Preprint
UCRL-JC-147226

Statistical Modeling of Large-Scale Simulation Data

T. Eliassi-Rad, T. Critchlow, G. Abdulla

This article was submitted to
The Eighth ACM SIGKDD International Conference of Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 23-26, 2002

February 22, 2002

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

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 shear size of these data sets has made efficient exploration of them impossible. To aid scientists in gathering knowledge 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 access times in two stages. First, it creates and stores mathematical and statistical models of the data. 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 highly effective statistical modeling techniques for simulation data. Our first modeling technique computes the true 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. In our second statistical modeling technique, we use the Andersen-Darling goodness-of-fit method on systematic partitions of the data. This second method evaluates a model by how well it passes the normality test on the data. Both of our statistical models summarize the data so as to answer range queries in the most effective way. We calculate precision on an answer to a query by scaling the one-sided Chebyshev Inequalities with the original mesh's topology. Our experimental evaluations on two scientific simulation data sets illustrate the value of using these statistical modeling techniques on 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

Scientific experiments ran on the latest super computers are producing large-scale simulation data. The size of these data sets is typically on the order of terabytes, which makes even the best 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][9].

In the *DataFoundry* Project, we have created a system, called *AQSim* (short for Ad-hoc Queries for Simulation). Figure 1 illustrates AQSim's two processors. The first processor (*a.k.a, model generator*) builds statistical and mathematical models of the data. Subsequently, the second processor (*a.k.a, query processor*) executes user queries on the generated models to explore the data set.

Since most scientific simulation code generate *mesh* data, AQSim uses data in mesh format to build its models. A mesh data set consists of interconnected grids of small zones (see Figure 2). Data points are stored in the zones. Mesh data sets usually vary with time, contain multiple dimensions (*i.e.*, variables), and have a huge number of irregular grids. Musick and Critchlow provide a nice introduction to scientific mesh data [3].

The main advantages of AQSim are two-folds. First, the model generator reduces the data storage requirements since models take less space than the original data set, which typically resides on tertiary storage. Second, the query processor decreases the access times since models of the data are queried.

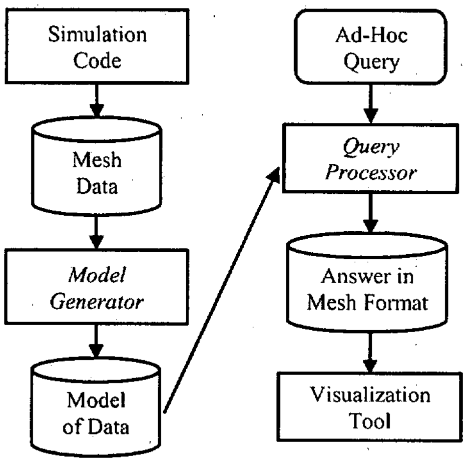

Figure 1. AQSIM Architecture

In this paper, we describe and evaluate two statistical modeling techniques for AQSim. The first model captures the true mean of systematic partitions of the data. We call this model the *mean modeler*. The error on this model is a variant of the *root mean square error* (RMSE). The main advantages of mean modeler are as follows: (*i*) it makes no assumptions about the distribution of the data, and (*ii*) it calculated its model parameters through one sweep of the data. 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*. The error on this model is the *Type I error* associated with the goodness-of-fit test.

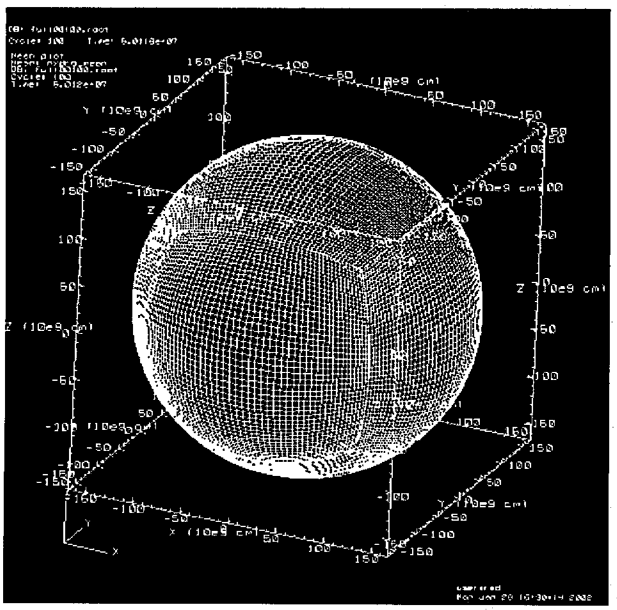

Figure 2. A Mesh Data Set Representing a Star

Despite their simplicity, these models have performed extremely well on our empirical studies of range queries. The answer to a query is judged by its precision to the original data. We calculate precision associated with a query's answer by scaling the one-sided *Chebyshev* inequality with the original mesh topology.

In the next section, we will describe our modeling algorithms. Then, we will present two case-studies, which illustrate the value of our modeling techniques, in section 3. We follow that with a discussion of some related, current, and future work. Finally, the paper concludes with some final remarks.

2. AQSIM'S MODEL GENERATOR

AQSim's model generator systematically partitions the data and builds models on each partition. This section describes two partitioning strategies and two statistical modeling techniques for AQSim.

2.1 Partitioning Strategies

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

AQSim has two distinct partitioning strategies. The first is a top-down approach, where the data is divided in a four-way bisection on the spatial-temporal space (see Figure 3). The second approach is a bottom-up strategy, where the data points are conglomerated based on their zones in the mesh topology (see Figure 4).

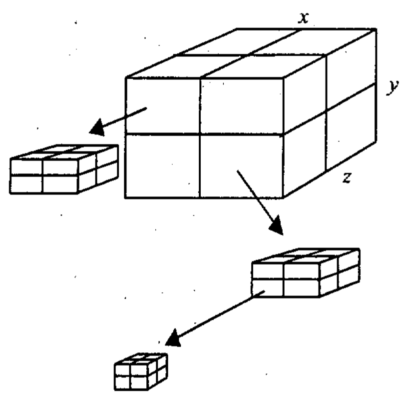

Figure 3. Top-Down Partitioning of the Data at a Particular Time Step

Table 1 summarizes the advantages of the two approaches. The bottom-up strategy is preferred over the top-down approach since it is cheaper computationally and it captures the mesh topology.

Table 1. Properties of AQSim's Partitioning Strategies

Strategies	Bottom-Up	Top-Down
Mesh Topology Captured	Yes	No
Computationally Expensive	No $O(N_{data} \times \log(N_{level}))$	Yes $O(N_{data} \times N_{level})$

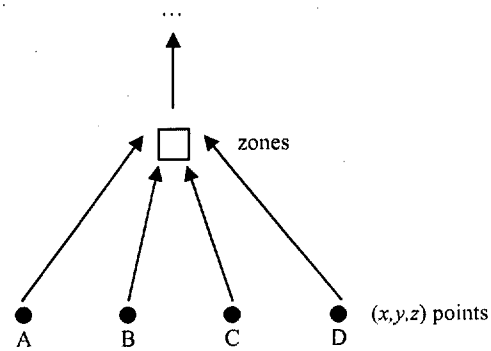

Figure 4. Bottom-Up Partitioning of the Data at a Particular Time Step

2.2 Mean Modeler

Each partition 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:

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

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. The *RMSE* represents the error associated with the mean modeler.

2.3 Goodness-of-Fit Modeler

For each variable v_i in partition p_k, the goodness-of-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, partitioning 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*. Otherwise, we accept H_0.

For each variable v_i, the error on this model is defined to be *Pr(reject H_0 / H_0 is true)*, where H_0 is the null hypothesis and 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 a value for the amount of time that the user is willing to wait for an answer. Then, while its running time is less than the user's constraint, the query processor searches the hierarchical partitions (which were made by the model generator) for the partitions that answer the user's query with the highest *precision*.

Precision(Q_{user}, 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_{user}, as a *percentage of partition$_i$'s mesh topology*. Specifically, *Precision(Q_{user}, model$_j$, partition$_i$)* = *(partition$_i \rightarrow$filled_volume)* × *P(Q_{user}, model$_j$, parition$_i$)*, where *filled_volume* corresponds to the *percentage of non-empty space in the partition's spatial bounding box* and is defined to be

$$filled\_volume_{parent} = \frac{\sum_{child=1}^{\#\text{ of children}}(filled\_volume_{child} \times volume_{child})}{volume_{parent}}$$

P(Q_{user}, model$_j$, node$_i$) is calculated by using the *one-sided Chebyshev inequalities* [6], which are defined to be

- $P(X \le \mu - \alpha) \le \frac{\sigma^2}{\sigma^2+\alpha^2}$
- $P(X \ge \mu + \alpha) \le \frac{\sigma^2}{\sigma^2+\alpha^2}$

Here is a simple example of how precision is calculated. Suppose we are given the following query, *pressure* $\le$ *0.5*. Then, for a partition, say p, the precision is equal to

Precision(pressure $\le$ 0.5, mean modeler, p) = *($p\rightarrow$filled_volume)* × *P*(pressure $\le$ 0.5) = *($p\rightarrow$filled_volume)* × *P*(pressure $\le \mu_{pressure} - \alpha) \le$ *($p\rightarrow$filled_volume)* × $\frac{\sigma^2_{pressure}}{\sigma^2_{pressure}+(\mu_{pressure}-0.5)^2}$, where $\alpha = \mu_{EQPS} - 0.5$.

For more complicated queries, we make new random variables and calculate mean and standard deviations for them based on the original mean and standard deviations. The advantage of using the Chebyshev inequalities is that no assumption is made on the distribution of the data in a node.

4. EXPERIMENTAL EVALUATION

4.1 The Can Data Set

Our first data set represents a wall crushing a can. It has 14 variables,[1] 44 time steps, and 443,872 data points. Figure 5 depicts this data set in its first time step when all the points 440K points are plotted.

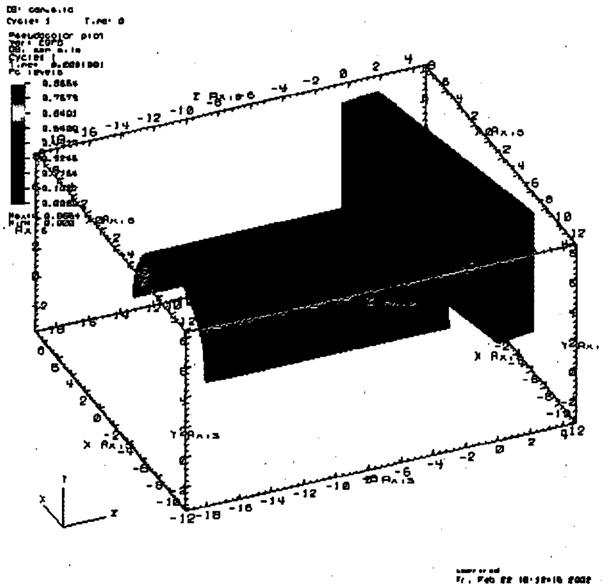

Figure 5. Can Data Set at its First Time Step

Table 2 lists the compression results on the can data for the mean modeler. Recall that for 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.

Table 2. 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

[1] The can data set's variables are as follows: time, x axis, y axis, z axis, pressure, acceleration in x axis, acceleration in y axis, acceleration in z axis, velocity in x axis, velocity in y axis, velocity in z, displacement in x axis, displacement in y axis, displacement in z axis.

For our mean modeler experiments, Figures 6 through 8 show the can data set at its first time step when the query *time > 0* is posed with no constraint on execution time (that is precision equals 100%) 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 me larger). However, as you see in Figure 8 even with 82.6% compression, we are able to return an answer with 100% precision.

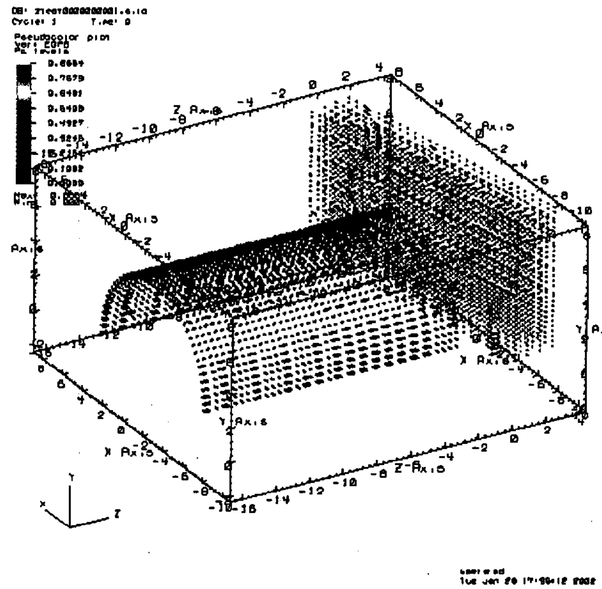

Figure 6. Can Data Set at its First Time Step with Partition Threshold of 1.00, Query Time > 0, and Precision = 100%

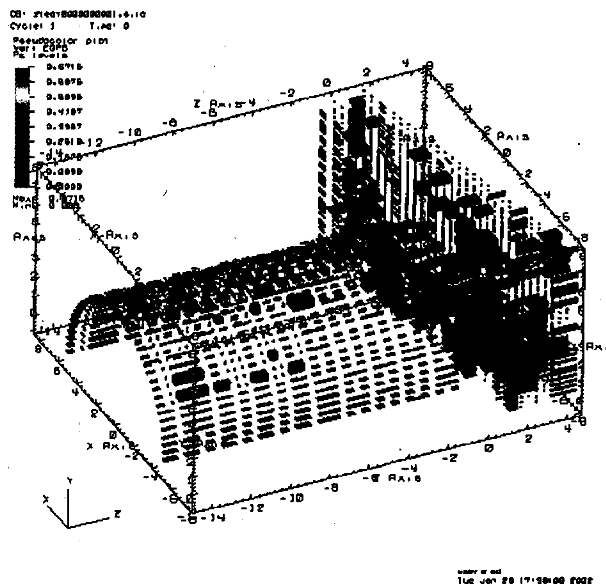

Figure 7. Can Data Set at its First Time Step with Partition Threshold of 2.00, Query Time > 0, and Precision = 100%

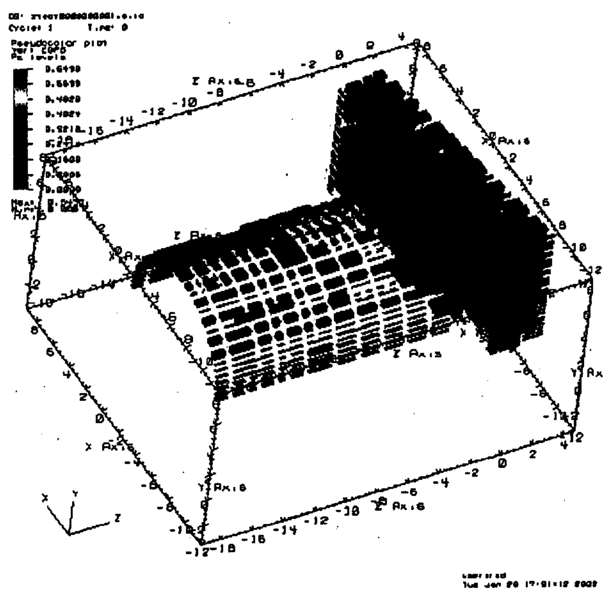

Figure 8. Can Data Set at its First Time Step with Partition Threshold of 3.00, Query Time > 0, and Precision = 100%

Table 3 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 $100 \times (1 - \text{Type I error})$.

Table 3. 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 9 through 11 show the can data set at its first time step when the query *time > 0* is posed with no constraint on execution time (that is precision equals 100%) and with partition thresholds of 99.99%, 95%, and 50% 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 11 even with 91.4% compression, we are able to return an answer with 100% precision.

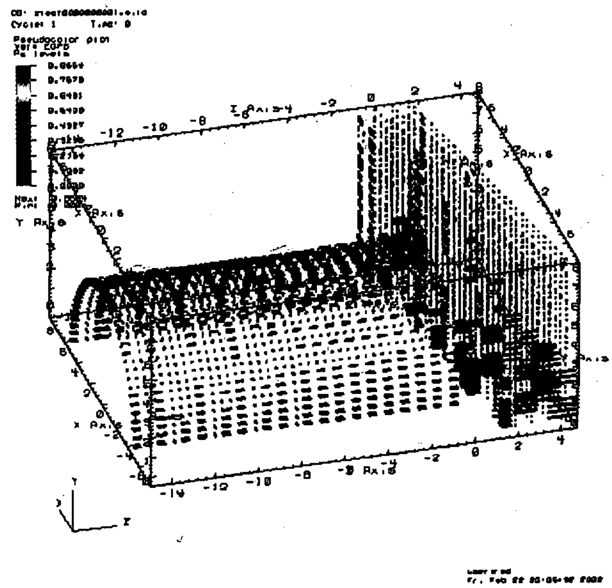

Figure 9. Can Data Set at its First Time Step with Partition Threshold of 50%, Query Time > 0, and Precision = 100%

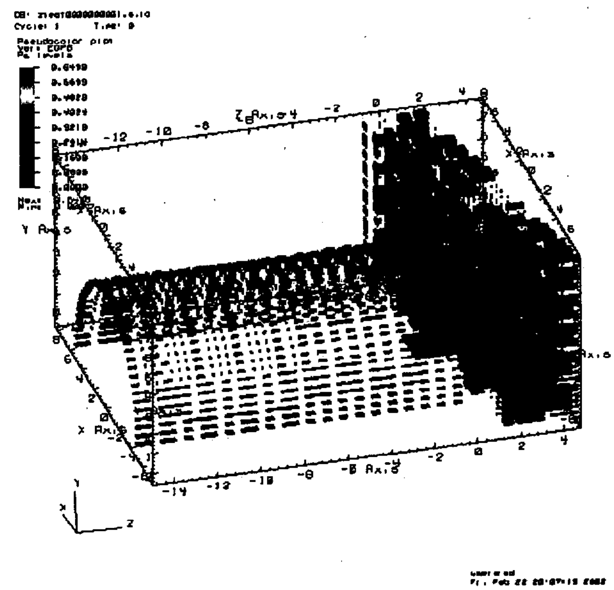

Figure 10. Can Data Set at its First Time Step with Partition Threshold of 95%, Query Time > 0, and Precision = 100%

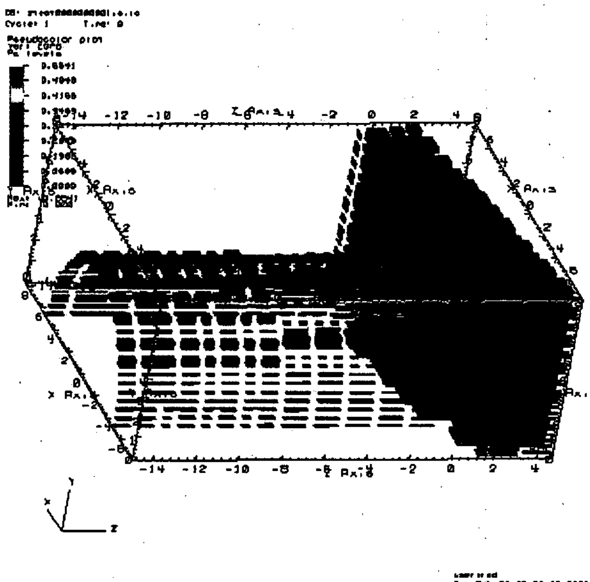

Figure 11. Can Data Set at its First Time Step with Partition Threshold of 99.99%, Query Time > 0, and Precision = 100%

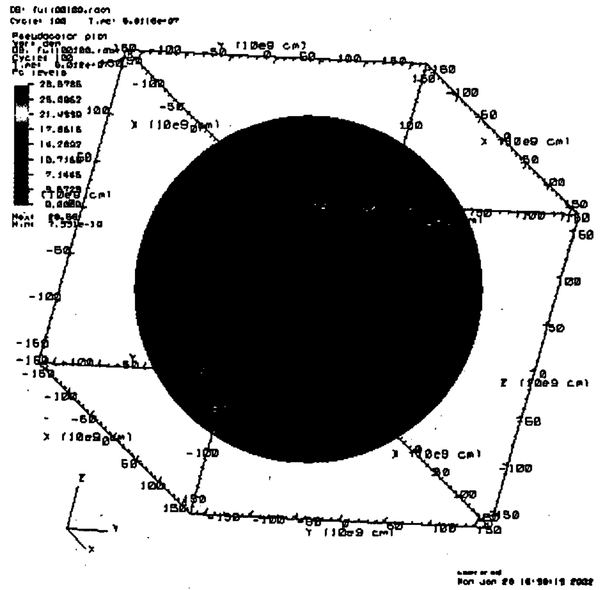

Figure 12. Astrophysics Data Set at its First Time Step

4.2 The Astrophysics Data Set

Our second data set represents a star in its mid-life. It has 18 variables,[2] 16 time steps, and 1,708,852 zones. Figure 12 depicts this data set in its first time step when all the points 1.7 million points are plotted.

Table 4 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 can data set at its first time step when the query *time > 0* is posed with no constraint on execution time (that is precision equals 100%) 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 me larger). However, as you see even with 92.1% compression, we are able to return an answer with 100% precision.

Table 4. 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

[2] The astrophysics data set's variables are as follows: time, x axis, y axis, z axis, distance, grid vertex values, grid movement in x axis, grid movement in y axis, d(energy)/d(temperature), density, electron temperature , temperature due to radiation, pressure, artificial viscosity, material temperature, material velocity in x axis, material velocity in y axis, material velocity in z axis.

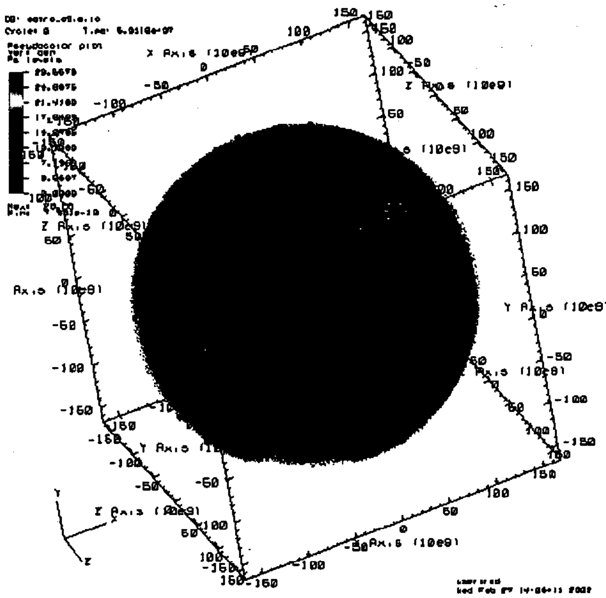

Figure 13. Astrophysics Data Set at its First Time Step with Partition Threshold of 3.00, Query Time > 0, and Precision = 100%

Table 5 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 $100 \times (1 - \text{Type I error})$.

Table 5. 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, Figures 14 shows the astrophysics data set at its first time step when the query *time > 0* is posed with no constraint on execution time (that is precision equals 100%) 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 an answer with 100% precision.

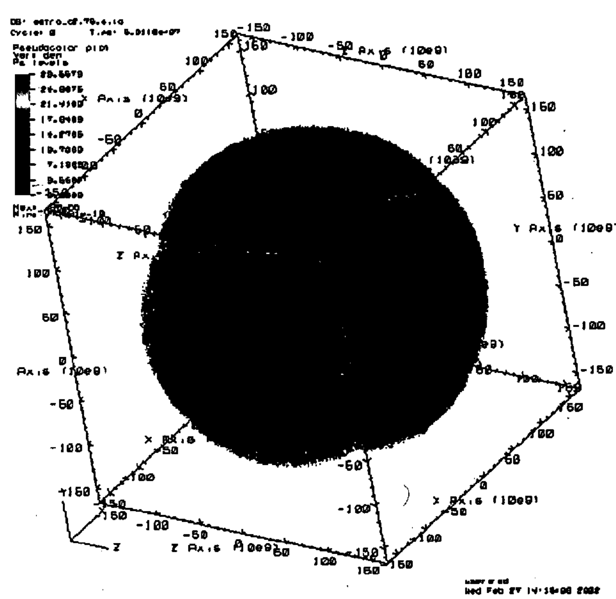

Figure 14. Astrophysics Data Set at its First Time Step with Partition Threshold of 99.99%, Query Time > 0, and Precision = 100%

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 don't allow the user to query the octree. Instead, the user can view the tree at different resolutions.

STING [9] is also similar to AQSim except that it assumes that the distribution of the data is known. It has also 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 using sampling and histogram techniques, which are not good for scientific data sets because by sampling you might miss outliers (which are important in scientific data sets) and histograms are computationally expensive when you have high dimensional data.

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. Moreover, parallelizing AQSim's query processor is also part of our future work. Finally, we are conducting experiments on other larger data sets.

7. CONCULSION

To help scientists in gathering knowledge from their large-scale simulation data, we have developed an ad-hoc query infrastructure, called AQSim. Our system reduces the data storage requirements and access times in two stages. First, it creates and stores mathematical and statistical models of the data. 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 highly effective statistical modeling techniques for simulation data. Our first modeling technique computes the true 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. In our second statistical modeling technique, we use the Andersen-Darling goodness-of-fit method on systematic partitions of the data. This second method evaluates a model by how well it passes the normality test on the data. Both of our statistical models summarize the data so as to answer range queries in the most effective way. We calculate precision on an answer to a query by scaling the one-sided Chebyshev Inequalities with the original mesh's topology. Our experimental evaluations on two scientific simulation data sets illustrate the value of using these statistical modeling techniques on large simulation data sets.

8. ACKNOWLEDGMENTS

Our thanks to Chuck Baldwin, Kevin Durrenberger, Roy Kamimura, Ed Smith, and Nu Ai Tang for their useful comments and 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 Sciencfes*, 3rd edition. Brooks/Cole Publishing Company, Pacific Grove, CA, 1991.

[7] Freitag, L.A., and Loy, R.M. Adaptive, multiresolution 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] 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.