



Faster numerical weather forecasting using parallel computing with multi-mesh topology

Somen De

Department of Physics Bijoy Krishna Girls' College, Howrah 711204, India

Email: somenbkgc1@gmail.com

ABSTRACT

Today for different Meteorological observations attempts are being made for a faster parallel computing method for the purpose of numerical weather forecasting. Highly faster computation technique is required to study raw observation data into easily understandable and accurate weather forecast today. Faster technique will make it possible for meteorological data investigations successfully as well as to run complex weather forecasting models in lesser time, using highly complex set of weather models. The methodology that is used by meteorological study of weather forecasting consists usually of technique called numerical weather prediction. Numerical modeling of atmosphere is a complex process of solving a number of equations involving the future changes of weather variables, like temperature, pressure, humidity, wind speed etc. A simultaneous calculation of different parameters is a necessary part for accurate weather prediction. The multi-mesh topology is getting popular for its efficient topological properties for example: existence of Hamiltonian cycle, simple routing etc. In an $n \times n$ multi-mesh network used n^2 meshes of size $n \times n$ each, which themselves are again arranged in n rows and n columns so that there will be n^4 processors in total. Each $n \times n$ mesh in this network is termed as a block. Using this topology different numerical equation for weather prediction can be solved very faster. This paper presents an efficient and faster way for solving different numerical equation using multi-mesh topology with multiple processors acting simultaneously.

Keywords: Multi-mesh Topology, Parallel Computing, Weather Forecasting.

1. INTRODUCTION

The task of predicting the weather that will be observed at a future time is called weather forecasting. As one of the primary objectives of the science of meteorology, weather forecasting has depended critically on the scientific and technological advances in meteorology that have taken place since the latter half of the 19th century.

Throughout most of history, forecasting efforts at any given site depended solely on observations that could be made at that site. Observations of sky, wind, and temperature conditions and knowledge of local climate history permitted a limited predictive ability. Current weather-forecasting techniques were initiated by the theoretical work based on by solving complex numerical equations involving different meteorological parameters. Various numerical techniques are used in Numerical Weather Prediction (NWP) [1] to find the solution of the governing equations of the atmosphere. In this paper we have proposed and presented technical idea about different mathematical tools using multi-mesh topology for which this network is found to be more efficient with regard to computational time than the corresponding mesh with the same number of processors. Time efficient implementations of algorithms for solving numerical problems, e.g., partial

differential equation, Lagrange's interpolation computation have been discussed. The time complexity of Lagrange's interpolation on this network is $O(n)$ for n^2 data points compared to $O(n^2)$ time on mesh of the same number of processors. These techniques can be used efficiently for solving numerical equations involving different data points regarding NWP.

2. MULTI-MESH TOPOLOGY

The multi-mesh topology [2] is getting popular for its efficient topological properties for example: existence of Hamiltonian cycle, simple routing etc. In an $n \times n$ multi-mesh network, used n^2 meshes of size $n \times n$ each, which themselves are again arranged in n rows and n columns so that there will be n^4 processors in total. Each $n \times n$ mesh in this network is termed as a block. In a simple $n \times n$ mesh only $(n - 2)^2$ internal processors have degree four, the four corner processors are of degree two and $4(n - 2)$ boundary processors have degree three, as opposed to degree four for all processors on the multi-mesh shown in Figure 1.

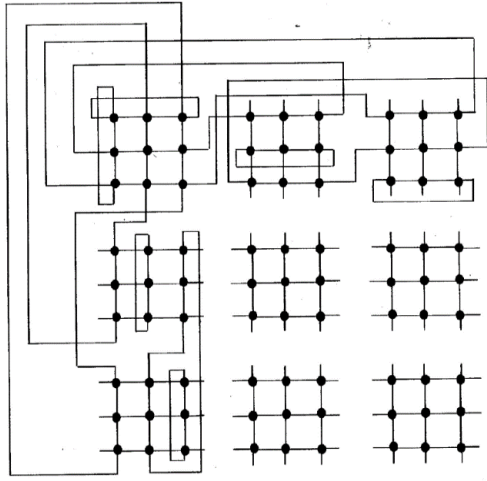


Figure 1. A Multi-mesh network with $n=3$ (all inter-block links are not shown)

A processor on the multi-mesh can be identified with a four tuple $P(\alpha, \beta, x, y)$ where α, β give the block address as the α^{th} row and β^{th} column, and x, y denote the processor address as x^{th} row and y^{th} column within that block. Each processor $P(\alpha, \beta, x, y)$ is connected to $P(\alpha, \beta, x \pm 1, y \pm 1)$, if they exist, using bi-directional links referred as intra-block links. There exist however, some additional bi-directional connections termed as inter-block links among the corners and boundary processors defined as follows:

- (i) $P(\alpha, \beta, 1, y)$ is connected to $P(y, \beta, n, \alpha)$ for $1 \leq y, \alpha, \beta \leq n$. As a special case, for $\alpha = y$, these links connect processors within the same block.
- (ii) $P(\alpha, \beta, x, 1)$ is connected to $P(\alpha, x, \beta, n)$ for $1 \leq x, \alpha, \beta \leq n$. As a special case, for $\beta = x$, these links connect processors within the same block.

3. EMULATION OF AN $n^2 \times n^2$ MESH TO A MULTI-MESH NETWORK

Different numerical weather prediction algorithms are mapped onto a mesh network because some useful properties of the mesh. In a simple $n^2 \times n^2$ mesh, a processor designated as $P(x,y)$ can know the stored in four of its neighbors, $P(x-1, y-1), 1 \leq x, y \leq n^2$, in constant time. Regarding this the 4-neighbor property of a mesh one can think that this property is destroyed in multi-mesh network. But, in the MM network, the processors $P(*, *, x, n)$ are directly connected to processors $p(*, *, x-1, n)$ by two intra-block links in the vertical direction, for $x \neq 1$ or n . They are also connected to the processors $P(*, *, x, n-1)$ by an intra-block link in the horizontal direction. So, the processors $P(*, \beta, x, n)$ for odd β can and the processors $P(*, \beta+1, x, n), \beta < n$, can exchange data in three steps, i.e. in $O(1)$ time. Thus, for odd $\beta, P(*, \beta, x, n)$ and $P(*, \beta+1, x, n), 1 \leq \beta < n$, are in a sense, neighbors of each other. Similarly, it is also true for even β . Thus the 4-neighbor property of the mesh interconnection can be emulated [3] by the multi-mesh network in constant time. The initial distribution of data to preserve desired adjacency among different data elements on an MM network with $n = 3$ is shown in Figure 2.

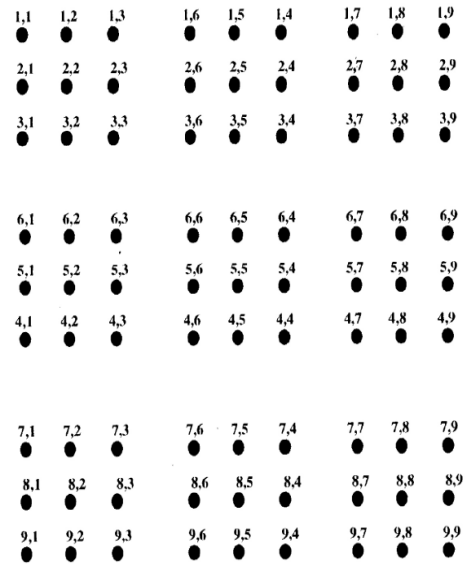


Figure 2. Initial distribution of processors of a 9×9 matrix on the MM network for emulating 9×9 mesh

4. NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATION

One of the greatest needs in Numerical Weather Prediction is a general and reasonably short method of solving partial differential equations by numerical methods [4]. Certain types of boundary-value problems can be solved by replacing the partial differential equations, such as Laplace's, Poisson's and several others, by the corresponding difference equations and then solving the later by a process of iteration.

A difference quotient is the quotient obtained by dividing the difference between two values of a function by the difference between the two corresponding values of the independent variable. Thus for function $f(x)$ of a single variable, the difference quotient is $\frac{f(x+h)-f(x)}{h}$, whose limiting value is the derivative of $f(x)$ with respect to x , the approximation becoming closer as h becomes smaller.

Partial difference quotients of the second and higher orders are best constructed with reference to a network of the points in the xy plane for a function, say, $u(x, y)$, of two variables as shown in the Figure 3. The points of intersection of the lines are called the lattice points.

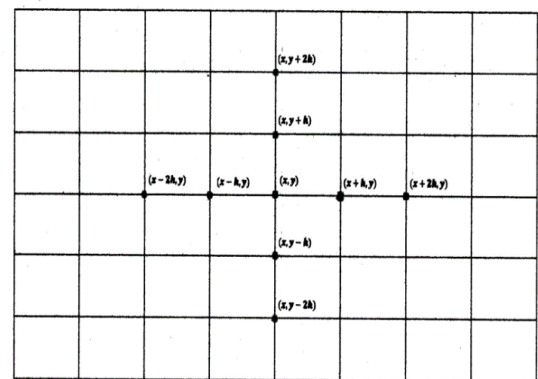


Figure 3. Lattice points around (x, y) in $x-y$ plane

Now, we shall consider as an example the Laplace's equation for two dimensions,

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0$$

Replacing $\frac{\partial^2 V}{\partial x^2}$ and $\frac{\partial^2 V}{\partial y^2}$ by $u_{\bar{x}\bar{x}}$ and $u_{\bar{y}\bar{y}}$ respectively (x and \bar{x} are two different points on x -axis, y and \bar{y} are similar points in y -axis), we get

$$\frac{u(x+h,y)-2u(x,y)+u(x-h,y)}{h^2} + \frac{u(x,y+h)-2u(x,y)+u(x,y-h)}{h^2} = 0$$

(Using forward and backward second difference quotient of $u(x, y)$ with respect to x and y respectively.)

From the above, we can get

$$u(x, y) = \frac{1}{4} [u(x+h, y) + u(x, y+h) + u(x-h, y) + u(x, y-h)]$$

To solve Laplace's equation in two variables with the given boundary conditions, we start with the given boundary values (the 'a' values in Figure 3) and the initial value of the interior lattice points (the 'u' values in Figure 4). The rough values of u 's are computed as

$$\begin{aligned} \hat{u}_1 &= \frac{1}{4} (u_2 + a_2 + a_{32} + u_8) \\ \hat{u}_2 &= \frac{1}{4} (u_3 + a_3 + \hat{u}_1 + u_9) \\ \hat{u}_3 &= \frac{1}{4} (u_4 + a_4 + \hat{u}_2 + u_{10}) \\ &\dots \\ \hat{u}_{49} &= \frac{1}{4} (\hat{u}_{48} + a_{16} + a_{18} + \hat{u}_{42}) \end{aligned}$$

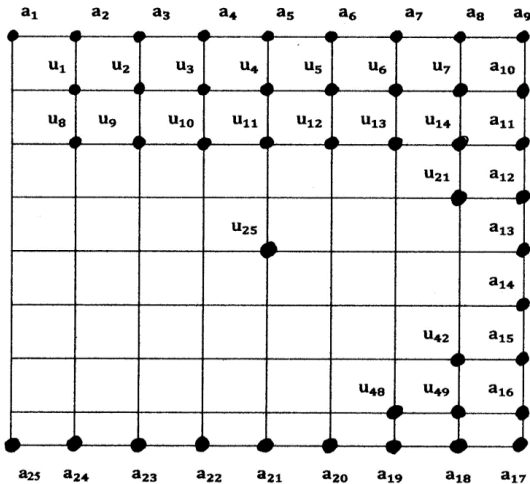


Figure 4. The boundary values and the interior lattice values in the grid

To get the initial values for the interior points of the network, first the value of u_{25} is calculated at the centre of the square by taking mean of four boundary values, i.e., a_{29} , a_{13} , a_5 , and a_{21} . Then the values are calculated for the centre of four large squares, i.e., u_9 , u_{13} , u_{37} and u_{41} . These values are found by taking the means of the values at the ends of the diagonals of the large squares and the process will be

repeated in this way if still some points are left.

A. Implementation using Multi-Mesh Network

Since the blocks in multi-Mesh are connected by their inter-block links, the computation of the initial values of the interior lattice points can be done quicker than in a mesh. For a $n^2 \times n^2$ mesh the calculation of initial value of the center lattice points requires $O(n^2)$ communication time to bring the four boundary values together in center lattice points whereas in Multi-Mesh it can be done in $O(n)$ communication time, since all the four blocks which contain the four boundary points are $O(n)$ distance far from the center block. The same is applicable for all successive computations of other interior lattice points.

For iterative computations of the rough values of the lattice points, the four neighborhood property of mesh can be achieved in Multi-Mesh as shown in the emulation of mesh in Multi-Mesh in section

5. LAGRANGE'S INTERPOLATION

One of the powerful mathematical tools for solving numerical equations is Lagrange's Interpolation. This technique is widely used by meteorological stations for Numerical Weather Prediction.

We can assume v_1, v_2, \dots, v_n be the given values of $F(u)$ at u_1, u_2, \dots, u_n respectively, so that we can find $v_i = F(u_i)$. We can interpolate $F(u)$ at the value \bar{u} using the N -point Lagrange's interpolation formula given by,

$$F(\bar{u}) = \Pi(\bar{u}) \sum_i [v_i / \{(\bar{u} - u_i) \pi'(u_i)\}]$$

where

$$v_i = F(u_i), \pi(\bar{u}) = (\bar{u} - u_1)(\bar{u} - u_2)(\bar{u} - u_3) \dots (\bar{u} - u_N),$$

And

$$\pi'(u_i) = \frac{(u_i - u_1)(u_i - u_2) \dots (u_i - u_{i-1})(u_i - u_{i+1}) \dots (u_i - u_N)}{(u_i - u_{i+1}) \dots (u_i - u_N)}$$

A. Parallel Implementation Using the Multi-Mesh Network

One can use an MM network of n^4 processors for $N(= n^2)$ -point Lagrange's interpolation. The basic idea of this algorithm [5] is given below:

At first the data elements $u_{(\beta-1)n+\alpha}$ and $u_{(\alpha-1)n+\beta}$ are fed to the processors $P(\alpha, \beta, n, 1)$ and $P(\alpha, \beta, 1, n)$ respectively, $\forall \alpha, \beta, 1 \leq \alpha, \beta \leq n$. Also, \bar{u} is fed to the processors $P(1, 1, 1, 1)$. $v_{(\alpha-1)n+\beta}$ are fed to the processors $P(\alpha, 1, \beta, 1)$, $\forall \alpha, \beta, 1 \leq \alpha, \beta \leq n$.

The differences $(\bar{u} - u_i)$ s is then computed at the diagonal processors of the diagonal blocks, which are partially multiplied in each block. These are then brought to a single block by using the inter block links, the product term $\pi(\bar{u})$ is computed there, and stored in processors $P(1, 1, 1, 1)$.

Similarly, each of the differences $(\bar{u} - u_i)$, $(u_i - u_1)$, etc., required for computing $(\bar{u} - u_i) \pi'(u_i)$ for each i , $i = 1, 2, \dots, n$, is computed in a separate processors. These differences are then partially multiplied in each block and then brought to a single block for final multiplication using the inter-block links for all i , v_i s is then divided by this product $(\bar{u} - u_i) \pi'(u_i)$. By using the inter-block links again,

these results are brought to a single block, summed up there, and then multiplied by $\Pi(\bar{u})$ to give the final result. The detailed steps of the algorithm are given the Ph.D. thesis of De M. [5].

6. CONCLUSIONS

New numerical models continue to be developed as supercomputers become more powerful. It is not simply a matter of doing more and more computations, however. Some approximations in such models depend on other parts of the solution being sufficiently simple to make the resulting approximation satisfactory.

As numerical models improve, meteorologists are reconsidering the concept of predictability. How far ahead can time- or area-averaged quantities be usefully predicted? Is it possible to identify occasions when the atmosphere is more predictable than at other times? Meteorologists recognize that in the prediction step of forecasting, current statistical models should in time be replaced with expert systems — that is, artificial intelligence systems. This idea, however, is only in the beginning stages of development. The greatest potential for improvement in forecasting appears to lie in the short and medium ranges, while experimental work

will characterize the extended range. Improvements in daily forecasting are likely to increase at a relatively minor pace.

In this paper, attempted have been made to give a technical idea for solving numerical equations regarding Numerical Weather Prediction efficiently in Multi-Mesh network.

REFERENCES

- [1] Richardson L.F. (1922). *Weather prediction by numerical process*, Cambridge University Press, Cambridge.
- [2] De M., Das D., Sinha B.P. (1994). A new network topology with multiple meshes, Tech. Rep. No. T/Rep / E – 94 / 01, Indian Statistical Institute, Calcutta.
- [3] Das D., De M., Sinha B.P. (1999). A new network topology with multiple meshes, *IEEE Trans. On Computer*, Vol. 48, pp. 536-551. DOI: [10.1109/12.769436](https://doi.org/10.1109/12.769436)
- [4] *Numerical Mathematical Analysis* (1966). J.B. Scarborough, Sixth Edition, Oxford and IBH Publishing Co., London, pp. 392-400.
- [5] De M. (1997). Design of efficient parallel algorithms and architectures for some numeric and non-numeric problems, Ph.D. thesis, J.U., Kolkata.