1.5.1 Starting the Package

Bookmark this page

1.5.1 Starting the Package

Reading Assignment

0 points possible (ungraded)
Read Unit 1.5.1 of the notes. [LINK]

☑ Done

Submit

Correct

1.5.1 Discussion

Topic: Week 1 / 1.5.1

Hide Discussion

Add a Post

Show all posts by recent activity

?	So we are coding in MATLAB but using the format resembling C and Fortran? Are we using MATLAB as a platform and using routines coded in a way resembling C and Fortran? My guess is that C and Fortran use less comp...	3
?	matlab i am currently involved with a multilayered data acquisition and analysis project involving python, its pipelines, jquery and javascript. I DON'T HA...	6
	Typo graceously → graciously just mentioning	2
?	Late Homework My internet is down for my computer and I don't have a ride to go somewhere that has free WiFi. Trying to do the homework on my Smartphone ...	2

In this course, we will explore and use a rudimentary dense linear algebra software library. The hope is that by linking the abstractions in linear algebra to abstractions (functions) in software, a deeper understanding of the material will be the result.

We will be using the MATLAB interactive environment by MathWorks for our exercises. MATLAB is a high-level language and interactive environment that started as a simple interactive “laboratory" for experimenting with linear algebra. It has since grown into a powerful tool for technical computing that is widely used in academia and industry.

For our Fall 2017 offering of LAFF on the edX platform, MathWorks has again graceously made temporary licenses available for the participants. Instructions on how to install and use MATLAB can be found in Week 0 (Section 0.3).

The way we code can be easily translated into other languages. For example, as part of our FLAME research project we developed a library called `libflame`. Even though we coded it in the C programming language, it still closely resembles the MATLAB code that you will write and the library that you will use.

A library of vector-vector routines

The functionality of the functions that you will write is also part of the "laff" library of routines. What this means will become obvious in subsequent units.

Below is a table of vector functions, and the routines that implement them, that you will be able to use in future more complete list of routines is given in the "laff routines" tab.

Calculator

Operation Abbrev.	Definition	Function	MATLAB	Approx. cost	
-	-	-	intrinsic	flops	memops
Vector-vector operations					
Copy (*copy*)	$y := x$	`y = laff_copy( x, y )`	`y = x`	0	$2n$
Vector scaling (*scal*)	$x := \alpha x$	`x = laff_scal( alpha, x )`	`x = alpha * x`	n	$2n$
Scaled addition (*axpy*)	$y := \alpha x + y$	`y = laff_axpy( alpha, x, y )`	`y = alpha x + y`	$2n$	$3n$
Dot product (*dot*)	$\alpha := x^T y$	`alpha = laff_dot( x, y )`	`alpha = x' * y`	$2n$	$2n$
Length (*norm2*)	$\alpha := \|x\|_2$	`alpha = laff_norm2( x )`	`alpha = norm2( x )`	$2n$	n

A couple of comments:

- The operations we will implement are available already in MATLAB. So why do we write them as routines? Because
  1. It helps us connect the abstractions in the mathematics to the abstractions in code; and
  2. Implementations in other languages (e.g. C and Fortran) more closely follow how we will implement the operations as functions/routines.
- In, for example, `laff_copy`, why not make the function

  `y = laff_copy( x )`?

  1. Often we will want to copy a column vector to a row vector or a row vector to a column vector. By also passing `y` into the routine, we indicate whether the output should be a row or a column vector.
  2. Implementations in other languages (e.g. C and Fortran) more closely follow how we will implement the operations as functions/routines.

The way we will program translates almost directly into equivalent routines for the C or Python programming languages.

Now, let's dive right in! We'll walk you through it in the next units.

< Previous   Next >

Calculator