*S*uper *M*atrix *S*olver-MF:

Direct solver based on Multifrontal Method

February 2011

VINAS Co., Ltd.
Project Development Dept.

URL: http://www.vinas.com

What is *Super Matrix Solver*-MF?

- *Super Matrix Solver*-MF is direct solver for symmetric/ unsymmetric sparse systems of linear equations that is based on Multifrontal Method (MF).

- The solver stably calculates a wide range of sparse linear systems (e.g. matrices with no diagonal dominant, or those with non-symmetric pattern*)

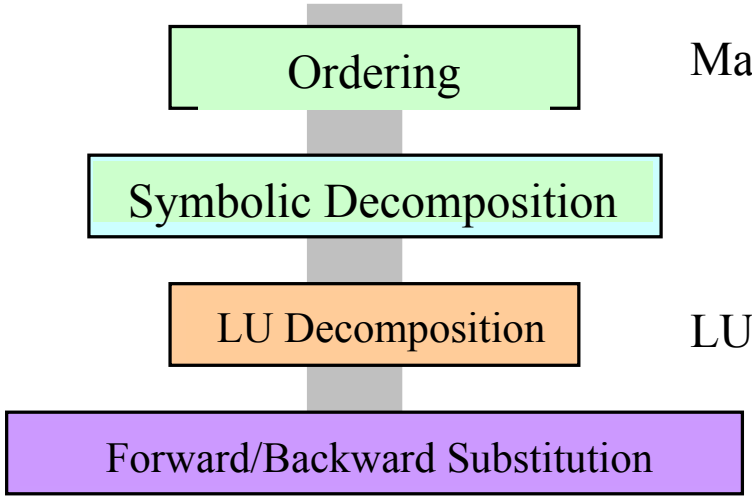
*symmetric element pattern means that the pattern of elements in the matrix is symmetric, such as the location of nonzero-elements; the values of elements do not have to be the same.

Super Matrix Solver-MF Algorithm

- The enhanced ordering algorithm of Super Matrix Solver-MF, which is indicated by the fewer number of fill-ins, allows faster calculation with less memory
- Fill-ins occur when non-zero elements are generated in the zero element portion of the coefficient matrix A.
- When fill-ins occur, the amount of memory use increases, and the amount of calculation also increases.
- The number of fill-ins indicates algorithm improvement: fewer the number of fill-ins, better the performance of direct method.
- It is critical to minimize the number of fill-ins (to keep the sparseness of the original coefficient matrix) for direct method performance.

Super Matrix Solver-MF Algorithm (continued…)

- *Super Matrix Solver*-MF employs combined Nested Dissection – Minimum Degree Ordering algorithm to reduce the number of fill-ins.



Matrix's graph is ordered with ND-MD Ordering

LU Decomposition by MF with full pivoting

- S*uper Matrix Solver*-MF employs Full Pivoting for improving stability of decomposition without increasing the number of fill-ins.
(c.f. general conventional MF-based algorithms employ Partial Pivoting or No Pivoting)

- S*uper Matrix Solver*-MF calculates faster using less memory compared with conventional direct methods.

- Due to optimal memory usage in the decomposition phase and low number of fill-ins, *Super Matrix Solver*-MF can operate possibly largest problems for a particular amount of memory.

Special Features of *Super Matrix Solver*-MF

***Super Matrix Solver*-MF** has the following feature compared with conventional direct method:

- Outstanding Calculation Performance
  - Fast calculation speeds (optimal performance for 2D/3D problems)*
  - Small memory consumption
  - Out-of –Core calculation capability
  - Storing decomposition results in internal memory/hard disk (for later reuse)
    - -Calculates multiple right-hand-side vector ***b*** with fixed coefficient ***A***
    - -Solves equations of slightly different coefficient ***A*** with iterative method
- Easy-to-use Interface
  - Few parameters/arguments
  - Program package includes product manual/sample programs

*Matrices arising from FD/FE problems on structured/unstructured grids.

Specifications of *Super Matrix Solver*-MF(1)

Items	Descriptions	Notes
Intended Matrix	Sparse matrices	
Symmetry of Matrix	Calculates both symmetric and asymmetric matrices. Also supports four symmetry types: symmetric,, hermitian,, skew-symmetric and skew-hermitian.	When calculating symmetric matrix, enter the data from either upper-half or lower-half of the matrix only.
Zero Diagonal Elements	Calculates matrices with zero diagonal elements.	
Types of Unknown	Real and Complex Numbers	
Limit on Number of Elements	Depending on OS, the amount of memory a process can use will be limited.	
Data Format	Executable modules such as DLL.	No disclosure of the source code.
Parameters	Coefficient matrix $\boldsymbol{A}$, right-hand-side vector $\boldsymbol{b}$, upper limit value of memory, etc.	
Solver Integration to Other Programs	Super Matrix Solver-MF may be integrated to the programs that are written in C and Fortran.	Integration to programs that are written in other languages is yet to be confirmed.
Out-of-Core Calculation Capability	Capable of efficiently storing/reading the data for which the amount of memory is insufficient, based on the amount of usable memory. Faster calculation speed compared with simple disk swapping that depends on OS.	

Specifications of *Super Matrix Solver*-MF(2)

Items	Descriptions	Notes
Operation Environment	Windows, Linux	
Input Data	Coefficient matrix and right-hand-side vector	
Output Data	Solution vector	
Indication of Error Messages	Warnings and error messages returned as return value (calculation information, system information, etc).	
Method of Provision	DLL format for Windows; Static library format for Linux and UNIX.	Source code will not be disclosed.
Attached Materials	Product manual (with explanations about data format, parameters, integration procedures, etc.), sample data, sample program for integrating SMS-MF (C and FORTRAN).	

with 1 Gbyte* of actual memory available (for finite element grids of modern topological complexity**):

2D Problems

Not using hard disk: up to 1 million unknowns

Using hard disks: up to 5 million unknowns

3D Problems

Not using hard disk: up to 0.2 million unknowns

Using hard disks: up to 0.8 million unknowns

*1 Gbyte of memory solely available to be used by Super Matrix Solver-MF.

** The limit size can vary several times depending on complexity of particular grid.

Environment supported by *Super Matrix Solver*-MF

Rev A: November 15, 2010

	OS	Recommended Environment	Recommended Compiler	Environment for which operation is noted	Compilers for which operation is noted	Remarks
32-bit machine	Windows	Windows 2000 Windows XP	(1)Fortran ・Compaq Visual Fortran 6.5 and later (2) C / C++ ・Microsoft Visual Studio 6.0 ・Microsoft Visual C++ 6.0 and later		・Intel Fortran 9.0 ・Intel Fortran 9.1	
	Linux	Red Hat Enterprise Linux 4 gcc : 3.4.6 glibc : 2.3.4 kernel : 2.6.9	gcc 3.4.6 Intel Fortran 9.0 and later			Environment under which the module might operate: gcc 3.4.6 and later, glibc 2.3.4 and later, kernel 2.6.9 and later
64-bit machine	Windows (AMD64/ EM64T)	Windows XP x64	(1)Fortran ・Intel Fortran 9.0 and later (2) C / C++ ・Microsoft Visual Studio 2005 ・Microsoft Visual C++ 2005 and later			
	Linux (AMD64/ EM64T)	Red Hat Enterprise Linux 4 gcc : 3.4.6 glibc : 2.3.4 kernel : 2.6.9	gcc 3.4.6 Intel Fortran 9.0 and later			Environment under which the module might operate: gcc 3.4.6 and later, glibc 2.3.4 and later, kernel 2.6.9 and later

Standard Call: Obtaining solutions by a single call, which is the standard procedure.

- rtc=smsmfd(u ad, alu, lnt, lnd, b, nd, ns, mtype, inout)

Multiple-step Call:

When the coefficient matrix ***A*** is fixed, and only the right-hand-side vector ***b*** changes, intermediate steps (inverse matrix) are stored to reassign the changing values of the right-hand-side vector ***b*** only.

- rtc=smsmfd_dec(ad, alu, lnt, lnd, nd, ns, mtype, inout) //Decompose
- rtc=smsmfd_calc(x, b, nd , mtype, inout) //Calculate (fixed ***A***)
- rtc=smsmfd_iter(u ad, alu, lnt, lnd, b, nd, ns, mtype, inout) //Calculate (slightly changed ***A***)
- rtc=smsmfd_free( ) //Delete decomposition results

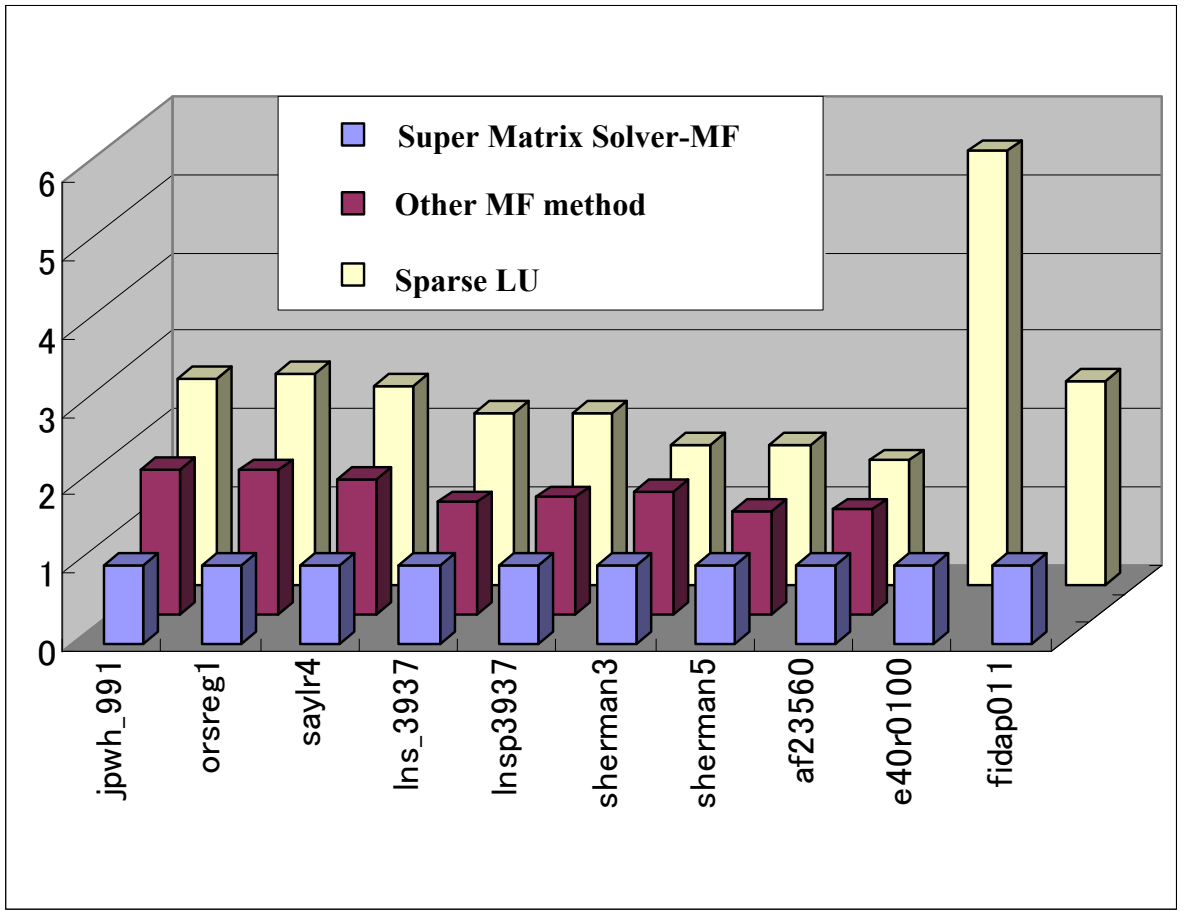
List of Arguments

Argument	Type	Description
u	real*8	Solution
ad	real*8	Store the value of diagonals of coefficient matrix ***A***
alu	real*8	Store the value of non-diagonals of coefficient matrix ***A***
lnt	integer*4	Store the number of non-diagonals in each row of the coefficient matrix ***A***
lnd	integer*4	Store the components of columns of the coefficient matrix ***A***
b	real*8	Right-hand-side vector ***b***
nd	integer*4	Number of unknowns
ns	integer*4	Number of non-diagonal and non-zero elements of the coefficient matrix ***A***
mtype	integer*4	Symmetry type of coefficient matrix ***A***
inout	integer*4	On/Off switch of Out-of-Core feature

Example of Super Matrix Solver-MF Calculation Performance

Comparison of Super Matrix Solver-MF with Other Solution Methods

(suppose the Super Matrix Solver-MF's Fill-in = 1)



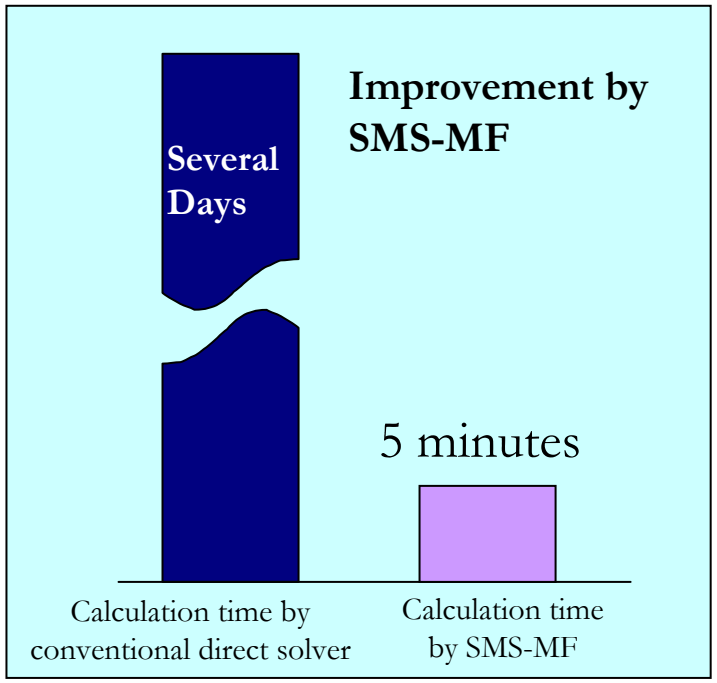
Sparse LU and Other MF Method were compared based on the fill-in performance, not on actual calculation time. Fill-in performance for Spares LU and Other MF method are published in paper, and therefore the comparison was made on the same fill-in performance. Theoretically, the fewer the number of fill-ins, the faster the calculation speed.

SparseLU : http://www.zib.de/visual/projects/par/lulong.en.html

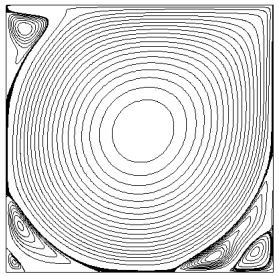
Application Example

Dr. Haruhiko Kohno
Visiting researcher at Massachusetts Institute of Technology
(currently visiting researcher at Coventry University in England)

Application field: Applied SMS-MF to research of fluid flow analysis using FCBI (Flow-Condition-Based Interpolation) finite element scheme

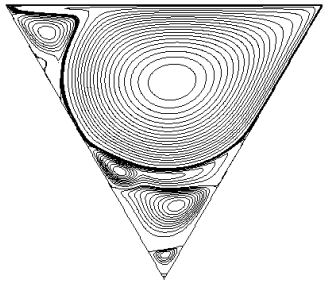


Problems:
Cannot employ iterative solvers due to 0 elements of the coefficient matrix.
Too much calculation time and memory usage of conventional direct solvers such as Gaussian elimination method.
Feature of SMS-MF: Fast calculation speed, Small memory usage.
Able to calculate a 150000-dimension matrix using 1GB-memory.



Re = 10,000

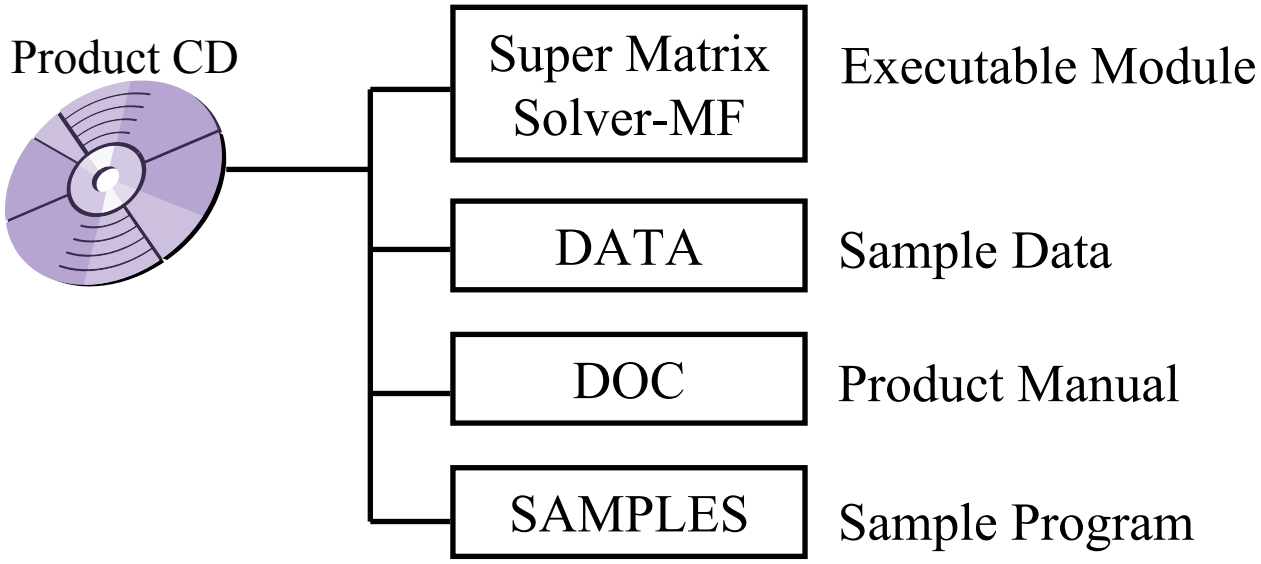
Figure 1: Calculation example (streamline) of 2-dimension stable non-compressed viscosity flow in a square cavity



Re = 5000

Figure 2: Calculation example (streamline) of 2-dimension stable non-compressed viscosity flow in a triangle cavity

Super Matrix Solver-MF software product consists of:



For further information on *S*uper *M*atrix *S*olver-MF such as

- Benchmark Testing (BMT)
- Evaluation module
- Other inquiries

Please contact:

VINAS Co., Ltd. **Project Development Dept.**
Kazuya Goto
URL http://www.vinas.com
E-mail : sms@vinas.com