Framework

Workflow

CUTEst.jl

Practical example

Future work

◆□▶ ◆□▶ ◆□▶ ◆□▶ □ のへ⊙

A Workflow for Designing Optimization Methods in the Julia Language

Optimization Days 2016

Abel Soares Siqueira

Mathematics Department, Federal University of Paraná, Curitiba/PR, Brazil

Dominique Orban

Mathematics and Industrial Engineering Department, École Polytechnique de Montréal

May, 4th

Workflow	Framework	CUTEst.jl	Practical example	Future work

Workflow

- Test Driven Development
- Problems as blocks
- Outline

2 Framework

- Julia
- Optimize.jl
- NLPModels.jl

3 CUTEst.jl

- CUTEst
- CUTEst.jl
- Flavors
- Some functions
- Practical example
 - TRON
 - Make it work
 - Make it right
- 5 Future work
 - Future work

◆ロ → ◆帰 → ◆ 国 → ◆ 国 → ● ● ● ● ● ● ● ●

Workflow	Framework	CUTEst.jl	Practical example	Future work

1 Workflow

- Test Driven Development
- Problems as blocks
- Outline

2 Framework

- Julia
- Optimize.jl
- NLPModels.jl

3 CUTEst.jl

CUTEst

- CUTEst.jl
- Flavors
- Some functions
- Practical example
 - TRON
 - Make it work
 - Make it right
- 5 Future work
 - Future work

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

Workflow	Framework	CUTEst.jl	Practical example	Future work

"... We should forget about the small efficiencies, say about 97% of the time: **premature optimization is the root of all evil**..." Donald Knuth

"The strategy is definitely: first **make it work**, then **make it right**, and, finally, **make it fast**."

Kent Beck

Workflow	Framework	CUTEst.jl	Practical example	Future work		
0000						
Test Driven Development						

Test Driven Development



Workflow	Framework	CUTEst.jl	Practical example	Future work
0●000 Problems as blocks	00000	0000000	000000000000	

Problems as blocks

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

- Simplest representative problems;
- Classes of problems;
- Selection of problems from some specific repository.

Workflow	Framework	CUTEst.jl	Practical example	Future work
0000				
Problems as blocks				

Test nonzero exit flag

- Infinite loop;
- Budget limitations;
- Domain error.

Workflow	Framework	CUTEst.jl	Practical example	Future work
	00000	00000000	000000000000	
Problems as blocks				

Scale the problem

- Dense matrices;
- Inplace operations;
- Small efficiencies.

Workflow	Framework	CUTEst.jl	Practical example	Future work
00000				
Outline				

Use a test library (FactCheck.jl);



Workflow	Framework	CUTEst.jl	Practical example	Future work
00000				
Outline				

- Use a test library (FactCheck.jl);
- Write small problem;
 - Write simplest code solving it (theory \checkmark , efficiency $\cancel{\times}$);
 - 8 Repeat

Workflow	Framework	CUTEst.jl	Practical example	Future work
00000				
Outline				

- Use a test library (FactCheck.jl);
- Write small problem;
 - O Write simplest code solving it (theory ✓, efficiency ✗);
 - 8 Repeat
- Solution Write limitation tests (time, iteration, etc.) and code for it;

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

Workflow	Framework	CUTEst.jl	Practical example	Future work
00000				
Outline				

- Use a test library (FactCheck.jl);
- Write small problem;
 - O Write simplest code solving it (theory ✓, efficiency ✗);
 - 8 Repeat
- Write limitation tests (time, iteration, etc.) and code for it;
- • Write class os problems (hundreds of runs, randomized, larger size);

イロン 不良 アイヨン トヨー ろくの

- Write code solving it;
- 8 Repeat

Workflow	Framework	CUTEst.jl	Practical example	Future work
00000				
Outline				

- Use a test library (FactCheck.jl);
- Write small problem;
 - O Write simplest code solving it (theory ✓, efficiency ✗);
 - 8 Repeat
- Solution Write limitation tests (time, iteration, etc.) and code for it;
- • Write class os problems (hundreds of runs, randomized, larger size);

イロン 不良 アイヨン トヨー ろくの

- Write code solving it;
- 8 Repeat
- Choose problems from specific repository;
 - Try to solve it;
 - 8 Repeat

Workflow	Framework	CUTEst.jl	Practical example	Future work
00000				
Outline				

- Use a test library (FactCheck.jl);
- Write small problem;
 - O Write simplest code solving it (theory ✓, efficiency ✗);
 - 8 Repeat
- Solution Write limitation tests (time, iteration, etc.) and code for it;
- • Write class os problems (hundreds of runs, randomized, larger size);

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ □ ● ● ●

- Write code solving it;
- 8 Repeat
- Choose problems from specific repository;
 - Try to solve it;
 - S Repeat
- Improve the code.

Workflow	Framework	CUTEst.jl	Practical example	Future work

Workflow

- Test Driven Development
- Problems as blocks
- Outline
- 2 Framework
 - Julia
 - Optimize.jl
 - NLPModels.jl

3 CUTEst.jl

- CUTEst
- CUTEst.jl
- Flavors
- Some functions
- Practical example
 - TRON
 - Make it work
 - Make it right
- 5 Future work
 - Future work

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 三臣 - のへで

Workflow	Framework	CUTEst.jl	Practical example	Future work
00000 Julia	• 0 000	0000000	000000000000	00

Julia

- High level, High performance;
- Open source, multiplatform;
- Great C/Fortran interface;
- Easy syntax;
- Good practices (git, automated testing, code coverage);

Workflow	Framework	CUTEst.jl	Practical example	Future work
	0000			
Julia				

Framework

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

- Develop with Optimize.jl
- Create the tests with NLPModels.jl;
- Easily access CUTEst with CUTEst.jl.

Workflow	Framework	CUTEst.jl	Practical example	Future work
	00000			
Optimize.jl				

Optimize.jl

- Methods;
- Auxiliary algorithms and tools (trust region, line search, etc.);

• Testing and benchmarking.

Workflow	Framework	CUTEst.jl	Practical example	Future work
	00000			
NLPModels.jl				

NLPModels.jl

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

- Define AbstractNLPModel;
- Define JuMPNLPModel and SimpleNLPModel;
- AMPL and CUTEst models are derived from it;
- Allows future models.







・ロト ・ 理 ト ・ ヨ ト ・ ヨ ト ・ ヨ

Workflow	Framework	CUTEst.jl	Practical example	Future work

Workflow

- Test Driven Development
- Problems as blocks
- Outline

2 Framework

- Julia
- Optimize.jl
- NLPModels.jl

3 CUTEst.jl

- CUTEst
- CUTEst.jl
- Flavors
- Some functions
- Practical example
 - TRON
 - Make it work
 - Make it right
- 5 Future work
 - Future work

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

Workflow	Framework	CUTEst.jl	Practical example	Future work
		0000000		
CUTEst				

CUTEst

- Repository of Nonlinear Optimization problems;
- Provides subroutines to obtain the problem's information;
- Decodes the problem, compiles your code with the problem's and runs your main code;

イロン 不良 イヨン イヨン ヨー ろくの

• Widely used.

Workflow	Framework	CUTEst.jl	Practical example	Future work
		0 00 0000		
CUTEst.jl				

CUTEst.jl

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

- Easy to install;
- Easy to use;
- Helps in many stages of the development.

Workflow	Framework	CUTEst.jl	Practical example	Future work
		000000		
CUTEst.jl				

CUTEst (Fortran)

◆ロ → ◆帰 → ◆ 国 → ◆ 国 → ● ● ● ● ● ● ● ●

```
... !open problem and define variables
CALL cutest_cdimen(st, ifile, n, m)
if (m.GT.0) THEN
STOP
ENDIF
CALL cutest_usetup(st, ifile, 7, 11, n, x, bl, bu)
CALL cutest_ufn(st, n, x, f)
... !close and end
```

Workflow	Framework	CUTEst.jl	Practical example	Future work
		0000000		
CUTEst.jl				

CUTEst.jl

◆ロ → ◆帰 → ◆ 国 → ◆ 国 → ● ● ● ● ● ● ● ●

Pkg.add("CUTEst") # Once (Eventually)

```
using CUTEst
```

```
nlp = CUTEstModel("ROSENBR")
```

```
x = nlp.meta.x0
```

```
f = obj(nlp, x)
```

cutest_finalize(nlp)

Workflow	Framework	CUTEst.jl	Practical example	Future work
		0000000		
Flavors				

Flavors

CALL cutest_ufn(st, n, x, f) CALL cutest_cfn(st, n, m, x, f, c)

Wrapper

◆ロ → ◆帰 → ◆ 国 → ◆ 国 → ● ● ● ● ● ● ● ●

```
st = Cint[0]
f = [0.0]
c = zeros(m)
ufn(st, Cint[n], x, f)
cfn(st, Cint[n], Cint[m], x, f, c)
```

Workflow	Framework	CUTEst.jl	Practical example	Future work
		00000000		
Flavors				

Flavors

CALL cutest_ufn(st, n, x, f) CALL cutest_cfn(st, n, m, x, f, c)

Julian way

◆ロ → ◆帰 → ◆ 国 → ◆ 国 → ● ● ● ● ● ● ● ●

```
f = ufn(nlp, x)
f = ufn(n, x)
f, c = cfn(nlp, x)
f, c = cfn(n, m, x)
f = cfn!(nlp, x, c) # Inplace
f = cfn!(n, m, x, c) # Inplace
```

Workflow	Framework	CUTEst.jl	Practical example	Future work
		00000000		
Flavors				

Flavors

CALL cutest_ufn(st, n, x, f) CALL cutest_cfn(st, n, m, x, f, c)

AbstractModel way

イロン 不良 イヨン イヨン ヨー ろくの

f = obj(nlp, x)
c = cons(nlp, x)
f = objcons(nlp, x) # If unconstrained
f, c = objcons(nlp, x) # If constrained

Workflow	Framework	CUTEst.jl	Practical example	Future work
		0000000		
Some functions				

Some functions

◆□▶ ◆□▶ ◆□▶ ◆□▶ □ のへ⊙

```
g = grad(nlp, x)
H = hess(nlp, x) # Sparse
H = hess(nlp, x, y) # Sparse
hrow, hcol, hval = hess_coord(nlp, x)
hrow, hcol, hval = hess_coord(nlp, x, y)
J = jac(nlp, x) # Sparse
jrow, jcol, jval = jac_coord(nlp, x)
Hv = hprod(nlp, x, v)
```

Workflow	Framework	CUTEst.jl	Practical example	Future work

Workflow

- Test Driven Development
- Problems as blocks
- Outline

2 Framework

- Julia
- Optimize.jl
- NLPModels.jl

3 CUTEst.jl

- CUTEst
- CUTEst.jl
- Flavors
- Some functions



- TRON
- Make it work
- Make it right
- 5 Future work
 - Future work

Workflow	Framework	CUTEst.jl	Practical example	Future work
			•••••	
TRON				

TRON: A practical example

Newton's Method for Large Bound-Constrained Optimization Problems

Chih-Jen Lin and Jorge J. Moré SIAM Journal on Optimization Vol. 9, No. 4, pp. 1100-1127, 1999.

Workflow	Framework	CUTEst.jl	Practical example	Future work
			000000000000	
TRON				

$$\min f(x)$$
 s. to $x \in \Omega$,

where $\boldsymbol{\Omega}$ is

$$\Omega = \{ x \in \mathbb{R}^n \mid \ell \le x \le u \},\$$

but can be extended to

$$\Omega = \{ x \in \mathbb{R}^n \mid \ell \le c_i^T x \le u, \ i \in \mathcal{I} \}.$$

Workflow	Framework	CUTEst.jl	Practical example	Future work
			00000000000	
TRON				

Outline of iteration k

Compute a model

$$m_k(d) = \frac{1}{2}d^T B_k d + d^T g_k.$$

3 Compute a gradient step $s_k = P[x_k - \alpha_k g_k] - x_k$ such that

$$m_k(s_k) \le \mu_0 g_k^T s_k, \quad \text{and} \quad \|s_k\| \le \mu_1 \Delta_k.$$

- Compute a step d_k better than s_k , i.e., further minimizing m_k with $||d_k|| \le \mu_1 \Delta_k$, and without leaving the bounds at $x_k + s_k$.
- Update x_k and Δ_k using Trust Region rules.

Workflow	Framework	CUTEst.jl	Practical example	Future work
			000 000000 000	
Make it work				

Simplest problems

• min
$$f(x) = \frac{1}{2}(x_1^2 + x_2^2);$$

- min $f(x) = \frac{1}{2}(x_1^2 + x_2^2)$, subject to $1 \le x_1, x_2 \le 2$;
- min $f(x) = \frac{1}{2}(x_1^2 + x_2^2)$, subject to $-1 \le x_1, x_2 \le 2$;
- min $f(x) = \frac{1}{2}(x_1^2 + x_2^2)$, subject to $-1 \le x_1 \le 1$, $1 \le x_2 \le 2$;
- min $f(x) = \frac{1}{2}(x_1^2 + x_2^2)$, subject to $0 \le x_1 \le 1, \ 1 \le x_2 \le 2$;

Rosenbrock with and without bounds;

	amework	CUTEst.jl	Practical example	Future work
			00000000000	
Make it work				

Minimum implementation (satisfying theory)

- Use $B_k = \nabla^2 f(x_k)$;
- **②** Use simple backtracking to find s_k ;
- $I Se d_k = s_k;$
- Use the Trust Region implemented in the framework (similar enough).

Workflow	Framework	CUTEst.jl	Practical example	Future work
			000000000000	
Make it work				

FactCheck

```
using FactCheck
facts("Simple test") do
 x0 = [1.2; 1.8]
  f(x) = dot(x,x)/2
 g(x) = x
 H(x) = eye(2)
  1 = [1.0; 1.0]
  u = [2.0; 2.0]
  nlp = SimpleNLPModel(x0, f, grad=g, hess=H, lvar=l,
    uvar=u)
```

◆ロ → ◆帰 → ◆ 国 → ◆ 国 → ● ● ● ● ● ● ● ●

Workflo 00000	w	Framework 00000	CUTEst.jl 00000000	Practical example 00000●●000000	Future work
Make it	work				
	x, fx, @fact x	dual = tron(nl > roughly(1	.p) .)		
	@fact f	x> roughly(f(1))		
	@fact d	ual> roughl	y(0.0)		
	end				

▲口> ▲園> ▲園> ▲園> ― 園 … 少々⊙

Workflow	Framework	CUTEst.jl	Practical example	Future work
			000 0000000	
Make it work				

Minimum implementation

◆ロ → ◆帰 → ◆ 国 → ◆ 国 → ● ● ● ● ● ● ● ●

```
function tron(nlp; m0 = 1e-2, m1 = 1.0)
 f(x) = obj(nlp, x)
 g(x) = grad(nlp, x)
 H(x) = hess(nlp, x)
 P(x) = max(min(x, u), 1)
 dual(x) = norm(P(x - g(x)) - x)
 tr = TrustRegion(100.0)
 D() = get_property(tr, :radius)
 x = nlp.meta.x0
```

```
Workflow
               Framework
                               CUTEst.jl
                                               Practical example
                                                                   Future work
                                               0000000000
Make it work
       while dual(x) > 1e-6
         q(d) = 0.5*dot(d, H(x) * d) + dot(d, g(x))
         s(a) = P(x - a*g(x)) - x
         a = 1
         while q(s(a)) > m0*dot(g(x), s(a)) ||
             norm(s(a)) > m1*D()
           a *= 0.9
         end
         xp = x + s(a)
         rho = ratio(f(x), f(xp), q(xp-x))
         if acceptable(tr, rho)
           x = xp
         end
         update!(tr, rho, norm(s(a)))
                                                イロト イ理ト イヨト イヨト ヨー わえで
```

Workflo 00000 Make it	w Framework 00000 work	CUTEst.jl 00000000	Practical example ○○○○○○●●●○○○	Future work 00
	end return x, f(x), o end	dual(x)		

▲口> ▲園> ▲園> ▲園> ― 園 … 少々⊙

Workflow	Framework	CUTEst.jl	Practical example	Future work
			000000000000000000000000000000000000000	
Make it right				

Limits

```
f(x) = begin sleep(0.1); sum(exp(x)) end
g(x) = begin sleep(0.1); exp(x) end
H(x) = begin sleep(0.1); spdiagm(exp(x), 0, 10, 10) end
x0 = 10*ones(10)
```

イロト イ理ト イヨト イヨト ヨー わえで

Workflow	Framework	CUTEst.jl	Practical example	Future work
			000000000000000000000000000000000000000	
Make it right				

Class os tests

• min
$$f(x) = \frac{1}{2}(x-r)^T Q^T \Lambda Q(x-r) + 1$$
, where
• $r = (1, \dots, 1)^T$:

• Q is an orthogonal matrix;

•
$$\Lambda = \operatorname{diag}(10^{-2}, \dots, 1)^T$$

• min $f(x) = \frac{1}{2}x^T B x + g^T x$, subject to $\ell \le x \le u$, where

•
$$B = Q^T \Lambda Q > 0.$$

- $\bullet\,$ Build solution and choose g
- Generalized rosenbrock with bounds.

Workflow	Framework	CUTEst.jl	Practical example	Future work
00000 Make it right	00000	0000000	000000000 00 0	

Improvements

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

- Matrix-free (hprod instead of hess);
- Store repeated function calls;
- Use inplace operations;

Workflow	Framework	CUTEst.jl	Practical example	Future work



5 Future work

• Future work

Workflow	Framework	CUTEst.jl	Practical example	Future work
				•0
Future work				

Future Work

ション 人間 アメヨア イヨア しゅうろう

- Implement various methods;
- Create a documentation/tutorial/example;
- Improve the benchmark;
- Benchmark many methods;
- Problem selector (Simple and CUTEst).

Workflow	Framework	CUTEst.jl	Practical example	Future work
				00
Thanks				



This presentation is licensed under the Creative Commons Attributions-ShareAlike 4.0 International License.