DESIGN ANALYSES OF HELICOPTER AEROELASTIC STABILITY WITH DISTRIBUTED COMPUTING AND GENERALIZED FLOQUET THEORY

Guifa Ma V. Sridharan G.H. Gaonkar

Department of Mechanical Engineering, Florida Atlantic University
Boca Raton, Florida, USA

Abstract

Floquet (fast–Floquet) theory requires excessive run time and a full finite-state representation. Thus advanced modeling, such as free-wake and full nonlinear finite-element modeling, is precluded. Accordingly, an exploratory investigation of distributed computing and the generalized Floquet (fast-Floquet) theory has been conducted toward using advanced modeling in routine stability predictions. For illustration, finite-state modeling with hundreds of states has been used; results from the fast-Floquet theory serve as the exact values, and the model size is quantified by the model order or total number of states. In the generalized Floquet (fast-Floquet) theory, the order of the Floquet transition matrix is successively increased and the convergence characteristics of the eigenvalues or damping levels are studied, and then the converged values are compared with the exact, and also with the approximate from the widely used Sparse Time Domain technique. Furthermore, distributed computing is compared with serial computing on the basis of the run-time growth with the model order. It is also compared with massively parallel computing on the basis of the run-time growth with the model order as well as with the number of processors, and of speedup and efficiency. (Speedup compares the parallel run time with the predicted uniprocessor run time and efficiency guards against processor underutilization). The results demonstrate that the generalized Floquet (fast-Floquet) theory and distributed computing offer considerable promise in routine stability predictions with advanced modeling.

Introduction

Floquet (fast-Floquet) theory requires a finite-state representation of all the states, from structural to aerodynamic to control (Refs. 1, 2). For models with more than 100 or so states, the run time is prohibitive for routine applications (Ref. 2). For example in Ref. 3, the fast-Floquet theory with periodic shooting is used to predict the isolated-rotor stability, and the run time is found to grow between quadratically and cubically with the model order or total number of states; specifically, as the model order varies from 94 to 169, the run-time on a VAX 4320 mainframe computer varies from 6 hours and 45 minutes to 2 days and 11 hours. Despite the variability of these run-time data from computer to computer, such a run-time requirement is a severe barrier to full nonlinear finite-element modeling that typically involves hundreds of structural states alone (Refs. 4, 5). It is good to recall that full nonlinear finite-element modeling approximates the structural details and nonlinearities much better than modal modeling (modal reduction) and that it also provides a better means of matching the sophistication used in approximating the flow field. Moreover in some cases, it is virtually indispensable in that the possible loss of accuracy due to modal modeling may not be acceptable (Refs. 4, 5).

For flow-field approximation, the barrier is the requirement of excessive run time as well as a full finite-state representation. Advanced aerodynamic models, such as free-wake and CFD models, are precluded a fortiori; these models are computationally intensive and do not lend themselves well to a finite-state representation. This should explain the current lack of consistency in modeling sophistication for the trim and stability analyses; generally in comprehensive analyses, the trim analysis is based on advanced modeling and in contrast, the stability analysis is based on relatively simpler finite-state modeling with fewer than 100 or so total states (e.g. Ref. 6). It is also plausible that stability predictions under dynamically stalled conditions may require modeling that goes well beyond current finite-state modeling; typical examples include high-thrust maneuvers at the extremes of the flight envelope (Ref. 7), and high-speed high-thrust flight regimes (Ref. 8).

Accordingly, an exploratory study of distributed computing and the generalized

Floquet (fast-Floquet) theory has been conducted toward removing the current barrier: the requirement of excessive run time and a full finite state representation. The results are based on the fast-Floquet theory, generalized Floquet (fast-Floquet) theory and the Sparse Time Domain or STD technique. Finite-state modeling is used throughout since it provides a simple means of measuring the model size by the total number of states. It also simplifies the study of the results of convergence and accuracy, and the variation of the serial- and parallel-performance metrics with the model size. Despite this limitation of full finite-state modeling, the results demonstrate that the generalized Floquet (fast-Floquet) theory and distributed computing offer considerable promise in removing the current barrier.

Distributed Computing

Distributed computing is parallel computing on a cluster of networked processors or computers, and these individual processors can range from a workstation to a high-performance computer. However, in practice, a distributed computing system represents an assemblage of heterogeneous workstations networked together and provides an effective means of treating computationally intensive cases for routine applications. Since this networking does not interfere with the stand-alone operation of individual workstations, distributed computing provides a means of combining and realizing considerable untapped computing power of the individual workstations during off-load (e.g. after-office) hours. Thus, it can be built as a separate dedicated system and also on existing hardware (Refs. 9, 10). By comparison, the massively parallel computers (e.g. IBM SP-2) are costly, maintained by only a few organizations and are heavily used. The result is high computing cost. The turnaround time is high as well. For a stability-analysis based on Floquet theory, say with about 400 states, the turnaround time often runs into days. Another significant feature is portability. The same code developed on a distributed computing system can be run on a massively parallel computer. Thus with advanced modeling, distributed computing can be used for almost all of the computations. If necessary, only a few final-stage demonstration cases can be run on a massively parallel computer.

Performance Metrics

Three widely used parallel-performance metrics are the growth of run time with the model size, speedup, S_p, and Efficiency, E_P. Let t_p represent the parallel run time with P processors ($P \geq 2$). The speedup and efficiency are defined as follows:

$$S_p = t_1 / t_p \qquad (1)$$

$$E_p = S_p / P \qquad (2)$$

In Eq. (1), t_1 is the predicted uniprocessor run time and comprises a serial part t_{1s}, which is the run time for the serial portion of the code, and a parallel part t_{1p}, which is the run time for the parallel portion. Parallelization with N processors speeds up only the parallel part. Thus,

$$t_1 = t_{1S} + t_{1P} \qquad (3)$$

$$t_N = t_{1S} + \frac{t_{1P}}{N} \qquad (4)$$

The same code is run with several values of N and the corresponding values of t_N are measured. Then t_{1s} and t_{1p} are computed by the least-squares solution and finally t_1 is predicted from Eq. (3). Speedup, S_p, compares the predicted uniprocessor runtime t_1 with the observed parallel run time t_p. Efficiency guards against processor underutilization in speeding up the computations. Ideally $S_p = P$ and $E_p = 1$; that is, the algorithm is perfectly parallel and the 'best' the processors can deliver has been realized (Refs. 9, 10).

Generalized Floquet (fast-Floquet) Theory

The generalized Floquet and generalized fast-Floquet theories are briefly described to help present the results. For details see Refs. 1, 11 and 12.

Generalized Floquet Theory

The generalized Floquet theory combines Floquet theory and the embedding theory, which "is turning out to be one of the major technical contributions of chaos theory (Ref. 13)." This combination provides an analytical basis for using an arbitrary set of excitations and measurements and thereby to approximate the Floquet transition matrix according to the least squares principle (singular value decomposition). Translated in the present context of predicting stability, the embedding theory says that even for large systems, the Floquet exponent can be predicted by observing the response or time history of just one state variable. This is significant since numerical and experimental measurements typically generate a set of discrete data points of a single weakly damped state variable.

For an appreciation of the simplicity of algorithmic aspects, it is expedient to begin with Floquet theory and to consider a system with M total number of states. When each of the M states is perturbed independently to generate the perturbation matrix P, the complete Floquet transition matrix connects this P and the corresponding response matrix R, which is measured after one period T:

$$[R]_{M\times M} = [FTM]_{M\times M}\,[P]_{M\times M} \tag{5}$$

The generalized Floquet theory also generates an equation analogous to Eq. (5), not in the original state space of dimension M, which is not known in several cases, but in a reconstructed or embedded space, say of dimension D. To this end, a series of vectors are reconstructed, and a typical D-dimensional vector is typified by

$$X(t) = \left\lfloor x_i, x_{i+\Delta}, x_{i+2\Delta}, \ldots\ldots, x_{i+(D-1)\Delta} \right\rfloor \tag{6}$$

In the above equation, Δ is the time shift or time lag, which is an integer multiple of the integration step size, and $x_{i+\Delta}, x_{i+2\Delta}$ etc. are the scalar measurements of a selected state variable at time $t_i, t_{i+\Delta,}$ etc. (Refs. 12, 13). Bypassing specific algorithmic details, the basic equation of the generalized Floquet theory can be expressed as (Refs. 12, 13)

$$[R]_{i\times j} = [FTM]_{i\times i}\,[P]_{i\times j} \tag{7}$$

Inclusion of more periods increases the number of columns and inclusion of more time shifts increases the number of rows. The singular value decompositions of P and R, and then the generalized inverse of P give the FTM, and its eigenvalues lead to the damping levels and frequencies.

Generalized fast-Floquet Method

A rotor with Q identical blades has Q planes of symmetry. The generalized Floquet theory plus exploitation of this symmetry is the generalized fast-Floquet theory. The result is that the original period T reduces to T/Q, one blade passage, and the integration interval also reduces to $(T/Q) + \Delta$. Other details are basically similar to those of the generalized Floquet theory.

Modeling and Analyses

The results refer to the isolated-rotor stability, and the rotor has identical blades. The stability analysis is based on the fast-Floquet theory, generalized Floquet (fast-Floquet) theory and Sparse Time Domain or STD Technique. Modal reduction, the ONERA dynamic Stall Models of lift, drag and pitching moment, and a finite-state three-dimensional wake model are used (Refs. 10, 12, 14). Two independent sets of results are generated. The first set refers to serial, massively parallel and distributed computing. Here, periodic shooting based on the fast-Floquet theory is used to predict the trim condition of periodic response and four control settings: shaft tilt, collective- and two cyclic-pitch angles. The Floquet transition matrix comes out as a byproduct and its eigenvalues lead to the damping levels and frequencies. The second set refers to the convergence characteristics and accuracy of the damping levels from the generalized Floquet and generalized fast-Floquet theories. These two theories give virtually identical results and are simply referred to as the generalized Floquet (fast-Floquet) theory. The damping levels from periodic shooting based on the fast-Floquet theory serve as the exact values. The damping levels from the STD technique are also included for comparison. In the second set, serial computing is used. The analytical model approximates the torsionally soft test model of Ref. 14, and the stability analysis is conducted for a given trim condition. Reference 12 gives algorithmic details such as reconstruction of vectors with time delay in an embedded space, the number of periods covered by time marching and the computation of the Floquet transition matrix.

Results

Figure 1 shows the run time versus the model order or total number of states, M. Blade modeling is based on rigid flap and lag motions, and M varies from about 50 to 400. For a job with fixed M, the number of processors is selected automatically by the compiler system. The variation of the serial run time for $94 \le M \le 169$ is taken from Ref. 3. The serial run time varies approximately as $M^{2.4}$. It is about 2½ days for M = 169. For models with hundreds of states it could run into weeks even with the fast-Floquet theory. Simply put, such a run time is prohibitive for routine stability predictions. In comparison, the two parallel run times are shorter and their rates of growth with respect to *M* are much slower. Although the differences in the architecture do not permit a quantitative comparison among the serial and two parallel run times, these data on the run time and its growth with *M* show that the two parallel computing systems provide a means of treating systems with hundreds of states.

In Figs. 2-4, distributed computing is compared with massively parallel computing on the basis of how run time, speedup and efficiency vary with the number of processors, *P*, and the model order, *M*. Blade modeling is based on elastic flap-lap-torsion motions. Three isolated-rotor models with *M* = 167, 302 and 410 are considered, and *P* varies from 2 to 32 for each model. A job with fixed *M* is executed for a stipulated value of *P* through a program directive and then this execution is repeated by successively increasing *P.*

Figure 2 shows the run times from the two parallel computing systems as a function of *P* and *M*. In both, the run time decreases for increasing *P* and fixed *M,* and increases for increasing *M* and fixed *P*. For example, the distributed computing system with *P* = 2 takes nearly 135 hours for *M* = 410. This run time comes down to about 24 hours as *P* increases to 13. Overall, both parallel computing systems are strikingly similar as to how the run time grows with *M* and how it can be controlled by increasing *P* with *M.* The only difference is that for a given *P* and *M,* the run time of the massively parallel computing system is much shorter since the constituent IBM SP-2 processors are much faster than the SunSPARC workstations.

Speedup S_P, and efficiency, E_P, are presented in Figs. 3 and 4, respectively. Overall, the results of S_P and E_P from both parallel computing systems are comparable. For example, with a fixed number of processors *P*, as the model order *M* increases the effectiveness of parallelization increases; that is, the parallel run time generally decreases, and in turn, speedup S_P increases. Moreover, the processors are getting closer to delivering the 'best'. Thus the idle time of the individual processors decreases and thereby efficiency also increases. Both massively parallel and distributed computing systems well demonstrate this simultaneous increase in S_P and E_P with fixed *P* and increasing *M*. As an illustration, the distributed computing system is considered. For *P* = 5, as *M* increases from 167 to 410, S_P increases from 3 to 4.5 and the corresponding E_P also increases from 60% to 92%. In fact, for *M* = 410,and $P \leq 5$, both S_P and E_P are close to the ideal: $S_P = P$ and E_P = 100%. These results have considerable bearing on the practical utility of distributed computing as an effective alternative to massively parallel computing.

Tables 1 and 2 show the convergence and accuracy of the damping from the stability analysis for a known trim condition; the analytical model has 228 total states, structural and aerodynamic. Five cases are presented with advance ratio μ = 0, 0.05, 0.1, 0.2, and 0.31; the case with μ = 0 also leads to a periodic coefficient system. In each case, the generalized Floquet and generalized fast-Floquet theories are applied with a successive increase in the order of the FTM: 80,120,160, and 200. Table 1 also shows that the converged damping can be predicted with about three trials. Similarly, Table 2 shows that the converged values agree with the exact and the STD-technique results within 3% error.

Concluding Remarks

1. The generalized Floquet (fast-Floquet) theory gives the converged damping values with about three trials with a successive increase in the order of the Floquet transition matrix. These converged values agree extremely well with the exact values and also with the values from the widely used Spares Time Domain (STD) technique.
2. In both massively parallel and distributed computing, the run time and its growth with the model order can be controlled by increasing the number of processors with the order. This increase can be determined on the basis of how fast is fast enough and how effectively the processors are utilized; that is, a compromise between run time and speedup on the one hand and efficiency on the other
3. The performance metrics of speedup and efficiency of distributed computing are comparable to those of massively parallel computing. Moreover, a distributed computing system has significant advantages over a massively parallel computer with respect to cost effectiveness and turnaround time, and these advantages

far outweigh the fact that the processing speed is faster in the latter.

4. To sum up: The generalized Floquet (fast-Floquet) theory and distributed computing offer considerable promise in routine stability predictions with state-of-the-art modeling of structural and aerodynamic components. This remark should be tempered by the fact that the results are based on full finite-state modeling.

References

1. Gaonkar, G.H., "Serial and Parallel Computing of Helicopter Stability from Floquet and Approximate Methods," Paper presented at the 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, April 3-6, 2000, Atlanta, Georgia, AIAA Paper No. 2000-1692.

2. Subramanian, S., and Gaonkar, G.H., Nagabhushanam, J., and Nakadi, R.N., "Parallel Computing Concepts and Methods for Floquet analysis of Helicopter Trim and Stability," Journal of the American Helicopter Society, Vol. 41, (4), October 1996, pp. 370-382.

3. Subramanian, S., and Gaonkar, G.H, "Parallel Fast-Floquet Analysis of Trim and Stability for Large Helicopter Models," Mathematical and Computer Modeling, Vol. 33, May-June 2001, pp. 1155-1176.

4. Bauchau, O.A., and Guernsey, D., "On the Choice of Appropriate Bases for Nonlinear Dynamic Model Analysis," Journal of the American Helicopter Society, Vol. 38, (4), October 1993, pp. 28-36.

5. Bauchau, O.A., Bottasso, C.L., and Nikishkov, Y.G., "Modeling Rotorcraft Dynamics with Finite Element Multibody Procedures," Mathematical and Computer Modeling, Vol.33, May-June 2001, pp. 1113-1137.

6. Wang, J.M., "Challenges in Rotor Dynamics Correlation," Proceedings of the 54th Annual Forum of the American Helicopter Society, Washington, D.C. may 20-22, 1998.

7. Bagai, A., Leishman, J.G., and Park, J., "Aerodynamic Analysis of a Helicopter in Steady Maneuvering Flight Using a Free-Vortex Rotor Wake Model," Journal of the American Helicopter Society, Vol. 44 (2), April 1999, pp. 109-120

8. McHugh, F.J., "What Are the Lift and Propulsive Force Limits at High Speed for the Conventional Rotor?" Presented at the 34th Annual National Forum of the American Helicopter Society, Washington, D.C., May 1978, Preprint No. 78-2.

9. Venketaratnam, S., Subramanian, S., and Gaonkar, G.H., "Fast-Floquet Analysis of Trim and Stability with Distributed and Massively parallel Computing," Proceedings of the 23rd European Rotorcraft Forum, Vol. 2, Dresden, Germany, September 16-18,1997, Paper No. 92.

10. Subramanian, S., Venketaratnam, S., and Gaonkar, G.H., "Serial and Parallel Fast-Floquet Analyses in Rotorcraft Aeroelasticity," Proceedings of the Fourth International Symposium in Fluid-Structure Interactions, Aeroelasticity, Flow-Induced Vibration and Noise, ASME International mechanical Engineering Congress and Exposition, Dallas, Texas, November 16-21, 1997.

11. Wang Xin, and Peters D.A., "The Method of Generalized Floquet Theory Applied to Flap-Lag Dynamics with Inflow," Paper presented at the 24th European Rotorcraft Forum, Marseilles, France, September 15-17, 1998, paper No. DY05

12. Ma Guifa, *Fast-Floquet method of trim and Stability including Comparisons with Experiments and Approximate Methods*, Ph.D dissertation, Mechanical Engineering Department, Florida Atlantic University, July 2000.

13. Hilborn, Robert C., *Chaos and Nonlinear Dynamics*, Second Edition, Oxford University Press, 2000, Chapter 9.

14. Subramanian, S., Ma, Guifa., Gaonkar, G.H., and Maier, T.H., "Correlation of Several Aerodynamic Models and Measurements of Hingeless Rotor Trim and Stability," Journal of the American Helicopter Society, Vol.45 (2), April 2000, pp.106-117.

Table 1 - Convergence of Lag Regressive-Mode Damping from the Generalized Floquet (fast-Floquet) theory.

Advance Ratio μ		0	0.05	0.1	0.20	0.31
Theory	FTM Order	Damping	Damping	Damping	Damping	Damping
Generalized Floquet (fast-Floquet) Theory	80 120 160 200	0.808 0.788 0.785 0.781	0.862 0.840 0.836 0.832	1.218 1.141 1.003 0.971	1.441 1.341 1.202 1.196	1.442 1.347 1.246 1.241

Table 2 - Lag Regressive-Mode Damping from the Generalized Floquet (fast-Floquet) theory and the STD Technique, and Comparison with the Fast-Floquet Theory.

Advance Ratio μ	0	0.05	0.1	0.20	0.31
Fast-Floquet Theory (228 states)	0.774	0.824	0.966	1.214	1.213
Converged value from the Generalized Floquet (fast-Floquet) Theory	0.781	0.832	0.971	1.196	1.241
STD Technique	0.794	0.838	0.975	1.216	1.218

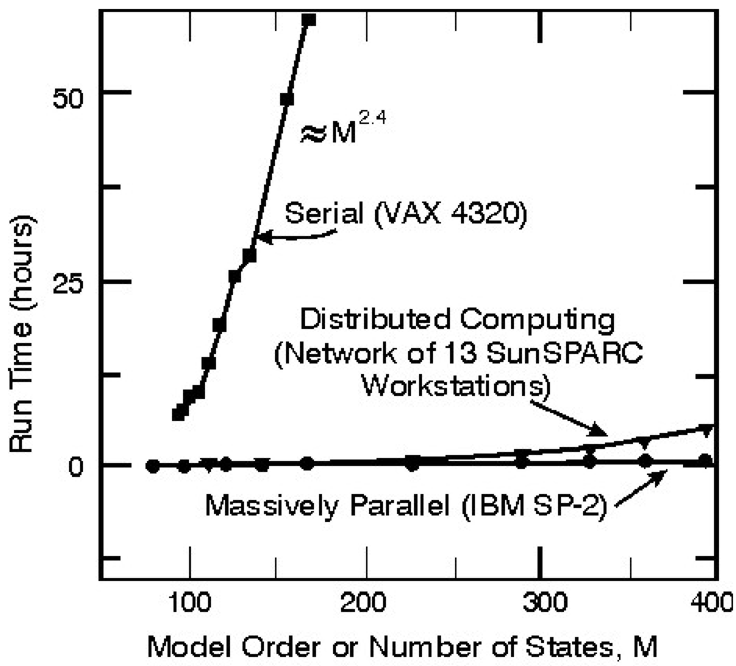

Figure 1 - Run-Time Variations on Serial, Massively Parallel and Distributed Computing Systems for the Trim and Stability Analyses with Periodic Shooting based on the Fast-Floquet Theory.

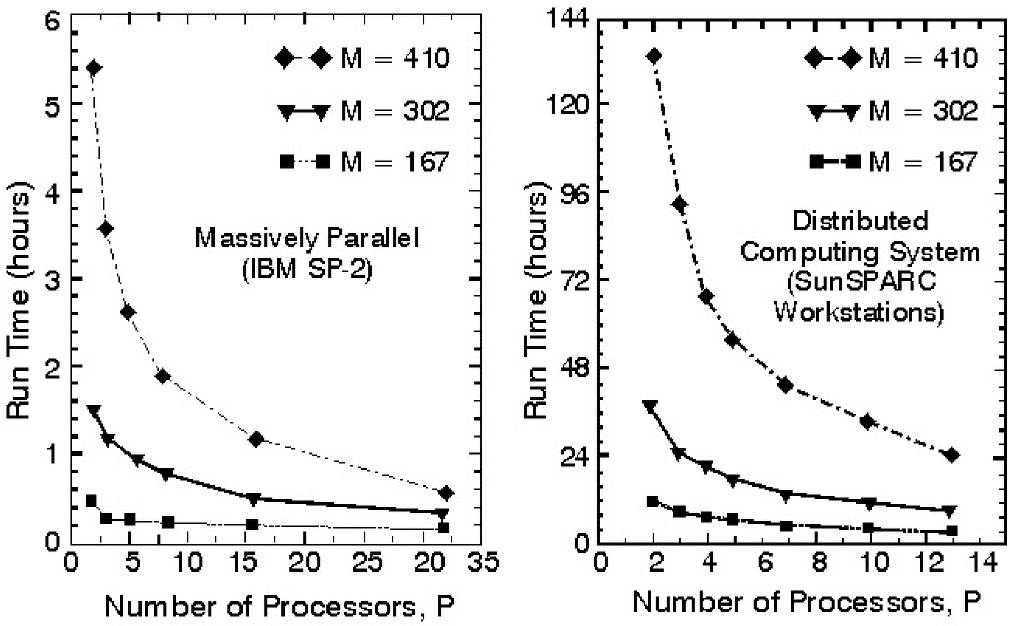

Figure 2 –Run-Time Variations with the Model Order M and Number of Processors P for the Trim and Stability Analyses with Periodic Shooting based on the Fast-Floquet Theory,

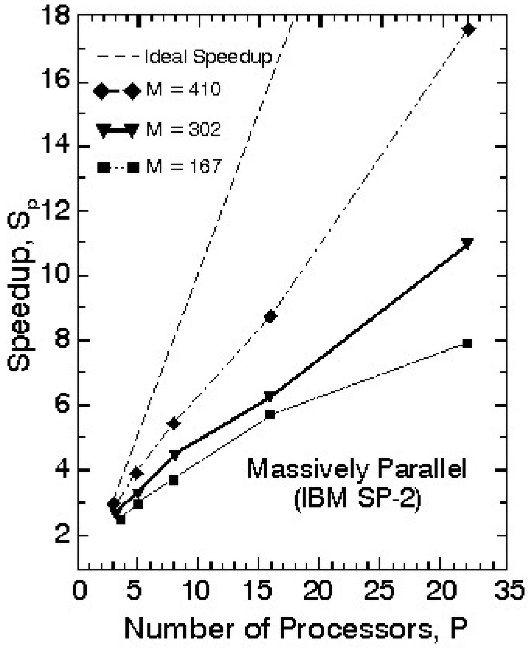

Figure 3 –Speedup Variations with the Model Order M and Number Processors P for the Trim and Stability Analyses with Periodic Shooting based on the Fast – Floquet Theory.

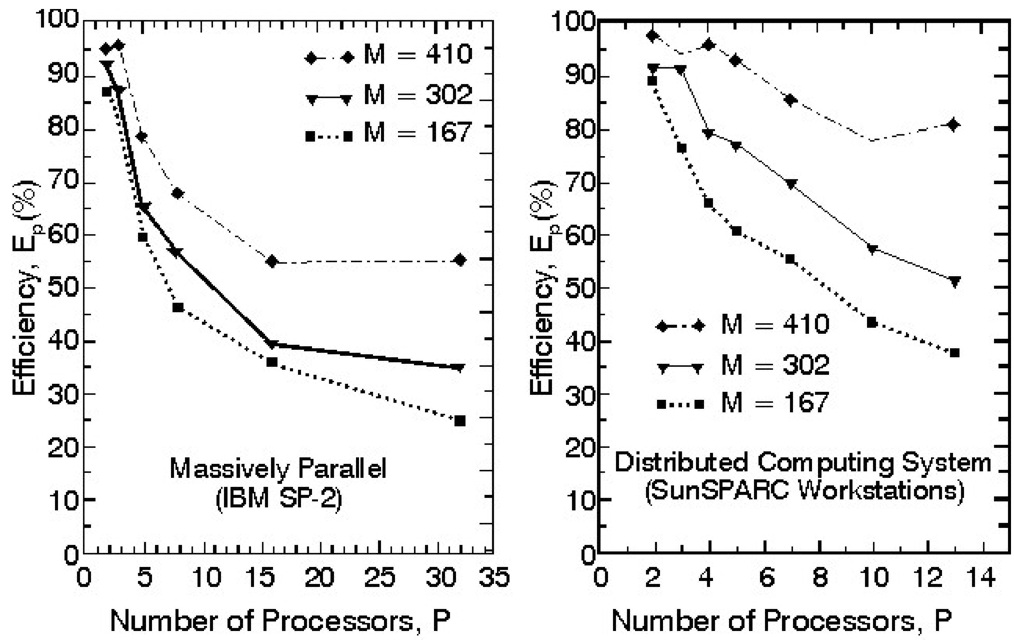

Figure 4 – Efficiency Variations with the Model Order M and Number of Processors P for the Trim and Stability Analyses with Periodic Shooting based on the Fast-Floquet Theory