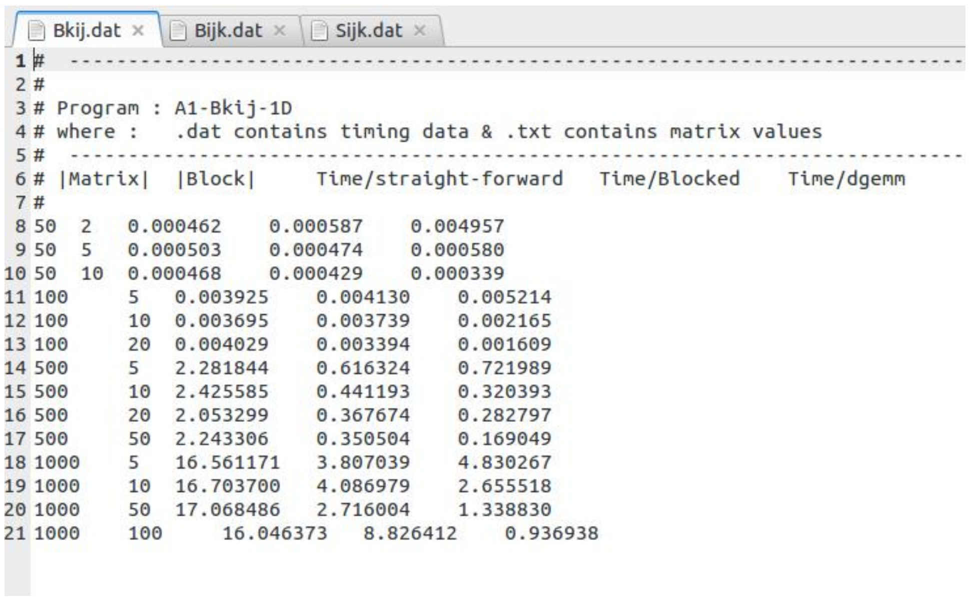


```
# -----------------------------------------------------------------------
#
# Program : A1-Bijk-1D
# where :    .dat contains timing data & .txt contains matrix values.
#
# -----------------------------------------------------------------------
# |Matrix|  |Block|      Time/straight-forward   Time/blocked     Time/dgemm
#
50 2     0.000655    0.000700    0.005538
50 5     0.000725    0.000460    0.000555
50 10    0.000502    0.000432    0.000282
100 5    0.004109    0.003843    0.004984
100 10   0.003823    0.003968    0.002412
100 20   0.004111    0.003202    0.001705
500 5    2.240573    0.806531    0.904348
500 10   2.156464    0.534275    0.407040
500 20   2.366476    0.442594    0.268614
500 50   2.244838    0.479174    0.194900
1000 5   15.573684   5.743443    7.544161
1000 10      16.272705   5.037051    4.053527
1000 50      17.351065   3.039281    1.365313
1000 100     15.775873   9.097575    0.829299
```

Blocked KIJ





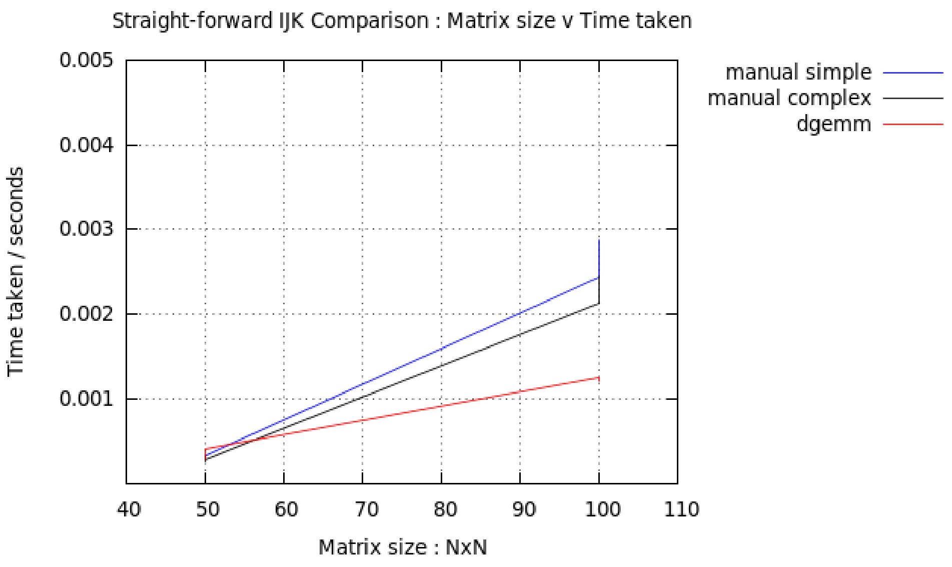


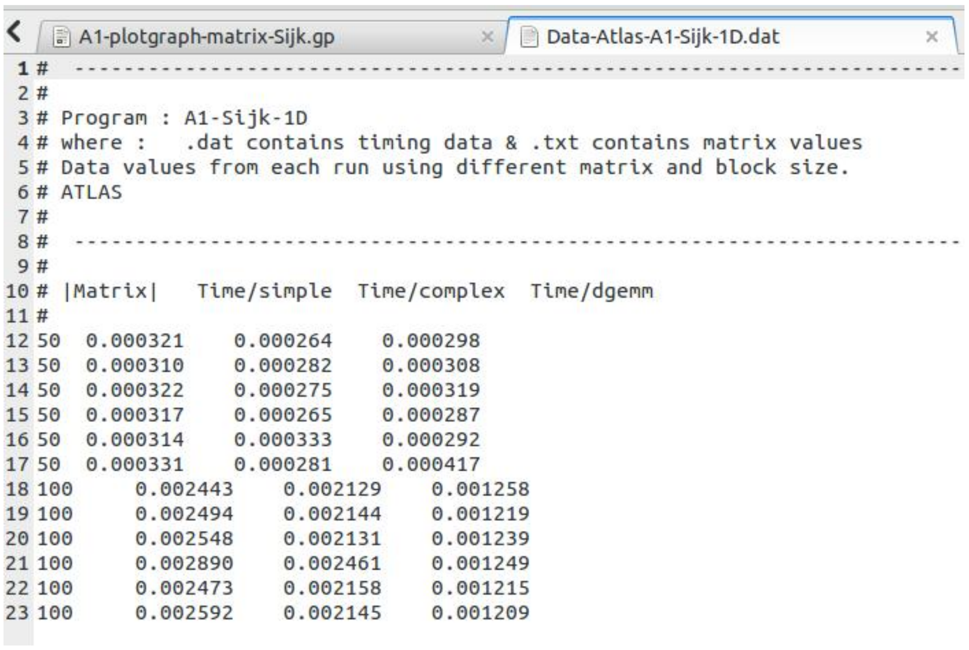
```
# ------------------------------------------------------------------------------------
#
# Program : A1-Bkij-1D
# where :   .dat contains timing data & .txt contains matrix values
# ------------------------------------------------------------------------------------
# |Matrix|  |Block|     Time/straight-forward   Time/Blocked    Time/dgemm
#
50  2   0.000462    0.000587    0.004957
50  5   0.000503    0.000474    0.000580
50  10  0.000468    0.000429    0.000339
100     5   0.003925    0.004130    0.005214
100     10  0.003695    0.003739    0.002165
100     20  0.004029    0.003394    0.001609
500     5   2.281844    0.616324    0.721989
500     10  2.425585    0.441193    0.320393
500     20  2.053299    0.367674    0.282797
500     50  2.243306    0.350504    0.169049
1000    5   16.561171   3.807039    4.830267
1000    10  16.703700   4.086979    2.655518
1000    50  17.068486   2.716004    1.338830
1000    100     16.046373   8.826412    0.936938
```

GNUPLOT GRAPHS – ATLAS USING MATRIX SIZES (50 50 50 50 50 50 100 100 100 100 100 100)

$ ./RUNASSIGNMENT1.SH -A <-I|-R> -V -D1

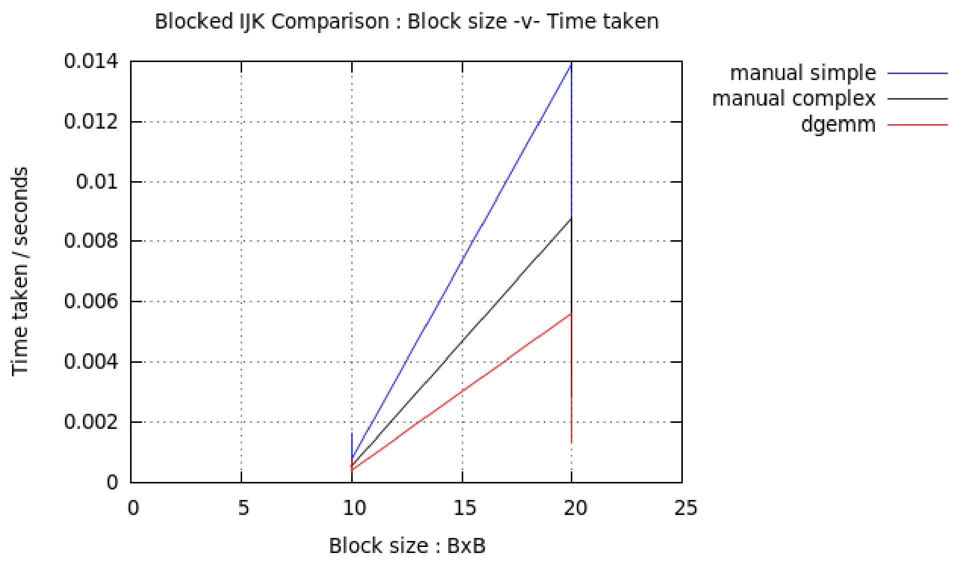
Simple IJK

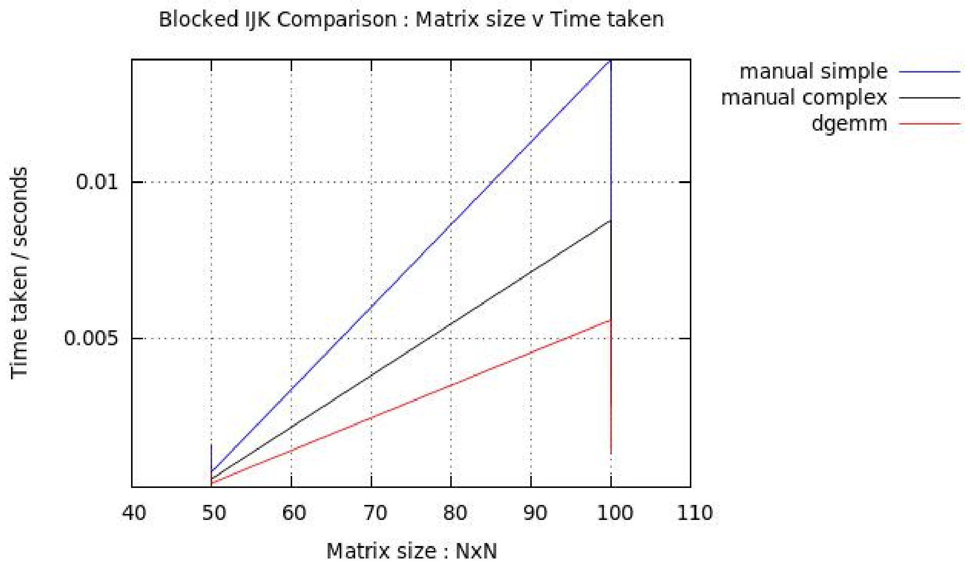


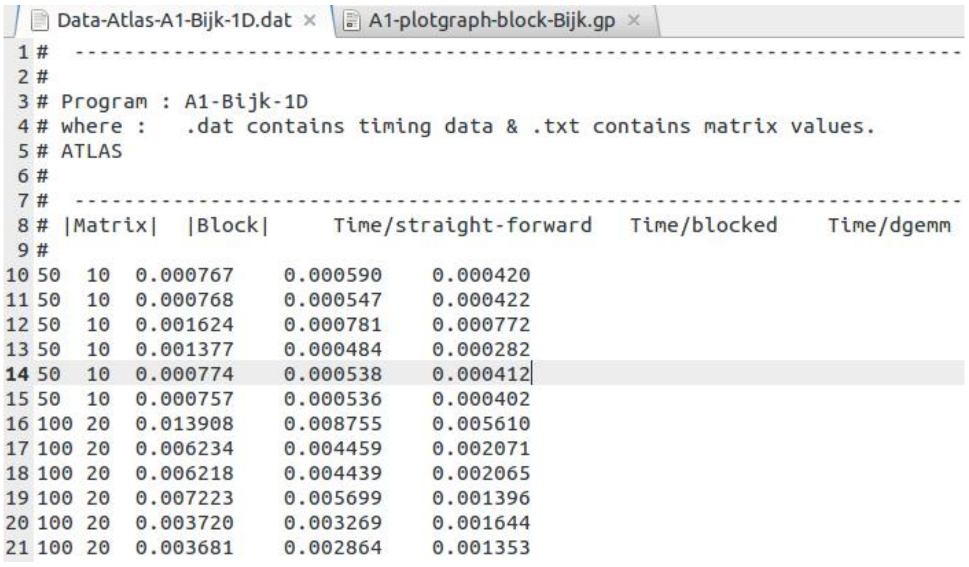


```
#  ----------------------------------------------------------------------
#
# Program : A1-Sijk-1D
# where :   .dat contains timing data & .txt contains matrix values
# Data values from each run using different matrix and block size.
# ATLAS
#
#  ----------------------------------------------------------------------
#
# |Matrix|   Time/simple  Time/complex  Time/dgemm
#
50  0.000321    0.000264    0.000298
50  0.000310    0.000282    0.000308
50  0.000322    0.000275    0.000319
50  0.000317    0.000265    0.000287
50  0.000314    0.000333    0.000292
50  0.000331    0.000281    0.000417
100     0.002443    0.002129    0.001258
100     0.002494    0.002144    0.001219
100     0.002548    0.002131    0.001239
100     0.002890    0.002461    0.001249
100     0.002473    0.002158    0.001215
100     0.002592    0.002145    0.001209
```

Blocked IJK

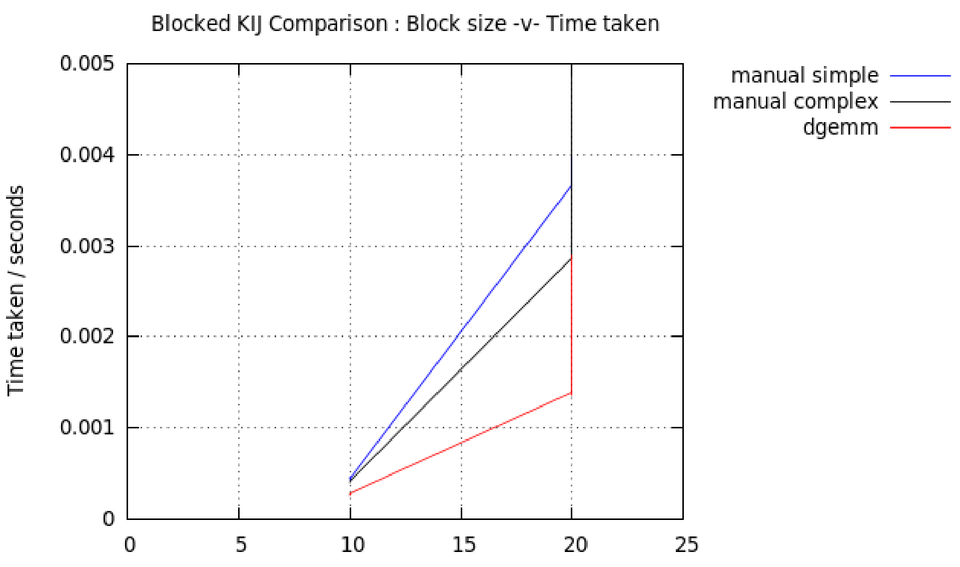


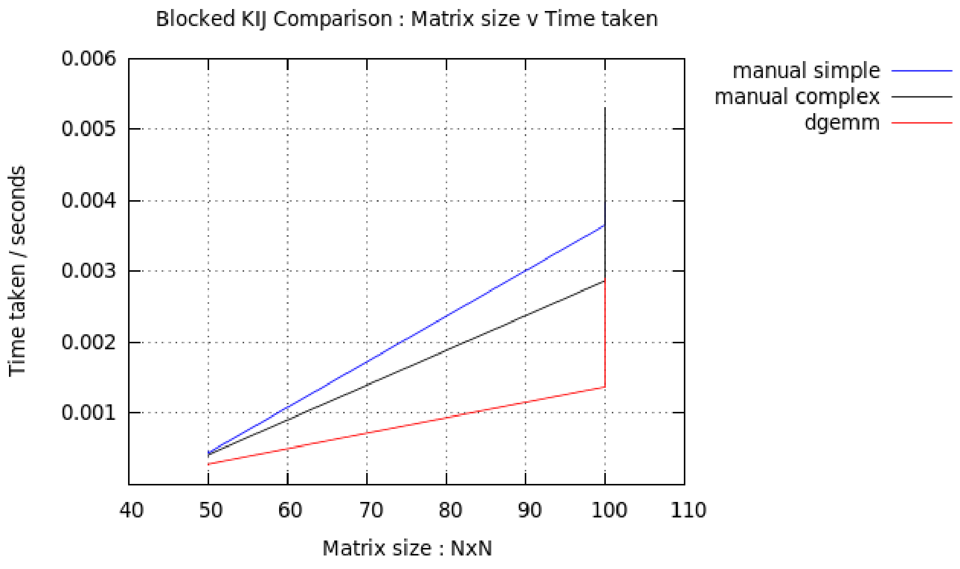


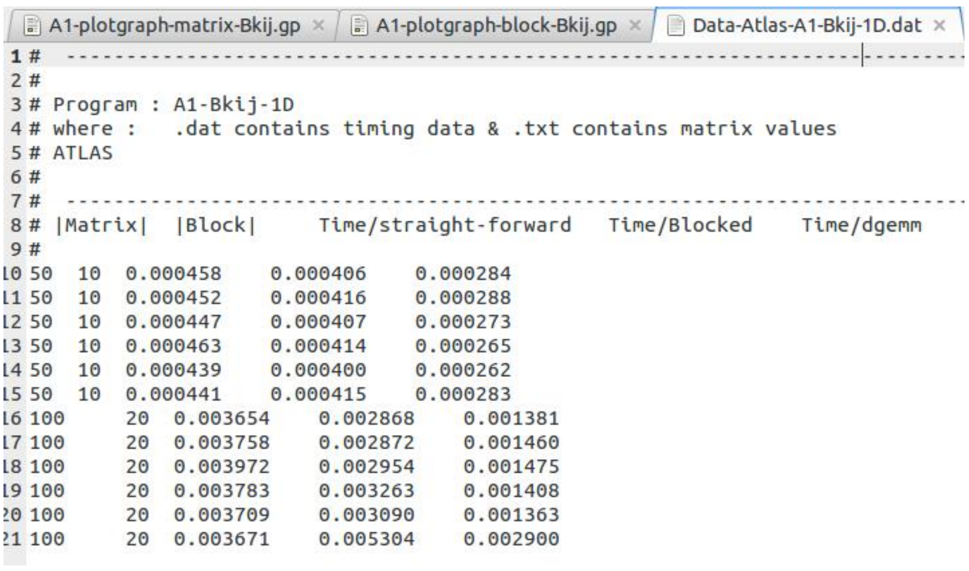


```
# --------------------------------------------------------------------------
#
# Program  : A1-Bijk-1D
# where  :   .dat contains timing data & .txt contains matrix values.
# ATLAS
#
# --------------------------------------------------------------------------
# |Matrix|  |Block|     Time/straight-forward    Time/blocked     Time/dgemm
#
50   10   0.000767     0.000590     0.000420
50   10   0.000768     0.000547     0.000422
50   10   0.001624     0.000781     0.000772
50   10   0.001377     0.000484     0.000282
50   10   0.000774     0.000538     0.000412
50   10   0.000757     0.000536     0.000402
100  20   0.013908     0.008755     0.005610
100  20   0.006234     0.004459     0.002071
100  20   0.006218     0.004439     0.002065
100  20   0.007223     0.005699     0.001396
100  20   0.003720     0.003269     0.001644
100  20   0.003681     0.002864     0.001353
```

Blocked KIJ





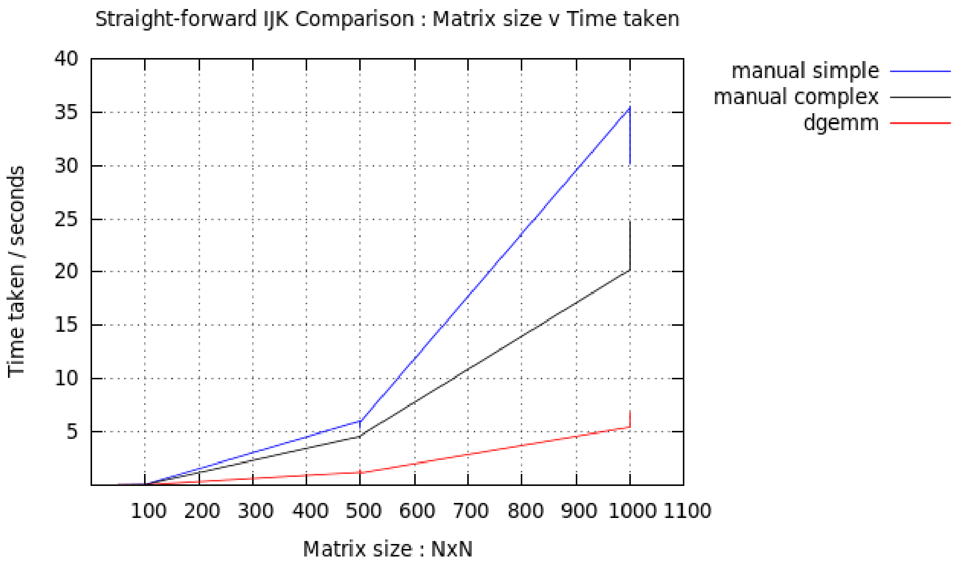

A1-plotgraph-matrix-Bkij.gp
A1-plotgraph-block-Bkij.gp
Data-Atlas-A1-Bkij-1D.dat

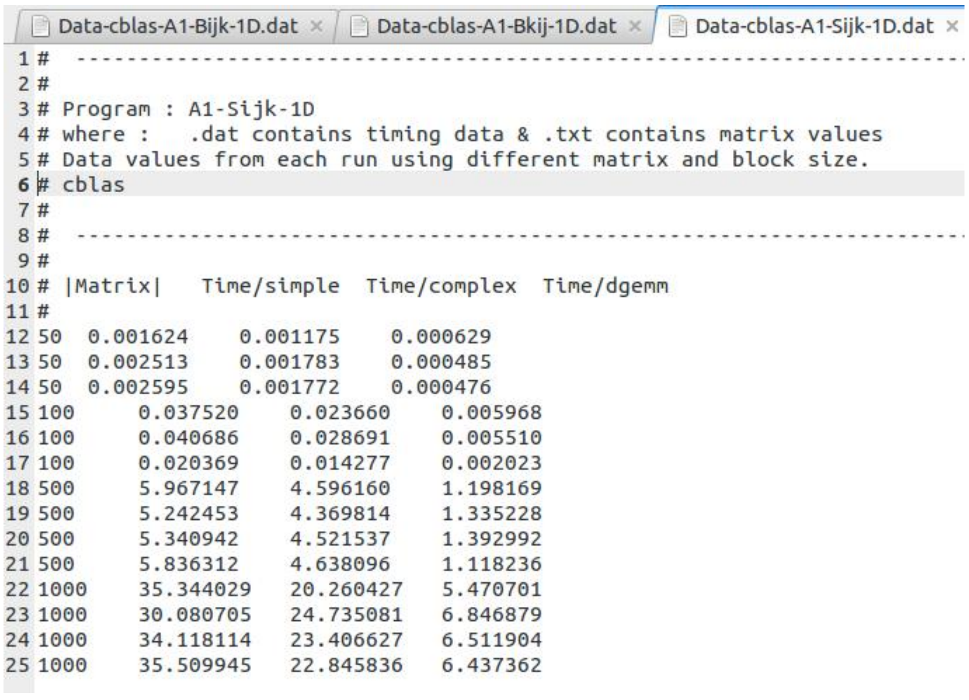
```
#  ----------------------------------------------------------------------------
#
# Program : A1-Bkij-1D
# where :   .dat contains timing data & .txt contains matrix values
# ATLAS
#
#  ----------------------------------------------------------------------------
# |Matrix|  |Block|     Time/straight-forward    Time/Blocked     Time/dgemm
#
50  10  0.000458    0.000406    0.000284
50  10  0.000452    0.000416    0.000288
50  10  0.000447    0.000407    0.000273
50  10  0.000463    0.000414    0.000265
50  10  0.000439    0.000400    0.000262
50  10  0.000441    0.000415    0.000283
100     20  0.003654    0.002868    0.001381
100     20  0.003758    0.002872    0.001460
100     20  0.003972    0.002954    0.001475
100     20  0.003783    0.003263    0.001408
100     20  0.003709    0.003090    0.001363
100     20  0.003671    0.005304    0.002900
```

GNUPLOT GRAPHS – CBLAS USING MATRIX SIZES (50 50 50 100 100 100 500 500 500 500 1000 1000 1000 1000)

$ ./RUNASSIGNMENT1.SH -A <-R|-I> -V -D2

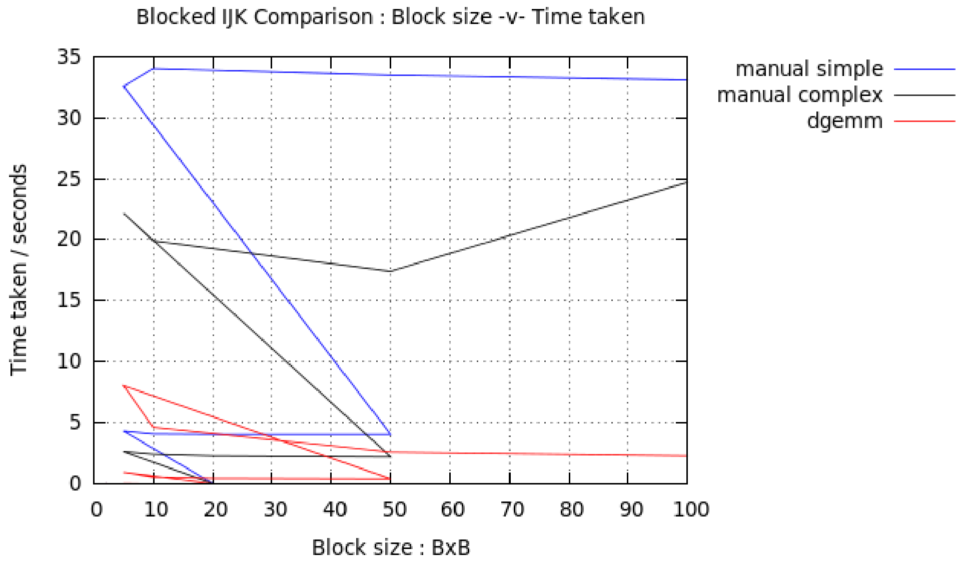
Simple IJK

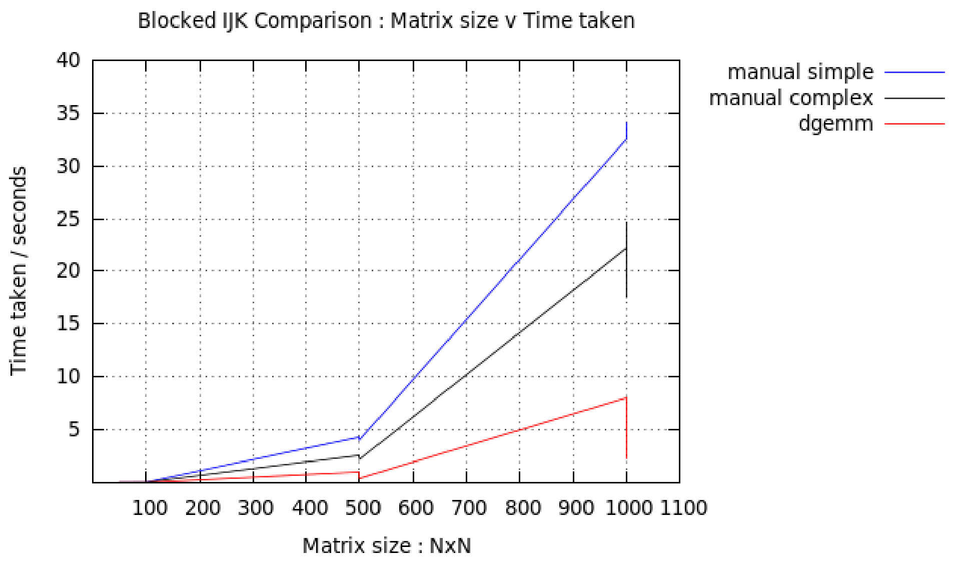


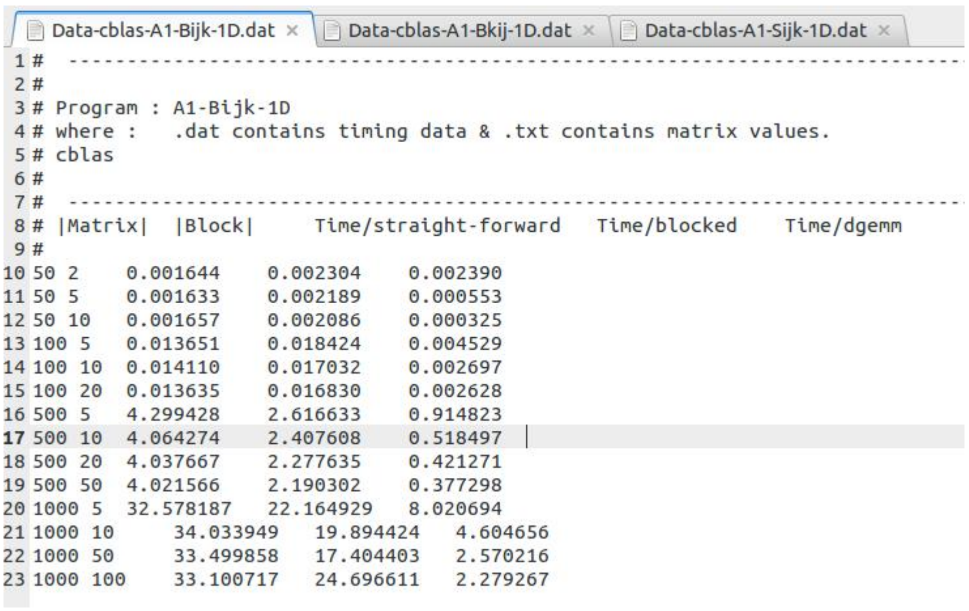

Data-cblas-A1-Bijk-1D.dat
Data-cblas-A1-Bkij-1D.dat
Data-cblas-A1-Sijk-1D.dat

```
#  ----------------------------------------------------------------------
#
# Program : A1-Sijk-1D
# where :   .dat contains timing data & .txt contains matrix values
# Data values from each run using different matrix and block size.
# cblas
#
#  ----------------------------------------------------------------------
#
# |Matrix|   Time/simple  Time/complex  Time/dgemm
#
50  0.001624     0.001175     0.000629
50  0.002513     0.001783     0.000485
50  0.002595     0.001772     0.000476
100     0.037520     0.023660     0.005968
100     0.040686     0.028691     0.005510
100     0.020369     0.014277     0.002023
500     5.967147     4.596160     1.198169
500     5.242453     4.369814     1.335228
500     5.340942     4.521537     1.392992
500     5.836312     4.638096     1.118236
1000    35.344029    20.260427    5.470701
1000    30.080705    24.735081    6.846879
1000    34.118114    23.406627    6.511904
1000    35.509945    22.845836    6.437362
```

Blocked IJK

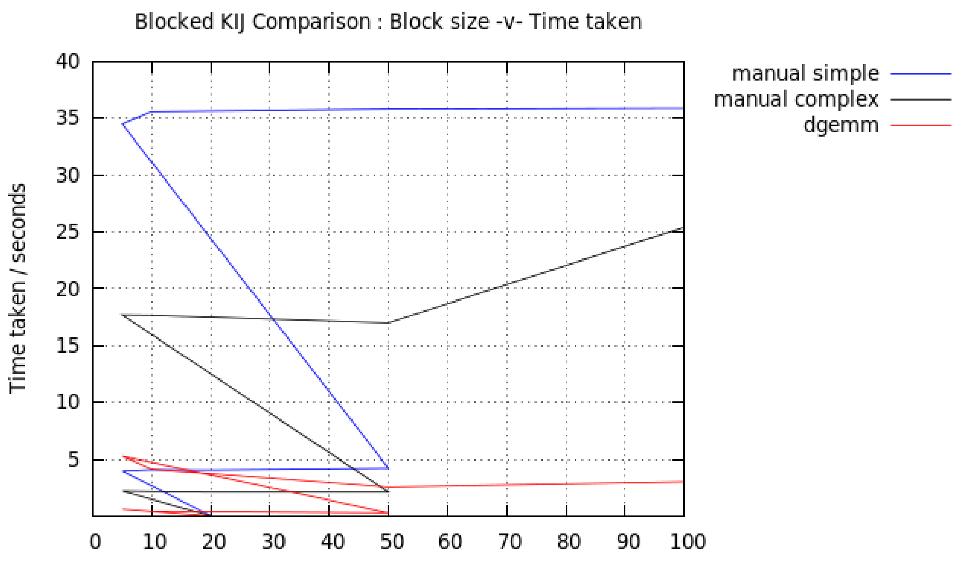


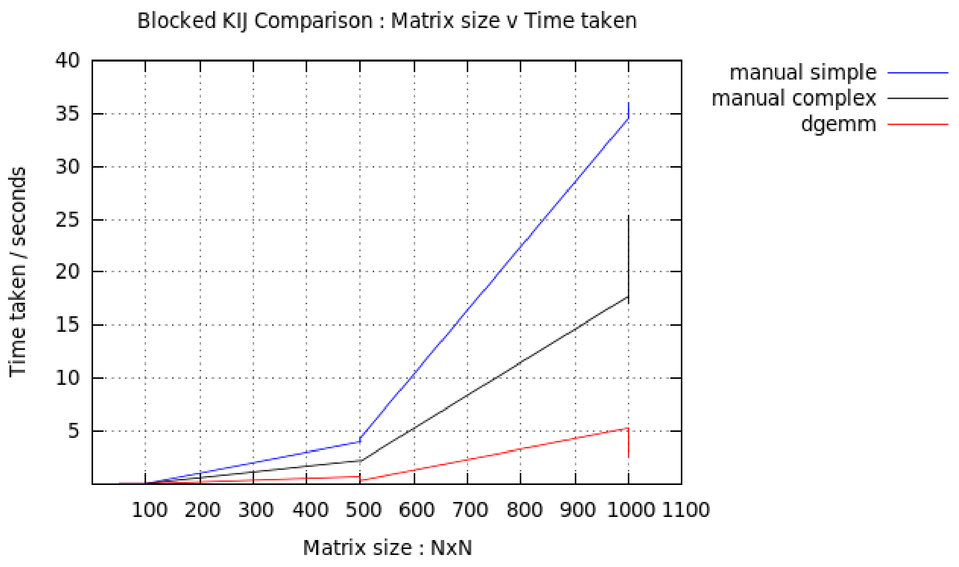


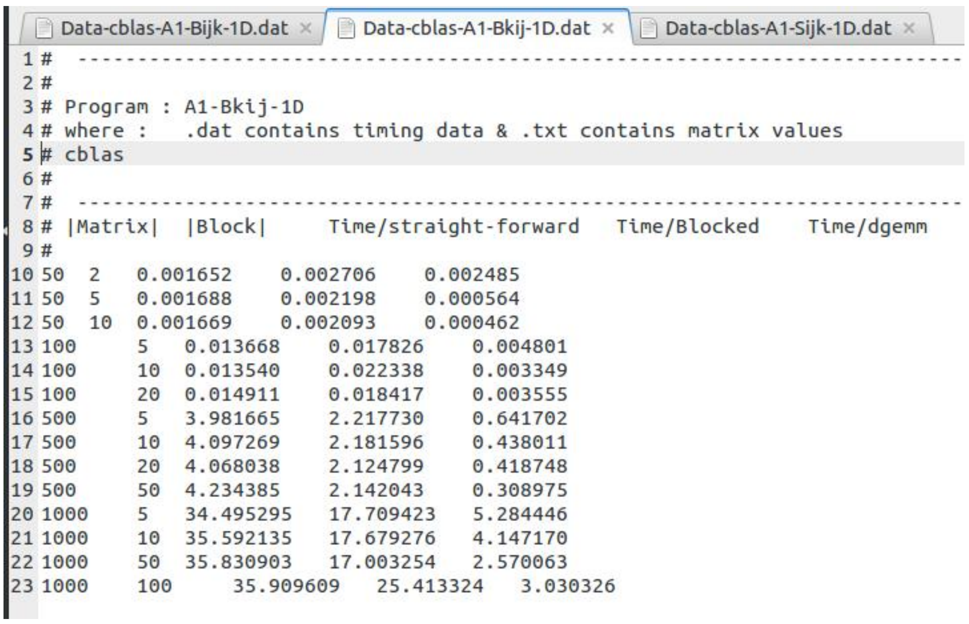

Data-cblas-A1-Bijk-1D.dat
Data-cblas-A1-Bkij-1D.dat
Data-cblas-A1-Sijk-1D.dat

```
# ---------------------------------------------------------------------
#
# Program : A1-Bijk-1D
# where :   .dat contains timing data & .txt contains matrix values.
# cblas
#
# ---------------------------------------------------------------------
# |Matrix|  |Block|     Time/straight-forward   Time/blocked    Time/dgemm
#
50 2    0.001644    0.002304    0.002390
50 5    0.001633    0.002189    0.000553
50 10   0.001657    0.002086    0.000325
100 5   0.013651    0.018424    0.004529
100 10  0.014110    0.017032    0.002697
100 20  0.013635    0.016830    0.002628
500 5   4.299428    2.616633    0.914823
500 10  4.064274    2.407608    0.518497
500 20  4.037667    2.277635    0.421271
500 50  4.021566    2.190302    0.377298
1000 5  32.578187   22.164929   8.020694
1000 10     34.033949   19.894424   4.604656
1000 50     33.499858   17.404403   2.570216
1000 100    33.100717   24.696611   2.279267
```

Blocked KIJ





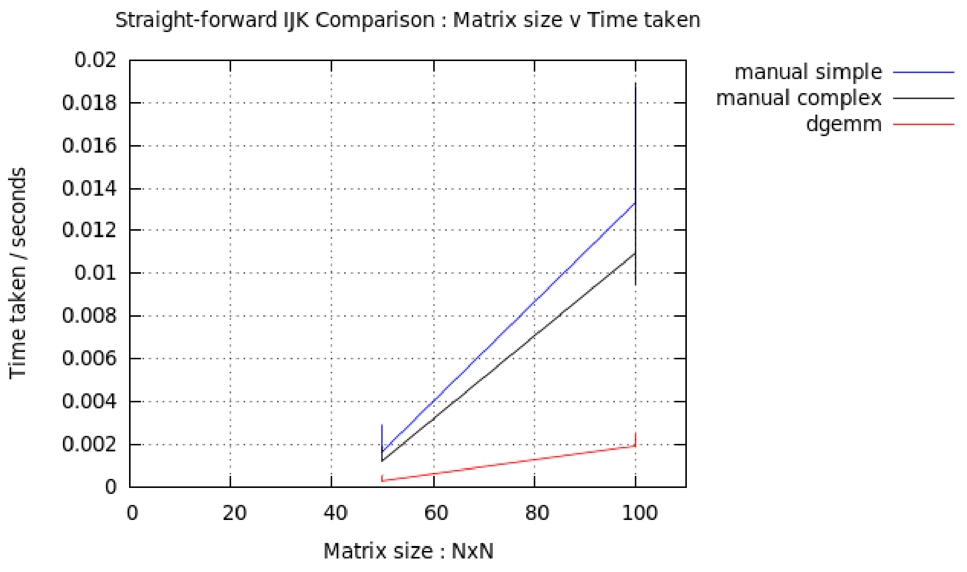

Data-cblas-A1-Bijk-1D.dat
Data-cblas-A1-Bkij-1D.dat
Data-cblas-A1-Sijk-1D.dat

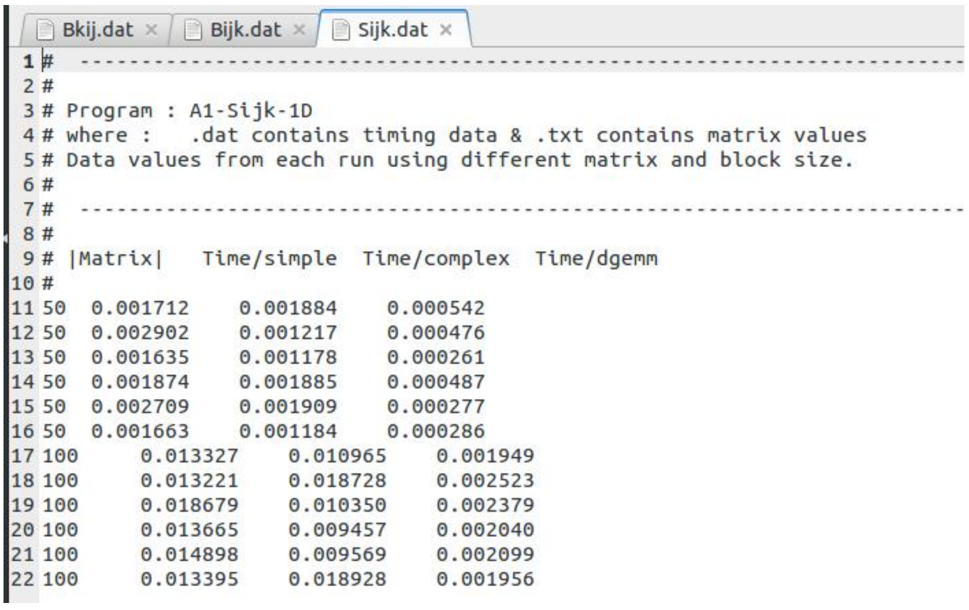
```
#  ------------------------------------------------------------------------
#
# Program : A1-Bkij-1D
# where :   .dat contains timing data & .txt contains matrix values
# cblas
#
#  ------------------------------------------------------------------------
# |Matrix|  |Block|     Time/straight-forward    Time/Blocked      Time/dgemm
#
50  2   0.001652     0.002706     0.002485
50  5   0.001688     0.002198     0.000564
50  10  0.001669     0.002093     0.000462
100     5   0.013668     0.017826     0.004801
100     10  0.013540     0.022338     0.003349
100     20  0.014911     0.018417     0.003555
500     5   3.981665     2.217730     0.641702
500     10  4.097269     2.181596     0.438011
500     20  4.068038     2.124799     0.418748
500     50  4.234385     2.142043     0.308975
1000    5   34.495295    17.709423    5.284446
1000    10  35.592135    17.679276    4.147170
1000    50  35.830903    17.003254    2.570063
1000    100     35.909609    25.413324    3.030326
```

GNUPLOT GRAPHS – CBLAS USING MATRIX SIZES (50 50 50 50 50 50 100 100 100 100 100 100)

$ ./RUNASSIGNMENT1.SH -A <-R|-I> -V -D2

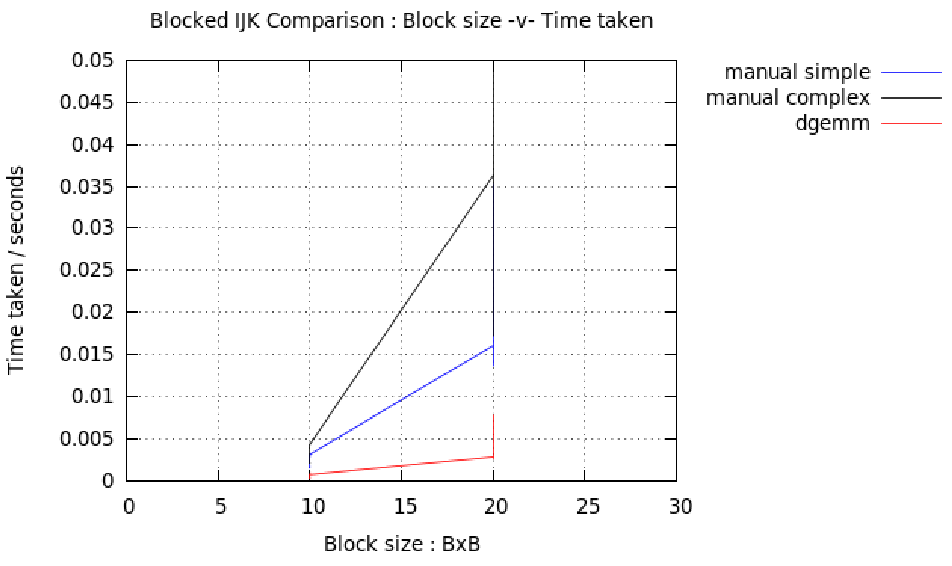
Simple IJK

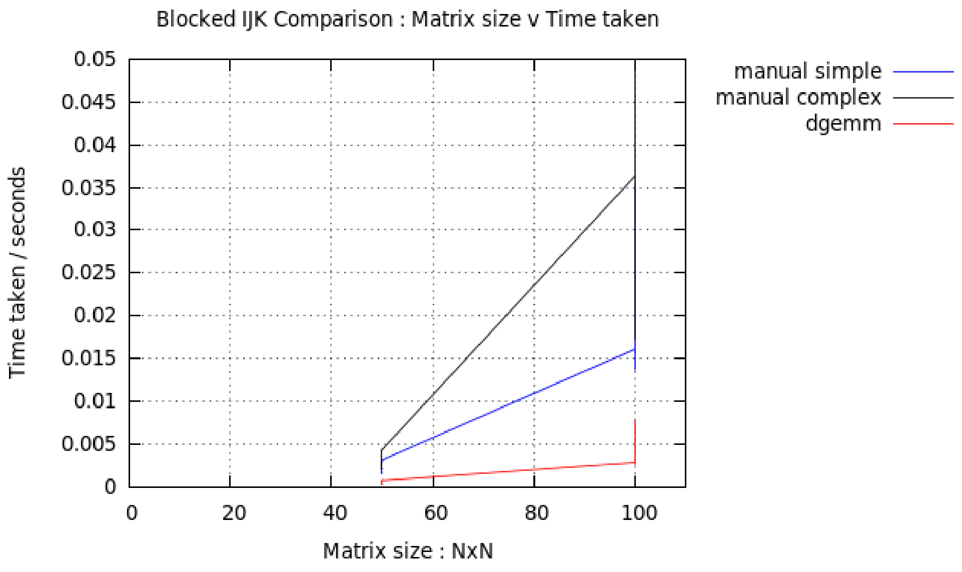


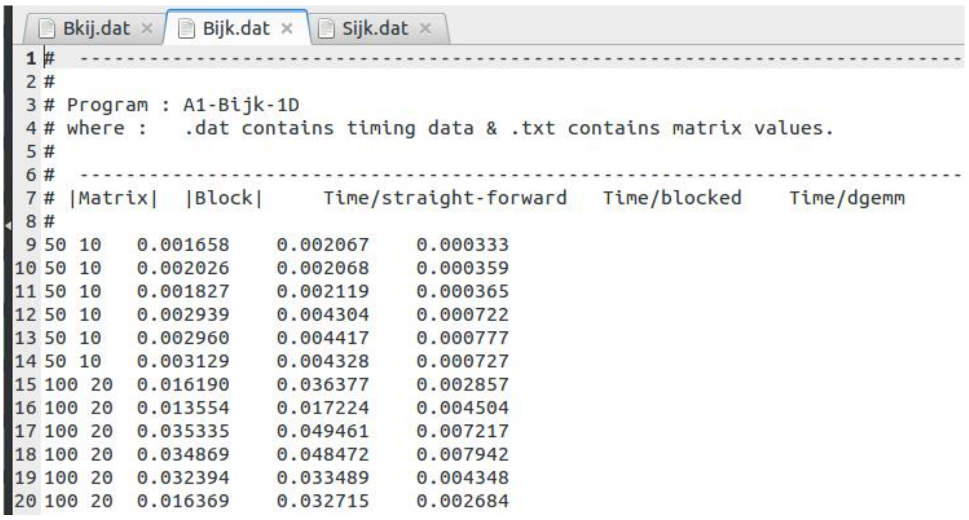


```
# ------------------------------------------------------------------------
#
# Program : A1-Sijk-1D
# where :    .dat contains timing data & .txt contains matrix values
# Data values from each run using different matrix and block size.
#
# ------------------------------------------------------------------------
#
# |Matrix|   Time/simple  Time/complex  Time/dgemm
#
50  0.001712    0.001884    0.000542
50  0.002902    0.001217    0.000476
50  0.001635    0.001178    0.000261
50  0.001874    0.001885    0.000487
50  0.002709    0.001909    0.000277
50  0.001663    0.001184    0.000286
100     0.013327    0.010965    0.001949
100     0.013221    0.018728    0.002523
100     0.018679    0.010350    0.002379
100     0.013665    0.009457    0.002040
100     0.014898    0.009569    0.002099
100     0.013395    0.018928    0.001956
```

Blocked IJK

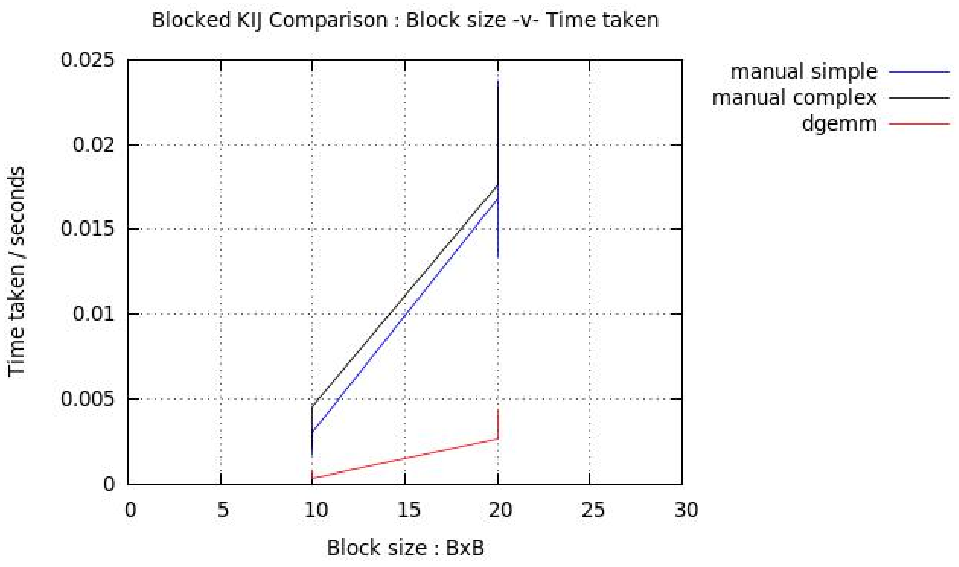


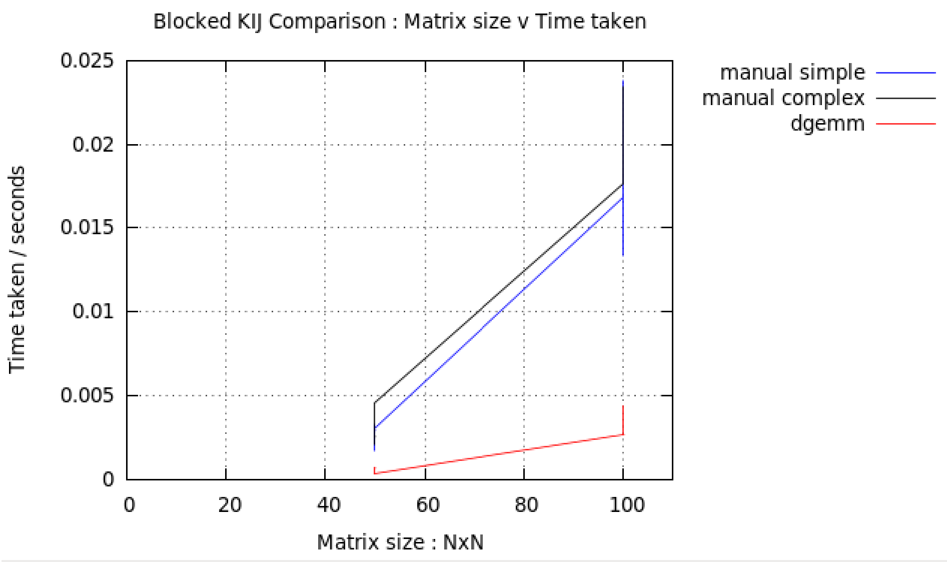


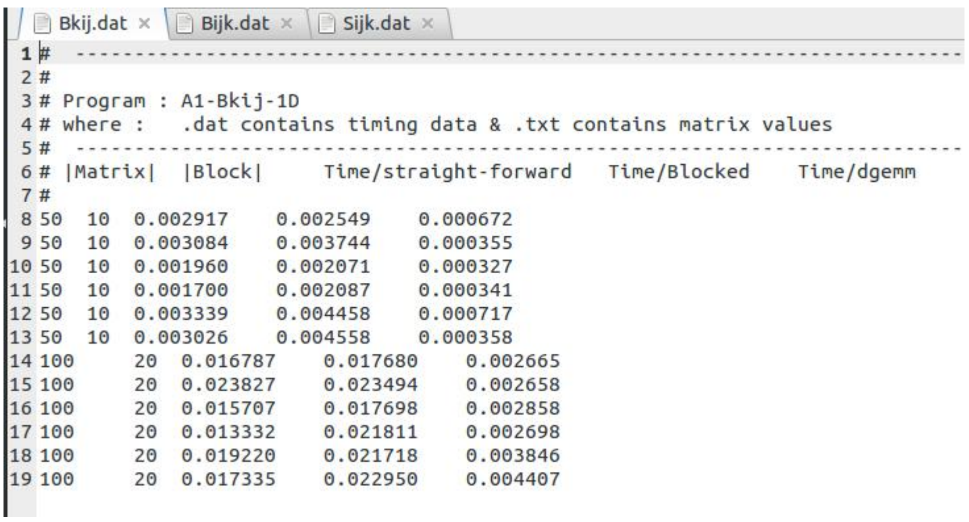


```
#  -----------------------------------------------------------------------------
#
# Program : A1-Bijk-1D
# where :   .dat contains timing data & .txt contains matrix values.
#
#  -----------------------------------------------------------------------------
# |Matrix|  |Block|     Time/straight-forward   Time/blocked    Time/dgemm
#
50  10    0.001658    0.002067    0.000333
50  10    0.002026    0.002068    0.000359
50  10    0.001827    0.002119    0.000365
50  10    0.002939    0.004304    0.000722
50  10    0.002960    0.004417    0.000777
50  10    0.003129    0.004328    0.000727
100  20   0.016190    0.036377    0.002857
100  20   0.013554    0.017224    0.004504
100  20   0.035335    0.049461    0.007217
100  20   0.034869    0.048472    0.007942
100  20   0.032394    0.033489    0.004348
100  20   0.016369    0.032715    0.002684
```

Blocked KIJ






Bkij.dat × Bijk.dat × Sijk.dat ×

```
# ---------------------------------------------------------------------------
#
# Program : A1-Bkij-1D
# where :   .dat contains timing data & .txt contains matrix values
# ---------------------------------------------------------------------------
# |Matrix|  |Block|     Time/straight-forward    Time/Blocked     Time/dgemm
#
50  10  0.002917    0.002549    0.000672
50  10  0.003084    0.003744    0.000355
50  10  0.001960    0.002071    0.000327
50  10  0.001700    0.002087    0.000341
50  10  0.003339    0.004458    0.000717
50  10  0.003026    0.004558    0.000358
100     20  0.016787    0.017680    0.002665
100     20  0.023827    0.023494    0.002658
100     20  0.015707    0.017698    0.002858
100     20  0.013332    0.021811    0.002698
100     20  0.019220    0.021718    0.003846
100     20  0.017335    0.022950    0.004407
```

Conclusions

Matrix multiplication is limited by memory speed and the size of the matrices being multiplied. When applying the three algorithms I used matrices |A| and |B| of varying sizes from [10x10] to [1000x1000]. The values in each matrix was Dependant on the switch -r (random from 1 to 10) and -r (iterative column + 1, so all cells in column 1000 had a value of 1001.

For the basic straight-forward algorithm of $|\mathbf{C}|_{[ij]} \mathrel{+}= |\mathbf{A}|_{[ik]} * |\mathbf{B}|_{[kj]}$ uses row major as follows (e.g.: [4x4] matrix:

0	*1*	*2*	*3*
4	*5*	*6*	*7*
8	*9*	*10*	*11*
12	*13*	*14*	*15*

Block matrix computation is usually faster than is straightforward computation. And as block size was increased for the same matrix size speed remained constant or did not increase significantly for manual computation – straight-forward. There was minimal improvements in blocked IJK and blocked KIJ computation. I did not notice any major difference in time taken for the later two even on large matrices. Atlas, however was significantly faster for the same calculations than was cblas. Also, while it took straight-forward IJK c. 35s when compiled using cblas to calculate |C| for a [1000 x 1000] matrix., the same computation took only c. 15s when compiled using atlas.

Also when block size is increased we should see a noticeable improvement in speed of calculation for atlas but not for manual calculations.

I calculated [100 x 100] matrices using the same block size to verify if a discrepancy occurred and indeed one does about 3 milliseconds in one instance for n=100 and b=20. This is not a great deal but it should be noted that due to other demands on the UCD servers some of these results are not 100% accurate as a standalone test for this computation alone.

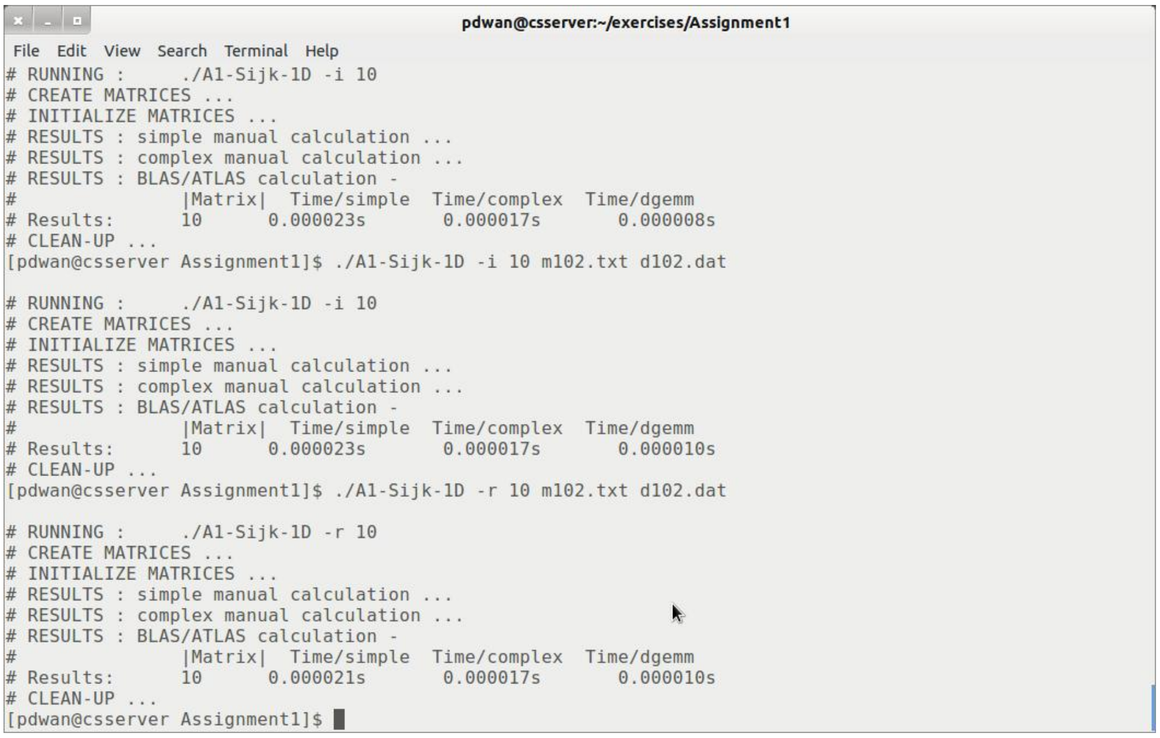
Finally for straight-forward IJK computation, I calculated using|C| = |A| * |B|, I also used a temporary variable to store the value of |C| (***sum***). This resulted in a small improvement but nothing hugely significant. The main improvement overall was when using atlas. As mentioned, cblas was better than manual and blocked were better than non-blocked.

APPENDICES

APPENDIX I – VALIDATE RESULTS

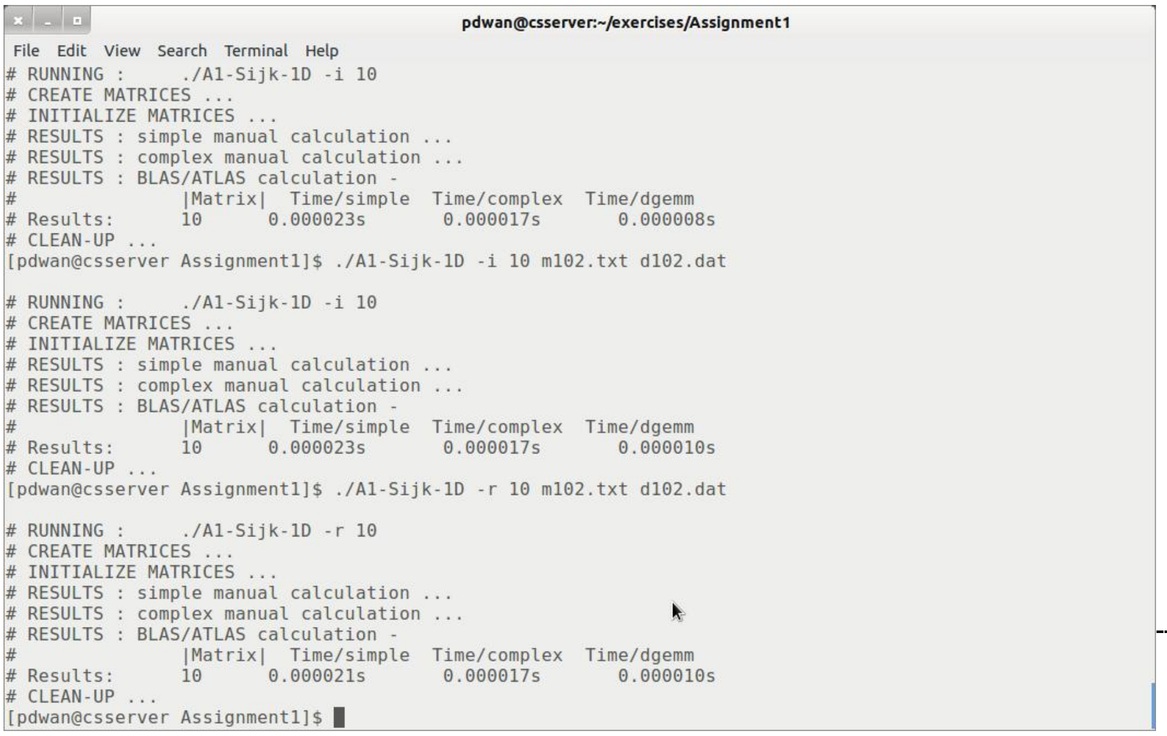
Spot check only using 10x10 matrices, initializing matrices |A| and |B| using successive column values.

Build using :



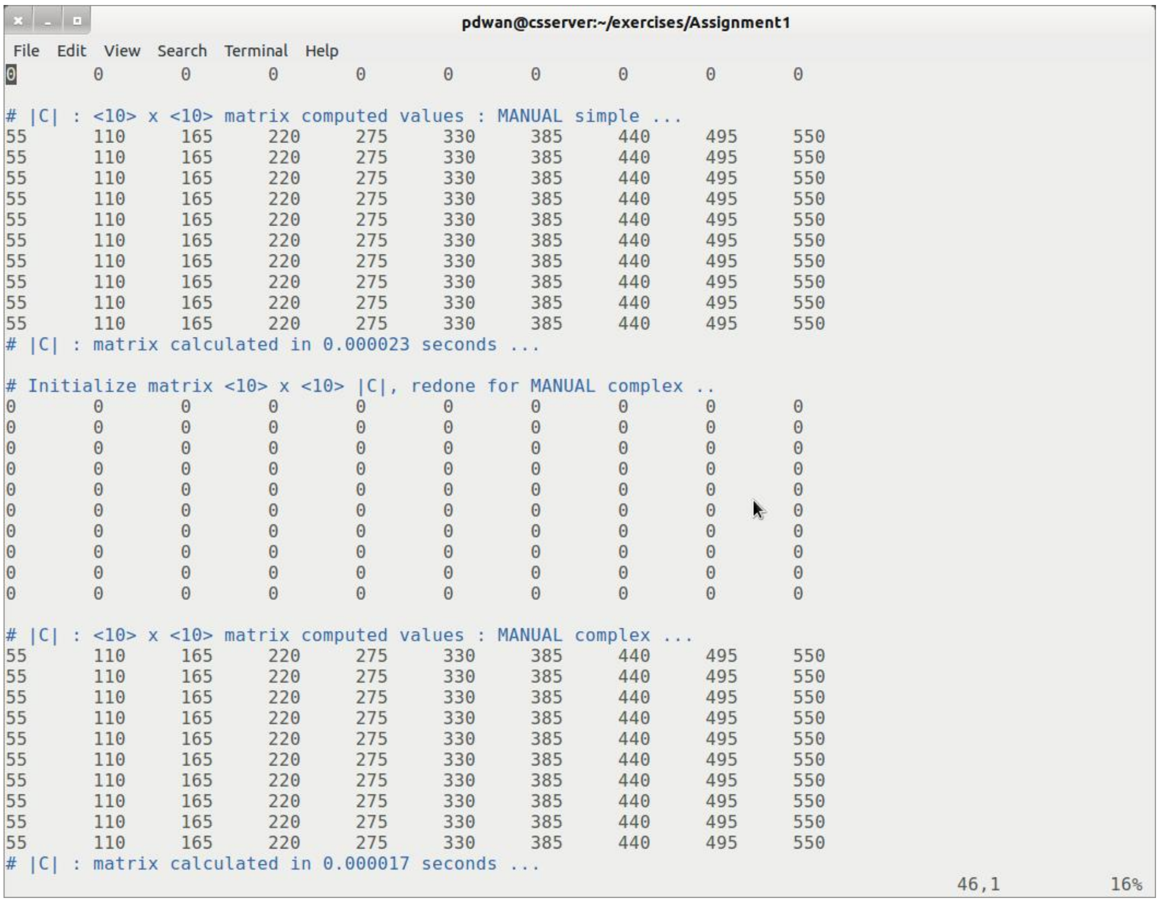
```
# RUNNING :     ./A1-Sijk-1D -i 10
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual calculation ...
# RESULTS : complex manual calculation ...
# RESULTS : BLAS/ATLAS calculation -
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      10      0.000023s       0.000017s       0.000008s
# CLEAN-UP ...
[pdwan@csserver Assignment1]$ ./A1-Sijk-1D -i 10 m102.txt d102.dat

# RUNNING :     ./A1-Sijk-1D -i 10
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual calculation ...
# RESULTS : complex manual calculation ...
# RESULTS : BLAS/ATLAS calculation -
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      10      0.000023s       0.000017s       0.000010s
# CLEAN-UP ...
[pdwan@csserver Assignment1]$ ./A1-Sijk-1D -r 10 m102.txt d102.dat

# RUNNING :     ./A1-Sijk-1D -r 10
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual calculation ...
# RESULTS : complex manual calculation ...
# RESULTS : BLAS/ATLAS calculation -
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      10      0.000021s       0.000017s       0.000010s
# CLEAN-UP ...
[pdwan@csserver Assignment1]$
```

Resulting sample Matrix .txt file contains :

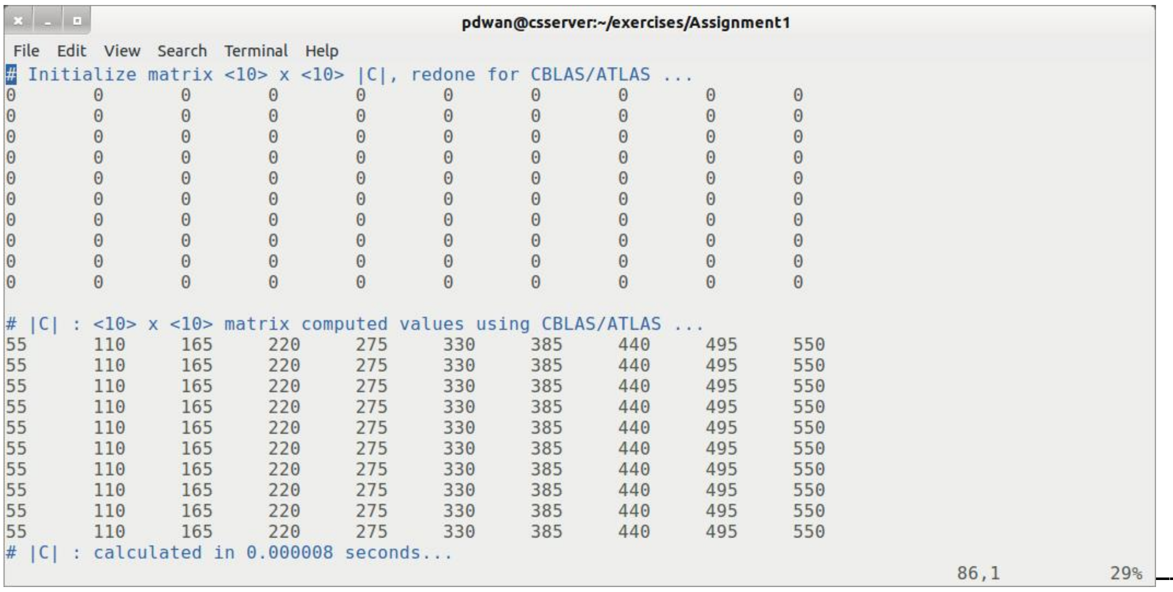


```
# RUNNING :     ./A1-Sijk-1D -i 10
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual calculation ...
# RESULTS : complex manual calculation ...
# RESULTS : BLAS/ATLAS calculation -
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      10      0.000023s       0.000017s       0.000008s
# CLEAN-UP ...
[pdwan@csserver Assignment1]$ ./A1-Sijk-1D -i 10 m102.txt d102.dat

# RUNNING :     ./A1-Sijk-1D -i 10
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual calculation ...
# RESULTS : complex manual calculation ...
# RESULTS : BLAS/ATLAS calculation -
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      10      0.000023s       0.000017s       0.000010s
# CLEAN-UP ...
[pdwan@csserver Assignment1]$ ./A1-Sijk-1D -r 10 m102.txt d102.dat

# RUNNING :     ./A1-Sijk-1D -r 10
# CREATE MATRICES ...
# INITIALIZE MATRICES ...
# RESULTS : simple manual calculation ...
# RESULTS : complex manual calculation ...
# RESULTS : BLAS/ATLAS calculation -
#               |Matrix|  Time/simple  Time/complex  Time/dgemm
# Results:      10      0.000021s       0.000017s       0.000010s
# CLEAN-UP ...
[pdwan@csserver Assignment1]$
```

--

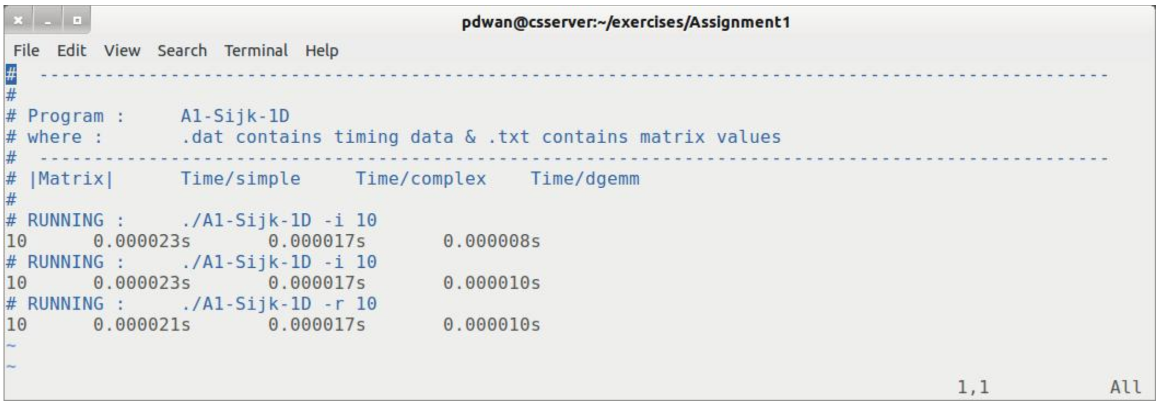

pdwan@csserver:~/exercises/Assignment1
File Edit View Search Terminal Help

```
0       0       0       0       0       0       0       0       0       0

# |C| : <10> x <10> matrix computed values : MANUAL simple ...
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
# |C| : matrix calculated in 0.000023 seconds ...

# Initialize matrix <10> x <10> |C|, redone for MANUAL complex ..
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0

# |C| : <10> x <10> matrix computed values : MANUAL complex ...
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
# |C| : matrix calculated in 0.000017 seconds ...
                                                                          46,1          16%
```


pdwan@csserver:~/exercises/Assignment1
File Edit View Search Terminal Help

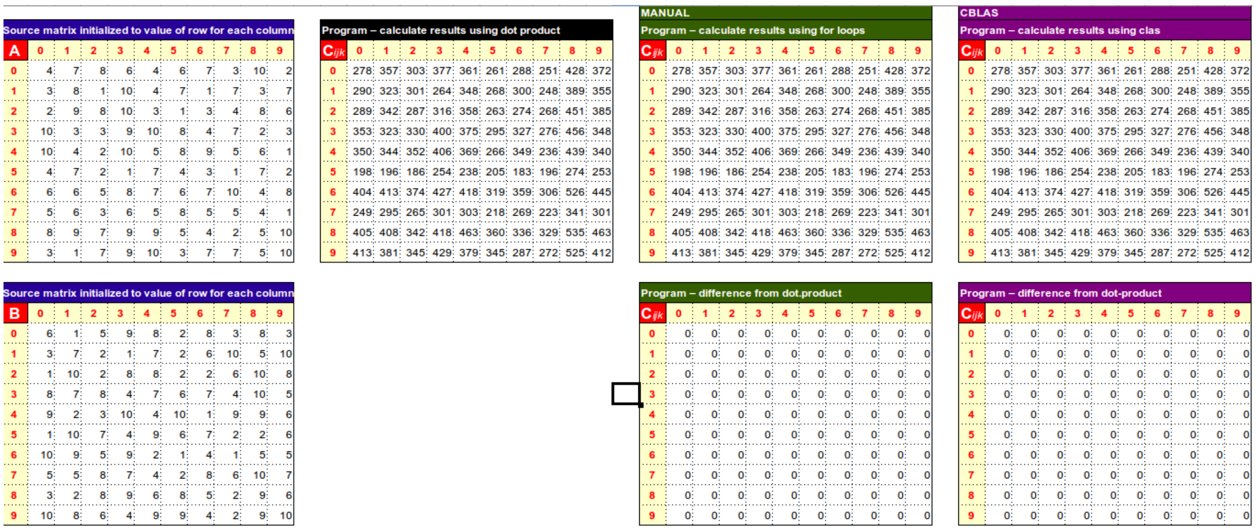
```
# Initialize matrix <10> x <10> |C|, redone for CBLAS/ATLAS ...
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0

# |C| : <10> x <10> matrix computed values using CBLAS/ATLAS ...
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
55      110     165     220     275     330     385     440     495     550
# |C| : calculated in 0.000008 seconds...
                                                                          86,1          29%
```

Resulting sample summary timing data file contains :



```
pdwan@csserver:~/exercises/Assignment1
File  Edit  View  Search  Terminal  Help
#  -------------------------------------------------------------------------------------------------
#
# Program :     A1-Sijk-1D
# where :       .dat contains timing data & .txt contains matrix values
#  -------------------------------------------------------------------------------------------------
# |Matrix|      Time/simple     Time/complex    Time/dgemm
#
# RUNNING :     ./A1-Sijk-1D -i 10
10      0.000023s       0.000017s       0.000008s
# RUNNING :     ./A1-Sijk-1D -i 10
10      0.000023s       0.000017s       0.000010s
# RUNNING :     ./A1-Sijk-1D -r 10
10      0.000021s       0.000017s       0.000010s
~
~
                                                                        1,1           All
```

Validating results gives :



Source matrix initialized to value of row for each column

A	0	1	2	3	4	5	6	7	8	9
0	4	7	8	6	4	6	7	3	10	2
1	3	8	1	10	4	7	1	7	3	7
2	2	9	8	10	3	1	3	4	8	6
3	10	3	3	9	10	8	4	7	2	3
4	10	4	2	10	5	8	9	5	6	1
5	4	7	2	1	7	4	3	1	7	2
6	6	6	5	8	7	6	7	10	4	8
7	5	6	3	6	5	8	5	5	4	1
8	8	9	7	9	9	5	4	2	5	10
9	3	1	7	9	10	3	7	7	5	10

Program – calculate results using dot product

C_{ijk}	0	1	2	3	4	5	6	7	8	9
0	278	357	303	377	361	261	288	251	428	372
1	290	323	301	264	348	268	300	248	389	355
2	289	342	287	316	358	263	274	268	451	385
3	353	323	330	400	375	295	327	276	456	348
4	350	344	352	406	369	266	349	236	439	340
5	198	196	186	254	238	205	183	196	274	253
6	404	413	374	427	418	319	359	306	526	445
7	249	295	265	301	303	218	269	223	341	301
8	405	408	342	418	463	360	336	329	535	463
9	413	381	345	429	379	345	287	272	525	412

MANUAL

Program – calculate results using for loops

C_{ijk}	0	1	2	3	4	5	6	7	8	9
0	278	357	303	377	361	261	288	251	428	372
1	290	323	301	264	348	268	300	248	389	355
2	289	342	287	316	358	263	274	268	451	385
3	353	323	330	400	375	295	327	276	456	348
4	350	344	352	406	369	266	349	236	439	340
5	198	196	186	254	238	205	183	196	274	253
6	404	413	374	427	418	319	359	306	526	445
7	249	295	265	301	303	218	269	223	341	301
8	405	408	342	418	463	360	336	329	535	463
9	413	381	345	429	379	345	287	272	525	412

CBLAS

Program – calculate results using clas

C_{ijk}	0	1	2	3	4	5	6	7	8	9
0	278	357	303	377	361	261	288	251	428	372
1	290	323	301	264	348	268	300	248	389	355
2	289	342	287	316	358	263	274	268	451	385
3	353	323	330	400	375	295	327	276	456	348
4	350	344	352	406	369	266	349	236	439	340
5	198	196	186	254	238	205	183	196	274	253
6	404	413	374	427	418	319	359	306	526	445
7	249	295	265	301	303	218	269	223	341	301
8	405	408	342	418	463	360	336	329	535	463
9	413	381	345	429	379	345	287	272	525	412

Source matrix initialized to value of row for each column

B	0	1	2	3	4	5	6	7	8	9
0	6	1	5	9	8	2	8	3	8	3
1	3	7	2	1	7	2	6	10	5	10
2	1	10	2	8	8	2	2	6	10	8
3	8	7	8	4	7	6	7	4	10	5
4	9	2	3	10	4	10	1	9	9	6
5	1	10	7	4	9	6	7	2	2	6
6	10	9	5	9	2	1	4	1	5	5
7	5	5	8	7	4	2	8	6	10	7
8	3	2	8	9	6	8	5	2	9	6
9	10	8	6	4	9	9	4	2	9	10

Program – difference from dot.product

C_{ijk}	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

Program – difference from dot-product

C_{ijk}	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

Appendix II – Summary of time taken using predefined values

# [pdwan@csserver Assignment1]$ ./runAssignment1.sh -a -i -v

# RUNNING :   ./A1-Sijk-1D-cblas -i

|Matrix|	Time / simple	Time / complex	Time / dgemm
50	0.002397	0.001817	0.000563
50	0.001835	0.001610	0.000271
50	0.002929	0.001241	0.000525
100	0.017382	0.011552	0.001930
100	0.013228	0.018884	0.001930
100	0.017741	0.011016	0.002145
500	3.992115	3.373840	0.816194
500	4.042341	3.321079	0.813839
500	4.060413	3.506160	0.808669
500	3.935975	3.414704	0.814437
1000	32.106745	21.988344	6.468189
1000	32.680508	22.111431	6.142010
1000	32.168037	22.785491	6.010852
1000	32.829488	21.783892	6.466903

# RUNNING : ./A1-Bijk-1D-cblas -i ....

|Matrix|	|Block|	Time / straight-forward	Time / blocked	Time / dgemm
50	2	0.001728	0.002292	0.002351
50	5	0.001742	0.002156	0.000575
50	10	0.001969	0.002344	0.000331
100	5	0.013452	0.017792	0.004812
100	10	0.013417	0.017832	0.002805
100	20	0.013346	0.017548	0.002674
500	5	4.093077	2.589791	1.108811
500	10	3.975883	2.311699	0.460090
500	20	4.061668	2.230458	0.423325
500	50	4.103544	2.194383	0.314732
1000	5	32.692186	21.335487	7.811334
1000	10	32.900493	19.119611	4.422699
1000	50	32.806413	17.282943	2.553168
1000	100	32.740458	24.899779	2.610839

# RUNNING : ./A1-Bkij-1D-cblas -i ....

|Matrix|	|Block|	Time / straight-forward	Time / blocked	Time / dgemm
50	2	0.001714	0.002352	0.002412
50	5	0.001696	0.002159	0.000556
50	10	0.001726	0.002065	0.000365
100	5	0.013371	0.017705	0.004513
100	10	0.013345	0.017616	0.003136
100	20	0.013503	0.017083	0.003213
500	5	4.088723	2.174808	0.658462
500	10	4.101730	2.140955	0.446238
500	20	4.105657	2.124068	0.419177
500	50	4.023817	2.105757	0.300349
1000	5	33.503170	17.998477	5.296227
1000	10	32.901040	17.215492	3.753910
1000	50	32.942782	17.135165	2.513635
1000	100	33.385318	25.788746	2.967292

# [pdwan@csserver Assignment1]$ ./runAssignment1.sh -a -r -v

#RUNNING : ./A1-Sijk-1D-cblas -r -v -a .....

|Matrix|	Time / simple	Time / complex	Time / dgemm
50	0.001758	0.004282	0.000277
50	0.001680	0.001182	0.000260
50	0.001691	0.001271	0.000304
50	0.001684	0.001194	0.000267
50	0.001657	0.001204	0.000264
50	0.001673	0.001176	0.000268
100	0.013685	0.009317	0.001959
100	0.013239	0.009549	0.001946
100	0.013343	0.009363	0.001955
100	0.014888	0.010414	0.002103
100	0.014103	0.009863	0.002158
100	0.014835	0.010157	0.002142

RUNNING : ./A1-Bijk-1D-cblas -r -v -a .....

|Matrix|	|Block|	Time / straight-forward	Time / blocked	Time / dgemm
50	10	0.001918	0.002379	0.000421
50	10	0.001812	0.002362	0.000409
50	10	0.001926	0.002369	0.000473
50	10	0.002199	0.002324	0.000423
50	10	0.001887	0.002348	0.000376
50	10	0.001910	0.002284	0.000405
100	20	0.014558	0.018774	0.003174
100	20	0.014488	0.018067	0.002812
100	20	0.019229	0.021085	0.002695
100	20	0.014279	0.016852	0.002676
100	20	0.013526	0.016964	0.002613
100	20	0.013715	0.017646	0.002662

RUNNING : ./A1-Bkij-1D-cblas -r -v -a ....

|Matrix|	|Block|	Time / straight-forward	Time / blocked	Time / dgemm
50	10	0.001715	0.002055	0.000351
50	10	0.001714	0.002082	0.000350
50	10	0.001823	0.002062	0.000346
50	10	0.001826	0.002070	0.000322
50	10	0.001770	0.002131	0.000335
50	10	0.001702	0.002069	0.000324
100	20	0.013753	0.017111	0.002642
100	20	0.013370	0.017060	0.002695
100	20	0.013380	0.017424	0.002672
100	20	0.013362	0.017756	0.002689
100	20	0.014066	0.017370	0.002714
100	20	0.013301	0.017247	0.002862

Appendix III – References / Acknowledgements

www.stackoverflow.com

www.cs.indiana.edu