A PARALLEL CODE FOR SOLVING LINEAR SYSTEM EQUATIONS WITH MULTIMODULAR ALGEBRA

Héctor Eduardo González[1], División de Estudios de Postgrado de la Facultad de Ingeniería, Universidad Nacional Autónoma de México/ITTLA
Enrique Cruz Martinez[2], Dirección General de Servicios de Cómputo Académico, Universidad Nacional Autónoma de México

ABSTRACT
Parallel automatic OpenMp codes for solving simultaneous linear equations with integral coefficients is presented. The solution is obtained by applying the "Chinese Remainder Theorem" avoiding floating point operations. The algorithm used can be extended to sets of equations with the same algebraic structure with real coefficients.

Key words: exact solution of simultaneous linear equations, multimodular arithmetic, Chinese Remainder Theorem, linear system equations over finite fields.

RESUMEN
Aquí se presenta un código paralelo para resolver sistemas de ecuaciones lineales simultáneas con coeficientes enteros. La solución se obtiene aplicando el "Teorema Chino del Residuo" evitando así operaciones de punto flotante. Este algoritmo puede extenderse a conjuntos de ecuaciones con la misma estructura algebraica y coeficientes reales.

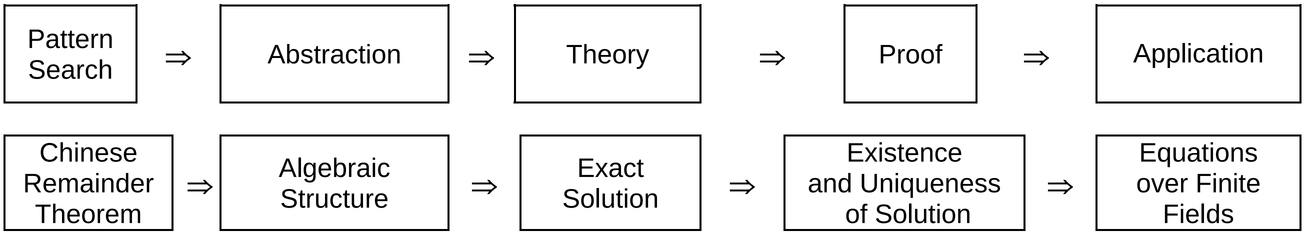
Palabras clave: solución exacta de ecuaciones lineales simultáneas, aritmética multimodular, Teorema Chino del Residuo, ecuaciones del sistema lineal sobre los campos finitos.

MSC:65F05

1. INTRODUCTION

We have solved sets of linear equations with dense matrices of integral coefficients using the direct Gauss – Jordan method and modular arithmetic of ancient origin to solve in a supercomputer the equations exactly (with no roundoff error) avoiding floating point operations.

The way in which the application of a modern computing tool was conceived for solving an old problem is sketched in the following graph


Pattern Search
⇒
Abstraction
⇒
Theory
⇒
Proof
⇒
Application
Chinese Remainder Theorem
⇒
Algebraic Structure
⇒
Exact Solution
⇒
Existence and Uniqueness of Solution
⇒
Equations over Finite Fields

2. ALGEBRAIC STRUCTURE

Modular Arithmetic [Conway-Guy (1996)] is a very versatile tool discovered by K.F.Gauss (1777-1855) in 1801. Two numbers **a** and **b** are said to be equal or congruent modulo **N** written **N|**(**a-b**), if their difference is exactly divisible by **N**. Usually (and in this paper) **a**,**b**, are nonnegative integers and **N** is a positive integer. We write **a** ≡ **b** (**mod N**).

The set of numbers congruent modulo **N** is denoted $\mathbf{[a]_N}$. If $\mathbf{b} \in \mathbf{[a]_N}$ then, by definition, **N|(a-b)** or, in other words, **a** and **b** have the same remainder on division by **N**. Since there are exactly **N** possible remainders upon division by **N**, there are exactly **N** different sets $\mathbf{[a]_N}$.

Email:[1]heg53@prodigy.net.mx
[2]ecm@super.unam.mx

Quite often these **N** sets are simply identified with the corresponding remainders: **$[0]_N = 0, [1]_N = 1, \ldots, [N-1]_N = N-1$**. Remainders are often called *residues*; accordingly, the **[a]'s** are also known as the *residue classes*.

It is easy to see that if **$a \equiv b \pmod N$** and **$c \equiv d \pmod N$** then **$(a+c) \equiv (b+d) \pmod N$.** The same is true for multiplication. This allows us to introduce an **algebraic structure** [2] into the set **$\{[a]_N: a=0,1,\ldots,N-1\}$.**

By definition:

1. $[a]_N + [b]_N = [a + b]_N$

2. $[a]_N \text{ x } [b]_N = [a \text{ x } b]_N$

Subtraction is defined similarly:

$$[a]_N - [b]_N = [a - b]_N$$

and it can be verified that the set $\{[a]_N: a=0,1,\ldots,N-1\}$ becomes a **ring** with **commutative** addition and multiplication.

Division can not be always defined. To give an obvious example:

$$[5]_{10} * [1]_{10} = [5]_{10} * [3]_{10} = [5]_{10} * [5]_{10} = [5]_{10} * [7]_{10} = [5]_{10} * [9]_{10} = [5]_{10}.$$

So $[5]_{10}/[5]_{10}$ can not be defined uniquely.

We also see that:

$$[5]_{10} * [2]_{10} = [5]_{10} * [4]_{10} = [5]_{10} * [6]_{10} = [5]_{10} * [8]_{10} = [5]_{10} * [0]_{10} = [0]_{10}.$$

something we never had either for integer or real numbers.

The situation improves for prime **N's** in which case division can be defined uniquely.

Observe the multiplication tables below for prime **N**. For the multiplication and division table we have removed the **0** column and row.

Every row (and column) contains all non-zero remainders mostly messed up. So every row is a permutation of the first row in the table. This provides an easy way to construct division tables too. For prime **N**, the set **$\{[a]_N: a=0,1,\ldots,N-1\}$** can be upgraded to a **field.**

Table N = 5.

For **non prime N**, most **rows contain zeros** and **repeated entries**.

Addition

	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Multiplication

	1	**2**	**3**	**4**
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Subtraction

	0	1	2	3	4
0	0	1	2	3	4
1	4	0	1	2	3
2	3	4	0	1	2
3	2	3	4	0	1
4	1	2	3	4	0

Division

	1	2	3	4
1	1	2	3	4
2	3	1	4	2
3	2	4	1	3
4	4	3	2	1

The tables exhibit a variety of patterns. To mention a few [Kevin (1997), Davis-Hersh 81981)]:

a) For addition, consecutive rows result from the first one by circular rotation of entries.

b) Addition and multiplication tables are symmetric with respect to the main diagonal (the line that goes from the top left to the bottom right corner.)

c) Subtraction tables are not symmetric but the rows are still obtained from the first one by rotation of entries. (We subtract numbers in the leftmost column from the numbers in the topmost row.)

d) In multiplication tables, the last row is always a reverse of the first row.

e) In multiplication tables modulo **N**, rows corresponding to numbers *coprime* with **N** contain permutations of the first row.

f) For prime **(N+1)**, multiplication tables offer multiple and simultaneous solutions to the rook problem: On an **NxN** board position **N** rooks so that they command the whole board and none may capture another. To solve, select a digit, replace all its occurrences with a rook, remove all other digits.

g) Under the same conditions, *1* always appears in the upper left and lower right corners and nowhere else on the main diagonal.

h) 6/5 = 4 (mod 7) or, which is the same, $[6]_7/[5]_7 = [4]_7$

i) One can use addition and subtraction tables to play the same game as with the Calendar Tables.

j) For multiplication tables, this is also true provided selected entries are multiplied instead of being added up.

k) For multiplication tables, both diagonals are palindromic, i.e. each is the same in both directions.

l) If an addition table has an odd number of rows, then every remainder occurs on the main diagonal.

m) In subtraction tables with an odd number of rows, the second diagonal is a permutation of the first row.

n) In addition tables with an even number of rows, the main diagonal contains only a half of all the remainders. The remainders on the diagonal appear twice each.

o) In multiplication tables with a number of rows **N** where **(N+1)** is prime, the same is also true: the main diagonal contains only a half of all the remainders. The remainders on the diagonal appear twice each.

p) In the table of multiplication by **N**, rows corresponding to the numbers coprime with **N** consist of permutations of the first row. The reverse does not hold.

3. CHINESE REMAINDER THEOREM

According to D.Wells, the following problem was posed by Sun Tsu Suan-Ching (4th century AD):

There are certain things whose number is unknown. Repeatedly divided by 3, the remainder is 2; by 5 the remainder is 3; and by 7 the remainder is 2. What is the number?

Oystein Ore mentions another puzzle with a dramatic element from *Brahma-Sphuta-Siddhanta* (Brahma's Correct System) by Brahmagupta (born 598 AD):

An old woman goes to market and a horse steps on her basket and crashes the eggs. The rider offers to pay for the damages and asks her how many eggs she had brought. She does not remember the exact number, but when she had taken them out two at a time, there was one egg left. The same happened when she picked them out three, four, five, and six at a time, but when she took them seven at a time they came out even. What is the smallest number of eggs she could have had?

Problems of this kind are all examples of what universally became known as the *Chinese Remainder Theorem*. In mathematical parlance the problems can be stated as finding **n**, given its remainders of division by several numbers $\mathbf{m_1}$, $\mathbf{m_2,...,m_k}$:

$$n = n_1 \text{ (mod } m_1)$$

$$n = n_2 \text{ (mod } m_2)$$

$$...$$

$$n = n_k \text{ (mod } m_k)$$

The modern day theorem is best stated with a couple of useful notations. For non-negative integers $\mathbf{m_1,m_2,...,m_k}$, their *greatest common divisor* is defined as:

$$\gcd(m_1,m_2,...,m_k) = \max\{s: s|m_i, \text{ for } i=1,...,k\},$$

where, as usual, **"s|m"** means that **s** divides **m** exactly. The *least common multiple* of **k** numbers is defined as

$$\text{lcm}(m_1,m_2,...,m_k) = \min\{s: s>0 \text{ and } m_i|s, \text{ for } i=1,...,k\},$$

Both **gcd()** and **lcm()** are **symmetric functions** of their arguments. They are complementary in the sense that, for **k = 2**, $\mathbf{\gcd(m_1,m_2)\text{lcm}(m_1,m_2) = m_1m_2}$.

However, for **k>2** a similar identity does not in general hold. For an example, consider two triplets: {2,4,16} and {2,8,16}. Both have exactly the same gcd and lcm but obviously different products. On the other hand, both **gcd** and **lcm** are *associative*:

$$\gcd(m_1, (\gcd(m_2, m_3)) = \gcd(\gcd(m_1, m_2), m_3)$$

and, both equal $\mathbf{\gcd(m_1, m_2, m_3).}$

Similarly,

$$\text{lcm}(m_1, (\text{lcm}(m_2, m_3)) = \text{lcm}(\text{lcm}(m_1, m_2), m_3)$$

Associativity allows one to proceed a step at a time with an inductive argument without putting all eggs into a basket at once. Jumping at the opportunity we will prove the most basic case of **k = 2**.

Theorem

Two simultaneous congruences $\mathbf{n \equiv n_1 \text{ (mod } m_1)}$ and $\mathbf{n \equiv n_2 \text{ (mod } m_2)}$ are only solvable when $\mathbf{n_1 \equiv n_2 \text{ (mod } \gcd(m_1,m_2))}$. The solution is unique modulo $\mathbf{\text{lcm}(m_1,m_2)}$.

When $\mathbf{m_1}$ and $\mathbf{m_2}$ are **coprime** their **gcd is 1**. By convention, $\mathbf{a \equiv b \text{ (mod } 1)}$ is simply understood as the usual **equality a = b**.

Proof

The first congruence is equivalent to $\mathbf{n = tm_1 + n_1}$, the second to $\mathbf{n = sm_2 + n_2}$, for some integers **t** and **s**.

Equating we obtain

$$tm_1 - sm_2 = n_2 - n_1. \qquad (1)$$

The left-hand side is divisible by $\mathbf{gcd(m_1,m_2)}$. So, unless the right-hand side is also divisible by $\mathbf{gcd(m_1,m_2)}$, there could not possibly exist **t** and **s** that satisfy the identity.

Now let's assume that $\mathbf{gcd(m_1,m_2)|(n_2 - n_1)}$ and denote $\mathbf{n_0 = (n_2 - n_1)/gcd(m_1,m_2)}$. Then, (1) can be written as

$$t(m_1/gcd(m_1,m_2)) = n_0 \text{ (mod } (m_2/gcd(m_1,m_2))) \qquad (2)$$

By definition, $\mathbf{m_1/gcd(m_1,m_2)}$ and $\mathbf{m_2/gcd(m_1,m_2)}$ are coprime; since we are dividing $\mathbf{m_1}$ and $\mathbf{m_2}$ by their largest common factor. Therefore, by a generalization of Euclid's Proposition, (2) has a solution. Given **t**, we can determine $\mathbf{n = tm_1 + n_1}$. This proves the existence part.

To prove the uniqueness part, assume **n** and **N** satisfy the two congruences. Taking the differences we see that

$$N\text{-}n = 0 \text{ (mod } m_1) \text{ and } N\text{-}n = 0 \text{ (mod } m_2) \qquad (3)$$

which implies $\mathbf{N\text{-}n = 0 \ (mod\ lcm(m_1,m_1))}$.

4. A RESIDUE SYSTEM OF EQUATIONS

Consider the linear algebraic system of equations $Ax = b$. Even if A and b are required to be integral there is no guarantee that x, the solution vector, will be integral. On the other hand, when we write the residue system of equations $\left|A\bar{x}\right|_M = \left|b\right|_M$ with A and b integral, we seek an integral vector $\left|\bar{x}\right|_M$ which satisfies it. In general, then, x and $\left|\bar{x}\right|_M$ are different and it would appear that $\left|\bar{x}\right|_M$ would not aid us in finding x ; however, it turns out that this is not the case. We can use residue arithmetic in solving it and this will lead us to a solution of $Ax = b$, where A and b are integral.

Gauss-Jordan Elimination

We seek a nonsingular nxn matrix J for which $JAx = Jb$ with $JA = \mathbf{I}$ thus, $J = A^{-1}$ and $x = Jb$ is the solution. The reduction of A to the identity matrix consists of n major steps which correspond to n matrix row operations. These are similar to the n steps in ordinary Gauss elimination which reduce A to an upper triangular matrix. However, Jordan's variation eliminates the elements above the diagonal as it eliminates the elements below the diagonal and also scales the diagonal elements to unity. Thus, there is no need for back substitution. If the pivotal candidate does vanish, we must do row interchanges before we can carry out that step of the elimination [Golub-Van Loan (1983)].

The analog of Gauss-Jordan elimination for residue arithmetic using more than one modulus

It is more practical in a general situation [Young-Todd (1972)] to select a set of moduli $m_1, m_2, ..., m_s$, with $M = m_1 m_2 ... m_s$ because, this enables us to obtain results modulo M by doing most of the arithmetic modulo m_i, for $i = 1,2,...,s$. To be more specific, we select a set of moduli $m_1, m_2, ..., m_s$, with $(m_i, m_j) = 1$ for $i \neq j$. Let $d = \det A$ and $y = A^{adj} b$. We shall assume there are sufficient moduli (large enough) so that M satisfies $(d, M) = 1$ and $M > 2\max\left(|d|, \max_i |y_i|\right)$.

We solve the residue systems $\left|A\overline{x}\right|_{m_i} = \left|b\right|_{m_i}$ for $i = 1,2,...,s$, (for each of the moduli) and obtain the residue representations $d \sim \left\{\left|d\right|_{m_1}, \left|d\right|_{m_2}, ..., \left|d\right|_{m_s}\right\}$ and $y \sim \left\{\left|y\right|_{m_1}, \left|y\right|_{m_2}, ..., \left|y\right|_{m_s}\right\}$. From these two s-tuples we can determine $\left|d\right|_M = \left|y\right|_M$ and if M is large enough, we can determine d and y and, ultimately, $x = A^{-1}b$.

The Chinese Remainder Theorem

There are various algorithms for obtaining $\left|d\right|_M$ and $\left|y\right|_M$. Perhaps the best known procedure makes use of a classic theorem from the theory of numbers called the Chinese Remainder Theorem.

Theorem. Let $m_1, m_2, ..., m_s$ be the base for a residue number system with $(m_i, m_j) = 1$ for $i \neq j$, and let $M = m_1 m_2 ... m_s$. Also, let $\hat{m}_j = \frac{M}{m_j}$. Now, if q has the residue representation $q \sim \{r_1, r_2, ..., r_s\}$ where

$$r_i = \left|q\right|_{m_i} \quad i = 1,2,...,s \text{ then } \left|q\right|_M = \left|\sum_{j=1}^{s} \hat{m}_j \left| r_j \, \hat{m}_j^{-1}(m_j)\right|_{m_j}\right|_M .$$

Obtaining x from $\left|\overline{x}\right|_M$.

Fortunately, we can obtain x if we are willing to do the additional work. Obviously, we need to compute $\left|d\right|_M$ and $\left|y\right|_M$ since x is obtained from y by dividing the components of y by d. It should be pointed out that only at this point do we leave residue arithmetic, and so only at this point do we introduce rounding errors. Actually, if the division is merely indicated, but never carried out, then there will be no errors introduced.

Theorem. If the modulus M is chosen so that

i) $M > 2\left|d\right|_M$

and if d' is formed from $\left|d\right|_M$ so as to satisfy

ii) $\left|d'\right|_M = \left|d\right|_M$

iii) $\left|d'\right|_M < M/2$

Then d' = d.

In addition, the modulus M is chosen so that

iv) $M > 2\max_i \left|y_i\right|_M$

and if y' is formed from $\left|y\right|_M$ so as to satisfy

v) $|y'|_M = |y|_M$

vi) $\max_i |y'|_M < M/2$

Then y’ = y.

Briefly, these conditions are satisfied for $M > 2\max\left(|d|, \max_i |y_i|\right)$

Selection of the m_i as primes.

In practice, the moduli are chosen as large prime numbers. This choice increases the probability that $(d, m_i) = 1$ and that $|d|_{m_i} \neq 0$. We recall that if these two conditions are satisfied, then A is nonsingular modulo m_i, and the residue system $|A\bar{x}|_{m_i} = |b|_{m_i}$ can be solved for $|d|_{m_i}$ and $|y|_{m_i}$. If the two conditions are not satisfied, then we simply select another prime for a modulus. Futhermore, by choosing prime numbers for the moduli, we guarantee that $(m_i, m_j) = 1$ for $i \neq j$, and hence, there is a unique integer in the interval (0, m-1) with the given residue representation.

Alternatively, we propose the following scheme to solve a linear system of equations using residue arithmetic $A|A^{adj}b|_{\frac{M}{2}} = |d|_{\frac{M}{2}}b$ where $M = m_1 m_2 ... m_s$ is the composite module, A^{adj} is the adjoint matrix, d is the determinant of A and M/2 is the symmetrical module. This scheme enables us to find the minimal lower bound of the prime module which solves the equations.

5. RESULTS

As an illustration we use the matrix

$$A = (y + \delta_{ij} x); \quad \delta_{ij} = 1 \quad \forall \quad i = j;$$

$$\delta_{ij} = 0 \quad \forall \quad i \neq j \;\; ; \;\; b = \begin{pmatrix} ny + x \\ . \\ . \\ . \\ ny + x \end{pmatrix} \quad ;$$

The value of the determinant and the adjoint matrix are known

$$|A| = x^{n-1}(x + ny)$$

$$A^{adj} = \begin{pmatrix} x^{n-2}[x + (n-1)] & -x^{n-2}y & . \;\; . \;\; . & -x^{n-2}y \\ -x^{n-2}y & x^{n-2}[x + (n-1)] & . \;\; . \;\; . & -x^{n-2}y \\ . & . & . \;\; . \;\; . & . \\ . & . & . \;\; . \;\; . & . \\ . & . & . \;\; . \;\; . & . \\ -x^{n-2}y & -x^{n-2}y & . \;\; . \;\; . & x^{n-2}[x + (n-1)] \end{pmatrix} .$$

We choose $x = 100, y = 1$

The program we used was written in the style of **FORTRAN 77** using some intrinsic subroutines of the new standard **FORTRAN 90**. The code was tested on an **Origin 2000** at UNAM's Supercomputing Center.

We conducted experiments for three numbers **N** of equations, namely **N = 1000,2000** and **5000** and we used from one to six processors (**PE's**).

The following tables shown the **wall clock times** and **speed-ups** [Silicon Graphics Computer System (1998)] for the solutions using the **Chinese Remainder Theorem**.

Table 1.
Times required (in seconds) for the solution of the system Ax = b

Size of the System **N = 1000**	
No. PE's	Wall Clock(s)
1	149.14
2	74.55
4	39.94
6	29.49

Table 2
Speed-up(s) of the solution of the system Ax = b

Size of the System **N = 1000**	
No. PE's	Speed-up(s)
1	1
2	2.000
4	3.691
6	4.998

Table 3.
Times (in seconds) for the solution of the system Ax = b

Size of the System **N = 2000**	
No. PE's	Wall Clock(s)
1	667.17
2	305.31
4	156.65
6	109.37

Table 4.
Speed-up(s) of the solution of the system Ax = b

Size of the System **N = 2000**	
No. PE's	Speed-up(s)
1	1
2	2.186
4	4.258
6	6.100

Table 5.
Times (in seconds) for the solution of the system Ax = b

Size of the System **N = 5000**	
No. PE's	Wall Clock(s)
1	10167.72
2	5174.41
4	2580.64
6	1735.40

Table 6.
Speed-up(s) of the solution of the system Ax = b

Size of the System **N = 5000**	
No. PE's	Speed-up(s)
1	1
2	1.965
4	3.937
6	5.859

The **Scalability** of the algorithm is measured when we calculate the theoretic **Speed-up** using some mixture of **Amdhal`s Law** and two experimental data points to obtain the curve for the particular size of the problem. The following three graphs show some of the results of using **the Chinese Remainder Theorem.**

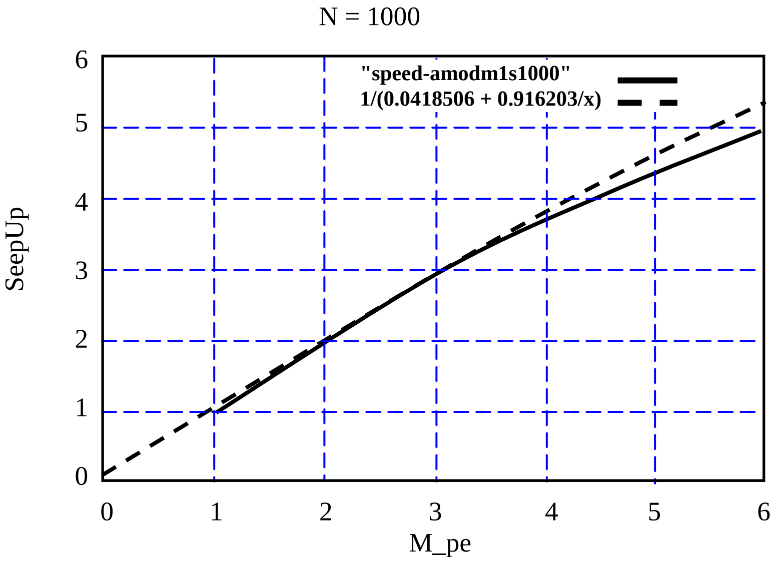


Figure 1. The speed-up curve for the size N = 1000 and the Amdhal Law for this problem.

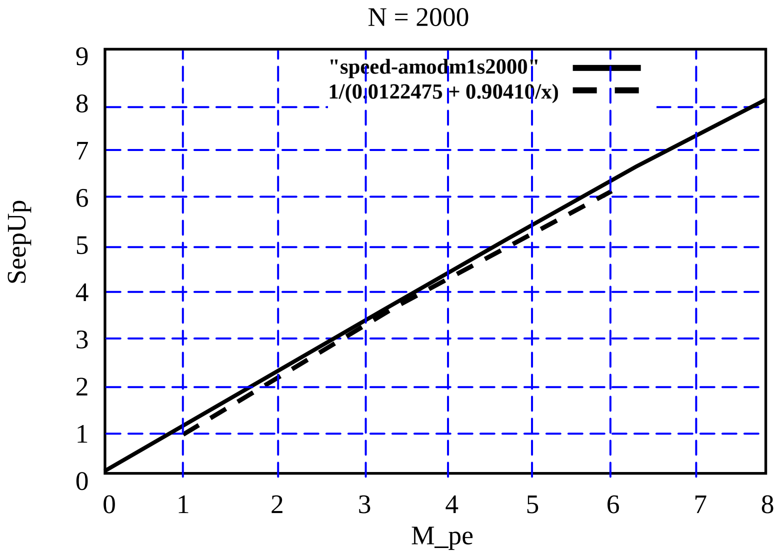


Figure 2. The speed-up curve for the size N = 2000 and the Amdhal Law for this problem.

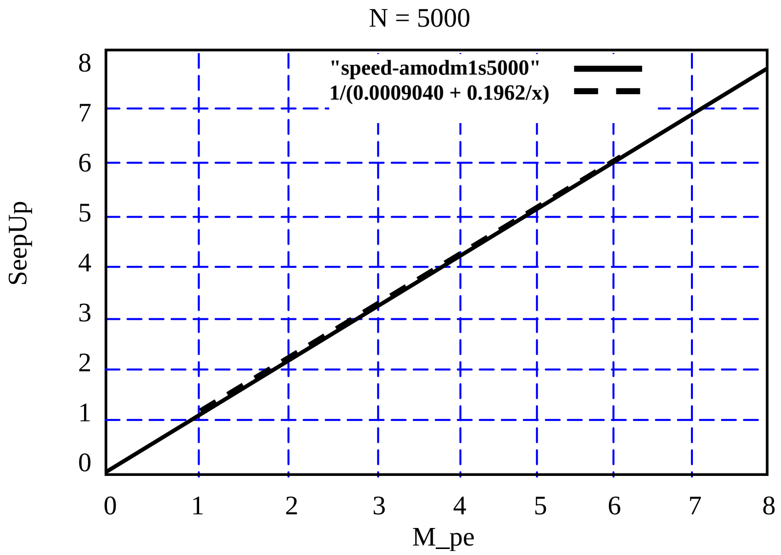


Figure 3. The speed-up curve for the size N = 5000 and the Amdhal Law for this problem.

5. CONCLUSIONS

The traditional user of scientific computing prefers to use well proven routines that are recognized by the international scientific community. Among the well known collections are **IMSL**, **NAG**, and **LINPACK**. However, a direct solution method should be used in the case of dense non structured matrices. If, in addition the coefficients and the right hand side are integers and an exact solution is desired, the only alternative is to use residual arithmetic.

For this type of problems we observe a behavior very close to **ideal speedup**, and in the case of **N = 2000** we observe a **superlinear speedup**.

In any problem with a dense matrix there is an easy decomposition in terms of parallel tasks with the corresponding savings for the appropriate computer architecture.

REFERENCES

CONWAY, J.H.and R.K.GUY 1996): **The Book of Numbers**, Springer-Verlag, N. York.

DAVENPORT, H. (1990): **The Higher Arithmetic**, Harper&Brothers, N. York.

DEVLIN, K. (1997): **Mathematics: The Science of Patterns**, Scientific American Library.

GRAHAM, R.; D. KNUTH and O.POTASHNIK (1994): **Concrete Mathematics**, 2nd edition, Addison-Wesley.

HILTON, P.; D.HOLTON and J.PEDERSON (1997): **Mathematical Reflections**, Springer Verlag, N. York.

OYSTEIN, O. (1976): **Number Theory and Its History**, Dover Publications, London.

STEIN, S.K. (2000): **Mathematics: The Man-Made Universe**, 3rd edition, Dover Publication, London.

DAVIS, P.J. and R. HERSH (1981): **The Mathematical Experience**, Houghton Mifflin Company, Boston.

GROSSWALD, E. (1966): **Topics from the Theory of Numbers**, Mc Millan Company, N. York.

WELLS, D. (1992): **The Penguin Book of Curious and Interesting Puzzles**, Penguin Books, London.

GOLUB, G.H. and Ch.F. VAN LOAN (1983): **Matrix Computations**.Johns Hopkins University Press.

YOUNG, DAVID M. and ROBERT TODD GREGORY (1972): **A Survey of Numerical Mathematics**, II. Dover Publications, Inc., London.

Silicon Graphics Computer System, Origin 2000 (1998): Application Development and Optimization. Student Handbook Vol. 1 & 2. Part Number: 02KAPPL-1.2-6.4-S-SD-W. Silicon Graphics Computer Systems.