(NASA-CR-199768) AN EXPERIMENT IN HURRICANE TRACK PREDICTION USING PARALLEL COMPUTING METHODS (Oklahoma Univ.) 18 p

N96-16144

Unclas

G3/47 0084500

Parallel Algorithms and Applications, Vol. 2, pp. 315–332
Reprints available directly from the publisher
Photocopying permitted by license only

AN EXPERIMENT IN HURRICANE TRACK PREDICTION USING PARALLEL COMPUTING METHODS

CHANG G. SONG, JUNG-SING JWO, S. LAKSHMIVARAHAN and
S. K. DHALL

Parallel Processing Institute, School of Computer Science, University of Oklahoma, Norman, OK 73019, USA

JOHN M. LEWIS

National Severe Storms Laboratory, NOAA/ERL, Norman, OK 73069, USA

CHRISTOPHER S. VELDEN

Cooperative Institute for Meteorological Satellite Studies, University of Wisconsin, Madison, WI 53706, USA

(Received June 2, 1993)

The barotropic model is used to explore the advantages of parallel processing in deterministic forecasting. We apply this model to the track forecasting of hurricane Elena (1985).

In this particular application, solutions to systems of elliptic equations are the essence of the computational mechanics. One set of equations is associated with the decomposition of the wind into irrotational and nondivergent components—this determines the initial nondivergent state. Another set is associated with recovery of the streamfunction from the forecasted vorticity. We demonstrate that direct parallel methods based on accelerated block cyclic reduction (BCR) significantly reduce the computational time required to solve the elliptic equations germane to this decomposition and forecast problem. A 72-h track prediction was made using incremental time steps of 16 min on a network of 3000 grid points nominally separated by 100 km. The prediction took 30 sec on the 8-processor Alliant FX/8 computer. This was a speed-up of 3.7 when compared to the one-processor version.

The 72-h prediction of Elena's track was made as the storm moved toward Florida's west coast. Approximately 200 km west of Tampa Bay, Elena executed a dramatic recurvature that ultimately changed its course toward the northwest. Although the barotropic track forecast was unable to capture the hurricane's tight cycloidal looping maneuver, the subsequent northwesterly movement was accurately forecasted as was the location and timing of landfall near Mobile Bay.

KEY WORDS: Barotropic model, Poisson's equation, block cyclic reduction, parallel computing.

C.R. CATEGORIES: G.1.8, J.2, F.2.1.

1. INTRODUCTION

The use of barotropic models to track hurricanes (typhoons) was initiated in Japan during the early 1950s (Sasaki and Miyakoda, 1954). Research continued into the 1960s in both Japan and the U.S.A., where the success of research models eventually led to operational implementation at the National Hurricane Center (NHC) in

1968 (Sanders et al., 1975). For a history of barotropic modeling as applied to the hurricane tracking problem and commentary on its relevance in current research, see DeMaria (1985) and Shapiro and Ooyama (1990).

The sensitivity of barotropic track forecasting to perturbations in the initial conditions was convincingly demonstrated in the paper by Sanders and Burpee (1968). Determination of the initial state is especially difficult due to the scarcity of data over the oceans where tropical storms develop. Most observations over the ocean are obtained from weather satellites. Typically, these observations are wind estimates made from a sequence of photographs or images—so-called "cloud-tracked winds" or "water-vapor winds" (Velden et al., 1992). Since the barotropic model assumes a nondivergent wind, the observations of wind must be decomposed or partitioned into nondivergent and irrotational components.

In an effort to improve the initial conditions for barotropic forecasting, recent efforts have been made to include a time history of wind analysis into the estimate of the initial state. To accomplish this task in a dynamically consistent fashion, adjoint methods have been used (Lewis et al., 1987; DeMaria and Jones, 1991). These assimilation/initialization methods require a model of comparable complexity to the forecast model. The so-called adjoint model is integrated backward in time and is used iteratively with the forward model (forecast model) to find an optimal initial condition that minimizes the discrepancy between model forecast and observations. Although aesthetically pleasing and theoretically attractive, the computational demands have made this strategy prohibitive in the operational environment. With the research models tested to date, the computational cost of an adjoint assimilation with barotropic models is ≈ 10–20 times as costly as the forecast alone (Lewis and Derber, 1985; Talagrand and Courtier, 1987). Realistic implementation of adjoint assimilation in operations will require the maturing technology of multicomputer and multiprocessors (Lakshmivarahan and Dhall, 1990). We refer to this technology as parallel processing or parallel computing.

Our ultimate aim is to apply parallel computing technology to both the forecasting and assimilation problem (adjoint assimilation). We concentrate on the forward problem in this initial effort, but the transfer of the computational algorithms to the backward or adjoint problem should be straightforward. We apply our parallel processing mechanics to the track forecast of hurricane Elena (1985). This case is especially challenging because of the looping maneuver that the storm executed in the northeastern Gulf of Mexico. This case study also benefits from an unusually large set of synoptic-scale wind analyses produced at the Space Science and Engineering Center (University of Wisconsin–Madison).

In Section 2, following Lynch (1988, 1989), we describe a computational scheme for decomposing a windfield into non-divergent and irrotational components. This scheme leads to the solution of two related Poisson's equations. The baratropic model itself is described in Section 3. This has two components—one is the time-dependent evolution of vorticity and the second is a Poisson's equation relating the stream function and vorticity. Since the solution of Poisson's equation is central to the computations, we use a direct parallel method, called the block cyclic reduction, for solving these equations. A summary of this method is given in Appendix A. Section 4 discusses various issues related to choice of grid, boundary conditions, nature and type of discrete approximations, and conditions for stability. Speed-up analysis resulting from solving this model on the state of the art multivector processor Alliant FX/8 is given in Section 5. Concluding statements are given in Section 6.

2. DECOMPOSITION

To facilitate our work, we have mapped the earth onto the Mercator projection. In the kinematic and dynamics equations that follow, the east–west and north–south coordinates—x and y, respectively—are distances on the Mercator plane and the image scale factor, m, is the ratio of distance on this projection to the corresponding distance on earth. This image factor is given by $(\cos\phi_0/\cos\phi)$ where ϕ_0 is the latitude of intersection between the Mercator cylinder and the spherical earth ($\phi_0 = 30°$N in our case). Saucier (1955) discusses the geometry of the Mercator projection and includes the derivation of the image scale factor.

Let **V** denote the *horizontal wind vector*, χ the *velocity potential*, Ψ the *stream function*, ζ the *vorticity* and δ the *divergence*. Let $\mathbf{V}_\psi$ and $\mathbf{V}_\chi$ be the nondivergent and the irrotational components of this wind field, respectively. Helmholtz's theorem permits decomposition of **V** as follows:

$$\mathbf{V} = \mathbf{V}_\psi + \mathbf{V}_\chi, \tag{1}$$

where

$$\mathbf{V}_\chi = \mathbf{k} \times \nabla\Psi, \tag{2}$$

$$\mathbf{V}_\chi = \nabla\chi, \tag{3}$$

and $\nabla = (m(\partial/\partial x), m(\partial/\partial y))$ is the $2-d$ gradient operator. To simplify notation, the image-scale factor will be assumed to be associated with all differential operators, but it will not be explicitly written. Thus, $\partial/\partial x$ is in actuality $m(\partial/\partial x), (\partial^2/\partial x^2)$ means $m(\partial/\partial x)(m(\partial/\partial x))$, etc. Let **i**, **j** and **k** be unit vectors in the 2 orthogonal horizontal directions (x, y) and the vertical direction (**z**), respectively. Since $\nabla \cdot \mathbf{V}_\psi = 0$ and $\nabla \times \mathbf{V}_\chi = 0$, it follows that

$$\zeta = \mathbf{k} \cdot \nabla \times \mathbf{V} = \mathbf{k} \cdot \nabla \times (\mathbf{k} \times \nabla\Psi) = \nabla^2\Psi \tag{4}$$

and

$$\delta = \nabla \cdot \mathbf{V} = \nabla^2\chi \tag{5}$$

where

$$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right).$$

Since $\mathbf{V} = \mathbf{i}u + \mathbf{j}v$, then

$$\zeta = \mathbf{k} \cdot \nabla \times \mathbf{V} = \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) \tag{6}$$

Table 1 Lynch's Eight Different Variations of Boundary Conditions for Decomposing the Wind Field (Dirichlet and Neuman conditions are denoted by D and N, respectively)

Version Number	First Boundary Condition	Second Boundary Condition	Type of First and Second B.C.
1	$\chi = 0$	$\Psi = \int_{S_0}^{S}\left(\frac{\partial\chi}{\partial n} - V_n\right) dS$	DD
2	$\frac{\partial\chi}{\partial n} = \gamma_n$	$\Psi = \int_{S_0}^{S}(\gamma_n - V_n) dS$	ND
3	$\Psi = 0$	$m\frac{\partial\chi}{\partial n} = V_n$	DN
4	$m\frac{\partial\Psi}{\partial n} = \gamma_s$	$m\frac{\partial\chi}{\partial n} = V_n + m\frac{\partial\Psi}{\partial s}$	NN
5	$\chi = 0$	$m\frac{\partial\Psi}{\partial n} = V_s$	DN
6	$m\frac{\partial\chi}{\partial n} = \gamma_n$	$m\frac{\partial\Psi}{\partial n} = V_s - m\frac{\partial\chi}{\partial n}$	NN
7	$\Psi = 0$	$\chi = \int_{S_0}^{S}\left(V_S - m\frac{\partial\Psi}{\partial n}\right) dS$	DD
8	$m\frac{\partial\Psi}{\partial n} = \gamma_S$	$\chi = \int_{S_0}^{S}(V_S - \gamma_S) dS$	ND

and

$$\delta = \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right). \tag{7}$$

To extract the nondivergent wind component from the observed wind, we first compute the vorticity and divergence which serve as the forcing functions in the solution to Poisson's equations, Eqs. (4) and (5). These equations are coupled through the boundary condition (b.c.'s) which can assume a variety of forms. As discussed by Lynch and collaborators (Bijlsma et al., 1986; Lynch, 1988, 1989), the decomposition is not unique in a limited domain and a wide variety of results are possible that depend on the choice of b.c.'s. Table 1 displays eight different sets of b.c.'s that have been tested by Lynch (1988). The first four versions use the normal component ($V_n\hat{n}$) of the observed wind on the boundary whereas the last four use the tangential component ($V_s\hat{s}$). The parameters

$$\gamma_n\left(= \frac{\partial\chi}{\partial n} \equiv \frac{1}{L}\oint \mathrm{V}_n\, dS\right) \quad \text{and} \quad \gamma_S\left(= \frac{\partial\Psi}{\partial n} \equiv \frac{1}{L}\oint \mathrm{V}_S\, dS\right)$$

are constant gradient conditions on the boundary that satisfy divergence theorem identities—so-called compatibility conditions. Section 2 of Lynch (1988) can be read for a more detailed discussion of these boundary condition choices.

Our choices are narrowed by choosing b.c.'s that minimize the kinetic energy in the divergent component (K_χ). This is dictated by the desire to minimize the kinetic energy in the "largely spurious divergent part of the wind" (Sanders and Burpee, 1968). As shown by Davies–Jones (1988) and Lynch (1988), $\chi = 0$ on the boundary is sufficient to minimize K_χ and furthermore causes the nonphysical cross-term $K_{\psi\chi}$ to vanish (again see Lynch (1988) for details). Version 1 (due to Sangster (1960)) and version 5 are, therefore, candidates. Dirichlet b.c.'s are easier to implement than Neumann conditions since the solution is unique, i.e., it does not involve an arbitrary additive constant, nor is there a compatibility condition. Thus, version 1 is preferred. Care must be exercised, however, to insure that

$$\Psi_B(S) = \int_{S_0}^{S} \left(\frac{\partial\chi}{\partial n} - V_n\right) dS$$

is single valued; subscript B is used to indicate the value on the boundary. This requirement can be satisfied by first solving (5) for χ with $\chi_B = 0$, computing $\partial\chi/\partial n\,|_B$ and adjusting $\mathbf{V}_n$ (to $\hat{\mathbf{V}}_n$) by a *constant amount* ε *around the boundary* such that

$$\int_S \left(\frac{\partial\chi}{\partial n} - V_n - \varepsilon\right) dS = 0.$$

The final step is to solve (4) with

$$\Psi_B = \int_{S_0} \left(\frac{\partial\chi}{\partial n} - \hat{V}_n\right) dS \qquad \text{where} \quad \Psi_B(S_0) = 0$$

by choice. Thus, version 1 has the same desirable properties as version 5 without the complexities of a Neumann b.c. The only difference is that version 1 uses $\mathbf{V}_n$ on the boundary while version 5 uses $\mathbf{V}_S$. There doesn't appear to be any physical reason for preferring $\mathbf{V}_S$ over $\mathbf{V}_n$, or *vice versa*.

3. THE BAROTROPIC MODEL

The approach to barotropic forecasting follows Leith (1978) and DeMaria (1987). The prognostic equation is given by

$$\frac{\partial\zeta}{\partial t} = \mathbf{J}(\zeta + f, \Psi), \tag{8}$$

where $\mathbf{J}$ is the Jacobian and f is the Coriolis parameter.

In this prognostic equation, ζ represents a modified vorticity given by

$$\zeta = (\nabla^2 - \sigma^2)\Psi. \tag{9}$$

The derivation of (8) is found in Leith (1978) and follows from operations on the shallow water equations. As discussed in DeMaria (1987), this form of the equation amounts to a divergence-corrected barotropic model which prevents the retrogression of ultralong Rossby waves. The parameter σ is empirically determined and is taken to be $1/800(\text{km}^{-1})$ based on a recommendation from DeMaria (personal communication). We still use the nondivergent wind to initialize the model.

Given the wind components (u, v) at each of the grid points, a scheme for computing the solution of (8) is described as follows:

(a) Compute ζ and δ from the initial wind field, then solve (4) and (5) for Ψ and χ following the decomposition process described in Section 2.

(b) Use (8) to step forward and find ζ at the next incremental time. The vorticity at the boundary is calculated differently, depending on whether there is inflow or outflow at the boundary. If outflow, the vorticity is calculated by a one-sided finite difference version of (8). If inflow, the vorticity is unchanged.

(c) Using the forecasted vorticity, (9) is solved for Ψ.

(d) Repeat steps (b) and (c) until the entire forecast period is covered.

Both the decomposition of the wind field and the barotropic forecast hinge on the solution to Poisson's equation. To find the solution to this equation, we apply a direct method called block cyclic reduction (BCR). Since the use of this method in a parallel processing mode is critical to our work, salient features of this method are given in the appendix. For more details, refer to Buzbee, Golub and Nielson (1970), Hockney (1965), Lakshmivarahan and Dhall (1991), Swartztrauber (1984), and Sweet (1977, 1988).

4. APPLICATION TO HURRICANE ELENA

Hurricane Elena (1985) had its origin in a rapidly moving tropical wave that was first analyzed over the Saharan Desert (Velden, 1987). It moved over the Atlantic Ocean at a rate of ≈ 15 ms^{-1} and appeared as a tropical depression over Cuba on August 28th. The depression intensified and was classified as a tropical storm by 00 UTC on the 29th. It was upgraded to hurricane status 12 hours later when it was centered at $\sim$ 25N, 85W. Subsequent positions at 12 h intervals are indicated by dots along the solid line in Figure 1. The initial time for our forecast was 12 UTC 31 August, when the storm occupied the position indicated by (§) in Figure 1.

a. Numerics: All experiments were conducted on the ALLIANT FX/8 using double precision arithmetic. Five separate modes of computation were used starting with the scalar mode—use of a single processor without the benefit of vectorization. This was followed by computation with 1, 2, 4, and 8 processors with vectorization. Since the BCR-partial fraction method was more efficient than the polynomial factorization form of BCR (Jwo et al., 1992), all experiments used the partial fraction approach (see Appendix).

The equations are solved on the staggered mesh of grid points displayed in Figure 2. This staggering is in accord with Arakawa's C-grid convention (Arakawa, 1966). The finite-difference approximation to the Jacobian conserves the vorticity, the kinetic energy, and the squared vorticity.

Following DeMaria (1985, 1987), we adopted the Adams–Bashford scheme for integrating the model (Anderson et al., 1984). As suggested by DeMaria (personal communication), a reasonable limitation on the time-step is given by

$$\Delta_t \leq \frac{\Delta}{2m\sqrt{2}(u^2+v^2)_{\max}}(25 \text{ min}),$$

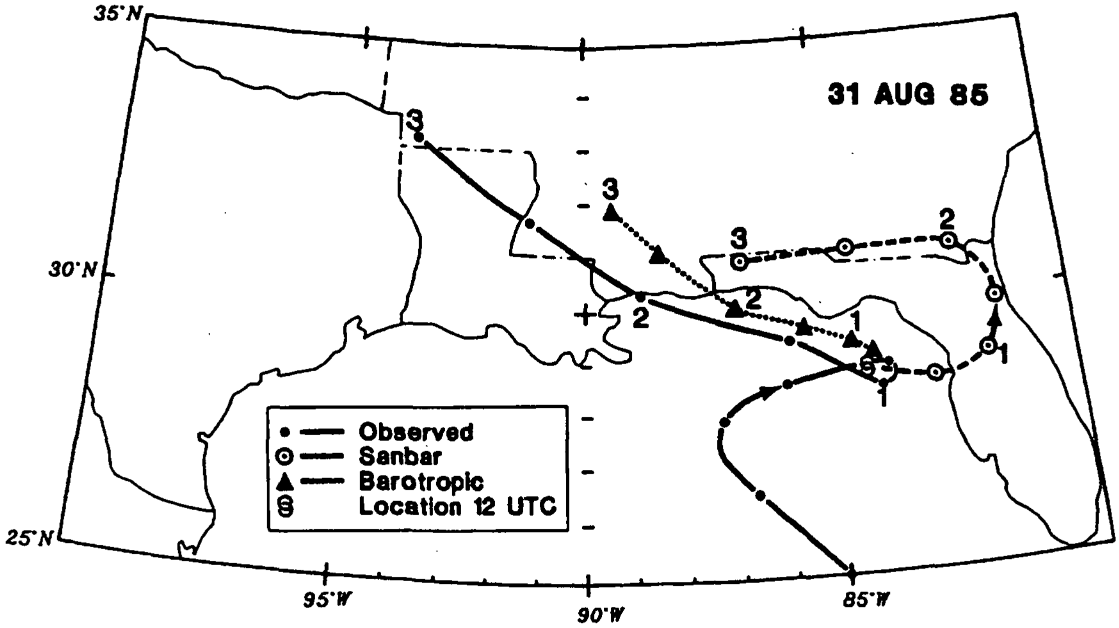

Figure 1 The observed and forecasted tracks of Elena (1985). The location of Elena at the initial time (t = 0, 12 UTC Aug. 31) for barotropic forecasts is indicated by ( ). The numbers 1, 2, and 3 indicate the location of Elena at 1, 2, and 3 days after t = 0. Our forecast is termed barotropic and the operational model is SANBAR.

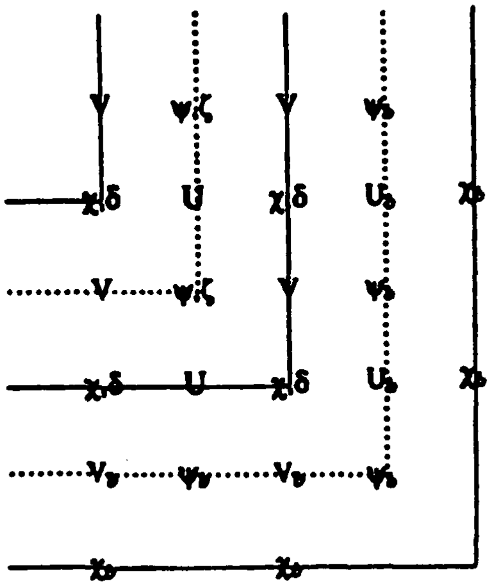

Figure 2 The lower right hand corner of the staggered grid used in our study. The boundary values are denoted by the use of subscript 'b'.

where max signifies the largest wind speed, Δ and Δ_t are the space and time increments, and m is the image scale factor at the location of the maximum speed. As indicated in parentheses, the limiting Δ_t is 25 min for our grid and wind field. A time step of 16 minutes was used in all of our computations.

Prior to the forecast, an artificial axisymmetric vortex—representing the hurricane circulation—is superimposed on the Ψ-field found from the decomposition process. This helps us locate the hurricane at subsequent times. The vortex remains intact through the 72 h forecast period, but there is a modest amount of numerical smoothing that weakens the vortex. The mathematical form of the vortex is:

$$\mathbf{V}_H(r) = \mathbf{V}_m \left(\frac{r}{r_m}\right) \exp\left\{\frac{1}{b}\left[1 - \left(\frac{r}{r_m}\right)^b\right]\right\} \quad \text{and} \quad \zeta_H(r) = \frac{1}{r}\frac{\partial}{\partial r}(r\mathbf{V}_H), \tag{10}$$

where $\mathbf{V}_H$ is the tangential wind and ζ_H the associated vorticity. The parameters in the wind equation are: $\mathbf{V}_m$ (the maximum tangential wind), r (the distance from the storm center), r_m (the radius of maximum wind), and b (an empirical parameter that determines the rate at which $\mathbf{V}_H$ decreases at large radii). We have used DeMaria's values for these parameters to typify observed Atlantic tropical storms (DeMaria, 1987). These values are $V_m = 25\ \mathrm{ms}^{-1}$ and $r_m = 150$ km, and the parameter b is chosen such that $\mathbf{V}_H = 5\ \mathrm{ms}^{-1}$ at 600 km (this gives a value of $b = 0.998$).

b. Track forecast: Barotropic forecasting for a particular level in the tropical atmosphere has been unsuccessful for a variety of reasons that have been enumerated in Sanders and Burpee (1968). They determined " ... that forecasting should be carried out for the tropospheric mean flow ... rather than by selection of a single, presumably representative level." This work was the foundation of the operational model developed at NHC (Sanders et al., 1975). The operational model has become known as the Sanders Barotropic model, or SANBAR in acronymic form. Following the work of Sanders and Burpee (1968) and Sanders et al. (1975), we calculate a mean wind throughout the troposphere using satellite-derived winds and radiosondes following the procedures established at the Space Science and Engineering Center in Madison, Wisconsin (UW–SSEC) (Velden et al,, 1984). In subsequent discussion, we refer to these winds as deep-layer-mean (DLM) winds.

Figure 3 shows the DLM wind field at $t = 0$ (12 UTC 31 Aug.) plotted over our analysis and forecast domain. A well-defined trough that extends from New England down through the northeast corner of the Gulf of Mexico is apparent in the DLM wind vectors. The initial streamfunction field associated with the DLM wind is shown in Figure 4. The artificial hurricane vortex described in Section 4a has been superimposed on this large-scale Ψ field. The forecasted stream function fields at $t = 72$ h is shown in Figure 5. The path of the hurricane over the three-day period has been superimposed on the 72 h forecast of streamfunction.

We estimated Elena's position in our forecasts by finding the center of the imbedded vortex at 12 h intervals. This is done by finding the minimum Ψ value or the maximum ζ value. We found insignificant difference in positioning Elena by either method and the forecasted positions at 12 h intervals are displayed in Figure 1. The observed track as well as the forecasted track from SANBAR are also shown in this figure. The observed tight cycloidal motion (re-curvature) between $t = 0$ and $t = 24$ h was not captured by our model forecast. This small scale feature in the trajectory is near the limit of the model resolution, i.e., the area outlined by this "teardrop" shaped part of the trajectory is $\approx (0.5°\ \mathrm{lat})^2 = (50\ \mathrm{km})^2$. A grid

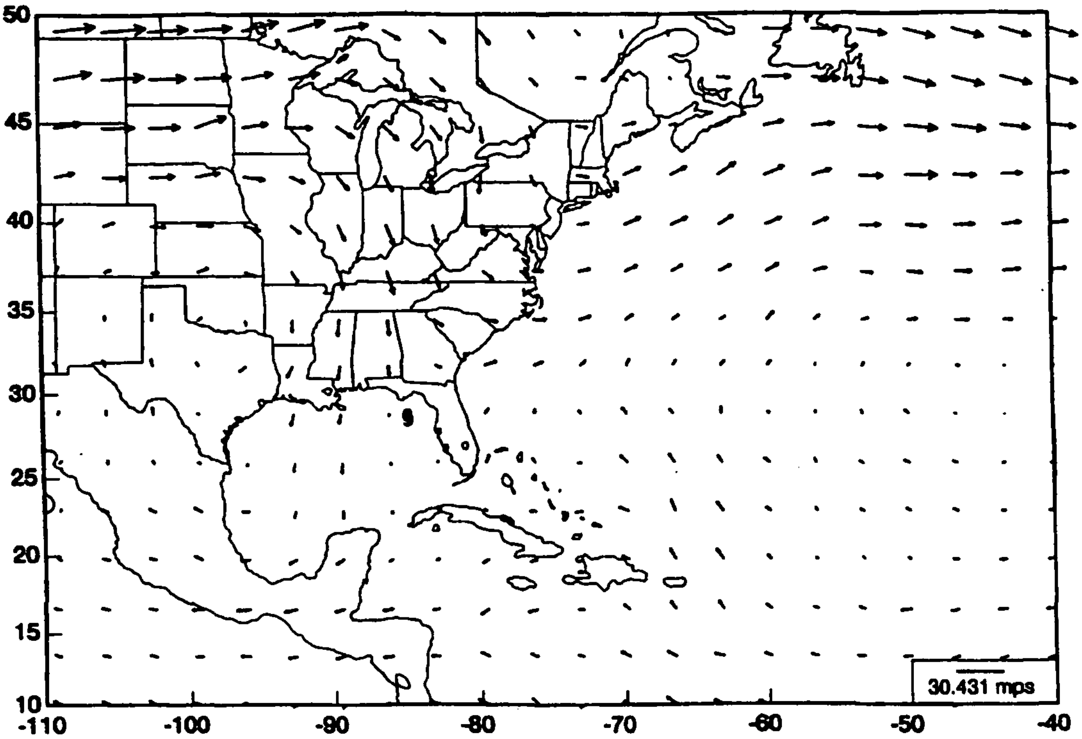

Figure 3 Distribution of deep-layer-mean winds (DLM) at t = 0 (12 UTC, 31 Aug 85). The wind speed is scaled by the largest speed which is shown in the lower right hand corner ($\approx$ 30 ms^{-1}). Elena's position at this time is indicated by ( ).

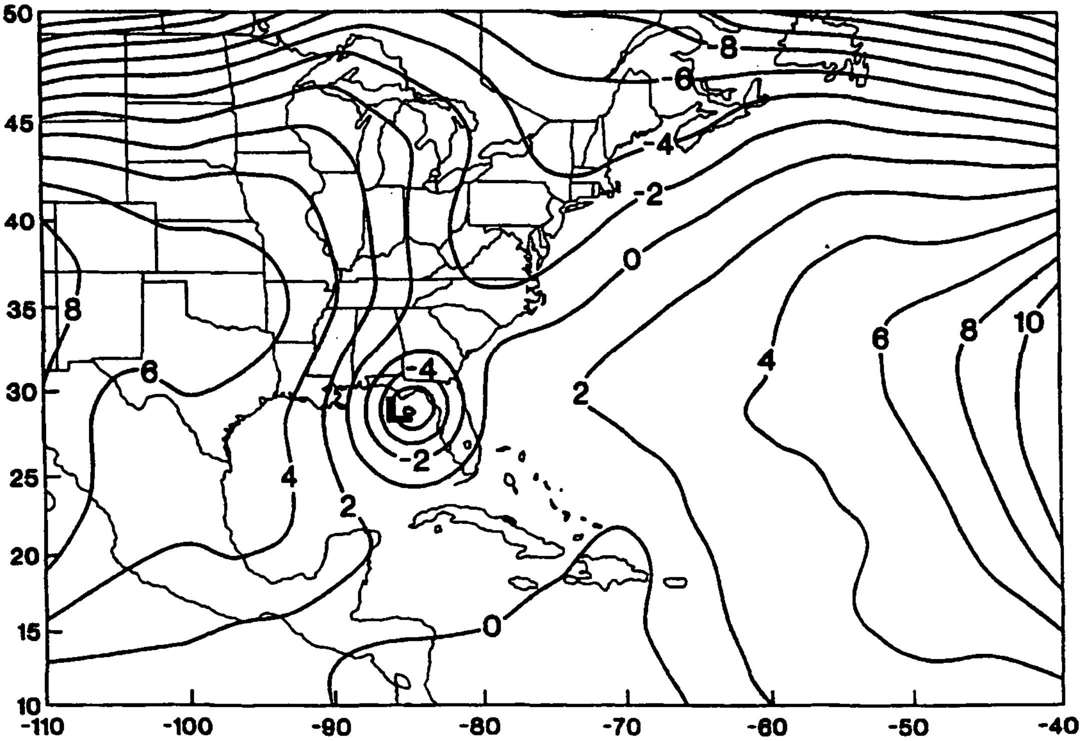

Figure 4 Streamfunction associated with the DLM winds and imbedded vortex at t = 0. The contours of the streamfunction are drawn at intervals of 10^6 m^2s^{-1}.

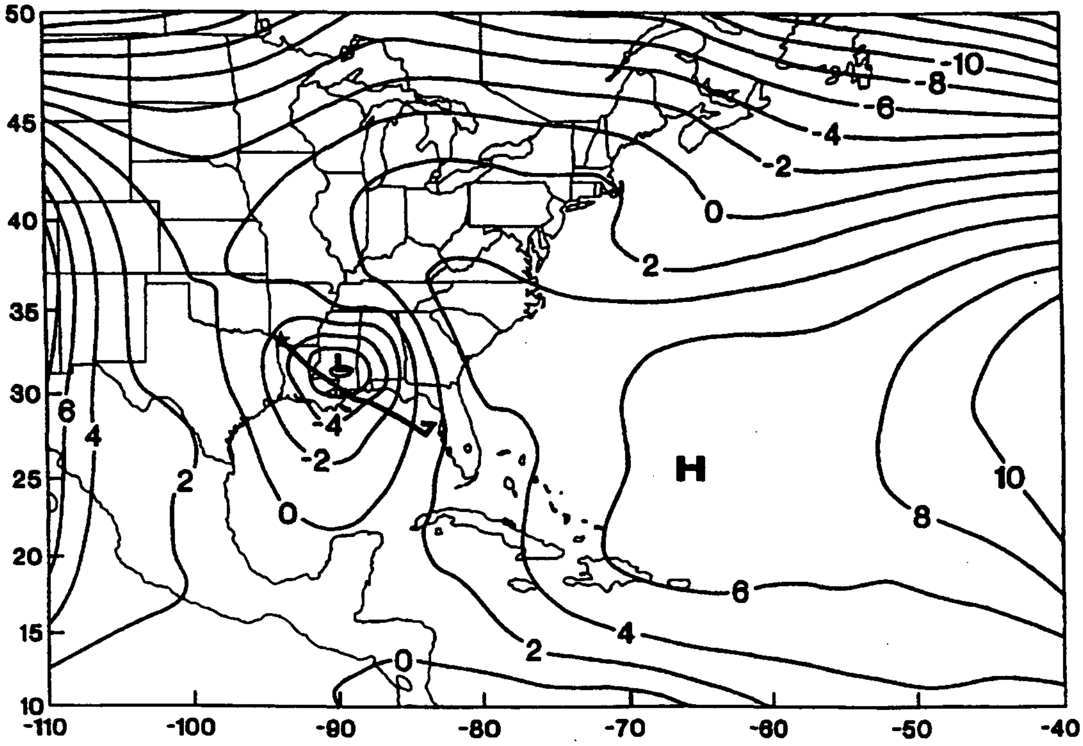

Figure 5 Same as Figure 4 except map is valid at t = 72 h. The observed track of Elena between 0–72 h has been superimposed.

cell in the model is ≈ (100 km)^2. Although the forecasted location of landfall was ≈ 150 km east of the observed location (near Mobile Bay), the time for landfall was accurate. The positioning error was less than the average error for 48 h predictions (NHC, personal communication).

The parallel-barotropic forecast was an improvement over SANBAR because it did not use the "initial storm motion" (ISM) vector as part of the forecast. It has been found that operational forecasts with SANBAR generally benefit from the inclusion of an ISM vector based on the storm's previous path. In cases exhibiting re-curvature, however, incorporation of recent history (persistence) is generally detrimental to the track forecast. Since re-curvature is difficult to anticipate and since it is a relatively rare event, the operational forecasts always include the ISM vector in the initial condition. Examination of the forecast from SANBAR indicates that the imposition of the ISM constraint at t = 0 forced Elena to erroneously move over the northern portion of the Florida peninsula during the early part of the 72 h forecast.

5. SPEED-UP ANALYSIS

Table 2 summarizes the results for our barotropic forecast using the Alliant multi-vector processor. This table illustrates two types of speed-up computations. The speed-up ratio 1 is the ratio with respect to the serial time. Using one vector pro-

Table 2 Speed-Up Ratios for 72 h Forecasting Process

Number of Processors	Elapsed Time (Sec)	Speed-Up Ratio 1	Speed-Up Ratio 2
Serial	216.18	1.0	—
1 Vector	109.47	1.97	1.0
2 Vector	61.03	3.54	1.79
4 Vector	38.82	5.57	2.82
8 Vector	29.53	7.32	3.71

cessor we obtain a speed-up of nearly 2, using 8 vector processors a speed-up of 7.32. In the latter case, there are two factors contributing to the speed-up, namely, parallelism (8 processors) and vectorization in each of the eight processors. In contrast, the speed-up ratio 2 measures the speed-up resulting from multivectors to single vector processing. Thus, eight vector processors result in a speed-up ratio of 3.71. Since vectorization is a common factor in the second ratio, the effect of parallelism on speed-up is isolated.

Referring to the results in Table 2, one might wonder why a speed-up ratio of ≈ 3.7 is achieved when 8 processors are used. In practice, a number of factors affect the speed-up achievable. First, the algorithm itself is a factor. It could happen that there are segments of the computation that are intrinsically serial and cannot be parallelized at all. This will lead to loss of speed-up as dictated by the well-known Amdahl's law (Lakshmivarahan and Dhall, 1990). Referring to the model equations (8) and (9), it can be verified that (refer to Appendix A), while BCR method for solving (9) exhibits "good" parallelism, the degree of parallelism does *not* remain constant, which is an intrinsic property of this class of algorithms. Indeed, the degree of parallelism *decreases* during the course of the reduction phase, and then *increases* in the back-substitution phase. In other words, the processor utilization decreases and increases during the reduction and back-substitution phases respectively. Secondly, stability of numerical integration dictates the use of Adams–Bashford scheme in combination with Arakawa's approximation for the Jacobian. However, the computational process resulting from the use of these schemes do not readily vectorize and/or parallelize. These factors limit the speed-up that is achievable.

Second, the machine architecture is a factor. In the Alliant FX/8, for example, the processors are connected to the main memory through a fast cache memory. The size of this cache limits the data transfer rate between the memory and the "thirsty" processors. Thus, when all eight processors are concurrently working in the vector mode, the data needed by these processors may not be readily available in cache memory which may affect the data transfer rate between memory and the processors. Thus, in such a situation, increase in number of processors results in diminishing returns in terms of the speed-up. Increasing the cache size will only postpone this saturation effect, but on architectures with no cache memory, such as the CRAY research machines, linear increase in speed-up can be achieved.

Thirdly, when one increases the number of processors, unless the problem size is also increased, the given quantum of work is merely smeared across a large number of processors. Oftentimes it may take more time to distribute the data to the processors than to compute the results. This contributes to idling of processors and in turn reduces the processor efficiency, which is speed-up per processor. Since the data set available on Elena was fixed *a priori*, increasing the number of processors would only result in diminishing returns as evidenced in Table 2. Thus, success in parallel computing critically depends on the proper match between the algorithm, the architecture, and the problem size.

6. CONCLUSION

We have taken the simplest deterministic weather prediction model that is still used operationally—the barotropic model—and have brought the maturing technology of parallel processing to bear on its solution. We have applied the computational mechanics to the problem of tracking hurricane Elena (1985).

The results are encouraging from both a pragmatic computational viewpoint as well as a practical weather forecasting viewpoint. A significant saving of computational resources is possible with parallel processing. Essential to the time-saving is the use of parallel direct methods for solving the elliptic equations associated with the barotropic forecast. We have found that the block cyclic reduction (BCR) method using partial fraction expansion is an efficient strategy. As a measure of the efficiency, we calculated the speed-up ratios with respect to both the serial mode and the one vector processor mode and found that a speed-up ratio of 3–4 is possible when an 8 vector processor is compared to a single vector machine.

A complete set of DLM winds for Elena (1985) were derived at the UW-SSEC and used as input to the barotropic model. A decomposition of these winds was accomplished in such a way that the kinetic energy in the nondivergent component was maximized. The streamfunction extracted from the decomposition process was used to initialize the barotropic model and application was made to the period of time when Elena executed a dramatic re-curvature in the eastern Gulf of Mexico. Although the model did not capture the small scale cycloidal motion associated with the abrupt change in Elena's path, the larger-scale redirection from easterly to northwesterly was well handled. Especially encouraging was an accurate 48 h prediction of landfall.

Based on this success, we are prepared to address the problem of coding the adjoint model that can be used in model sensitivity studies and data assimilation. Since the adjoint equations also hinge on the solution to Poisson's equation, the implementation should proceed expeditiously.

Appendix. Solution to Poisson's Equation Using a Parallel Processing Version of Block Cyclic Reduction (BCR)

BCR methods were first discussed by Hockney (1965) and were further developed by Buzbee et al. (1970). However, applicability to large-scale problems was hampered by the presence of serial bottleneck which inhibited the exploitation of full parallelism (Lakshmivarahan and Dhall, 1990). Only recently have methods been developed to overcome this bottleneck (Sweet, 1988; Gallopoulos and Saad, 1989). The classical BCR, when combined with acceleration schemes, gives rise to an algorithm which holds promise for both parallelism and vectorization (Jwo et al., 1992). We have now developed codes for the BCR method that include two basic options: polynomial factorization and the partial fraction expansion. Both versions can accommodate three types of boundary conditions: Dirichlet, Neumann, or periodic. They can also handle an arbitrary $M \times N$ grid as well as special grids with M (or N) $= 2^k - 1$, k an integer (Sweet, 1977; Jwo et al., 1992). Poisson's equation is discretized on a uniform Mercator grid (100 km spacing at latitude 30°N). In the experiments, variables on the staggered grid have the following dimensions (M,N): $u(65,43)$, $v(64,44)$, $\Psi(65,44)$ and $\chi(66,45)$ where M is the index in the east–west direction and N is the corresponding index in the north–south direction.

Poisson's equation on an $M \times N$ grid is represented by

$$\mathbf{Ax} = \mathbf{b}, \tag{A.1}$$

where $\mathbf{A}$ is a *sparse symmetric* matrix. When Dirichlet boundary conditions are used, $\mathbf{A}$ assumes the form:

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} & \mathbf{C} & 0 & 0 & \cdot & \cdot & 0 & 0 & 0 \\ \mathbf{C} & \mathbf{B} & \mathbf{C} & 0 & \cdot & \cdot & 0 & 0 & 0 \\ 0 & \mathbf{C} & \mathbf{B} & \mathbf{C} & \cdot & \cdot & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \mathbf{C} & \mathbf{B} & \mathbf{C} \\ 0 & 0 & 0 & 0 & \cdot & \cdot & 0 & \mathbf{C} & \mathbf{B} \end{bmatrix} \tag{A.2}$$

where there are $(N-2)$ block matrices $\mathbf{B}$ along the principal diagonal of $\mathbf{A}$, and $\mathbf{B}$ is a tridiagonal matrix of order $(M-2) \times (M-2)$. The form of $\mathbf{B}$ is:

$$\mathbf{B} = \begin{bmatrix} -4 & 1 & 0 & 0 & \cdot & \cdot & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & \cdot & \cdot & 0 & 0 & 0 \\ 0 & 1 & -4 & 1 & \cdot & \cdot & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdot & \cdot & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & 0 & 1 & -4 \end{bmatrix} \tag{A.3}$$

and $\mathbf{C}$ is the identity matrix of order $(M-2) \times (M-2)$.

We shall illustrate block cyclic reduction in the case where A is a 7×7 block tridiagonal matrix. The system $\mathbf{Ax} = \mathbf{b}$ can be rewritten as

$$\begin{gathered} \mathbf{Bx}_1 + \mathbf{x}_2 = \mathbf{b}_1 \\ \mathbf{x}_{i-1} + \mathbf{Bx}_i + \mathbf{x}_{i+1} = \mathbf{b}_i \qquad 2 \le i \le 6 \\ \mathbf{x}_6 + \mathbf{x}_7 = \mathbf{b}_7. \end{gathered} \tag{A.4}$$

Consider the sets of three consecutive equations centered around $\mathbf{x}_i$, for $i = 2$, 4 and 6

$$\left.\begin{aligned} \mathbf{x}_{i-2} + \mathbf{B}\mathbf{x}_{i-1}\mathbf{x}_i = \mathbf{b}_{i-1} \\ \mathbf{x}_{i-1}\mathbf{B}\mathbf{x}_i + \mathbf{x}_{i+1} = \mathbf{b}_i \\ \mathbf{x}_i + \mathbf{B}\mathbf{x}_{i+1} + \mathbf{x}_{i+2} = \mathbf{b}_{i+1} \end{aligned}\right\} \qquad i = 2,4,6, \tag{A.5}$$

where $\mathbf{x}_0 = \mathbf{x}_8 = 0$.

Multiplying the second equation by $-\mathbf{B}$ and adding the three equations of the set, we eliminate the odd indexed terms $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i+1}$. These operations can be accomplished in parallel for $i = 2$, 4 and 6, and we obtain

$$\mathbf{x}_{i-2} + (2\mathbf{I} - \mathbf{B}^2)\mathbf{x}_i + \mathbf{x}_{i+2} = \mathbf{b}_{i+1} + \mathbf{b}_{i-1} - \mathbf{B}\mathbf{b}_i, \qquad i = 2,4,6. \tag{A.6}$$

This constitutes the first step of the reduction process and results in the split of the 7 × 7 system into two subsystems—one for odd indexed $\mathbf{x}_i$'s called the *eliminated* system and the other for even indexed $\mathbf{x}_i$'s called the *reduced* system

$$\begin{aligned} \mathbf{B}(1)\mathbf{x}_2 + \mathbf{x}_4 &= \mathbf{b}_2 \\ \mathbf{x}_2 + \mathbf{B}(1)\mathbf{x}_4 + \mathbf{x}_6 &= \mathbf{b}_4 \\ \mathbf{x}_4 + \mathbf{B}(1)\mathbf{x}_6 &= \mathbf{b}_6 \end{aligned} \tag{A.7}$$

is the reduced system where $\mathbf{B}(1) = 2\mathbf{I} - \mathbf{B}(0)^2$, $(\mathbf{B}(0) = \mathbf{B})$, which is a matrix polynomial in $\mathbf{B}$ of degree 2 and $\mathbf{b}_i(1) = \mathbf{b}_{i+1} + \mathbf{b}_{i-1} - \mathbf{B}(0)\mathbf{b}_i$, for $i = 2$, 4 and 6. The eliminated system is given by

$$\begin{bmatrix} \mathbf{B}(0) & 0 & 0 & 0 \\ 0 & \mathbf{B}(0) & 0 & 0 \\ 0 & 0 & \mathbf{B}(0) & 0 \\ 0 & 0 & 0 & \mathbf{B}(0) \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_3 \\ \mathbf{x}_5 \\ \mathbf{x}_7 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 - \mathbf{x}_2 \\ \mathbf{b}_3 - \mathbf{x}_2 - \mathbf{x}_4 \\ \mathbf{b}_5 - \mathbf{x}_4 - \mathbf{x}_6 \\ \mathbf{b}_7 - \mathbf{x}_6 \end{bmatrix}. \tag{A.8}$$

Thus solving (A.7) for $\mathbf{b}_2$, $\mathbf{b}_4$, and $\mathbf{b}_6$, we can readily solve for $\mathbf{x}_1$, $\mathbf{x}_3$, $\mathbf{x}_5$ and $\mathbf{x}_7$ from the eliminated system. Notice (A.7) is again a block tridiagonal system and we can once again multiply the second equation in (A.7) by $-\mathbf{B}(1)$ and add the equations together to eliminate $\mathbf{x}_2$ and $\mathbf{x}_6$. The new reduced system is

$$\mathbf{B}(2)\mathbf{x}_4 = \mathbf{b}_4(2), \tag{A.9}$$

where

$$\mathbf{B}(2) = 2\mathbf{I} - [\mathbf{B}(1)]^2,$$

which is a matrix polynomial in $\mathbf{B}$ of degree 4 and

$$\mathbf{b}_4(2) = \mathbf{b}_2(1) + \mathbf{b}_6(1) - \mathbf{B}(1)\mathbf{b}_4(1).$$

The new eliminated system is

$$\begin{bmatrix} \mathbf{B}(1) & 0 \\ 0 & \mathbf{B}(1) \end{bmatrix} \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{x}_6 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_2(1) - \mathbf{x}_4 \\ \mathbf{b}_6(1) - \mathbf{x}_4 \end{bmatrix}. \tag{A.10}$$

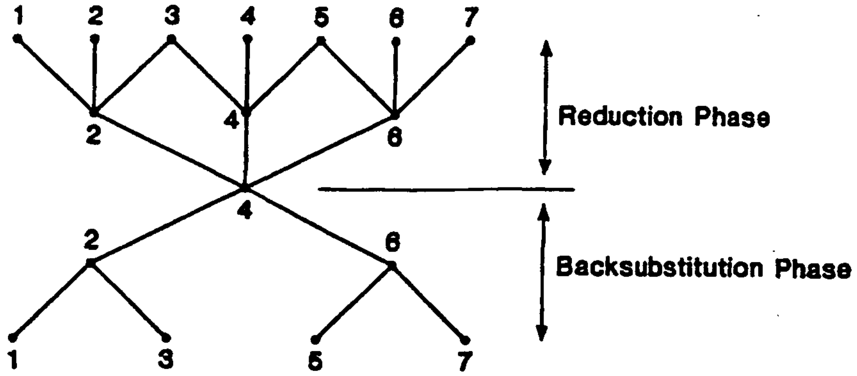

Figure 6 Pictorial display of the block cyclic reduction (BCR) process for a linear system of 7 equations where each node in the diagram represents an equation in one of the variables: $X_1, X_2, \ldots, X_7$.

This algorithm is diagrammatically illustrated in Figure 6. It consists of two phases. In the first step of the reduction phase, eliminate $\mathbf{x}_1$, $\mathbf{x}_3$, $\mathbf{x}_5$, and $\mathbf{x}_7$ in parallel to get a reduced system in $\mathbf{x}_2$, $\mathbf{x}_4$, and $\mathbf{x}_6$. In the second step eliminate $\mathbf{x}_2$ and $\mathbf{x}_6$ in parallel to obtain the reduced system in $\mathbf{x}_4$. By solving this system we can recover other unknowns in the back-substitution phase. Knowing $\mathbf{x}_4$, $\mathbf{x}_2$ and $\mathbf{x}_6$ are recovered first in parallel using (A.10) and then from (A.8), $\mathbf{x}_1$, $\mathbf{x}_3$, $\mathbf{x}_5$, and $\mathbf{x}_7$ are recovered in parallel.

Computationally, the back substitution involves solving

$$\mathbf{B}(2)\mathbf{x}_4 = \mathbf{b}_4(2) \tag{A.11}$$

$$\mathbf{B}(1)\mathbf{x}_i = \mathbf{b}_i(1) - \mathbf{x}_4, \qquad \text{for} \quad i = 2, 6, \tag{A.12}$$

$$\mathbf{B}(0)\mathbf{x}_i = \mathbf{b}_i - \mathbf{x}_{i+1} - \mathbf{x}_{i-1} \tag{A.13}$$

for $i = 1, 3, 5$, and 7, where $\mathbf{x}_0 = \mathbf{x}_8 = 0$.

Consider the system in (A.11). From $\mathbf{B}(2) = 2\mathbf{I} - [\mathbf{B}(1)]^2 = 2\mathbf{I} - (2\mathbf{I} - [\mathbf{B}(0)]^2)^2$, it follows that computation of $\mathbf{B}(2)$ involves expensive matrix multiplications. Besides, while $\mathbf{B}(0)$ is tridiagonal, $[\mathbf{B}(0)]^2$ is not tridiagonal. Thus $\mathbf{B}(1)$ and $\mathbf{B}(2)$ do not inherit the sparse structure of $\mathbf{B}(0)$. Thus solving (A.11) by directly computing $\mathbf{B}(2)$ is not computationally worthwhile. An alternate approach is to exploit the structure of the matrix polynomial and to cleverly factor it so that each factor inherits the tridiagonal structure of $\mathbf{B}$ which can then be exploited in computation. To this end, define a class of polynomials recursively as follows:

$$p_{2^{r+1}}(a,t) = 2t^{2^{r+1}} - [p_{2^r}(a,t)]^2 \tag{A.14}$$

where

$$p_2(a,t) = 2t^2 - a^2.$$

$p_{2^r}(a,1)$ is known as the Chebyshev polynomial of the first kind and degree 2^r. The importance of this class of polynomials stems from the fact that

$$\mathbf{B}(r) = p_{2^r}(a,t)\,|_{a=\mathbf{B}, t=\mathbf{I}}\,.$$

Thus, $\mathbf{B}(1) = 2\mathbf{I} - \mathbf{B}^2$. By setting $a = -2t\cos\theta$, it can be shown that

$$p_{2^r}(a,t) = 2t^{2^r}\cos 2^r\theta.$$

Clearly,

$$p_{2^r}(a,t) = 0,$$

if

$$\frac{a}{2t} = -\cos\left(\frac{2j-1}{2^{r+1}}\right)\pi, \qquad j = 1,2,\ldots,2^r.$$

Using this we obtain a factorization

$$p_{2^r}(a,t) = -\prod_{j=1}^{2^r}(a + 2t + \cos\theta_j(r)),$$

where

$$\theta_j(r) = \left(\frac{2j-1}{2^{r+1}}\right)\pi.$$

The corresponding factorization for $\mathbf{B}(r)$ is given by

$$\mathbf{B}(r) = -\prod_{j=1}^{2^r}\mathbf{H}_j(r),$$

where

$$\mathbf{H}_j(r) = (\mathbf{B} + 2\mathbf{I}\cos\theta_j(r)).$$

Notice $\mathbf{H}_j(r)$ is tridiagonal since $\mathbf{B}$ is tridiagonal and $2\mathbf{I}\cos\theta_j(r)$ is diagonal.

Given this factorization, there are two possibilities for solving systems of the type $\mathbf{B}(r)\mathbf{y} = \mathbf{d}$ for $r = 0,1,2,\ldots$. We illustrate this using (A.11).

Recall

$$\mathbf{B}(2) = -\mathbf{H}_1(2)\mathbf{H}_2(2)\mathbf{H}_3(2)\mathbf{H}_4(2).$$

Let $\mathbf{Z}_0 = \mathbf{b}_4(2)$. Then (A.11) becomes

$$\mathbf{H}_1(2)\mathbf{H}_2(2)\mathbf{H}_3(2)\mathbf{H}_4(2)\mathbf{x}_4 = \mathbf{Z}_0.$$

Solve

$$\mathbf{H}_i(2)\mathbf{Z}_i = \mathbf{Z}_{i-1}$$

successively for $i = 1$, 2, 3 and 4. Clearly $\mathbf{x}_4 = \mathbf{Z}_r$. The advantage of this approach called the *polynomial factorization* is that each $\mathbf{H}_i(2)$ is sparse. Yet the collection of systems is solved sequentially which inhibits parallelism.

In search of a method that admits more parallelism, rewrite (A.11) as

$$\begin{aligned}\mathbf{x}_4 &= [\mathbf{B}(2)]^{-1}\mathbf{b}_4(2)\\ &= [p_{2^r}(a,t)]^{-1}_{a=\mathbf{B},t=1}\mathbf{b}_4(2).\end{aligned}$$

Expressing $[p_{2^r}(a,t)]^{-1}$ in partial fraction expansion we obtain

$$\mathbf{x}_4 = [\alpha_1\mathbf{H}_1^{-1}(2) + \alpha_2\mathbf{H}_2^{-1}(2) + \alpha_3\mathbf{H}_3^{-1}(2) + \alpha_4\mathbf{H}_4^{-1}(2)]\mathbf{b}_4(2).$$

In this expression each of the terms on the right hand side can be independently computed in parallel. Swarztrauber (1984) and Lakshmivarahan and Dhall (1990) can be consulted for further elaboration on the steps involved in parallelization of this algorithm.

Acknowledgments

We received partial support for this work from the Center for the Analysis and Prediction of Storms (CAPS) at the University of Oklahoma, and Argonne National Laboratory generously allowed us to use the Alliant FX/8. Drs. Mark DeMaria (AOML/ERL), Bob Davies–Jones (NSSL/ERL), and Dave Stensrud (NSSL/ERL) showed genuine interest in this research and offered cogent suggestions which strengthened the contribution. Competent technical assistance from Mr. Dave Keller (computer graphics) and Ms. Joy Walton (typing) is gratefully acknowledged.

References

A. Arakawa, Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow, Part 1, *J. Computational Physics* **1** (1966), 119–143.

D. A. Anderson, J. C. Tannehill and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Pub. Co., (1984), 599 pp.

S. J. Bijlsma, L. M. Hafkenscheid and P. Lynch, Computation of the stream function and velocity potential and reconstruction of the wind field, *Mon. Wea. Rev.* **114** (1986), 1547–1551.

B. L. Buzbee, G. H. Golub and C. W. Nielson, On direct methods for solving Poisson's equation, *SIAM J. Numerical Analysis* 7 (1970), 627–655.

R. Davies-Jones, On the formulation of surface geostrophic streamfunction, *Mon. Wea. Rev.* **116**, 9 (1988), 1824–1826.

M. DeMaria, Tropical cyclone motion in a nondivergent barotropic model, *Mon. Wea. Rev.* **113** (1985), 2346–2357.

———, Tropical cyclone track prediction with a barotropic spectral model, *Mon. Wea. Rev.* **115** (1987), 1199–1210.

——— and R. W. Jones, Optimization of a hurricane track forecast model using the adjoint model equations, *Ninth Conf. on Num. Wea. Pred.*, Amer. Meteor. Soc. (1991), 547–550.

E. Gallopoulos and Y. Saad, A parallel block cyclic reduction algorithm for the fast solution of elliptic equations, *Parallel Computing* **10** (1989), North-Holland, 143–159.

G. J. Haltiner and R. T. Williams, *Numerical Prediction and Dynamic Meteorology*, 2nd ed. John Wiley and Sons (1980), 477 pp.

R. W. Hockney, A fast direct solution of Poisson's equation using Fourier analysis, *J. Assoc. Comp. Mach. (JACM)* **12** (1965), 95–113.

J. Jwo, S. Lakshmivarahan, S. K. Dhall and J. M. Lewis, Comparison of the performance of three parallel version of the block cyclic reduction algorithm for solving linear elliptic partial differential equations, *Computers Math. Applic.* **24**, 5/6 (1992), 83–101.

S. Lakshmivarahan and S. K. Dhall, *Analysis and design of Parallel Algorithms*, McGraw Hill, see Ch. 5, 1990.

J. M. Lewis and J. C. Derber, The use of adjoint equations to solve a variational adjustment problem with advective constraints, *Tellus* **37A** (1985), 885–893.

———, A. Van Tuyl and C. Velden, A dynamical method for building continuity into the deep layer mean wind, *Mon. Wea. Rev.* **115** (1987), 885–893.

C. E. Leith, Objective methods for weather prediction, *Ann. Rev. Fluid Mech.* **10** (1978), 107–128.

P. Lynch, Deducing the wind from vorticity and divergence, *Mon. Wea. Rev.* **116** (1988), 86–93.

———, Partitioning the wind in a limited domain, *Mon. Wea. Rev.* **117** (1989), 1492–1500.

F. Sanders, A. C. Pike and J. P. Gaertner, A barotropic model for operational prediction of tracks of tropical storms, *J. Appl. Meteor.* **14** (1975), 265–280.

——— and R. W. Burpee, Experiments in barotropic hurricane track forecasting, *J. Appl. Meteor.* **7** (1968), 313–323.

W. E. Sangster, A method of representing the horizontal pressure force without reduction of pressure to sea level, *J. Meteor.* **17** (1960), 166–176.

Y. Sasaki and K. Miyakoda, Numerical forecasting of the movement of cyclone, *J. Meteor. Soc. Japan* **32** (1954), 9–19.

W. J. Saucier, *Principles of Meteorological Analysis*, Univ. of Chicago Press (1955), 438 pp.

L. J. Shapiro and K. V. Ooyama, Barotropic vortex evolution on a beta-plane, *J. Atmos. Sci.* **47** (1990), 170–187.

P. A. Swarztrauber, Fast Poisson solvers in *Studies in Numerical Analysis* (G. H. Golub, ed.), *MAA Studies in Mathematics* **24** (1984), 319–370.

R. A. Sweet, A cyclic reduction algorithm for solving block tridiagonal systems of arbitrary dimension, *SIAM J. on Numerical Analysis* **143** (1977), 706–720.

———, A parallel and vector variant of the cyclic reduction algorithm, *SIAM J. on Scientific and Statistical Computing* **9** (1988), 761–765.

O. Talagrand and P. Courtier, Variational assimilation of meteorological observations with the adjoint vorticity equation, Part I: Theory, *Quart. J. Roy. Meteor. Soc.* **113** (1987), 1311–1328.

C. S. Velden, Satellite observations of hurricane Elena (1985) using the VAS 6.7 μm "water vapor" channel, *Bull. Amer. Meteor. Soc.* **68** (1985), 210–215.

———, W. L. Smith and M. Mayfield, Application of VAS and TOVS to tropical cyclones, *Bull. Amer. Meteor. Soc.* **65** (1984), 1059–1067.

———, C. M. Hayden, W. P. Menzel, J. L. Franklin and J. S. Lynch, The impact of satellite-derived winds on numerical hurricane track forecasting, *ul Weather and Forecasting* (1992), (to appear).