A small numerical analysis library in Common Lisp

Gene Michael Stover

created Saturday, 2007 January 6
updated Monday, 2007 January 22

Contents

1 What is this?

fixme: As of 2007 January 6, this is unfinished. In fact, it's just started. It's an experimental work in progress.

This is a small library to approximate perform some "numerical analysis" calculations, which include:

- interpolating polynomials over one variable &
- approximating numerical solutions to linear systems.

This library is small & provides a few specific features which I needed for a special case[1]. It is probably most useful if you have an application in which you need only a few types of numerical analysis operations & your data sets aren't

[1] The special case happened to be graphically rendering virtual worlds.

large. For more demanding numerical analysis needs, look at the "mathematics" entry at CL^{iki} the common lisp wiki.

2 License

The library is licensed according to the Gnu Lesser General Public License. [2]
The text of that agreement is at `http://www.gnu.org/licenses/licenses.html#LGPL`.

This documentation (which are are reading now) is not covered by an open source license. See the copyright notice at the top of this document.

3 Examples of use

Most (all?) of the algorithms in the library are from *Numerical Analysis* by Burden & Faires. [1]

4 Obtaining

- matrix.lisp
- numerical-analysis.lisp
- loadall.lisp
- my Lisp unit test framework [3]

5 How to use it

1. Download the source file from `http://cybertiggyr.com/gene/big/bigo.lisp`. (Also, that file is in Appendix **??**.)
2. You might want `loadall.lisp`.[2]
3. Youw will need `test.lisp` from My Lisp Unit Test Framework [3].
4. Load `test.lisp` into your Lisp. Then load `bigo.lisp` into your Lisp. A file called `loadall.lisp`, in Appendix **??**, is an example of this.
5. Obtain your performance measurements. They must be in the form of a list of two-list elements. Each two-list element consists of a FIRST which is the size of the input & a SECOND which is the cost. The cost can be number of times some function was called, the amount of wall-time required for your function to run, the amount of memory it used, or whatever else you want to measure. Let's assume that you bound your measurements to a global variable called *MEASUREMENT*.

[2] Yep, it would be nice if I made an ASDF file for it.

6. `(in-package "COM.CYBERTIGGYR.BIGO")`

7. `(bigo *measurement*)`

8. The string you get in return will be he Big-O cost for the function you measured. The string is formatted for LaTeX.

A Other File Formats

- This document is available in multi-file HTML format at http://cybertiggyr.com/gene/linear/.
- This document is available in Pointless Document Format (PDF) at http://cybertiggyr.com/gene/linear/linear.pdf.

References

[1] Richard L. Burden J. Douglass Faires. *Numerical Analysis*. PWS-KENT Publishing Company, fourth edition, 1989. ISBN 0-53491-585-X.

[2] Free Software Foundation. Lesser general public license. world wide web. http://www.gnu.org/licenses/licenses.html#LGPL.

[3] Gene Michael Stover. My lisp unit test framework. *cybertiggyr.com/gene*, June 2005. http://cybertiggyr.com/gene/lut/.