Good Practices for Mathematical Software

Sou-Cheng (Terrya) Choi
NORC at UChicago, IIT  www.iit.edu/~schoi32

Meshfree Methods Seminar
Dept. of Applied Mathematics, IIT

Thanks: GAIL Team, MATH573 (Fall 2013)

May 21, 2014
Memorable Memorial Day!

All gave some
Some gave all
More do some
Some do more
The GAIL Company

- **CEO:** Fred Hickernell
- **Engineering Manager/Mislead:**
- **Release Lead:** Yuhan Ding
- **GAIL sites master:** Lan Jiang
- **Repository Specialist:** Yizhi Zhang
- **Documentation Lead:** Xuan Zhou
- **Test Lead:** Tony Jiménez Rugama
- **Alumni:** Xincheng Sheng
About GAIL—words from the CEO, May 6, 2013

- Before GAIL: Automatic numerical integration algorithms have inherent flaws in their error estimation based on balls of integrands.
- GAIL overcomes the flaws by considering cones of integrands. This allows us to construct upper bounds on costs of our integration routines with rigorous guarantees of accuracy and develop algorithms that provide the value of the integral with an error of no more than the user-defined tolerance.
- Mission (possible): To create a well-documented and well-tested library of univariate & multivariate integration routines that have rigorous guarantees.
- GAIL version 1: By the end of the summer we hope to have our automatic routines for function recovery, univariate integration, and Monte Carlo estimation of mean on the GAIL site in good form, meaning that these routines need to be
  - well-documented
  - well-tested
  - optimized for speed
  - accompanied by examples
  - in a repository where they can be modified and re-tested as needed
- Later we will improve and add to these routines.
### DONE

<table>
<thead>
<tr>
<th>Owner</th>
<th>Description</th>
<th>Priority</th>
<th>Due Date</th>
<th>Resolution notes</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lan</td>
<td>Add &quot;cubMC_test\un\put&quot; to <a href="http://bit.ly/1tjOe8q">http://bit.ly/1tjOe8q</a></td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>Sou-Cheng, I added my put file to the page you provided, please check if it is the way you want</td>
<td>✔️</td>
</tr>
<tr>
<td>Lan</td>
<td>Add doctest test for cubMC_g on nunit hypercube if none exist yet</td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Lan</td>
<td>cubMC_g workout test functions for $n=3,\ldots,7$</td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Lan</td>
<td>Add doctest test for cubMC_g with measure &quot;normal&quot; if none exist yet</td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Lan</td>
<td>Add phone number(s) on <a href="http://bit.ly/1ISxXli">http://bit.ly/1ISxXli</a></td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Lan</td>
<td>Which google drive is GAIL 1.3 release zip hosted?</td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Tony</td>
<td>Add phone number(s) on <a href="http://bit.ly/1ISxXli">http://bit.ly/1ISxXli</a></td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Xuan</td>
<td>Add phone number(s) on <a href="http://bit.ly/1ISxXli">http://bit.ly/1ISxXli</a></td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Yizhi</td>
<td>Add phone number(s) on <a href="http://bit.ly/1ISxXli">http://bit.ly/1ISxXli</a></td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Yuhan</td>
<td>Unit tests for funapp_g extend x to more than 1 point.</td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Yuhan</td>
<td>Add phone number(s) on <a href="http://bit.ly/1ISxXli">http://bit.ly/1ISxXli</a></td>
<td>Medium</td>
<td>March 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>

### TODO

<table>
<thead>
<tr>
<th>Owner</th>
<th>Description</th>
<th>Priority</th>
<th>Due Date</th>
<th>Resolution notes</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>GAIL version 2.0 release</td>
<td>High</td>
<td>September 1, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Fred, Tony, Lan, and Xuan</td>
<td>Prepare MCMC conference proceedings papers</td>
<td>Medium</td>
<td>August 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>REU student</td>
<td>Alpha version of integral_open_g (based on open integration rule)</td>
<td>Medium</td>
<td>June 30, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Sou-Cheng</td>
<td>Draft GAIL 2.0 for ACM TOMS</td>
<td>Medium</td>
<td>June 30, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Xuan</td>
<td>Alpha version of funapp_kernel_g</td>
<td>Medium</td>
<td>June 16, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>ALL</td>
<td>Register for SIAM Annual Meeting</td>
<td>Medium</td>
<td>June 9, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Fred &amp; Tony</td>
<td>Alpha version of cubMC_g</td>
<td>Medium</td>
<td>May 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Lan</td>
<td>Read Tempore's paper and give us a report in the seminar</td>
<td>Medium</td>
<td>May 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Lan</td>
<td>Alpha version of mean_IMC_g</td>
<td>Medium</td>
<td>May 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Lan</td>
<td>Alpha version of meanMC_g with relative error</td>
<td>Medium</td>
<td>May 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Tony</td>
<td>Alpha version of cul_calc_g</td>
<td>Medium</td>
<td>May 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Yizhi</td>
<td>Alpha version of integral_g with Simpson's rule</td>
<td>Medium</td>
<td>May 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Yuhan</td>
<td>Make funapp_g return piecewise polynomial form that can be evaluated using polyv</td>
<td>Medium</td>
<td>May 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Yuhan</td>
<td>Alpha version of local adaptive funapp_g</td>
<td>Medium</td>
<td>May 31, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Sou-Cheng</td>
<td>Check the documentation help in funapp_g.m</td>
<td>Medium</td>
<td>May 24, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>ALL</td>
<td>GAIL Meeting May 21, 4:00pm-6:00pm @E1-112</td>
<td>Medium</td>
<td>May 21, 2014</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Sou-Cheng</td>
<td>Meshfree talk on Good Practices for Mathematical Software</td>
<td>Medium</td>
<td>May 21, 2014</td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>
GAIL Milestones & Targets

- **First-year milestones**
  - Sep 3, 2013: Release of GAIL version 1
  - Sep 4, 2013: Development of GAIL version 1.3 commenced
  - Fall 2013: MATH 573 Seminar/Elective “Reliable Mathematical Software” (Instructors: Fred Hickernell & C. Students: 7 registered, 2 regular sit-in)
  - Feb 14, 2014: Release of GAIL version 1.3
  - Feb 15: Development of GAIL version 2.0 commenced

- **Targets for the next few years**
  - July 9–10: SIAM Annual Meeting, Minisymposium on “Reliable Mathematical Software”
  - **Labor Day**, Sep 1: Release of GAIL version 2
  - Feb 2015: Release of GAIL version 2.5
  - Sep 2015: Release of GAIL version 3
  - TBD: Apply for a research grant for GAIL
# GAIL algorithms at a glance

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Developers</th>
<th>GAIL functions</th>
<th>Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function recovery</td>
<td>Yuhan</td>
<td>funappx_g</td>
<td>1.0–2.0</td>
</tr>
<tr>
<td>Univariate integration</td>
<td>Yizhi</td>
<td>integral_g</td>
<td>1.0–2.0</td>
</tr>
<tr>
<td>Monte Carlo mean estimation</td>
<td>Lan</td>
<td>meanMC_g</td>
<td>1.0–2.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>meanMCBinomial_g</td>
<td>2.0</td>
</tr>
<tr>
<td>Multivariate integration</td>
<td>Lan</td>
<td>cubMC_g</td>
<td>1.3–2.0</td>
</tr>
<tr>
<td>Univariate optimization</td>
<td>Xin</td>
<td>funmin_g</td>
<td>2.0</td>
</tr>
<tr>
<td>Multivariate function recovery</td>
<td>Xuan</td>
<td>funappxmulti_g</td>
<td>2.0</td>
</tr>
<tr>
<td>Quasi Monte Carlo</td>
<td>Tony</td>
<td>cubQMC_g</td>
<td>2.0</td>
</tr>
<tr>
<td>Multi-level Monte Carlo</td>
<td>Aleks</td>
<td>cubQMLMC_g</td>
<td>2.0</td>
</tr>
</tbody>
</table>
A “disintegrating” integral

Spiky \( f \) with \( I = \int_0^1 f(x) \, dx \approx 0.3694 \).
\[
\text{quad}(f, [0, 1], 1e-14) = 0 \text{ giving error } = I.
\]

Strategy: ↑ number of points to ↓ error

Q: How many number of points (or function evaluations), \( n \), are required in a quadrature rule to guarantee that a given error tolerance, \( \epsilon \) is met with a confidence level \( 1 - \alpha \)?

A: Clancy et al. 2013, [CDH\( ^+ \)14b], Hickernell et al. 2014 [HJLO14], GAIL 1.0 [CDH\( ^+ \)13], GAIL 1.3 [CDH\( ^+ \)14a]
Reproducible research (RR) pioneers and champions

Jon Claerbout
SEP (Stanford Exploration Project, 1973–present)

David Donoho
What is reproducible research (RR)?

- Claerbout, “The markings [ER], [CR], and [NR] are promises by the author(s) about the reproducibility of each figure result.” (url: j.mp/VM7Xq4)
  - ER: Easily reproducible: “programs, parameters, and makefiles ... data”
  - CR: Conditionally reproducible: “processing requires 20 minutes or more, or commercial packages”
  - NR: Not reproducible: drawings
  - M: Movie “in a figure”; ER, CR, or NR

- Donoho paraphrasing Claerbout, “an article about computational result is advertising, not scholarship. The actual scholarship is the full software environment, code and data, that produced the result.” (Buckheit & Donoho 1995)
Examples of RR—ER

Lomask and Fomel 2006

Figure 1: The dipping planes synthetic model. Although a trivial flattening test case, the boundaries of the divergence of the dip are not periodic.

[jesse3-plane3D] [ER]
Bernd Flemisch’s survey results (Aug 2013)

Online Survey: Reproducibility in Computational Science and Engineering (CSE)

- 13 questions on opinions and experiences concerning the reproducibility of computational results.
- Results collected on August 1.

Direct Emails
- Call went out on July 5 to ~ 500 addresses.
- Resulted in ~ 80 answers.

InterPore Newsletter
- Newsletter was sent out on July 6 to ~ 1000 addresses.
- Resulted in 2 answers.

SIAM Activity Group on CSE Mailing List
- Call went out on July 5 to ~ 2000 addresses.
- Resulted in ~ 300 answers.

Survey Results I (n = 385)

- I understand what the reproducibility of computational results means:
  - No
  - Yes
  - 0% 20% 40% 60% 80%

Survey Results II (n = 385)

- My strategy to ensure the reproducibility of my comp. results:
  - Use of tools like Madagascar
  - I make the source code available
  - I make the problem data available
  - Use of version control
  - Detailed description in my papers
  - I don’t have/need a strategy
  - 0% 20% 40% 60% 80%

- My current education/position:
  - PhD student
  - BSc/MSc student
  - Postdoc
  - Professor
  - 0% 10% 20% 30%

- My age in years:
  - ≤ 20
  - 21 – 30
  - 31 – 40
  - 41 – 50
  - 51 – 60
  - > 60
  - 0% 10% 20% 30%

Survey Results: A Slightly Deeper Look

- The estimation of the effort to reproduce does not influence the estimation of the importance of reproducibility.
- The effort estimated for oneself influences the effort estimated for others, and the effort for the others is considered to be higher.
- The estimated effort to reproduce does not influence the number of employed strategy items.
- The amount of work related to coding influences the estimation of the importance, but not the number of employed strategy items.
- Age does not have an influence on the quantitative results, apart from the time devoted to coding.
How reliable or limiting is RR?

- Bounded above by the underlying theory, data, code, and software
- Platform and version dependent
- Less reliable over time due to new software versions
- Lack or loss of data or code; confidential/sensitive data
- Big data inputs hard to clean; outputs corruptable by sharing or transferring processes
- Commercial software expensive with many restrictions—Cf. quality free software
- Big (binary) code of poor design or documentation, slow to run
- Lack of testing
- No bug-fix patches or slow new releases
- Lack of communication, community support & feedback
- Hamper creativity and/or productivity?
Reliable RR via Staunch Scientific Software (SSS)

SSS heavyweights

Richard Stallman
GNU, Free software movement

Ian Foster
Father of the Grid, Globus Online, Galaxy, SWIFT, CIM-EARTH
**Scientific research vs. Software engineering**

**Sciences:**
- Wavelets
- Signal processing
- Image processing
- Biostatistics
- PDEs
- Economics
- Physics
- Chemistry
- Mathematics
- Algorithms
- Monte Carlo simulation
- Numerical integration

**Software:**
- Test-driven development
- Object oriented design
- API and GUI
- Software reuse
- Logging, error handling
- Paired programming
- Nightly build
- Client-server functionality tests
- Load testing
- Documentation
- Continuous release
- Research project websites
- Licenses (BSD), copyleft
GAIL team working with a remote, central repository

Lan initialized GAIL-dev repository

Everybody cloned the repository

Yizhi worked on his local repository

Yuhan worked on her local repository

Yizhi published his code

Yuhan pulled Yizhi's commits

Yuhan resolved merge conflicts

Yuhan published her code

Pull before push

Image credit: https://www.atlassian.com/git/workflows

Sou-Cheng Terrya Choi schoi32@iit.edu

Good Practices for Mathematical Software

May 21, 2014
Interacting with our GAIL development repository

Use SourceTree

```
Commit: ad01c73
Author: Lan Jiang <tojianglan@gmail.com>
Date: May 1, 2014 at 5:13:01 PM CDT
labels: HEAD origin/master origin/HEAD master
release steps instruction

1. Follow the instructions to prepare a new version release:
2. Make sure all the contents.m are correct, please note: you need
   "/include". Zero 2. Make sure the contents.m contains the list for both directories.
```

Sou-Cheng Terrya Choi schoi32@iit.edu

Good Practices for Mathematical Software

May 21, 2014 17 / 33
Managing a repository

Avoid checking in binary files

"Cleanliness is next to godliness"

Image source: http://bit.ly/1i4p7by

Credits: Xuan
GAIL Central, http://code.google.com/p/gail/

Guaranteed Automatic Integration Library

About Guaranteed Automatic Integration Library (GAIL)

GAIL is a suite of algorithms for integration problems in one, many, and infinite dimensions, and whose answers are guaranteed to be correct.

GAIL is created, developed, and maintained by Fred Hickernell (Illinois Institute of Technology), Sou-Cheng Choi (University of Chicago, Argonne National Laboratory, and IIT), and their collaborators including Yuhan Ding (IIT), Lan Jiang (IIT), and Yish Zhang (IIT).

News
- We plan to release GAIL version 2.0 in Fall 2014.
- GAIL version 1.3 is released on February 14, 2014 (Valentine’s Day, for our love of the package).
- GAIL version 1.0 is released on September 3, 2013.

If you find GAIL helpful in your work, please support us by citing the following papers and software.

Free Software
- Sou-Cheng T. Choi, Yuhan Ding, Fred J. Hickernell, Lan Jiang, and Yish Zhang, GAIL: Guaranteed Automatic Integration Library (Version 1.3), MATLAB Software, 2014. (Download zip, or download and run MATLAB script, or clone repository. Documentation. BibTeX.)
- Sou-Cheng T. Choi, Yuhan Ding, Fred J. Hickernell, Lan Jiang, and Yish Zhang, GAIL: Guaranteed Automatic Integration Library (Version 1.2), MATLAB Software, 2013. (Download zip, or download and run MATLAB script, or clone repository. Documentation. BibTeX.)

Papers and Reports

Credits: Lan

Sou-Cheng Terrya Choi schoi32@iit.edu

Good Practices for Mathematical Software

Introduction
After Math
Practices
Last Words
Repository
Installation
APIs, input parsing, messages
Documentation
Tests

May 21, 2014 19 / 33
GAIL home directory & help

One step to install:

```bash
>> DownloadInstallGail_1_3_0
The GAIL package is now being downloaded...
```

```bash
>> help GAIL_1_3_0
GAIL_MATLAB

Files
- GAILstart - Initialize all the GAIL paths and system parameters.
- GAIL_Install - Install GAIL. Add GAIL paths to MATLAB search path.
- GAIL_Reinstall - Reinstall GAIL. Remove existing GAIL paths and add new ones.
- GAIL_Uninstall - Uninstall GAIL. Remove GAIL paths from MATLAB search path
- LICENSE - License of GAIL
- README - Installation and introduction to GAIL.

Folders
- Algorithms - GAIL algorithms
- Documentation - GAIL documentation
- OutputFiles - Output generated by GAIL routines
- Papers - Papers and slides related to GAIL
- ThirdParty - Open-source tools used but not produced by the GAIL team
- UnitTests - Unit tests of GAIL algorithms
- Utilities - Tools for the GAIL package
- Workouts - Workouts of GAIL algorithms
```
GAIL’s Application Programming Interfaces (APIs)

Our key algorithms have three API patterns:

\[
\begin{align*}
[x, \ out\_param] &= \text{algo}_g(f, \ in\_param); \\
[x, \ out\_param] &= \text{algo}_g(f, \ inputVal_1, \ldots, \ inputVal_n); \\
[x, \ out\_param] &= \text{algo}_g(f, \ 'input1', \ inputVal_1, \ldots, \ 'inputn', \ inputVal_n)
\end{align*}
\]

- \(f\): compulsory, function handle
- \(in\_param, \ out\_param\): optional, structures
- \(input1, \ldots, inputn\): optional, string
- \(inputVal_1, \ldots, inputVal_n\): optional, numeric

Automatically correct out-of-range input values with warning or error messages to users.

Credits: Fred
GAIL help function

>> help funappx_g
FUNAPPX_G 1-D guaranteed function recovery on closed interval [a,b]

funappx = FUNAPPX_G(f) recovers function f on the default interval [0,1]
by a piecewise linear interpolant funappx to within the guaranteed
absolute error tolerance of 1e-6. Default initial number of points is
100 and default cost budget is 1e7. Input f is a function handle. The
statement y=f(x) should accept a vector argument x and return a vector
y of function values that is the same size as x.

funappx = FUNAPPX_G(f,a,b,abstol,nlo,nhi,nmax) for given function f and
the ordered input parameters that define the finite interval [a, b], a
guaranteed absolute error tolerance bstol, lower bound of initial
number of points nlo, upper bound of initial number of points nhi, and
cost budget nmax. nlo and nhi can be input as a vector or just one
value as an initial number of points.

funappx = FUNAPPX_G(f,'a',a,'b',b,'abstol',abstol,'nlo',nlo,'nhi',nhi,'nmax',nmax)
recovers function f on the finite interval [a, b], guaranteed absolute
error tolerance abstol, lower bound of initial number of points nlo,
upper bound of initial number of points nhi, and cost budget nmax. All
six field-value pairs are optional and can be supplied in different
order.

funappx = FUNAPPX_G(f,in_param) recovers function f on the finite
interval [in_param.a, in_param.b], guaranteed absolute error tolerance
in_param.abstol, lower bound of initial number of points in_param.nlo,
upper bound of initial number of points in_param.nhi, and cost budget
in_param.nmax. If a field is not specified, the default value is used.

in_param.a --- left end point of interval, default value is 0
GAIL’s searchable HTML documentation

Credits: Yuhan
GAIL doctest for documentation examples

Example 1

```matlab
>> doctest funappx_g
TAP version 13
1..9
  ok 1 - "f = @(x) x.^2; [fappx, out_param] = funappx_g(f)"
  ok 2 - "f = @(x) x.^2;"
  ok 3 - "[fappx, out_param] = funappx_g(f,-2,2,1e-7,10,10,1000000)"
  ok 4 - "f = @(x) x.^2;"
  ok 5 - "[fappx, out_param] = funappx_g(f,'a',-2,'b',2,'nhi',100,'nlo',10)"
  ok 6 - "clear in_param; in_param.a = -10; in_param.b = 10; "
  ok 7 - "in_param.abstol = 10^(-7); in_param.nlo = 10; in_param.nhi = 100;"
  ok 8 - "in_param.nmax = 10^6; f = @(x) x.^2;"
  ok 9 - "[fappx, out_param] = funappx_g(f,in_param)"
```

Example 2

```matlab
[fappx, out_param] = funappx_g(@(x) x.*2,0,100,1e-7,10,1000000,1e0)
* Approximate function x^2 on [0,100] with error tolerance 1e-7, cost
  budget 10000000 and initial number of points 100

fappx = @(x)interpl(xl,yl,x,'linear')
out_param =
  f: @(x)X.*2
  a: 0
  b: 100
  abstol: 1.00000e-07
  nlo: 10000
  nhi: 10000000
  nmax: 100000000
  ninit: 956
  nstar: 954
  exceedbudget: 0
  npoints: 976021
  errorbound: 2.6944e-11
```
MATLAB unit tests

Every time the code is changed, unit tests are run.
Every time a bug is found, unit tests are expanded.

```matlab
>> results = run(ut_funappx_g)
Running ut_funappx_g
....
Done ut_funappx_g

results =
1x4 TestResult array with properties:

    Name     Passed     Failed     Incomplete     Duration
    Total: 4 Passed, 0 Failed, 0 Incomplete.
    0.10896 seconds testing time.

classdef ut_funappx_g < matlab.unittest.TestCase
    methods(Test)
    function funappx_g0fx(testCase)
        f = @(x) x;
        in_param.tol = 10^(-8);
        in_param.tau = 15;
        in_param.Nmax = 10^6;
        [appxf, result] = funappx_g(f,in_param);
        x = sqrt(2)-1;
        actualerr = abs(appxf(x)-f(x));
        testCase.verifyLessThanOrEqual...
        (actualerr,in_param.tol);
        testCase.verifyLessThanOrEqual...
        (result.npoint,in_param.Nmax);
    end
```

(More test cases, omitted)
>> runtests
TAP version 13
1..9
ok 1 - "f = @(x) x.^2; [fappx, out_param] = funappx_g(f)"
ok 2 - "f = @(x) x.^2;"
ok 3 - "[fappx, out_param] = funappx_g(f,-2,2,1e-7,10,10,1000000)"
ok 4 - "f = @(x) x.^2;"
ok 5 - "[fappx, out_param] = funappx_g(f,’a’,-2,’b’,2,’nhi’,100,’nlo’,10)"
ok 6 - "clear in_param; in_param.a = -10; in_param.b = 10; "
ok 7 - "in_param.abstol = 10^(-7); in_param.nlo = 10; in_param.nhi = 100;"
ok 8 - "in_param.nmax = 10^6; f = @(x) x.^2;"
ok 9 - "[fappx, out_param] = funappx_g(f,in_param)"
...
ok 1 - "f=@(x) sin(x);interval = [1;2];"
ok 2 - "Q = cubMC_g(f,interval,’uniform’,1e-3)"
ok 3 - "f=@(x) exp(-x(:,1).^2-x(:,2).^2);hyperbox = [0 0;1 1];"
ok 4 - "Q = cubMC_g(f,hyperbox,’uniform’,1e-3)"
ok 5 - "d=3;f=@(x) 2^d*prod(x,2)+0.555;hyperbox = [zeros(1,d);ones(1,d)];"
ok 6 - "Q = cubMC_g(f,hyperbox,’uniform’,1e-3)"
  0.4842
Measuring and improving performance

MATLAB software development/analysis tools

▶ Matlab Profiler
▶ Matlab Mex
▶ Matlab 2013 Unit Testing Framework
▶ Matlab Central (url: j.mp/anwdaP)
▶ Matlab mex to interface with C/Fortran functions
▶ Parallelization with SWIFT
▶ Matlab reports
Conclusions

1. RR is a philosophy that also serves as a practical guide for producing quality scholarly publications in computational sciences
2. SSS involves a number of software engineering principles and tools that enable experiments and actions. Open source plays a key role
3. RR is made more reliable by SSS development
4. Numerical quadrature with a guarantee of accuracy is new and worthy
5. GAIL is to demonstrate the theorems and make an impact
6. Eventually and ideally, to be introduced into first NA/SC courses
7. Handling and producing data/code/software taken to the extreme are challenging research and practical problems
8. HPC, mobile apps, industrial employment
9. Please cite high-quality software
10. Think a wedge (cone), not a whole (ball)
Wish list

Would you be our Santa?

- User guide
- Stress tests
- Real-world applications
- Automation of HTML help
- Automation of test suites
- Graphical user interfaces to guide user inputs
- Algorithmic usage measurements and citation reminders
- Publish GAIL 2.0 with ACM TOMS
Readings

About GAIL: [CDH$^{+}$14b, HJLO14, CDH$^{+}$14a]

About reliable reproducible research via staunch / sustainable scientific software: [CH13, Cho13, KCL$^{+}$14]
“God does not care about our mathematical difficulties. He integrates empirically.” —Albert Einstein

Thank you!
References I


Sou-Cheng Terrya Choi schoi32@iit.edu

Good Practices for Mathematical Software

May 21, 2014 32 / 33
References II
