035067-8-T

HYBRID FINITE ELEMENT AND MOMENT METHOD SOFTWARE FOR THE SERAT ARRAY

6th Quarterly Report

Sanders, A Lockheed Martin Co.
95 Canal Street NCA1-6268
P.O. Box 868
Nashua, NH 030601-0868

April 1998

35067-8-T = RL-2488

PROJECT INFORMATION

PROJECT TITLE: Hybrid Finite Element Design Codes for the SERAT Array

REPORT TITLE: 6th Quarterly Report

U-M REPORT No.: 035067-8-T

CONTRACT
START DATE: October 1996
END DATE: September 1998

DATE: April 10, 1998
(6th Quarterly Report)

SPONSOR: Roland Gilbert
SANDERS, INC, A Lockheed Martin Co.
MER 24-1583
PO Box 868
Nashua, NH 030601-0868
Phone: (603) 885-5861
Email: RGILBERT@mailgw.sanders.lockheed.com

SPONSOR
CONTRACT No.: P.O. QP2047

U-M PRINCIPAL
INVESTIGATOR: John L. Volakis
EECS Dept.
University of Michigan
1301 Beal Ave
Ann Arbor, MI 48109-2122
Phone: (313) 764-0500 FAX: (313) 747-2106
volakis@umich.edu
http://www-personal.engin.umich.edu/~volakis/

CONTRIBUTORS
TO THIS REPORT: T. Eibert(UM), Y. Erdemli (UM), K. Sertel(UM),
D. Jackson(UH), J. Volakis(UM) and D. Wilton(UH)

TABLE OF CONTENTS

CHRONOLOGY of Events (Updated Every Quarter)

• April 1996	Proposal Submission
• July 1996	Answers to Proposal Questions
• August 1996	Began Contract Negotiations
• 20 Sept. 1996	Kickoff meeting at Ann Arbor (attended by Sanders, UM and UH)
• October 1996	Contract Signed between U-M and Sanders in Mid October
• October 1996	Subcontract to the Univ of Houston (formalized in early November)
• 15 Nov. 1996	**SERAT Review meeting** (at Nashua)
• 9 January 1997	**Submitted First Quarterly Report** Report Described Code Plan and Progress on the Moment Method FSS Code. Specifically, a new scheme was developed to accelerate the convergence of the periodic Green's function
• 28 February 1997	**Prepared viewgraphs on the project's progress review**. Showed first validation results for the moment method FSS code with the new accelerated Green's function; showed results for a new algorithm to accelerate the boundary integral truncation of the planar and curved FSS hybrid FEM code using the Adaptive Integral Method(AIM) and CVSS, the new LU solver specialized to sparse matrices
• 5 April 1997	**Submission of Second Quarterly Report.** Report included the first validation results for the stand alone small array FEM code (with dipole FSS elements and dipole antenna elements). A similar validation was done for the moment method FSS developed at Houston. The fast AIM algorithm was described for boundary truncation and the TRIANGLE surface mesher was introduced to generate the aperture mesh, subsequently grown down to the FSS.
• 30 May 1997	**Semi-annual review at the Univ. of Michigan** (attended by all parties) Review covered progress up-to-date. At this meeting, emphasis was on the validation of the three codes which took 'shape and form' between March-May 1997 in accordance with the proposed schedule. Theory, validations and comparisons among the codes were presented.
• 25 June 1997	**SERAT review Meeting** (Nashua)
• 5 July 1997	**Submission of Third Quarterly Report** The major component of this report was the description and validation of the periodic hybrid FEM code, FSS-PRISM. Comparisons among the FSS-EIGER, FSS-BRICK and FSS-PRISM were given for the first time.

Also, the geometry drivers for the FSS-BRICK and FSS-EIGER were given.

- 6 Oct 1997 — Submission of Fourth Quarterly Report

  Delivered FSS-EIGER and FSS-BRICK, both with manuals and preprocessors. Primitives are used to specify antenna and FSS elements in each code. Report gives example calculations for non-commensurate FSS panels for the first time using the scaling approach implemented in FSS-PRISM. Another first is the hybridization of the fast multipole method with the finite element code PRISM as a step toward the curved FSS modeling.
- 24 Oct. 1997 — **Review at the Univ. of Michigan**

  Attended by U-M and Houston project people, Gibert, Pirrung and Asvestas.

  Presented status of FSS-BRICK, FSS-EIGER, FSS-PRISM and FSS-CURVE.

  Delivered manuals for FSS-EIGER and FSS-BRICK (and codes); Successes for non-commensurate array modeling were presented (5 layer example); Update on FSS_EIGER for non-commensurate was given; Initial implementation of finite curved array code was presented with the fast multipole method for mesh truncation.
- Dec 1997 — **5th Quarterly Report**

  The major highlights of this progress report are
  - Implementation of the fast integral method into FSS-PRISM, making prism a practical analysis code, even when the sampling requirements are very high
  - Completion of FSS/Antenna geometry Driver for FSS-PRISM
  - First implementation of curved arrays with the FMM for mesh truncation.
  - Additional validations for non-commensurate FSS
- April 1998 — **6th Quarterly Report**

  The major highlights of this progress report are
  - Delivered FSS-PRISM for modeling SERAT antennas and FSS
  - Delivered Users Manual for FSS-PRISM. Manual included several test cases (input files and examples)
  - Delivered FSSBUILD, the preprocessor to FSSEIGER, the SERAT moment method code developed at the Univ. of Houston.
  - FSSEIGER
  - Curved SERAT code (FSS-CURVE) can now handle transmission and reflection coefficient computations. It has also been upgraded to include lumped loads

MEETINGS

No Meetings were held during this quarter

UPDATED MILESTONE CHART (page 6)

Task	1st Q.	2nd Q.	3rd Q.	4th Q.	5th Q.	6th Q.	7th Q.	8th Q
	Quarterly Progress							
FSS Green's function and Code ***(U of Houston)***	→	→						
Mesh Generator for Antenna Elements	→	→				Delivered		
Mesh generator for FSS elements			→	→		Delivered		
Single Element and Small Array								
Planar and Curved-IBC								
Planar-FEM/Moment Method		→	→	→	Delivered/ parts via ONR			
Curved-FEM for antenna and FSS								
Planar Periodic Array								
FEM with IBCs								
Simple Moment Method code	FSS-EIGER	→	→	→	→	Delivered		
FEM and Moment Method for FSS	FSS-PRISM			→	→	Delivered		
FEM for antenna and FSS	FSS-PRISM			→	→	Delivered		
Curved Array								
Cylindrical								
Approximate Doubly Curved					→	→		
Doubly Curved with fast integral algorithms for mesh truncations	FSS-CURVE				→	→		
Software Integration and I/O Displays				→	→	→	→	
Validation			→	→	→	→	→	
Software Support							→	→

Executive Summary and Project Status

All code modules (see Table below) and code testing are in accordance with the proposed Milestone chart. (refer to page 6). We have already delivered all code modules for FSS-PRISM and FSS-EIGER one quarter ahead of schedule. Also, we run numerous examples and applications (see attached papers and documents). Further, the curved SERAT modeling code is already in working condition and we are currently working on improving its I/O and modeling capabilities.

Of most importance is that the delivered codes are based on state-of-the-art algorithms in terms of speed and memory. More specifically, the codes employ accelerated periodic Green's function computations and fast algorithms (AIM for FSS-PRISM and FMM for FSS-CURVE) for CPU speed-up and memory reductions. These improvements are added to the finite element and moment method implementations, and proved essential to the development of practical codes. Typically these solver and algorithmic improvements result in speed-ups that are two orders of magnitude better in comparison to traditional implementations. It is truly remarkable that only three years ago, these faster algorithms did not exist in any research or industry code.

Below we present a summary status of each module/code.

Summary of Code Modules and Capabilities

CODE	DESCRIPTION AND STATUS
FSS-BRICK	**Small (finite) Array Code** • Finite element-Boundary Integral (FE-BI) code using brick elements • Completed and Validated • Geometry Driver described in Report 035067-5-T
FSS-EIGER	**Moment Method code** • Uses multilayered Green's function and triangular boundary elements (Rao-Wilton-Glisson formulation) • Uses Ewald acceleration for periodic Green's function • Validated (slot and dipole elements) and delivered for commensurate FSS • Geometry Driver Manual delivered in October 1997 • Geometry Driver Manual Updated April 1998 and Delivered • FSSBUILD (FSS-EIGER geometry Driver) delivered April 1998. • FSS-EIGER to be delivered through ONR by Univ of Houston

- FSS-EIGER is capable of Non-commensurate FSS modeling
- FSSBUILD is being upgraded for entering geometries of non-commensurate FSS/arrays

FSS-PRISM

Finite Element-Boundary Integral Code

- Combines flexibility of finite elements for volume and boundary element for robust mesh truncation
- Uses Ewald acceleration for periodic free space Green's function
- Incorporates Adaptive Integral Method for fast Boundary element evaluation
- Uses layer de-coupling to handle non-commensurate FSS.
- Validated for commensurate and non-commensurate FSS and slot and printed antenna elements
- Geometry Driver and code were delivered April 1998 on 2 disks (one for commensurate and another for non-commensurate)
- Manual with test case report were delivered April 1998 (see UM Radiation Lab Report 035067-7-T)
- Theory and many test cases are described in UM Radiation Lab Report 035067-9-T

FSS-CURVE

Curved SERAT array code

- Based on the non-periodic version of PRISM
- Incorporates the fast multipole method (FMM) for fast non-planar boundary integral evaluation
- Tested for planar large array simulations
- Extended to non-planar simulations
- Preliminary testing for curved array simulations has been done
- Transmission and Reflection coeff. can be extracted.
- Incorporates lumped loads (horizontal and vertical)
- FSS Geometry driver is in progress

This Quarter's Summary of Accomplishments

- Delivered commensurate and non-commensurate version of FSS-PRISM
- Delivered complete users manual for FSS-PRISM with 3 example input files and results
- Delivered a new version of SERATBUILD as a geometry and I/O preprocessor for FSS-EIGER
- FSS-EIGER and SERATBUILD are still being upgraded for non-commensurate arrays
- Completed several new antenna and FSS examples as described in UM report 035067-9-T. These examples demonstrate the flexibility of the code and its capability to handle large volumes and complex geometries. This is a primary result of the adaptive integral method for mesh truncation and the accelerated periodic Green's function
- Our finite doubly curved FSS/Array code FSS-CURVE has been upgraded to handle transmission computations. Also, FSS-CURVE can now model vertical and horizontal lumped loads and was modified to handle scattering as well as radiation geometries. It includes the fast multipole method(FMM) for mesh truncation and this leads to $O(N^{1.5})$ CPU even for curved structures. We are currently in the process of validating FSS-CURVE for large arrays with curvature. We also need to add resistive sheet capability, improve I/O and write a users manual.

Project Goals

The goal of the SERAT project at the University of Michigan (with subcontract to Univ. Of Houston) is to develop a suite of software for the analysis of strip and slot dipoles on multilayered substrates backed by a frequency selective surface. The dipoles are equipped with photonic switches permitting variable electrical dipole lengths for broadband performance and the FSS is suitably designed to simulate a variable substrate thickness for optimal operation. A general view of the geometry is given in Figure 1.

The UM/UH team proposed to construct a code which combines various computational modules interfaced with appropriate pre-processors and post-processors. The computational modules include:

- Stand-alone moment method simulation of the FSS with up to 10 layers with commensurate and non-commensurate periodicities.
- Simple moment method simulation of the antenna elements on the FSS panels
- Hybrid FEM simulation modules for small arrays, planar periodic arrays and curved arrays on FSS panels.

Various options for modeling the FSS and for mesh truncation were proposed to provide a compromise between speed and accuracy. These are outlined in the proposal and summarized in the included milestone chart (repeated from the proposal).

As called for in the milestone chart, we are proceeding in accordance with the schedule in our proposal. In most cases we are ahead of schedule by one quarter or so. Our many examples and validations are testaments to the practical utility and speed of the codes. The FE-BI codes include algorithm speed-ups based on fast algorithms. These can deliver as much are 2 orders of magnitude in CPU speed-up and memory reduction.

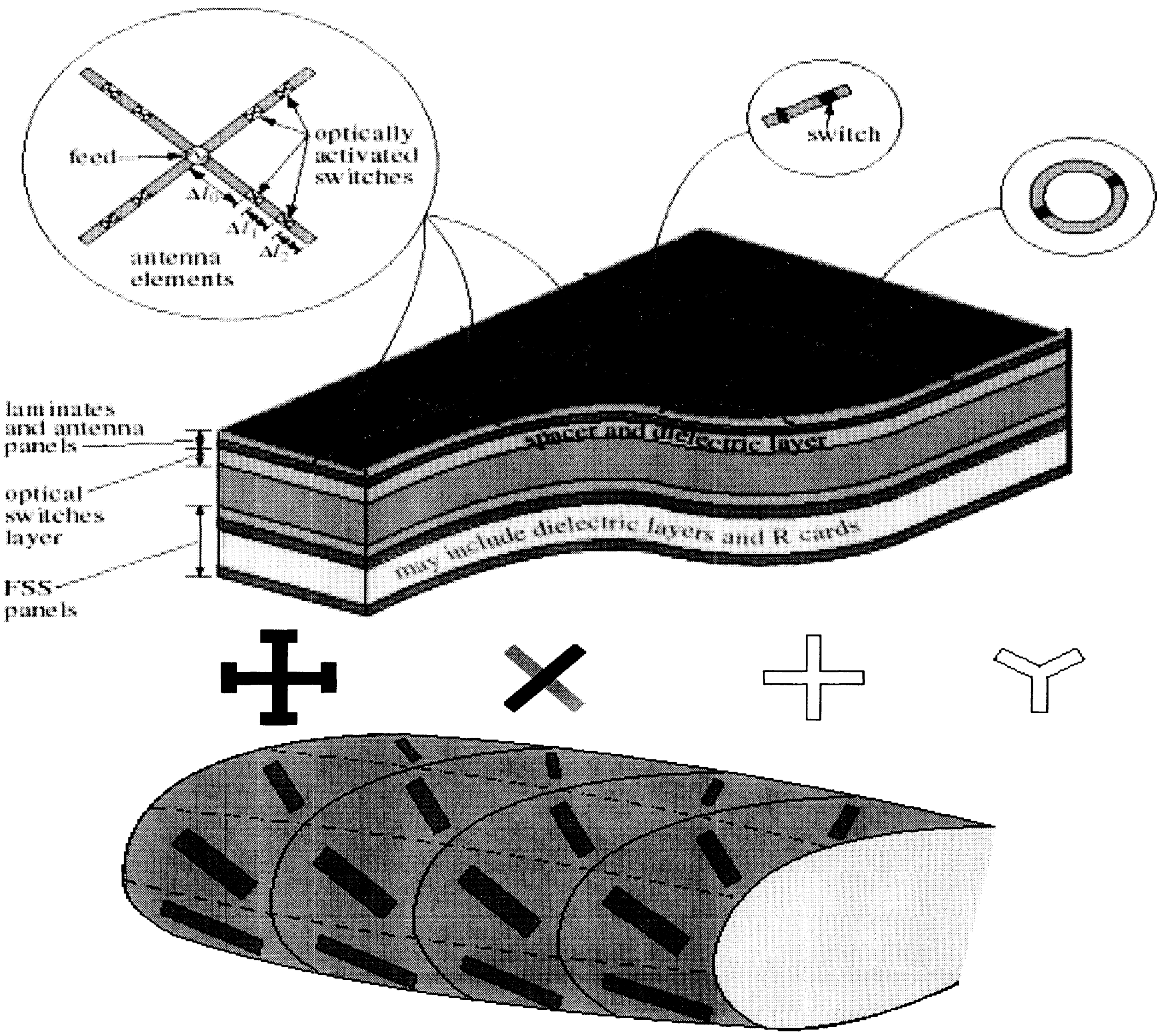

Figure 1. Illustration of the SERAT panel (Planar and Curved)

FSSEIGER — Current Progress by the University of Houston

The FSSEIGER code suite has been slightly reorganized. Since the code EIGER requires separate licensing, the pre- and post-processing subroutines have now been completely separated from EIGER for ease in distributing the code in two separate parts. It is envisioned that FSSEIGER would be treated as an application suite comprising the modules FSSBUILD and FSSEIGER, to be separately distributed from EIGER, but requiring a licensed copy of EIGER to complete the suite.

The FSSEIGER suite now comprises the following three codes:

- FSSBUILD – The pre-processor module, which is used to define the problem geometry and to prescribe the excitation parameters. FSSBUILD is essentially a user interface used to create an EIGER input file for modeling frequency selective surfaces and phased arrays with specific elements.
- EIGER – The general purpose code EIGER; its only specialization to FSS and array problems is the addition of a call to FSSEIGER, the post-processor.
- FSSEIGER – The post-processor module, which is used to write the problem output data file and to determine array or FSS parameters appropriate to the problem.

A table summarizing the current capabilities of the three modules follows. It is noted that the general purpose code EIGER supports a number of features that are available to the user via the EIGER input file, but which may not be currently supported by FSSBUILD or FSSEIGER. These features are listed because they may be capabilities which are of future interest for phased array or FSS design. (EIGER has a number of additional features which are not listed, but which are not considered particularly pertinent to array or FSS design.)

FSSBUILD, EIGER and FSSEIGER Capabilities

Features	FSSBUILD	EIGER	FSSEIGER
EXCITATIONS:	▬	▬	▬
Plane wave	√	√	√
Voltage source	√	√	√
ELEMENT GEOMETRIES:	▬	▬	▬
Planar dipole	√	√	√
Planar cross	√	√	√
Planar slot	√	√	√
Planar cross slot	√	√	√
SUBDOMAINS:	▬	▬	▬
Triangular	√	√	√
Rectangular		√	√
Wire		√	√
Wire/triangle junctions		√	√
BOUNDARY CONDITIONS:	▬	▬	▬
EFIE	√	√	√
MFIE	(slots only)	√	√
CFIE		√	√
Dielectric (PMCHW)		√	√
LOADING:	▬	▬	▬
Lumped impedance	√	√	√
Surface impedance		√	√
GREEN'S FUNCTIONS:	▬	▬	▬
Homogeneous nonperiodic		√	
Layered media:	▬	▬	▬
Planar periodic (skew lattice)	√	√	√
Noncommensurate periodic		√	√
Nonperiodic		√	

Parameters Computed by FSSEIGER

PLANE WAVE EXCITATION:	▬
Equivalent currents	√
Reflection coefficient	√
Transmission coefficient	√
VOLTAGE EXCITATION:	▬
Equivalent currents	√
Input power per element	√
Active impedance & admittance	√
Array element pattern	√

Update on FSS-PRISM and Geometry Driver

FSS-PRISM was delivered April 1998

The geometry driver along with test cases and capabilities are described in the UM Radiation Laboratory Report 035067-7-T

The theory of the FSS-PRISM code is described in the UM Radiation Laboratory Report 035067-9-T. This report excludes details pertaining to the fast integral. These details are currently being written Many test cases are included in the report demonstrating various practical uses of the code, including band gap antennas as well as multilayered FSS (up 8 layers have been modeled so far).

Curved Finite Array Code(FSS-CURVE) Update

K. Sertel

The development of the computer code FSS-CURVE for the solution of finite conformal FSS structures is being continued as scheduled. As of April 1st, the following features were added.

- Till now the code was able to handle structures embedded in a finite size cavity residing in an infinite ground plane. The cavity was metal backed and therefore only reflection and radiation problems were considered so far. Currently the code handles FSS transmission computations as well. This was done by opening the base of the cavity. Basically, same boundary-integral truncation was implemented for the bottom surface of the cavity. As was for the top aperture, fast multipole method (FMM) is employed to speed-up the solution time and keep memory requirement low.

- Once the solution of the finite-size open-backed cavity problem is obtained, the equivalent surface currents on the bottom surface are radiated to find the transmission coefficient of the structure. For this computation we assume that the equivalent currents on the surfaces are periodic. Our computations were verified by comparing the results with the FSS-BRICK and the FSS-PRISM results (see Fig 1).

- For radiation computations, the user can specify arbitrary number of probe feeds, impedance loads and shorting pins. During this quarter, we added capability for modeling impedance loads and shorting pins.

1 Results

To validate the code, the single layer FSS structure with slot elements (also presented in the FSS-PRISM manual) is analysed using FSS-CURVE. The results obtained show very good agreement with those obtained using FSS-BRICK code and the infinite array code (see Fig. 1). This 2 × 2 finite slot array is then wrapped on a 3.82 cm. radius cylinder to form a conformal structure. The transmission coefficient of the curved structure is also presented in Fig 1. Since the transformation used to obtain the mesh of the curved structure does not change the horizontal length of the slots, we did not observe a shift in the resonance frequency of the structure.

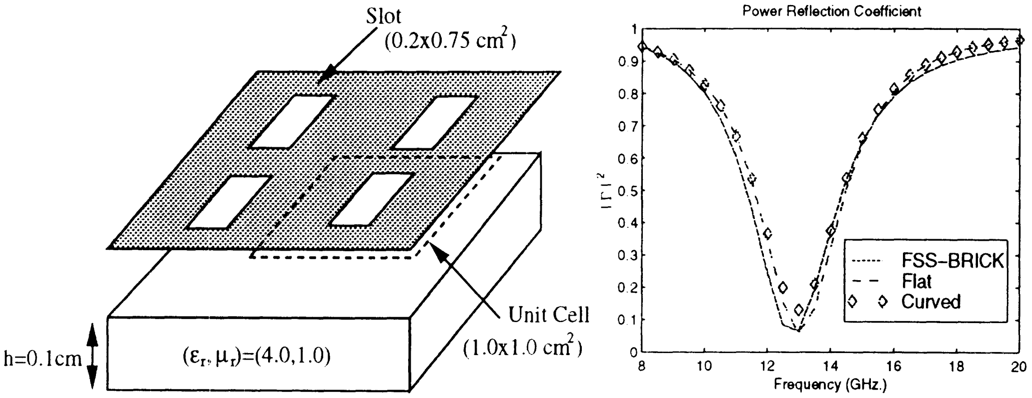

Figure 1: Reflection coefficient for the single layer FSS with slot elements.

To obtain quantitative results regarding the complexity of the FE-FMM implementation compared to the conventional FE-BI method, we obtained the dependencies of the matrix fill and matrix solution time as well as time consumed per iteration and the required memory. Using curve-fitting algorithms (LSQ), we generated a quantitative comparison between the computational complexities of the two methods. The various CPU time and memory curves are presented in Figs. 2-3.

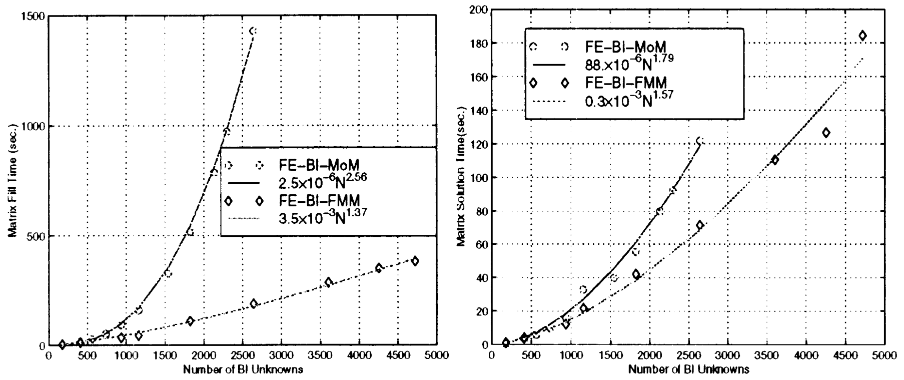

Figure 2: Matrix filling time and iterative solution time of conventional FE-BI and FE-FMM algorithms.

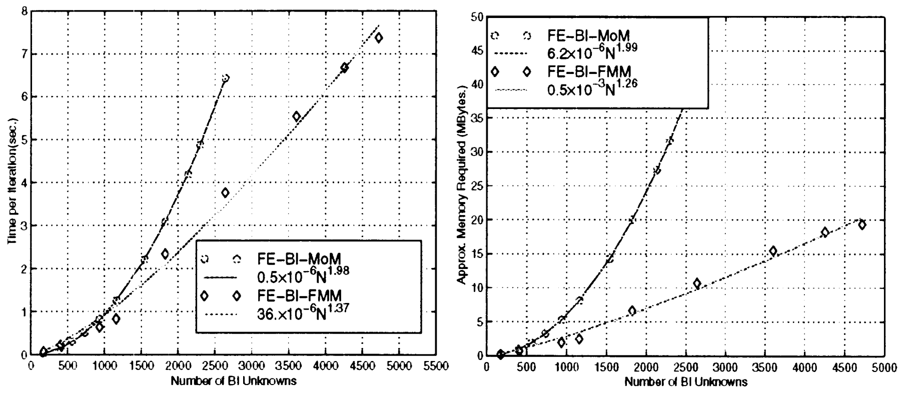

Figure 3: Time consumed per one iteration and memory required for the matrix for conventional FE-BI and FE-FMM algorithms.