Solution of Simultaneous Equations Models in High Performance Systems

José-Juan López-Espín
Universidad Miguel Hernández (Elche, Spain)
Domingo Giménez
Universidad de Murcia (Murcia, Spain)

This work has been funded in part by Spanish MCYT and FEDER under Grant TIC2003-08238-C02-02

Contents

- Introduction
- Simultaneous equations models
- Methods and algorithms
  - OLS
  - ILS
  - 2SLS
- Experimental results
- Conclusions and future works

Introduction

- Applications of S.E.M.: econometric, psychology, medicin, networks' simulation, ...
- The solution of a S.E.M. in high performance parallel systems is studied.
- The methods analyzed here are ILS and 2SLS.
- Parallel algorithms for shared an distributed memory have been developed.
- The methods have been analyzed in different parallel systems.

Simultaneous Equations Models

The scheme of a system with M equations, M endogenous variables and k predetermined variables is (structural form)

$$Y_{1t} = \beta_{12}Y_{2t} + \beta_{13}Y_{3t} + \ldots + \beta_{1M}Y_{Mt} + \gamma_{11}X_{1t} + \ldots + \gamma_{1k}X_{kt} + u_{1t}$$
$$Y_{2t} = \beta_{21}Y_{1t} + \beta_{23}Y_{3t} + \ldots + \beta_{2M}Y_{Mt} + \gamma_{21}X_{1t} + \ldots + \gamma_{2k}X_{kt} + u_{2t}$$
$$\ldots$$
$$Y_{Mt} = \beta_{M1}Y_{1t} + \beta_{M2}Y_{2t} + \beta_{M3}Y_{3t} + \ldots + \beta_{MM-1}Y_{M-1t} + \gamma_{M1}X_{1t} + \ldots + \gamma_{Mk}X_{kt} + u_{Mt}$$

These equations can be represented in matrix form

$$BY_t + \Gamma X_t + u_t = 0$$

Simultaneous Equations Models

The structural form can be expressed in reduced form

$$Y_t = \Pi X_t + v_t$$

with $\Pi = -B^{-1}\Gamma$ and $v_t = -B^{-1}u_t$

$$Y_{1t} = \pi_{11} X_{1t} + \ldots + \pi_{1k} X_{kt} + v_{1t}$$

$$\ldots$$

$$Y_{Mt} = \pi_{M1} X_{1t} + \ldots + \pi_{Mk} X_{kt} + v_{Mt}$$

Simultaneous Equations Models

Identification problem

- Underidentified >> not solve
- Overidentified >> 2SLS
- Just-identified >> ILS (also 2SLS)

- Order conditions (necessary)
- Range conditions (necessary and sufficient)

Contents

- Introduction
- Simultaneous equations models
- Methods and algorithms
  - OLS
  - ILS
  - 2SLS
- Experimental results
- Conclusions and future works

OLS (Method)

OLS (**Ordinary Least Square**) can be used to solve a regression model

$$Y_t = a_1 X_{1t} + \ldots + a_n X_{nt} + u_t$$

In matrix form

$$Y = bX + u$$

The expression of the estimator is

$$\hat{b} = (XX)^{-1} XY$$

The determination coefficient is also calculated

ILS (Method)

The technique ILS needs the equation to be **exactly identified**, which means the values of structural coefficients can be univocally obtained from the reduced form

$$-B_i\Pi = \Gamma_i$$

- The models in reduced form are calculated using OLS
- The structural form is calculated from reduced models

2SLS (Method)

- OLS can not be used in structural form because random variable and endogenous variables are correlated
- Endogenous variables are replaced for approximations (proxys variables)
- The proxy of Y is calculated using OLS with Y and the exogenous in the system.
- When the endogenous have been replaced, we use OLS with the variables of this equation (special OLS).

Parallel Algorithms

- Use of ScaLAPACK
- Two type of parallel systems:
  - Shared memory
  - Distributed memory
- One equation
- A complete SEM

OLS (Shared Memory)

- Create X and Y
- Calculate X'X
- Calculate $(X'X)^{-1}$
- Calculate $(X'X)^{-1}X'$
- Calculate $(X'X)^{-1}X'Y$

- ScaLAPACK is used
- $(X'X)^{-1}X'$ is used in successive OLS

ILS for 1 equation (Shared Memory)

- ILS is
  - A loop to find Pi Matrix → Use OLS
  - SolutionILS (create and solve a system) → Not important time when there is a big sample size
- Make the first iteration (All the threads work together)
- Each thread make several iterations (N/np), and all of them use $(X'X)^{-1}X'$ in OLS (it was calculated in the first iteration)
- Parallel at low level in SolutionILS

ILS for a system (Shared Memory)

- ILS in different equations can share the Pi matrix
- ILS needs more time to solve the first equations (it doesn't have Pi matrix).
- In a system (with a lot of just-identified equations)
  - All the threads work to solve the first equation
  - The rest of the equations are distributed between the threads

2SLS for 1 equation (Shared Memory)

- 2SLS is
  - A loop to find proxys variables → Use OLS
  - Special OLS
- Calculate the first proxy (All threads working together)
- Each thread calculates several proxys, and all of them use $(X'X)^{-1}X'$ in OLS (it was calculate in the first proxy).
- Parallel at low level in special OLS

Distributed Memory

- Version 1:

Process 0 sends the data to the other processes.

Each process solves N/np equations.

Each process sends the solutions to process 0.

- Version 2:

All the processes work in the solution of the first equation.

Each process solves N/np equations.

Each process sends the solutions to process 0.

Contents

- Introduction
- Simultaneous equations models
- The identification problem
- Methods and algorithms
  - OLS
  - ILS
  - 2SLS
- Experimental results
- Conclusions and future works

Computer System

- HPC160: 32GFlops, 4 nodes each node with 4 processors (1Ghz) (OnpenMP algorithms have been tested).
- A cluster of ten biprocessors Intel Xeon 2 with SCI connection (MPI algorithms have been tested).
- The MPI algorithms have also been tested in Marenostrum. Marenostrum compromises 2282 JS20 compute nodes and 42 p615 servers. Each node has two processors at 2.2 Ghz running Linux operating system with 4 GB of memory RAM and 40 GB local disk storage.

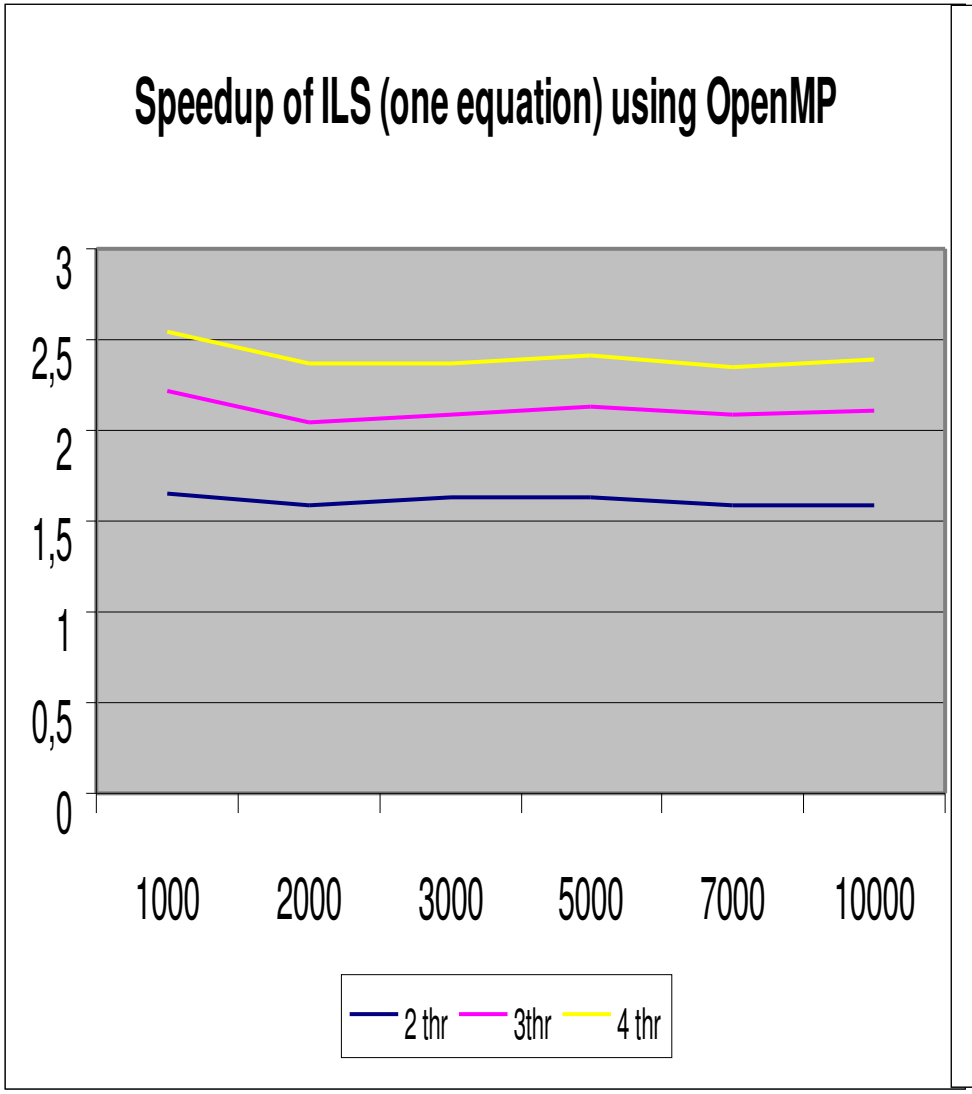
ILS time (sequential)

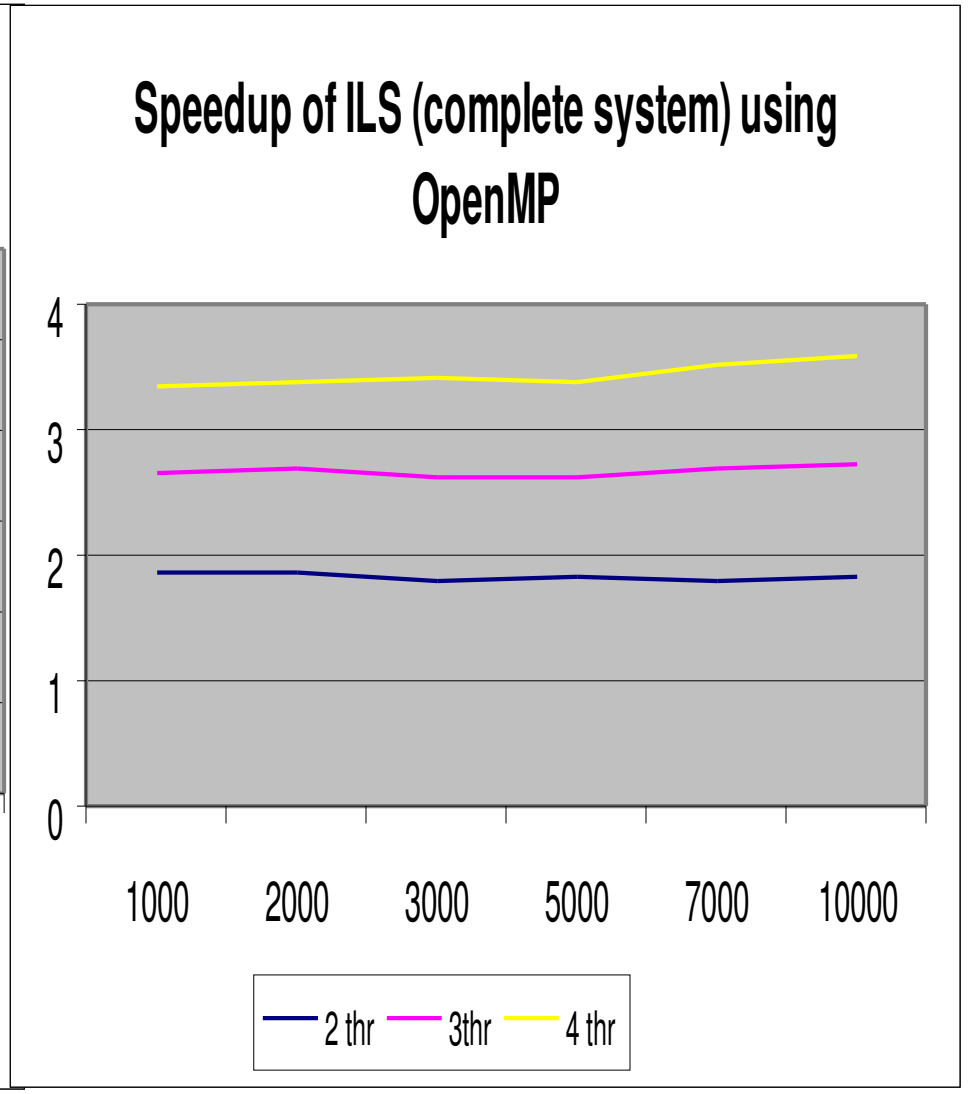
N	K	n	k	d	ILS	OLS	%	Loop	%	solution ils	%
500	200	201	0	100	0,120115	0,044921	37,40%	0,055663	46,34	0,01953	16,26%
500	200	201	0	500	0,415031	0,141599	34,12%	0,252925	60,94	0,01953	4,71%
500	200	201	0	1000	1,42966	0,896467	62,70%	0,512685	35,86	0,01953	1,37%
1000	400	401	0	100	0,586903	0,226558	38,60%	0,218746	37,27	0,14062	23,96%
1000	400	401	0	500	3,225525	1,981408	61,43%	1,099588	34,09	0,14258	4,42%
1000	400	401	0	1000	6,720575	4,204021	62,55%	2,372025	35,29	0,14258	2,12%
1500	600	601	0	100	1,627899	0,629871	38,69%	0,537099	32,99	0,458	28,13%
1500	600	601	0	500	8,538899	5,655166	66,23%	2,420852	28,35	0,45605	5,34%
1500	600	601	0	1000	20,277933	12,77222	62,99%	7,042835	34,73	0,45898	2,26%
2000	800	801	0	100	3,345639	1,345677	40,22%	0,918928	27,47	1,07225	32,05%
2000	800	801	0	500	21,808178	15,07881	69,14%	5,652236	25,92	1,06932	4,90%
2000	800	801	0	1000	41,598812	29,4799	70,87%	11,04081	26,54	1,06834	2,57%

ILS parallel

N	K	n	1h	2 th	%	speedup	3 th	%	speedup	4 th	%	speed
1000	400	401	**6,78991**	4,095625	**60,32%**	**1,657845**	3,0761	**45,30%**	**2,2073**	2,6689	**39,31%**	**2,5441**
2000	800	801	**41,445496**	26,23973	**63,31%**	**1,579494**	20,206	**48,75%**	**2,05118**	17,455	**42,11%**	**2,3745**
3000	1200	1201	**125,63448**	77,20849	**61,45%**	**1,627211**	60,203	**47,92%**	**2,08685**	52,839	**42,06%**	**2,3777**
5000	2000	2001	**430,63632**	265,1268	**61,57%**	**1,624266**	202,049	**46,92%**	**2,13135**	178,243	**41,39%**	**2,416**
7000	2800	2801	**975,65448**	612,4561	**62,77%**	**1,59302**	467,08	**47,87%**	**2,08884**	414,346	**42,47%**	**2,3547**
10000	4000	4001	**2024,6293**	1268,925	**62,67%**	**1,595547**	961,9	**47,51%**	**2,10482**	845,408	**41,76%**	**2,3949**

Sample size = 1000


Speedup of ILS (one equation) using OpenMP
3
2,5
2
1,5
1
0,5
0
1000
2000
3000
5000
7000
10000
2 thr
3thr
4 thr


Speedup of ILS (complete system) using OpenMP
4
3
2
1
0
1000
2000
3000
5000
7000
10000
2 thr
3thr
4 thr

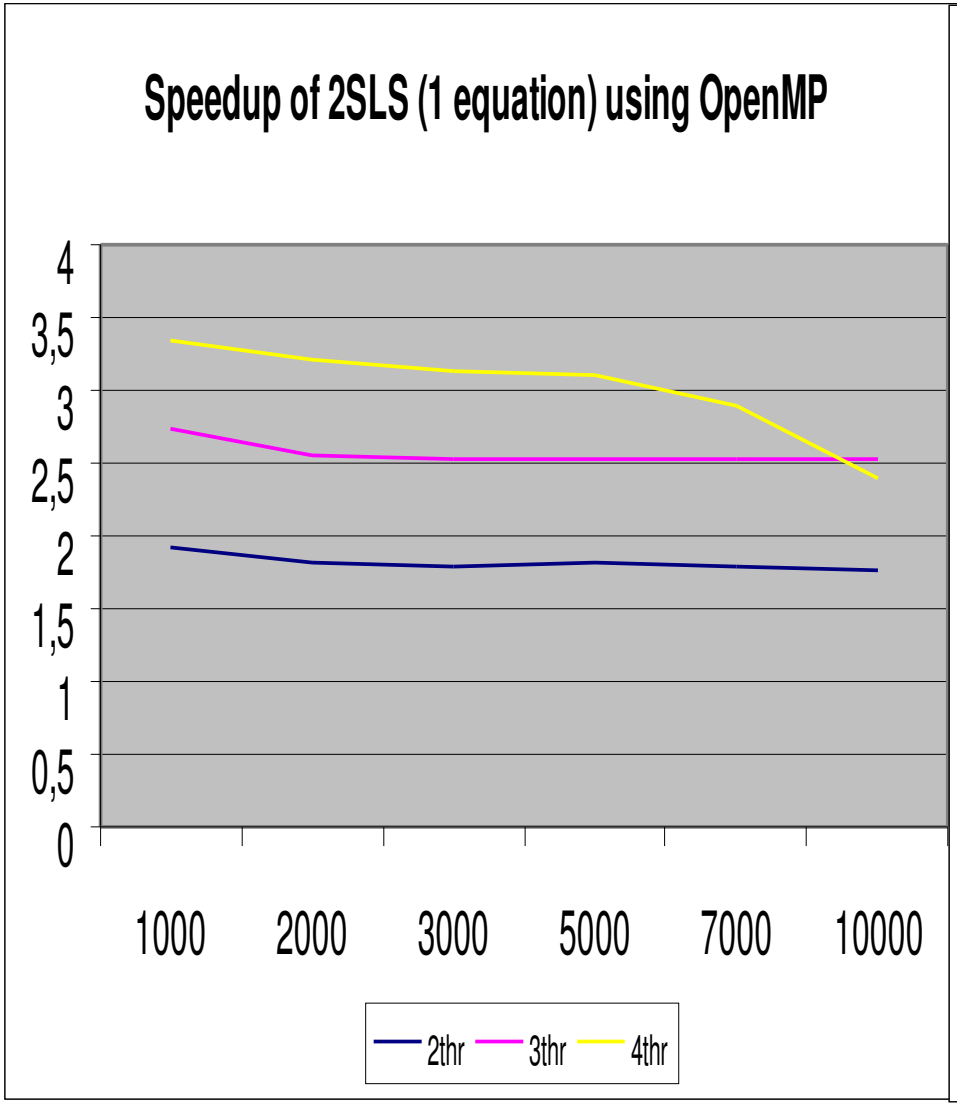
2SLS time (sequential)

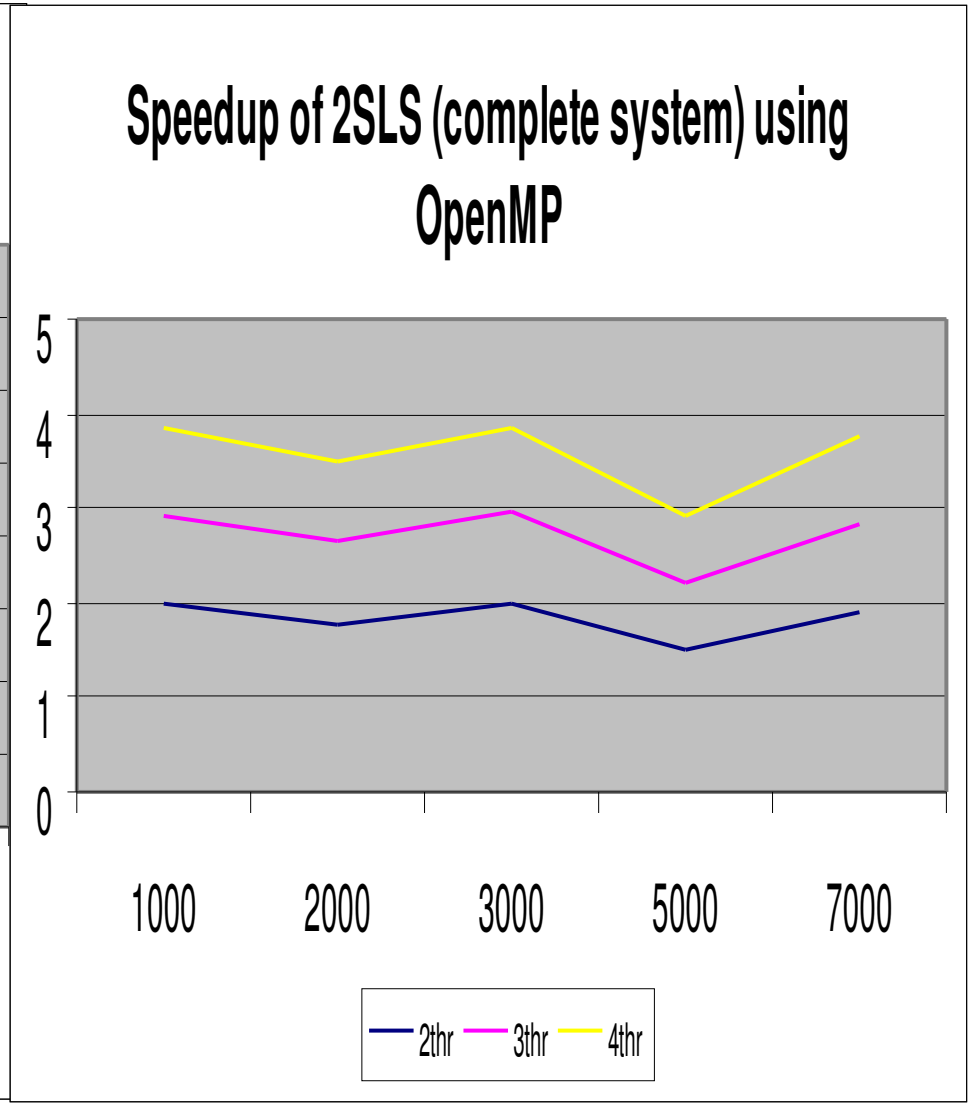
N	K	n	k	d	2LSL	First proxy	%	Rest of proxys	%	Special OLS	%
500	200	201	0	100	0,10449	0,044921	42,99%	0,02637	25,23%	0,0332	31,78%
500	200	201	0	500	0,3496	0,148433	42,46%	0,11035	31,56%	0,09277	26,54%
500	200	201	0	1000	2,049752	0,969701	47,31%	0,21191	10,34%	0,86131	42,02%
1000	400	401	0	100	0,519518	0,230463	44,36%	0,10449	20,11%	0,18457	35,53%
1000	400	401	0	500	4,412018	2,074176	47,01%	0,43357	9,83%	1,85543	42,05%
1000	400	401	0	1000	9,448	4,417855	46,76%	1,0488	11,10%	3,98037	42,13%
1500	600	601	0	100	1,449181	0,64842	44,74%	0,25195	17,39%	0,54296	37,47%
1500	600	601	0	500	12,012446	5,662968	47,14%	1,09666	9,13%	5,27626	43,92%
1500	600	601	0	1000	27,688742	13,00943	46,98%	2,54486	9,19%	12,1042	43,72%
2000	800	801	0	100	3,05168	1,374965	45,06%	0,47949	15,71%	1,20798	39,58%
2000	800	801	0	500	30,612495	14,63634	47,81%	2,44135	7,97%	13,7868	45,04%
2000	800	801	0	1000	62,037785	29,41932	47,42%	4,88938	7,88%	27,6703	44,60%

2SLS parallel

N	K	n	1 th	2 th	%	Speed up	3 th	%	Speed up	4 th	%	Speed Up
1000	400	401	**9,448**	4,890532	**51,76%**	**1,9319**	3,45987	**36,62%**	**2,73074**	2,821216	**29,86%**	**3,348911**
2000	800	801	**62,037785**	34,04818	**54,88%**	**1,82206**	24,2591	**39,10%**	**2,5573**	19,35887	**31,20%**	**3,204618**
3000	1200	1201	**193,61516**	107,871	**55,71%**	**1,79488**	76,7583	**39,64%**	**2,5224**	61,58336	**31,81%**	**3,143953**
5000	2000	2001	**680,42969**	376,9001	**55,39%**	**1,80533**	268,853	**39,51%**	**2,53086**	219,6535	**32,28%**	**3,097742**
7000	2800	2801	**1503,1566**	836,4596	**55,65%**	**1,79705**	594,646	**39,56%**	**2,52782**	520,465	**34,62%**	**2,888103**
10000	4000	4001	**3238,9675**	1833,813	**56,62%**	**1,76625**	1285,38	**39,68%**	**2,51985**	1349,368	**41,66%**	**2,400358**

Sample size = 1000


Speedup of 2SLS (1 equation) using OpenMP
4
3,5
3
2,5
2
1,5
1
0,5
0
1000
2000
3000
5000
7000
10000
2thr
3thr
4thr


Speedup of 2SLS (complete system) using OpenMP
5
4
3
2
1
0
1000
2000
3000
5000
7000
2thr
3thr
4thr

Distributed Memory (ILS complete system using MPI)

N	1 th	2 th	sp	4 th	sp	8 th	sp	16 th	sp	24 th	sp	32 th	sp	64 th	sp
1000	**54,4**	27,7	**2,0**	16,2	**3,3**	8,2	**6,6**	4,5	**12,1**	3,2	**16,9**	2,6	**20,7**	1,2	**45,7**
2000	**1473,3**	763,7	**1,9**	432,1	**3,4**	217,5	**6,8**	110,6	**13,3**	74,6	**19,8**	57,6	**25,6**	30,7	**48,1**
3000	**6680,2**	3387,1	**2,0**	1970,5	**3,4**	996,7	**6,7**	492,7	**13,6**	332,2	**20,1**	248,7	**26,9**	130,7	**51,1**
4000	**18117,8**	9358,9	**1,9**	5373,4	**3,4**	2622,4	**6,9**	1339,0	**13,5**	900,3	**20,1**	675,6	**26,8**	349,1	**51,9**
5000	**43289,0**	21363,3	**2,0**	12332,4	**3,5**	6221,6	**7,0**	5845,2	**7,4**	2079,9	**20,8**	1569,7	**27,6**	827,7	**52,3**
6000	**83270,4**	41971,2	**2,0**	23551,9	**3,5**	11993,3	**6,9**	5984,5	**13,9**	4021,2	**20,7**	3064,2	**27,2**	1553,6	**53,6**

Distributed Memory (2SLS complete system using MPI)

N	1 th	2 th	sp	4 th	sp	8 th	sp	16 th	sp	24 th	sp	32 th	sp	64 th	sp
1000	**377,7**	188,7	**2,0**	106,1	**3,6**	53,2	**7,1**	26,5	**14,3**	17,8	**21,2**	13,8	**27,5**	7,0	**53,6**
2000	**4995,4**	2505,7	**2,0**	1415,6	**3,5**	726,4	**6,9**	379,1	**13,2**	245,2	**20,4**	180,3	**27,7**	94,9	**52,6**
3000	**20037,0**	9987,2	**2,0**	5778,2	**3,5**	2983,5	**6,7**	1494,4	**13,4**	1013,1	**19,8**	720,2	**27,8**	377,4	**53,1**
4000	**58256,5**	29324,6	**2,0**	16844,4	**3,5**	8289,9	**7,0**	4109,5	**14,2**	2811,3	**20,7**	2091,0	**27,9**	1098,2	**53,0**

Conclusions and Future works

- Parallelization of algorithms for Simultaneous Equations Models in High Performance Systems
- Tools will be made freely available to the scientific community
- Optimize the developed methods
- Develop other methods (full information)
- Application to real problems