Complementarity Problems: Examples and Preprocessing

Michael Ferris

University of Wisconsin, Computer Sciences

ferris@cs.wisc.edu

Jeffrey Horn

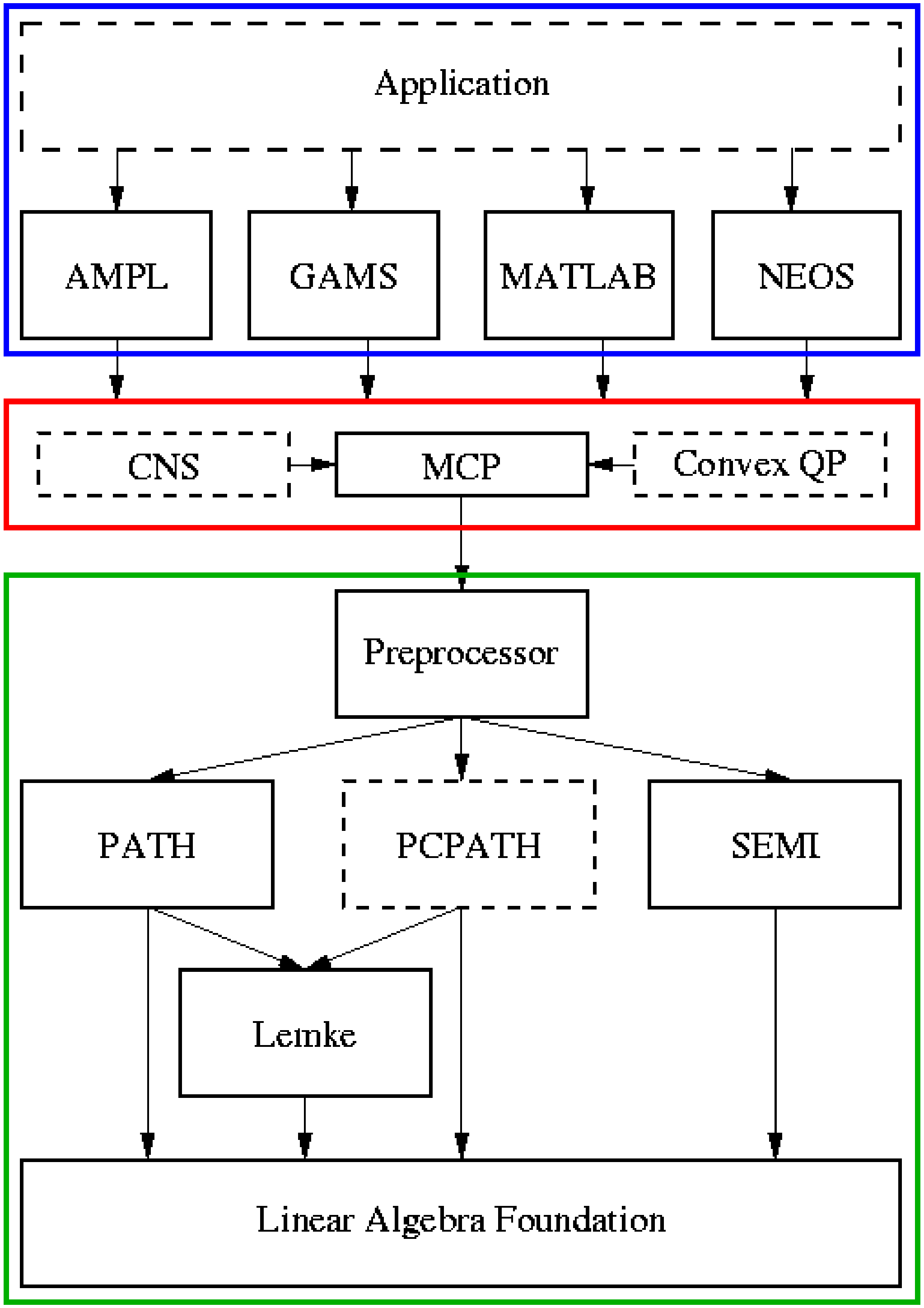
Todd S. Munson

Argonne National Labs

Supported by NSF and AFOSR

World Dairy Market Model

- Spatial equilibrium model of world dairy sector
  - 5 farm milk types
  - 8 processed goods
  - 21 regions
- Regions trade raw and processed goods
- Barriers to free trade
  - Import policies: quotas, tariffs
  - Export policies: subsidies
- Study impact of policy decisions
  - GATT/URAA
  - Future trade negotiations


Application
AMPL
GAMS
MATLAB
NEOS
CNS
MCP
Convex QP
Preprocessor
PATH
PCPATH
SEMI
Lemke
Linear Algebra Foundation

Formulation

- Quadratic program
  - Variables: quantities
  - Constraints: production and transportation
  - Objective: maximize net social welfare
- Difficulty is *ad valorem* tariffs
  - Tariff based on value of goods
  - Market value is multiplier on constraint
- Complementarity problem
  - Formulate optimality conditions
  - Market price is now a variable
  - Directly model *ad valorem* tariffs

NLP2MCP

$$\min\ f(x)\ s.t.\ g(x) \le 0, r \le x \le s$$

Introduce multipliers on constraints, form Lagrangian:

$$L(x,?) \equiv f(x) - l'g(x)$$

Symbolic differentiation to create first-order optimality conditions automatically:

$$F(x,l) \equiv \begin{bmatrix} \nabla_x L(x,l) \\ -\nabla_l L(x,l) \end{bmatrix} = \begin{bmatrix} \nabla f(x) - l'\nabla g(x) \\ g(x) \end{bmatrix}$$

Small number of blocks of equations even for large problems

F(x;õ) ? (x;õ) 2 [r;s]â [0;1)

```
variables x, y, z, obj;
equations cost, cons1, cons2;

cost..  obj =e= -x;
cons1..   y =g= exp(x);
cons2..   z =g= exp(y);

x.lo = 0; x.up = 100;
y.lo = 0; y.up = 100;
z.lo = 0; z.up = 10;

model nonlin /cost,cons1,cons2/;
* solve nonlin using nlp minimizing obj;
```

```
equations d_x, d_y, d_z;
positive variables m_cons1, m_cons2;

d_x .. -1-m_cons1*(-exp(x)) =e= 0;
d_y .. (-m_cons1)-m_cons2*(-exp(y)) =e= 0;
d_z .. -m_cons2 =e= 0;

model m_nonlin / d_x.x, d_y.y, d_z.z,
cons1.m_cons1, cons2.m_cons2 /;

solve m_nonlin using mcp;
```

Nonlinear Programs via MCP

- Matrix balancing via nonlinear optimization in GAMS. Class of problems where number of superbasic variables at solution grows rapidly with problem dimension.
  - GTAP(4) dataset: formulate first order conditions via NLP2MCP

	MINOS5	CONOPT	CONOPT2	PATH
20	2.902	2.405	1.497	0.308
25	14.028	8.766	4.868	0.377
30	25.861	24.484	10.492	0.548

  - PATH produces more reliable and precise solutions, particularly for larger models.

NLP2MCP

- GAMS: separate model parsing and differentiation
  - Perl script to parse model
  - AD tool to create MCP (as a GAMS model file)
  - Used to solve Matrix Balancing Problems, MARKAL-MACRO model (65 agencies in 37 countries)
- AMPL: use second order derivatives at solver level
  - Hookup to PATH exists
  - Tested on Hock/Schittkowski suite
- Both allow complementarity codes to be used for solving existing (practical) NLP models

NLP and MCP

NLP

$$\begin{array}{ll} \min & f(x) \\ \text{subject to} & g(x) \leq 0, r \leq x \leq s \end{array}$$

Lagrangian $\mathcal{L}(x,\lambda) := f(x) - \lambda^T g(x)$

MCP(F,ℓ,u) $F(x,\lambda) := \begin{bmatrix} \nabla_x \mathcal{L}(x,\lambda) \\ -\nabla_\lambda \mathcal{L}(x,\lambda) \end{bmatrix}$

$$\ell := \begin{bmatrix} r \\ -\infty \end{bmatrix}, u := \begin{bmatrix} s \\ 0 \end{bmatrix}$$

Automatic translation

- Source to source: instructional
- Symbolic differentiation of objective and constraint functions
- Transposition of constraint gradients

$$F(x,\lambda) = \begin{bmatrix} \nabla_x f(x) - \lambda^T \nabla_x g(x) \\ g(x) \end{bmatrix}$$

- Small number of blocks of equations (even in large scale setting)

Why convert to MCP?

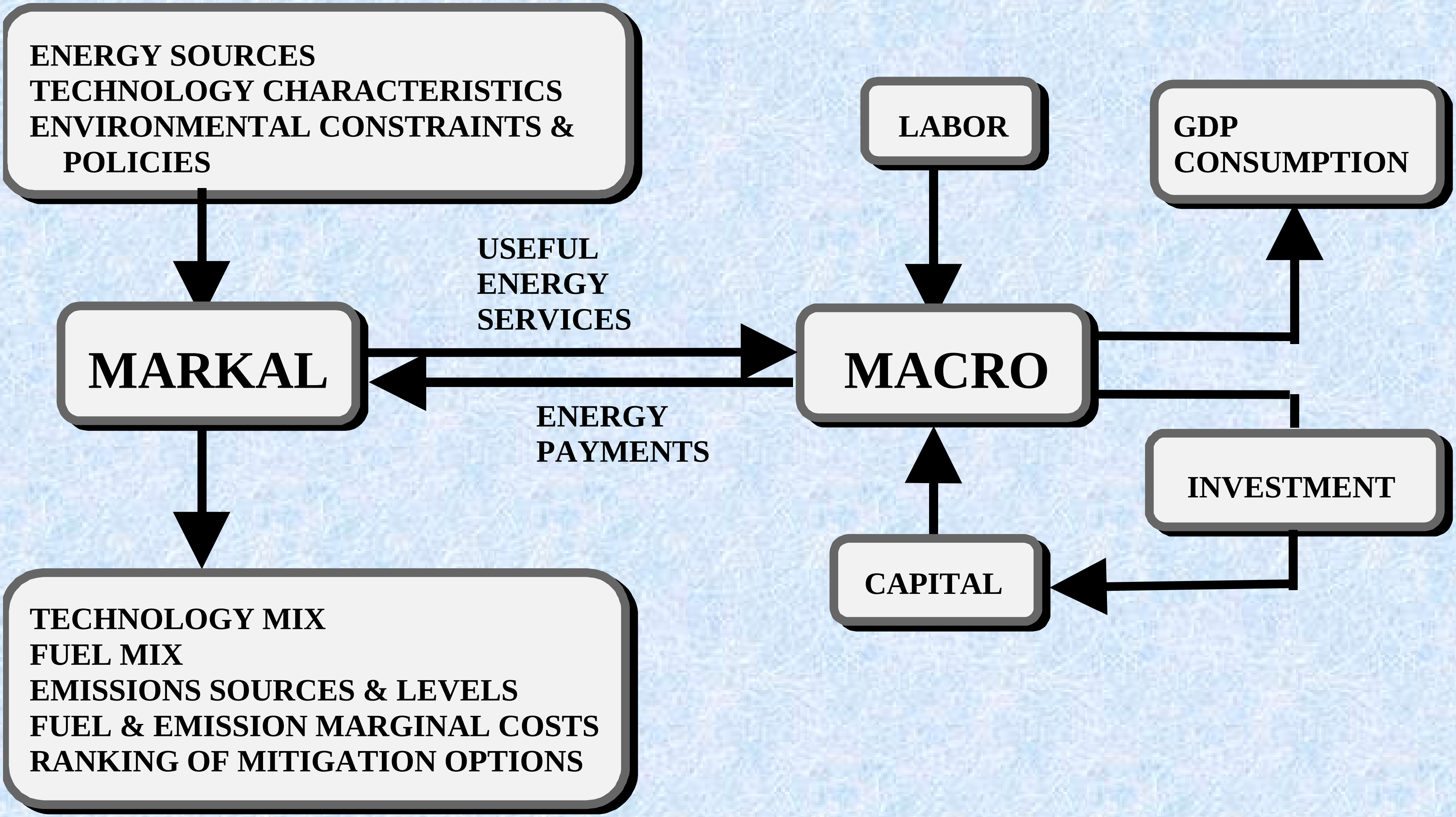
- Second order information available
- Remove “superbasic” dependence
- Incorporate into MPEC
- Exploit multiplier information

- Modify models to “formulate” correct complementarity constraints

Second Order Information

- GAMS provides first order derivatives
- MCP is formulated in GAMS with explicit first derivatives
  - GAMS provides second order derivatives
- Alternatively, PATHNLP: AMPL and GAMS provide second order derivatives to solvers allowing direct implementation

MARKAL-MACRO Overview



ENERGY SOURCES
TECHNOLOGY CHARACTERISTICS
ENVIRONMENTAL CONSTRAINTS & POLICIES

MARKAL

USEFUL ENERGY SERVICES

ENERGY PAYMENTS

MACRO

LABOR

GDP
CONSUMPTION

INVESTMENT

CAPITAL

TECHNOLOGY MIX
FUEL MIX
EMISSIONS SOURCES & LEVELS
FUEL & EMISSION MARGINAL COSTS
RANKING OF MITIGATION OPTIONS

MARKAL-MACRO

- Complex GAMS model defined by user front-end
- Multiple models within framework
- Large scale linear programs and nonlinear programs
  - Automatic conversion in less than 1 minute CPU
  - Model benchmarks at solution of LP/NLP
  - Solution found using PATH
  - Many new features possible in MCP model format

Other examples of usage

- Estimating AIDs equations using GME approach (nonlinear)
- Data is from mexico houseshould comsumption data
- sample size 100, 5 goods, 12 demographic variables

Example from Amos Golan, size 43,000; 2,400 iterations, 9,600 seconds with PATH. Not solvable by MINOS or CONOPT.

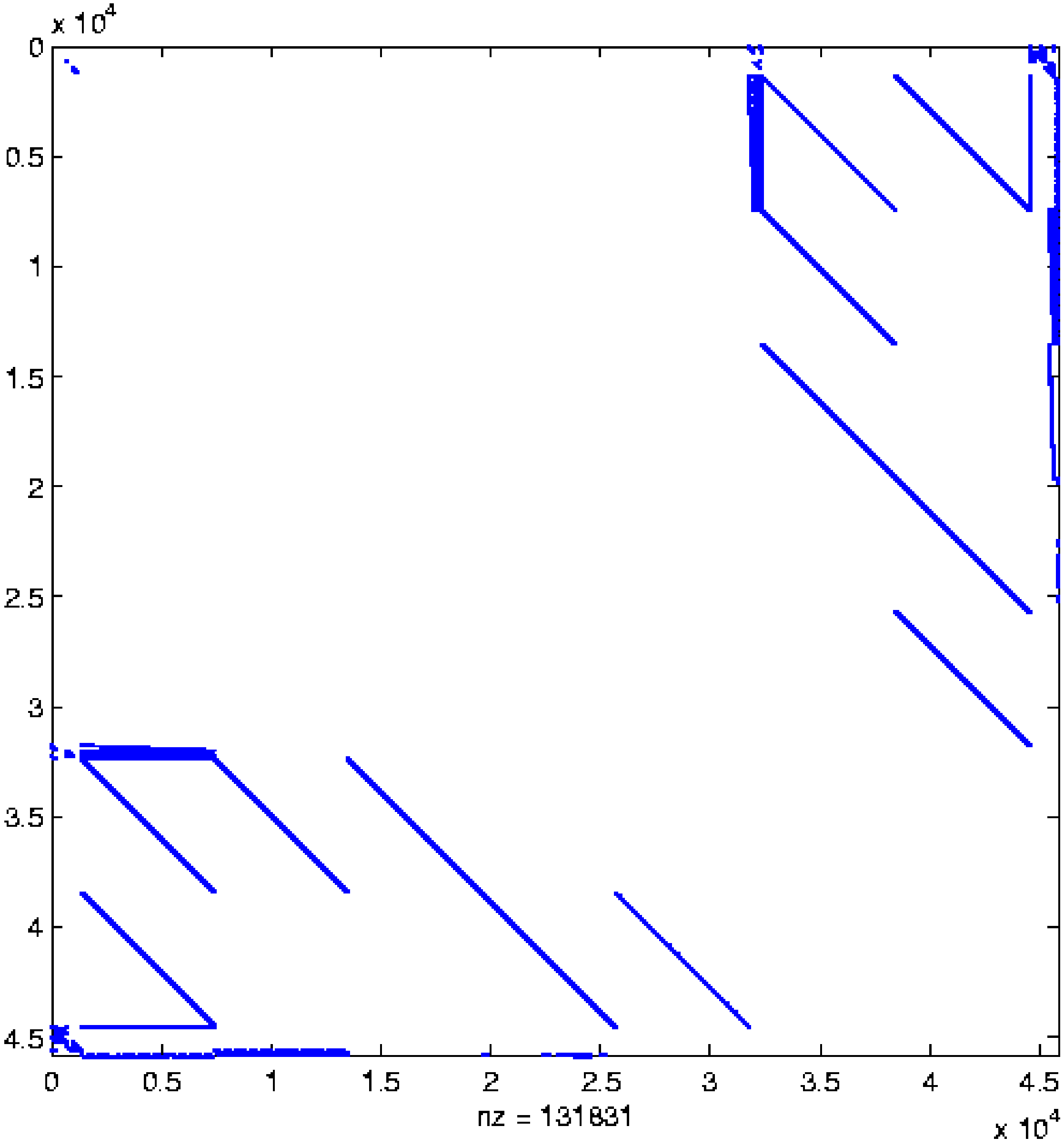
Negishi iterations

- Many models in the economics literature use Negishi iterations (essentially a Jacobi process) to process equilibria that are not integrable (thus not expressible as NLP)
- MCP derived from the NLP is easily modified to express the complete equilibrium problem directly
- Replace a sequence of NLP's by a single MCP
- Hand-coded work has shown these modifications improve robustness and speed
- Multiplier estimates at large time horizons are much better

World Dairy Market Model Statistics

- Quadratic program
  - 31,772 variables
  - 14,118 constraints
- Linear complementarity problem
  - 45,890 variables and constraints
  - 131,831 nonzeros

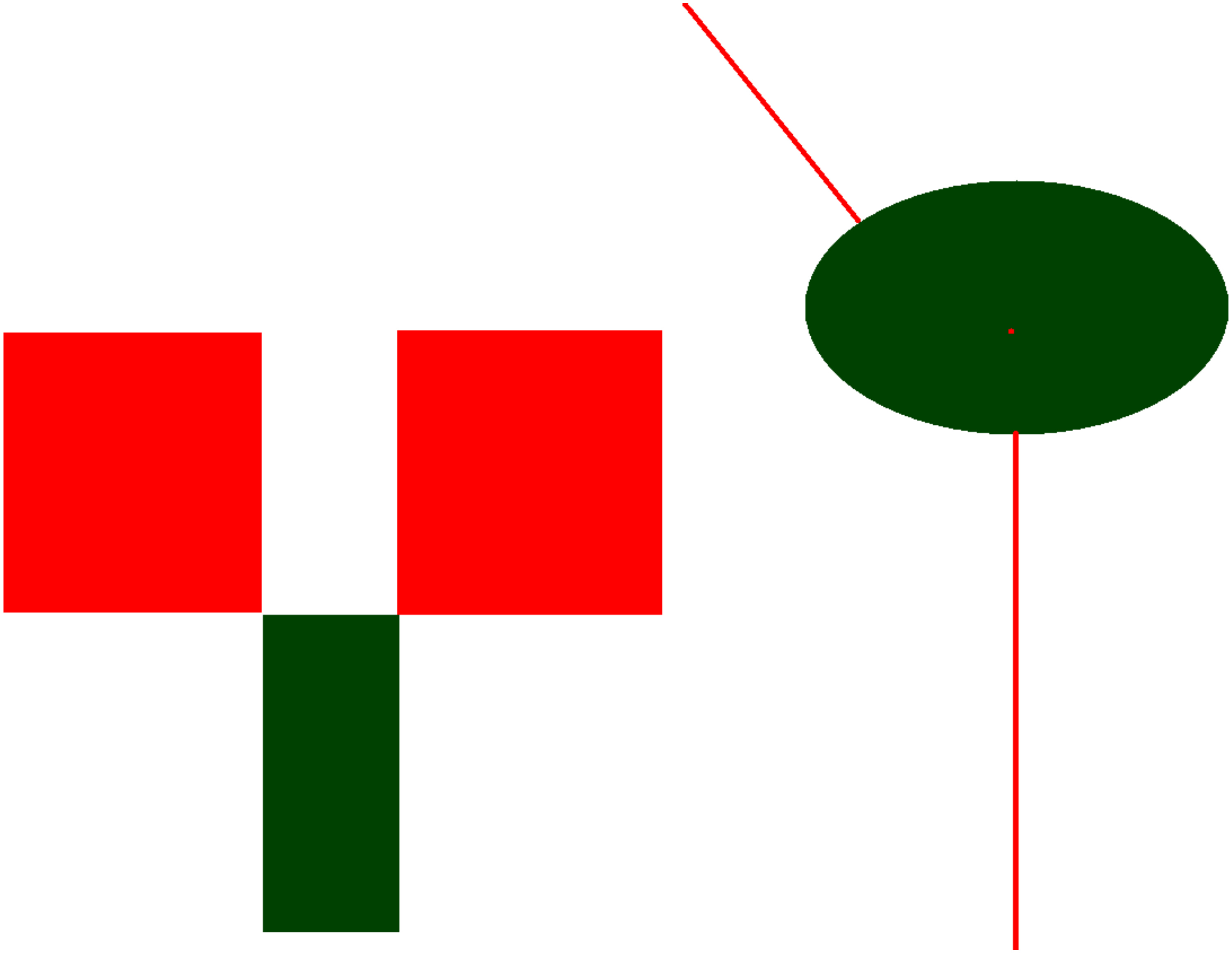
Jacobian



Preprocessing

- Discover information about a problem
- Use to reduce size and complexity
  - Improve algorithm performance
  - Detect unsolvable models
- Found in linear and mixed integer programming codes

Normal Cones



Definition

$N_C(z)$ = f y j y^T(c à z) ô 0; 8c 2 Cg

à F(z) 2 $N_C(z)$ ( ) à $F(z)^T$(c à z) ô 0; 8c 2 C

c = z + e_i =) F(z) õ 0

c = 0 and c = 2z =) z^TF(z) = 0

Relation to Normal Map

$$x \text{ à } x_{+} \; 2 \; N_{C}(x_{+})$$

$$\text{à } F(x_{+}) = x \text{ à } x_{+} \; 2 \; N_{C}(x_{+})$$

Thus $z=x_{+}$ solves the complementarity problem

The converse result can be proven slightly differently

Normal Cone

$$N_{[\ell,u]}(x) = \prod_{i=1}^{n} N_{[\ell_i,u_i]}(x_i)$$

and

$$N_{[\ell_i,u_i]}(x_i) = \begin{cases} \Re & \text{if } \ell_i = x_i = u_i \\ \Re_- & \text{if } \ell_i = x_i < u_i \\ \{0\} & \text{if } \ell_i < x_i < u_i \\ \Re_+ & \text{if } \ell_i < x_i = u_i \\ \emptyset & \text{otherwise.} \end{cases}$$

Variational Inequality Formulation

- Let $F : \Re^n \rightarrow \Re^n$ and $X \subseteq \Re^n$ be a nonempty,closed, and convex set

$$0 \in F(x) + N_X(x)$$

- Special Cases
  - Nonlinear Equations ($X \equiv \Re^n$)

$$F(x) = 0$$

  - Nonlinear Complementarity Problem ($X \equiv \Re^n_+$)

$$0 \leq F(x) \quad \perp \quad x \geq 0$$

Generalized Equation Representation

$$\begin{array}{ccc} 0 \in F(x) & + & N_{\Re^n_+}(x) \\ & \Leftrightarrow & \\ 0 \leq F(x) & \perp & x \geq 0 \end{array}$$

One-to-one correspondence between solutions to above generalized equation and the complementarity problem

Preprocessing

- Discover information about a problem
- Use to reduce size and complexity
  - Improve algorithm performance
  - Detect unsolvable models
- Main idea
  - Identify special structure
    - Polyhedral constraints
    - Separability
  - Use complementarity theory to eliminate variables

Polyhedral Constraints X

- X and $N_{\mathbf{X}}(\cdot)$ are geometric objects
- Free to choose algebraic representation
- Partition into two components: $X \equiv B \cap C$
  - B - simple bounds – treated specially by algorithm
  - C - polyhedral set
- Reduce complexity of C
- Must find X automatically

Polyhedral Structure

- Partition variables into (x, y)
- Identify skew symmetric structure

$$0 \in \begin{bmatrix} F(x) - A^T y \\ Ax - b \end{bmatrix} + \begin{bmatrix} N_{\Re^n_+}(x) \\ N_{\Re^n_+}(y) \end{bmatrix}$$

- Equivalent polyhedral problem (Robinson)

$$0 \in F(x) + N_{\Re^n_+ \cap \{x | Ax - b \geq 0\}}(x)$$

- Implementation finds a single constraint at a time

Relationship

1. If $(\bar{x}, \bar{y})$ solves box constrained problem then $\bar{x}$ solves the polyhedral problem
2. If $\bar{x}$ solves the polyhedral problem then there exist multipliers $\bar{y}$ such that $(\bar{x}, \bar{y})$ solves the box constrained problem

How do we calculate the multipliers, $\bar{y}$?

Calculating Multipliers

- Given an $\bar{x}$ solving the polyhedral problem
- Choose $\bar{y}$ solving the following linear program

$$\begin{array}{ll} \min_{y \in \Re^n_+} & y^T(A\bar{x} - b) \\ \text{s.t.} & 0 \in F(\bar{x}) - A^T y + N_{\Re^n_+}(\bar{x}) \end{array}$$

If $\bar{x}$ solves the polyhedral problem then

1. The linear program is solvable
2. Given any $\bar{y}$ in the solution set, $(\bar{x}, \bar{y})$ solves the box constrained problem

Separable Structure

- Partition variables into (x, y)
- Identify separable structure

$$0 \in \begin{bmatrix} F(x) \\ G(x,y) \end{bmatrix} + \begin{bmatrix} N_{\Re^n_+}(x) \\ N_{\Re^n_+}(y) \end{bmatrix}$$

- Reductions possible if either
  1. $0 \in F(x) + N_{\Re^n_+}(x)$ has a unique solution
  2. $0 \in G(x,y) + N_{\Re^n_+}(y)$ has solution for all x
- Theory provides appropriate conditions
- Solve F and G sequentially

Presolve

1. Identify a constraint with skew symmetric property
2. Convert problem into polyhedral form
3. Modify representation of polyhedral set
   - Singleton and doubleton rows
   - Forcing constraints
   - Duplicate rows
4. Recover box constrained problem with reduced size
   - Multipliers fixed and function modified
   - Additional polyhedral constraints uncovered
5. Repeat 1–4 until no changes
6. Identify separable structure

Example

- Original problem

$$0 \in \begin{bmatrix} x^2 - y - 1 \\ x - 1 \end{bmatrix} + \begin{bmatrix} N_{\Re_+}(x) \\ N_{\Re_+}(y) \end{bmatrix}$$

- Polyhedral problem

$$0 \in x^2 - 1 + N_{\Re_+ \cap \{x | x - 1 \geq 0\}}(x)$$

- Equivalent problem

$$0 \in x^2 - 1 + N_{[1,\infty)}(x)$$

Example (continued)

- $0 \in x^2 - 1 + N_{[1,\infty)}(x)$ has one solution $\bar{x} = 1$
- Solve optimization problem

$$\begin{array}{ll} \min_{y \in \Re_+} & y^T(\bar{x} - 1) \\ \text{s.t.} & 0 \in \bar{x}^2 - y - 1 + N_{\Re_+}(\bar{x}) \end{array}$$

- Equivalent model

$$\begin{array}{ll} \min_{y \in \Re_+} & 0 \\ \text{s.t.} & y = 0 \end{array}$$

- Obtain $\bar{y} = 0$
- Solution is $(1, 0)$

Example

- Original problem

$$0 \in \begin{bmatrix} x - y - 1 \\ x + 1 \end{bmatrix} + \begin{bmatrix} N_{\Re_+}(x) \\ N_{\Re_+}(y) \end{bmatrix}$$

- Polyhedral problem

$$0 \in x - 1 + N_{\Re_+ \cap \{x | x+1 \geq 0\}}(x)$$

- Equivalent problem

$$0 \in x - 1 + N_{\Re_+}(x)$$

Example (continued)

- $0 \in x - 1 + N_{\Re_+}(x)$ has one solution $\bar{x} = 1$
- Solve optimization problem

$$\begin{array}{ll} \min_{y \in \Re_+} & \langle \bar{x} + 1,\ y \rangle \\ \text{s.t.} & 0 \in \bar{x} - y - 1 + N_{\Re_+}(\bar{x}) \end{array}$$

- Equivalent model

$$\begin{array}{ll} \min_{y \in \Re_+} & 2y \\ \text{s.t.} & y = 0 \end{array}$$

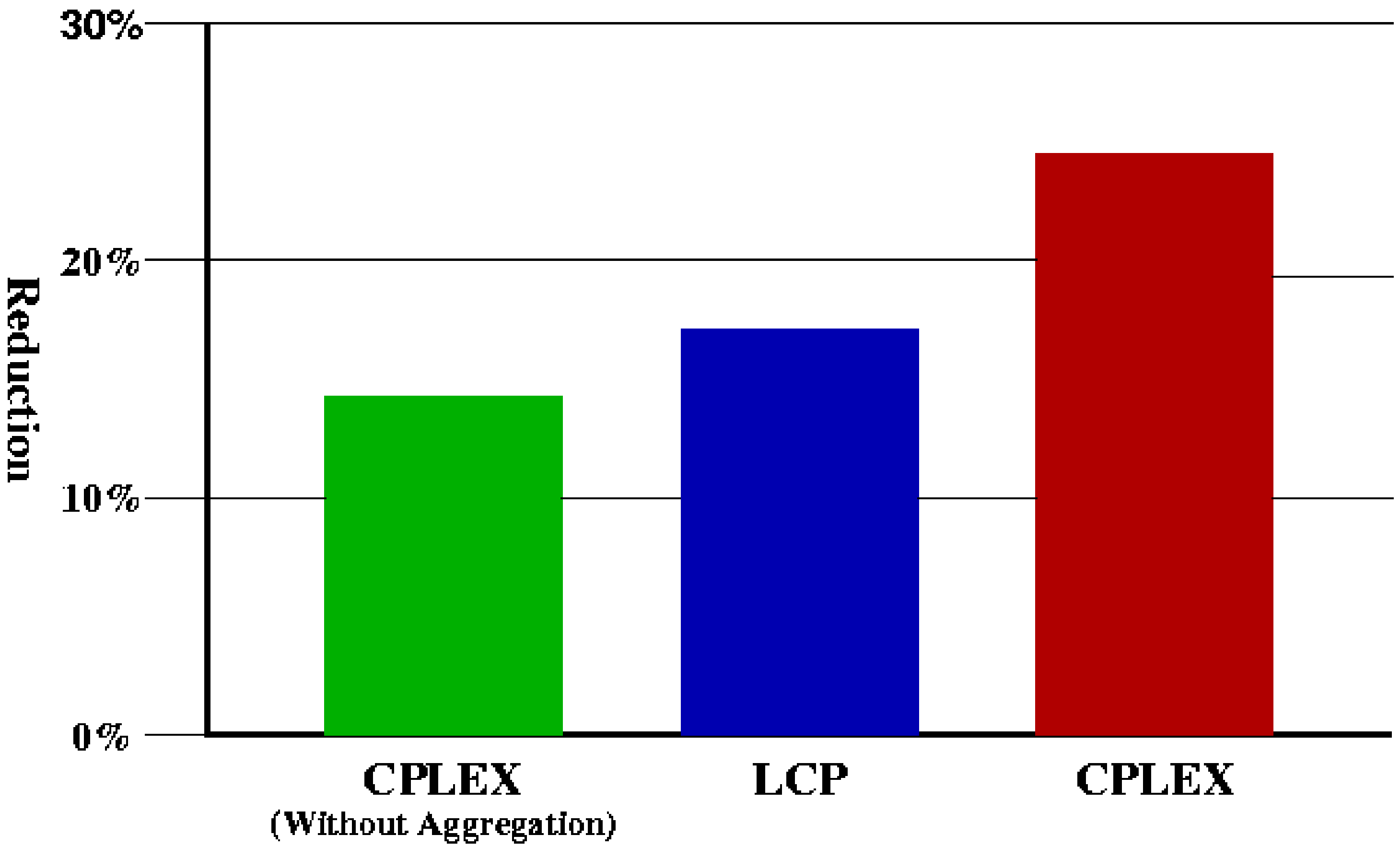
- Obtain $\bar{y} = 0$
- Solution is $(1, 0)$

Availability of Preprocessor

- PATH 4.x and SEMI for GAMS and AMPL
  - Finds polyhedral structure
  - Exploits separable structure
- Capability exists in other environments
  - User needs to provide information
  - Listing of linear/nonlinear elements in Jacobian
  - Optional - interval evaluation routines

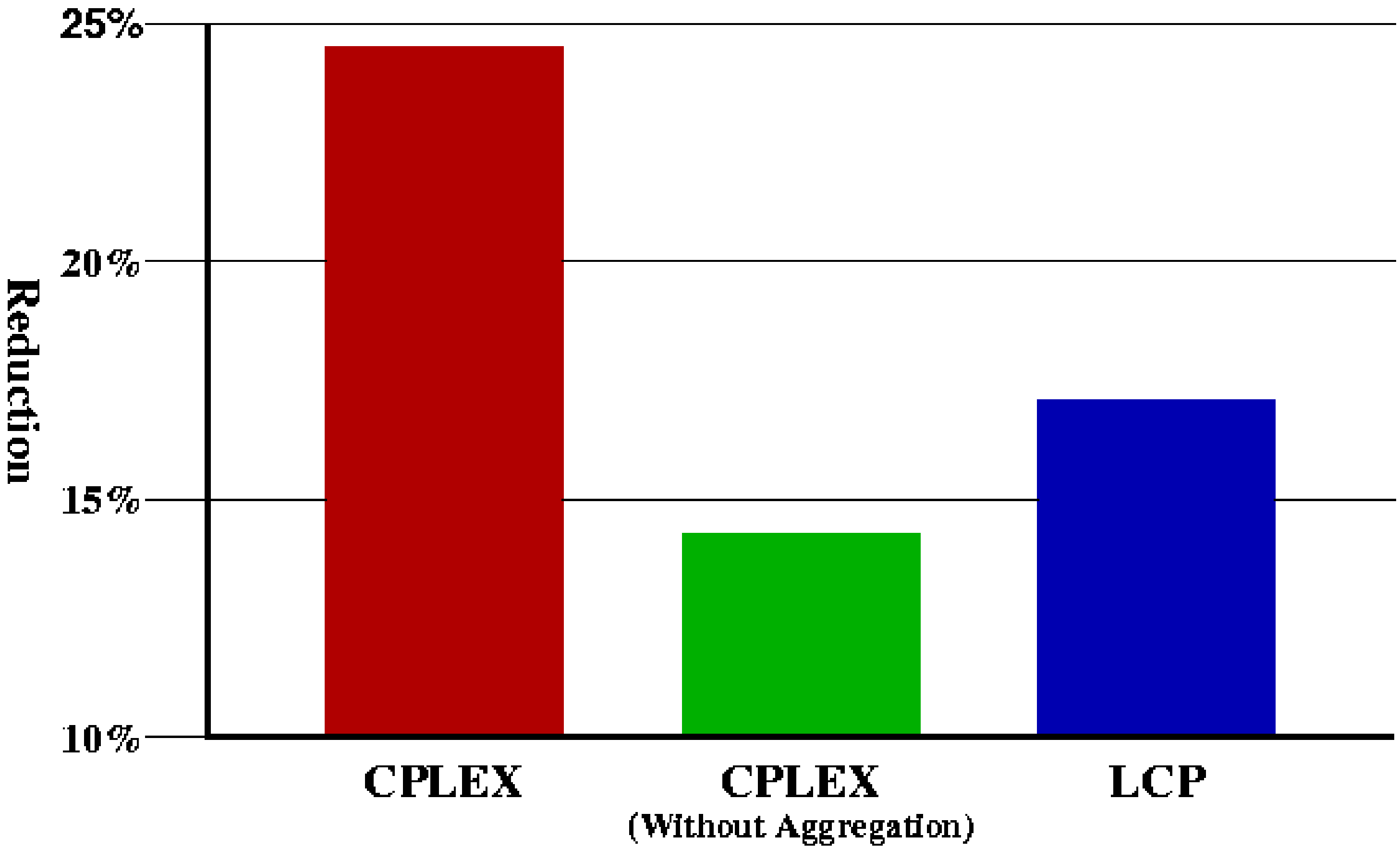
Results: Linear Programs

- Formulated first order conditions of NETLIB problems
- Polyhedral structure *not* supplied to LCP preprocessor



Results: Linear Programs

- Formulated first order conditions of NETLIB problems
- 73 models used



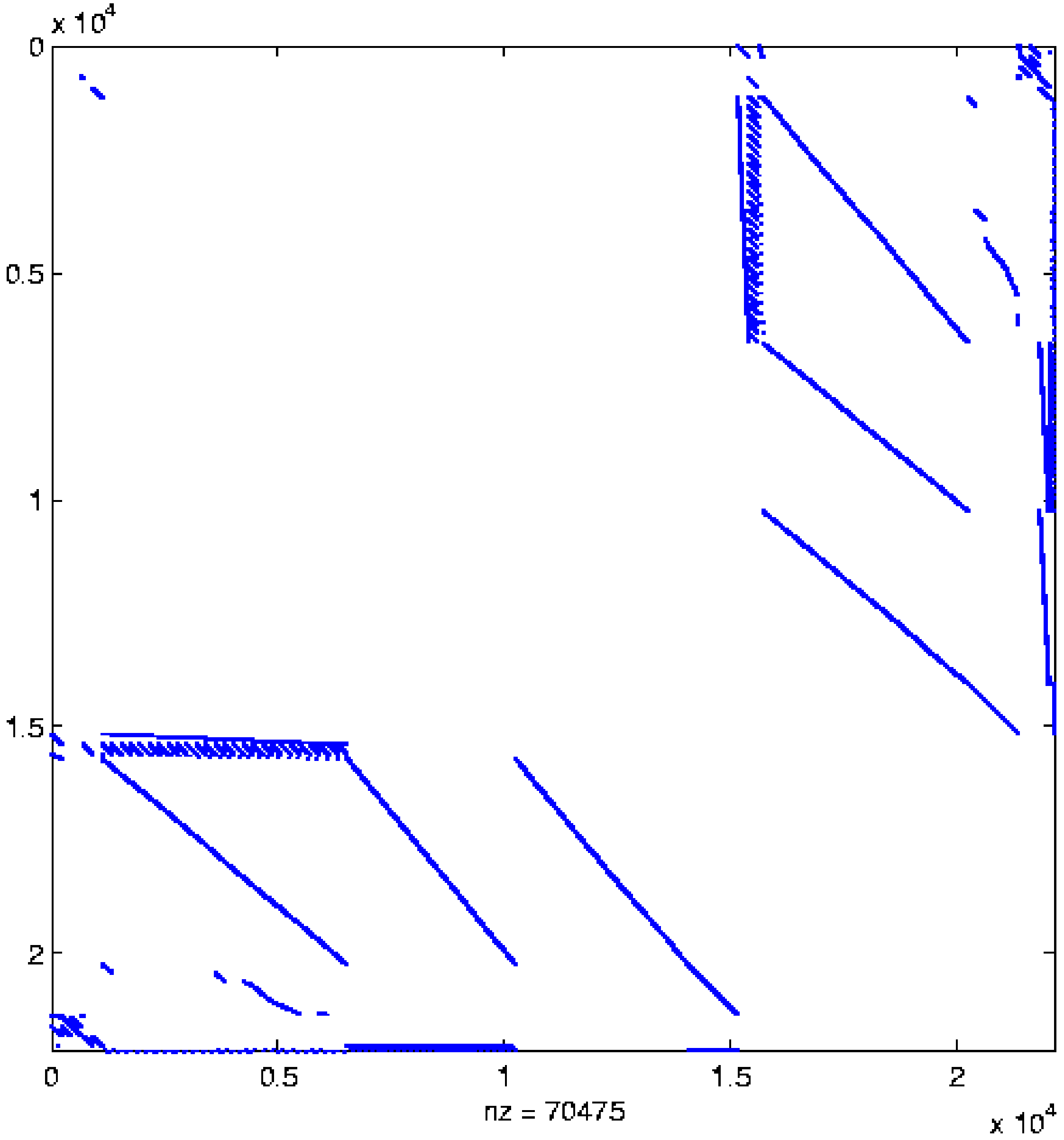
Results: Quadratic Programs

- Solve optimality conditions
- Synthetic models
  - NETLIB problems with $\frac{1}{2}\|x\|^2$ added to objective
  - Selected 8 models
  - 17.6% reduction in size
  - 29.2% reduction in time with PATH 4.x
- World Dairy Market Model
  - Failed on original model (4.5 hours)
  - 70.4% reduction in size
  - Solved preprocessed model 23 minutes
  - 91.5% reduction in time with PATH 4.x

World Dairy Market Model Statistics

- Quadratic program
  - 31,772 variables
  - 14,118 constraints
- Linear complementarity problem
  - 45,890 variables and constraints
  - 131,831 nonzeros
- Preprocessed problem
  - 22,159 variables and constraints
  - 70,475 nonzeros

Preprocessed Jacobian



World Dairy Market Model Results

- Note: want to analyze large number of scenarios
- Gauss-Seidel method
  - Solves 96 quadratic programs
    - Uses MINOS with nonstandard options
  - Approximates equilibrium in 42 minutes
- Complementarity formulation
  - Solves a single complementarity problem
  - Computes equilibrium in
    - 117 minutes without preprocessing
    - 21 minutes with preprocessing
    - 11 minutes with nonstandard options
  - Obtain more accurate result in less time!

Results: Nonlinear Complementarity Problems

- Models from GAMSLIB and MCPLIB
- Selected 6 models
- 9.7% reduction in size
- 15.3% reduction in time with PATH 4.x

Simple Example: NLP

$$\min x \text{ subject to } \sqrt{x} \geq 0, x \geq 0$$

Note that need $x \geq 0$ for problem to be well posed, (convex, CQ).

KKT point (from NLP2MCP):

$$\begin{aligned} 0 \leq x \quad &\perp \quad 1 - u/(2\sqrt{x}) \geq 0 \\ 0 \leq u \quad &\perp \quad \sqrt{x} \geq 0 \end{aligned}$$

Ill posed - cannot evaluate F at solution (0,0) using IEEE

AMPL Format

```
var x := 1;
var u := 2;

subject to

f:
   0 <= x complements 1 - u/(2*sqrt(x)) >= 0;

g:
   0 <= u complements sqrt(x) >= 0;

solve;
```

Ill Posed Problems

Ideally, we would like the following to hold:

1. a solution exists,
2. the solution is unique,
3. the solution depends continuously on the problem data.

Otherwise, problem is ill posed.

$$u = 2\sqrt{x}$$

$$(0, 0)$$

yes

But F is not defined everywhere on $[\ell, u]$.

Typical behavior in Economic Models (CES functions).

So what to do?

Von Thunen Land Model

- central market - prices p of commodities
- concentric regions of land - land rent w_r
- labour - wage rate w_L
- transport - porter rate w_P

Key variables: prices of commodities, land, labour and transport

Agents: landowner, worker, porter (closed economy)

Economy: Demand, given prices

Each agent demands goods to maximize utility subject to budget constraint.

Closed form solution (for Cobb-Douglas utility):

$$\frac{\alpha_c^a \langle w^a, e^a \rangle}{p_c}$$

Assumes prices of commodities p_c are positive.

e^a is endowment bundle for agent a.

Economy: Supply, given prices

Each sector supplies goods that maximize profit subject to technology constraint.

Closed form solution (for CES production functions):

$$\frac{\beta_c w_L^{\beta_c} w_r^{1-\beta_c}}{w_L}$$

Assumes prices of inputs w_L, w_r are positive.

Spatial Price Equilibrium

$$0 \le x_{c,r} \perp \text{ prod cost } + \text{ trans cost } \ge p_c$$

Theory: Homogeneous in prices, fix a numeraire

Walras Law

- labour: $0 \leq w_L \perp$ labour used $\leq$ labour endowed
- land: $0 \leq w_r \perp$ ....
- transport: $0 \leq w_P \perp$ ....

Now Walras Law (Supply = Demand) gives complementarity problem where variables p and w may go to zero, and problem is ill posed.

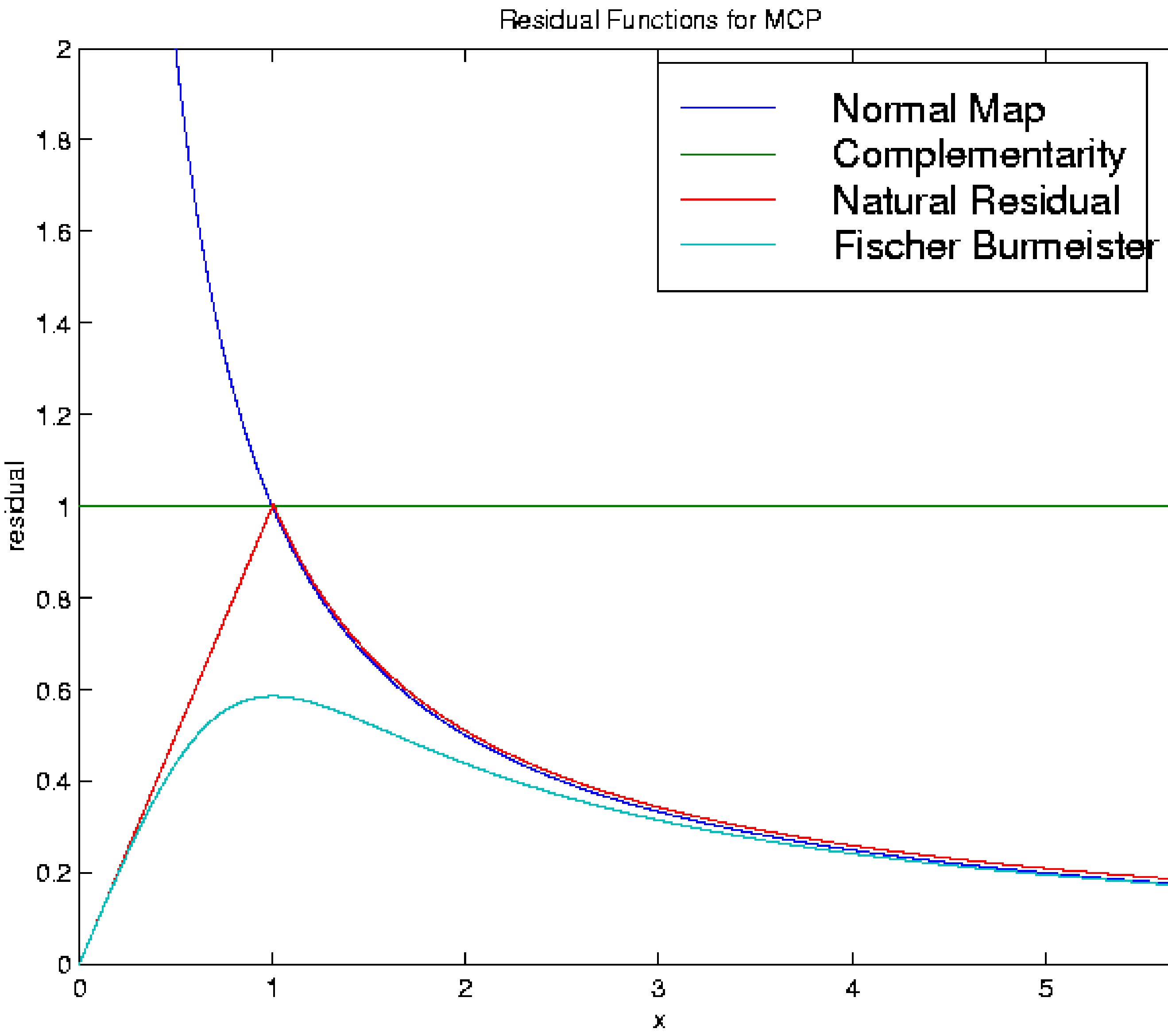
Are we done?

There are a number of measures of optimality available:

- Natural Residual: $x - (x - F(x))_+$
- Complementarity: $\approx x'F(x)$
- Normal Map: $F(x_+) + x - x_+$
- Fischer Function: $\phi(x_i, F_i(x))$
- $\nabla\Phi$: $\Phi(x) := \frac{1}{2}\sum \phi(x_i, F_i)^2$

Example:

$$0 \leq x \quad \perp \quad 1/x \geq 0$$


Residual Functions for MCP
residual
x
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
1
2
3
4
5
6
Normal Map
Complementarity
Natural Residual
Fischer Burmeister

Solver requirements: input

- Preprocessing (problems: optimization, nonlinear - in progress)
- Infeasible problem - check if $[\ell, u]$ is nonempty (general VI is harder)
- Large entries in Jacobian / scaling

Is it possible to differentiate between the following cases:

1. Ill posed
2. Ill conditioned
3. Singular

Solver requirements: output

- Return best solution found to user (nonmonotone search or restarts)
- Report worst violated equation at each iteration
- Report which equations are violated at solution
- Do multiple checks of optimality criteria
- Is F defined at solution?
- Check linear feasibility?

Scaling

- Responsibility of modeler: scaling works in MCP models from GAMS
- Default is to scale residual in PATH and also to scale the rows using diag($\nabla F(x^0)$). Simple test example of multiply repeated model.
- What about large Jacobian entries?
  - Currently reported at solution
  - At initial point, quit and force modeler to scale
  - or, carry on regardless

Solver requirements: singular models

Many models generate singularities during development:

- Rank deficiency (report constraint name) require good factorization code
- Artificial variables
- Proximal perturbation
- Empty rows and/or empty columns? (Preprocessor or linearization)

Domain errors

- PATH: $F(x_+)$ - only evaluates F and ∇F on $[\ell, u]$
- SEMI: $\Phi(x^k)$ - only guarantees $x \in [\ell, u]$ at solution; can evaluate F outside box - potential problems

All domain errors reported to modeler from PATH (with explicit naming of equations)

No domain errors in Jacobian reported to modeler

Both codes use backtracking to overcome this problem

Key issues

- Use problem knowledge to avoid generating bad models
- Solve the problem
- If fail, return best point found (different to NLP codes, since typically the last solution is the best ever found)
- Analyse problem at best point found
- Domain errors: functions and/or gradients