Statistical Modeling of Large-Scale Simulation Data

Tina Eliassi-Rad
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory
Livermore, CA 94551
+1 (925) 422-1552
eliassi@llnl.gov

Terence Critchlow
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory
Livermore, CA 94551
+1 (925) 423-5682
critchlow@llnl.gov

Ghaleb Abdulla
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory
Livermore, CA 94551
+1 (925) 423-5947
abdulla1@llnl.gov

ABSTRACT

With the advent of fast computer systems, scientists are now able to generate terabytes of simulation data. Unfortunately, the sheer size of these data sets has made efficient exploration of them impossible. To aid scientists in gleaning insight from their simulation data, we have developed an ad-hoc query infrastructure. Our system, called AQSim (short for Ad-hoc Queries for Simulation) reduces the data storage requirements and query access times in two stages. First, it creates and stores mathematical and statistical models of the data at multiple resolutions. Second, it evaluates queries on the models of the data instead of on the entire data set. In this paper, we present two simple but effective statistical modeling techniques for simulation data. Our first modeling technique computes the "true" (unbiased) mean of systematic partitions of the data. It makes no assumptions about the distribution of the data and uses a variant of the root mean square error to evaluate a model. Our second statistical modeling technique uses the Andersen-Darling goodness-of-fit method on systematic partitions of the data. This method evaluates a model by how well it passes the normality test on the data. Both of our statistical models effectively answer range queries. At each resolution of the data, we compute the precision of our answer to the user's query by scaling the one-sided Chebyshev Inequalities with the original mesh's topology. We combine precisions at different resolutions by calculating their weighted average. Our experimental evaluations on two scientific simulation data sets illustrate the value of using these statistical modeling techniques on multiple resolutions of large simulation data sets.

Categories and Subject Descriptors

E.4 [**Data**]: Coding and Information Theory – *data compaction and compression.* G.3 [**Mathematics of Computing**]: Probability and Statistics – *distribution functions, multivariate statistics, nonparametric statistics, statistical computing.* H.2.4 [**Database Management**]: Systems – *query processing.* H.2.8 [**Database Management**]: Database Applications – *data mining, scientific databases.* H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *indexing methods.*

General Terms

Algorithms, Management, Measurement, Performance, Experimentation.

Keywords

statistical modeling, large-scale scientific data sets, approximate ad-hoc queries.

1. INTRODUCTION

By utilizing the enormous computing power currently available, scientific experiments are producing tera-scale simulation data. The size of these data sets makes even the best available visualization tools inadequate. The need to efficiently explore these large simulation data sets has led to a surge of interest in scalable modeling and visualization tools [1][2][3][4][7][10].

To help explore these huge data sets, we have created *AQSim* (short for Ad-hoc Queries for Simulation). AQSim utilizes *multi-resolution models* to reduce both the data storage requirements and the query response times. Figure 1 illustrates an overview of AQSim's data flow. AQSim has two components: (*i*) *the model generator* and (*ii*) *the query processor*. The model generator builds statistical and mathematical models of systematic partitions of the data. By generating models in this manner, we create models at different resolutions of the data. Moreover, since models take less storage space than the original data set,[1] we are able to keep the models on secondary storage. Subsequently, the query processor executes queries on these models to regenerate the appropriate subset of data. This processor decreases the query response time since models of the data are queried.

AQSim's model generator builds models from *mesh* data, which is produced by most scientific simulation code. A mesh data set consists of interconnected grids of small zones, in which data points are stored. Figure 2 depicts the mesh produced from an astrophysics simulation of a star in its mid-life. Mesh data usually varies with time, consists of multiple dimensions (*i.e.*, variables), and can contain irregular grids. Musick and Critchlow provide a nice introduction to scientific mesh data [8].

SIGKDD '02, July 23-26, 2002, Edmonton, Alberta, Canada.

[1] The original data set typically resides on tertiary storage.

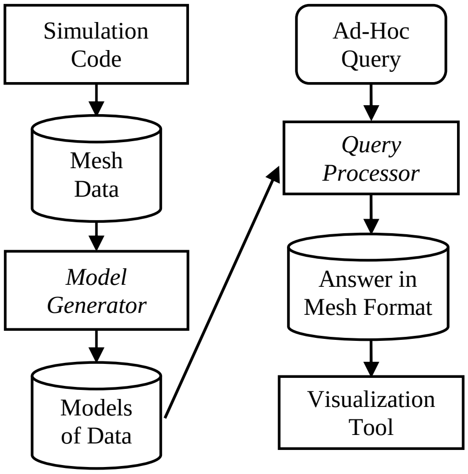

Figure 1. AQSIM Data Flow

In this paper, we describe and evaluate two statistical modeling techniques for AQSim. The first model captures the "true" (unbiased) mean of systematic partitions of the data. We call this model the *mean modeler*. The mean modeler has two main advantages. First, it makes no assumptions about the distribution of the data. Second, it calculates its model parameters through one sweep of the data at each resolution.[2] The error metric on the mean modeler is a variant of the *root mean square error* (RMSE). Our second model captures the normality of systematic partitions of the data by utilizing the *Anderson-Darling* goodness-of-fit test [5]. This model is called the *goodness-of-fit modeler*. Similar to the mean modeler, the goodness-of-fit modeler is able to calculate its model parameters through one sweep of the data. However, this modeler attempts to fit the data to a normal distribution. The error on this model is the *Type I error* associated with the goodness-of-fit test. Section 2 describes these two approaches in details.

Despite their simplicity, these models have performed extremely well on our empirical studies of range queries (see Section 4). The answer to a query is judged by its *weighted average precision* to the original data. At each partition, we calculate the precision associated with a query's answer by scaling the one-sided *Chebyshev* inequality with a metric representing the topology of the original mesh in that partition [6]. We chose to utilize the one-sided *Chebyshev* inequality since it does not make any assumptions about the data. Section 3 describes AQSim's query processor and our precision measure in details.

Section 4 presents two case-studies, which illustrate the value of our approach. Sections 5 and 6 discuss some related and future work, respectively. Section 7 summarizes our work.

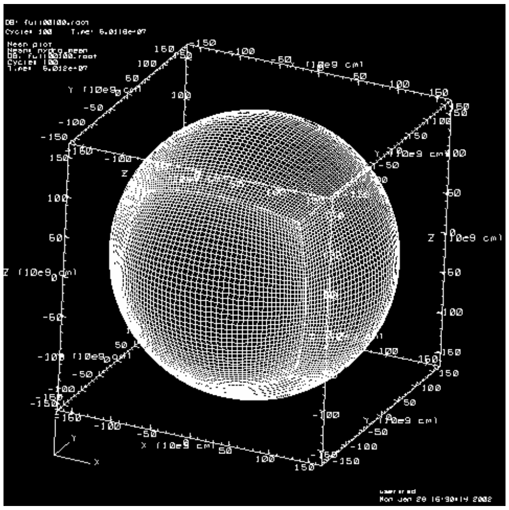

Figure 2. A Mesh Data Set Representing a Star

2. AQSIM'S MODEL GENERATOR

AQSim's model generator systematically partitions the original data and builds models on each partition. Partitioning stops when models are accurate within a user-defined error threshold.

AQSim has two different partitioning strategies: (*i*) top-down and (*ii*) bottom-up. Due to limitation in space, we will only discuss the first strategy in this paper. In the top-down approach, the data is divided in a four-way bisection on the spatial-temporal space (see Figure 3). The computational complexity of this partitioning approach is O($N_{data} \times N_{level}$), where N_{data} is the size of the original data set and N_{level} is the number of partitioning levels. For example, in Figure 3, the number of partitioning levels is 2.

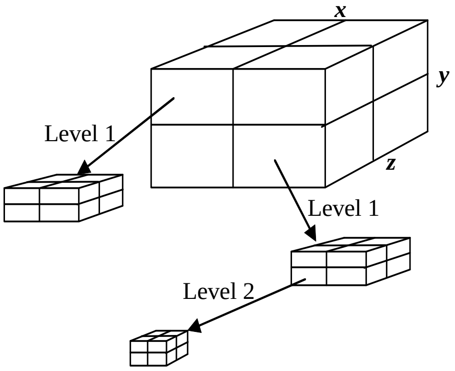

Figure 3. Top-Down Partitioning of the Data at a Particular Time Step

The remainder of this section describes AQSim's two statistical modeling techniques.

2.1 Mean Modeler

Each partition, p_k, of the data has a set of variables associated with it. For each variable v_i, the mean modeler is μ_i, where μ_i is the mean of the data points associated with v_i in partition p_k.

For the mean modeler, partitioning of the data stops when either one of the following two conditions is true:

[2] The terms *resolution* and *partition* are interchangeable in this paper.

1. $\forall v \in NonPartitioningVariables$ in node η, $\sigma_v = 0$.
2. $\forall v \in NonPartitioningVariables$ in node η, $(\mu_v - c.\sigma_v \leq \min_v)$ & $(\max_v \leq \mu_v + c.\sigma_v)$.

The first stopping criterion represents the simple case of partitions with either 1 data point or a set of data points with standard deviation of zero. In the second stopping criterion, the partition threshold, *c*, is a real number greater than or equal to zero. This user-defined threshold is a scaling factor for the standard deviation of variable *v*. For example, *c* = 1 means that the minimum and maximum values for each non-partitioning variable must be within 1 standard deviation of the mean of the data points in the node. The advantage of the above stopping criteria is that it does not assume any distribution on the data points.

For the mean modeler, standard deviation is the same as *RMSE* (root mean square error) since the *true mean*, which is an unbiased estimator, is used as the model. Thus, the *RMSE* is the error metric associated with the mean modeler.

2.2 Goodness-of-Fit Modeler

For each variable v_i in partition p_k, the goodness-of-fit modeler is $N(\mu_i, \sigma_i)$. That is, the model for v_i is a normal distribution with mean, μ_i, and standard deviation, σ_i.

For the goodness-of-fit modeler, the partitioning step stops when the hypothesis test for normality is *not rejected*. We use the *Anderson-Darling test for normality* (which is considered to be the most powerful goodness-of-fit test for normality) for our goodness-of-fit test [5].

The Anderson-Darling test involves calculating the *A^2 metric* for variable $v_i \sim N(\mu_i, \sigma_i)$, which is defined to be

$$A^2 = -\frac{1}{n}\left(\sum_{j=1}^{n}(2j-1)\left(\ln(z_j)+\ln(1-z_{n+1-j})\right)\right)-n$$

where n = number of data points for v_i and $z_j = \Phi(\frac{x_j-\mu_i}{\sigma_i})$. $\Phi(\bullet)$ is the standard normal distribution function.

We reject H_0 if $A^2\left(1+\frac{0.75}{n}+\frac{2.25}{n^2}\right)$ exceeds the *critical value* associated with the *user-specified error threshold* [5]. Otherwise, we accept H_0.

For each variable v_i, the error on this model is defined to be $Pr(\textit{reject } H_0 \mid H_0 \textit{ is true})$, where H_0 is the null hypothesis. H_0 states that the distribution of a variable v_i is normal. In other words, the model error is equal to the Type I error.

3. AQSIM's QUERY PROCESSOR

AQSim's query processor takes a user's query and the amount of time that the user is willing to wait for an answer. Then, while its running time is less than the user-defined time limit, the query processor searches the hierarchical partitions (which were made by the model generator) for those partitions that contain highly *precise* models for the given query.

$Precision(Q, model_j, partition_i)$ is defined to be the precision of the answer that $model_j$ of $partition_i$ would produce for the query, Q, as a *percentage of* $partition_i$*'s mesh topology*.[3] Specifically, $Precision(Q, model_j, partition_i) = Filled\_Volume(partition_i) \times P(Q, model_j, partition_i)$, where *Filled_Volume* returns the *percentage of non-empty space in the given partition's spatial bounding box* and is defined to be $Filled\_Volume(parent\_partition) =$

$$\frac{\sum_{child=1}^{\#\,of\,children}\left(Filled\_Volume(child\_partition)\times Volume(child\_partition)\right)}{Volume(parent\_partition)},$$

where $Filled\_Volume(leaf\_partition) = 1$.

$P(Q, model_j, partition_i)$ is calculated by using the *one-sided Chebyshev inequalities* [6], which are defined to be

- $P(X \leq \mu-\alpha) \leq \dfrac{\sigma^2}{\sigma^2+\alpha^2}$
- $P(X \geq \mu+\alpha) \leq \dfrac{\sigma^2}{\sigma^2+\alpha^2}$

X is a random variable with mean μ and variance σ^2. The variable α is a real number. The advantage of using the Chebyshev inequalities is that no assumption is made on the distribution of the data in a partition. For example, suppose we are given the query, *pressure* ≤ *0.5*. Then, for any partition, p (and $\alpha = \mu_{pressure} - 0.5$), the precision is equal to

$$Precision(pressure \leq 0.5, \text{mean modeler}, p) =$$
$$Filled\_Volume(p) \times P(pressure \leq 0.5) =$$
$$Filled\_Volume(p) \times P(pressure \leq \mu_{pressure} - \alpha) \leq$$
$$Filled\_Volume(p) \times \frac{\sigma^2_{pressure}}{\sigma^2_{pressure}+(\mu_{pressure}-0.5)^2}$$

For more complicated queries, we make new random variables and calculate mean and standard deviation values for them based on the original means and standard deviations. We assume independence between the original variables when calculating the means and standard deviations of the new random variables. For example, suppose we are given the query, $(temperature/pressure) \leq density$. This query is equivalent to $(temperature/pressure) - density \leq 0$. We create a new random variable, R, where $R = (temperature/pressure) - density$. So, our query is now $R \leq 0$. By calculating R's mean and standard deviation, we can use the aforementioned formula, namely $Precision(Q, model_j, partition_i) = Filled\_Volume(partition_i) \times P(Q, model_j, partition_i)$, to calculate our query's precision. Mean of R, μ_R, is equal to $E[(temperature/pressure) - density] = E[(temperature/pressure)] - E[density] =$

$$\left(E[temperature]\cdot\left(\frac{1}{E[pressure]}+\frac{Var[pressure]}{(E[pressure])^3}\right)\right)-E[density]$$

To get this formula, we assume that the random variables *temperature* and *pressure* are independent. Moreover, we use the formula $E[g(X)\cdot h(Y)] = E[g(X)]\cdot E[h(Y)]$, where X and Y are two

[3] The percentage of a partition's mesh topology corresponds to how well the partition's bounding box matches the underlying mesh topology.

independent random variables. The functions, g and h, are over X and Y, respectively. Finally, we use a lemma from Ross [9], which states the following:

> *"Let Z be a random variable having finite expectation μ and variance σ^2. Let g(•) be a twice differentiable function. Then $E[g(Z)] \approx g(\mu) + (g''(\mu)*0.5*\sigma^2)$."*

Similarly, we calculate the standard deviation for R, by using the equation $E[R^2] - (E[R])^2$ and assuming independence between *temperature*, *pressure*, and *density*. Due to space limitations, we have omitted the formula for R's standard deviation.

To get a single number representing the overall precision of our answer, we compute the *weighted average of the precisions* (which were calculated on all the explored partitions). In particular, the weighted average of the precisions is defined to be

$$\frac{\sum_{i=1}^{\#\,of\ explored\ partitions} \left(Precision(Q, m_j, p_i)*Volume(p_i)\right)}{\sum_{i=1}^{\#\,of\ explored\ partitions} Volume(p_i)},$$

where m_j is set to a particular model (*e.g.*, the mean modeler).

4. EXPERIMENTAL EVALUATION

4.1 The Can Data Set

Our first data set represents a wall crushing a can. It has 14 variables, 44 time steps, and 443,872 data points. The variables associated with this data set are: time, x axis, y axis, z axis, pressure, acceleration along each axis, velocity along each axis, and displacement along each axis. Figure 4 depicts this data set in its first time step when all the 440K points are plotted.

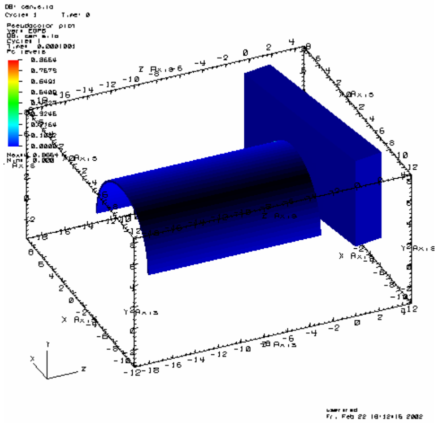

Figure 4. The Can Data Set at its First Time Step

Table 1 lists the compression results on the can data for the mean modeler. Recall that the partition threshold for this modeler restricts the distance between minimum and maximum values of a variable and its mean value with respect to RMSE.

For our mean modeler experiments, Figures 5 through 7 show the can data set at its first time step when the query *time > 0* is posed with no constraint on the query processor's execution time and with partition thresholds of 1.00, 2.00, and 3.00, respectively. As expected, we get better compression as the partition threshold for the mean modeler gets larger (since we are allowing the range of values for a variable to be larger). However, as you see in Figure 7 even with 82.6% compression, we are able to return a highly precise answer. The weighted average of all the precisions is 100% on the answer to the query, *time* > 0.

Table 1. Mean Modeler's Compression Results on the Can Data

Partition Threshold	% of Compression	Total # of partitions	% of non-leaf partitions	% of leaf partitions	Avg. # of data point in a partition
1.00	4.2	425,075	19.4	80.6	1.3
1.50	33.0	297,566	13.4	86.6	1.7
1.75	40.1	265,939	12.5	87.5	1.9
2.00	51.5	215,255	11.1	88.9	2.3
2.25	62.4	166,986	10.3	89.7	3.0
2.50	71.6	125,912	9.6	90.4	3.9
2.75	78.1	97,410	9.1	90.9	5.0
3.00	82.6	77,277	8.6	91.4	6.3

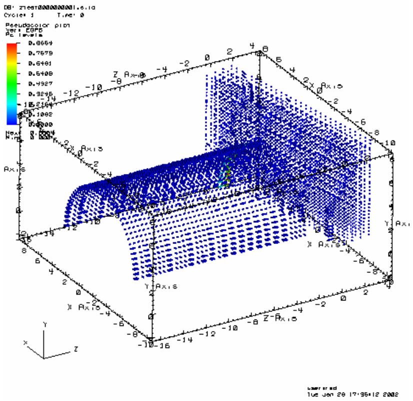

Figure 5. Can Data Set at its First Time Step with Partition Threshold = 1.00, Query = Time > 0

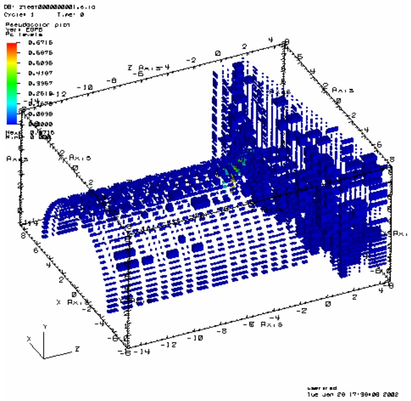

Figure 6. Can Data Set at its First Time Step with Partition Threshold = 2.00, Query = Time > 0

Table 2 lists the compression results on the can data for the goodness-of-fit modeler. The partition threshold in this table represents the confidence region of our normality test, which is equal to 100 × (1 – Type I error).

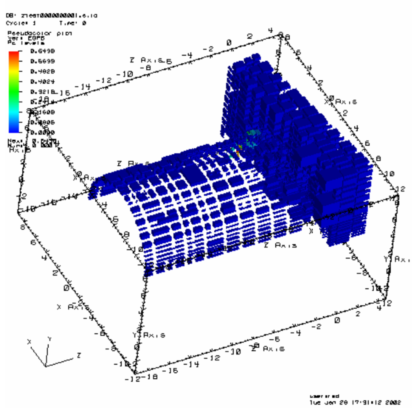

Figure 7. Can Data Set at its First Time Step with Partition Threshold = 3.00, Query = Time > 0

Table 2. Goodness-of-Fit Modeler's Compression Results on the Can Data

% Partition Threshold	% of Compression	Total # of partitions	% of non-leaf partitions	% of leaf partitions	Avg. # of data point in a partition
50.0	39.6	272,583	12.6	87.4	1.9
80.0	57.3	189,533	10.1	89.9	2.6
85	60.9	173,766	9.7	90.3	2.8
90.0	65.8	151,818	9.3	90.7	3.2
95.0	73.7	116,948	8.8	91.2	4.2
99.99	91.4	38,344	7.3	92.7	12.5

For our goodness-of-fit modeler experiments, Figures 8 through 10 show the can data set at its first time step when the query *time > 0* is posed with no constraint on execution time and with partition thresholds of 50%, 95%, and 99.99% respectively. Again not surprisingly, we get better compression as the partition threshold for the goodness-of-fit modeler gets larger (since the confidence region shrinks). However, as you see in Figure 10 even with 91.4% compression, we are able to return a highly precise answer. The weighted average of all the precisions is 100% on the answer to the query, *time* > 0.

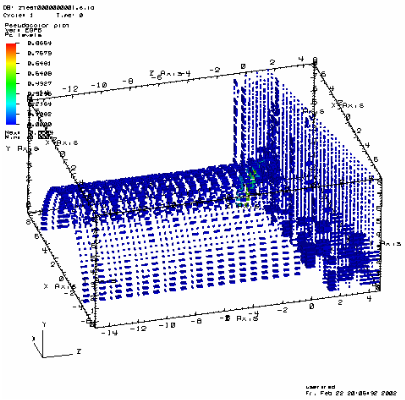

Figure 8. Can Data Set at its First Time Step with Partition Threshold = 50%, Query = Time > 0

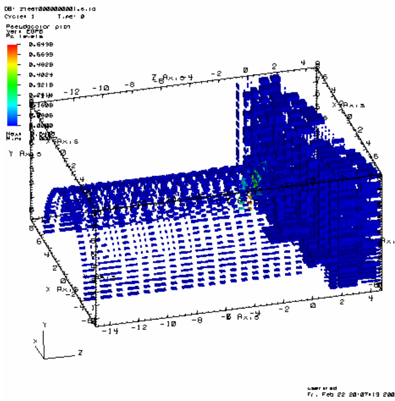

Figure 9. Can Data Set at its First Time Step with Partition Threshold = 95%, Query = Time > 0

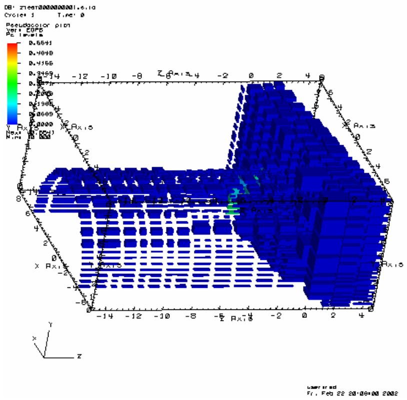

Figure 10. Can Data Set at its First Time Step with Partition Threshold = 99.99%, Query = Time > 0

The mean modeler achieves approximately 40% compression when the partition threshold is 1.75 (see Table 1). The goodness-of-fit modeler produces nearly the same level of compression with a partition threshold of 50% (see Table 2). Figures 8 and 11 illustrate the query processor's results on the query *time* > 0, for models built by the goodness-of-fit modeler (with threshold = 50%) and the mean modeler (with threshold = 1.75), respectively.

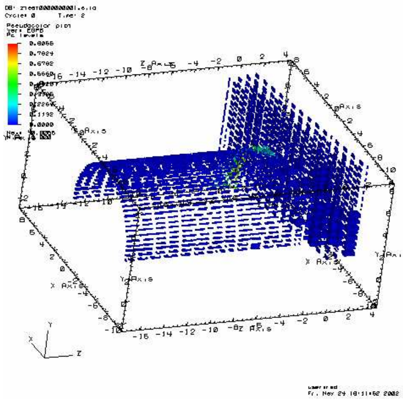

Figure 11. Can Data Set at its First Time Step with Partition Threshold = 1.75, Query = Time > 0

4.2 The Astrophysics Data Set

Our second data set represents a star in its mid-life. It has 18 variables, 16 time steps, and 1,708,852 zones. The variables associated with this data set are: time, *x* axis, *y* axis, *z* axis, distance, grid vertex values, grid movement along the *x* and *y* axes, d(energy)/d(temperature), density, electron temperature, temperature due to radiation, pressure, artificial viscosity, material temperature, material velocity along the *x*, *y*, and *z* axes. Figure 12 depicts this data set in its first time step when all the 1.7 million points are plotted.

Table 3 lists the compression results on the astrophysics data for the mean modeler. Again, recall that the partition threshold for this modeler restricts the distance between minimum and maximum of a variable and its mean value with respect to RMSE.

For our mean modeler experiments, Figure 13 shows the astrophysics data set at its first time step when the query *time > 0* is posed with no constraint on execution time and with partition thresholds of 3.00. Similar to our experiments on the can data set, we get better compression as the partition threshold for the mean modeler gets larger (since we are allowing the range of values for a variable to be larger). However, as you see even with 92.1% compression, we are able to return a highly precise answer. The weighted average of all the precisions is 100% on the answer to the query, *time* > 0.

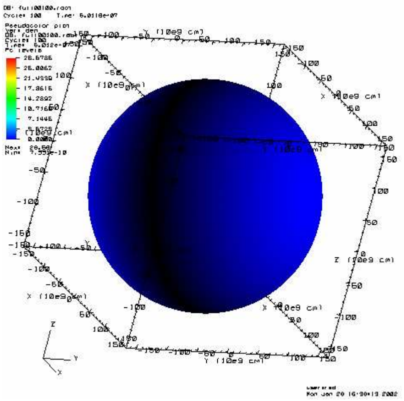

Figure 12. Astrophysics Data Set at its First Time Step

Table 3. Mean Modelers' Compression Results on the Astrophysics Data

Partition Threshold	% of Compression	Total # of partitions	% of non-leaf partitions	% of leaf partitions	Avg. # of data point in a partition
1.75	67.4	728,081	17.9	82.1	2.9
2.00	70.1	511,395	17.8	82.2	4.1
2.25	79.7	347,471	17.7	82.3	6.0
2.50	85.8	242,840	18.7	81.3	8.7
2.75	89.6	177,448	19.0	81.0	11.9
3.00	92.1	135,548	17.8	82.2	15.3

Table 4 lists the compression results on the can data for the goodness-of-fit modeler. Recall that the partition threshold in this table represents the confidence region of our normality test, which is equal to 100 × (1 – Type I error).

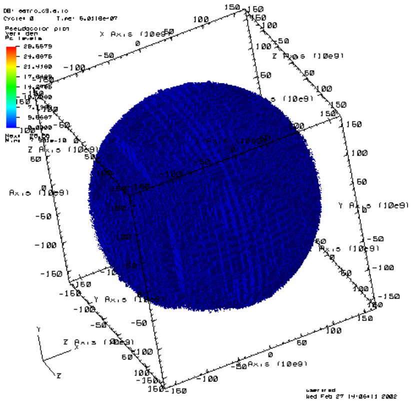

Figure 13. Astrophysics Data Set at its First Time Step with Partition Threshold of 3.00, Query Time > 0

Table 4. Goodness-of-Fit Modeler's Compression Results on the Astrophysics Data

% Partition Threshold	% of Compression	Total # of partitions	% of non-leaf partitions	% of leaf partitions	Avg. # of data point in a partition
80.0	66.7	564,718	16.8	83.2	3.6
85	71.2	492,029	16.7	83.3	4.2
90.0	76.4	404,136	16.9	83.1	5.1
95.0	82.8	293,585	16.8	83.2	7.0
99.99	94.3	97,819	13.3	86.7	20.2

For our goodness-of-fit modeler experiments, Figure 14 shows the astrophysics data set at its first time step when the query *time > 0* is posed with no constraint on execution time and with partition thresholds of 99.99%. Again not surprisingly, we get better compression as the partition threshold for the goodness-of-fit modeler gets larger (since the confidence region shrinks). However, as you see in Figure 14 even with 94.3% compression, we are able to return a highly precise answer. The weighted average of all the precisions is 100% on the answer to the query, *time* > 0.

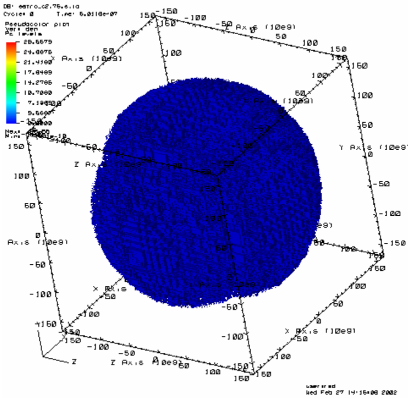

Figure 14. Astrophysics Data Set at its First Time Step with Partition Threshold of 99.99%, Query Time > 0

4.3 Discussion

Our experimental results illustrate the value of using simple statistical modeling techniques on scientific simulation data sets. Both of our approaches require only one sweep of the data and generate models that compress the data up to 94%.

The goodness-of-fit modeler performed better than the mean modeler on the two data sets presented in this paper. This is not surprising to us since our two data sets describe physical phenomena and the goodness-of-fit modeler is biased towards such normally distributed data sets. In general, we prefer the mean modeler since it makes no assumption on the data.

5. RELATED WORK

Our work is similar to Freitag and Loy's work at Argonne National Laboratory [7]. Their system builds distributed octrees from large scientific data sets. They, however, reduce their data by constraining the points to their spatial locations. They also do not allow the user to query the octree. Instead, the user can view the tree at different resolutions.

STING [10] is also similar to AQSim except that it assumes that the distribution of the data is known. It has been tested only on small data sets containing only tens of thousands of data points.

AQUA [2] uses cached summary data in an OLAP domain. Unfortunately, they use sampling and histogram techniques, which can remove outliers from data sets. In our experiences, outliers are very important to scientists. Moreover, histograms are computationally expensive on high-dimensional data sets.

6. CURRENT AND FUTURE WORK

We are investigating other modeling techniques for AQSim's model generator. Specifically, we are constrained to models that (*i*) require only one sweep of data, (*ii*) are good at finding outliers, (*iii*) can be easily parallelized, and (*iv*) can efficiently answer non-range queries (see [3])

We are also interested in *optimal* disk layout of the index tree. In particular, we are investigating techniques which will minimize seek time. In addition, parallelizing AQSim's query processor is part of our future work. Finally, we are conducting experiments on other larger data sets.

7. CONCULSION

To help scientists gather knowledge from their large-scale simulation data, we are developing the ad-hoc query infrastructure, AQSim. Our system consists of two components: (*i*) the model generator and (*ii*) the query processor. The model generator reduces the data storage requirements by creating and storing mathematical and statistical models of the data at multiple resolutions. The query processor decreases the query access times by evaluating queries on the models of the data instead of on the original data set. We describe two simple but effective statistical modeling techniques for simulation data. Our mean modeler computes the unbiased mean of systematic partitions of the data. It makes no assumptions about the distribution of the data and uses a variant of the root mean square error to evaluate a model. Our goodness-of-fit modeler utilizes the Andersen-Darling goodness-of-fit method on systematic partitions of the data. This modeler evaluates a model by how well it passes the normality test on the data. Both of our statistical modelers generate models that effectively answer range queries. At each resolution of the data, we calculate the precision of the query's answer by scaling the one-sided Chebyshev Inequalities with the original mesh's topology. We combine different precisions by computing their weighted average. Our empirical analyses on two scientific simulation data sets illustrate the value of using these statistical modeling techniques on large simulation data sets.

8. ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy by the University of California Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.1. UCRL-JC-147226-REV-1. Our thanks to Chuck Baldwin, Kevin Durrenberger, Roy Kamimura, Ed Smith, and Nu Ai Tang for their assistance.

9. REFERENCES

[1] Abdulla, G., Baldwin, C., Critchlow, T., Kamimura, R., Lozares, I., Musick, R., Tang, N.A., Lee, B., and Snapp, R. Approximate ad-hoc query engine for simulation data. In *Proceedings of JCDL 2001* (Roanoke VA, June 2001), ACM Press, 255-256.

[2] Acharya, S., Gibbsons, P.B., Poosala, V., and Ramaswamy, S. The Aqua approximate query answering system. In *Proceedings of the 1999 ACM SIGMOD*, ACM Press, 574-576.

[3] Baldwin, C., Abdulla, G., and Critchlow, T. Multi-Resolution Modeling of Large Scale Scientific Simulation Data. LLNL Technical Report, 2002.

[4] Chakrabarti, K., Garofalakis, M., Rastogi, R., and Shim, K. Approximate query processing using wavelets, In *Proceedings of VLDB 2000* (Cairo Egypt, September 2000), ACM Press, 111-122.

[5] D'Agostino, R.B., and Stephens, M.A. *Goodness-of-fit Techniques*, Marcel Dekker, Inc., 1986.

[6] Devore, J.L. *Probability and Statistics for Engineering and the Sciences*, 3rd edition. Brooks/Cole Publishing Company, Pacific Grove, CA, 1991.

[7] Freitag, L.A., and Loy, R.M. Adaptive, multi-resolution visualization of large data sets using a distributed memory octree. In *Proceedings of SC 1999* (Portland OR, November 1999), ACM Press, Article 60.

[8] Musick, R., and Critchlow, T. Practical lessons in supporting large-scale computational science, In *Proceedings of SIGMOD Record 1999*, ACM Press, 28(4):49-57.

[9] Ross, S. *A First Course in Probability*, 4th edition. Prentice Hall, Englewood Cliffs, NJ, 1994.

[10] Wang, W., Yang, J., and Muntz, R. STING: A statistical information grid approach to spatial data mining. In Proceedings of the VLDB (Athens Greece, August 1997), Morgan Kaufmann Publishers, 186-195.