An Algorithm for the Approximate and Fast Solution of Linear Complementarity Problems.

José Luis Morales[1 3] Jorge Nocedal[1] Mikhail Smelyanskiy[2]

[1]EECS, Northwestern University

[2]The INTEL Corporation

[3]On sabbatical leave from ITAM, México.

Argonne National Laboratory. March 21, 2007

Aims of the talk

In this talk we explore the solution of mixed symmetric linear complementarity problems (mLCP).
The focus is on the fast and approximate solution of medium to large size problems.
Source of mLCPs:

- lubrication problems
- computer game simulations (examples)
- American option pricing

We study three classes of methods: a) Projected Gauss-Seidel; b) IPMs c) Projected Gauss-Seidel + subspace minimization steps.

Formulation of the problem

The form of the linear complementarity problem considered here is

$$\begin{aligned} Au + Cv + a &= 0 \\ C^T u + Bv + b &\geq 0 \\ v^T (C^T u + Bv + b) &= 0 \\ v &\geq 0, \end{aligned}$$

where the variables of the problem are u and v. The matrix

$$\begin{bmatrix} A & C \\ C^T & B \end{bmatrix},$$

is an $n \times n$ positive definite matrix.

Organization of the talk

1. Motivation.
2. Structure of the mLCP.
3. Notation.
4. Brief description of the methods.

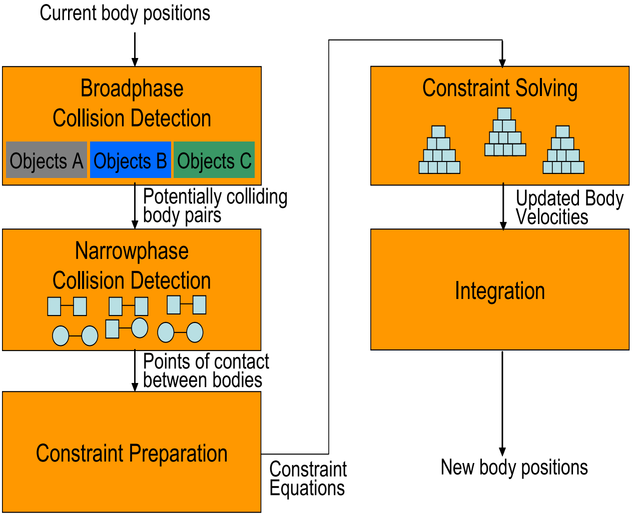
    Projected Gauss-Seidel (PGS).
    Interior point methods (IPM).
    PGS + subspace minimization

5. Numerical experiments.
6. Observations.

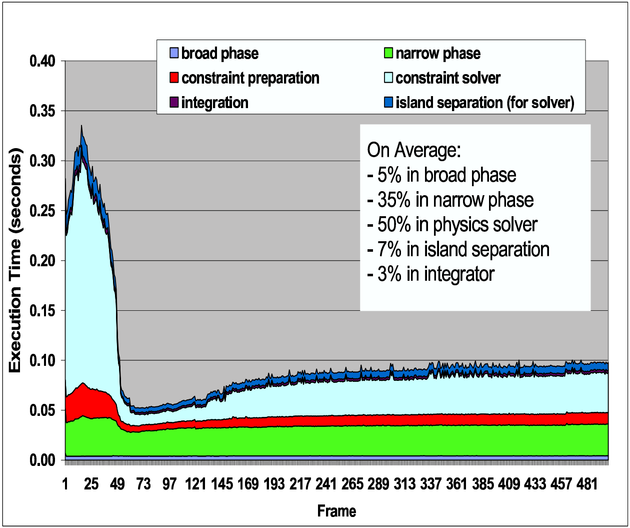
Motivation

- mLCP's come from modeling contact forces in physical simulation used in computer games industry. Limited amount of resources: a) CPU time (real time); b) memory; c) stability; d) low accuracy.
- Modern systems: a model that takes into account interactions between pairs of bodies. Each interaction is modeled by 1 inequality that amounts for contact, and 2 equalities that model friction between the bodies.
- More realism in games demands the incorporation of more complex models in terms of physics.

Physical Simulation Pipeline



Time breakdown of Physical Simulation.



Structure of the mLCP.

- The matrix has the form

$$JDJ^T + E,$$

where J is rectangular where rows correspond to constraints, and columns correspond to bodies.

- D is a block-diagonal matrix that incorporates inertia into the model.
- E is a diagonal matrix with positive entries. E has some physical meaning

$$(JDJ^T + E)\lambda = e.$$

- λ is the vector of contact forces.
- Examples. Open Dynamics Engine (ODE). Castle destruction demo.

A useful reformulation

$$
\begin{aligned}
Au + Cv + a &= 0 \\
C^T u + Bv + b &= w \\
v^T w &= 0 \\
v, w &\geq 0.
\end{aligned}
$$

A standard LCP

$$
\begin{aligned}
Mv + q &= w \\
v^T w &= 0 \\
v, w &\geq 0,
\end{aligned}
$$

The QP connection

$$M = B - C^T A^{-1} C, \qquad q = b - C^T A^{-1} a.$$

$$\begin{array}{lll} \min_{v} & \phi(v) = & \frac{1}{2} v^T M v + q^T v \\ \text{s.t.} & v \geq 0, & \end{array}$$

Projected Gauss-Seidel. 70's

$$A = A_L + A_D + A_U, \qquad B = B_L + B_D + B_U,$$

Given u^k, $v^k \geq 0$, consider the auxiliary linear complementarity problem

$$
\begin{aligned}
(A_L + A_D)u + A_U u^k + Cv^k + a &= 0 \qquad (1)\\
C^T u + (B_L + B_D)v + B_U v^k + b &\geq 0 \qquad (2)\\
v^T(C^T u + (B_L + B_D)v + B_U v^k + b) &= 0 \qquad (3)\\
v &\geq 0 \qquad (4)
\end{aligned}
$$

Deriving PGS for mLCPs

Let us define (u^{k+1}, v^{k+1}) to be a solution of (1)-(4). Then we have from (1)

$$A_D u^{k+1} = -a - A_L u^{k+1} - A_U u^k - C v^k.$$

Let us define

$$\hat{b} = b + C^T u^{k+1} + B_U v^k.$$

Then we can write (2)-(4) as

$$\begin{aligned}
\hat{b} + (B_L + B_D)v \quad & \geq 0 && (5)\\
v_i[\hat{b} + (B_L + B_D)v]_i \quad & = 0, \quad i = 1, 2, \ldots, n_b && (6)\\
v \quad & \geq 0 && (7)
\end{aligned}$$

We can satisfy (5)-(6) by defining the ith component of the solution v^{k+1} so that

$$[(B_L + B_D)v^{k+1}]_i = -\hat{b}_i;$$

We cannot guarantee that $v_i^{k+1} \geq 0$, but if it is not, we can simply set $v_i^{k+1} = 0$.

Algorithm PGS

Projected Gauss-Seidel (PGS) Method

Initialize $u^0, v^0 \geq 0$.
For $k = 0, 1, 2, \ldots,$ until a convergence test is satisfied
  for $i = 1, \ldots, n_a$
    compute u_i^{k+1} by previous slide
  end
  for $i = 1, \ldots, n_b$
    compute v_i^{k+1} by previous slide
  end
End

Numerical results with PSOR

n_a	n_b	nz(JDJ^T)	cond(JDJ^T)	10^{-1}	10^{-2}
7	18	162	5.83e+01	4	6
8	45	779	2.92e+03	17	120
8	48	868	2.38e+03	17	111
235	1 044	14 211	4.58e+04	61	312
449	1 821	28 010	4.22e+04	132	414
907	5 832	176 735	5.11e+07	21	16 785
948	7 344	269 765	9.02e+07	3 123	>50 000
966	8 220	368 604	9.19e+07	1 601	39 103
976	8 745	373 848	6.45e+07	7 184	>50 000
977	9 534	494 118	1.03e+08	1 246	>50 000

Identifying the active set

name	$k = 2$	$k = 20$	$k = 1\,000$	$k = 10\,000$
1	3/4	4/4		
2	7/8	7/8		
3	8/10	8/10		
4	12/58	40/58	58/58	
5	156/254	233/254	254/254	
6	1 253/1 512	1 301/1 512	1 399/1 512	1 471/1 512
7	1 504/1 828	1 523/1 828	1 614/1 828	1 707/1 828
8	2 112/2 321	2 106/2 321	2 178/2 321	2 253/2 321
9	1 728/2 158	1 743/2 158	1 870/2 158	1 976/2 158
10	2 513/2 728	2 495/2 728	2 578/2 728	2 670/2 728

The proposed method

A few iterations of PGS give:

$$
\begin{aligned}
Au + \hat{C}\hat{v} + a &= 0 && (8)\\
\hat{C}^T u + \hat{B}\hat{v} + \hat{b} &\geq 0 && (9)\\
\hat{v}^T(\hat{C}^T u + \hat{B}\hat{v} + \hat{b}) &= 0 && (10)\\
\hat{v} &\geq 0, && (11)
\end{aligned}
$$

where

$$
\hat{C} = C[1 : n_b, m+1 : n_b], \qquad \hat{B} = B[m+1 : n_b, m+1 : n_b]
$$
$$
\hat{b} = b[m+1 : n_b], \qquad \hat{v} = v[m+1 : n_b];
$$

The proposed method

Since we follow an active set approach and our prediction is that $\hat{v} > 0$ at the solution, we set the complementarity term in (10) to zero. Thus (8) gives the reduced system

$$Au + \hat{C}\hat{v} + a = 0 \qquad (12)$$

$$\hat{C}^T u + \hat{B}\hat{v} + \hat{b} = 0, \qquad (13)$$

together with the condition $\hat{v} \geq 0$.

We compute an approximate solution of this problem by solving (12)-(13) and then projecting the solution $\hat{v}$

$$\hat{v} \leftarrow \max(0, \hat{v}). \qquad (14)$$

The components of $\hat{v}$ that are set to zero by the projection operation (14) are then removed to obtain a smaller vector $\check{v}$.

The proposed algorithm

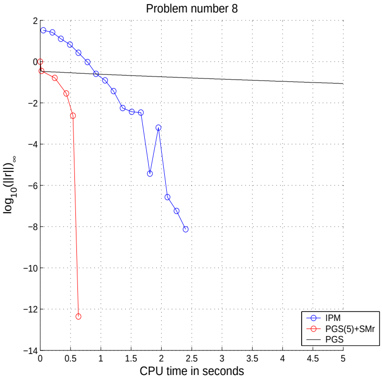
Projected Gauss-Seidel with Subspace Minimization

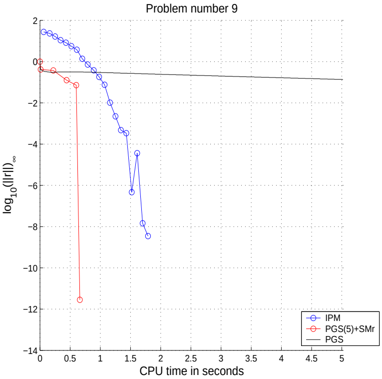
Initialize $u, v \geq 0$. Choose a constant $tol > 0$.
repeat until a convergence test for problem (1) is satisfied
  Perform k_{gs} iterations of the projected Gauss-Seidel iteration to obtain an iterate (u, v);
  Define $\hat{v}$ to be the subvector of v whose components satisfy $v_i > tol$;
  repeat at most k_{sm} times (subspace minimization)
    Form and solve the reduced system (12);
    If the solution $\hat{v}$ satisfies $\hat{v}$ >tol, **break**;
    Project the solution by setting $\hat{v} \leftarrow \max(0, \hat{v})$;
  end repeat
end repeat

Identifying the active set

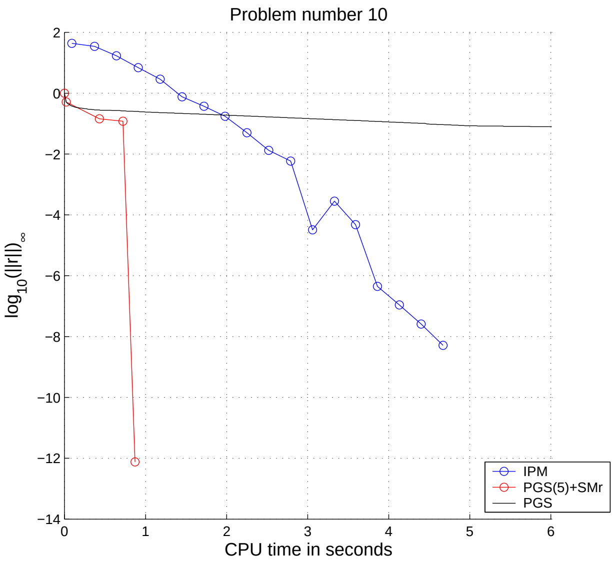
name	cpu time	nz(L)	# Chol. fact.
6	0.50/0.22	216 080/114 864	17/7
7	1.02/0.45	406 152/218 522	18/8
8	2.40/0.63	797 989/398 821	16/7
9	1.79/0.66	646 929/341 911	19/9
10	4.67/0.87	1 222 209/604 892	17/6

Error behaviour as a function of the CPU time





Error behaviour as a function of the CPU time



Observations.

- PGS is able to detect a very high proportion of the active set at the solution during the first iterations.
- PGS can be very slow on difficult problems.
- PGS + subspace minimization iterations $\Rightarrow$ flexible method
- Applying the subspace minimization repeatedly $\Rightarrow$ robust method.
- PGS + subspace minimization iterations $\Rightarrow$ robust and flexible.