

## The Computational Complexity and Parallel Scalability of Atmospheric Data Assimilation Algorithms

P. M. LYSTER,<sup>\*</sup> J. GUO,<sup>+</sup> T. CLUNE, AND J. W. LARSON<sup>#</sup>

*Global Modeling and Assimilation Office, Laboratory for Atmospheres, NASA Goddard Space Flight Center, Greenbelt, Maryland*

(Manuscript received 1 August 2003, in final form 15 December 2003)

### ABSTRACT

This paper quantifies the computational complexity and parallel scalability of two algorithms for four-dimensional data assimilation (4DDA) at NASA's Global Modeling and Assimilation Office (GMAO). The first, the Goddard Earth Observing System Data Assimilation System (GEOS DAS), uses an atmospheric general circulation model (GCM) and an observation-space-based analysis system, the Physical-Space Statistical Analysis System (PSAS). GEOS DAS is very similar to global meteorological weather forecasting data assimilation systems but is used at NASA for climate research. The second, the Kalman filter, uses a more consistent algorithm to determine the forecast error covariance matrix than does GEOS DAS. For atmospheric assimilation, the gridded dynamical fields typically have more than  $10^6$  variables; therefore, the full error covariance matrix may be in excess of a teraword. For the Kalman filter this problem will require petaflop  $s^{-1}$  computing to achieve effective throughput for scientific research.

### 1. Four-dimensional data assimilation

Four-dimensional data assimilation (4DDA) is the process of combining observations with a dynamical model to generate a gridded best estimate, or analysis, of the state of the system (Daley 1991). It is thus a mapping problem, whereby scattered observations are converted into accurate maps of wind, temperature, moisture, and other variables. Figure 1.11 of Daley (1991) shows a schematic of the data assimilation cycle. The model propagates in time the estimate of the state; for example, for the global atmosphere we use a general circulation model (GCM). The analysis is a statistics-based algorithm for combining the model output, or forecast, with observations to produce the best-estimate state. This is a cycled algorithm whereby the analysis state is used to reinitialize the model, and so on. 4DDA is used in weather forecasting to initialize model forecasts, for example, at the National Centers for Environmental Prediction (NCEP; Parrish and Derber 1992;

Parrish et al. 1997) and at the European Centre for Medium-Range Weather Forecasts (ECMWF; Courtier et al. 1998; Rabier et al. 1998; Andersson et al. 1998). 4DDA is also used to perform reanalyses of past datasets to obtain consistent, gridded, best estimates of the state variables of the atmosphere (e.g., wind, temperature, moisture, etc.), for example, at the National Aeronautics and Space Administration (NASA) Global Modeling and Assimilation Office (GMAO; Schubert et al. 1993, 1995), at NCEP (Kalnay et al. 1996; Kistler et al. 2001), and at ECMWF (Gibson et al. 1997). These gridded reanalysis datasets are a valuable resource for the earth science research community (GMAO 2000).

This paper quantifies the computational complexity and the scalability of distributed-memory parallel implementations of two algorithms for 4DDA at the GMAO. The first is the Goddard Earth Observing System Data Assimilation System (GEOS DAS), which uses a gridpoint-based atmospheric GCM and an observation-space-based analysis system, the Physical-Space Statistical Analysis System (PSAS). GEOS DAS is very similar to global weather forecasting algorithms, where the analysis fields are used to initialize the GCM for a model forecast. Operational global 4DDA systems such as GEOS DAS with model grids of the order  $10^5$  m have about  $10^6$  variables.<sup>1</sup> We also present a timing profile of the baseline-resolution GEOS-2 DAS on computers with 8 or 10 processors using shared-memory

<sup>\*</sup> Additional affiliation: Earth System Science Interdisciplinary Center, and Department of Meteorology, University of Maryland, College Park, College Park, Maryland.

<sup>+</sup> Additional affiliation: Science Applications International Corporation/General Sciences Operation, Beltsville, Maryland.

<sup>#</sup> Additional affiliation: Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois.

*Corresponding author address:* Dr. Peter M. Lyster, National Institutes of Health, Bethesda, MD 20892.  
E-mail: lysterpe@nigms.nih.gov

<sup>1</sup> More recent versions of GEOS DAS have a horizontal resolution of  $5 \times 10^4$  and  $2.5 \times 10^4$  m.

parallelism. The second algorithm is the Kalman filter, which offers the promise of more accurate analyses because it evolves error statistics in a dynamically consistent manner. However, the full error covariance matrix is of dimension the square of the number of model state variables, so the algorithm will require petaflop per second computing to achieve effective throughput for scientific research. A two-dimensional (latitude–longitude) Kalman filter for the assimilation of constituent gas mixing ratio in the stratosphere was developed by our group as a prototype and research tool (Lyster et al. 1997a; Ménard et al. 2000; Menard and Chang 2000). Some of the results of this work are used to extrapolate to the complexity of a full Kalman filter with three-dimensional meteorological fields.

The results presented here are of interest to scientific software developers who make compromises between algorithmic improvements or approximations on the one hand and computer resources on the other. They also are a useful starting point for computer administrators who make strategic decisions about computer architecture, maintenance, and purchases based on many parameters, including future estimates of the problem size (e.g., the resolution of the model and the number of observations to be assimilated) and the performance and parallel scalability of the algorithms. Section 2 starts with an overview of the GEOS DAS. The discussion on complexity provides estimates of floating-point operations of the components of the data assimilation systems. Specific results are provided for the version 2, GEOS-2 DAS, which was the main production system in use at the GMAO in the late 1990s. It consists of GEOS-2 GCM with resolution 2.5° latitude, 2° longitude, and 70 vertical levels, and GEOS-2 PSAS, which assimilated about 50 000 observations per 6 h; details of these algorithms are given in section 2a,b. Section 2c provides timing profiles of the baseline production system GEOS-2 DAS that used shared-memory parallelism. Section 3 presents the algorithm for the Kalman filter and describes the two-dimensional distributed-memory parallel implementation that was used for scientific and computational research. The GCM and PSAS have tightly coupled core algorithms with computation- and communication-intensive parallel implementations: these are hydrodynamic transport (GCM) and nonsparse large matrix–vector multiplications (PSAS). Section 4 discusses technical issues and limitations in developing scalable distributed-memory parallel implementations of the GCM and PSAS, and then extends the discussion to GEOS DAS.

## 2. Goddard Earth Observing System Data Assimilation System (GEOS DAS)

Derivations of analysis algorithms abound (Daley 1991). We motivate briefly and derive the analysis equations for GEOS DAS based on a statistical least squares approach. Cohn (1997) places this discussion in the con-

text of general filtering methods. The optimal estimate of the state is the value of the control variable  $\mathbf{w}$  that minimizes the cost function  $J$ :

$$J(\mathbf{w}) = \frac{1}{2}[(\mathbf{w}^f - \mathbf{w})^T(\mathbf{P}^f)^{-1}(\mathbf{w}^f - \mathbf{w}) + (\mathbf{w}^o - \mathbf{H}\mathbf{w})^T \times \mathbf{R}^{-1}(\mathbf{w}^o - \mathbf{H}\mathbf{w})], \quad (1)$$

where

- $\mathbf{w}$  is the control vector of  $n$  state variables;
- $\mathbf{w}^f$  is the forecast vector of  $n$  state variables;
- $\mathbf{w}^o$  is the vector of  $p$  observations;
- $\mathbf{P}^f$  is the  $(n \times n)$  given forecast error covariance matrix;
- $\mathbf{R}$  is the  $(p \times p)$  given observation error covariance matrix; and
- $\mathbf{H}$  is the (here linearized) forward operator that models the observations by acting on the state vector (e.g., if the observations come from direct measurements of the state, then  $\mathbf{H}$  can be implemented by interpolation from the state grid to the observation locations).

The value of  $\mathbf{w}$  that minimizes  $J$  is the analysis state:

$$\mathbf{w}^a = \mathbf{w}^f + \mathbf{K}(\mathbf{w}^o - \mathbf{H}\mathbf{w}^f), \quad (2)$$

where the Kalman gain is

$$\mathbf{K} = \mathbf{P}^f \mathbf{H}^T (\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R})^{-1}. \quad (3)$$

GEOS-2 DAS uses a 6-h assimilation window [0, 6 h] for the analysis cycle that is shown schematically in Fig. 1.11 of Daley (1991). Starting from a prior analysis, the GCM generates a forecast by iterating a time-stepping algorithm:

$$\mathbf{w}_{k+1}^f = \mathcal{M}_k \mathbf{w}_k^f, \quad (4)$$

where  $k$  is a time index and  $\mathcal{M}_k$  is the model operator. By convention (e.g., Daley 1991; GMAO 2000), the forecast for each data assimilation cycle ends at (0000, 0600, 1200, 1800) UTC. GEOS-2 DAS evaluates Eq. (2) for each of these 6-h forecasts using data that are accumulated  $\pm 3$  h (i.e., evenly) about the forecast time. Operational algorithms at weather centers and laboratories (Daley 1991) have more constraints and attributes than the simple form of Eqs. (1), (2), and (3). Indeed, GEOS-2 DAS uses a slight modification of the cycling method just described, which involves 9 h of model iteration for each 6-h data assimilation cycle (Bloom et al. 1996). However, these caveats do not substantially modify the evaluation of computational complexity and parallel scalability in the present work.

### a. The computational algorithm for GEOS DAS

We describe the complexity and timing profile for a baseline version 2, GEOS-2 DAS, which was the main production system in use at the GMAO in the late 1990s. The GEOS-2 GCM (Takacs et al. 1994) comprises a spatial fourth-order-accurate finite-difference dynamical

core to model hydrodynamical processes, plus physics components for moist convection, turbulence, and shortwave and longwave radiation. The state, or prognostic, variables are horizontal winds, potential temperature, specific humidity, and surface pressure. A high-latitude spectral filter and a global Shapiro filter and polar rotation algorithm provide smoothing and numerical stability. The GEOS-2 GCM used a model resolution of  $2.5^\circ$  latitude,  $2^\circ$  longitude, and 70 vertical levels. This corresponds to three-dimensional fields with horizontal resolution 91 grid points in latitude and 144 grid points in longitude. The GEOS-2 GCM uses a multiple-time-scale computational technique (Brackbill and Cohen 1985). The dynamical core has the smallest time step of 3 min at baseline resolution. The physics components generate time tendencies at longer intervals: moist convection 10 min, turbulence 30 min, shortwave radiation 1 h, and longwave radiation 3 h. These tendencies are applied to the state variables incrementally at the shortest time scale (3 min). Fuller details are described in Takacs et al. (1994), and the next sections will discuss the complexity and timing profile of the GCM in the context of the whole data assimilation system. The number of state variables at the baseline resolution is approximately  $n \approx 3 \times 91 \times 144 \times 70 + 91 \times 144 \approx 2.6 \times 10^6$ , corresponding to the 3 upper-air (i.e., three-dimensional) field arrays and 1 surface (i.e., two-dimensional) field array, although in practice up to 14 upper-air field arrays are carried by the algorithm.

Currently, the GCM is run with  $1^\circ \times 1^\circ \times 48$  levels, and developmental versions achieve even higher resolution. An extensive land surface model with associated prognostic variables has also been implemented in the GCM, but we will not include that in the baseline numbers. The actual resolution is not critical to this paper, which discusses scaling properties starting from the baseline resolution of the GEOS-2 DAS. Note also that this is not the same model as the finite-volume fvGCM that is used for current-generation data assimilation systems at the GMAO. Between these two GCMs some general quantities, such as asymptotic scalability, may be similar, but specific values of quantities like the model time step or wall-clock time of runs are different.

The algorithm for solving Eq. (2), that is, the analysis in Fig. 1.11 of Daley (1991), is the PSAS (Cohn et al. 1998). This solves

$$(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})\mathbf{x} = \mathbf{w}^o - \mathbf{H}\mathbf{w}^f, \quad \text{and} \quad (5)$$

$$\mathbf{w}^a - \mathbf{w}^f = \mathbf{P}^f\mathbf{H}^T\mathbf{x}. \quad (6)$$

The time subscript  $k$  will be dropped where it is not important to the discussion. The right-hand side of Eq. (5) is sometimes called the “observed minus forecast residual” or the “innovation,” and  $\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R}$  is called the “innovation matrix.” To generate the analysis fields at the end of each 6-h cycle, GEOS-2 DAS adds the “analysis increment”  $\mathbf{w}^a - \mathbf{w}^f$  incrementally to the state variables in a similar way as the physics tendencies are

applied as described above (Takacs et al. 1994; Bloom et al. 1996). The error covariance matrices  $\mathbf{P}^f$  and  $\mathbf{R}$  are implemented using models for variances and correlations whose parameters are obtained from prior statistics and simplifying assumptions such as stationarity (Daley 1991; GMAO 2000). Sophisticated multivariate formulations of error covariances are used to improve the quality of the analysis (Guo et al. 1998). Although this has a significant impact on the software complexity (Larson et al. 1998) it has only a secondary impact on the computational complexity and will not be considered here. The resulting matrices  $\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R}$  and  $\mathbf{P}^f\mathbf{H}^T$  are in principle dense; however, correlation models with compact support (Gaspari and Cohn 1999) are used, which reduces the computational complexity by setting the correlation to zero beyond a fixed length. As described above, Eqs. (5) and (6) are solved for data that are aggregated over the [0, 6 h] analysis cycle. This interval will be shortened to make better use of synoptic observations (e.g., retrievals from satellites) and accommodate shorter temporal and spatial scales of high-resolution GCMs, but the numbers in this paper refer to baseline GEOS-2 DAS with a 6-h analysis cycle. The PSAS consists of solving one  $p \times p$  linear system [Eq. (5)] for the intermediate vector  $\mathbf{x}$  using a parallel nested-preconditioned conjugate gradient solver [Cohn et al. (1998); Golub and van Loan (1989); The technical documents for the PSAS are da Silva and Guo (1996), Guo et al. (1998), and Larson et al. (1998)]. Machine-precision solutions for  $\mathbf{x}$  are not required because the error in the analysis field  $\mathbf{w}^a$  is only required to be consistent with the error covariance  $\mathbf{P}^a$  (Cohn et al. 1998). For the baseline GEOS-2 DAS during the late 1990s there were typically  $p \approx 5 \times 10^4$  observations worldwide in each 6-h period. From experience, we found that  $\mathcal{N}_i \approx 10$  iterations of the outer loop of the solver provides a satisfactory solution; this reduces the residual of the solver by about an order of magnitude.

The GEOS-2 DAS was run in a number of production modes (Stobie 1996). These may be generally categorized as real-time, near-real-time, and reanalysis modes. Real-time mode requires model forecast and analyses to take place sufficiently in excess of 1 day of assimilation per wall-clock day so that the results may be studied and disseminated to customers such as satellite instrument teams with real-time needs. Reanalyses are multiyear studies designed to provide long-term datasets from a frozen scientific software configuration. For example, the GMAO has completed a reanalysis for the years 1979–95 using the version GEOS-1 DAS (Schubert et al. 1993, 1995).

The data acquisition and storage system for 4DDA involves a worldwide instrumentation, telecommunication, databasing, computational, and administrative effort (Atlas 1997). We remark here only on the attributes and numbers that are relevant to the present work. In the last 60 yr about 2 billion observations that are appropriate for input to atmospheric data assimilation

TABLE 1. GEOS-2 DAS system performance and throughput.

Base line GEOS-2 DAS: $2.5^\circ \times 2^\circ \times 70$ level GCM resolution; 200 000 obs day <sup>-1</sup>	
Net throughput is 5 assimilation days (wall-clock day) <sup>-1</sup> using multitasking parallelism run on eight processors of the Origin 2000.	
Main memory (GB)	2.2 (per image)
Disk (GB)	8.0
Mass storage (GB)	2300.0 (this is output per year)
Volume of data (GB)	6.2 (produced per day per image)
Gflop s <sup>-1</sup> sustained	0.25 (per image)
Duration of run	5 days (wall-clock day) <sup>-1</sup> (continuous operation, single image)

systems have been accumulated. The volume of these data does not present the greatest computational complexity, and operational centers are more concerned with the accuracy of these data. Considerable energy is devoted to finding and validating old observations, that is, “data rehabilitation.” In the coming years, diverse new data types will be made available for data assimilation, and the volume and complexity of the data handling system will increase considerably. For example, satellite sea surface wind observations have been shown to be useful in increasing forecast accuracy of weather analyses (Atlas et al. 1996). The GMAO will also assimilate an increasing amount of nonmeteorological data, such as trace gas concentration in the atmosphere. During the late 1990s, when GEOS-2 DAS was the main operational data assimilation algorithm at the GMAO, about  $10^5$  observations were produced daily under the World Weather Watch and transmitted to worldwide weather centers and the GMAO via the Global Telecommunications System, which is under the supervision of the World Meteorological Organization (Atlas 1997). More than 70% of these were obtained from satellite measurements, mostly as temperature retrievals; the remaining were from in situ balloon-borne and land and sea surface instruments. Table 1 summarizes the baseline GEOS-2 DAS system performance and throughput. GEOS-2 DAS used shared-memory multitasking parallelism and ran on Cray J90/C90 and SGI Origin 2000 computers. At baseline resolution for the GEOS-2 GCM ( $2.5^\circ \times 2^\circ \times 70$  levels), a day of assimilation produced in excess of 1 GB of data. Hence data assimilation at real time (1 day of assimilation per wall-clock day) did not stretch the local disk capacity or bandwidth of most modern computer systems. However, extended runs at higher throughput than real time increases the burden on storage and data processing. The most severe challenge is for reanalysis projects in which multiyear datasets are analyzed by a fixed-version DAS and the products are made available to the scientific community. The standard benchmark is a rate of 30 days of assimilation per day of wall-clock time (i.e., a 15-yr reanalysis on order half a year). At this rate the GEOS-2 DAS produced about 10 TB of data per year.

#### b. The computational complexity of GEOS DAS

Where appropriate, estimates of actual floating-point counts are calculated. However, where this is too dif-

ficult or vague we simply specify the scaling. The computational complexity of different algorithms cannot be compared without careful specification of the spatiotemporal problem domains. In this paper we will state when we use two or three spatial dimensions. We use the notation  $[0, T]$  to specify a fixed simulation time interval. Beyond these, the computational complexity depends on a combination of numerical and physical parameters, including the number of state variables in the model ( $n$ ), the number of observations in an assimilation cycle ( $p$ ), as well as numerical parameters defined in the text.

For the GEOS-2 GCM we specify separately the number of grid points in the longitude, latitude, and vertical coordinates as  $N_x$ ,  $N_y$ , and  $N_z$ , respectively (i.e.,  $n \sim N_x N_y N_z$ ; we indicate here only proportionality because  $n$  includes the total number of field types—wind, height, surface pressure, and moisture—factored into the total number of gridpoints). The complexity of all four of the dynamics, moist convection, turbulence, and radiation components scale as  $N_x N_y$ . In any fixed interval  $[0, T]$  the complexity of the dynamics has an additional dependence on the number of time steps. Generally the number of time steps of the dynamics, that is, the temporal resolution, increases in proportion to the horizontal resolution,  $N_x$ . Also, as the update interval of the physics components is shortened there will be an additional impact on complexity (L. L. Takacs 1997, personal communication). The complexity of the dynamics, moist convection, and turbulence components scale as  $N_z$ , while the radiation scales as  $N_z^2$ . As the horizontal resolution is increased and the concomitant number of dynamics time steps in a fixed simulation interval is increased, the complexity of the dynamics dominates the other components. Asymptotically, for a fixed simulation interval the complexity of the dynamics scales as  $n^{4/3}$ . Thus, if the resolution of the GCM is doubled in all three dimensions the complexity of the dynamics increases 16-fold. The memory requirement for the GCM scales as  $n$ ; thus, the memory requirement in general scales less rapidly than the computational complexity. These asymptotic calculations help specify the size of computing requirements in a 10-yr or longer time frame; however, they can be misleading when applied to real developmental or production software in use today where, for example, there may be parameter regimes where the time step does not need to be reduced in

proportion to the horizontal resolution. In this case, it is important to instrument and generate timing profiles of the algorithms. The next section will present the timing profile for the GEOS-2 DAS and its components.

For the GEOS-2 PSAS, the solver [Eq. (5)] has complexity  $f\mathcal{N}_i s p^2$ , where  $s \approx 0.40$  is the density (fraction of nonzero elements) of the innovation matrix resulting from the use of a correlation function with compact support of  $6 \times 10^6$  m. The factor  $f$  equals 2 plus the number of floating-point operations required to form each element of the matrix. The GEOS-2 PSAS calculates the matrix elements using precalculated lookup tables at each iteration of the outermost loop during the conjugate gradient iteration cycle. This reduces the overall memory requirement and allows for scalability to larger numbers of observations beyond the current values (Guo et al. 1998; Larson et al. 1998). Therefore,  $f$  may be as high as 10, but the exact value depends on the optimization of the access to the tables (Lyster et al. 2000a). The complexity of the preconditioners is neglected here because the preconditioners involve sparse matrix–vector operations compared with the full matrix solver. Equation (6) evaluates the analysis increment, and this has complexity  $f s n p$ . The analysis increment is evaluated on a  $2.5^\circ \times 2^\circ \times 14$  level grid, and these fields are interpolated to the model GCM grid. For the baseline GEOS-2 DAS this means that the vertical coordinate systems are interpolated from 14 to 70 levels. Note that because the GCM and PSAS use different resolution grids the values of  $n$  are context-dependent in the complexity formulas.

The baseline GEOS-2 DAS used a 6-h analysis cycle, with  $p \approx 5 \times 10^4$  observations accumulated evenly about the analysis time, as described above. The analysis cycle can be made shorter, potentially leading to a more accurate algorithm. In section 3 this is discussed in the context of the Kalman filter. For now, note that as the analysis cycle time is reduced the computational complexity of the analysis equation (6) for the interval [0, 6 h] remains fixed at  $f s n p$ . However, for this fixed interval the complexity of the solver, Eq. (5), will be reduced to approximately  $\mathcal{N}_i f \mathcal{N}_i s (p/\mathcal{N}_i)^2 = f \mathcal{N}_i s p^2 / \mathcal{N}_i$ , where  $\mathcal{N}_i$  is the number of analysis cycles in [0, 6 h]. Thus, if the analysis cycle time were reduced to the 3-min time step of the model dynamics for baseline GEOS-2 GCM, the complexity of the analysis solver would be reduced by a factor of  $\mathcal{N}_i = 120$ . In the following section we show that for the baseline GEOS-2 PSAS, the implementations of Eqs. (5) and (6) contribute to the computational complexity of the PSAS in the ratio 35:62. Therefore, reducing the analysis cycle time reduces the overall complexity, thus allowing the use of an increasing number of observations.

### c. The timing profile of GEOS-2 DAS

The baseline GEOS-2 DAS uses shared-memory multitasking parallelism on Cray J series and SGI Origin

TABLE 2. The percentage of time taken by the components of shared-memory multitasking parallel baseline GEOS-2 DAS. Runs were performed on eight processors of an SGI Origin 2000.

GEOS-2 DAS component	Percentage of wall-clock time (%)
GCM	45.0
PSAS	39.0
Diagnostics	13.5
Interface	2.5

computers. Technical issues and limitations in developing scalable distributed-memory parallel implementations of the GCM and PSAS, and by extension GEOS DAS, are discussed in section 4. In this section we discuss the timing profile of shared-memory parallel GEOS-2 DAS.

Table 2 shows the percentage of time taken by the top-level components of the baseline GEOS-2 DAS run on eight processors of an SGI Origin 2000.<sup>2</sup> Note that the time taken for the diagnostics involves the CPU time to accumulate and process three-dimensional arrays and the time to write data to disk. The interface time accounts for the input and initial processing of the ( $p \approx 5 \times 10^4$ ) observations, plus the quality control component, which culls a priori unreliable observations (e.g., those observations whose locations or values are in gross error). The GCM, PSAS, diagnostics, and interface software, which comprise about 150 000 lines of FORTRAN 77 and FORTRAN 90 code, make substantial use of shared-memory multitasking parallelism. Overall, 0.6% of the serial time cost of GEOS-2 DAS (i.e., as timed on a single processor) arises from code that is not parallelized; of this, about half is in the initialization and data processing components of the PSAS and half is in the interface. As a check, we estimate the percentage of wall-clock time for the interface when GEOS-2 DAS is run on eight processors. Let  $u$  be the fraction of serial time cost of the interface; that is,  $u = 0.003$  (henceforth  $u$  is referred to as the “serial fraction”). Then the fraction of wall-clock time of the interface on  $N_p = 8$  is approximately  $u/(u + (1 - u)/N_p) = 0.024$ . The unparallelized component of the PSAS does not significantly modify the result; however, we will later have to take this into account when dealing with the scalability of GEOS DAS for larger numbers of processors. In the present case, for eight processors the figure 0.024 (2.4%) is in line with the value shown in Table 2.

Table 3 shows the percentage of time taken by the top-level components of the baseline GEOS-2 GCM. The GCM is run in “assimilation mode” using the Matsuno time-stepping scheme. The times for the dynamics, the Shapiro filter spatial smoother, the polar rotation,

<sup>2</sup> The numbers in this paper were obtained on an Origin 2000 with 64 processors and 16 GB of memory at NASA Ames Research Laboratory. Other numbers were obtained for a Cray J916 with 16 processors and 2 GB of memory at NASA Goddard Space Flight Center.

TABLE 3. The percentage of time taken by the top-level components of the GEOS-2 GCM (vc6.5; Takacs 1997). Although these numbers are for 10 processors of the Cray J90, they do not differ significantly from the baseline 8 processors on the SGI Origin 2000.

GCM component	Percentage of wall-clock time (%)
Dynamical core	43.0
Moist convection	16.0
Turbulence	10.0
Radiation	32.0

and other grid transformations are bundled into a single-component designated dynamical core (Takacs et al. 1994). Although these numbers are for 10 processors of the Cray J90, they do not differ significantly from the baseline 8 processors on the SGI Origin 2000.

The percentage of time taken by the top-level components of the baseline GEOS-2 PSAS is shown in Table 4. The solver [Eq. (5)] with complexity  $f\mathcal{N}_i sp^2$  takes about 35% of the time while the analysis [Eq. (6)] with complexity  $fsnp$  takes 62% of the time. These expressions for complexity can be checked approximately by taking the nominal values,  $f = 10$ ,  $p = 5 \times 10^4$ ,  $n = 10^6$ ,  $\mathcal{N}_i = 10$ , and  $s = 0.4$ . Using these numbers,  $f\mathcal{N}_i sp^2 = 10^{11}$  and  $fsnp = 2 \times 10^{11}$ , that is, the estimated count of floating-point operations for the PSAS is  $3 \times 10^{11}$  per analysis. The Cray J916 Hardware Performance Monitor reports  $5 \times 10^{11}$  floating-point multiplications and  $4.5 \times 10^{11}$  floating-point additions for the total complexity of GEOS-2 DAS (including the GCM, PSAS, diagnostics, and interface) per analysis. Therefore, Table 2 indicates that  $39/100 \times 9.5 \times 10^{11} \approx 3.7 \times 10^{11}$  is more like the actual number of flops per analysis for the baseline GEOS-2 PSAS.

### 3. The Kalman filter

The Kalman filter (Jazwinski 1970; Cohn 1997) assimilates observations sequentially with the model, interpolated to the nearest time step ( $t_k$ ) when they are taken. In this regard, it is like the PSAS with a shortened analysis update cycle:

$$\mathbf{w}_k^a = \mathbf{w}_k^f + \mathbf{K}_k(\mathbf{w}_k^o - \mathbf{H}_k \mathbf{w}_k^f), \quad (7)$$

where the Kalman gain is

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1}, \quad (8)$$

where  $s_k$  observations are assimilated at time  $t_k$ . For the Kalman filter analysis, the cycle also involves both a model forecast,

$$\mathbf{w}_{k+1}^f = \mathcal{M}_k \mathbf{w}_k^a, \quad (9)$$

and a dynamically consistent forecast of the state error covariance matrix,

$$\mathbf{P}_{k+1}^f = \mathbf{M}_k \mathbf{P}_k^a \mathbf{M}_k^T + \mathbf{Q}_k, \quad (10)$$

where  $\mathbf{M}_k$  is the tangent-linear model matrix, and  $\mathbf{Q}_k$  is the model (or system) error covariance matrix. The analysis error covariance matrix at the new time  $t_{k+1}$  is

TABLE 4. The percentage of time taken by the top-level components of the baseline GEOS-2 PSAS.

PSAS component	Percentage of wall-clock time (%) 8 processors of an SGI Origin 2000
Solver [Eq. (5)]	35.0
Analysis [Eq. (6)]	62.0
Utilities	3.0

$$\mathbf{P}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1}^f, \quad (11)$$

where  $\mathbf{I}$  is the identity matrix. The filter then proceeds sequentially in time through repeated iterations of Eqs. (7)–(11).

A two-dimensional (latitude–longitude) Kalman filter for the assimilation of stratospheric chemical constituents, developed by Lyster et al. (1997a), is being used for scientific study of stratospheric constituent gases (Ménard et al. 2000; Menard and Chang 2000). The dynamical model uses advective transport with a grid-point-based flux-conserving algorithm (Lin and Rood 1996). The transport is driven by prescribed winds from GEOS DAS. At  $2.5^\circ \times 2^\circ$  resolution the number of grid points is  $n = 91 \times 144 = 13\,104$  and the model time step is 15 min. This was used for the assimilation of retrieved methane from the Cryogenic Limb Array Etalon Spectrometer (CLAES) instrument aboard NASA's Upper Atmosphere Research Satellite (UARS). For CLAES, there were typically  $p_k \approx 15$  observations, per layer, per time step. The Kalman filter achieved 150 days of assimilation per wall-clock day, or 4.1 sustained gigaflop  $s^{-1}$ , on 128 processors of the Cray T3E-600 at NASA Goddard Space Flight Center.

For the gridpoint-based horizontal transport that is used for the two-dimensional Kalman filter, the complexity of a single time step of the model, Eq. (9), is  $hn$ , where  $h \approx 10 - 100$  takes into account the size of the finite-difference template. The complexity of Eq. (10) is  $(2h + 1)n^2$  per analysis cycle. The Kalman gain, Eq. (8), may be evaluated using a direct solver using  $\mathcal{O}(p_k^3)$  operations. Alternatively, Eqs. (5) and (6) may be employed; their computational complexity was discussed in section 2b. However, the method described in section 2b does not generate the Kalman gain  $\mathbf{K}_k$  explicitly. The complexity of Eq. (11) is approximately  $(p_k + 1)n^2$ . For the GEOS-2 DAS, observations are aggregated over a 6-h analysis cycle. As described above, the value of  $p_k$  for the Kalman filter is smaller than for the GEOS-2 DAS by the number of model time steps in 6 h. At baseline resolution for the GCM ( $2.5^\circ \times 2^\circ \times 70$  layers) the time step of the dynamics is 3 min, so  $p_k$  is 120 times smaller than for the PSAS. Only small experiments (e.g.,  $p_k < 10^3$ ) could afford to evaluate  $\mathbf{K}_k$  directly. A Kalman filter or an approximate Kalman filter for a large-scale multivariate meteorological system would have to use an iterative solver, such as the PSAS. The matrices  $\mathbf{P}_k^{fa}$  are of size  $n^2$ , and  $\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{R}_k$  is of size  $p_k^2$ .

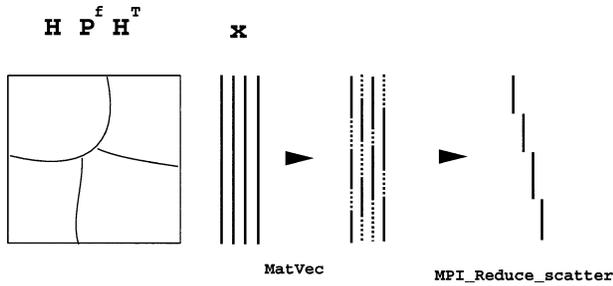


FIG. 1. Schematic of the parallel decomposition for dense matrix–vector multiply for four processors.

A Kalman filter based on a tangent-linear three-dimensional GCM would require considerably more resources than the two-dimensional filter described above for stratospheric analyses. The memory to store the error covariance matrices,  $\mathbf{P}^{fa}$ , would be approximately  $n^2 \approx 6.8 \times 10^{12}$  words at the baseline resolution of  $2.5^\circ \times 2^\circ \times 70$  levels. The floating-point operations in Eq. (10) are generated by  $2n$  applications of the tangent-linear matrix. Assuming that the resolution and throughput is fixed at that of GEOS-2 DAS, the required operations rate for Eq. (10) would be  $2n \times 250$  megaflop  $s^{-1} = 0.5$  petaflop  $s^{-1}$  (the value 250 megaflop  $s^{-1}$  is taken from the baseline GEOS-2 DAS in Table 1). This is clearly beyond the reach of current resources. GEOS DAS, with an analysis based on PSAS, is an approximate Kalman filter. Efforts are under way worldwide and at the GMAO to develop computationally feasible improvements to 4DDA algorithms, such as reducing the analysis cycle time for GEOS DAS and developing more physically based error covariance models (Riishøjgaard 1998).

#### 4. The scalable distributed-memory parallel GEOS DAS

The baseline GEOS-2 DAS uses shared-memory multitasking parallelism on Cray J series and SGI Origin computers. A distributed-memory parallel implementation of the GCM was designed (Lyster et al. 1997b) and prototyped (Sawyer and Wang 1999) using the Message-Passing Interface (MPI) and shmem libraries. Distributed-memory parallel PSAS was prototyped (Ding and Ferraro 1995), and an MPI PSAS kernel was developed (Guo et al. 1998; Larson et al. 1998). During the year 2001, development and validation of distributed-memory parallel GEOS DAS was completed. The GCM and PSAS have tightly coupled core algorithms with computation- and communication-intensive parallel implementations; these are hydrodynamic transport (GCM) and nonsparse large matrix–vector multiplications (PSAS; see Fig. 1). We discuss theoretical limits to the development of scalable distributed-memory parallel implementations of the GCM and PSAS. We then describe in terms of the well-known Amdahl’s law (1967) how the serial component of the GEOS DAS application impacts scalability. As we developed the

distributed-memory parallel application based on GEOS-2 DAS, this limit was the most important in determining the maximum number of processors that can be usefully employed to run the application.

Appendixes A and B quantify the limitations of scalable distributed-memory parallel implementations of the GCM and PSAS, respectively. We focus on the tightly coupled core hydrodynamic transport and nonsparse large matrix–vector multiply algorithms. The parallel speedup (SU), Eq. (A4), is defined as the time to run the application on one processor divided by the time to run it on  $N_p$  processors. An ideal parallelization would have  $SU = N_p$ . However, because of a combination of the inter-processor communication overhead and the difficulties in balancing the workload among processors, SU falls increasingly below the ideal linear scaling for an increasing number of processors. We define the maximum number of processors  $N_{pmax}$  to be where SU is one-half the ideal value. For gridpoint-based transport algorithms  $N_{pmax}$  is given by Eq. (A5). For parameters typical of current global transport algorithms ( $1^\circ \times 1^\circ$  resolution, using a two-dimensional horizontal parallel domain decomposition, a single-processor speed of 100 megaflop  $s^{-1}$ , and an interprocessor communication bandwidth of 10 MB  $s^{-1}$ ) Eq. (A5) gives  $N_{pmax} = 400$ . The parallel matrix–vector multiply at the core of the PSAS distributes the work in matrix–vector blocks across processors and uses collective MPI library routines `MPI_reduce_scatter()` and `MPI_allgather()` (Fig. 1). The maximum speedup is given by Eq. (B3). For typical parameters of the PSAS ( $p \approx 10^5$ ,  $s = 0.4$ ,  $f = 10$ )  $N_{pmax}$  is of the order of thousands of processors. Figure 2 shows that the scalability is further limited by load imbalance in the distribution of matrix–vector blocks to the processors.

We have shown that the highly coupled parallel subcomponents of distributed-memory parallel gridpoint GCM and PSAS have upper limits to their scalability in the range 400–1000 processors on SGI Origin 2000 series and similar computers. We have also shown in Tables 2, 3, and 4 that the main subcomponents of the GEOS-2 DAS (dynamical core, moist convection, turbulence, radiation, PSAS solver, PSAS analysis, diagnostics, and interfaces) have an approximately flat timing profile. This means that a large fraction of 150 000 lines of code are candidates for single-processor optimization. In addition to these issues of single-processor optimization and parallel scalability of core algorithms, we have to account for unparallelizable and unparallelized code. Similar to the discussion in section 2c, let  $u$  be the serial fraction of GEOS-2 DAS. As above, the speedup (SU) is defined as the time taken to run the application on one processor divided by the time to run it on  $N_p$  processors. Then

$$SU = N_p / (1 - u + N_p u). \quad (12)$$

Assuming  $u \ll 1$ , the maximum number of processors—defined as where the speedup is one-half the ideal value—is  $1/u$ , that is, the limit is approximately the

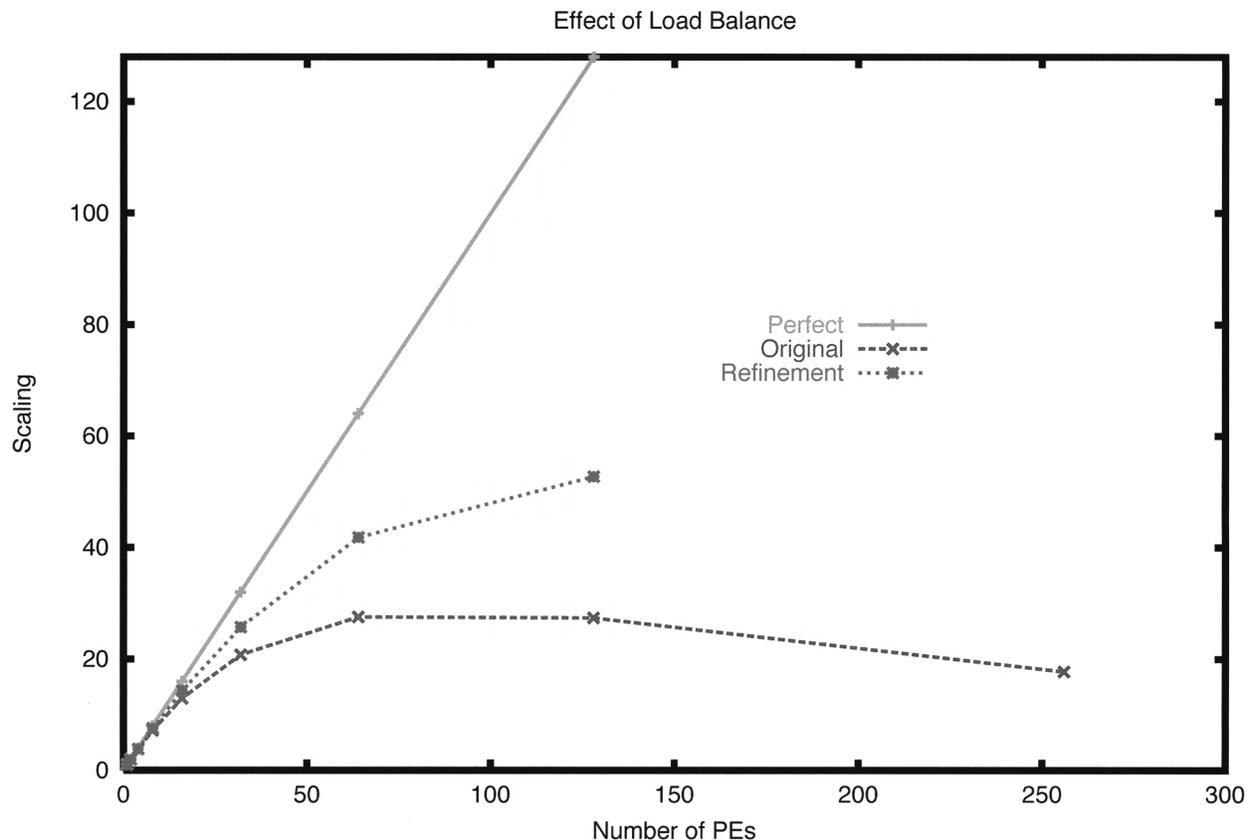


FIG. 2. Improvements in scalability of the MPI PSAS kernel due to load balancing.

inverse of the fraction of the serial time cost of the application; this is a statement of Amdahl's law (1967). For the baseline GEOS-2 DAS,  $u = 0.006$  (section 2c), so the Amdahl's limit on the entire parallel GEOS-2 DAS application is  $1.0/0.006 = 166$  processors.

This analysis shows that, regardless of how efficiently the core computation- and communication-intensive components of the GCM and PSAS are parallelized, the parallel application based on GEOS-2 DAS will not scale beyond 166 processors. This is due to the small but ultimately significant component of unparallelized and unparallelizable code. Increasing the resolution of the transport algorithm, and using more observations, will improve scalability because there is correspondingly more work to distribute among processors. Efforts to parallelize more of GEOS DAS, as well as single-processor optimization of the serial code, should also increase the scalability of the GEOS DAS application.

## 5. Summary

We have discussed the computational complexity of the GEOS-2 DAS, which was the baseline production system in use at NASA's Global Modeling and Assimilation Office (GMAO) in the late 1990s. The complexity of the general circulation model (GCM) gen-

erally scales linearly with the number of gridded state variables,  $n$ , per iteration of the algorithm (with the exception of the quadratic scaling of the radiation algorithms with respect to the number of vertical levels). The need to reduce the time step of the dynamics as the spatial resolution is increased results in an asymptotic  $n^{4/3}$  scaling for the dynamical core for the simulation of fixed time intervals. The Physical-Space Statistical Analysis System (PSAS) has asymptotic scaling  $sp^2$  and  $snp$ . The former arises from the solver, Eq. (5), and the latter from Eq. (6), the fundamental basis of which is the error correlation between all observations and all gridpoints in an analysis cycle. The computational complexity of the PSAS is reduced by using error correlation models with compact support so that the fraction of nonzero matrix elements is  $s \approx 0.4$ . Other modifications, such as reducing the analysis cycle time, may be used to reduce the computational complexity. We also present a timing profile of the baseline resolution GOES-2 DAS (Tables 1, 2, 3, and 4) on computers with 8 or 10 processors using shared memory parallelism. A simple application of Amdahl's law illustrates how the parallel scaling efficiency is affected by the fraction of unparallelized code. The computational complexity and the required computer memory of the Kalman filter is quadratic in  $n$ . We showed, using a simple estimate based

on the performance of the GEOS-2 GCM, that a Kalman filter for atmospheric global data assimilation would require petaflop  $s^{-1}$  computing to achieve effective throughput for scientific research. We have developed a computationally tractable Kalman filter suitable for research on stratospheric constituent gas assimilation where the dynamics are two-dimensional. We noted that the development of a full, petaflop  $s^{-1}$  scale, Kalman filter would be an ambitious and scientifically significant exercise, but the main thrust for practical or operational implementations concentrate on approximate Kalman filters with reduced computational complexity.

We developed parameterized formulas that estimate the limit to distributed-memory parallel scalability of the tightly coupled transport and large matrix–vector multiplications, which are important components of the CPU time cost of the gridpoint GCM and PSAS. For SGI Origin 2000 and similar computers the scalability is limited to 400–1000 processors. In addition, the unparallelizable and unparallelized code pose significant limits on the scalability of the end-to-end algorithms. For GEOS-2 DAS the serial fraction (i.e., the fraction of CPU time cost as run on a single processor) is only 0.006. This includes I/O and represents by far the bulk of the lines of code of GEOS-2 DAS. Therefore, the Amdahl's limit of scalability of the distributed-memory parallel implementation of this application is  $1/0.006 = 166$  processors. This result holds regardless of how efficiently the GCM and PSAS are parallelized. Therefore, efforts to improve the throughput of GEOS DAS necessarily involve efficient parallelization of a very large number of the 150 000 lines of code (aside from the core transport and matrix–vector multiply subcomponents), and improving their serial performance.

*Acknowledgments.* The authors acknowledge discussions with Steven Cohn, Anthony Colyandro, Chris Ding, Gregory Gaspari, Dan Kokron, William Sawyer, Harper Pryor, Richard Rood, Meta Sienkiewicz, Lawrence Takacs, Ricardo Todling, and Philip Webster. This work was funded under the NASA High Performance Computing and Communications Initiative (HPCC) Earth and Space Science (ESS) Program Cooperative Agreement NCCS5-150, the NASA EOS Interdisciplinary Science Program, and the NASA Research and Applications Program.

## APPENDIX A

### Asymptotic Scalability of Distributed-Memory Parallel Gridpoint General Circulation Models

We calculate the limit on the number of processors that can be usefully employed to reduce the wall-clock time of a distributed-memory parallel-gridpoint-based transport algorithm. In the parallel decomposition, compact domains of grid points and their associated floating-point operations are distributed across processors. The

limit on the number of processors is the result of the surface-to-volume effect (e.g., Foster 1995; section 2.4), whereby the impact of communication of domain surface data becomes comparable to the time to perform the floating-point operations of the algorithm. This is an approximation of the scalability in the sense that it does not account for a number of the typical complications that often occur in GCMs, namely:

- We are neglecting the algorithms for parameterized physics processes, which include moist convection, turbulence, and radiation; the grid transformations; the diagnostics; and the I/O.
- We are not assessing the impact of load imbalance.
- We cannot simply account for indeterminacy in communications, such as in semi-Lagrangian methods.

The embarrassingly parallel parts of the GCM (e.g., some algorithms for parameterized physics processes) tend to improve the overall scaling with respect to the present calculation, while load imbalance will tend to make the scaling worse. Other components (e.g., the parallel rotation grid transformation) need a separate analysis (Lyster 2000 and articles therein). The communication of domain surface data enables algorithmic consistency across the boundary between processor domains. The present calculation is very similar to the estimate of parallel scalability of particle-in-cell methods by Lyster et al. (1995), except that case involved communication of mobile particles, which represented plasma ions and electrons, across gridpoint domain boundaries. We assume that the communication time can be approximated in terms of the number of bytes communicated per processor and the bandwidth of the communication channel (i.e., latency effects make the scalability worse, so this approximation is still good in terms of evaluating an upper bound on scalability). With this, the following calculation provides a good approximation for the scalability of the distributed-memory parallel dynamical core of the GEOS GCM.

Define the following:

$N_p$	=	total number of processors employed
$N_g$	=	total number of grid points in the computational domain
$d$	=	dimension of the physical problem
$D$	=	dimension of the parallel decomposition
$M$	=	single-processor speed in megaflop $s^{-1}$
$B$	=	interprocessor communication bandwidth in MB $s^{-1}$
$F$	=	number of flops/grid point/time step for the relevant transport algorithm
$G$	=	the number of “layers” of guard cells in each dimension of the parallel decomposition (e.g., $G = 2$ for fourth-order finite difference)
$P$	=	the precision of the calculation in bytes per word (i.e., $P = 4$ or $8$ )

Typically  $D = 1, 2,$  or  $3,$  and  $d = 2$  or  $3,$  while  $d \geq D.$  Because it is difficult to parallelize global hydrody-

dynamic algorithms in the vertical dimension, most parallel implementations use a compact horizontal, or “checkerboard,” parallel decompositions (i.e.,  $D = 2$ ). It is therefore sufficient to quantify scalability in terms of the number of grid points in the latitude–longitude domain (i.e., horizontal transport), for which  $d = 2$ . The number of grid points around the border of each domain is then  $2DN_g^{(d-1)/d}/N_p^{(D-1)/D} \equiv 2DN_g^{1/2}/N_p^{1/2}$ . The communication time per time step per processor is

$$T_{\text{comm}} = 2DGPB^{-1}N_g^{(d-1)/d}/N_p^{(D-1)/D}. \quad (\text{A1})$$

The CPU time per time step per processor is

$$T_{\text{cpu}} = (F/M)(N_g/N_p). \quad (\text{A2})$$

Hence the ratio of communication to CPU time is

$$\tau := T_{\text{comm}}/T_{\text{cpu}} = \frac{2DGP}{F} \frac{M}{B} \frac{N_p^{1/D}}{N_g^{1/d}}. \quad (\text{A3})$$

The parallel speedup (SU) is defined as the time for the application to run on one processor divided by the time to run on  $N_p$  processors. With the present assumptions, we have

$$\text{SU} = N_p/(1 + \tau). \quad (\text{A4})$$

Therefore, we may nominally define the maximum speedup,  $N_{p\text{max}}$  as the number of processors for which  $\tau$  in Eq. (A3) is equal to 1:

$$N_{p\text{max}} = \left[ \frac{BFN_g^{1/d}}{2MDGP} \right]^D. \quad (\text{A5})$$

Beyond that, the floating-point operations in additional processors are effectively wasted.

The terms in  $\tau$  may be characterized as follows:

- $2DGP/F$ : parameters of the computational algorithm
- $M/B$ : parameters of the computer
- $N_g^{1/d}$ : the problem resolution
- $N_p^{1/D}$ : the surface-to-volume effect (i.e.,  $\tau$  gets larger in proportion to the number of processors to some geometry-dependent exponent.)

For parameters typical of current global transport algorithms,  $N_g = 360 \times 181$  (i.e.,  $1^\circ \times 1^\circ$  resolution),  $d = D = 2$ ,  $M = 100$ ,  $B = 10$ ,  $F = 50$ ,  $G = 2$ , and  $P = 8$ , so Eq. (A5) gives  $N_{p\text{max}} = 400$ .

## APPENDIX B

### Asymptotic Scalability of Distributed-Memory Parallel Matrix–Vector Multiply for the PSAS Solver

We calculate the limit on the number of processors that can be usefully employed to reduce the wall-clock time of a distributed-memory dense matrix–vector multiply. The dominant time cost of the PSAS, Eq. (5) and (6), are large, dimension  $p \approx 10^5$ , matrix–vector multiplications. For the present analysis the results do not

differ significantly between the symmetric [Eq. (5)] or rectangular [Eq. (6)] cases since the structure of each dimension of the matrix is determined by a compact spatial decomposition of the multidimensional data (see Guo et al. 1998). We therefore only show the scaling analysis for the symmetric case. Parallelism is achieved by assigning subsets of the matrix–vector multiplication to each processor. The left-hand side of Fig. 1 shows schematic matrix sub-blocks that are distributed among processors (only four are illustrated). Each processor performs matrix–vector multiplies (MatVec) corresponding to its sub-block, thus yielding four partial vectors. The partial vector results are then summed using the `MPIreduce_scatter()` library call. The cycle of the parallel matrix–vector multiply is then completed using the `MPIall_gather()` library call (not shown in the figure).

Advanced libraries such as PLAPACK (van de Geijn et al. 1997) have custom interfaces and decompositions to support dense matrix–vector operations. We chose not to use this because the more general interface of the MPI library is both simple and compatible with the pointer-specified multidimensional vectors (Larson et al. 1998). Using a 6000-km cutoff length for correlation functions, the matrices are semidense, with density  $s \approx 0.4$ . For the moment, we focus on the limitations on scalability due to the trade-off between communications in the `MPIreduce_scatter()` and `MPIall_gather()`, and the time cost of the sub-block matrix–vector multiplications. We ignore the costs of the floating-point operations in the reduction. As in appendix A, we ignore the cost of latency in the interprocessor communications.

Assuming that the collective MPI communication calls described above are implemented using an efficient method such as recursive halving (Foster 1995, section 11.2) the cost of communications is

$$T_{\text{comm}} = 2(pP/B)(N_p - 1)/N_p \approx 2pP/B, \quad (\text{B1})$$

where we have used the same definitions as in appendix A, and  $p \approx 10^5$  is the size of the vector. The CPU time per processor is

$$T_{\text{cpu}} = fsp^2/(N_pM), \quad (\text{B2})$$

where, as in section 2b,  $f$  equals 2 plus the number of floating-point operations to form each matrix element. The parallel speedup is given by Eq. (A4), and the maximum speedup is defined in the same way as in appendix A:

$$N_{p\text{max}} = \frac{fspB}{2PM}. \quad (\text{B3})$$

For typical values for these parameters as defined in appendix A and above,  $N_{p\text{max}} = 625fs$ . If the matrix is precalculated,  $f = 2$ , but it may be of order 10 when elements are calculated on the fly. Memory limitations prohibit storing entire matrices, so current implementations enable a combination of prestored and on-the-

fly calculation of matrix elements. The matrix density is  $s \approx 0.4$ , so it is clear that the upper limit of scalability of semidense matrix–vector multiplications, and hence the PSAS, is of the order of thousands of processors for current generation machines and current input datasets. The value is larger than the upper limit for a GCM because transport algorithms in the dynamical cores of GCMs are sparse matrix algorithms, which have more stringent scalability limits due to the surface-to-volume effect described in appendix A.

The calculation thus far presents an upper limit on scalability. We discuss here a number of factors that reduce the scalability of the PSAS below the theoretical limit. First, on large numbers of processors the size of the vector segments are sufficiently small that message latency and synchronization dominate the communication cost of collective MPI calls. Second, the PSAS has a nested preconditioner that involves successively sparser matrix–vector multiplications (Cohn et al. 1998; Larson et al. 1998). Through Eq. (B3) (i.e.,  $N_{pmax} \sim s$ ) these will negatively effect scalability. Third, workload imbalance has a serious impact on parallel scalability. The baseline MPI PSAS kernel has an upper limit of 57 600 matrix blocks, which should be sufficient to provide a statistically uniform distribution when their work is allocated across 1000 or 2000 processors (Lyster et al. 2000). However, these blocks are of widely differing size because their dimensions depend on the nonrepeatable distribution of observations in geographical areas of the earth. Early versions of the kernel used a method for load balancing that based the costs of the block matrix–vector multiplications on the dimensions of the blocks. This was later augmented, with only incremental improvement in scalability, by dynamic scheduling and work scheduling based on statistically tuned cost estimates. The lower curve of Fig. 2 (from Lyster et al. 2000). shows the scaling of the baseline MPI PSAS kernel, including the load-balancing algorithm for 52 738 observations covering a standard 6-h analysis cycle. The poorer scaling relative to the above calculation is from a combination of load imbalance and sparse preconditioners; using  $s = 0.1$  and  $f = 5$  in Eq. (B3) gives  $N_{pmax} = 312$ , which is in line with Fig. 2. The lower curve in Fig. 2 corresponds to the case of approximately 57 600s blocks. The improved scaling shown in the upper curve of the figure corresponds to the improved load balance that resulted from a refinement to 921 600s blocks. The value of the improved scaling at 256 processors was not obtained due to restricted availability of the computer at the time of the experiments. However, from the scaling up to 128 processors it is clear that the MPI PSAS kernel did not reach the theoretical limit that had been expected from the above calculation. Apart from our work on load-balancing algorithms, we have developed and continue to work on collective parallel algorithms using optimized communication procedures.

## REFERENCES

- Amdahl, G. M., 1967: Validity of the single-processor approach to achieving large scale computing capabilities. *Proc. AFIPS Spring Joint Computer Conf.*, Atlantic City, NJ, AFIPS, 483–485.
- Andersson, E., and Coauthors, 1998: The ECMWF implementation of three dimensional variational assimilation (3D-Var). Part III: Experimental Results. *Quart. J. Roy. Meteor. Soc.*, **124**, 1831–1860.
- Atlas, R., 1997: Atmospheric observations and experiments to assess their usefulness in data assimilation. *J. Meteor. Soc. Japan*, **75**, 111–130.
- , R. N. Hoffman, S. C. Bloom, J. C. Jusem, and J. Ardizzone, 1996: A multiyear global surface wind velocity dataset using SSM/I wind observations. *Bull. Amer. Meteor. Soc.*, **77**, 869–882.
- Bloom, S. C., L. L. Takacs, A. M. da Silva, and D. Ledvina, 1996: Data assimilation using incremental analysis updates. *Mon. Wea. Rev.*, **124**, 1256–1271.
- Brackbill, J. U., and B. I. Cohen, Eds., 1985: *Multiple Time Scales*. Academic Press, 442 pp.
- Cohn, S. E., 1997: An introduction to estimation theory. *J. Meteor. Soc. Japan*, **75**, 257–288.
- , A. da Silva, J. Guo, M. Sienkiewicz, and D. Lamich, 1998: Assessing the effects of data selection with the DAO Physical-space Statistical Analysis System. *Mon. Wea. Rev.*, **126**, 2913–2926.
- Courtier, P., and Coauthors, 1998: The ECMWF implementation of three dimensional variational assimilation (3D-Var). Part I: Formulation. *Quart. J. Roy. Meteor. Soc.*, **124**, 1783–1808.
- Daley, R., 1991: *Atmospheric Data Analysis*. Cambridge University Press, 457 pp.
- da Silva, A., and J. Guo, 1996: Documentation of the Physical-Space Statistical Analysis System (PSAS) part I: The conjugate gradient solver version, PSAS-1.00. NASA Data Assimilation Office Note 96-02, 66 pp. [Available online at <http://gmao.gsfc.nasa.gov/>.]
- Ding, H., and R. Ferraro, 1995: A general purpose parallel sparse matrix solver package. *Proc. Ninth Int. Parallel Processing Symp.*, Santa Barbara, CA, IEEE, 70.
- Foster, I., 1995: *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison-Wesley, 381 pp.
- Gaspari, G., and S. E. Cohn, 1999: Construction of correlation functions in two and three dimensions. *Quart. J. Roy. Meteor. Soc.*, **125**, 723–757.
- Gibson, J. K., P. Källberg, S. Uppala, A. Nomura, A. Hernandez, and E. Serrano, 1997: *ERA Description*. Vol. 1. ECMWF Re-Analysis Project Report Series, ECMWF, 71 pp.
- GMAO, cited 2000: Algorithm theoretical basis document version 2.01. Global Modeling and Assimilation Office, NASA Goddard Space Flight Center. [Available online at <http://gmao.gsfc.nasa.gov/>.]
- Golub, G. H., and C. F. van Loan, 1989: *Matrix Computations*. 2d ed. The Johns Hopkins University Press, 642 pp.
- Guo, J., J. W. Larson, P. M. Lyster, and G. Gaspari, cited 1998: Documentation of the Physical-space Statistical Analysis System (PSAS). Part II: The factored-operator error covariance model formulation. NASA Data Assimilation Office Note 98-04, 27 pp. [Available online at <http://gmao.gsfc.nasa.gov/>.]
- Jazwinski, A. H., 1970: *Stochastic Processes and Filtering Theory*. Academic Press, 276 pp.
- Kalnay, E., and Coauthors, 1996: The NCEP/NCAR 40-Year Reanalysis Project. *Bull. Amer. Meteor. Soc.*, **77**, 437–471.
- Kistler, R., and Coauthors, 2001: The NCEP–NCAR 50-Year Reanalysis: Monthly means CD-ROM and documentation. *Bull. Amer. Meteor. Soc.*, **82**, 247–268.
- Larson, J. W., J. Guo, and P. M. Lyster, 1998: Documentation of the Physical-space Statistical Analysis System (PSAS). Part III:

- Software implementation. NASA Data Assimilation Office Note 98-05, 189 pp. [Available online at <http://gmso.gsfc.nasa.gov/>.]
- Lin, S.-J., and R. B. Rood, 1996: Multidimensional flux-form semi-lagrangian transport schemes. *Mon. Wea. Rev.*, **124**, 2046–2070.
- Lyster, P. M., cited 2000: Final report on the NASA HPCC PI project: Four dimensional data assimilation. [Available online at <http://ct.gsfc.nasa.gov/lys/hpccfinal>.]
- , P. C. Liewer, R. D. Ferraro, and V. K. Decyk, 1995: Implementation and characterization of three-dimensional particle-in-cell codes on multiple-instruction-multiple-data parallel supercomputers. *Comput. Phys.*, **9**, 420–432.
- , S. E. Cohn, R. Ménard, L.-P. Chang, S.-J. Lin, and R. Olsen, 1997a: Parallel implementation of a kalman filter for constituent data assimilation. *Mon. Wea. Rev.*, **125**, 1674–1686.
- , W. Sawyer, and L. L. Takacs, 1997b: Design of the Goddard Earth Observing System (GEOS) Parallel General Circulation Model (GCM). NASA Data Assimilation Office Note 97-13, 17 pp. [Available online at <http://gmso.gsfc.nasa.gov/>.]
- , T. Clune, J. Guo, and J. W. Larson, cited 2000: Performance optimization of the Physical-space Statistical Analysis System (PSAS), 38 pp. [Available online at <http://ct.gsfc.nasa.gov/lys/hpccfinal>.]
- Ménard, R., and L.-P. Chang, 2000: Assimilation of stratospheric chemical tracer observations using a Kalman filter. Part II:  $\chi^2$ -validated results and analysis of variance and correlation dynamics. *Mon. Wea. Rev.*, **128**, 2672–2686.
- , S. E. Cohn, L.-P. Chang, and P. M. Lyster, 2000: Assimilation of stratospheric chemical tracer observations using a Kalman filter. Part I: Formulation. *Mon. Wea. Rev.*, **128**, 2654–2671.
- Parrish, D. F., and J. C. Derber, 1992: The National Meteorological Center's spectral statistical-interpolation analysis system. *Mon. Wea. Rev.*, **120**, 1747–1763.
- , ——, R. J. Purser, W.-S. Wu, and Z.-X. Pu, 1997: The NCEP global analysis system: Recent improvements and future plans. *J. Meteor. Soc. Japan*, **75**, 359–365.
- Rabier, F., A. Mc Nally, E. Andersson, P. Courtier, P. Undén, J. Eyre, A. Hollingsworth, and F. Bouttier, 1998: The ECMWF implementation of three dimensional variational assimilation (3D-Var). Part II: Structure functions. *Quart. J. Roy. Meteor. Soc.*, **124**, 1809–1830.
- Riishøjgaard, L. P., 1998: A direct way of specifying flow-dependent background error correlations for meteorological analysis systems. *Tellus*, **50A**, 42–57.
- Sawyer, W., and A. Wang, 1999: Benchmark and unit test results of the message-passing GEOS General Circulation Model. NASA Data Assimilation Office Note 99-04, 20 pp. [Available online at <http://gmso.gsfc.nasa.gov/>.]
- Schubert, S. D., R. B. Rood, and J. Pfaendtner, 1993: An assimilated dataset for earth science applications. *Bull. Amer. Meteor. Soc.*, **74**, 2331–2342.
- , and Coauthors, 1995: A multiyear assimilation with GEOS-1 System: Overview and results. Vol. 7. NASA Tech. Memo. 104606, 201 pp. [Available online at <http://gmso.gsfc.nasa.gov/>.]
- Stobie, J. G., 1996: Data assimilation computing and mass storage requirements for 1998. NASA Data Assimilation Office Note 96-16, 10 pp. [Available online at <http://gmso.gsfc.nasa.gov/>.]
- Takacs, L. L., A. Molod, and T. Wang, 1994: Documentation of the Goddard Earth Observing System (GEOS) General Circulation Model—Version 1. Vol. 1. NASA Tech. Memo. 104606, NASA Goddard Space Flight Center, Greenbelt, MD, 114 pp. [Available online at <http://gmso.gsfc.nasa.gov/>.]
- van de Geijn, R. A., P. Alpatov, G. Baker, and C. Edwards, 1997: *Using PLAPACK—Parallel Linear Algebra Package*. MIT Press, 194 pp.