

# zedhat Workshop

**Alistair Boyle**

Carleton University, Ottawa, Canada  
EIT2019, July 1-3, 2019

# What is this workshop?

1. A bit of background
2. Trying by doing
  - install
  - try it out

# Why zedhat?

THE VISION IS TO PROVIDE A BUILDING BLOCK WITH  
MINIMAL DEPENDENCIES WHICH CAN BE USED TO  
DEVELOP SOLUTIONS THAT INCORPORATE EIT  
TECHNIQUES.

## zedhat is...

1. A preliminary release of zedhat,
2. an open source C library and
3. command line interface  
for EIT image reconstruction.

# zedhat is Open Source

## BSD Licensed, uses git

boyle / zedhat

Impedance Imaging (EIT and ERT) <https://zedhat.org>

178 commits 4 branches 3 releases 1 environment 1 contributor BSD 3-Clause

boyle: initial testing improvements

File	Commit Message	Age
ini	initial commit: general clean up of configure.ac; switch to auto-c11	last month
man	added man page for zedhat	3 years ago
etc	initial testing improvements	16 hours ago
tests	initial testing improvements	16 hours ago
appveyor.yml	initial windows build; various portability improvements	4 days ago
athparse	initial SVD Tikhonov inverse solver	5 days ago
NEWS.yml	initial testing improvements	16 hours ago
AUTHORS	basic project template: autotool, autotest, cmake, travis, style	3 years ago
LICENSE	initial commit	3 years ago
Makefile.am	initial to enable parameterized tests with corrected input file fo...	22 hours ago
NEWS	basic project template: autotool, autotest, cmake, travis, style	3 years ago
README.md	initial windows build; various portability improvements	4 days ago
TCCD	initial windows build; various portability improvements	4 days ago
src/lex	fix: refactor modify parser	last month
affocal.in	initial dummy commit for travis	26 days ago
autogen	complete cxx program on macOS 10.14.4	27 days ago
configure.ac	initial windows build; various portability improvements	4 days ago
env	error: fixed tests (setup/hardware) with masked malloc using g++ -Dl	last month
testsuite.at	initial testing improvements	16 hours ago

README.md

### zedhat 2

A tool for Electrical Impedance Tomography (EIT) and Electrical Resistivity Tomography (ERT). Current is applied to electrodes on the boundary and voltages are measured at other electrodes. From the measured voltages, we reconstruct an estimate of the conductivity  $\sigma$  over the interior.

<https://github.com/boyle/zedhat>

# BSD License

Copyright (c) 2016-2019, Alistair Boyle All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of zedhat nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



# zedhat is Cross Platform

- Linux
- Windows
- MacOS



# zedhat depends on

- BLAS
- LAPACKE
- CHOLMOD

# zedhat Preliminary Release

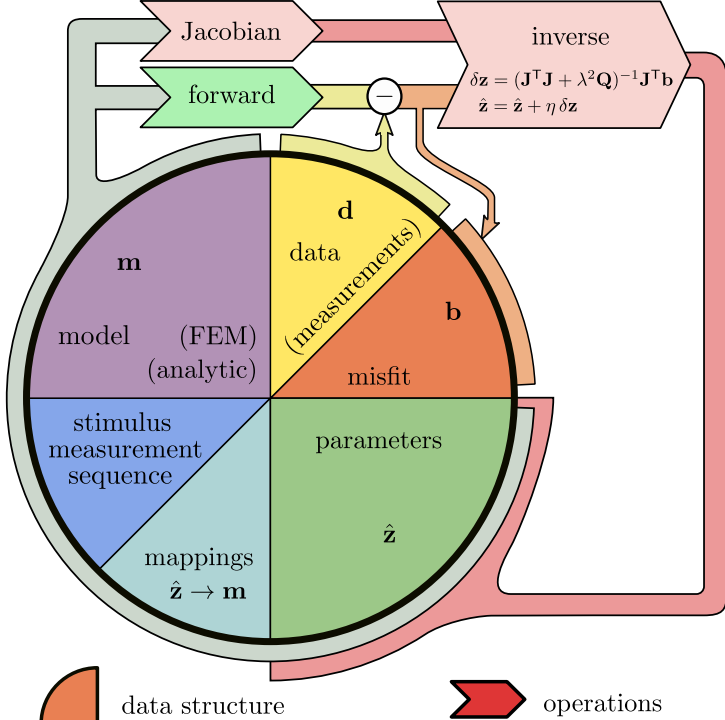
- **solvers**
  - **forward**  
linear elements, PEM/CEM, 2D and 3D
  - **inverse**  
single-step Gauss-Newton difference
- **command line interface**  
text input file format, Netgen meshes, .gz
- **continuous integration** all platforms
- **full test suite** 100% line coverage

## zedhat will be...

- be a linkable dynamic library
- have Python and Matlab bindings
- support a sample GUI
- support Gmsh meshes

# Quick Facts

Language	C	✓
Linkable Library	C/C++	
Language Bindings	Matlab Python	
Command Line Interface	text	✓
Graphical Interface	QT	
Supported File Formats	Netgen Gmsh	✓
Open Source	BSD	✓
Revision Controlled	git	✓
Continuous Integration	Linux	✓
<i>compiles on...</i>	MacOS	✓
	Windows	✓
Tested (Line Coverage)	100%	✓
Lines of Code	src	2761
	tests	6656
	build	141
Citations	0	...



# Text file format

## zedhat .zh format

zedhat

format

1

modeltype

forward

[Netgen mesh here]

# bcnr zc [ /m]

contactimpedances

1

1 1.0000000000000000e-03

# elementnr parameternr overlapfraction

parametermap

1

1 1 1.0000000000000000e+00

modeltype

reconstruction

[Netgen mesh here]

...

# A B M N amp\*gain

stimmeas

1

1 2 4 3 1.0000000000000000e+01

# \\_conductivity

hyperparameter

1

1.0000000000000001e-05

# conductivity [S/m]

parameters

1 3

1.0000e+00 9.9648e-01 -6.2833e-03

# measurements [V]

data

1 3

2.1912e+00 2.20959e+00 -5.4257e+01

# Text file format

## Netgen .vol meshes (inline)

```
dimension
2

geomtype
0

# surfnr   bcnr   domin  domout   np    p1    p2
surfacelements
3
1 0 1 0 2 111 112    <--- boundary
1 1 1 0 2 10 0      <--- PEM (non-standard Netgen)
1 2 1 0 2 114 115   <--- CEM

#          X          Y
points
145
  0.0000e+00  0.0000e+00

# matnr    np    p1    p2    p3
volumeelements
1
1 3 1 2 3
```

# zedhat Usage

--help

zedhat [options]

--help -h      this help

--version -V    version info

--forward-solver --fwd -f <fwd.zh>

    simulate measurements for model in fwd.zh

--inverse-solver --inv -f <inv.zh>

    solve for model parameters using model and data from inv.zh

--tolerance --tol -t <#.##e#>

    for checking --fwd and --inv solutions when available



# zedhat Usage

zedhat --fwd m1f.zh

```
-- loading from file --
reading ../zedhat/m1f.zh
zedhat
format 1
modeltype forward
dimension 2
geomtype 0
surfaceelements 36
points 145
volumeelements 256
contactimpedances 4
stimmeas 4
parametermap 477
modeltype reconstruction
dimension 2
geomtype 0
surfaceelements 20
points 41
volumeelements 64
contactimpedances 4
stimmeas 4
hyperparameter 1
parameters 64x3
data 4x3

-- forward solutions --
3 parameter frames
frame#1
meas#      calculated      from file
1          2.1912224      2.1912224
2          2.1912224      2.1912224
3          2.1912224      2.1912224
4          2.1912224      2.1912224
RMSE = 1.00191e-14
tolerance = inf
frame#2
meas#      calculated      from file
1          2.2095986      2.2095986
2          2.1973556      2.1973556
3          2.2095986      2.2095986
4          2.1973556      2.1973556
RMSE = 5.0341e-15
tolerance = inf
frame#3
meas#      calculated      from file
1          -51.152568      -51.152568
2          -1115.1087      -1115.1087
3          -51.152568      -51.152568
4          -1115.1087      -1115.1087
RMSE = 2.17333e-12
tolerance = inf

-- completed --
```

# zedhat Usage

zedhat --inv m1i.zh

```
-- loading from file --
reading ../zedhat/m1i.zh
zedhat
format 1
modeltype forward
dimension 2
geomtype 0
surfaceelements 36
points 145
volumeelements 256
contactimpedances 4
stimmeas 4
parametermap 477
modeltype reconstruction
dimension 2
geomtype 0
surfaceelements 20
points 41
volumeelements 64
contactimpedances 4
stimmeas 4
hyperparameter 1
parameters 64x3
data 4x3
```

```
-- inverse solutions --
```

```
2 measurement frames
```

```
model: 0 electrodes, 4 measurements, = 1e-05
```

```
3 parameter frames (64 parameters/frame)
```

```
3 measurement frames (4 measurements/frame)
```

```
forward model: 2D, 145 nodes, 256 elements, 36 surfaces, 477 entry p
```

```
reconstruction model: 2D, 41 nodes, 64 elements, 20 surfaces, 0 entr
```

```
frame#1
```

param#	calculated	from file
1	-6.28336459e-03	-6.28336594e-03
2	-6.19041309e-03	-6.19041139e-03
3	-6.28336459e-03	-6.28336552e-03
4	-6.19041309e-03	-6.19041282e-03
5	-1.00941504e-02	-1.00941514e-02
6	-9.28324133e-03	-9.28324206e-03
7	-1.00941504e-02	-1.00941520e-02

```
....
```

59	-1.42871004e-03	-1.42871011e-03
60	-4.12415678e-03	-4.12416017e-03
61	1.45142989e-03	1.45143240e-03
62	1.06213311e-03	1.06213291e-03
63	1.06213311e-03	1.06213116e-03
64	1.45142989e-03	1.45143737e-03

```
RMSE = 2.02372e-09
```

```
-- completed --
```

# Build your own .zh files

- **from EIDORS**  
zedhat() in dev/a\_boyle/zedhat.m  
see help for usage  
(input/output text format .zh)
- **roll your own**  
Netgen .vol meshes + data/parameters

# Lets try it!

- <https://github.com/boyle/zedhat>
- **follow instructions for installing**  
on your platform
- **try the forward or inverse solver**  
example .zh files are in tests/ directory