Some Parallel Numerical Methods in Solving Partial Differential Equations

Norma Alias
Institute of Ibnu Sina,
Universiti Teknologi Malaysia
81310 Skudai, Johor Bahru, Malaysia
norma@ibnusina.utm.my

Noriza Satam, Roziha Darwis, Norhafiza Hamzah,
Zarith Safiza Abd. Ghaffar, Md. Rajibul Islam
Department Of Mathematics
Universiti Teknologi Malaysia,
81310 Skudai, Johor Bahru, Malaysia
{zarithsafiza.ag, norizasatam, roziha.darwis,
norhafizahamzah, md.rajibul.islam}@gmail.com

***Abstract*—This paper will discuss the solution of two-dimensional partial differential equations (PDEs) using some parallel numerical methods namely Gauss Seidel and Red Black Gauss Seidel. The selected two-dimensional PDE to solve in this paper are of parabolic and elliptic type. Parallel Virtual Machine (PVM) is used in support of the communication among all microprocessors of Parallel Computing System. PVM is well known as a software system that enables a collection of heterogeneous computers to be used as coherent and flexible concurrent computational resource. The numerical results will be presented graphically and parallel performance measurement by Gauss Seidel and Red Gauss Seidel methods will be evaluated in terms of execution time, speedup, efficiency, effectiveness and temporal performance. Performance evaluations are critical as this paper aimed to fabricate an efficient Two-Dimensional PDE Solver (TDPDES). This new well-organized TDPDES technique will enhance the research and analysis procedure of many engineering and mathematic fields.**

Keywords-Two-dimensional Partial Differential Equations; Parabolic, Elliptic; Hyperbolic; Parallel Numerical Methods; Performance Evaluations

I. INTRODUCTION

It is abundantly clear that many important scientific problems are governed by partial differential equations according to [5-6]. The difficulty in obtaining exact solution arises from the governing partial differential equations and the complexities of the geometrical configuration of physical problems [7, 8, 9]. For example, imagine a metal rod insulated along its length with no heat can escape for its surface. If the temperature along the rod is not constant, then heat conduction takes place. In such situations, the numerical method is used to obtain the numerical solutions [10]. These partial differential equations may have boundary value problems as well as initial value problems. This study will discuss the two-dimensional partial differential equation solved using parallel Gauss-Seidel and Red Black Gauss-Seidel Methods. First, the PDEs will be written in matrix form to ease the work. Then, parallel algorithm for all three types of the PDEs will be developed and run in parallel computing environment to provide the numerical solution. Finally, the speed of convergences of using the above numerical methods will be compared. In general, the transient particle diffusion or heat conduction is Partial Differential Equations (PDE) of the parabolic type and Laplace's equation for temperature, diffusion, electrostatic conduction is elliptic and wave equation or transport equation is the PDE of hyperbolic type [5, 9, 6]. The parabolic partial differential equations are normally used in such fields like molecular diffusion, heat transfer, nuclear reactor analysis, and fluid flow [11, 12].

Partial differential equations (PDEs) widely used as mathematical models for phenomena in all branches of engineering and science.

A. *Parabolic Equation*

$$\frac{\partial u}{\partial t}=a_1(x,y,t)\frac{\partial^2 u}{\partial x^2}+a_2(x,y,t)\frac{\partial^2 u}{\partial y^2}+b_1(x,y,t)\frac{\partial u}{\partial x}+b_2(x,y,t)\frac{\partial u}{\partial y}-c(x,y,t) \tag{1}$$

where a < 0, c ≥ 0 and $b^2-4ac=0$. The PDE is said to be parabolic if $\det(Z)=0$. The heat conduction equation and other diffusion equation are examples. The heat equation is $\frac{\partial U}{\partial T}=\kappa\frac{\partial^2 U}{\partial X^2}$, κ is a constant. Initial-boundary conditions are used to give

u (x, t) = g(x, t) for x∈ ∂ Ω, t >0
u (x,0) =(x) for x∈ Ω,
where ux x = f (ux ,uy ,u, x, y) holds in Ω.

B. *Hyperbolic Equation*

$$\frac{\partial^2 u}{\partial t^2}-(a\frac{\partial^2 u}{\partial x^2}+2b\frac{\partial^2 u}{\partial x\partial y}+c\frac{\partial^2 u}{\partial y^2})+d\frac{\partial u}{\partial t}+e\frac{\partial u}{\partial x}+f\frac{\partial u}{\partial y}+gu=0 \tag{2}$$

where $b^2-4ac>0$. The PDE is said to be hyperbolic if $\det(Z)<0$. The wave equation is an example of a hyperbolic partial differential equation. The wave equation is $\frac{\partial^2 u}{\partial x^2}-\frac{1}{\beta}\frac{\partial^2 u}{\partial t^2}=0$, β is a constant. Initial-boundary conditions are used to give

u (x, y, t) = g(x, y, t) for x∈ ∂ Ω, t >0
u (x, y, 0) = v0 (x, y) in Ω
ut (x, y, 0) = v1 (x, y) in Ω
where ux y = f (ux ,ut , x, y) holds in Ω.

C. Elliptic Equation

$$a(x,y)\frac{\partial^2 u}{\partial x^2}+2b(x,y)\frac{\partial^2 u}{\partial x\partial y}+c(x,y)\frac{\partial^2 u}{\partial y^2}=d(x,y,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}) \quad (3)$$

Where $b^2-4ac>0$. The PDE is said to be elliptic if Z is a positive definite matrix with $\det(Z)<0$. Laplace's equation and Poisson's equation are examples. The Laplace's equation is $\frac{\partial^2 u}{\partial x^2}-\frac{\partial^2 u}{\partial y^2}=0$. Boundary conditions are used to give the constraint *u(x, y)* on $\partial\,\Omega$,
where ux x + ux y = f (ux ,uy ,u, x, y)

D. Finite Difference Method

Finite Difference Method is a classical and straightforward way to solve the partial difference equation [3, 4] numerically. It consists of transforming the partial derivatives in difference equations over a small interval and the continuous domain of the state variables by a network or mesh of discrete points. The partial differential equation is converted into a set of finite difference equations so that it can be solved subject to the appropriated boundary conditions.

Assuming that *u* is function of the independent variables x and y, then divided the x-y plan in mesh points equal to δx = *h* and δy = *k*,
Evaluate *u* at point *P* by:

$$u_p=u(ih,jk)=u_{i,j}$$

The value of the second derivative at *P* could also be evaluated by:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_p=\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j}\cong\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_p=\left(\frac{\partial^2 u}{\partial y^2}\right)_{i,j}\cong\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{k^2}$$

II. Two-Dimensional PDE Solver (TDPDES)

A. Hyperbolic Partial Differential Equations

Hyperbolic differential equations, includes the "wave equation" which is fundamental to the study of vibrating systems. It is instructive to outline the derivation of the simple wave equation in one dimension problem.

The wave equation is given by the differential equation

$$\frac{\partial^2 u}{\partial t^2}(x,t)-\alpha^2\frac{\partial^2 u}{\partial x^2}(x,t)=0, 0<x<L, t>0 \quad (4)$$

Subject to the boundary conditions

$$u(0,t)=u(L,t)=0, t>0$$

and the initial conditions

$$u(x,0)=f(x), 0\le x\le L,$$

$$\frac{\partial u}{\partial t}(x,0)=g(x), 0\le x\le L$$

where α is a constant.

To set up the finite-difference method, assume $u=f(x)$ is a function of the independent variables x and t. Subdivide the x-plane into sets of equal rectangles if sides $\delta x=h$ and $\delta t=k$. We introduce a time grid $t_n = n\Delta t$ for n = 0, 1, 2,.. and Δt is the time step size. We set $p^n(x) = p(x, t_n)$ as the nth iterate of the pressure at the global point x. The time derivatives in (4) are discretized by centered second-order finite difference, which gives the semi-discrete scheme:

$$\frac{P^{n+1}-2P^n+P^{n-1}}{\Delta t^2}+\gamma c^2\frac{P^{n+1}-P^{n-1}}{2\Delta t}=c^2\Delta^2 P^n \quad (14)$$

B. Two Dimensional Parabolic Equations

A forward finite difference is used to approximate the time derivative. Consider the two-dimensional of parabolic equations

$$\frac{\partial u}{\partial t}=c\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right),\ c \text{ constant} \quad (15)$$

Applying the Crank Nicolson scheme to the two-dimensional heat equation results in

$$\frac{u^{(n+1)}-u^{(n)}}{\Delta t}=\frac{c}{2}\left[\frac{\partial^2 u^{(n+1)}}{\partial x^2}+\frac{\partial^2 u^{(n+1)}}{\partial y^2}+\frac{\partial^2 u^{(n)}}{\partial x^2}+\frac{\partial^2 u^{(n)}}{\partial y^2}\right] \quad (16)$$

This leads to the following finite difference equation

$$\begin{aligned}&N_{ij}(t+\Delta t)=\\&N_{ij}(t)+\Delta t[P^j_{i-1,i}N_{i-1,j}(t)+R^{j-1,j}_i N_{i,j-1}(t)-P^j_{i,i+1}N_{ij}(t)\\&-R^{j,j+1}_i N_{ij}(t)+Q^j_{i-1,i}N_{i-1,j}(t)+Q^j_{i+1,i}N_{i+1,j}(t)\\&+Q^{j-1,j}_i N_{i,j-1}(t)+Q^{j+1,j}_i N_{i,j+1}(t)\\&-(Q^j_{i,i-1}+Q^j_{i,i+1}+Q^{j,j-1}_i+Q^{j,j+1}_i)N_{ij}(t)+\Gamma_{i,j}-L_{ij}N_{ij}(t)],\end{aligned} \quad (17)$$

where $\Gamma_{i,j}$ and $L_{i,j}$ are the generation and death rates, respectively. Under suitable regularity assumptions one can expand N, P, Q and R, use $N_{i,j}(t)\approx u(t,x_{i,j})\Delta V_{i,j}$ and write the word equation above mathematically (Angelis and Preziosi, 2000) as:

$$\frac{\partial u}{\partial t}=-\frac{\partial(Pu)}{\partial x}-\frac{\partial(Ru)}{\partial y}+\frac{\partial}{\partial x}(Q\frac{\partial u}{\partial x})+\frac{\partial}{\partial y}(Q\frac{\partial u}{\partial y})+\Gamma-Lu, \quad (18)$$

with $\Gamma(t,x_i,y_j)=\Gamma_{i,j}(t)/\Delta V_{i,j}$ and where the indices (i, j) have been substituted with the dependence of u and of all coefficients on the space variable. Equivalently, one can
write

$$\frac{\partial u}{\partial t} + \nabla.(\mathrm{Wu}) = \nabla.(Q\nabla u) + \Gamma - Lu, \qquad (19)$$

where, in two dimensions, W = (P, R). The general advection-diffusion model (19) requires the specification of the drift, diffusion, proliferation, and death coefficient in the terms W, Q, Γ, L and in particular of their dependence of the state variables. Based on central finite difference method, the discretization is shown as follow:

Let $\frac{\partial u}{\partial t} = \frac{N_{ij}(t+\Delta t) - N_{ij}(t)}{\Delta t}$ and applying the discretization to the right side, the equation (18) becomes,

$$\begin{aligned}
&\frac{N_{ij}(t+\Delta t) - N_{ij}(t)}{\Delta t} \\
&= [P^j_{i-1,i} N_{i-1,j}(t) - P^j_{i,i+1} N_{ij}(t)] + [R^{j-1,j}_i N_{i,j-1}(t) - R^{j,j+1}_i N_{ij}(t)] \\
&+ [Q^j_{i+1,i} N_{i+1,j}(t) - Q^j_{i,i-1} N_{ij}(t)] + [Q^j_{i-1,i} N_{i-1,j}(t) - Q^j_{i,i+1} N_{ij}(t)] \\
&+ [Q^{j+1,j}_i N_{i,j+1}(t) - Q^{j,j-1}_i N_{ij}(t)] + [Q^{j-1,j}_i N_{i,j-1}(t) - Q^{j,j+1}_i N_{ij}(t)] \\
&+ \Gamma_{i,j} - L_{ij} N_{ij}(t) \\
&= [P^j_{i-1,i} N_{i-1,j}(t) - P^j_{i,i+1} N_{ij}(t)] + [R^{j-1,j}_i N_{i,j-1}(t) - R^{j,j+1}_i N_{ij}(t)] \\
&+ [Q^j_{i-1,i} N_{i-1,j}(t) - (Q^j_{i,i-1} + Q^j_{i,i+1}) N_{ij}(t) + Q^j_{i+1,i} N_{i+1,j}(t)] \\
&+ [Q^{j-1,j}_i N_{i,j-1}(t) - (Q^{j,j-1}_i + Q^{j,j+1}_i) N_{ij}(t) + Q^{j+1,j}_i N_{i,j+1}(t)] \\
&+ \Gamma_{i,j} - L_{ij} N_{ij}(t)
\end{aligned} \qquad (27)$$

C. Two Dimensional Elliptic Equation

The two dimensional elliptic equation $\frac{\partial^2 r}{\partial x^2} + \frac{\partial^2 r}{\partial y^2} = 0$ can be further implemented to solve the large scale mathematical problem. Generally, finite-difference approximation to two dimensional elliptic equation is given by

$$\frac{r_{i-1,j} - 2r_{i,j} + r_{i+1,j}}{(\Delta x)^2} + \frac{r_{i,j-1} - 2r_{i,j} + r_{i,j+1}}{(\Delta y)^2} = 0 \qquad (28)$$

Where $\Delta x = h,\ \Delta y = k$

$$\frac{r_{i-1,j} - 2r_{i,j} + r_{i+1,j}}{h^2} + \frac{r_{i,j-1} - 2r_{i,j} + r_{i,j+1}}{k^2} = 0 \qquad (29)$$

By multiplying each side with h^2, we have

$$r_{i-1,j} - 2r_{i,j} + r_{i+1,j} + \frac{h^2}{k^2}\left(r_{i,j-1} - 2r_{i,j} + r_{i,j+1}\right) = 0 \qquad (30)$$

If we assume $\theta = \frac{h^2}{k^2}$, then we will have the finite-difference approximation equation as follows

$$\theta r_{i,j-1} + r_{i-1,j} - (2+2\theta) r_{i,j} + r_{i+1,j} + \theta r_{i,j+1} = 0 \qquad (31)$$

For $0 \le \theta \le 1$

The discretization of the mathematical model based on the finite-difference approximation to equation (28) can be written as,

$$\left(\frac{\partial^2 r}{\partial x^2} + \frac{\partial^2 r}{\partial y^2} + \bar{k}^2(r)\right) e(r) = 0 \qquad (38)$$

After applying the finite-difference approximation to equation (38) is given by

$$\left(\frac{r_{i+1,j} - 2r_{i,j} + r_{i-1,j}}{(\Delta x)^2} + \frac{r_{i,j+1} - 2r_{i,j} + r_{i,j-1}}{(\Delta y)^2} + \bar{k}^2(r_{i,j})\right) e(r_{i,j}) = 0 \qquad (39)$$

From equation (39), it becomes

$$\left(\frac{r_{i+1,j} - 2r_{i,j} + r_{i-1,j}}{h^2} + \frac{r_{i,j+1} - 2r_{i,j} + r_{i,j-1}}{k^2} + \bar{k}^2(r_{i,j})\right) e(r_{i,j}) = 0 \qquad (40)$$

Where $\Delta x = h, \Delta y = k$. If we bring the $e(r_{i,j})$ term to the right-hand side, it become $e(r_{i,j}) = 0$. Thus,

$$\frac{r_{i+1,j} - 2r_{i,j} + r_{i-1,j}}{h^2} + \frac{r_{i,j+1} - 2r_{i,j} + r_{i,j-1}}{k^2} + \bar{k}^2(r_{i,j}) = 0 \qquad (41)$$

By multiplying each side with h^2, equation (41) become

$$r_{i+1,j} - 2r_{i,j} + r_{i-1,j} + \left(\frac{h^2}{k^2}\right)[r_{i,j+1} - 2r_{i,j} + r_{i,j-1}] + h^2\bar{k}^2 r_{i,j} = 0 \qquad (42)$$

The exact solution to the discretized problem obeys the equation

$$r_{i+1,j} + r_{i-1,j} - \left[2 + 2\left(\frac{h^2}{k^2}\right) - h^2\bar{k}^2\right] r_{i,j} + \left(\frac{h^2}{k^2}\right) r_{i,j+1} + \left(\frac{h^2}{k^2}\right) r_{i,j-1} = 0 \qquad (43)$$

$$\left[2 + 2\left(\frac{h^2}{k^2}\right) - h^2\bar{k}^2\right] r_{i,j} = r_{i+1,j} + r_{i-1,j} + \left(\frac{h^2}{k^2}\right) r_{i,j+1} + \left(\frac{h^2}{k^2}\right) r_{i,j-1} \qquad (44)$$

Thus,

$$r_{i,j} = \frac{r_{i+1,j} + r_{i-1,j} + \left(\frac{h^2}{k^2}\right) r_{i,j+1} + \left(\frac{h^2}{k^2}\right) r_{i,j-1}}{\left[2 + 2\left(\frac{h^2}{k^2}\right) - h^2\bar{k}^2\right]} \qquad (45)$$

This equation cannot be solved explicit for fixed $r_{i,j}$ because there are five unknowns involved. Thus, if the n th iterate is denoted $r_{i,j}^{(n)}$.

III. PARALLEL COMPUTING

The classification of the parallel computer architecture can be divided into three categories: Flynn's taxonomy, Quinn classification and Cheong classification. The PVM system supports the message passing, shared memory, and hybrid paradigms, thus allowing applications to use the most appropriate computing model, for the entire application or for individual sub-algorithms. Processing elements such as scalar machines, distributed-and shared-memory multiprocessors, vector supercomputers and special purpose graphics engines, permitted the use of the best suited computing resource for each component of an application. This versatility is valuable for several large and complex applications including global environmental modeling, fluid dynamics simulations, and weather prediction applications.

PVM system is implemented on a hardware base which is consists of different machine architectures, including single CPU systems, vector machines, and multiprocessors. This computing element is interconnected by one or more networks, which may themselves be different like one implementation of PVM operates on Ethernet, Internet and a fiber optic network [9].

C, C++ and FORTRAN are all languages that can be used to write PVM codes. This project is done by using C languages by UNIX as an operating system. To execute an application, a user typically starts one copy on one task from a machine within the host pool. Task-to-task communication in PVM is done with message passing. Message passing is a set of tasks that use their own local memory during computation. Multiple tasks can reside on the same physical machine as well as across an arbitrary number of machines. Tasks exchange data through communications by sending and receiving messages. Data transfer usually requires cooperative operations to be performed by each process. For example, a send operation must have a matching receive operation.

IV. PARALLEL PERFORMANCE EVALUATION

The performance of the parallel algorithm will be analyzed in terms of the time execution, speedup, efficiency, effectiveness and temporal performance. The measurements are defined as follows:

Speedup: $S(p) = \frac{t_1}{t_p}$

Efficiency: $E(p) = \frac{S(p)}{p}$

Effectiveness: $F(p) = \frac{S(p)}{pt_p} = \frac{E_p}{t_p}$

Temporal performance: $L(p) = t_p^{-1} = \frac{1}{t_p}$

t_1 = execution time for a single processor and

t_p = execution time using p parallel processors.

Figure 1(a) shows that the execution time is decreasing with the increasing of the number of processors. The reduction of execution time as number of processors increase can also be seen in solving parabolic and hyperbolic problem. Figure 1(b) shows that the speedup increases when the number of processors is added. It is because the distributed memory hierarchy reduces the time consuming access to a cluster of workstations. The efficiency of a parallel program is a measure of processor utilization. Figure 1(c) shows that the efficiency decreases with the increasing of number of processors. As known, efficiency is the ratio of speedup with number of processors. So, efficiency is a performance closely related to speedup. The effectiveness is escalating with the increasing of the number of processors. The formula of the effectiveness is depending on the speedup, when the speedup increases, the effectiveness will also increase.

Figure 1(e) shows that the temporal performance graph is proportional to the number of processors increase. This is because the execution time is decreasing versus the number of processors. It can be conclude that, from the aspect of execution time, speedup, efficiency, effectiveness and temporal performance shows the performance of parallel algorithm is improved by the increasing of the number of processors. Communication and execution times is always affecting the performance of parallel computing. The Red Black Gauss Seidel which is effective is found to be well suited for parallel implementation on PVM where data decomposition is run synchronously and concurrently at every time level. The PVM system has been used for applications such as molecular dynamics simulations, superconductivity studies, distributed fractal computations, matrix algorithms, and in the classroom as the basis for teaching concurrent computing.

V. CONCLUSION

Numerical techniques in solving scientific and engineering problems are growing importance, and the subject has become an essential part of the training of applied mathematicians, engineers and scientists. The reason is numerical methods can provide the solution while the ordinary analytical methods fail [1]. Numerical methods have almost unlimited breadth of application. Other reason for lively interest in numerical procedures is their interrelation with digital computers [2]. Besides, parallel computing is a good platform to solve a large scale problem especially the numerical problem. This is proven through the successful implementation in solving elliptic, parabolic and hyperbolic problem. The outcomes of parallel performance measurements shows that parallel computing is time saving comparatively with the sequential computing as

well as other programming. Thus, the migration from sequential to parallel is worthwhile as it can reduce the execution time while maintaining computation accuracy.

ACKNOWLEDGMENT

The authors acknowledge Institute of Ibnu Sina, Universiti Teknologi Malaysia and Ministry of Science, Technology and Innovation, Malaysia for the financial support.

REFERENCES

[1] B. Bhat, Rama, Chakraverty, Snehasnish (2004), United Kingdom Numerical Analysis in Engineering, Alpha Science International Ltd.

[2] SMITH, G.D (1965), Numerical Solution of Partial Differential Equations, Oxford Universities Press

[3] Nakamura, Shoichiro (1993), Applied Numerical Methods In C, United State of America PTR Prentice Hall Inc.

[4] SMITH, G.D (1965), Numerical Solution of Partial Differential Equations, Oxford Universities Press

[5] Norma Alias, Md. Rajibul Islam, Nur Syazana Rosly (2009, March 16 – 18). "A Dynamic PDE Solver for Breasts' Cancerous Cell Visualization on Distributed Parallel Computing Systems", in Proc. The 8th International Conference on Advances in Computer Science and Engineering (ACSE 2009), Phuket, Thailand, Mar. 16-18, 2009.

[6] Evans D.J, Sukon K.S, (1995), The Alternating Group Explicit (AGE) Iterative Method for Variable coefficient Parabolic Equations, Intern. J. Computer Math. Vol 59, pp 107-121. 1995.

[7] Alias, N., Sahimi, M.S., and Abdullah, A.R., "The AGEB Algorithm for Solving the Heat Equation in Two Space Dimensions and Its Parallelization on a Distributed Memory Machine", Proceedings of the 10th European PVM/ MPI User's Group Meeting: Recent Advances In Parallel Virtual Machine and Message Passing Interface, Vol. 7, 2003, pp. 214–221.

[8] Alias, N., Sahimi, M.S., Abdullah, A.R., Parallel Strategies for the Iterative Alternating Decomposition Explicit Interpolation-Conjugate Gradient Method In solving Heat Conductor Equation on a Distributed Parallel Computer Systems. Proceedings of the 3rd International Conference on Numerical Analysis in Engineering. 3: 31-38. 2003.

[9] Norma Alias, Rosdiana Shahril, Md. Rajibul Islam, Noriza Satam, Roziha Darwis, "3D parallel algorithm parabolic equation for simulation of the laser glass cutting using parallel computing platform", The Pacific Rim Applications and Grid Middleware Assembly (PRAGMA15), Penang, Malaysia. Oct 21-24, 2008. (Poster presentation).

[10] SMITH, G.D (1965), Numerical Solution of Partial Differential Equations, Oxford Universities Press

[11] Nakamura, Shoichiro (1993), Applied Numerical Methods In C, United State of America PTR Prentice Hall Inc.

[12] SMITH, G.D (1985), United Kingdom, and Numerical Solution of Partial Differential Equations: Finite Difference Methods, Oxford Universities Press.

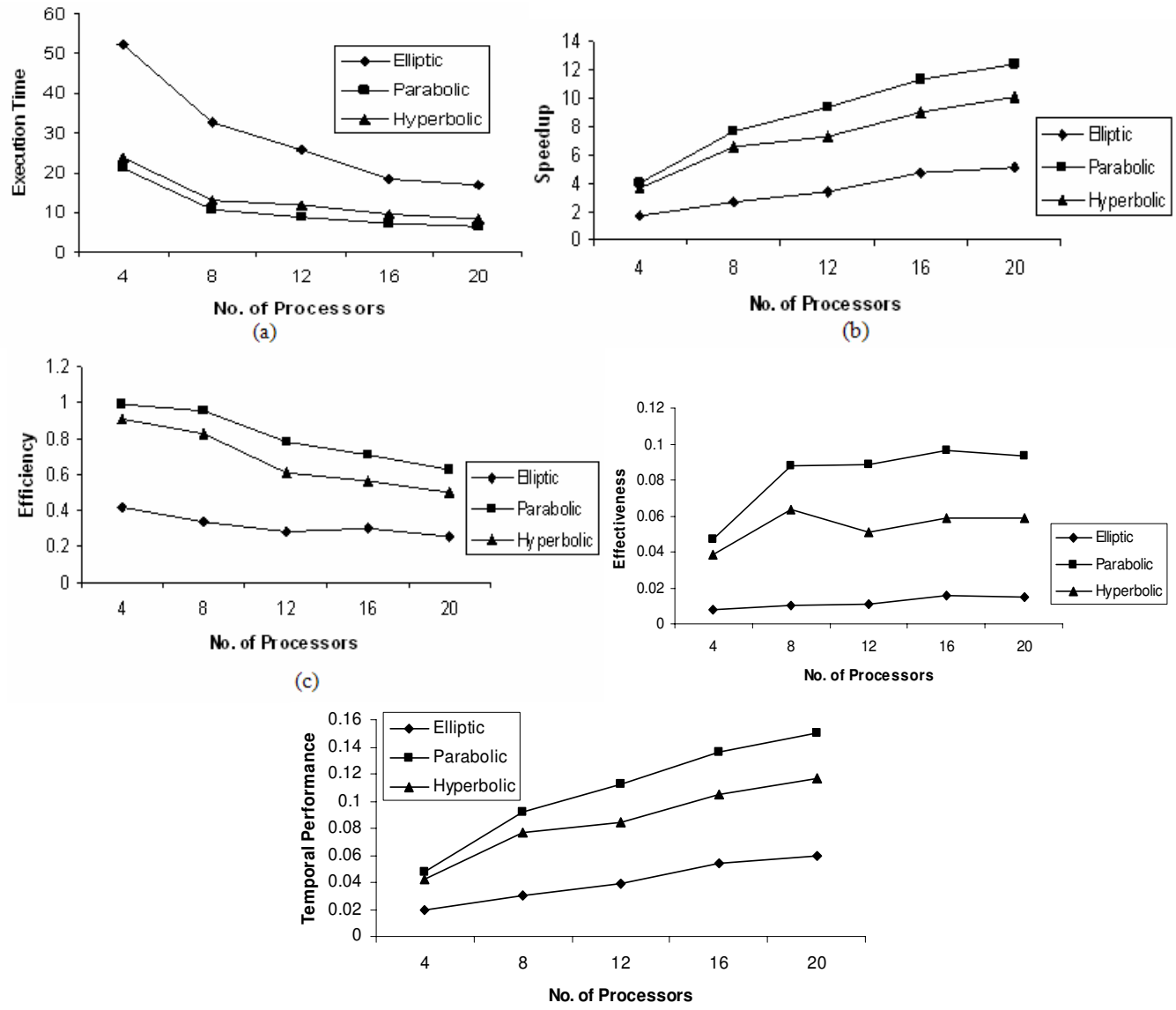

Figure 1. Parallel Performance Evaluation: (a)Execution Time, (b) Speedup, (c) Efficiency, (d) Effectiveness, (e) Temporal Performance