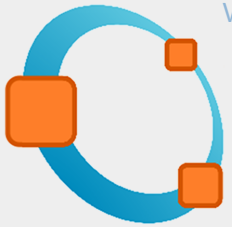
GNU Octave
A free high-level tool for Scientific Computing

Carlo de Falco, Jordi Gutiérrez Hermoso

November 8, 2012



Outline

Outline

Outline

What is Octave?

Octave

"A free [a] numerical environment mostly compatible with MATLAB"

- What is compatibility? A point of much debate...
- If it works in MATLAB, it should work in Octave.
- If it breaks it is considered a bug.
- If it works in Octave, it can break in MATLAB.

[a] "free" = "libero" ≠ "gratis"

Lines of code

The stuff Octave is made of...

Lines of code

The stuff Octave is made of...

Core

- About 600,000 lines of C++
- About 100,000 lines of m-scripts
- About 50,000 lines of Fortran

Lines of code

The stuff Octave is made of...

Core

- About 600,000 lines of C++
- About 100,000 lines of m-scripts
- About 50,000 lines of Fortran

Octave-Forge

- About 200,000 lines of C++
- About 330,000 lines of m-scripts
- About 50,000 lines of Fortran

Features

Current features

- N-d arrays, linear algebra, sparse matrices
- Nonlinear equations
- Ordinary/Algebraic Differential Equations,
- Image processing, statistics, special functions
- Many more...

Features in development

- GUI
- JIT compiling
- classdef OOP

What does it look like

- Primarily a CLI interface

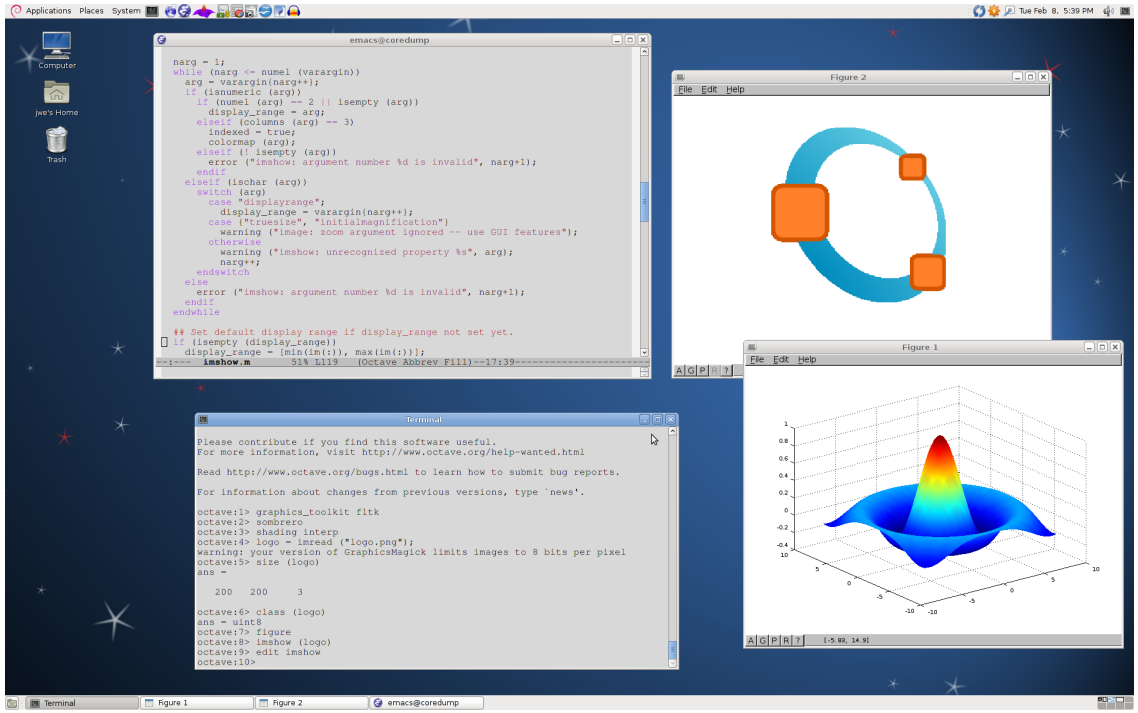


Figure: CLI screenshot

What does it look like

- Primarily a CLI interface
- Most requested feature: GUI! Will ship with next release (4.0)

What does it look like

- Primarily a CLI interface
- Most requested feature: GUI! Will ship with next release (4.0)

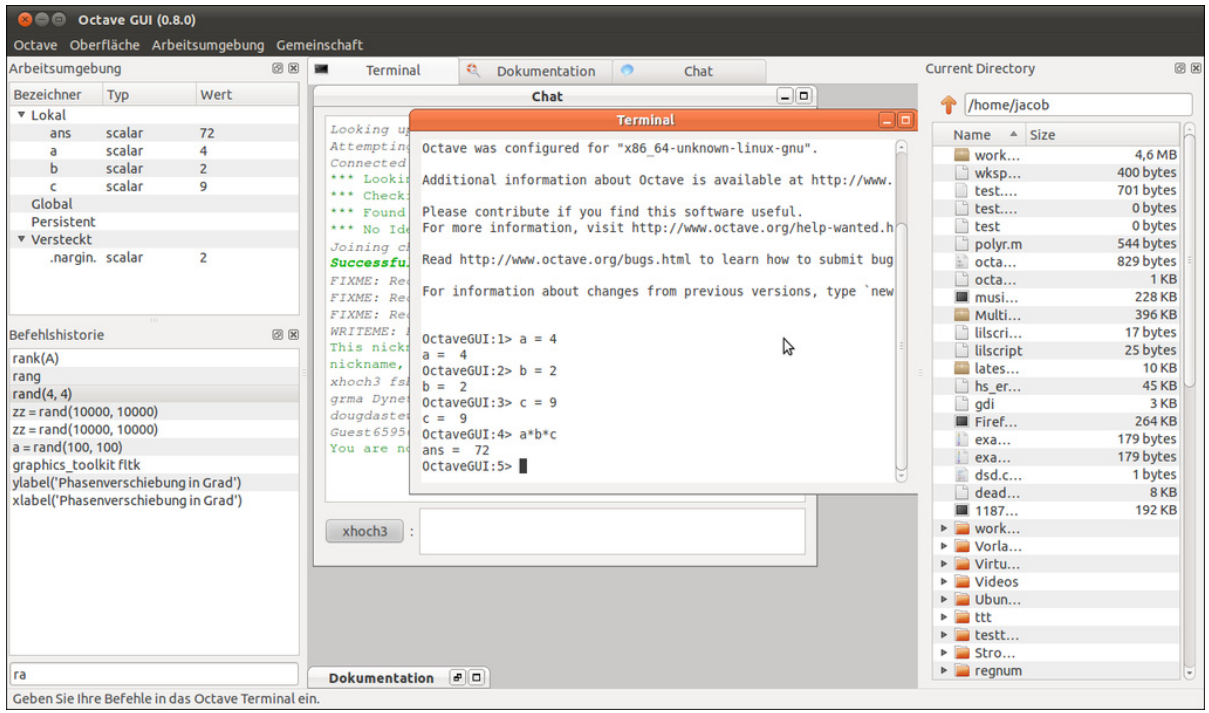


Figure: Qt based GUI development started as J. Dawid's GSoC2012 project

Outline

In the beginning...

- Companion software for chemical reactor textbook by James B. Rawlings and John G. Ekerdt
- John W. Eaton (hereafter, jwe) started coding in 1993

In the beginning...

- Companion software for chemical reactor textbook by James B. Rawlings and John G. Ekerdt
- John W. Eaton (hereafter, jwe) started coding in 1993

Rawlings said...

Why don't you call it "Octave"?

- Octave refers to Octave Levenspiel, nothing to do with music ...

jwe is a lone wolf...

jwe works almost completely alone for first four or five years.

jwe is a lone wolf...

jwe works almost completely alone for first four or five years.

In the very beginning...

- No mailing lists
- No widespread announcements
- No VCS (these were dark times)

Contributions slowly trickle in

Timeline

1989 Planning stages

1992 Development begins

1993 First public announcement

1994 Version 1.0

1996 Version 2.0

1998 Version 2.1 development

2004 Version 2.9 in preparation for 3.0 release

2007 Version 3.0 major upgrade

2010 Version 3.2.4, last before using hg

2011 Version 3.4.0

2012 Version 3.6.4

Contributions slowly trickle in

Milestones

1994 Most of the current basic functionality already in. (Much was written during its first two years!)

Contributions slowly trickle in

Milestones

1994 Most of the current basic functionality already in. (Much was written during its first two years!)

1995 Structs, MATLAB-style plot() command.

1998 Original sparse matrix implementation

2001 Octave-Forge's first commit

2006 MEX interface in core

2007 Implementation of handle graphics, full support for sparse matrices

2009 OpenGL plotting

2010 -forge option for pkg.m

2011 Profiler

2012 Nested functions

Contributions slowly trickle in

Milestones

1994 Most of the current basic functionality already in. (Much was written during its first two years!)

1995 Structs, MATLAB-style plot() command.

1998 Original sparse matrix implementation

2001 Octave-Forge's first commit

2006 MEX interface in core

2007 Implementation of handle graphics, full support for sparse matrices

2009 OpenGL plotting

2010 -forge option for pkg.m

2011 Profiler

2012 Nested functions

2013? GUI, JIT compiling

Outline

Web resources

Web pages

- Octave website
- Octave-Forge website
- Octave wiki

Users communication

- Octave users mailing list
- Octave-Forge mailing list
- #octave channel in Freenode
- Savannah bug tracker

Web resources

Web pages

- Octave website
- Octave-Forge website
- Octave wiki

Users communication

- Octave users mailing list
- Octave-Forge mailing list
- #octave channel in Freenode
- Savannah bug tracker

Developers collaboration

- Octave Mercurial repository
- Octave-Forge Subversion repository

Social structure

- Like all free projects, every user is a potential developer.
- 15 current Core developers (with write access to repo)

	Member
	Ben Abbott <bpabbott>
	Marco Caliari <caliari>
	Carlo de Falco <cdf>
	David Bateman <dbateman>
	Max Brister <fisheater>
	Michael Goffioul <goffioul>
	Jacob Dawid <jacobdawid>
	Jordi Gutiérrez Hermoso <jordigh>
	John W. Eaton <jwe>
	Kai Habel <kahacjde>
	Konstantinos Poulios <logari81>
	Mike Miller <mtmiller>
	Philip Nienhuis <philipnienhuis>
	Rik <rik5>
	Torsten <ttl>

Social structure

- Like all free projects, every user is a potential developer.
- 15 current Core developers (with write access to repo)
- 49 currently registered 'Forge developers (38 active)

Group	Users	Permissions
Admin	cdf (cdf) Thomas Treichl (treichl) Carnë Draug (carandraug) soren hauberg (hauberg) Paul Kienzle (pkienzle) David Bateman (adb014) Juan Pablo Carbajal (jpicarbajal) Add	 read admin create update
	All users in Admin group Francesco Potortë (fpoto) alex (abarth93) Rafael Vázquez (rafavzqz) Michele Martone (michelemartone) marco atzeri (matzeri) Alex (axkma) Andrius Sutas (eandrius) Etienne Grossmann (etienne) Thomas Weber (thomas-weber) andy buckle (blondandy) Paul Dreik (pauldreik) Marco Merlin (batman52) Pascal Dupuis (cdemills) Massimiliano Culpo (culpo) Ben Lewis (benjf5) Mike Miller (mtmiller) Dirk Schmidt (dreisamrd) jgpallero (jgpallero) slackydeb (slackydeb) Arno Onken (asnelt) Thomas Sailer (tsailer)	

Social structure

- Like all free projects, every user is a potential developer.
- 15 current Core developers (with write access to repo)
- 49 currently registered 'Forge developers (38 active)
- 296 total contributors over all time

Ben Abbott Andy Adler Giles Anderson Joel Andersson Muthiah Annamalai Marco Atzeri Shai Ayal Roger Banks Ben Barrowes Alexander Barth David Bateman Heinz Bauschke Roman Belov Karl Berry David Billinghurst Don Bindner Jakub Bogusz Moritz Borgmann Paul Boven Richard Bovey John Bradshaw Marcus Brinkmann Remy Bruno Ansgar Burchard Marco Caliari Daniel Calvelo John C. Campbell Juan Pablo Carbajal Jean-Francois Cardoso Joao Cardoso Larrie Carr David Castelow Vincent Cautaerts Clinton Chee Albert Chin-A-Young Carsten Clark J. D. Cole Martin Costabel Michael Creel Jeff Cunningham Martin Dalecki Jorge Barros de Abreu Carlo de Falco Jacob Dawid Thomas D. Dean Philippe Defert Bill Denney Fabian Deutsch Christos Dimitrakakis David M. Doolin Carnë Draug Pascal A. Dupuis John W. Eaton Dirk Eddelbuettel Pieter Eendebak Paul Eggert Stephen Eglen Peter Ekberg Rolf Fabian Gunnar Farnebäck Stephen Fegan Ramon Garcia Fernandez Torsten Finke Jose Daniel Munoz Frias Brad Froehle Castor Fu Eduardo Gallestey Walter Gautschi Klaus Gebhardt Driss Ghaddab Nicolo Giorgetti Michael D. Godfrey Michael Goffioul Glenn Golden Tomislav Goles Keith Goodman Brian Gough Steffen Groot Etienne Grossmann David Grundberg Peter Gustafson Kai Habel Patrick Häcker William P. Y. Hadisoeseno Jaroslav Hajek Benjamin Hall Kim Hansen Søren Hauberg Dave Hawthorne Daniel Heiserer Martin Helm Stefan Hepp Jordi Gutiérrez Hermoso Yozo Hida Ryan Hinton Roman Hodek A. Scottedward Hodel Richard Allan Holcombe Tom Holroyd David Hoover Kurt Hornik Christopher Hulbert Cyril Humbert Teemu Ikonen Alan W. Irwin Geoff Jacobsen Mats Jansson Cai Jianming Steven G. Johnson Heikki Junes

Social structure

- Like all free projects, every user is a potential developer.
- 15 current Core developers (with write access to repo)
- 49 currently registered 'Forge developers (38 active)
- 296 total contributors over all time
- How many users? Thousands? Millions?

From user to developer

This is a FAQ

From user to developer

This is a FAQ

How can I contribute?

- Code (obviously)
- Money (pay-what-you-need)
- Documentation (especially examples)
- Wiki maintenance
- Help in the mailing list
- Bug reporting

From user to developer

This is a FAQ

How can I contribute?

- Code (obviously)
- Money (pay-what-you-need)
- Documentation (especially examples)
- Wiki maintenance
- Help in the mailing list
- Bug reporting

Student projects

Google Summer of Code

- GSoC 2011
  - Daniel Kraft, Profiler
- GSoC 2012
  - Jacob Dawid, Qt GUI
  - Max Brister, JIT
  - Ben Lewis, Lest Squares Spectral Analysis

European Space Agency's Summer of Code in Space

- SOCIS 2012
  - Wendy Liu, Agora Octave
  - Andrius Sutas, Instrument-Control

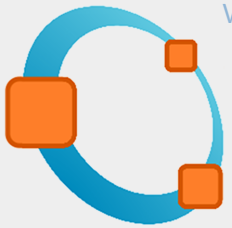
Student projects

Google Summer of Code

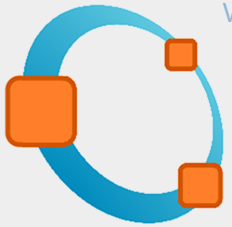
- GSoC 2011
  - Daniel Kraft, Profiler
- GSoC 2012
  - Jacob Dawid, Qt GUI
  - Max Brister, JIT
  - Ben Lewis, Lest Squares Spectral Analysis

European Space Agency's Summer of Code in Space

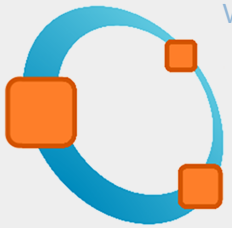
- SOCIS 2012
  - Wendy Liu, Agora Octave
  - Andrius Sutas, Instrument-Control



Outline



Outline

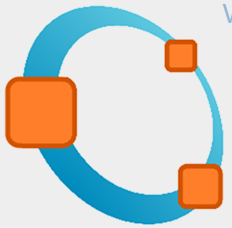


Octave-Forge

Octave-Forge

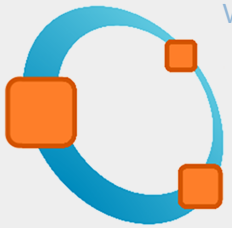
Octave Forge Is a place for concurrently developing and distributing extension packages for Octave.

- Each package has a *maintainer* responsible for updating and releasing new versions of the package
- Some packages are maintained by *The Community*
- Installation via an integrated *package manager*



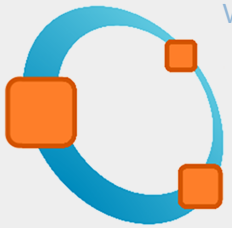
PKG

```
>> pkg install -forge miscellaneous
For information about changes from previous versions of the ↩
     miscellaneous package, run: news ("miscellaneous").
>> pkg list
Package Name           | Version | Installation directory
-----------------------+---------+----------------------
                 bim   |   1.1.1 | ~/octave/bim-1.1.1
                 fpl   |   1.3.3 | ~/octave/fpl-1.3.3
             general   |   1.3.1 | ~/octave/general-1.3.1
            geometry   |   1.6.0 | ~/octave/geometry-1.6.0
       miscellaneous   |   1.2.0 | ~/octave/miscellaneous-1.2.0
>> pkg load miscellaneous
>> pkg list
Package Name           | Version | Installation directory
-----------------------+---------+----------------------
                 bim   |   1.1.1 | ~/octave/bim-1.1.1
                 fpl   |   1.3.3 | ~/octave/fpl-1.3.3
             general   |   1.3.1 | ~/octave/general-1.3.1
            geometry   |   1.6.0 | ~/octave/geometry-1.6.0
       miscellaneous *|   1.2.0 | ~/octave/miscellaneous-1.2.0
```



PKG

```
>> pkg describe bim -verbose
---
Package name:
bim
Version:
1.1.1
Short description:
Package for solving Diffusion Advection Reaction (DAR) Partial Differential Equations
Status:
Not loaded
---
Provides:
Matrix assembly
bim1a_advection_diffusion
bim1a_advection_upwind
bim2a_advection_diffusion
...
Pre-processing and Post-processing computations
bim2c_mesh_properties
...
>>
```



Some interesting packages

bim

Package Version:	1.1.1
Last Release Date:	2012-10-26
Package Author:	Carlo de Falco, Culpo Massimiliano
Package Maintainer:	Carlo de Falco
License:	GPLv2+

Download Package
(older versions)

Function Reference

Description

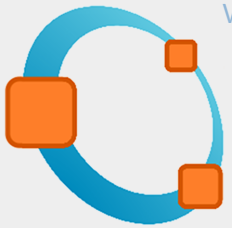
Package for solving Diffusion Advection Reaction (DAR) Partial Differential Equations

Details

Dependencies: Octave (>= 3.6.0) fpl (>= 0.0.0) msh (>= 0.0.0)
Autoload: No

Package: bim

usage examples in the wiki



Some interesting packages

msh

Package Version:	1.0.6
Last Release Date:	2012-10-21
Package Author:	Carlo de Falco, Massimiliano Culpo
Package Maintainer:	Carlo de Falco
License:	GPLv2+

Download Package
(older versions)

Function Reference

Description

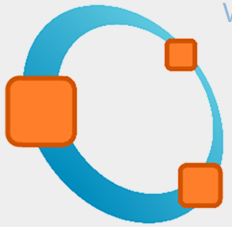
Create and manage triangular and tetrahedral meshes for Finite Element or Finite Volume PDE solvers.
Use a mesh data structure compatible with PDEtool. Rely on gmsh for unstructured mesh generation.

Details

Dependencies: Octave (>= 3.0.0) splines (>= 0.0.0)
Autoload: No

Package: msh

usage examples in the wiki



Some interesting packages

fpl

Package Version:	1.3.3
Last Release Date:	2012-11-01
Package Author:	Carlo de Falco, Massimiliano Culpo and others
Package Maintainer:	Carlo de Falco, Massimiliano Culpo
License:	GPLv3+

Download Package
(older versions)

Function Reference

Description

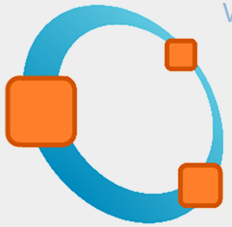
Collection of routines to export data produced by Finite Elements or Finite Volume Simulations in formats used by some visualization programs.

Details

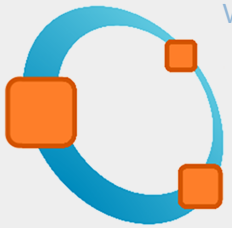
Dependencies: Octave (>= 3.2.3)
Autoload: No

Package: fpl

usage examples in the wiki

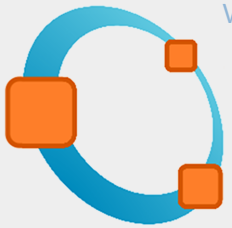


Outline



Broadcasting

- Since 3.6.0, Octave automatically broadcasts arrays when using elementwise binary operators.
- Corresponding array dimensions must either be equal or, one of them must be 1.
- In case all dimensions are equal, ordinary element-by-element arithmetic takes place.
- When one of the dimensions is 1, the array with that singleton dimension gets copied along that dimension until it matches the dimension of the other array.



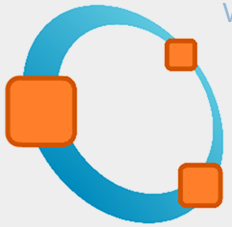
Broadcasting

```
x = [1 2 3; 4 5 6; 7 8 9];
y = [10 20 30];
x + y
    11   22   33
    14   25   36
    17   28   39
```

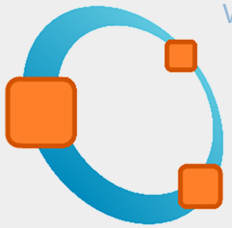
- Without broadcasting, x + y would be an error because dimensions do not agree.
- With broadcasting it is as if the following operation were performed

```
x = [1 2 3;4 5 6;7 8 9];
y = [10 20 30; 10 20 30; 10 20 30];
x + y
    11   22   33
    14   25   36
    17   28   39
```

Other notable differences with Matlab, listed in the wiki



Outline

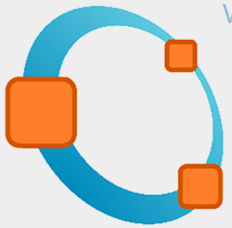


dld-functions

Implement an Octave interpreter function in C++

```
1  #include <octave/oct.h>
2
3  DEFUN_DLD(dld,args,nargout,"dld (array) \nreturn the elements of ↩
       the array in reverse order\n")
4  {
5    octave_value_list retval;
6    int nargin = args.length ();
7
8    if (nargin != 1)
9      print_usage ();
10   else
11     {
12       Array<double> a = args(0).array_value ();
13       if (! error_state)
14         {
15           Array<double> b (a);
```

source code of the example

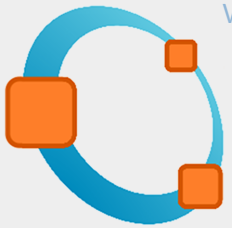


dld-functions

Implement an Octave interpreter function in C++

```
16              double* ap = a.fortran_vec ();
17              double* bp = b.fortran_vec ();
18              for (octave_idx_type i = a.numel () - 1, j = 0; i >= 0; ↩
                    i--, j++)
19                bp[i] = ap[j];
20              retval = octave_value (b);
21            }
22        }
23    return retval;
24  }
```

source code of the example

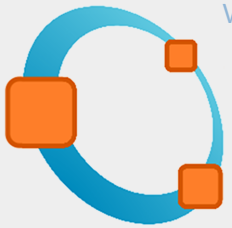


dld-functions

Implement an Octave interpreter function in C++

```
>> mkoctfile dld.cc
>> a = randn (5)
a =

   0.395421  -1.425232  -0.176544   1.055205   2.229371
  -0.241893   0.035004  -0.296543  -1.710613   0.444318
  -0.752467  -2.220469   2.380951   0.766246   1.196153
   1.404672   0.623112   1.182609   0.196125   0.609325
  -0.687019   0.646079   2.239012  -0.495169   1.488314

>> b = dld (a)
b =

   1.488314  -0.495169   2.239012   0.646079  -0.687019
   0.609325   0.196125   1.182609   0.623112   1.404672
   1.196153   0.766246   2.380951  -2.220469  -0.752467
   0.444318  -1.710613  -0.296543   0.035004  -0.241893
   2.229371   1.055205  -0.176544  -1.425232   0.395421

>>
```

source code of the example

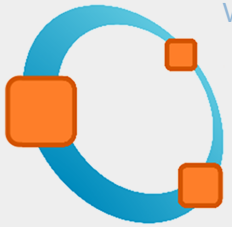


liboctave

Use Octave's Matrix/Array Classes in a C++ application

```
1  #include <iostream>
2  #include <octave/oct.h>
3
4  int main (void)
5  {
6
7    Matrix A (4, 4);
8    for (octave_idx_type i = 0; i < 4; i++)
9      for (octave_idx_type j = 0; j < 4; j++)
10       A(i,j) = 1.0 / (static_cast<double> (i) +
11                       static_cast<double> (j) + 1.0);
12
13   ColumnVector b (4, 1.0);
14   ColumnVector x = A.solve (b);
15
16   std::cout << "A = " << std::endl << A << std::endl
17             << "b = " << std::endl << b << std::endl
18             << "x = " << std::endl << x << std::endl;
19
20   return 0;
21 }
```

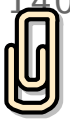
source code of the example



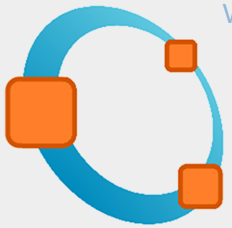
liboctave

Use Octave's Matrix/Array Classes in a C++ application

```
$ mkoctfile --link-stand-alone standalone.cc
$ ./a.out
A =
 1 0.5 0.333333 0.25
 0.5 0.333333 0.25 0.2
 0.333333 0.25 0.2 0.166667
 0.25 0.2 0.166667 0.142857
b =
1
1
1
1
x =
-4
60
-180
140
```



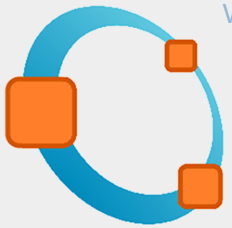
source code of the example



Embedding Octave

You can embed the Octave interpreter in your C++ application

```
1  #include <iostream>
2  #include <octave/oct.h>
3  #include <octave/octave.h>
4  #include <octave/parse.h>
5
6  int main (void)
7  {
8    string_vector octave_argv (2);
9    octave_argv(0) = "embedded";
10   octave_argv(1) = "-q";
11
12   octave_main (2, octave_argv.c_str_vec (), 1);
13
14
15   octave_value_list out = feval ("version", octave_value_list (), ↩
         1);
16   std::cout << out(0).string_value () << std::endl;
```

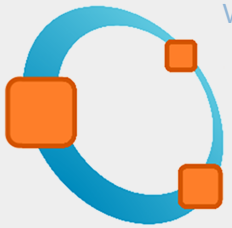


Embedding Octave

You can embed the Octave interpreter in your C++ application

```
18    Matrix A (4, 4);
19    for (octave_idx_type i = 0; i < 4; i++)
20      for (octave_idx_type j = 0; j < 4; j++)
21        A(i,j) = 1.0 / (static_cast<double> (i) +
22                        static_cast<double> (j) + 1);
23
24    ColumnVector b (4, 1.0);
25
26    out = feval ("mldivide", octave_value (A), octave_value (b), 1);
27
28    return 0;
29  }
```

source code of the example

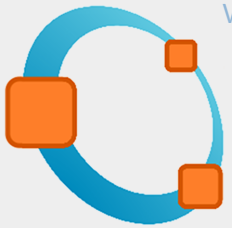


An advanced example

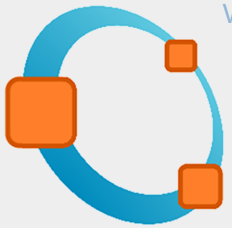
Add a new class to the Octave interpreter and work around Octave's pass-by-value semantics


source code of the example (.cc)


source code of the example (.h)



Outline

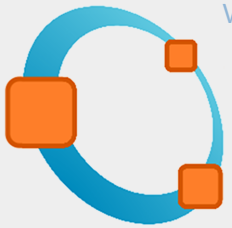


parcellfun and pararrayfun

Parcellfun is distributed in the package "general" it implements parallelization via `fork ()` and `pipe ()`

```
1  tic ();
2  nel = 100;
3  U0 = randn (200, 1);
4  us = zeros (101, 200);
5  for ii=1:numel (U0)
6
7    x = transpose (linspace (0, 1, nel+1));
8    A = bim1a_laplacian (x, 1, 1);
9    b = bim1a_rhs (x, 1, 1);
10
11   us(:,ii) = zeros (size (x));
12   us(1,ii) = U0(ii);
13
14   res = @(X) A(2:end-1, 2:end-1) * X - (b(2:end-1) - A(2:end-1, [1
         end]) * us([1 end], ii));
15   us(2:end-1,ii) = fsolve (res, us(2:end-1,ii));
16
17 endfor
18 toc ()
```

source code of the example

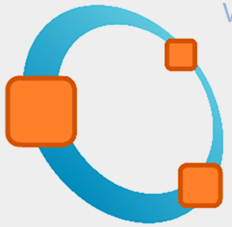


parcellfun and pararrayfun

Parcellfun is distributed in the package "general" it implements parallelization via `fork ()` and `pipe ()`

```
1  function u = poisson1d (u0)
2    nel = 100;
3    x = transpose (linspace (0, 1, nel+1));
4    A = bim1a_laplacian (x, 1, 1);
5    b = bim1a_rhs (x, 1, 1);
6
7    u = zeros (size (x));
8    u(1) = u0;
9
10   res = @(X) A(2:end-1, 2:end-1) * X - (b(2:end-1) - A(2:end-1, [1↩
         end]) * u([1 end]));
11   u(2:end-1) = fsolve (res, u(2:end-1));
12 endfunction
13 tic ();
14 U0 = num2cell (randn (1, 200));
15 up = parcellfun (2, @poisson1d, U0, "UniformOutput", true, "↩
     VerboseLevel", 2);
16 toc ()
```

source code of the example



openmpi_ext

The package `openmpi_ext` provides wrappers for the main MPI functions in `openmpi`

openmpi_ext

Package Version:	1.1.0
Last Release Date:	2012-8-29
Package Author:	Riccardo Corradini , Jaroslav Hajek, Carlo de Falco
Package Maintainer:	the Octave Community
License:	GPLv3+

Download Package
(older versions)

Function Reference

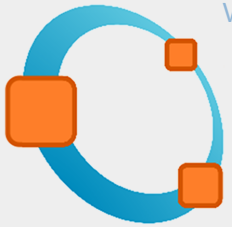
Description

MPI functions for parallel computing using simple MPI Derived Datatypes.

Details

Dependencies: Octave (>= 3.2.4)
Autoload: No
Package: openmpi_ext

source code of the example

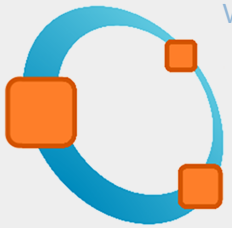


openmpi_ext

The package openmpi_ext provides wrappers for the main MPI functions in openmpi

```
1  T=clock;
2  MPI_ANY_SOURCE = -1;
3  MPI_Init ();
4  MPI_COMM_WORLD = MPI_Comm_Load ("NEWORLD");
5  rnk    = MPI_Comm_rank (MPI_COMM_WORLD);
6  siz    = MPI_Comm_size (MPI_COMM_WORLD);
7  SLV = logical(rnk);
8  MST = ~ SLV;
9  width=1/N; lsum=0;
10 i=rnk:siz:N-1;
11 x=(i+0.5)*width;
12 lsum=sum(4./(1+x.^2));
13 TAG=7;
14 if SLV
15   MPI_Send (lsum, 0, TAG, MPI_COMM_WORLD);
16 else
17   Sum =lsum;
18   for slv=1:siz-1
19     lsum = MPI_Recv (MPI_ANY_SOURCE, TAG, MPI_COMM_WORLD);
20     Sum += lsum;
21   endfor
22 endif
23 MPI_Finalize ();
```

source code of the example



openmpi_ext

The package openmpi_ext provides wrappers for the main MPI functions in openmpi

```
1  mpirun --hostfile $HOSTFILE -np $NUMBER_OF_MPI_NODES octave --eval↩
       "pkg load openmpi_ext; Pi ()"
```


source code of the example