



THE UNIVERSITY OF  
CHICAGO

# Stat 310: Numerical Optimization

Mihai Anitescu

- 1.1 Logistics for the class
- 1.2 Example of Optimization Problems and the Optimization Landscape
- 1.3 Modeling Optimization Problems
- 1.4 The object of continuous Optimization and course objectives.
- 1.5 Newton's Method.
- 1.6 The role of linear algebra; recap of direct methods.

# 1.1.Course Logistics

---

- Instructors: Mihai Animescu  
([animescu@galton.uchicago.edu](mailto:animescu@galton.uchicago.edu)) and Lek-Heng Lim
- Course Assistant: Han Han  
([han@galton.uchicago.edu](mailto:han@galton.uchicago.edu))
- The course is divided in 2 parts (tentative);
  - MA: Nonlinear Programming ~ 10 lectures
  - LHL: Convex Optimization ~ 10 lectures

# 1.1 Course Logistics

---

- Assignments
  - Combination of theoretical problems and computer projects using Matlab
  - 4-5 assignments  $\sim$  1 per week = your grade for this part.
- Office Hours: TTh, 3-4, Eck 104.
- Web site and contact: (remember my name ☺ ).
- How do I get in touch with you? What do you use for computing?

## 1.2 Context of Optimization

---

- What is nonlinear optimization optimization?
- Why? Example of optimization based on a subjective criterion.
- Why? Example of optimization problems derived from variational principles in physics.
- Thanks – Sven Leyffer, etc ...

# Nonlinear Optimization-Nonlinear Programming

$$\begin{array}{|l} \min \\ \text{s.t.} \end{array} \quad \begin{array}{l} f(x) \\ c(x) = 0, h(x) \leq 0 \end{array} \quad (\text{or}) \quad \begin{array}{|l} \min \\ \text{s.t.} \end{array} \quad \begin{array}{l} f(x) \\ c(x) = 0, h(x) + y = 0 \\ y \geq 0 \end{array} \quad (\text{or}) \quad \begin{array}{|l} \min \\ \text{s.t.} \end{array} \quad \begin{array}{l} f(x) \\ c(x) = 0 \\ x \in K \end{array}$$

- The variables  $y$  are called slacks.
- In the latter case, the “data” functions  $f, c$ , are not identical with the 2 preceding cases.
- The problem is called **nonlinear** when either  $f$  or  $(c, h)$  or both are nonlinear.
- The set  $K$  may include integrality constraints, MINLP.
- The above is a powerful modeling paradigm, in which many problems may be rephrased or approximated, though it is important to exploit the particularities of the problem – the “structure”.**

## ES: “Subjective” criteria of optimization.

---

- Trying to optimize an user defined criterion: time to accomplish a task, or cost, or matching criterion ....
- Generally a specialist makes a judgment and defines the criterion based on domain-specific knowledge.
- Optimal behavior and design of Engineering, Financial, Management applications.
- Examples: structural design and phase problems in crystallography.

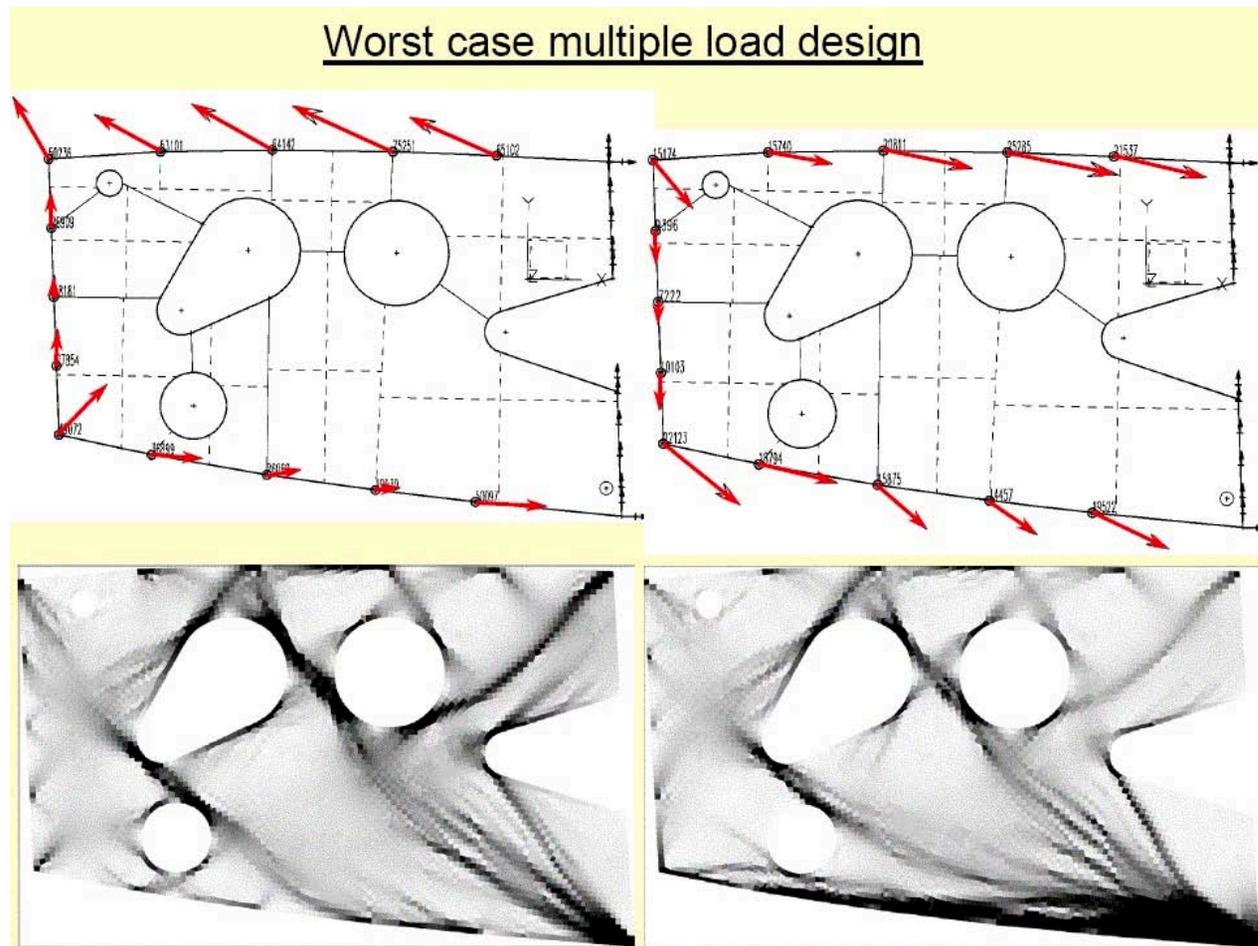
# ES1: The Airbus wing

---



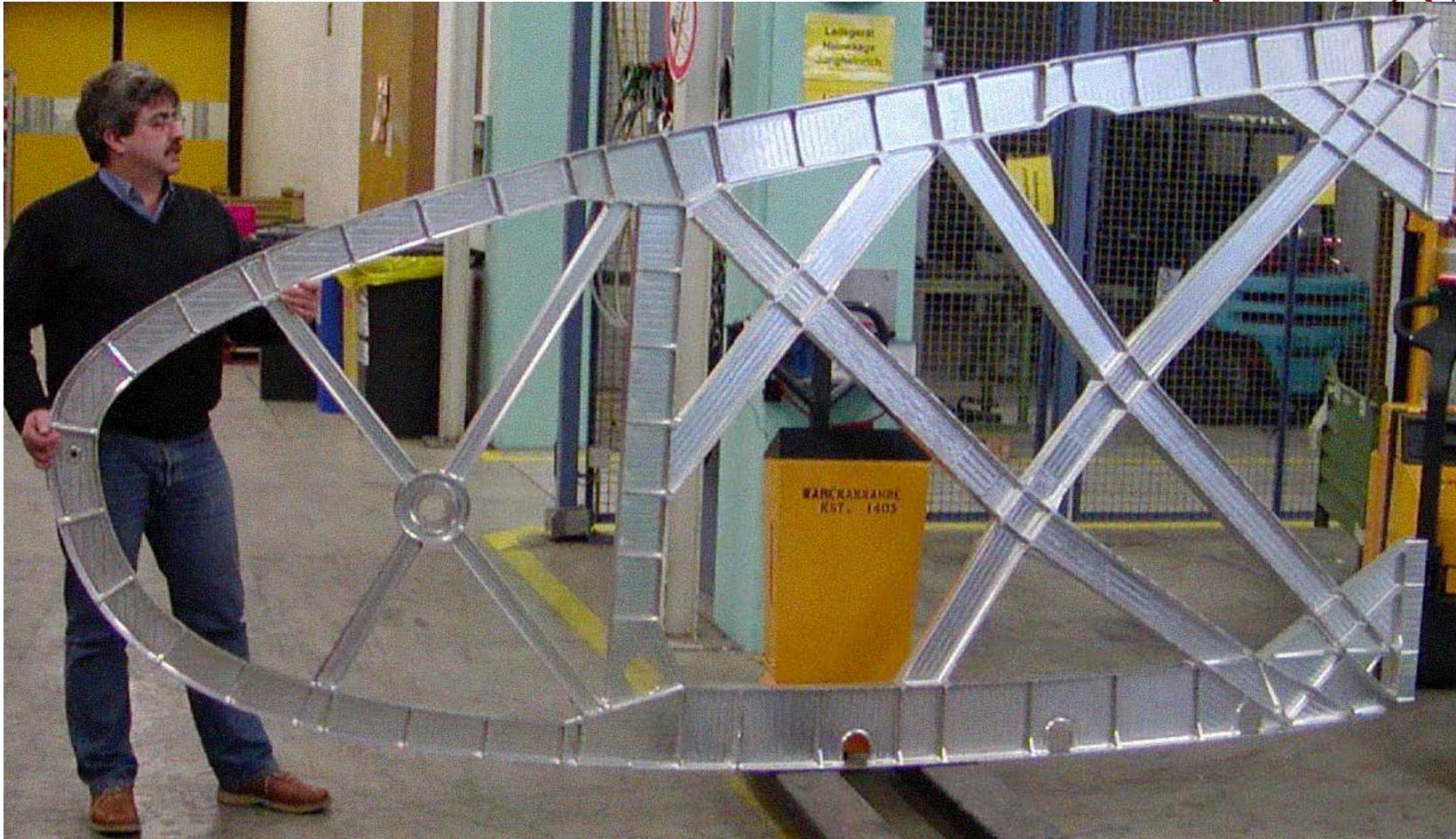
(From Sven Leyffer): Optimizing the inboard inner leading edge ribs.

# ES1: Design considerations (Kocvara et al.)



Minimize weight of the structure subject to load and design restrictions

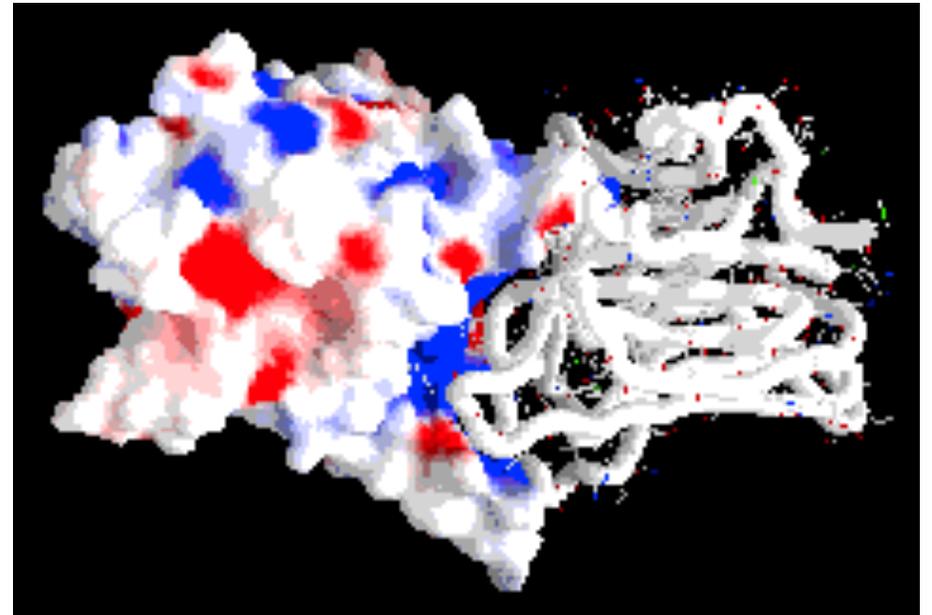
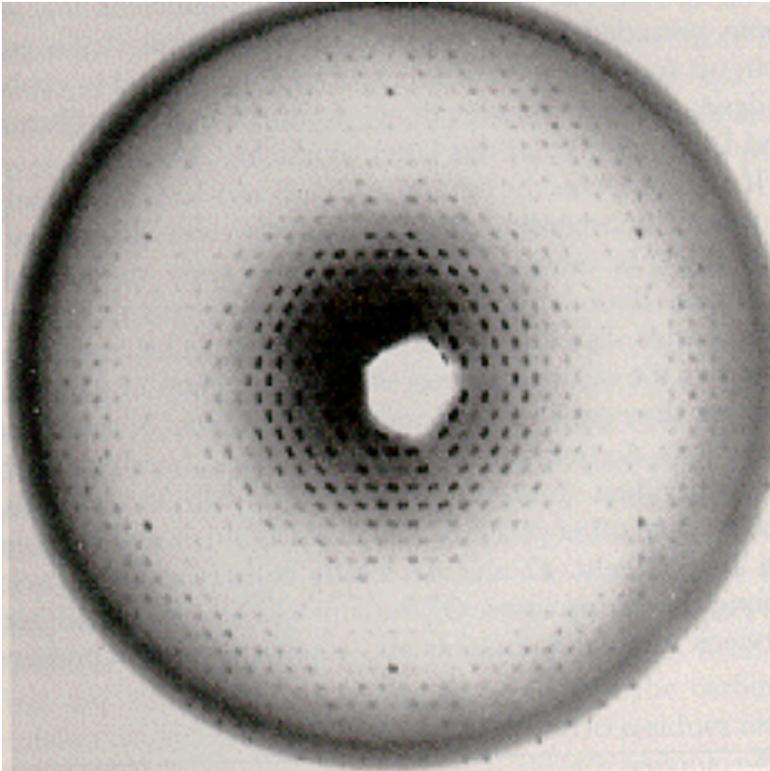
# ES1: Final Design—”Truss topology”



STRUCTURE: linear objective function, nonlinear inequality and equality constraints, continuous variables.

# ES2: Xray crystallography

---



- How do we obtain a 3D structure (right) from its diffraction pattern (left)?
- It is essentially the unique high resolution approach to detect protein structure irrespective of size.
- Problem: Find the atomic distributions that minimizes “discrepancy”.

# ES2: Phase problem-centrosymmetric

## (Sahinidis et al.)

### Indices

- $m$  index used for reflections ( $m = 1, \dots, M$ )  
 $t$  index used for triplet invariants ( $t = 1, \dots, T$ )

### Variables

- $\phi_m$  phase of the  $m$ th reflection  
 $\varphi_m$  normalized phase of the  $m$ th reflection equal to  $\phi_m / 2\pi$   
 $\omega_t$  triplet invariant defined by  $\omega_t = \phi_{m_t} + \phi_{m'_t} + \phi_{m''_t}$  where  $\mathbf{h}_{m_t} + \mathbf{h}_{m'_t} + \mathbf{h}_{m''_t} = \mathbf{0}$   
 $\alpha_t$  binary decision variable  
 $\beta_t$  binary decision variable equal to  $(1 - \cos \omega_t)$

### Parameters

- $M$  number of reflections  
 $n$  number of atoms in the unit cell  
 $T$  number of invariants  
 $|E_m|$  structure factor amplitude associated with reflection  $\mathbf{h}_m$   
 $A_t$  constant equal to  $2n^{-1/2}|E_{m_t}||E_{m'_t}||E_{m''_t}|$   
 $\bar{\omega}_t$  conditional expected value of the cosine of the triplet invariant, equal to  $I_1(A_t)/I_0(A_t)$  (Germain *et al.*, 1970)

### Model M1

$$\begin{aligned} \min \quad & f(\boldsymbol{\beta}) = \frac{\sum_{t=1}^T A_t (4\beta_t \bar{\omega}_t + (1 + \bar{\omega}_t^2 - 2\bar{\omega}_t))}{\sum_{t=1}^T A_t} \\ \text{s.t.} \quad & \varphi_{m_t} + \varphi_{m'_t} + \varphi_{m''_t} = 2\alpha_t + \beta_t, \quad t = 1, \dots, T \quad (1) \\ & \varphi_m \in \{0, 1\}, \quad m = 1, \dots, M \\ & \alpha_t, \beta_t \in \{0, 1\}, \quad t = 1, \dots, T. \end{aligned}$$

STRUCTURE: nonlinear objective, linear equality constraints,  
 Mixed continuous and integer variables

## ES3: Maximum Likelihood:

---

$$\log p(y | \theta) = -\frac{1}{2}W(\theta_1)^T K(\theta_2)^{-1}W(\theta_1) - \frac{1}{2} \log |K(\theta_2)| - \frac{n}{2} \log 2\pi$$

- STRUCTURE: Continuous Optimization, No Constraints.
- Difficulty, how do you compute the derivative of the determinant term?

# ES4: Stochastic Unit Commitment with Wind Power (SAA)

$$\min \quad \text{COST} = \frac{1}{N_s} \sum_{s \in \mathcal{S}} \left( \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{T}} c_{sjk}^p + c_{jk}^u + c_{jk}^d \right)$$

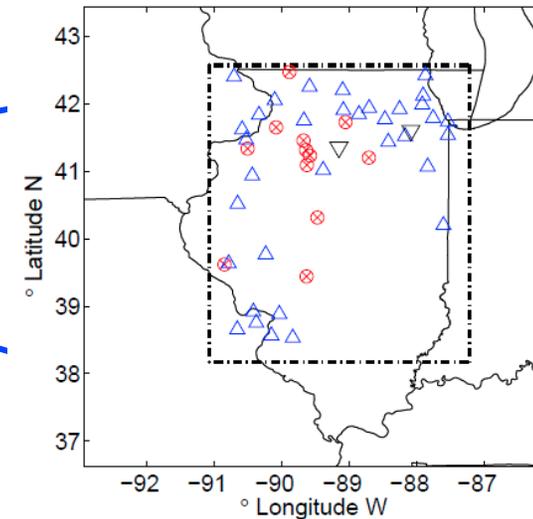
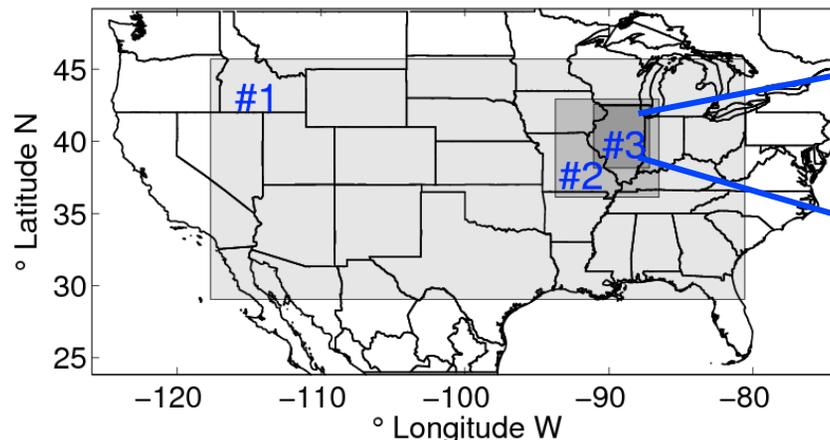
$$\text{s.t.} \quad \sum_{j \in \mathcal{N}} p_{sjk} + \sum_{j \in \mathcal{N}_{wind}} p_{sjk}^{wind} = D_k, s \in \mathcal{S}, k \in \mathcal{T}$$

$$\sum_{j \in \mathcal{N}} \bar{p}_{sjk} + \sum_{j \in \mathcal{N}_{wind}} p_{sjk}^{wind} \geq D_k + R_k, s \in \mathcal{S}, k \in \mathcal{T}$$

ramping constr., min. up/down constr.

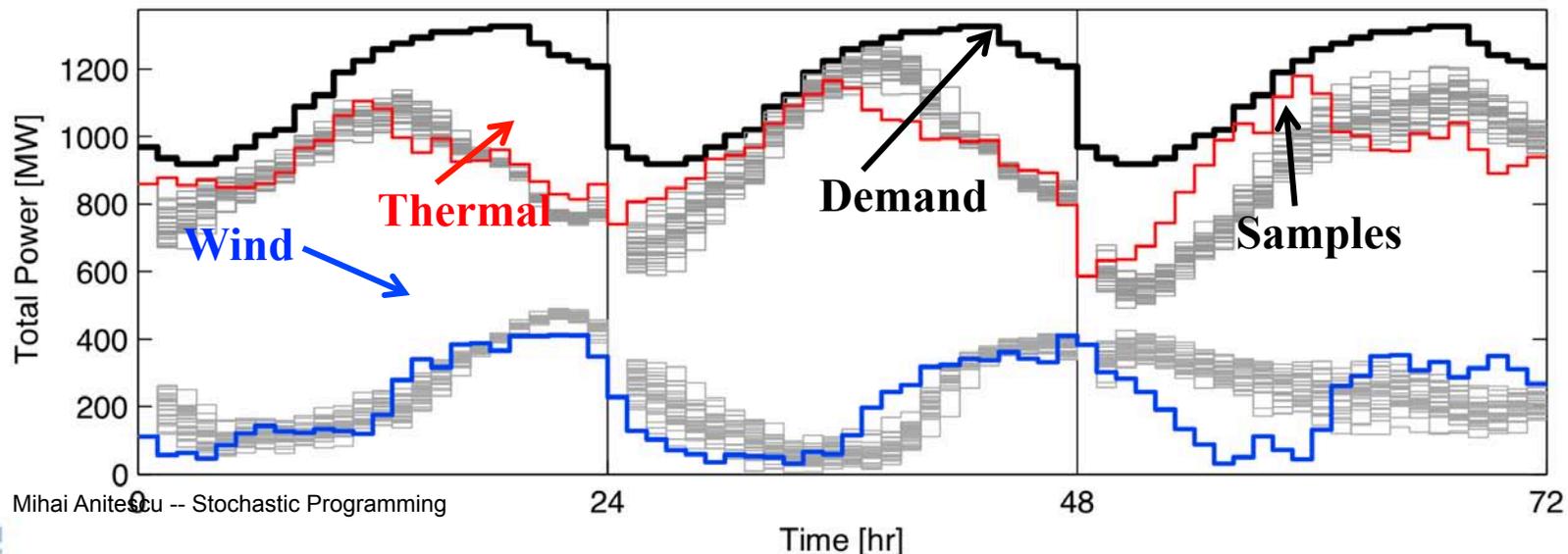


- Wind Forecast – WRF(Weather Research and Forecasting) Model
  - Real-time grid-nested 24h simulation
  - 30 samples require 1h on 500 CPUs (Jazz@Argonne)



# Wind power forecast and stochastic programming

- Unit commitment & energy dispatch with uncertain wind power generation for the State of Illinois, assuming 20% wind power penetration, using the same windfarm sites as the one existing today.
- Full integration with 10 thermal units to meet demands. Consider dynamics of start-up, shutdown, set-point changes
- Does uncertainty matter? ... Yes. **The solution is only 1% more expensive than the one with exact information. Solution on average infeasible at 10%.**

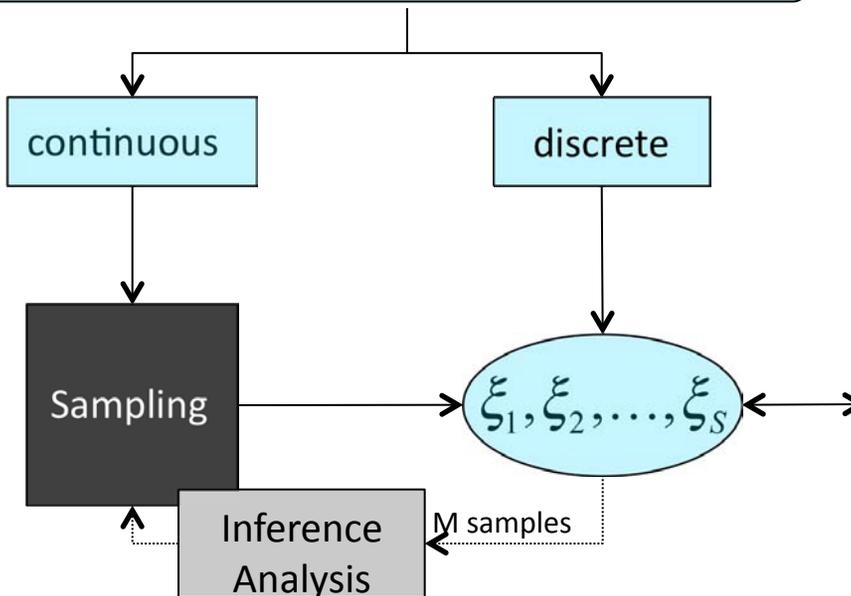


# Management under uncertainty paradigm: stochastic programming.

- Two-stage stochastic programming with recourse (“here-and-now”)

$$\begin{aligned} & \underset{x_0}{\text{Min}} \left\{ f_0(x_0) + \mathbb{E} \left[ \underset{x}{\text{Min}} f(x, \omega) \right] \right\} \\ \text{subj. to. } & A_0 x_0 = b_0 \\ & A(\omega) x_0 + B(\omega) x = b(\omega) \\ & x_0 \geq 0, \quad x(\omega) \geq 0 \end{aligned}$$

- $\xi(\omega) := (A(\omega), B(\omega), b(\omega), Q(\omega), c(\omega))$



Sample average approximation (SAA)

$$\begin{aligned} & \underset{x_0, x_1, x_2, \dots, x_S}{\text{Min}} \quad f_0(x) + \frac{1}{S} \sum_{i=1}^S f_i(x_i) \\ \text{subj. to. } & A_0 x_0 = b_0 \\ & A_k x_0 + B_k x_k = b_k, \\ & x_0 \geq 0, \quad x_k \geq 0, \quad k = 1, \dots, S \end{aligned}$$

# Linear Algebra of Primal-Dual Interior-Point Methods

Convex quadratic problem

$$\begin{aligned} \text{Min } & \frac{1}{2} x^T Q x + c^T x \\ \text{subj. to. } & Ax = b \\ & x \geq 0 \end{aligned}$$



IPM Linear System

$$\begin{bmatrix} Q + \Lambda & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = rhs$$



Multi-stage SP

Two-stage SP

**nested arrow-shaped linear system**  
(via a permutation)

$$\begin{bmatrix} \tilde{H}_1 & B_1^T & & & & & & 0 & 0 \\ B_1 & 0 & & & & & & A_1 & 0 \\ & & \tilde{H}_2 & B_2^T & & & & 0 & 0 \\ & & B_2 & 0 & & & & A_2 & 0 \\ & & & & \dots & & & \vdots & \vdots \\ & & & & & & \tilde{H}_S & B_S^T & 0 & 0 \\ & & & & & & B_S & 0 & A_S & 0 \\ 0 & A_1^T & 0 & A_2^T & \dots & 0 & A_S^T & \tilde{H}_0 & A_0^T \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & A_0 & 0 \end{bmatrix}$$

# The Direct Schur Complement Method (DSC)

- Uses the arrow shape of H

$$\begin{bmatrix} H_1 & & & G_1^T \\ & H_2 & & G_2^T \\ & & \ddots & \vdots \\ & & & H_S & G_S^T \\ G_1 & G_2 & \dots & G_S & H_0 \end{bmatrix} = \begin{bmatrix} L_1 & & & & \\ & L_2 & & & \\ & & \ddots & & \\ & & & L_S & \\ L_{10} & L_{20} & \dots & L_{S0} & L_c \end{bmatrix} \begin{bmatrix} D_1 & & & & \\ & D_2 & & & \\ & & \ddots & & \\ & & & D_N & \\ & & & & D_c \end{bmatrix} \begin{bmatrix} L_1^T & & & & L_{10}^T \\ & L_2^T & & & L_{20}^T \\ & & \ddots & & \vdots \\ & & & L_S^T & L_{S0}^T \\ & & & & L_c^T \end{bmatrix}$$

- Solving Hz=r

$$L_i D_i L_i^T = H_i, \quad L_{i0} = G_i L_i^{-T} D_i^{-1}, \quad i = 1, \dots, S,$$

$$C = H_0 - \sum_{i=1}^S G_i H_i^{-1} G_i^T, \quad L_c D_c L_c^T = C.$$

Implicit factorization

$$w_i = L_i^{-1} r_i, \quad i = 1, \dots, S,$$

$$w_0 = L_c^{-1} \left( r_0 - \sum_{i=1}^S L_{i0} w_i \right)$$

Back substitution

$$v_i = D_i^{-1} w_i, \quad i = 0, \dots, S$$

Diagonal solve

$$z_0 = L_c^{-1} v_0$$

$$z_i = L_i^{-T} (v_i - L_{i0}^T z_0), \quad i = 1, \dots, S.$$

Forward substitution

# Large-scale performance (with Miles Lubin, STATS)

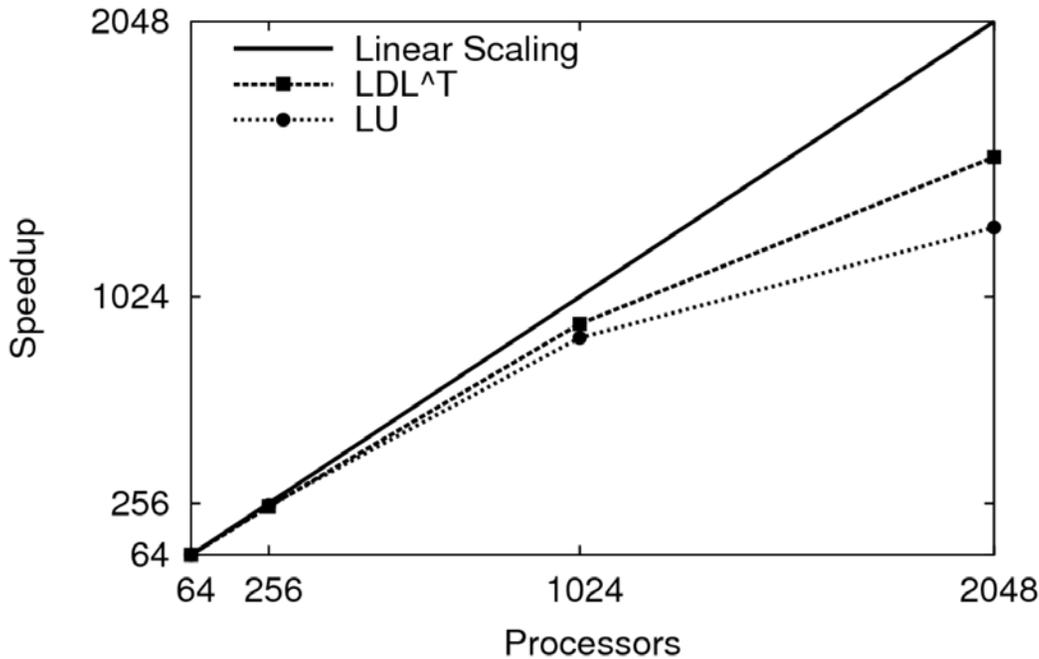
- Comparison of ScaLapack (LU), Elemental(LU), and  $LDL^T$  (1024 cores)

Units	1st Stage Size ( $Q+A$ )	Factor (Sec.)			Reduce (Sec.)	
		$LU(S)$	$LU(E)$	$LDL^T$	$LU$	$LDL^T$
300	23436+1224	16.59	20.04	<b>6.71</b>	54.32	<b>26.35</b>
640	49956+2584	60.67	83.24	<b>36.77</b>	256.95	<b>128.59</b>
1000	78030+4024	173.67	263.53	<b>90.82</b>	565.36	<b>248.22</b>

SAA problem:  
**189 million variables**

Total Walltime

- Strong scaling
  - 90.1%** from 64 to 1024 cores;
  - 75.4% from 64 to 2048 cores.
  - > 4,000 scenarios.



# ESO: Other applications.

---

- Data assimilation in weather forecasting → PDE constraint
- Image reconstruction from acoustic wave data → PDE constraint
- Crew scheduling, vehicle routing → integer variables
- Reactor core reloading nonlinear → integer variables
- Radio therapy treatment planning → nonlinear integer
- Oil field infrastructure design → PDE c/s & integer var
- Simulation of competition in electricity markets equilibrium c/s.

# EV: Variational Description of Phenomena in Physical Sciences

---

- In these problems, the “state” variable is the solution of an optimization problem, which is formulated based on a law of physics, rather than a subjective criterion.
- In electronic structure computation: the electronic density.
- In complex fluids the (density and species) distributions at thermodynamic equilibrium.
- In Hamiltonian systems, the trajectory is the solution of an optimization problem
- Fermat's variational principle states that a signal in anisotropic media propagates between two points along a curve which renders Fermat's functional  $I(l)$  stationary
- ....

# EV1: (Thomas-Fermi) Density Functional Theory

**Problem:** For a given atomic configuration, determine electronic density from using the variational principle. STRUCTURE: nonlinear objective, linear constraints, continuous variables.

$$\min_{\rho} E[\rho, \{\mathbf{R}_A\}]$$

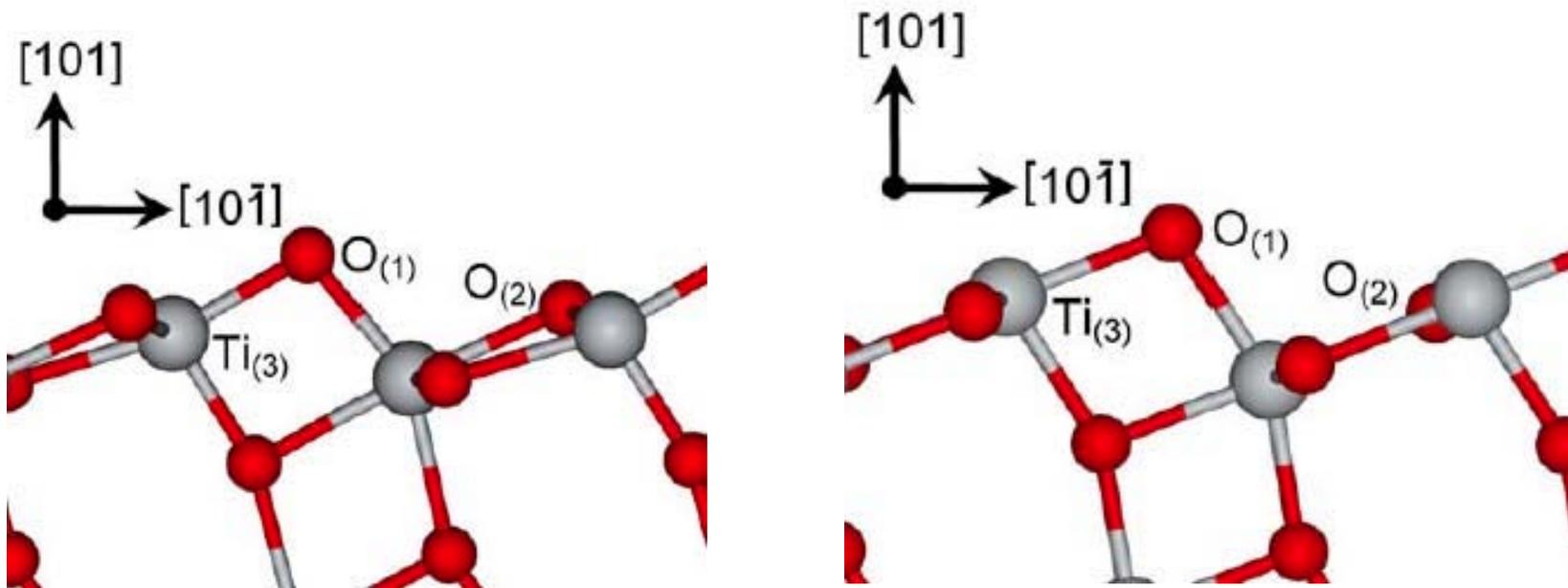
$$\text{s.t. } \int \rho = N_e$$

$$E[\rho, \{\mathbf{R}_A\}] = E_{ne}[\rho, \{\mathbf{R}_A\}] + J[\rho] + K[\rho] + T[\rho] + V_{nn}(\{\mathbf{R}_A\})$$

$$T[\rho] = C_F \int \rho^{\frac{5}{3}}(\mathbf{r}) d\mathbf{r}, \quad K[\rho] = -C_x \int \rho^{\frac{4}{3}}(\mathbf{r}) d\mathbf{r}$$

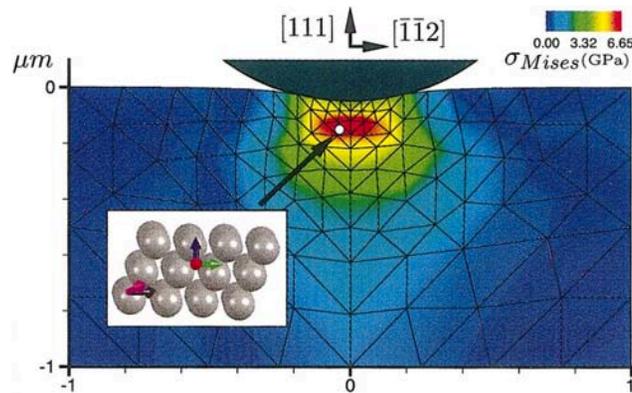
$$E_{ne}[\rho, \{\mathbf{R}_A\}] = \sum_{A=1}^M \int \frac{Z_A \rho(\mathbf{r})}{\|\mathbf{R}_A - \mathbf{r}\|} d\mathbf{r}, \quad J[\rho] = \frac{1}{2} \iint \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|} d\mathbf{r} d\mathbf{r}'$$

# EV1: Surface structure of the TiO<sub>2</sub> nanoparticle

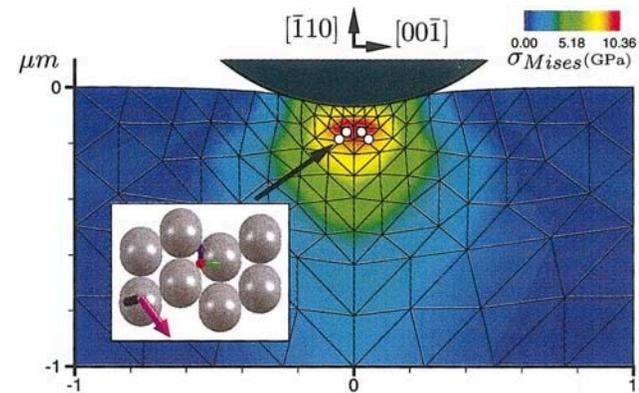


Computations carried out by Peter Zapol et al. from MSD-Argonne, using Kohn-Sham DFT.

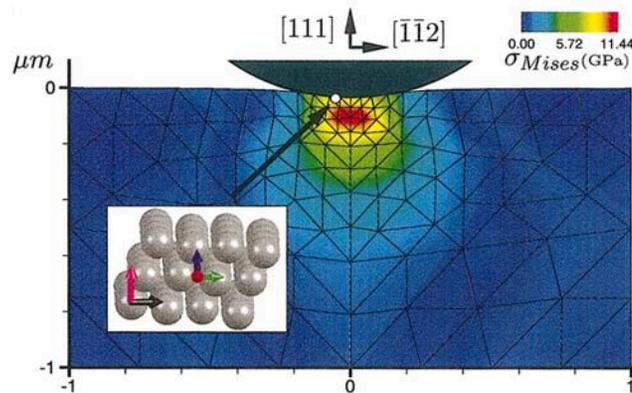
# EV1: Nano-indentation



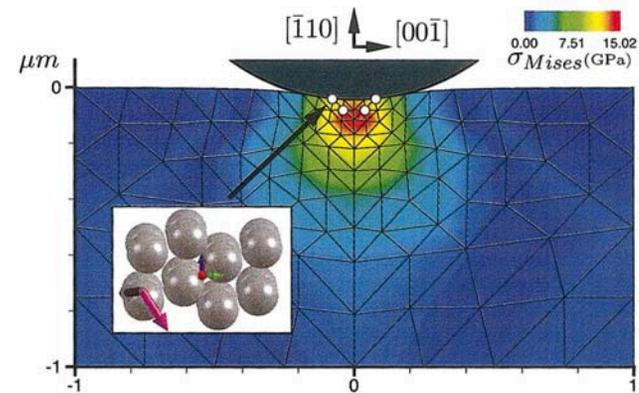
(a)



(a)



(b)



(b)

- One of the “hot pursuits” in mechanical engineering: Simulating complex phenomena starting from first principles, as opposed to empirical potentials.
- Density Functional Theory based defect nucleation (Carter, Ortiz, et al.)

# EV2: Multi-rigid-body dynamics with contact and friction

---

- A subject pursued by the author of this presentation for some time.
- Essential in the study of robotics, granular materials, pharmaceutical drug processing (powders).
- The velocity of the system at the next step is the solution of the minimum energy problem subject to nonpenetration and frictional constraints.
- Example: the study of size-based segregation in granular materials. STRUCTURE: quadratic objective, quadratic constraints.

$$\begin{aligned}
 v^{(l+1)} &= \operatorname{argmin}_{\hat{v}} \frac{1}{2} \hat{v}^T M \hat{v} + k^{(l)T} \hat{v} \\
 \text{subject to } &\frac{1}{h} \Phi^{(j)}(q^{(l)}) + \nabla \Phi^{(j)T} \hat{v} + \mu^{(j)} d_k^{(j)T} \hat{v} \geq 0, \\
 &j \in \mathcal{A}(q^{(l)}, \epsilon), \quad k = 1, 2, \dots, m^{(j)},
 \end{aligned} \tag{16}$$

## 1.3 Modeling Optimization

---

### Problems

- Here comes the big decision.
- Do we save the same problem over and over
  - Then performance is what matters.
  - Use C++, Fortran, MPI, PERL.
- Do we solve the problem once or only a few times? (e.g. algorithmics class like this)
  - Productivity matters; use higher-level language.
  - Matlab for general scientific computing
  - AMPL or GAMS for optimization
- Domain-specific languages may blur the line
- ...

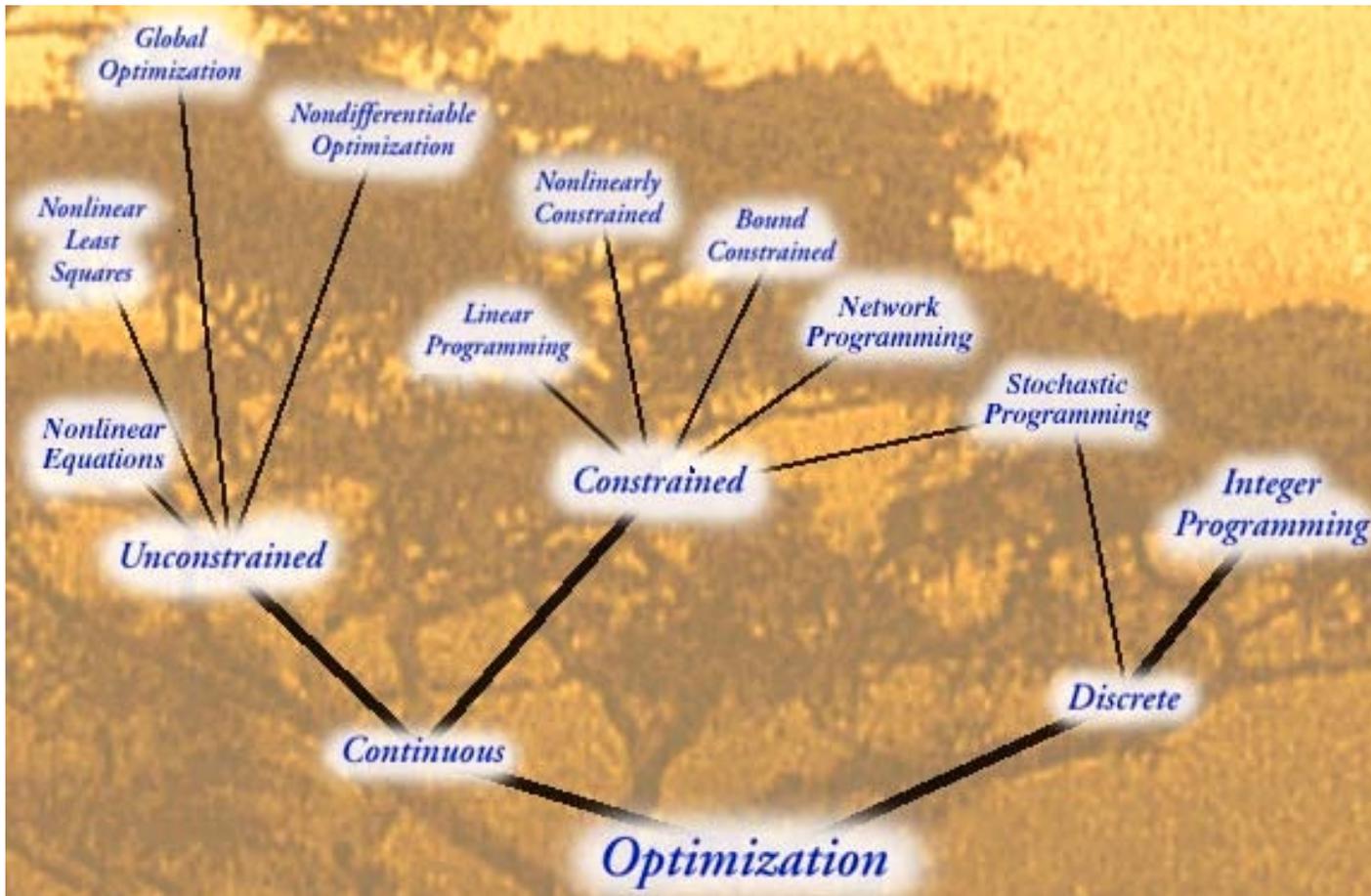
# Modeling: Ingredients

---

- Objective function
- Variables
- Constraints

**Find values of the variables  
that minimize or maximize the objective function  
while satisfying the constraints**

# MODELING: Structure (see NEOS)



Isn't any problem reducible to NLP? Sure, but it is very efficient to recognize and exploit structure.

# Modeling (Nonlinear) Optimization Problems

---

## AMPL & GAMS

- high level languages for nonlinear optimization
- interpret problem description, interface to solvers & returns results
- details of solver, derivatives & pre-solve are hidden from user
- modeling language (e.g. var, minimize, subject to, ...)
- programming language (e.g. while, if, ...)

- Has an exquisitely simple syntax, reminiscent of C (Kernighan is one of the authors), but adapted to optimization.
- Versions of it can be used even in parallel computation if done wisely.
- AMPL can be run:
  - Student version is free and easy to use.
  - Or you can run it using one of the online servers.

# AE1: AMPL Example 1

---

$$\begin{array}{ll} \min_{(x,y,z)} & z \\ \text{subject to} & g_0(x, y, z) = x^2 + 3x^4 - 2y^2 - z \leq 0, \\ & g_1(x, y, z) = -\frac{1}{2}(x^2 + y^2) + y^4 + 3xy - z \leq 0, \\ & g_2(x, y, z) = -2x^2 + y^2 - z \leq 0, \\ & g_3(x, y, z) = -\frac{1}{2}(x^2 + y^2) + x^2y^2 - 3xy - z \leq 0. \end{array}$$

## AE1: "Model"

```
var x1;
var x2;
var x3;
minimize objective: x3;
subject to constraint_1: x2**2-2*x1**2+3*x1**4-x3 <= 0;
subject to constraint_2: -2*x2**2+x1**2+x2**4-x3 <= 0;
subject to constraint_3: -0.5*x2**2-0.5*x1**2+3*x1*x2-x3<=0;
subject to constraint_4: -0.5*x2**2-0.5*x1**2-3*x1*x2+x1**2*x2**2-
x3<=0;
let x1:=1;
let x2:=1;
let x3:=1;
```

## AE2: “Commands”

---

```
solve;  
display x1,x2,x3;  
display constraint_1.dual, constraint_2.dual,  
constraint_3.dual, constraint_4.dual;
```

# AMPL: standalone

---

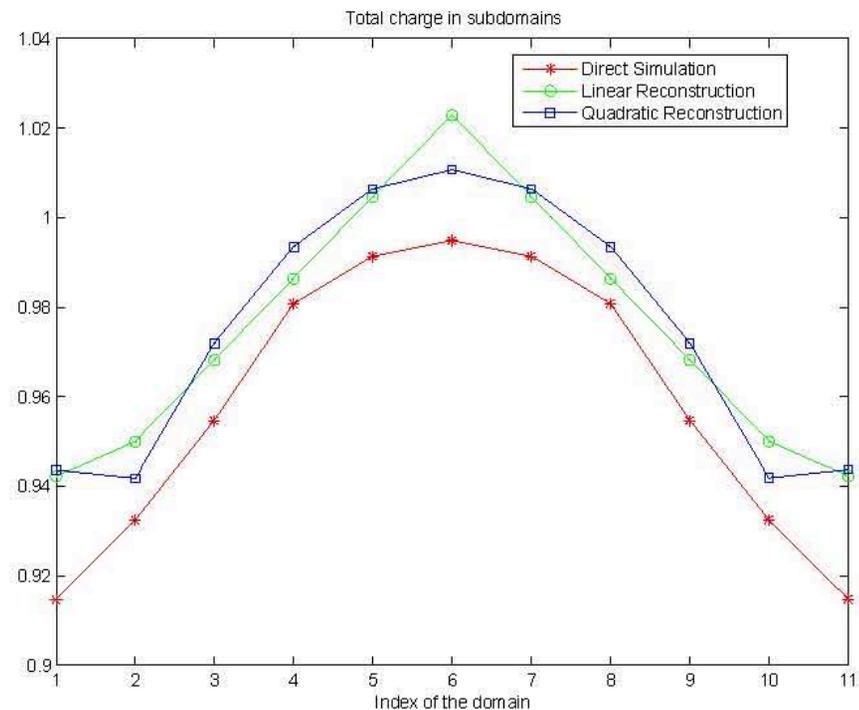
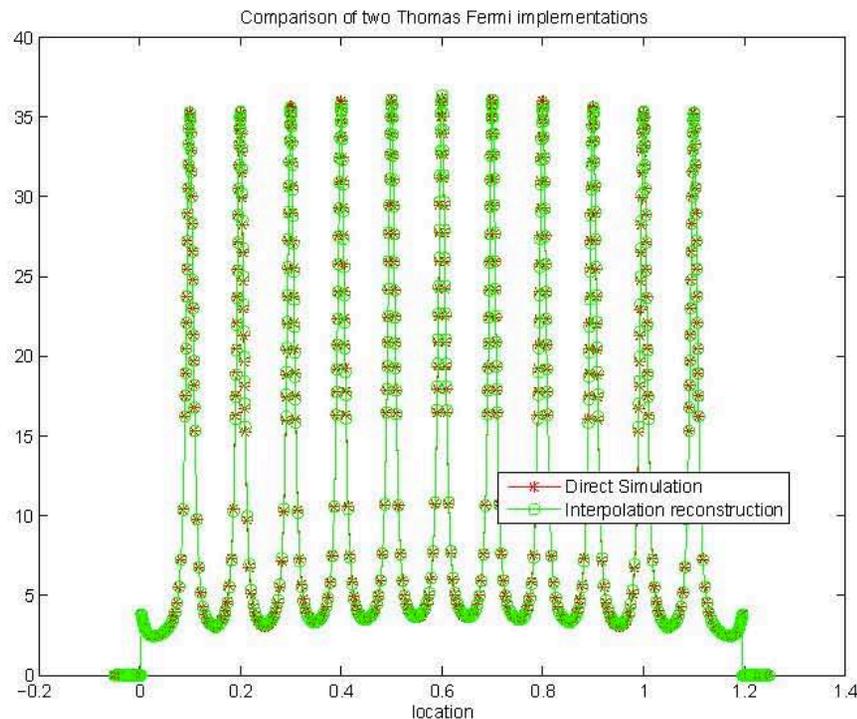
- EXPAND AND DEMO

# AMPL ONLINE:

---

- **EXPAND.** Discuss taxonomy. How do different solvers behave?

# AE2: AMPL example 2.



- One-dimensional Thomas-Fermi problem.
- Once you have created the model, you can even run it over the internet with the NEOS server.

# AE2: (Thomas-Fermi) Density Functional Theory

**Problem:** For a given atomic configuration, determine electronic density from using the variational principle.

$$\min_{\rho} E[\rho, \{\mathbf{R}_A\}]$$

$$\text{s.t. } \int \rho = N_e$$

$$E[\rho, \{\mathbf{R}_A\}] = E_{ne}[\rho, \{\mathbf{R}_A\}] + J[\rho] + K[\rho] + T[\rho] + V_{nn}(\{\mathbf{R}_A\})$$

$$T[\rho] = C_F \int \rho^{\frac{5}{3}}(\mathbf{r}) d\mathbf{r}, \quad K[\rho] = -C_x \int \rho^{\frac{4}{3}}(\mathbf{r}) d\mathbf{r}$$

$$E_{ne}[\rho, \{\mathbf{R}_A\}] = \sum_{A=1}^M \int \frac{Z_A \rho(\mathbf{r})}{\|\mathbf{R}_A - \mathbf{r}\|} d\mathbf{r}, \quad J[\rho] = \frac{1}{2} \iint \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|} d\mathbf{r} d\mathbf{r}'$$

# AE2: AMPL for Thomas Fermi

## DFT

```

#user-defined parameters
param n integer;
param ma integer;
param mg integer;
param dist;
param ratioGap;
param delta;
param cutoff integer;
param pi;
param Z;

#parameters of Thomas Fermi model
param CF:=0.3*(3*pi*pi)^(2/3);
param CX:=0.75*(3/pi)^(1/3);
param indexCutoff=2*(ma+mg)*cutoff;
#total number of nodes
param N:=(n+2)*(2*ma+2*mg);
param ZA {i in 1..n}=Z;

param Nelec:=sum {i in 1..n} ZA[i];

param atomicPosition {i in 1..n}:=dist*i; #atomic positions
param xloc {i in 1..(ma+mg)}:= if i <= ma then (i-0.5)/ma*ratioGap*dist*0.5 else (ma-0.5)/ma*ratioGap*dist*0.5 + 0.5*(1-ratioGap)*dist *(i-ma)/mg;
param xi {i in 1..N}:= if (i-2*(ma+mg)*floor((i-1)/(2*(ma+mg)))-ma-mg-0.5) < 0
    then floor((i-1)/(2*(ma+mg)))*dist-xloc[abs(i-2*(ma+mg)*floor((i-1)/(2*(ma+mg)))-ma-mg-1)]
    else floor((i-1)/(2*(ma+mg)))*dist+xloc[i-2*(ma+mg)*floor((i-1)/(2*(ma+mg)))-ma-mg];

var x;
var rho {i in 1..N} >=0;

```

# Objective function

minimize obj:  $\sum\{i \text{ in } 1..(N-1)\} 0.5*(CF*(rho[i]+\delta)^{(5/3)}-CX*(rho[i]+\delta)^{(4/3)} + CF*(rho[i+1]+\delta)^{(5/3)}-CX*(rho[i+1]+\delta)^{(4/3)})*(xi[i+1]-xi[i])$

#Kinetic and Exchange

-  $\sum\{i \text{ in } 1..(N-1), j \text{ in } 1..n\} ZA[j]*(rho[i+1]/\sqrt{(xi[i+1]-atomicPosition[j])^2+\delta})+\rho[i]/\sqrt{(xi[i]-atomicPosition[j])^2+\delta})*0.5*(xi[i+1]-xi[i])$

# Electron-Nucleus

+  $0.5 * 0.25*\sum\{i \text{ in } 1..(N-1), j \text{ in } \max(1,i-indexCutoff)..min(i+indexCutoff,N-1)\} (xi[i+1]-xi[i])*(xi[j+1]-xi[j])*(rho[i]*rho[j]/\sqrt{(xi[i]-xi[j])^2+\delta}) + rho[i]*rho[j+1]/\sqrt{(xi[i]-xi[j+1])^2+\delta}) + rho[i+1]*rho[j]/\sqrt{(xi[i+1]-xi[j])^2+\delta}) + rho[i+1]*rho[j+1]/\sqrt{(xi[i+1]-xi[j+1])^2+\delta})$ ;

# Electron-electron

subject to

constr:  $\sum\{i \text{ in } 1..(N-1)\} 0.5*(xi[i+1]-xi[i])*(rho[i]+\rho[i+1])=Nelec$ ;

# Then why study the algorithms at all if modeling is so advanced ?

---

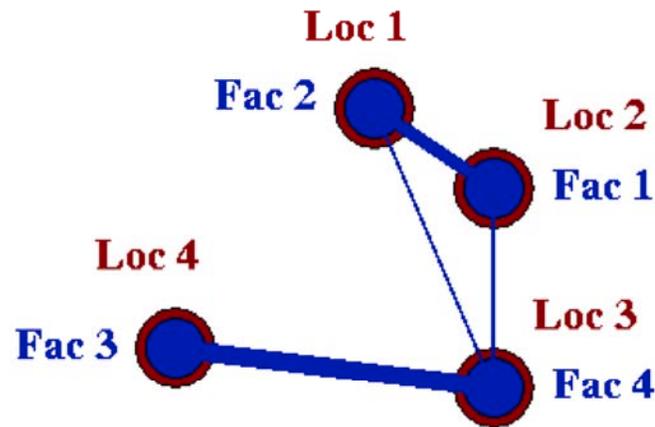
- Some problems CANNOT be solved well by current high-level languages.
  - Problems that have non smooth data (and need to understand limitations of algorithms if approximating them).
  - Problems that use non-intrinsics (e.g. max likelihood).
- One can take an enormous performance hit if the problem is large or has to be solved many times.

## 1.3 What is state of art in optimization?

---

- We can solve problems with  $10^9$ - $10^{12}$  variables LOCALLY.
- We have designed algorithms that make excellent use of massive parallelism (see unit commitment example).
- We can take advantage “smartly” of the latest architectures ...

# The Quadratic Assignment (facility location) Problem



$$\min_{p \in \Pi} \sum_{i=1}^n \sum_{j=1}^n f_{ij} d_{p(i)p(j)}$$

- How does one place  $n$  facilities at  $n$  location such that the total cost is minimized?
- Solved by the METANEOS team (Linderoth, Goux, Wright) and friends.
- The branch-and-bound procedure IDEAL for distributed computing (cloud, grid).
- The solution of the NUG30 – a 30 year open problem made the headlines.

# NUG30 statistics

---

Table 2: nug30 Run Statistics	
Average number of available workers	652.7
Maximum number of available workers	1009
Running wall clock time (sec)	597,872
Total cpu time (sec)	346,640,860
Average machine speed	0.56
Minimum machine speed	0.045
Maximum machine speed	1.074
Equivalent CPU time (sec) on an HP-C3000	218,823,577
Parallel Efficiency	93%
Number of times a machine joined the computation	19,063

# What we (optimizers) still need to figure out

---

- Global NONCONVEX (including integer) optimization problems are still far from being systematically solved on the large scale (many times we dance of joy when we solve  $N=30$  😊).
- Iterative solvers for constrained optimization beyond PDE constraints --- the preconditioning problems.
- There always appear new classes of problems that do not satisfy the usual assumptions – for example complementarity constraints.