Visual Computing Concept in Finite Element Analysis

Hideo Yamashita, Vlatko Cingoski, Mitsuru Mikami and Kazufumi Kaneda
Faculty of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-hiroshima 739 JAPAN

Reprinted from
IEEE TRANSACTIONS ON MAGNETICS
Vol. 33, No. 2, March 1997

Visual Computing Concept in Finite Element Analysis

Hideo Yamashita, Vlatko Čingoski, Mitsuru Mikami and Kazufumi Kaneda
Faculty of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-hiroshima 739 JAPAN

***Abstract*—A new FEA approach utilizing the concept of visual computing is proposed. Conventional FEA is improved by a new visual computing method. This new method is based on rapid data interchange among program modules through the use of shared computer memory. The concept enables easy control of the analysis process through interactive user input. It also provides the possibility for on-line visualization and animation in real time of various one-dimensional scalar quantities and one-dimensional and two-dimensional distributions, using a fast computer-graphics engine. By visualizing the numerical data in progress, the user can easily detect errors in the initial data, achieving rapid and accurate analysis with reduced computation cost. The proposed approach is also very useful for analysis of time-periodical, nonlinear or any other problems having many successive iterations and solutions.**

I. Introduction

FEM is widely employed in the field of engineering for analysis, design and optimization problems. With the development of computer technology, as well as hardware and software techniques, even significant three-dimensional problems can be solved easily using moderately priced and readily available computers. The conventional FEA method based on a modular solution of a problem – such as pre-processing (mesh generation and material properties assignment), processing (solution of the problem) and post-processing (visualization of the results) – is very common. This traditional approach, however, usually is time-consuming and memory-costly.

In this paper, a new FEA approach that utilizes the method of visual-computing (VC) [1] is proposed. The conventional modular concept is enhanced and enriched by developing a new VC concept based on data interchange between program modules through shared computer memory. The conventional modular FEA concept enables visualization only after the application program is entirely executed and the final results are obtained. It requires a restarting of the application program whenever a parameter or a value of the input data is changed. Therefore, if execution of the application program is time-consuming, which is almost always the case with FEA, total computation cost for obtaining the desired solution can be exceedingly high.

Manuscript received March 18, 1996.

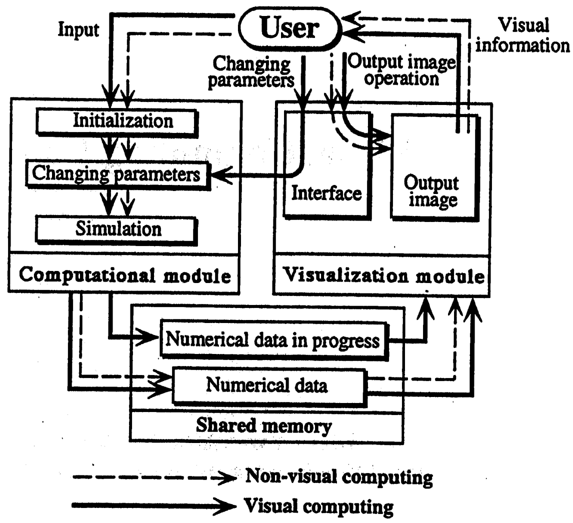


Fig. 1. Visual computing concept in FEA.

On contrary, VC enables easy control of the analysis process by visualizing the obtained results at any computational step, using numerical data in progress, thus speeding up the computation process and increasing its effectiveness. Visual computing also enables easy adjustment of parameters during computation when the need arises. Such parameters can be connected directly with the analysis problem or with the method for its solution. A simplified block diagram of the proposed VC concept in FEA is presented in Fig. 1.

Here, we treat a system of processing and visualization modules connected through shared computer memory. The shared memory maintains a constant connection linking numerically obtained data, visualization data and the operation that changes parameters in the analysis. At the top of the VC system is a user who controls the entire process according to the visualized results and the user's experience and desire. In Fig. 1 the dashed line represents the conventional, non-visual computing approach that, after completion of each module (i.e. computation), stores the obtained results in an external storage device. The results are then visualized through the use of a visualization module. The VC method, represented by the solid line in Fig. 1, performs continuous data flow between the computation and visualization modules, without the need for additional storage procedures. It also enables con-

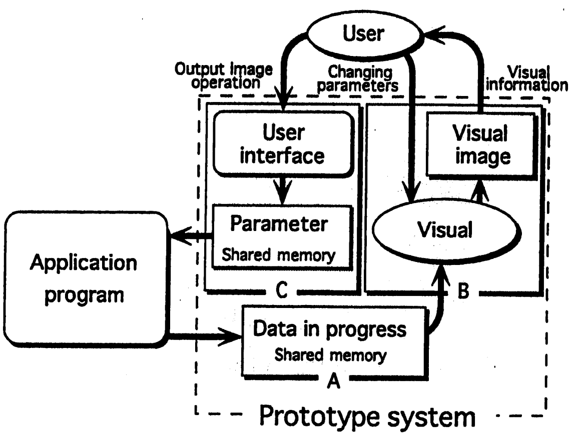


Fig. 2. Developed prototype VC system for SGI workstation.

tinuous feedback among the computational module, the visualization module and the user. Visual computing has another advantage: it does not require an additional storage device, as the results are visualized while the analysis proceeds. Storage of the final solution, however, does require a storage device.

In what follows, first, we introduce the basic ideas of the VC method and describe in detail the main components and the numerical implementation of the proposed prototype VC system. Next, we present two successful applications of the proposed VC concept for one-dimensional electrostatic and two-dimensional magnetostatic field analysis using the Hopfield neural network (HNN) as a solution method for minimizing the energy functional. Finally, we derive the conclusions and discuss points for further research and extension of the proposed system.

II. Developed Prototype of Visual Computing System

As mentioned above, the developed VC system prototype enables easy control over parameters, the need for which usually arises during the computational process. This control is enabled through shared computer memory maintaining a constant link among numerically obtained data, visualization data and the operation that alters several analysis parameters. All this can be performed interactively by the user, who constantly controls and regulates the entire computational process. To enable this operation, the VC system is constructed of several specific components or modules. A simplified block representation of the generated VC system prototype for an SGI workstation is presented in Fig. 2. As shown in Fig. 2, the proposed VC system consists mainly of three modules: a data sharing module (A), a visualization module (B) and a control module (C).

Transfer of the data in progress between the application program and the visualization program is performed through module (A) using shared computer memory. Visualization of the results is executed using module (B). Finally, by employing user interface module (C) the user can pass on the control information directly to the application program through the shared memory. Therefore, the interactive control of the application program and the feedback of the application program can be obtained simultaneously using the shared computer memory and the user interface. Each of the three main modules is explained in detail below.

A. Data Sharing Module – Module A

The main purpose of this module is to carry out data exchange between the application program and the visualization program using shared computer memory. The shared memory is the standard unit of communication in model V of the UNIX operating system, in which two or more processes can share the same data. When one application program sends data to the system memory, for example, the data is stored temporarily in the system's shared memory. Any other application program running at the same time is able to read the data directly from the shared memory space and process it further. In the developed VC system prototype, the temporary output data of the application program which performs FEA is visualized using the visualization module (B).

B. Visualization Module – Module B

In this module, the data flow from the application program is visualized through the pipeline established by shared memory. Depending on the application program for data visualization, the user can visualize: changes in visualization data over time, in space, or both; the change of convergence data, such as the residual value or the energy of the model; or the distribution of the unknown variables to better understand their physical appearance. The developed VC system prototype is designed to handle the following classes of visualization data:

1) Display of a Scalar Quantity as a Graph: This function allows the display of arbitrary scalar variables that change over time, such as the application program's convergence process. For example, when minimizing the energy functional using the neural network approach, it is useful to continuously observe changes in the objective function and its minimization. When the convergence rate of the iteration process is slow, the value of the parameter of the input-output function can be changed easily. A sample graphic window generated by the proposed VC system prototype is presented in Fig. 3. Not only is observation of the iterated energy function possible, but also the user can control interactively the entire iteration process with a user-friendly interface. The prototype has also the following functions: graph movement, zooming in or out, displaying any part of or the entire region, changing the maximum and minimum bounds of the displayed variables, setting grid-on and grid-off, simultaneously representing more than one variable on the same graph, etc. All these functions can be operated easily with a mouse or a keyboard.

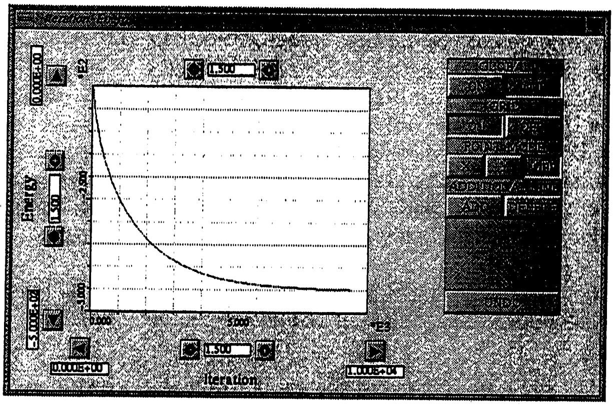


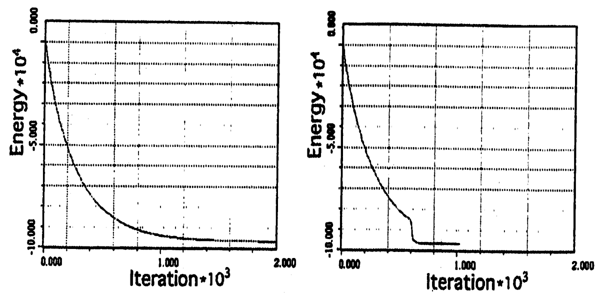
Fig. 3. Display of a scalar quantity as a graph.

2) Display of a Scalar Distribution in 1-D Space: This function enables visualization of any scalar physical quantity treated in the application program as an unknown variable in one-dimensional space (see Fig. 5). Due to the constant flow of output data from the application program to the visualization program, changes in physical quantity can be observed as animation in time. The user can therefore observe clearly how the solution of the problem continuously improves with time as the process converges.

3) Display of a Scalar Distribution in Two-dimensional Space: Two-dimensional quasi-color visualization of two-dimensional scalar variables is common in FEA. The VC system prototype enables simultaneous visualization of any two-dimensional physical quantity and its animation in time (see Fig. 6). Rapid visualization of two-dimensional distribution is accomplished using the computer's graphic engine. First, computation of the values of the unknown variable at each node of the two-dimensional finite element mesh of triangles is performed in the application program. Next, the problem's intermediate solution is transferred to the visualization program using the shared memory. The visualization program first associates the value of the unknown variables with the appropriate colors from a color palette. Next, using an SGI workstation's fast graphic engine, the visualization program interpolates the unknown variable inside each finite element, utilizing known nodal values. Finally, the generated two-dimensional distribution is displayed in a specially generated window. Through continuous performance of this process, animation of the physical quantity can be displayed nearly simultaneously with the numerical computation.

C. Control Module – Module C

This module enables control over the application program by sending various commands directly from the user interface using the shared computer memory. With the control panel of the user interface, the user can control the flow of information and data between the application program and the visualization program. The control can be obtained using various commands: *Start* the application, *Pause* the application, *Stop* the application, *Change* the parameters and *Quit* the VC system. The module's control panel can be operated easily with a mouse or a keyboard.



a) for constant k b) for variable k

Fig. 4. Display of the Energy-Iteration characteristics.

III. Applications

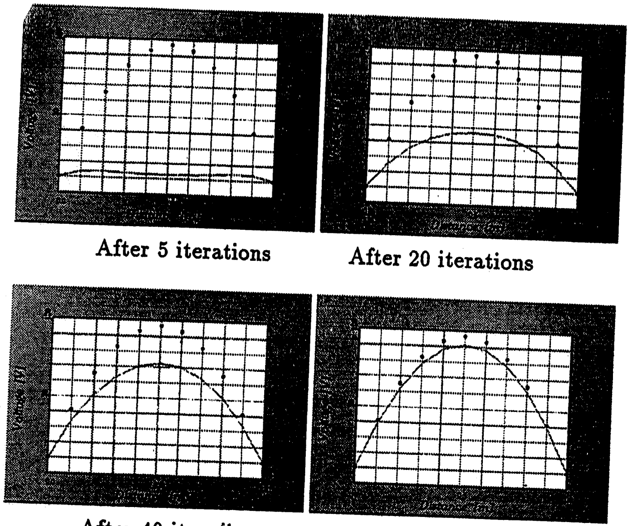
Using the HNN with linear input-output function $Y = kX$ for minimization of the energy functional arising from the finite element approximation [2], the VC system prototype was applied to problems dealing with the visualization and animation of a one-dimensional electrostatic field and a two-dimensional magnetostatic field.

A. One-dimensional Electrostatic Problem

A simple one-dimensional electrostatic model with a theoretically known solution was used to verify the VC system prototype. In [2], it was shown that parameter k in the input-output function $Y = kX$ has a large influence on the iteration process; even a small change in its value can result in a significant increase in the speed of the computation process. Therefore, by using the proposed VC concept, it is possible to change parameter k by monitoring the HNN minimization process. The obtained results for the minimization process are presented in Fig. 4 for constant coefficient k and for varying coefficient k, respectively. It is clear that when the user changes the value of the parameter k, the minimization process becomes faster. As the format of this paper does not allow the animation of the obtained results, which continues with the minimization of the network's energy, several intermediate frames are presented in Fig. 5. The dots in Fig. 5 represent the theoretical solution of the problem, while the continuous line represents the numerically obtained results at each iteration step. Clearly, as the network evolves with time, the numerically obtained results converge toward the problem's theoretical solution.

B. Two-dimensional Magnetostatic Problem

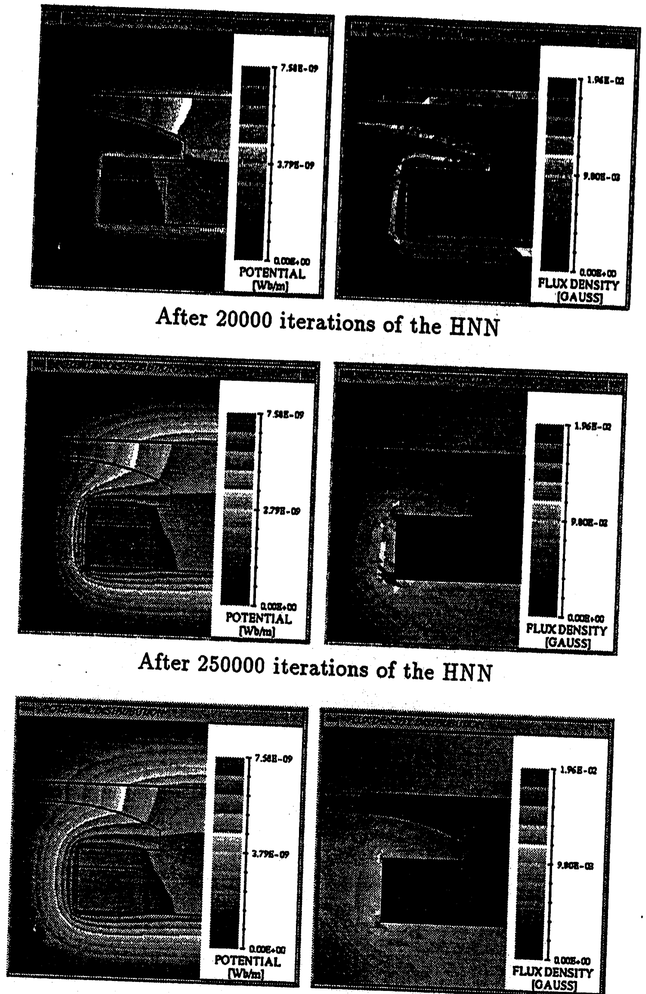
The proposed VC system was also applied successfully for visualizing the magnetic vector potential and magnetic flux density distributions of the half-pole area of a salient



After 5 iterations After 20 iterations

After 40 iterations After 80 iterations

Fig. 5. Comparison between theoretical and numerical solutions at several iteration steps.



After 20000 iterations of the HNN

After 250000 iterations of the HNN

Final results

Magnetic vector potential Magnetic flux density

Fig. 6. Visualization of magnetic vector potential and magnetic flux density distributions for two-dimensional magnetostatic model.

pole rotating machine. Figure 6 shows the obtained distribution for magnetic vector potential and magnetic flux density at two typical iteration steps during analysis, and at the end of the analysis.

From Fig. 6 the following conclusions can be derived: At the beginning of the calculation process the values of the magnetic vector potential are established more easily in the air-region than in the magnetic material region. As the computation evolves, however, the changes in the nagnetic vector potential are observed mainly inside the nagnetic material. On the other hand, due to an initially large difference between the magnetic vector potential values inside the magnetic material and those in he air-region, the magnetic flux density values are large long the boundary between the magnetic material and he air-region. Later, as the solution process continues, he maximum values for magnetic flux density decrease, s the high-field area increases toward a region of large ermeability.

It is apparent that the proposed VC system enables easy ontrol and understanding of the application program by isualization of the results while the program progresses. ı addition, the proposed method allows the easy detecon of any errors in the input-data and their correction, hich is advantageous for decreasing computation time ıd cost.

IV. Conclusions

We proposed and developed a new prototype of a vial FEA computing system that uses shared computer emory. The proposed system enables interactive conol and changes of various parameters in the application ogram, as well as the visualization and animation of varıs one-dimensional and two-dimensional scalar distributions almost in real time. The system is also very useful for on-line detection of any errors in the input-data and for their correction, decreasing computation time greatly. Further research is necessary to improve the user interface, to generalize the entire system and to upgrade its visualization of vector distributions in two-dimensional and three-dimensional space.

References

[1] R. Marshall, J. Kempf, S. Dyer and C. Yen, "Visualization Methods and Simulation Steering for a 3D Turbulence Model of Lake Erie," *Computer Graphics*, Vol. 24, No. 2, pp. 89-97, March 1990.

[2] H. Yamashita, N. Kowata, V. Čingoski and K. Kaneda, "Direct Solution Method for Finite Element Analysis Using Hopfield Neural Network," *IEEE Transactions on Magnetics*, Vol. 31, No. 3, pp. 1964-1967, May 1995.