Mathematical Fidelity and Open Source Libraries for Large Scale Simulation and Optimization

William Symes

The Rice Inversion Project

RICE

Mathematics $\neq$ Computation, but...

Mathematical concepts $\rightarrow$ computational solutions

Example: function jets

$\Rightarrow$ Natural solutions - (1) storage and (2) update decisions for intermediate data, values

Agenda

Optimization

Jets

Optimization with Jets

Trust Region Algorithm

- x = vector, current estimate of minimizer
- f = objective function
- g = gradient of f
- H = approx. to Hessian of f

Trust Region Algorithm

- s = trial step
- Δ = trust radius

Two major substeps in each iteration:

- constrained step computation
- step decision, constraint update

RICE

Trust Region Algorithm

Step Computation: given Δ, f, g, H,

$$s = \text{approx. minimizer of } \left(f + g^T s + \frac{1}{2} s^T H s \right)$$

$$\text{subject to } s^T s \leq \Delta^2$$

return s

Trust Region Algorithm

Step Decision: given $x, \Delta, s, f, f_+, g, H$,

$$\text{actred} = f - f_+$$

$$\text{predred} = -g^T s - \frac{1}{2} s^T H s$$

Trust Region Algorithm

Step Decision:

- if actred $< 0.1\,$predred: $\Delta \leftarrow 0.5\,\Delta$
- else $x \leftarrow x + s$
  - if actred $> 0.9\,$predred: $\Delta \leftarrow 1.8\,\Delta$

return x, Δ

Trust Region Algorithm

TR Algorithm: while (not done),

1. $f = f(x)$, $g = g(x)$, $H = H(x)$
2. $s = \text{Step Computation}(f, g, H, \Delta)$
3. $f_+ = f(x + s)$
4. $x, \Delta = \text{Step Decision}(x, s, f, f_+, g, H, \Delta)$

Agenda

Optimization

Jets

Optimization with Jets

RICE

Handling intermediate data

Heavyweight data common to $x \to f(x), g(x), \ldots$:

- FEM computation: meshing, partitioning, elt. assembly, also used in $g(x)$
- time stepping: checkpointing, boundary data - adj. state method for $g(x)$

Handling intermediate data

Two problems: how to

1. share intermediate data, avoid recomputation
2. force recomputation when necessary

???

Handling intermediate data

obvious solution: pass required data structure as argument, i.e.

$$f = f(x, stuff); g = g(x, stuff)$$

(C programmer's "void *" trick - TAO)

$\Rightarrow$ internal details of f, g,... invade abstract algorithm

Handling intermediate data

from Diff'l Geometry: *jet* of function at point = (value, gradient, Hessian,...) - math. natural!

Computational jet: single data structure - value, gradient, intermed. data, ... = attributes

solves problem 1

RICE

Handling intermediate data

The Right Stuff (problem 2): end of TR step

$x, \Delta = \text{Step Decision}(x, s, f, f_+, g, H, \Delta)$

at start of next step, has x changed (is "stuff" right)?

Handling intermediate data

Obvious solution: retain a copy x_{saved} of x, check whether x_{saved} same as x on return

BUT

- generates more storage, data motion
- requires comparison of float data

Handling intermediate data

Efficient alternative: vector versioning

version is vector attribute, increment whenever vector *assigned*: for instance,

$$x \leftarrow x + s$$

$\Rightarrow x$.version++ $\Rightarrow$ jets depending on x must re-compute value, gradient,...

Handling intermediate data

Jet semantics with vector versioning:

$$jet(f, x) = (f(x), g(x), \ldots) \text{ } \textit{for variable } x$$

solves problem 2

RICE

Agenda

Optimization

Jets

Optimization with Jets

TR with jets

TR initialization, jet version:

- x, x_+, s: current iterate, trial iterate, step
- $e = jet(f, x), e_+ = jet(f, x_+)$

TR with jets

TR step, jet version: while (not done)

1. $s = \text{Step Computation}(e, \Delta)$
2. $x_+ = x + s$
3. $e, \Delta = \text{Step Decision}(e, e_+, \Delta)$

TR with jets

Step Decision, jet version: compute *actred*, *predred* as before,

- if $\text{actred} < 0.1\,\text{predred})$: $\Delta \leftarrow 0.5\,\Delta$
- else $e \leftrightarrow e_+$
  - if $\text{acted} > 0.9\,\text{predred}$: $\Delta \leftarrow 1.8\,\Delta$

return e, Δ

Summary

(Partial) Realizations - jets, similar stuff:

- Trilinos - NOX (jet=*Group*), ROL (jet=*Objective*)
- Rice Vector Library (jet=*Evaluation*)
- Madagascar algorithm dialect
- ...

Summary

Jets: mathematically correct, plus

- assure coherence of values, gradients,...
- side data (meshes, checkpoints,...) retained if possible, recomputed if necessary
- exactly the necessary value, gradient,... computations!

RICE

Thanks to:

- Jon Claerbout, Dave Nichols, Lester Dye, Amr El Bakry, Roscoe Bartlett, Sergey Fomel
- Mark Gockenbach, Shannon Scott, Hala Dajani, Tony Padula, Yin Huang
- The Rice Inversion Project, NSF