A Revisit to Linear Solvers in RSS

Bulent N. Alemdar, Ph.D.
Senior Software Product Research Engineer

Abstract

RAMFE is the finite element analysis engine of RAM Structural System and it is used for several products including RAM Frame, RAM Concrete and RAM Elements (formerly RAM Advanse). In past years, ability of walls with openings is integrated into RSS, which involves meshing of walls and enforcing compatibility conditions between orthogonal wall systems. This process is completely automated and it takes care of all meshing details. And recently, lateral analysis of structures with semi rigid diaphragms and gravity analysis with two-way load distribution are added to RAM Structural System. Again, it involves meshing semi rigid diaphragms with openings considered as well as enforcing compatibility conditions between meshed diaphragms and walls. It is observed that these features substantially amplify the size of analytical model created and demand a significant increase for computer resources (particularly a crucial issue for big models). Observing the fact that a substantial amount of computer resources are usually needed for solving “n” number of equations, an efficient solver is desirable to provide a more productive analysis platform. To this end, the engine is recently updated with a new sparse solver to address these performance issues. In this paper, the new solver is compared to the existing solutions in RAM FE and two examples are provided to highlight the merits of the new solver.

A Revisit to Linear Solvers in RAM Structural System

Bulent N. Alemdar, Ph.D.
SR Software Product Research Engineer

Solution of "*n*" unknown equations is one of the most common topics in numerical analysis courses. Basically, the solution follows a typical Gauss elimination procedure, simply eliminating below diagonal terms first (i.e., factorization) and then solving the remaining system for unknowns by backward-forward procedure. In structural engineering applications, there are unique characteristics of assembled systems of equations, namely they form a symmetric matrix with a sparse structure (i.e. sparse matrix means most of terms in the matrix are zero). Most solvers exploit these unique characteristics, which not only provide substantial performance increase in solving equations, but also reduces amount of memory required for storing matrices.

RAMFE is the finite element analysis engine of RAM Structural Systems (RSS). It is currently used for RAM Frame, RAM Concrete and RAM Elements (formerly RAM Advanse). The engine is armed with several solvers, categorized mainly in two flavors: *in-core-solver* and *out-of-core solver*.

The main solver implemented and used extensively until recently is the *in-core direct solver*. It is based on a skyline matrix storage scheme (only the terms below column skylines are stored) supported with a bandwidth optimization solution to reduce bandwidth profile of the matrix. Assembled equations are stored in physical memory (RAM) of computer. The solution is particularly optimized for skyline storage scheme so that it takes full advantage of this type of storage and only works directly on terms inside the column skylines. For moderate size of models (n < 25000), the solution yields good performance. On the other hand, for large models, it may take too long to solve or it may even fail to solve equations due to out-of-memory error (i.e., failing to keep assembled global stiffness matrix in memory). A recent example is Torre Planetarium Towers (Structural Affiliates International. Inc[1]), shown in Figure 1 and it is also one of 2008 BE awards of Excellence projects. The analytical model was so big that the client had out-of-memory error while attempting to solve it. An out-of-core version of the existing in-core solver was developed within a few weeks, which eventually helped the client to finish the project on time. This new solver writes blocks of the matrix into hard disk, and the solution repeatedly accesses the disk during solution process. Even though the new solver helped the client to analyze their model, working off from the disk creates severe performance degradation issues, which easily leads to extended hours of analysis time.

[1] Structural Affiliates International, Inc., www.saii.com/projects/PLANETARIUM/PLANETARIUM.htm

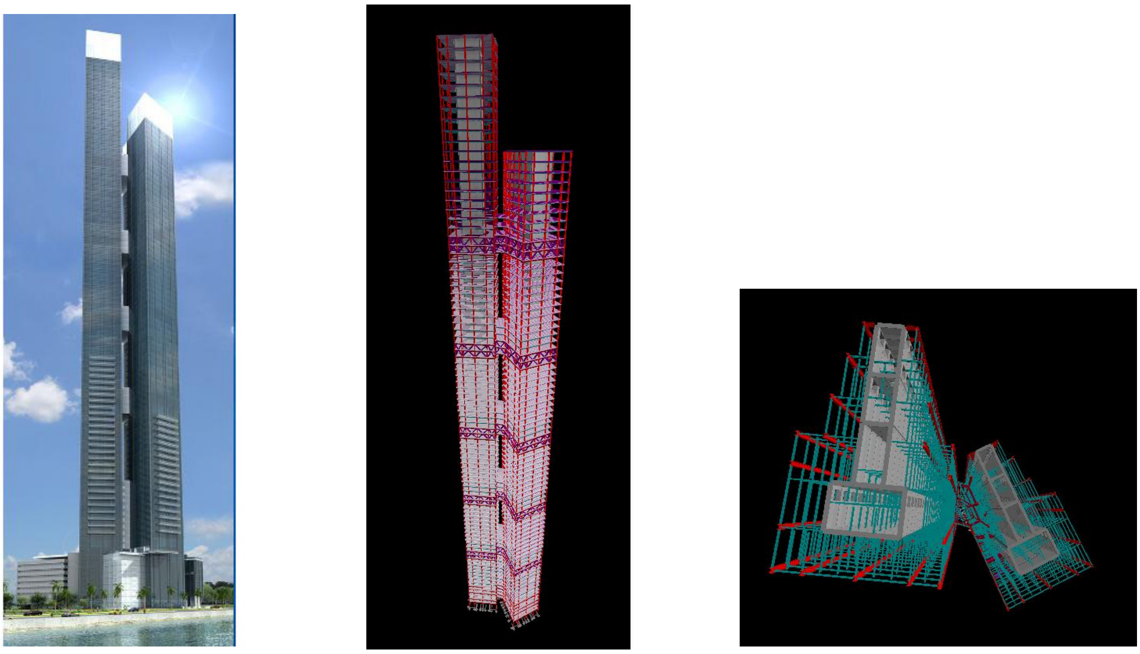

Figure 1. Torre Planetarium, Panama City, Panama (courtesy of Structural Affiliates International, Inc.), RSS analytical model is shown.

Recently, an in-core direct sparse solver (PARDISO[2]) has been integrated into RAM FE. The solver is also a part of INTEL-MKL[3]. The new solver is thread-safe, high-performance, robust, memory efficient and optimized for multi-core computer systems.

For testing purposes, the towers shown in Fig. 1 are analyzed[4]. It is worth mentioning that the model used in this paper is not the exactly the same model used by the client (the client model was extensively modified during design process). Table 1 gives information about number of elements and nodes in the generated analytical model. Analysis performances for both solvers are compared in Table 2. Referring to the new solver, it takes less than *20 seconds* to solve (factorize) the assembled stiffness matrix alone. Total analysis time (including time for assembling global stiffness matrix and global load vector, solving the stiffness matrix and then computing displacements, and finally calculating stresses\forces in all members and then saving results) for a single load case is almost 8 minutes. On the other hand, it takes almost an hour to factorize the assembled stiffness matrix with the existing out-of-core solver. Total time elapsed is 70 minutes to run one load case.

---

[2] PARDISO, http://www.pardiso-project.org
[3] INTEL – Math Kernel Library, www.intel.com
[4] Test computer has the following hardware: Intel Core2 Quad, 2.66 GHz and 3.25 GB of RAM

Table 1. Analytical Model Data for Twin-Tower Model

	Model with Rigid Diaphragms
Frame Elements	6,390
Shells	30,138
Nodes	13,909
Number of Equations	180,576

Table 2. Comparison of Solvers for Twin-Tower Model

	Model with Rigid Diaphragms	
	Existing solver	New solver
Memory Required* (MB)	2720	29.9
Time to factorize assembled stiffness matrix (seconds)	3411	19.9
Time to complete (minutes)	70	8.2

* amount of physical memory to store assembled stiffness matrix

Another example is given in Figure 2, which is used for internal testing purposes only. Model data info is briefed in Table 3. Note that the number of equations is 324,885 and number of non-zero terms in the assembled matrix is approximately 7.5 million. With the new out-of-core sparse solver, it only takes less than *30 seconds* to factorize the assembled stiffness matrix alone.

Table 3. Analytical Model Data for Buddha Example

	Model with Rigid Diaphragms
Frame Elements	28,106
Shells	46,039
Nodes	54,204
Number of Equations	324,885

The new solver provides staggering performance gains, which is a necessary companion for the analysis engine to address big models. It does not only reduce analysis time, but also it is very memory efficient. In v13.0, the new solver is added to RAM FE and available for several products in RSS (see Fig. 3 for RAM Frame). The new solver is offered in two versions: in-core and out-of-core.

Figure 2. Buddha Example (RAM Elements (formerly RAM Advanse))

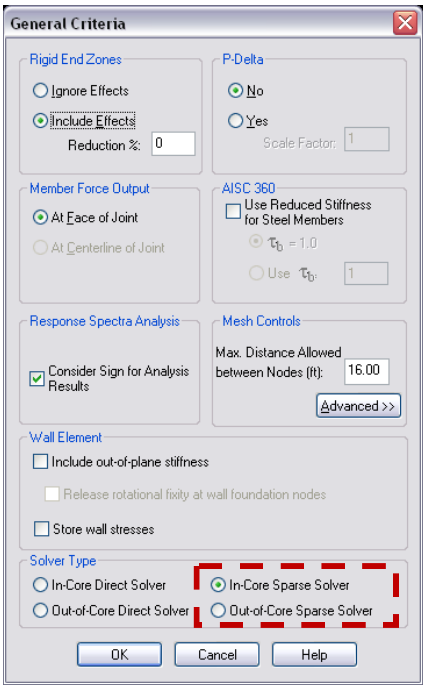

Figure 3. General Criteria dialog (RAM Frame) updated to accommodate the new solver