

Re: What is the point of octave? (And its presumed buddy, MatLab)

Re: What is the point of octave? (And its presumed buddy, MatLab)

Source: <http://sci.tech-archive.net/Archive/sci.math.symbolic/2005-06/msg00056.html>

- *From:* Jerzy Karczmarczuk <karczma@xxxxxxxxxxxxxxxx>
 - *Date:* Tue, 21 Jun 2005 10:06:32 +0200
-

djlogan2 wrote:

I installed octave, since I saw mention of it here and there, and it's purported similarities to MatLab.

Well, now that I have it, my fundamental question is: What do I do with it?

Consider that I am using Maxima extensively now, and have used Mathematica extensively in the past. They are obviously largely symbolic algebra systems. I can do differentiation, integration, various matrix equations, oodles and oodles of functions (such as "laplace()"), systems of equations, etc. etc. etc. I'm sure everybody here knows that.

But reviewing the Octave primer and Octave reference manual, I haven't really figured out its overall purpose. It sure doesn't seem to be symbolic at all. Is it numerical only? So equations would have to be "solved" in something like Maxima first?

Is it primarily for interfacing from programs I write? What problem exactly is it supposed to solve?

Julian Stoev gave you an answer:

Instead of solving these problems in compiled language (like Fortran), Matlab is solving this in the interpreted language, thus gaining programming speed and convenience.

Re: What is the point of octave? (And its presumed buddy, MatLab)

So, yes, it is MAINLY for numerical computations. Still, I have a few additional comments. Sorry, it will be long!

0. Matlab, Scilab, Octave, Numerical/Scientific Python, Yorick, Rlab, etc. are based on **vectorial** approach to programming. Instead of writing loops, you define and process arrays as compact entities with mathematical properties, say
x = (0:1000)*0.001*Pi; y=sin(x); plot(x,y,x,x.*y);
THIS is the main point, not the fact that those packages are interpreters. So, the typical usage concerns scientific problems where there is plenty of regular accesses to structured data. Signal processing, computer graphics, etc.
[Actually, I do not understand people who install programs before knowing what they really want from them, but this is another story.]
1. There is plenty of "semi-numerical" problems around, as Knuth would name them. Often you don't need "symbols" at all, but mathematical