Please Pass the FOIE

dominique.orban@gerad.ca

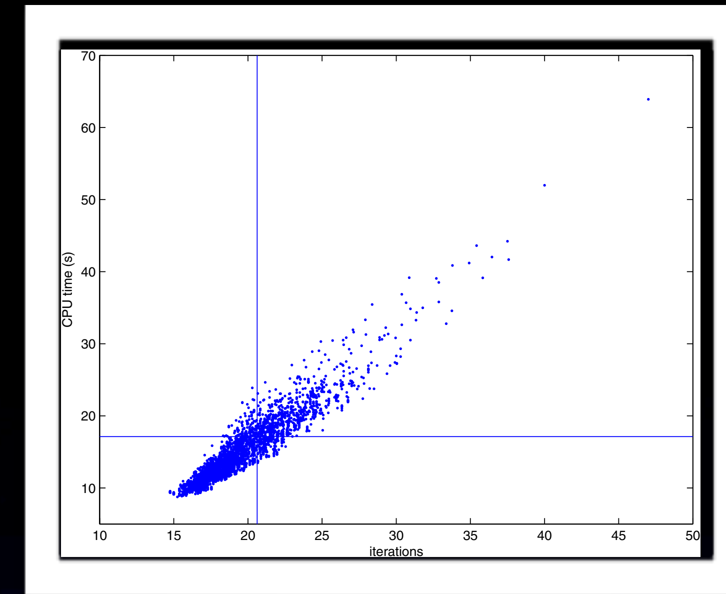
FOIE 2010, CERFACS, Toulouse

Overview of Interests

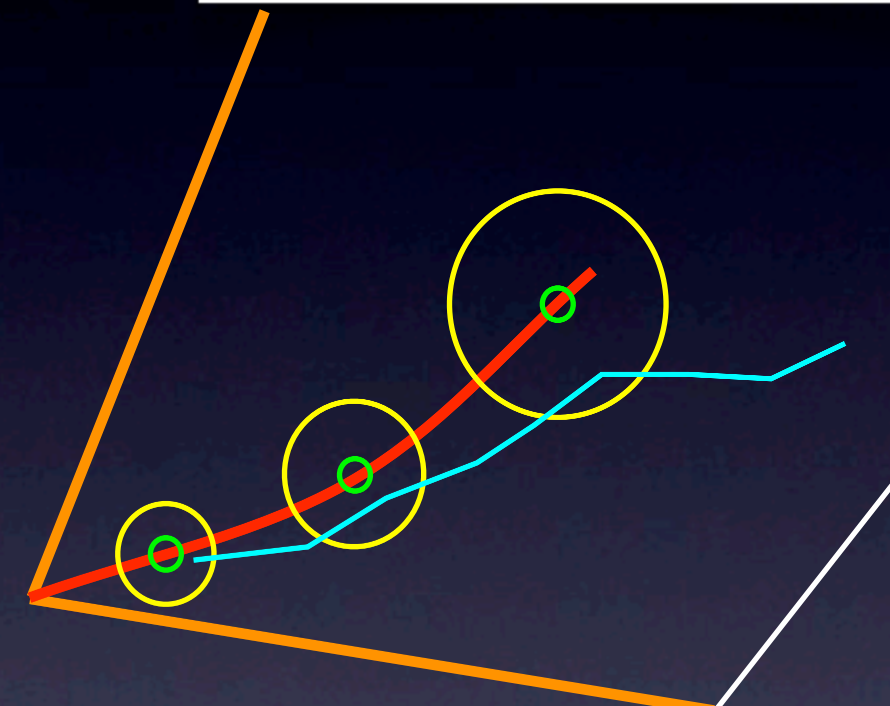
- Smooth Optimization
- Degeneracy / Regularization
- Programming Environments for Optimization
- Modeling Languages
- Application: Parameter Optimization

Please Interrupt!

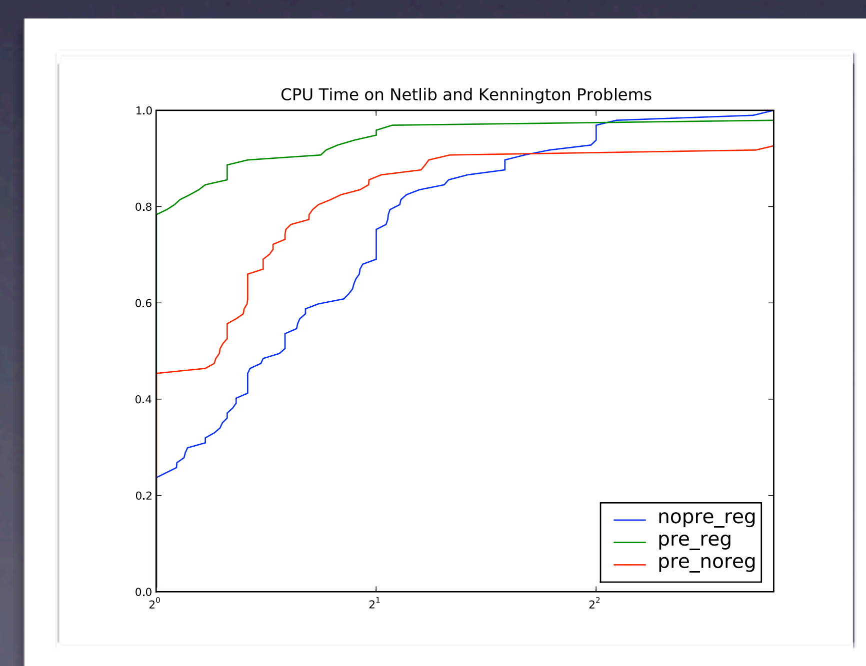
1997-2001: CERFACS



- Parameter Optimization: Trust-Region Method for Unconstrained Problems



- Primal-Dual Interior-Point Methods
- CUTEr (CUTE revisited)



Programming Environments for Optimization (& Co.)

NLPy

```
from nlpy.model import amplpy
from nlpy.optimize.solvers.lbfgs import LBFGSFramework

nlp = amplpy.AmplModel('woods')
lbfgs = LBFGSFramework(nlp, npairs=5, scaling=True, silent=True)
lbfgs.solve()
nlp.close()
```

Also various factorizations, linear solvers, preconditioners, constrained solvers, linesearches, trust-regions, etc.

PyKrylov

```
from numpy import ones
from pykrylov.cgs import CGS
from pykrylov.tfqmr import TFQMR
from pykrylov.bicgstab import BiCGSTAB

for KSolver in [CGS, TFQMR, BiCGSTAB]:
    ks = KSolver(lambda v: A*v, reltol = 1.0e-8)
    ks.solve(rhs, guess=ones(n), matvec_max=2*n)
```

CUTEr (with N. I. M. Gould and Ph. L. Toint)

- Over 1000 standard test problems
- Interfaces for C/C++, Fortran 77/90/95/2003 and Matlab
- Easy to install. Yes, really!

DrAMPL (with R. Fourer)

- Dissection lab for problems in AMPL format
- Some level of nonlinear preprocessing
- Some level of solver recommendation

GALAHAD (with N. I. M. Gould and Ph. L. Toint)

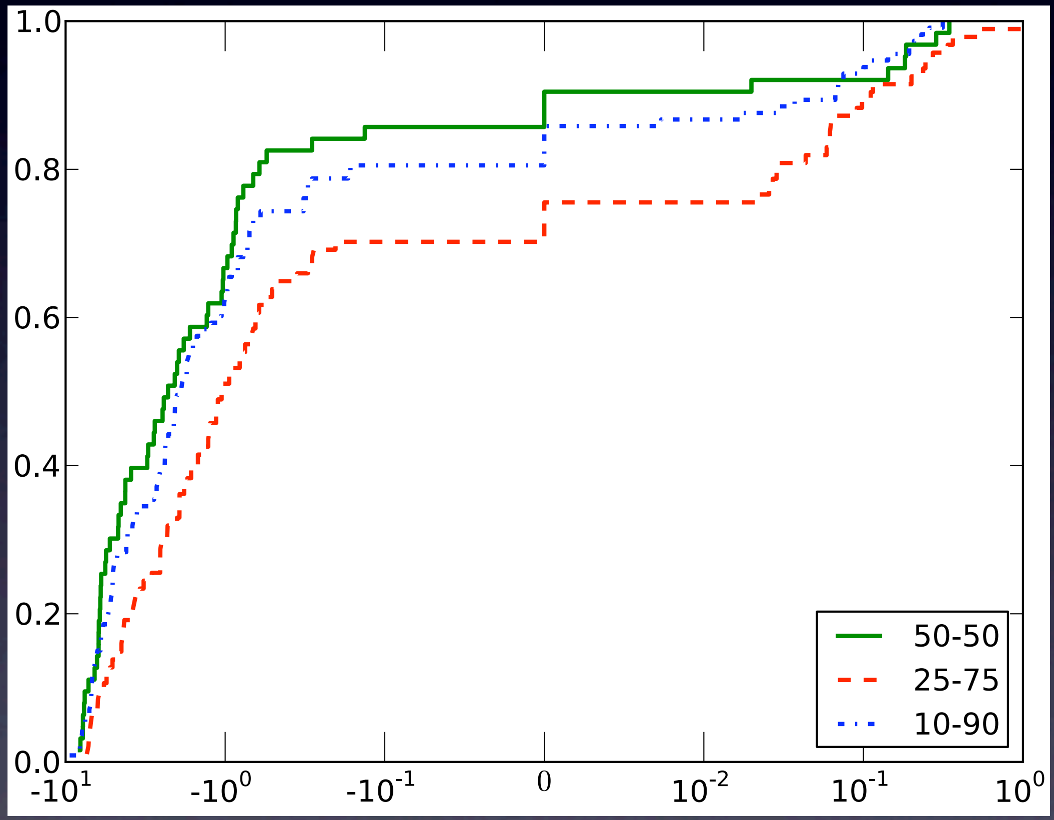
- Thread-safe library of Fortran 2003 modules
- Covers optimization, linear systems, least-squares, equations, …
- Features LANCELOT-B, SUPERB for NLP
- Various QP solvers, and more.

Parameter “Tuning” via Non-smooth Optimization

OPAL: Optimization of Algorithms

Python package for modeling parameter optimization problems

```python
from opal.Algorithms import DFO
from opal.TestProblemCollections import CUTEr
from opal.Solvers import NOMAD
from opal import ModelStructure, ModelData, BlackBoxModel

def avg_time(p,measures):
    cpuTime = measures[0]
    return cpuTime(p).sum() / len(cpuTime(p))

# Select real parameters for DFO
params = [par for par in DFO.parameters if par.is_real]

# Select tiny unconstrained HS problems
probs = [pb for pb in CUTEr.HS if pb.nvar<=5 and pb.ncon==0]

# Build model structure and model data
data = ModelData(DFO, probs, params)
structure = ModelStructure(objective=avg_time) #Unconstrained

blackbox = BlackBoxModel(modelData=data, modelStructure=structure)
NOMAD.solve(blackbox)
```



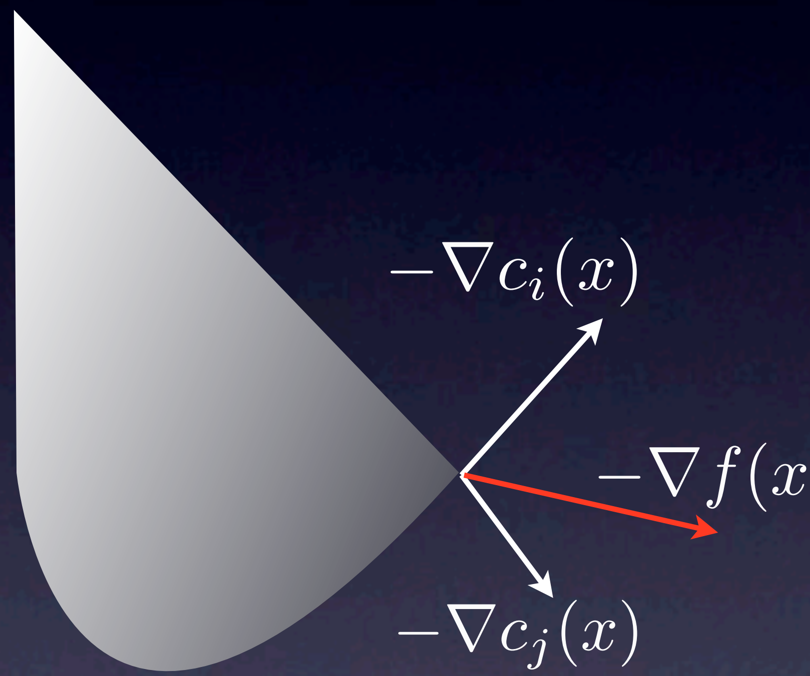
Joint work with C. Audet, C.-K. Dang

Degeneracy in NLP

Geometry of Optimality

If x is a local minimizer, then

$$\begin{aligned} \nabla f(x) &= \sum_i y_i \nabla h_i(x) + \sum_j z_j \nabla c_j(x), \\ z_j c_j(x) &= 0, \quad \text{for all } j, \\ h(x) &= 0, \quad (z, c(x)) \geq 0. \end{aligned}$$



- Only active inequality constraints matter
- Depends on *algebraic description* of feasible set

LP and Convex QP

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \ c^Tx + \tfrac{1}{2}x^TQx \quad \text{subject to}\ \ Ax = b,\ \ x \geq 0$$

- IPM in augmented form
- Degenerate = (rank(*A*) < *m*)

$$\begin{bmatrix} -(Q + X^{-1}Z) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \dots$$

Joint work with M. P. Friedlander

Primal-Dual Regularization

$$\begin{bmatrix} -(Q + X^{-1}Z + \rho I) & A^T \\ A & \delta I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \dots$$

- Iteration-dependent regularization params
- RHS *unchanged*!
- May be interpreted as:

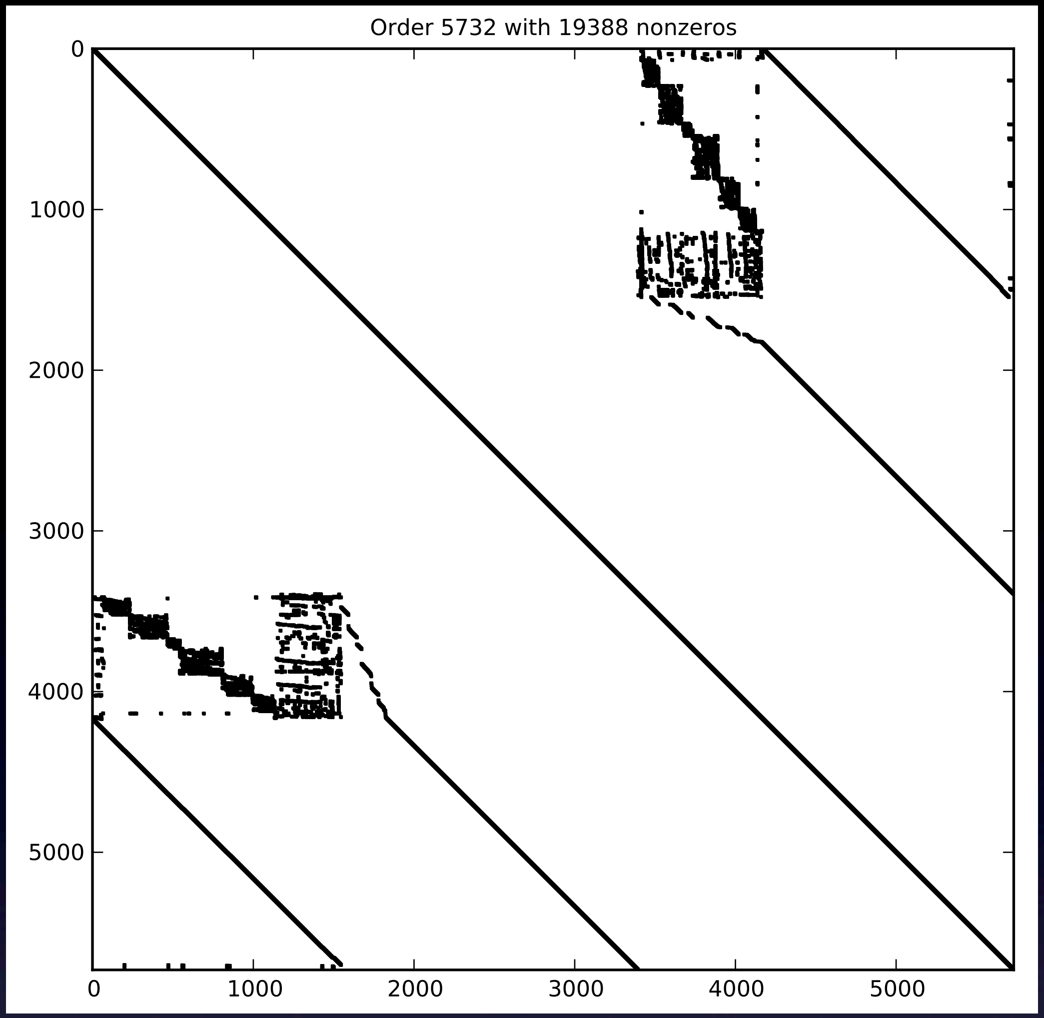
$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; c^T x + \tfrac{1}{2} x^T Q x + \tfrac{1}{2}\rho \|x - x_k\|^2 + \tfrac{1}{2}\delta \|r + y_k\|^2 \\ &\text{subject to} \;\; Ax + \delta r = b, \;\; x \geq 0 \end{aligned}$$

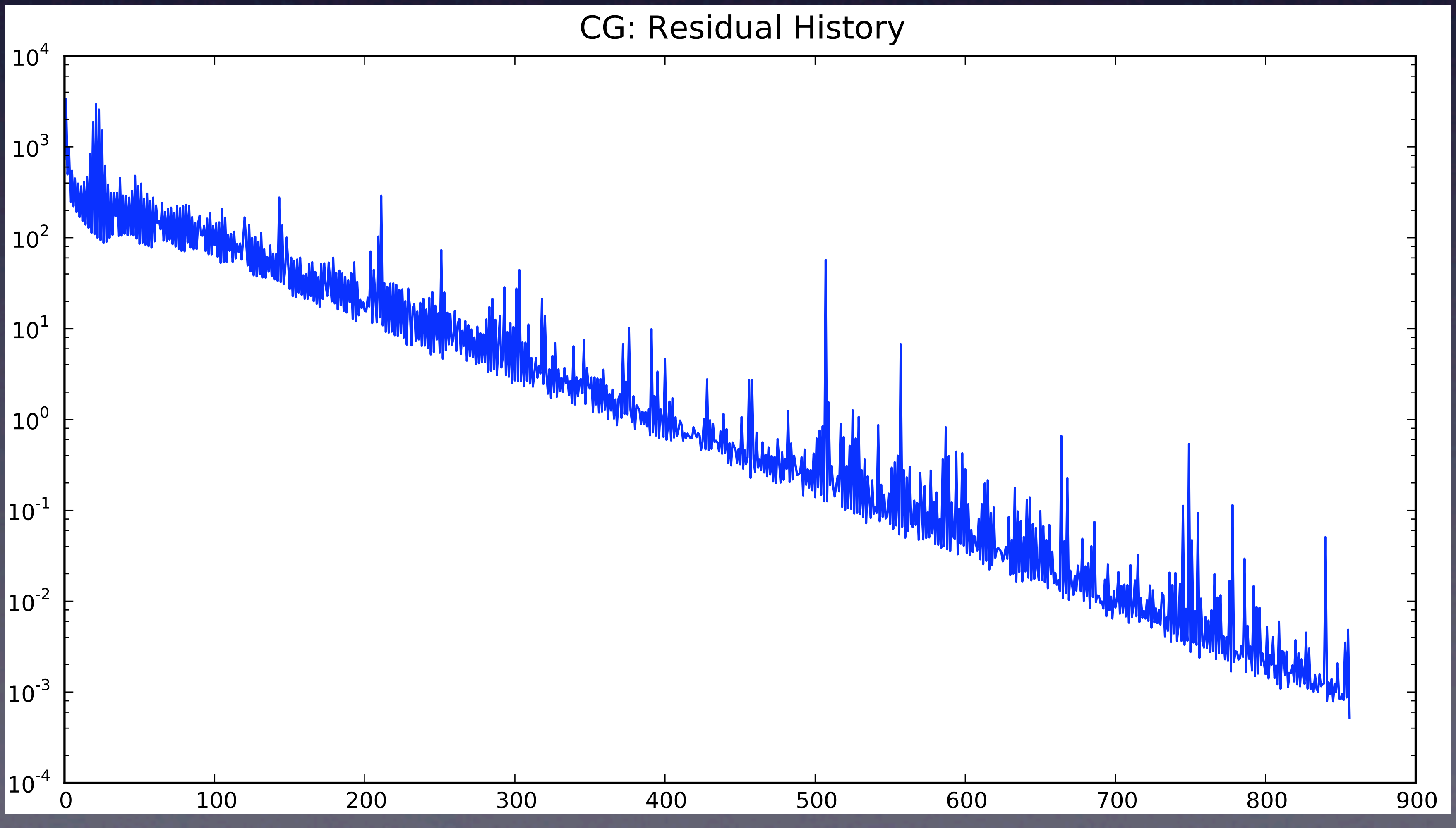
SQD Matrices

$$K = \begin{bmatrix} -E & A^T \\ A & F \end{bmatrix} \qquad E, F \succ 0 \qquad\qquad P^T K P = LDL^T$$

- Direct factorization much faster than as a general symmetric indefinite matrix!
- Iterative Methods: CG cannot break down. Can be implemented as variant of GK.
- Appropriate preconditioner?

Joint work with Mario Arioli


Order 5732 with 19388 nonzeros
0
1000
2000
3000
4000
5000
0
1000
2000
3000
4000
5000


CG: Residual History
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
100
200
300
400
500
600
700
800
900

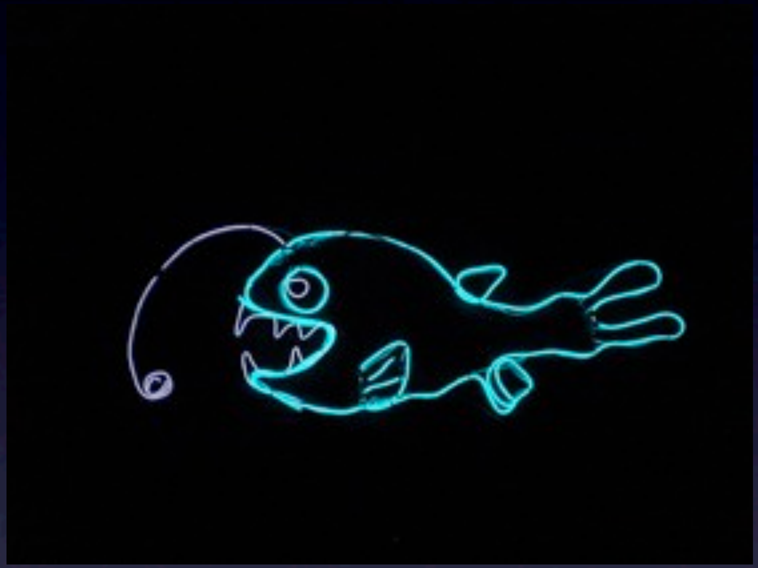
NLP: Elastic Variables

- Degeneracy = failure of appropriate CQ
- New variables to regularize the problem
- Nice convergence and degeneracy-revealing properties

Joint work with N. Gould, Ph. L. Toint, Z. Coulibaly, P.-R. Curatolo

Elastic Problem

$$\begin{array}{ll} \underset{x,s,t}{\text{minimize}} & f(x) + \nu \sum_i t_i + \nu \sum_j s_j \\ \text{subject to} & -t \le h(x) \le t, \quad t \ge 0, \\ & c(x) + s \ge 0, \qquad s \ge 0 \end{array}$$



- Always satisfies weakest CQ for NLP, MPCC, MPVC
- Only has inequality constraints
- Solve with an interior-point method: *SUPERB*

Lack of Strict Complementarity

- Puiseux expansion of the central path
- Extrapolation to obtain x^*

Joint work with N. Gould, Z. Coulibaly, A. Wächter and D. Robinson

In the Making

- With L. Giraud & E. Agullo : optimization of PLASMA
- With S. Gratton : variable-memory quasi-Newton
- With M. Mouffe : a sloppy method (in a good way)
- With N. Gould and S. Thorne : PDE-constrained opt

A Few Final Words

- CERFACS is well known and will stay with you
- In Toulouse you have access to many engineers: Talk to them!
- Look for innovation: breadth-first, not depth-first search
- In “Applied Mathematics”, there is “Applied”
- Think Python

À une prochaine FOIE !

www.gerad.ca/~orban

dominique.orban@gerad.ca