Implementing Symmetric Matrix-Vector Multiply with FLAMEC

September 27, 2016

Abstract

This document walks you through how to translate the FLaTeX algorithms for symmetric matrix-vector multiplication (Symv) into C using the FLAMEC API.

1 Step 1.

On your favorite CS machines, perform

```
cp -r ~rvdg/class/CS378F16/CS378_PfCandP/Assignments/Week2/C/  <your favorite directory>/C/
```

(Alternatively, do a ”pull” from github.)

2 Step 2

In the directory you just copied, you will find a number of files:

`Makefile`
The Makefile that lets you compile and link.

`driver.c`
The ”driver” program that tests different implementations.

`SymMatVec[1-5].c`
in which you will program ”algorithmic variants” 1, 4, and 5.

`SymMatVecPlot.m`
which helps you plot timing data.

To compile and link all the routines with the driver program, type

```
% make
```

To then execute the program, type

```
% make test
```

To understand what is going on, you may want to sneak a peek at the file `driver.c`.

3 Step 3: Plot the performance data

Start up Matlab in the background

```
% matlab &
```

and execute SymMatVecPlot in the command window. This will pop up a window with a graph that shows the time required to perform the symmetric matrix-vector multiplication as a function of the matrix size n.

If you look at the files `SymmMatVec1.c`, `SymmMatVec4.c`, and `SymmMatVec5.c`, you will notice that all three implement "Variant 1".

4 Step 4: Implement Variants 4 and 5

Your assignment, if you choose to accept it, is to now implement Variants 4 and 5, replacing the code in `SymMatVec4.c` and `SymMatVec5.c`.

5 Translating Math to FLAMEC Calls

There are three routines that you may want to employ to implement your routines:

- `FLA_Axpy( alpha, x, y )` computes $y := \alpha x + y$, where x and y can be column and/or row vectors.
- `FLA_Dot( x, y, alpha )` computes $\alpha := x^T y$, where x and y can be column and/or row vectors.
- `FLA_Dots( alpha, x, y, beta, gamma )` computes $\gamma := \alpha x^T y + \beta\gamma$, where x and y can be column and/or row vectors.

6 Useful links

If you want to implement the `libflame` and BLIS libraries to which this assignment links (e.g., on your own computing), you can get these from

- https://github.com/flame/blis.
- https://github.com/flame/libflame.
- Field Van Zee. `libflame` Users' Guide. (Please do not print.)