COMMUNICATIONS IN APPLIED NUMERICAL METHODS, Vol. 4, 373–378 (1988)

ON SOME RECENT EVOLUTIONS IN PERSONAL SUPERCOMPUTING AND WORKSTATION GRAPHICS

J. F. HAJJAR,* L. F. MARTHA,† T. F. O'CONNOR‡ AND J. F. ABEL§

Program of Computer Graphics, Cornell University, Ithaca, NY 14853 U.S.A.

SUMMARY

Recent hardware advances for both enhanced computing and interactive graphics are used to improve the effectiveness of three-dimensional engineering simulations. The two examples presented, drawn from structural engineering, deal with the fully nonlinear transient dynamic analysis of frames and boundary element stress analysis.

INTRODUCTION

Engineers in both research and industry are currently faced with rapid changes in the hardware environment for computational mechanics. On the one hand there is the rapid development of enhanced computing devices, ranging from attached processors to innovative stand-alone vectorized or parallel computers to giant supercomputers. On the other hand, there continue to be dramatic improvements to engineering workstations which combine a devoted processor with large amounts of local memory and disk storage, with networking capabilities, and with integral high-performance graphics including hardware implementations of display algorithms.

These two lines of development are strongly related when it comes to the needs of the engineer engaged in computational simulation. In particular, the combination of local processing, advanced graphics, networking and enhanced computing provides — in principle — the capability to address multidimensional problems of greater size and complexity than has previously been possible.

In choosing among the sometimes bewildering number of alternatives which offer improved performance and capability, the computational engineer will be governed by his particular needs and resources. The purpose of this brief paper is to present some examples of choices, and the resulting achievements, made in the writers' engineering research. While the directions and experiences indicated herein are clearly of interest to others in the computational engineering community, the writers do not pretend that the approach described is a panacea.

The two specific complementary directions described here are the choice and implementation of enhanced computing for structural and stress analysis, and the adaptation of high-performance workstation graphics to visualize and interpret stress analysis results. The choice for enhanced computing lay between local minicomputers, a readily accessible minisupercomputer with a small user group, and a nearby supercomputing mainframe with a large user base. The last two alternatives each entailed some recoding of programs originally developed on the minicomputer. The choice was determined mainly by the desire of the individual researcher to obtain the fastest possible elapsed-time turnaround (rather than the fastest CPU time), particularly for applications requiring adaptive and/or repeated analysis (e.g. as is needed in fracture propagation simulations) and eventual graphical monitoring of calculations as they are being performed. Therefore the local minisupercomputer, an FPS–264 attached processor, was selected. Because this device is shared by only a few users, its use may be termed 'personal supercomputing'.

With the improvement of simulation size and complexity achieved by this personal supercomput-

* Graduate Research Assistant, Department of Structural Engineering.
† Graduate Student, Department of Structural Engineering.
‡ Visualization Engineer/Technical Specialist.
§ Professor of Structural Engineering (Correspondent; address: 120 Rand Hall).

0748–8025/88/030373–06$05.00

Received April 1987

ing, the need for rapid visualization became more acute. To this end existing three-dimensional colour postprocessing, developed earlier for a raster graphical display hosted by a minicomputer, was extended to exploit the higher-performance graphics of a modern engineering workstation. The particular workstation was an HP 9000 Series 320 with a 320SRX graphics engine.

PERFORMANCE GAINS BY PERSONAL SUPERCOMPUTING

The results of two studies of the use of an FPS–264 advanced processor for structural analysis are briefly presented. The FPS is attached to a VAX 11/750 which acts as a host machine. Access to this attached processor is obtained by running a program on the host machine which calls subroutines of the application program that are downloaded and executed automatically on the FPS. The host may operate asynchronously with the attached processor, and this parallel arrangement allows flexibility in achieving an optimal division of tasks between the two computers.

The first application investigated was a fully nonlinear dynamic time history analysis of three-dimensional steel framed structures subjected to nonproportional multicomponent timedependent loading, and, in particular, to earthquake loading. Hilmy[1] originally developed a fully interactive and adaptive analysis program for this application.[1–3] The analysis runs on a VAX 11/780 or VAX 8300 and uses three-dimensional computer graphics for all user interaction (a vector refresh graphics display and a digitizing tablet and pen are used for this graphical interface). Beam-column elements are used for the finite element model, with direct time integration. Both geometric and material nonlinearity are taken into account for the duration of the loading, so that a new coefficient matrix must be formulated and solved during each timestep of the dynamic analysis (see References 1–3 for details of this finite element model). For problems with more than about 1000 degrees of freedom, with about 100 or more timesteps, this type of analysis is too time-consuming on the VAX.

The transformation to the FPS system of the approximately 20 000 line code took about four person-weeks and required three major changes. First, both the original and the revised program are written completely in FORTRAN; however, the minor differences between VAX FORTRAN and the standard FORTRAN-77 language used in the FPS had to be reconciled. Second, software provided with the FPS contains a number of features which accommodate this code development. For example, a library of mathematical functions,[4] such as vector additions, dot products and matrix multiplication may be easily incorporated into the finite element code. An in-core direct solution subroutine of this library is also used to solve the incremental displacements at each timestep. Most importantly, the use of these routines insures the optimal use of the attached processor hardware architecture.

The third change to the code was to establish an interface between the VAX host and the FPS. The flowchart shown in Figure 1 indicates the division of tasks between the host machine and the attached processor. The driver program on the host computer first reads the input data for the problem from its disk and then calls a subroutine to begin the dynamic analysis on the FPS. All input is passed to the FPS through the arguments of this call statement. This method of transferring information to the attached processor is significant for two reasons. First, it greatly simplifies communication of information between the host and the attached processor by allowing the interface to be no different than calling any other subroutine. Second, since the floating point format of the two computers is different, this direct calling mechanism allows built-in hardware to transform automatically all floating point data to the FPS format.

During the course of the analysis, the response data are written to the disk provided with the FPS. After the analysis is complete, control returns to the host, which then calls a subroutine on the FPS, which retrieves the data from the FPS disk and returns the data to the VAX through the arguments of its subroutine statement. Hardware conversion of the floating point numbers to the VAX format is done automatically during this return. The host then asynchronously writes this response set to the VAX disk while the next set of responses is retrieved from the attached processor.

The FPS analysis may be started by a user from the VAX during a session using the interactive graphics program by activating an appropriate menu item (see Reference 3 for a description of the interactive graphics interface). The user may continue any other work with the interactive program while the analysis is executing. When the analysis is complete, the user is notified and, during the same session (or, if desired, at a later time), the results may be viewed by using the

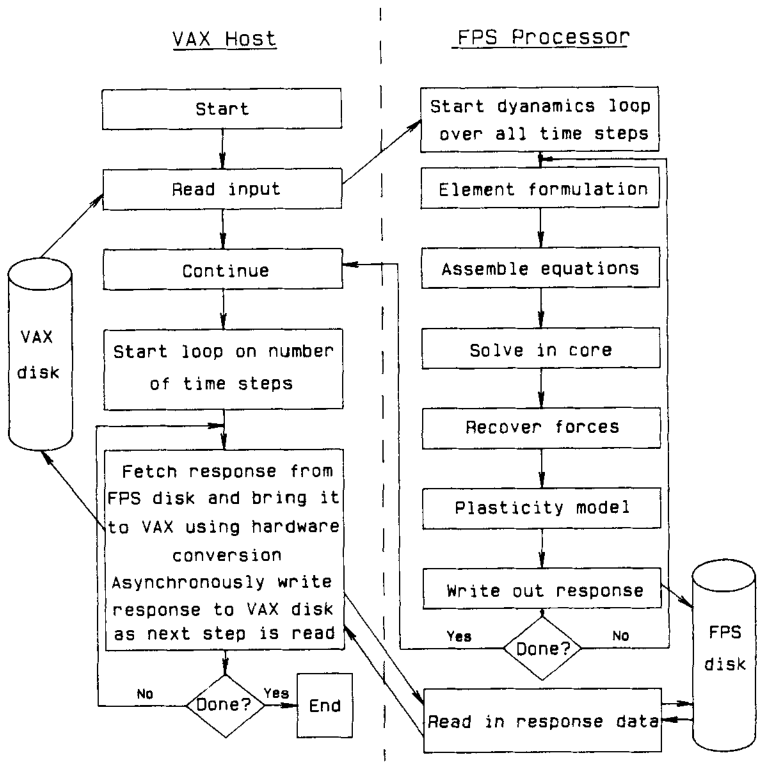

Figure 1. Flow of control for nonlinear dynamic frame analysis in a host/attached processor environment

dynamic playback capabilities of the program. This interactive environment provides a simple but effective interface to the attached processor.

The timing statistics for a number of steel frame problems of various sizes are summarized in Table I. The increase in speed over the VAX 11/780 is evident. Just as the solution time of a set of equations (and therefore of the total finite element analysis) increases approximately with the square or cube of the size of the problem, so too does the speed-up of the FPS solution over the VAX solution. Only minutes are required for sizeable dynamic analyses, although lengthier analysis times would be required to simulate the behaviour for the full history of an earthquake. In all cases, however, the results of the FPS analysis indicate the advantages of using an attached processor.

The second study of the use of the FPS was a static stress analysis of three-dimensional continua using the boundary element method. The propagation of cracks through the solids are modelled, which requires repeated reformulation and solution of the problem. (Gerstle *et al.*[5] discuss the implementation and results of this analysis on the FPS system; the best VAX/FPS time ratio they report is about 75.)

This boundary element analysis implementation not only uses the library of mathematical functions supplied by the FPS, but also takes advantage of the more powerful FPS fast matrix solution library (FMSLIB) routines.[6] This library accepts element or domain submatrices and

Table I. Comparison of total solution times on the VAX 11/780 and the FPS-264 for nonlinear transient dynamic analysis of 3D frames

Problem	Degrees of freedom	Nodes	Members	Time-steps	VAX elapsed time	FPS elapsed time	VAX/ FPS ratio
1	860	144	287	100	00: 24: 00	00: 05: 50	4:1
2	1080	181	398	100	01: 20: 38	00: 07: 26	11:1
3	2200	369	840	100	02: 35: 09	00: 10: 14	15:1
4	3600	620	1530	50	18: 21: 04	00: 10: 25	105:7

performs all assembly and solution operations, making the best use of the array processor hardware architecture. Both in-core and out-of-core solutions are possible. This library is geared primarily towards the finite element method, and it makes finite element analysis particularly straightforward to implement on this system. However, the library was successfully used to assemble the rectangular submatrices and to solve the unsymmetric coefficient matrices which arise in boundary element analysis. In the next section the use of workstation graphics to evaluate the results of this type of analysis is discussed further.

INTERACTIVE VISUALIZATION USING WORKSTATION GRAPHICS

This section discusses the implementation of a three-dimensional graphical colour postprocessor of engineering numerical analysis in a graphical workstation. There are two main advantages in using a graphical workstation as a driver for an engineering numerical simulation process. The first is to bring all the data that describe the model to the analyst's desk; this reduces to a minimum the amount of data to be transferred to and from the computing node since mainly only response parameters then need to be transferred. The second advantage is that several graphics functions that were performed through software in the past are now provided in hardware by advanced workstations, and thus are faster.

For the purposes of this paper, an engineering workstation is defined as a single-user computer with a 32-bit word length, at least a 1000×1000 colour bit-mapped terminal, a window manager and an operating system that supports multi-tasking, virtual memory and network communication. Its three-dimensional graphics processor consists of a hardware, microcode and software package that is responsible for display transformations, clipping, culling, polygonal filling and polygonal shading computations. It is assumed that performance requirements will dictate that the graphics processor be tightly coupled with the workstation architecture. It is also assumed that the display processor memory will either be directly mapped into the workstation virtual memory or can be quickly accessed by the workstation CPU.

The existing postprocessor is described in Reference 7. Originally, this program was implemented on a VAX 11/780 and used a Lexidata System 3400 (approximate vintage 1982), a 640×512 frame buffer, for the graphical colour display. This raster display was subsequently replaced by a Rastertech Model One/380 (approximate vintage 1984), a high-resolution 1280×1024 frame buffer colour monitor.[8] Both hardware devices provide hardware hidden surface removal by means of a depth buffer.

Neither of these devices, however, completely provides the desired viewing performance for such a highly interactive postprocessor. The bottlenecks are usually caused by the rendering capabilities of the frame buffer and by the bandwidth constraints of sending evaluated display images through a network from the computing node to the graphical display. In addition, shading computations, which are important in obtaining an adequate representation of complex three-dimensional geometries,[7] need to be performed in software and are consequently slow. For example, the low rendering speed of these devices allows for interactive rotation of only a simplified wire-frame outline of the models.[7]

To enhance the interactive characteristics of the postprocessor, it has recently been completely restructured and implemented on an HP 9000 series 320 workstation (approximate vintage 1986). This device improves shaded polygon rendering capability by about two to three orders of magnitude over the older technologies. The display list is stored in virtual memory, and hence the list's size is governed by the virtual memory capacity of the workstation. Segmentation of the display list is a feature that allows specific parts of the image display list to be edited and displayed. Three-dimensional transformations (rotation, translation, scaling and viewing), clipping, polygonal filling, simple diffuse and specular polygonal shading, and hit-testing are done in hardware. The resolution of the monitors is 1280×1024. The frame buffer is a three-channel, real-colour system which can be configured as a 24-bit single buffer or two 12-bit image planes for double buffering. A 16-bit depth buffer is used for hidden surface removal.

Examples of the current implementation of the postprocessor are shown in Plates 1 to 4 taken from the colour display device of the Hewlett-Packard workstation. All the examples are from a boundary element analysis (described in the previous section) of a hypothetical mechanical bracket, a component which presents geometric complexities that are challenging for graphics rendering. The material was specified as steel. The support and loading conditions are such that the upper-right support in Plate 1 has displacements in all directions restrained, the lower-right support is

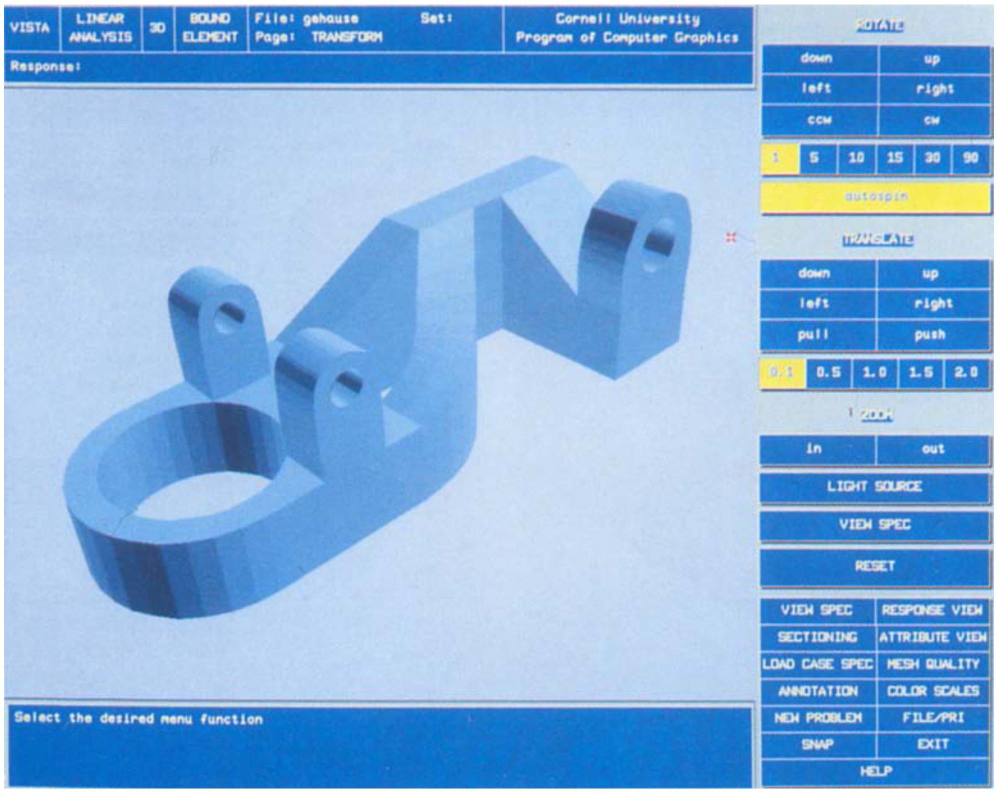

Plate 1. Shaded image of the bracket analysed by the boundary element method

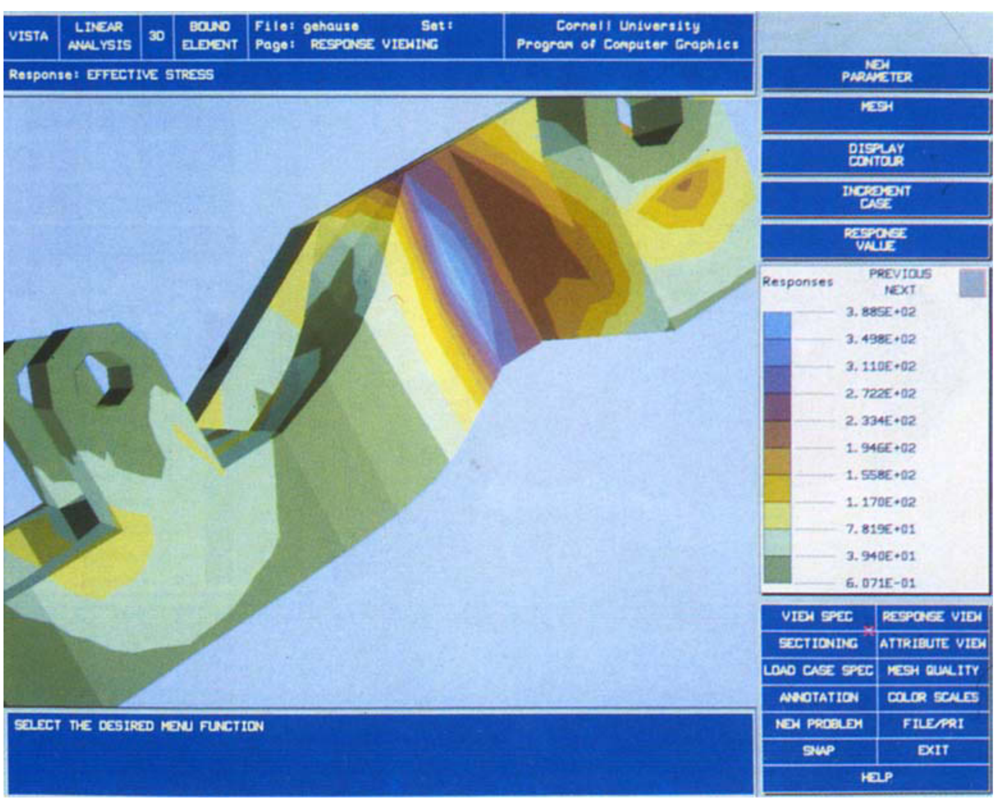

Plate 2. Colour contours of the effective stress on the surface of the bracket

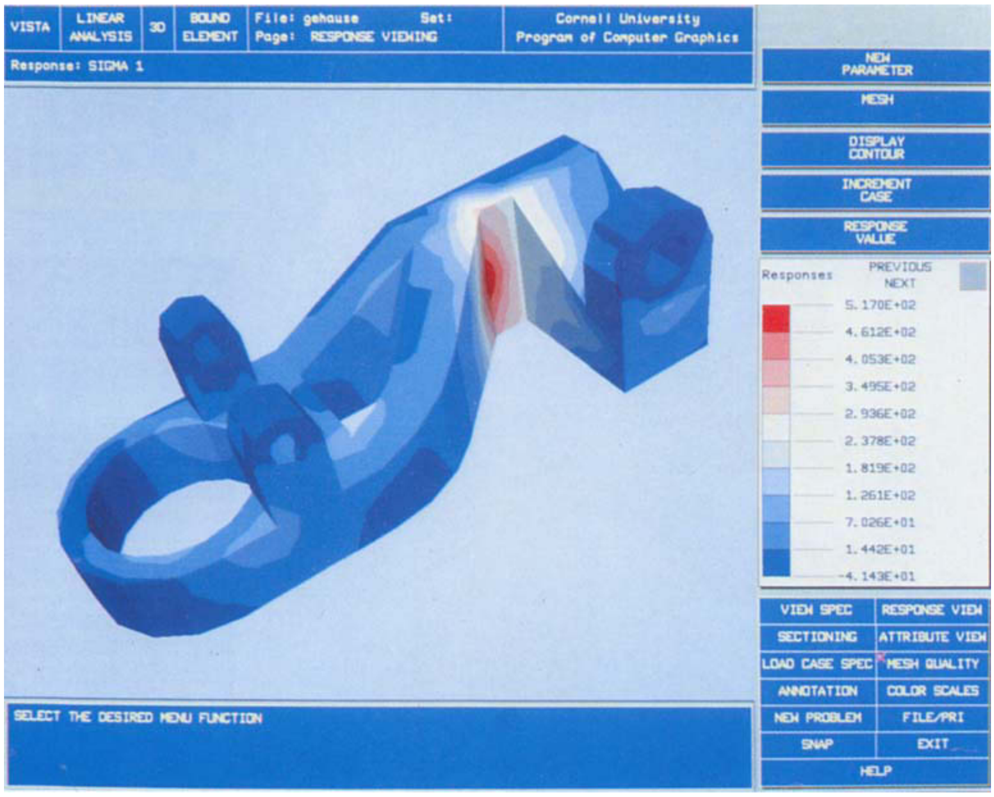

Plate 3. First view of the maximum principal stress on the bracket

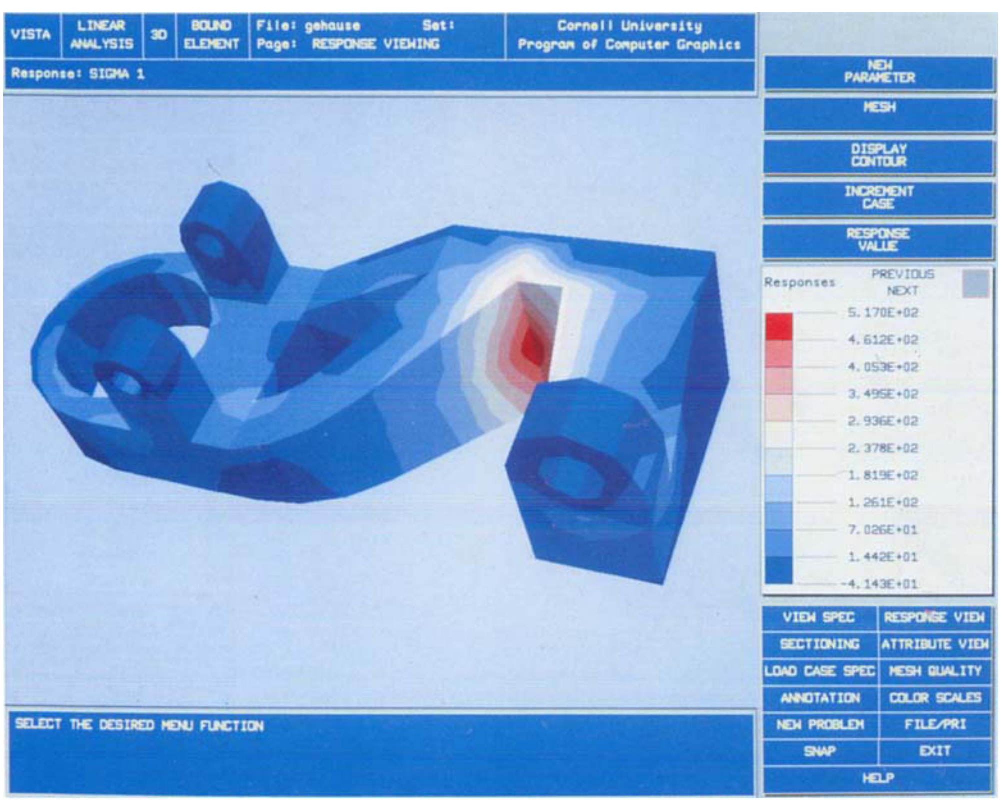

Plate 4. Second view of the maximum principal stress on the bracket

constrained to displace in the direction perpendicular to the bracket, an outward internal pressure acts on the lower-left quadrant of the large circular hole, and a pressure is applied perpendicularly to the bracket in that same region. Plate 1 shows a solid shaded view of the part: Plate 2 shows the colour contouring of effective stresses on the surface of the bracket; the automatic clipping performed by the graphics processor is evident in this view. Plates 3 and 4 show two views of colour contouring of the maximum principal stress.

One of the important features of this advanced workstation implementation is the existence of a hardware 'transform engine' to drive the three-dimensional transformations of the display list. This significantly speeds up the polygonal rendering. Complex polygonal environments can now be rotated, with double buffering to smooth out the process. Complex contouring such as shown in Plates 2, 3 and 4, which each have approximately 2500 polygons, can now be rotated interactively by the user. For example, the dynamic change from the view in Plate 3 to that in Plate 4 is achieved in less than a second. Because shading calculations are done in hardware, it is also possible to allow the user to reorient interactively one or more light sources to illuminate the object as required. Selective or complete changing of base colours (e.g. as in Plate 2), is fast, even if a great number of polygons have to be redrawn.

CONCLUSION

These examples of how computational engineers can take advantage of today's developments in hardware indicate that significant gains in simulation complexity and effectiveness can be achieved. In each case, the utilization of hardware features and/or software provided by the equipment manufacturer has led to gains in performance and capability. This reliance on vendor-supplied, hardware-specific features would be a disadvantage if portability of analysis and postprocessing software were to become a requirement. However, at Cornell's Program of Computer Graphics, as well as at other sites, research and development is continuing with device-independent, high-performance interactive graphics for engineering and scientific applications of enhanced computing. An example is the menu and graphics-manager approach used for the postprocessing illustrated in Plates 1 to 4; this approach is to be described elsewhere.

ACKNOWLEDGEMENTS

Portions of the work described here were sponsored by the National Science Foundation under Grant CEE-8117028; by the National Center for Earthquake Engineering Research, SUNY at Buffalo, under Project Number NCEER 86–4011; by the Cornell Program of Computer Graphics; and by the Cornell University Center for Theory and Simulation in Science and Engineering (which is in turn sponsored by the National Science Foundation (Grant ASC–8500641), IBM Corporation, and the New York State Science and Technology Foundation; and has received equipment grants from Floating Point Systems, Inc.). The second author is supported by a fellowship from CNPq, a Brazilian agency. Equipment grants to the Program of Computer Graphics from Hewlett-Packard Company, Digital Equipment Corporation, Raster Technologies, Inc., and Lexidata Corporation are gratefully acknowledged.

The writers are grateful to their colleagues at the Program of Computer Graphics who contributed to the implementations described. In particular, G. W. Dykes has led the efforts on the menu and graphics-manager approach and on the visualization of engineering analysis results. M. J. Panthaki[8] played a key role in the planning and execution of enhanced postprocessing. Among those who coded portions of the colour postprocessing were B. H. Aubert, T. J. Boone, S. C. Lin, S. Srivastav, D. W. White and P. C. Wong. Faculty members D. P. Greenberg and A. R. Ingraffea also supervised the graphics and engineering research.

Any opinions expressed herein are those of the writers only and do not reflect the views of sponsors, equipment donors or colleagues.

REFERENCES

1. S. I. Hilmy, 'Adaptive nonlinear dynamic analysis of three-dimensional steel framed structures with interactive computer graphics', Ph. D. dissertation, Cornell University, 1984.
2. S. I. Hilmy and J. F. Abel, 'A strain-hardening concentrated plasticity model for nonlinear dynamic analysis of steel buildings', in *NUMETA 85 Numerical Methods in Engineering: Theory and Applications*, Vol. 1, Proceedings of the International Conference on Numerical Methods in Engineering, Swansea, Jan. 1985, (J. Middleton and G. N. Pande, eds) A. A. Balkema, Boston, pp. 305–314 (1985).

3. S. I. Hilmy and J. F. Abel, 'Interactive-adaptive nonlinear dynamic analysis of three-dimensional steel frames', *NUMETA 85 Numerical Methods in Engineering: Theory and Applications*, Vol. 2, (J. Middleton, and G. N. Pande, eds), A. A. Balkema, Boston, pp. 935–944 (1985).
4. *FPS-264 APMATH64 Manual*, Floating Point Systems, Inc., Beaverton, OR, 1985.
5. W. H. Gerstle, L. F. Martha and A. R. Ingraffea, 'Finite and boundary element modeling of crack propagation in two and three dimensions', *Eng. Computers*, **2**, 167–183 (1987).
6. *FMSLIB (Fast Matrix Solution Library) Manual*, Floating Point Systems, Inc., Beaverton, OR, 1986.
7. B. C. Bailey, J. F. Hajjar and J. F. Abel, 'Towards effective interactive three-dimensional colour postprocessing', *Eng. Computations*, **3**, 90–98 (1986).
8. M. J. Panthaki, 'Colour postprocessing for three-dimensional finite element mesh quality evaluation and evolving graphical workstations', M. S. thesis, Cornell University, 1987.