GNU Octave
A free high-level tool for Scientific Computing

Carlo de Falco, Jordi Gutiérrez Hermoso

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

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

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

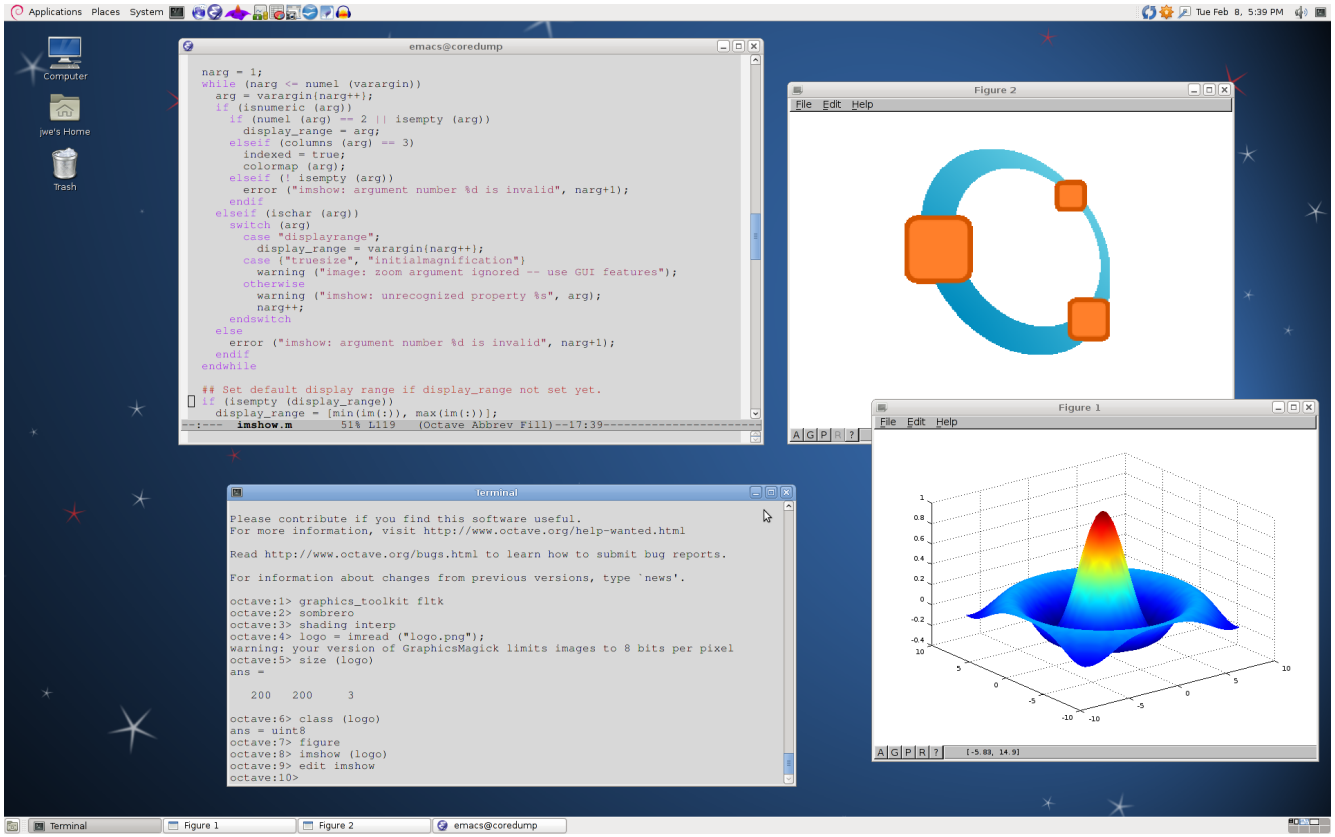

Figure : CLI screenshot

What does it look like

- Most requested feature: GUI!

What does it look like

- Most requested feature: GUI!

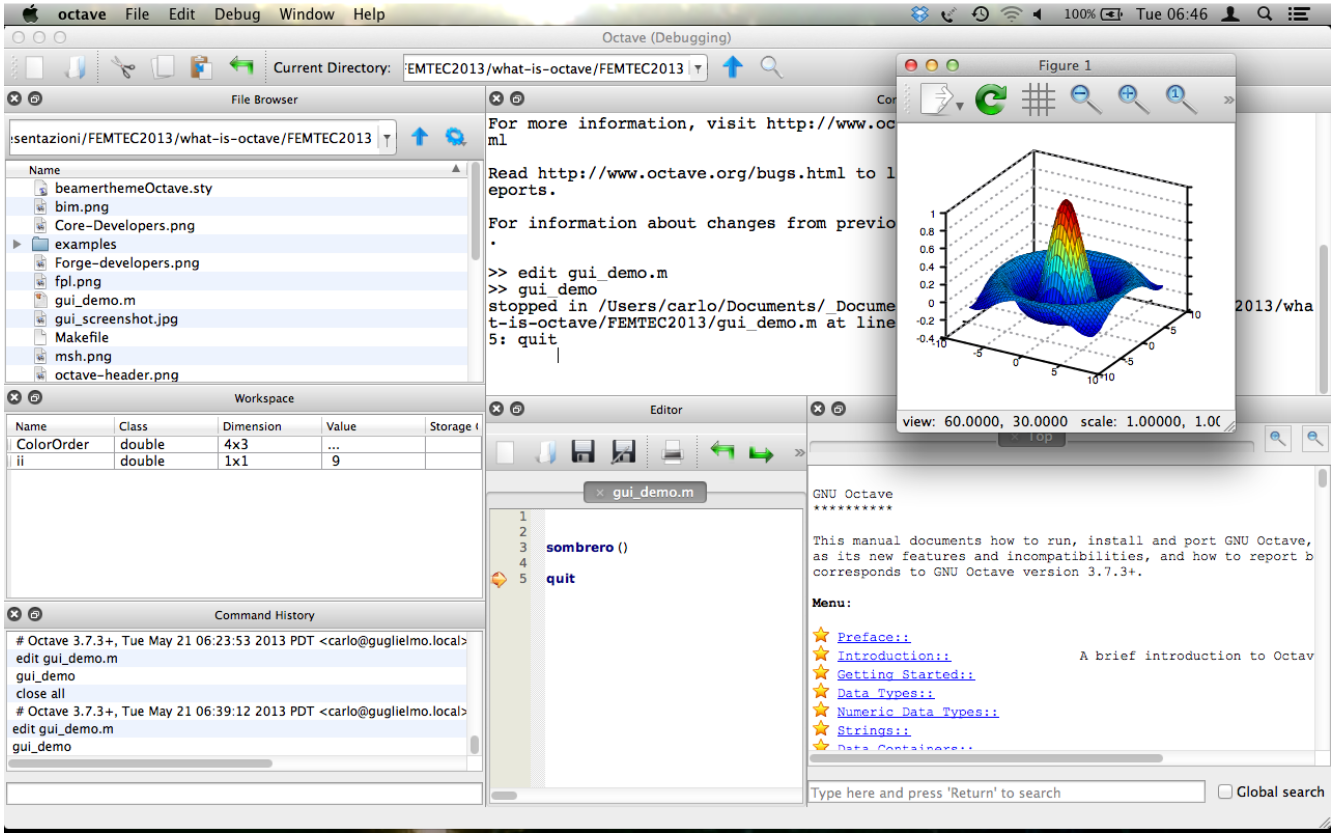

Figure : Qt based GUI Will ship with next release (4.0, expected 2013)

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>
	Jordi Gutiérrez Hermoso <jordigh>
	John W. Eaton <jwe>
	Kai Habel <kahacjde>
	Konstantinos Poulios <logari81>
	John Donoghue <lostbard>
	Mike Miller <mtmiller>
Squad	Octave Forge <octave-forge>
	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) Carnë Draug (carandraug) soren hauberg (hauberg) David Bateman (adb014) Add
Developer	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 (asnelt) Adam H Aitkenhead (adam78a) Olaf Till (i7tiol) JohanB (johanb88) Michael Goffioul (goffioul) Moreno Marzolla (mmarzolla) Jordi Gutiérrez Hermoso (jordigh) Philip Nienhuis (prnienhuis) lmarkov (lmarkov) Nir Krakauer (nir-krakauer) Juan Pablo Carbajal (jpicarbajal) Alois Schloegl (schloegl) paramaniac (paramaniac) John W. eaton (jwe)

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	Larrie Carr	Peter Ekberg	Kim Hansen	Lute Kamstra
Joel Andersson	David Castelow	Rolf Fabian	Søren Hauberg	Fotios Kasolis
Muthiah Annamalai	Vincent Cautaerts	Gunnar Farnebäck	Dave Hawthorne	Thomas Kasper
Marco Atzeri	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 Gallestey	Ryan Hinton	Geoffrey Knauth
Julien Bect	Martin Dalecki	Walter Gautschi	Roman Hodek	Heine Kolltveit
Roman Belov	Jorge Barros de Abreu	Klaus Gebhardt	A. Scottedward Hodel	Ken Kouno
Karl Berry	Carlo de Falco	Driss Ghaddab	Richard Allan Holcombe	Kacper Kowalik
David Billinghurst	Jacob Dawid	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 Diribarne	Brian Gough	Teemu Ikonen	Miroslaw Kwasniak
Marcus Brinkmann	Vivek Dogra	Steffen Groot	Alan W. Irwin	Rafael Laboissiere
Max Brister	John Donoghue	Etienne Grossmann	Geoff Jacobsen	Kai Labusch
Remy Bruno	David M. Doolin	David Grundberg	Mats Jansson	Claude Lacoursiere
Ansgar Burchard	Carnë Draug	Kyle Guinn	Cai Jianming	Walter Landry
Marco Caliari	Pascal A. Dupuis	Peter Gustafson	Steven G. Johnson	Bill Lash
Daniel Calvelo	John W. Eaton	Kai Habel	Heikki Junes	Dirk Laurie
John C. Campbell	Dirk Eddelbuettel	Patrick Haecker	Matthias Jäschke	Maurice LeBrun
Juan Pablo Carbajal	Pieter Eendebak	W,P.Y. Hadisoeseno	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-miin Lin	Kai P. Mueller	Tom Poage	Aleksej Saushev	John Swensen	Fook Fah Yap
Timo Lindfors	Hannes Müller	Orion Poplawski	Alois Schlögl	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	Jarno Rajahalme	Ludwig Schwardt	Kris Thielemans	
Massimo Lorenzin	Al Niessner	James B. Rawlings	Thomas L. Scofield	Olaf Till	
Emil Lucretiu	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 Reifenberger	Andriy Shinkarchuck	Utkarsh Upadhyay	
Rob Mahurin	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	E. Joshua Rigler	John Smith	Peter Van Wieren	
Axel MathÃ©i	Arno Onken	Petter Risholm	Julius Smith	James R. Van Zandt	
Makoto Matsumoto	V. Ortega-Clavero	Matthew W. Roberts	Shan G. Smith	Risto Vanhanen	
Tatsuro Matsuoka	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 Mamonov	Sylvain Pelissier	Joe Rothweiler	Christoph Spiel	Andreas Weber	
Christoph Mayer	Per Persson	Kevin Ruland	Richard Stallman	Olaf Weber	
J. Hoffimann Mendes	Primozz 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	Jonathan 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 Plesser	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

Outline

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

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
...
>>
```

Outline

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

```
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

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,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

```
1  $ mkoctfile --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 (), ↩
         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

```
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"↩
       );
```

```
1  oid load_myobject_type ()
2  {
3    myobject::register_type ();
4    myobject_type_loaded = true;
5    mlock ();
6    std::cout << "myobject_type_loaded" << std::endl;
7  }
```

 source code of the example (.cc)  source code of the example (.h)

Outline

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↩
         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

Outline

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