

GNU Octave

A free high-level tool for Scientific Computing

Carlo de Falco, Jordi Gutiérrez Hermoso

June 24, 2013 - OctConf - MOX - Politecnico di Milano



1 What is Octave?

- Definition
- History
- Community dynamics

2 Octave and ...

- Octave and Octave-Forge
- Octave and Matlab
- Octave and C++
- Octave and Parallel Computing
- PDEs in Octave-Forge



Outline



1 What is Octave?

- Definition
- History
- Community dynamics

2 Octave and ...

- Octave and Octave-Forge
- Octave and Matlab
- Octave and C++
- Octave and Parallel Computing
- PDEs in Octave-Forge



Outline



- 1 What is Octave?**
 - Definition
 - History
 - Community dynamics



Definitions



What is Octave?

"A free numerical environment mostly compatible with MATLAB"

"free" = "libero" ≠ "gratis"

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.



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
- Java?



Features



Current features

- N-d arrays, linear algebra, sparse matrices
- Nonlinear equations
- Partial/Ordinary/Algebraic Differential Equations,
- Image processing, statistics, special functions
- OOP (*old style*)
- Package manager
- Many more...

Features in development

- GUI
- JIT compiling
- classdef OOP



What does it look like

- Primarily a CLI interface

The screenshot shows a Linux desktop environment with several windows open:

- Terminal window:** Displays Octave code for generating a logo and a 3D surface plot.
- Figure 1:** A 2D plot of the Octave logo, which is a blue circle with orange squares at the top and bottom.
- Figure 2:** A 3D surface plot of a bell-shaped curve on a grid.
- Help window:** Shows the documentation for the `imshow` function, including usage examples and a warning about using GUI features.

Figure : CLI screenshot



What does it look like



- Most requested feature: GUI!



What does it look like



- Most requested feature: GUI!

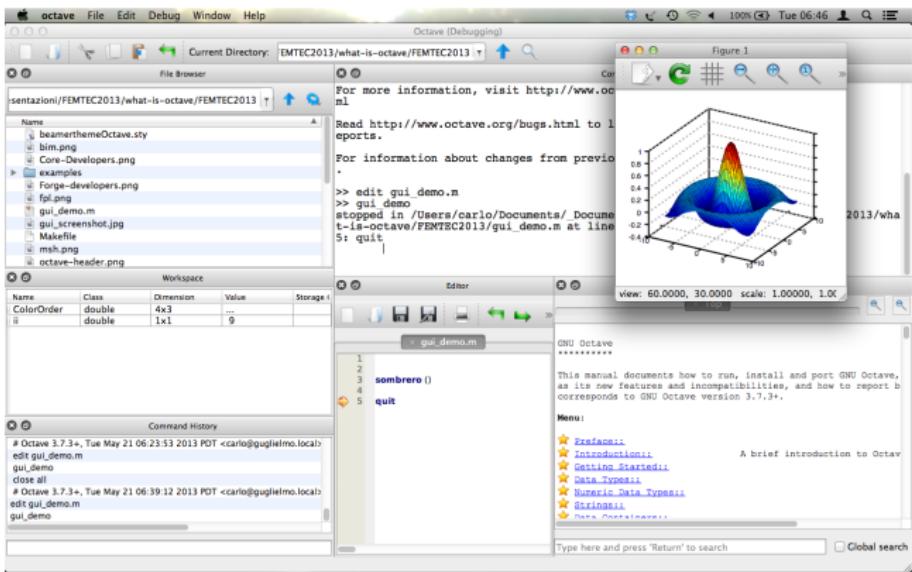


Figure : Qt based GUI Will ship with next release (4.0, expected 2013)



Outline



1 What is Octave?

- Definition
- History
- Community dynamics



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
- 2013 20th birthday! (and Version 4.0 Hopefully!)



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, Java, 64bit indexing



Outline



1 What is Octave?

- Definition
- History
- Community dynamics



Web resources



Web pages

- Octave website
- Octave-Forge website
- Octave wiki

Users communication

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

Developers collaboration

- Octave Mercurial repository
- Octave-Forge Subversion (+ Mercurial) repository



Social structure



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

	Member
	Ben Abbott <bpabbott>
	Marco Caliari <caliari>
	Carnë Draug <carandraug>
	Carlo de Falco <cdf>
	David Bateman <dbateman>
	Max Brister <fisheater>
	Michael Goffioul <goffioul>
	Jacob Dawid <jacobdawid>
	<u>Jordi Gutiérrez Hermoso <jordigh></u>
	<u>John W. Eaton <jwe></u>
	Kai Habel <kahacjde>
	Konstantinos Poulios <logari81>
	John Donoghue <lostbard>
	Mike Miller <mtnmiller>
<u>Squad</u>	<u>Octave Forge <octave-forge></u>
	Philip Nienhuis <philipnienhuis>
	Rik <rik5>
	Torsten <ttl>



Social structure

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

Admin



- cdf (cdf)
- Camë Draug (carandraug)
- soren hauberg (hauberg)
- David Bateman (adb014)
- Add

All users in Admin group

- alex (abarth93)
- Rafael Vázquez (rafavzqz)
- Michele Martone (michelemartone)
- marco atzeri (matzeri)
- andy buckle (blondandy)
- Pascal Dupuis (cdemills)
- Mike Miller (mtmiller)
- Arno Onken (asneit)
- Adam H Aitkenhead (adam78a)
- Olaf Till (i7tiol)
- JohanB (johanb88)
- Michael Goffioul (goffioul)
- Moreno Marzolla (mmarzolla)

Developer



- Jordi Gutiérrez Hermoso (jordigh)
- Philip Nienhuis (prnienhuis)
- imarkov (imarkov)
- Nir Krakauer (nir-krakauer)
- Juan Pablo Carbajal (jpicarbajal)
- Alois Schloegl (schloegl)
- paramaniac (paramaniac)
- John W. eaton (jwe)
- jgh (jgh)





Social structure



- Like all free projects, every user is a potential developer.
- 16 current Core developers (with write access to repo)
- 29 currently registered 'Forge' developers
- 305 total contributors over all time

Ben Abbott	Jean-Francois Cardoso	Paul Eggert	Jaroslav Hajek	Jarkko Kaleva
Andy Adler	Joao Cardoso	Stephen Eglen	Benjamin Hall	Mohamed Kamoun
Giles Anderson	Lorrie Carr	Peter Ekberg	Kim Hansen	Lute Kamstra
Joel Andersson	David Castelow	Rolf Fabian	Soren Hauberg	Fotios Kasolos
Muthiah Annamalai	Vincent Cautaerts	Gunnar Farnebeck	Dave Hawthorne	Thomas Kasper
Marcos Atzori	Clinton Chee	Stephen Fegan	Daniel Heiserer	Joel Keay
Shai Ayal	Albert Chin-A-Young	R. Garcia Fernandez	Martin Helm	Mumit Khan
Roger Banks	Carsten Clark	Torsten Finke	Stefan Hepp	Paul Kienzle
Ben Barrowes	J. D. Cole	J.D.M. Frias	Martin Hepperle	Aaron A. King
Alexander Barth	Martin Costabel	Brad Froehle	Jordi Gutierrez Hermoso	Arno J. Klaassen
David Bateman	Michael Creel	Castor Fu	Yozo Hida	Alexander Klein
Heinz Bauschke	Jeff Cunningham	Eduardo Galleteay	Ryan Hinton	Geoffrey Knauth
Julien Bect	Martin Dalecki	Walter Gautschi	Roman Hodek	Heime Kolitveit
Roman Belov	Jorge Barros de Abreu	Klaus Gebhardt	A. Scottedward Hodel	Ken Kouno
Karl Berry	Carlo de Falco	Driss Ghaddab	Richard Allan Holcombe	Kasper Kowalik
David Billinghurst	Jacob David	Nicolo Giorgetti	Tom Holroyd	Daniel Kraft
Don Bindner	Thomas D. Dean	Michael D. Godfrey	David Hoover	Aravindh Krishnamoorthy
Jakub Bogusz	Philippe Defert	Michael Goffioul	Kurt Hornik	Oyvind Kristiansen
Moritz Borgmann	Bill Denney	Glenn Golden	Christopher Hulbert	Piotr Krzyzanowski
Paul Boven	Fabian Deutsch	Tomislav Goles	Cyril Humbert	Volker Kuhlmann
Richard Bovey	Christos Dimitrakakis	Keith Goodman	John Hunt	Tetsuro Kurita
John Bradshaw	Pantxo Diribaŕane	Brian Gough	Teemu Ikonen	Miroslaw Kwasniak
Marcus Brinkmann	Vivek Dogra	Steffen Groot	Alan W. Irwin	Rafael Laboissiere
Max Brister	John Donoghue	Etienne Grossmann	Geoff Jacobson	Kai Labusch
Remy Bruno	David M. Doolin	David Grundberg	Mats Jonsson	Claude Lacoursiere
Ansgar Burchard	Carné Draug	Kyle Quinn	Cai Jianming	Walter Landry
Marco Caliari	Pascal A. Dupuis	Peter Gustafson	Steven G. Johnson	Bill Lash
Daniel Calvelo	John W. Eaton	Kai Haberl	Heikkil Junes	Dirk Laurie
John C. Campbell	Dirk Eddelbuettel	Patrick Haeker	Matthias Jäschke	Maurice LeBrun
Juan Pablo Carbajal	Pieter Eendebak	W.P.Y. Hadisoens	Atsushi Kajita	Friedrich Leisch



Social structure



- Like all free projects, every user is a potential developer.
- 16 current Core developers (with write access to repo)
- 29 currently registered 'Forge' developers
- 305 total contributors over all time

Jyh-min Lin	Kai P. Mueller	Tom Poage	Aleksej Saushev	John Swensen	Fook Fah Yap
Timo Lindfors	Hannes Müller	Orton Poplawski	Alois Schlogl	Daisuke Takago	Sean Young
Benjamin Lindner	Victor Munoz	Ondrej Popp	Michel D. Schmid	Ariel Tankus	Michael Zeising
Ross Lippert	Iain Murray	Jef Poskanzer	Julian Schnidder	Matthew Tenny	Federico Zenith
David Livings	Carmen Navarrete	Francesco Potorti	Nicol N. Schraudolph	Georg Thimm	Alex Zvoleff
Sebastien Loisel	Todd Neal	Konstantinos Poulios	Sebastian Schubert	Duncan Temple Lang	
E. de Castro Lopo	Philip Nienhuis	Jarne Rajahalma	Ludwig Schwabe	Kris Thielemans	
Massimo Lorenzin	Al Niessner	James B. Rawlings	Thomas L. Scofield	Olaf Till	
Emil Lucetiu	Rick Niles	Eric S. Raymond	Daniel J. Sebald	Christophe Tournery	
Hoxide Ma	Takuji Nishimura	Balint Reczey	Dmitri A. Sergatskov	Thomas Treichl	
James Macnicol	Kai Noda	Joshua Redstone	Vanya Sergeev	Karsten Trulsen	
Jens-Uwe Mager	Eric Norum	Lukas Reichlin	Baylis Shanks	Frederick Umminger	
Colin Macdonald	Krzesimir Nowak	Michael Reifенberger	Andryl Shinkarchuk	Utkarsh Upadhyay	
Rob Mohurin	Michael O'Brien	Anthony Richardson	Robert T. Short	Daniel Wagenaar	
Ricardo Marranita	Peter O'Gorman	Jason Riedy	Joseph P. Skudlarek	Stefan van der Walt	
Orestes Mas	Thorsten Ohl	F. Joshua Rigler	John Smith	Peter Van Wieren	
Axel Mathääl	Arne Onken	Petter Risholm	Julius Smith	James R. Van Zandt	
Makoto Matsumoto	V. Ortega-Clavero	Matthew W. Roberts	Shan G. Smith	Risto Vanhanen	
Tatsuro Matsuka	Luis F. Ortiz	Andrew Ross	Peter L. Sondergaard	Gregory Vanuxem	
Laurent Mazet	Scott Pakin	Fabio Rossi	Joerg Specht	Ivana Varekova	
G. D. McBain	Gabriele Pannocchia	Mark van Rossum	Quentin H. Spencer	Thomas Walter	
Alexander Monakov	Sylvain Pelissier	Joe Rothweiler	Christoph Spiel	Andreas Weber	
Christoph Mayer	Per Persson	Kevin Ruland	Richard Stallman	Olaf Weber	
J. Hoffmann Mendes	Primoz Peterlin	Kristian Rumberg	Russell Standish	Thomas Weber	
Ronald van der Meer	Jim Peterson	Ryan Rusaw	Brett Stewart	Rik Wehbring	
Thorsten Meyer	Danilo Piazzalunga	Olli Saarela	Doug Stewart	Bob Weigel	
Petr Mikulik	Nicholas Piper	Toni Saarela	Jonathon Stickel	Andreas Weingessel	
Mike Miller	Elias Pipping	Juhani Saastamoinen	Judd Storrs	Martin Weiser	
Stefan Monnier	Robert Platt	Radek Salac	Thomas Stuart	Michael Weitzel	
Antoine Moreau	Hans Ekkehard Plessner	Ben Sapp	Ivan Sutoris	David Wells	



Social structure



- Like all free projects, every user is a potential developer.
- 16 current Core developers (with write access to repo)
- 29 currently registered 'Forge' developers
- 305 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

Stick around this conference if you want to learn more about how to get involved!



Student projects



Google Summer of Code

- GSoC 2011
 - Daniel Kraft, Profiler
- GSoC 2012
 - Jacob Dawid, Qt GUI; Max Brister, JIT; Ben Lewis, LSSA
- GSoC 2013
 - 9 Slots this year!
 - JIT, Sparse Matrix Factorizations, FEM, Agora, Binary Packaging, Audio

European Space Agency's Summer of Code in Space

- SOCIS 2012
 - Wendy Liu, Agora Octave; Andrius Sutas, Instrument-Control
- SOCIS 2013
 - Students apply!



Student projects



Google Summer of Code

- GSoC 2011
 - Daniel Kraft, Profiler
- GSoC 2012
 - Jacob Dawid, Qt GUI; Max Brister, JIT; Ben Lewis, LSSA
- GSoC 2013
 - 9 Slots this year!
 - JIT, Sparse Matrix Factorizations, FEM, Agora, Binary Packaging, Audio

European Space Agency's Summer of Code in Space

- SOCIS 2012
 - Wendy Liu, Agora Octave; Andrius Sutas, Instrument-Control
- SOCIS 2013
 - Students apply!



Outline



1 What is Octave?

- Definition
- History
- Community dynamics

2 Octave and ...

- Octave and Octave-Forge
- Octave and Matlab
- Octave and C++
- Octave and Parallel Computing
- PDEs in Octave-Forge



2 Octave and ...

- Octave and Octave-Forge
- Octave and Matlab
- Octave and C++
- Octave and Parallel Computing
- PDEs in 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



```
1 >> pkg install -forge miscellaneous
2 For information about changes from previous versions of the ←
   miscellaneous package, run: news ("miscellaneous").
3 >> pkg list
4 Package Name          | Version | Installation directory
5
6      bim    | 1.1.1  | ~/octave/bim-1.1.1
7      fpl    | 1.3.3  | ~/octave/fpl-1.3.3
8      general | 1.3.1  | ~/octave/general-1.3.1
9      geometry | 1.6.0  | ~/octave/geometry-1.6.0
10     miscellaneous | 1.2.0  | ~/octave/miscellaneous-1.2.0
11 >> pkg load miscellaneous
12 >> pkg list
13 Package Name          | Version | Installation directory
14
15      bim    | 1.1.1  | ~/octave/bim-1.1.1
16      fpl    | 1.3.3  | ~/octave/fpl-1.3.3
17      general | 1.3.1  | ~/octave/general-1.3.1
18      geometry | 1.6.0  | ~/octave/geometry-1.6.0
19     miscellaneous *| 1.2.0  | ~/octave/miscellaneous-1.2.0
```



PKG



```
1 >> pkg describe bim --verbose
2
3 Package name:
4 bim
5 Version:
6 1.1.1
7 Short description:
8 Package for solving Diffusion Advection Reaction (DAR) Partial ←
     Differential Equations
9 Status:
10 Not loaded
11
12 Provides:
13 Matrix assembly
14 bim1a_advection_diffusion
15 bim1a_advection_upwind
16 bim2a_advection_diffusion
17 ...
18 Pre-processing and Post-processing computations
19 bim2c_mesh_properties
20 ...
21 >>
```



2 Octave and ...

- Octave and Octave-Forge
- Octave and Matlab
- Octave and C++
- Octave and Parallel Computing
- PDEs in Octave-Forge



How is Octave different from Matlab?



- Nested Functions
- Differences in core syntax
- Just-In-Time compiler
- Compiler
- Graphic handles
- GUI functions
- Simulink
- MEX-Files
- Block comments
- Profiler
- Short-circuit & and | operators
- Solvers for singular, under- and over-determined matrices
- Octave extensions



How is Octave different from Matlab?



- Nested Functions
- Differences in core syntax
 - Some limitations on the use of function handles.
 - Some limitations of variable argument lists on the LHS of an expression.
 - Matlab classdef object oriented programming is not yet supported.
- Just-In-Time compiler
- Compiler
- Graphic handles
- GUI functions
- Simulink
- MEX-Files
- Block comments
- Profiler
- Short-circuit & and | operators
- Solvers for singular, under- and over-determined matrices
- Octave extensions



How is Octave different from Matlab?



- Nested Functions
- Differences in core syntax
- Just-In-Time compiler
- Compiler
- Graphic handles
- GUI functions
- Simulink
- MEX-Files
- Block comments
- Profiler
- Short-circuit & and | operators
- Solvers for singular, under- and over-determined matrices
- Octave extensions
 - Octave comments can start with #
 - Indexing can be applied to all objects in Octave, e.g. `sin(x)(1 : 10)`
 - Octave has the operators `++`, `--`, `- =`, `+ =`, `* =`, ...
 - Character strings in Octave can be denoted with double or single quotes.



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



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

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

Other notable differences with Matlab, listed in the wiki



2 Octave and ...

- Octave and Octave-Forge
- Octave and Matlab
- Octave and C++**
- Octave and Parallel Computing
- PDEs in Octave-Forge



dld-functions



Implement an Octave interpreter function in C++

```
1 #include <octave/oct.h>
2
3 DEFUN_DLD(dld, args, nargin, "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



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; ←
19             i--, j++)
20             bp[i] = ap[j];
21         retval = octave_value (b);
22     }
23 }
24 }
```



source code of the example



Implement an Octave interpreter function in C++

```
1 >> mkoctfile dld.cc
2 >> a = randn (5)
3 a =
4
5 0.395421 -1.425232 -0.176544 1.055205 2.229371
6 -0.241893 0.035004 -0.296543 -1.710613 0.444318
7 -0.752467 -2.220469 2.380951 0.766246 1.196153
8 1.404672 0.623112 1.182609 0.196125 0.609325
9 -0.687019 0.646079 2.239012 -0.495169 1.488314
10
11 >> b = dld (a)
12 b =
13
14 1.488314 -0.495169 2.239012 0.646079 -0.687019
15 0.609325 0.196125 1.182609 0.623112 1.404672
16 1.196153 0.766246 2.380951 -2.220469 -0.752467
17 0.444318 -1.710613 -0.296543 0.035004 -0.241893
18 2.229371 1.055205 -0.176544 -1.425232 0.395421
19
20 ➜
```

 source code of the example



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



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

```
1 $ mkostfile --link-stand-alone standalone.cc
2 $ ./a.out
3 A =
4 1 0.5 0.333333 0.25
5 0.5 0.333333 0.25 0.2
6 0.333333 0.25 0.2 0.166667
7 0.25 0.2 0.166667 0.142857
8 b =
9 1
10 1
11 1
12 1
13 x =
14 -4
15 60
16 -180
17 140
```



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 (), ←
16        1);
17    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

```
1 class myobject : public octave_base_value
2 {
3     private:
4     DECLARE_OCTAVE_ALLOCATOR;
5     DECLARE_OV_TYPEID_FUNCTIONS_AND_DATA;
6 };
7
8 DEFINE_OCTAVE_ALLOCATOR(myobject);
9 DEFINE_OV_TYPEID_FUNCTIONS_AND_DATA(myobject, "myobject", "myobject"↔
    );
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
678
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
727
728
729
729
730
731
732
733
734
735
736
737
737
738
739
739
740
741
742
743
744
745
746
747
747
748
749
749
750
751
752
753
754
755
756
757
757
758
759
759
760
761
762
763
764
765
766
767
767
768
769
769
770
771
772
773
774
775
776
777
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
796
797
798
798
799
799
800
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
847
847
848
849
849
850
851
852
853
854
855
856
857
857
858
859
859
860
861
862
863
864
865
866
867
867
868
869
869
870
871
872
873
874
875
876
877
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
907
908
909
909
910
911
912
913
914
915
916
916
917
918
918
919
919
920
921
922
923
924
925
926
927
927
928
929
929
930
931
932
933
934
935
936
937
937
938
939
939
940
941
942
943
944
945
946
946
947
948
948
949
949
950
951
952
953
954
955
956
957
957
958
959
959
960
961
962
963
964
965
966
966
967
968
968
969
969
970
971
972
973
974
975
976
976
977
978
978
979
979
980
981
982
983
984
985
986
986
987
988
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1016
1017
1018
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1026
1027
1028
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1036
1037
1038
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1046
1047
1048
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1056
1057
1058
1058
1059
1059
1060
1061
1062
1063
1064
1065
1065
1066
1067
1067
1068
1068
1069
1069
1070
1071
1072
1073
1074
1075
1075
1076
1077
1077
1078
1078
1079
1079
1080
1081
1082
1083
1084
1085
1085
1086
1087
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1094
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1104
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1111
1112
1113
1114
1114
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1121
1122
1123
1124
1124
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1131
1132
1133
1134
1134
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1141
1142
1143
1144
1144
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1151
1152
1153
1154
1154
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1161
1162
1163
1164
1164
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1171
1172
1173
1174
1174
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1181
1182
1183
1184
1184
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1194
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1204
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1211
1212
1213
1214
1214
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1221
1222
1223
1224
1224
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1231
1232
1233
1234
1234
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1241
1242
1243
1244
1244
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1251
1252
1253
1254
1254
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1261
1262
1263
1264
1264
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1271
1272
1273
1274
1274
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1281
1282
1283
1284
1284
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1294
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1304
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1311
1312
1313
1314
1314
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1321
1322
1323
1324
1324
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1331
1332
1333
1334
1334
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1341
1342
1343
1344
1344
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1351
1352
1353
1354
1354
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1361
1362
1363
1364
1364
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1371
1372
1373
1374
1374
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1381
1382
1383
1384
1384
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1394
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1404
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1411
1412
1413
1414
1414
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1421
1422
1423
1424
1424
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1431
1432
1433
1434
1434
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1441
1442
1443
1444
1444
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1454
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1461
1462
1463
1464
1464
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1471
1472
1473
1474
1474
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1481
1482
1483
1484
1484
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1494
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1504
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1511
1512
1513
1514
1514
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1521
1522
1523
1524
1524
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1531
1532
1533
1534
1534
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1541
1542
1543
1544
1544
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1554
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1561
1562
1563
1564
1564
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1571
1572
1573
1574
1574
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1581
1582
1583
1584
1584
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1594
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1604
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1611
1612
1613
1614
1614
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1621
1622
1623
1624
1624
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1631
1632
1633
1634
1634
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1641
1642
1643
1644
1644
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1654
1655
1656
1656
1657
1657
1658
1658
1659
1659
1660
1661
1662
1663
1664
1664
1665
1666
1666
1667
1667
1668
1668
1669
1669
1670
1671
1672
1673
1674
1674
1675
1676
1676
1677
1677
1678
1678
1679
1679
1680
1681
1682
1683
1684
1684
1685
1686
1686
1687
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1694
1695
1696
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1704
1705
1706
1706
1707
1707
1708
1708
1709
1709
1710
1711
1712
1713
1714
1714
1715
1716
1716
1717
1717
1718
1718
1719
1719
1720
1721
1722
1723
1724
1724
1725
1726
1726
1727
1727
1728
1728
1729
1729
1730
1731
1732
1733
1734
1734
1735
1736
1736
1737
1737
1738
1738
1739
1739
1740
1741
1742
1743
1744
1744
1745
1746
1746
1747
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1754
1755
1756
1756
1757
1757
1758
1758
1759
1759
1760
1761
1762
1763
1764
1764
1765
1766
1766
1767
1767
1768
1768
1769
1769
1770
1771
1772
1773
1774
1774
1775
1776
1776
1777
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1784
1785
1786
1786
1787
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1794
1795
1796
1796
1797
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1804
1805
1806
1806
1807
1807
1808
1808
1809
1809
1810
1811
1812
1813
1814
1814
1815
1816
1816
1817
1817
1818
1818
1819
1819
1820
1821
1822
1823
1824
1824
1825
1826
1826
1827
1827
1828
1828
1829
1829
1830
1831
1832
1833
1834
1834
1835
1836
1836
1837
1837
1838
1838
1839
1839
1840
1841
1842
1843
1844
1844
1845
1846
1846
1847
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1854
1855
1856
1856
1857
1857
1858
1858
1859
1859
1860
1861
1862
1863
1864
1864
1865
1866
1866
1867
1867
1868
1868
1869
1869
1870
1871
1872
1873
1874
1874
1875
1876
1876
1877
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1884
1885
1886
1886
1887
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1894
1895
1896
1896
1897
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1904
1905
1906
1906
1907
1907
1908
1908
1909
1909
1910
1911
1912
1913
1914
1914
1915
1916
1916
1917
1917
1918
1918
1919
1919
1920
1921
1922
1923
1924
1924
1925
1926
1926
1927
1927
1928
1928
1929
1929
1930
1931
1932
1933
1934
1934
1935
1936
1936
1937
1937
1938
1938
1939
1939
1940
1941
1942
1943
1944
1944
1945
1946
1946
1947
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1954
1955
1956
1956
1957
1957
1958
1958
1959
1959
1960
1961
1962
1963
1964
1964
1965
1966
1966
1967
1967
1968
1968
1969
1969
1970
1971
1972
1973
1974
1974
1975
1976
1976
1977
1977
1978
1978
1979
1979
1980
1981
1982
1983
1984
1984
1985
1986
1986
1987
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1994
1995
1996
1996
1997
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2004
2005
2006
2006
2007
2007
2008
2008
2009
2009
2010
2011
2012
2013
2014
2014
2015
2016
2016
2017
2017
2018
2018
2019
2019
2020
2021
2022
2023
2024
2024
2025
2026
2026
2027
2027
2028
2028
2029
2029
2030
2031
2032
2033
2034
2034
2035
2036
2036
2037
2037
2038
2038
2039
2039
2040
2041
2042
2043
2044
2044
2045
2046
2046
2047
2047
2048
2048
2049
2049
2050
2051
2052
2053
2054
2054
2055
2056

```



2 Octave and ...

- Octave and Octave-Forge
- Octave and Matlab
- Octave and C++
- Octave and Parallel Computing**
- PDEs in Octave-Forge



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←
15     end]) * us([1 end], ii));
16   us(2:end-1,ii) = fsolve (res, us(2:end-1,ii));
17
18 endfor
19 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 ←
11 end]) * u([1 end]));
12 u(2:end-1) = fsolve (res, u(2:end-1));
13 endfunction
14 tic ();
15 U0 = num2cell (randn (1, 200));
16 up = parcellfun (2, @poisson1d, U0, "UniformOutput", true, "←
VerboseLevel", 2);
17 toc ()
```



source code of the example



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



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



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



2 Octave and ...

- Octave and Octave-Forge
- Octave and Matlab
- Octave and C++
- Octave and Parallel Computing
- PDEs in Octave-Forge



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



How to get Octave



- Source on www.octave.org
- Binaries on octave.sourceforge.net
- Through package managers
- In NCLab