The Astrophysical Journal, 654:2–11, 2007 January 1

© 2007. The American Astronomical Society. All rights reserved. Printed in U.S.A.

PICO: PARAMETERS FOR THE IMPATIENT COSMOLOGIST

William A. Fendt[1] and Benjamin D. Wandelt[1,2,3,4]

Received 2006 June 28; accepted 2006 August 4

ABSTRACT

We present a fast, accurate, robust, and flexible method of accelerating parameter estimation. This algorithm, called Pico, can compute the CMB power spectrum and matter transfer function, as well as any computationally expensive likelihoods, in a few milliseconds. By removing these bottlenecks from parameter estimation codes, Pico decreases their computational time by 1 or 2 orders of magnitude. Pico has several important properties. First, it is extremely fast and accurate over a large volume of parameter space. Furthermore, its accuracy can continue to be improved by using a larger training set. This method is generalizable to an arbitrary number of cosmological parameters and to any range of l-values in multipole space. Pico is approximately 3000 times faster than CAMB for flat models, and approximately 2000 times faster than the *WMAP* 3 yr likelihood code. In this paper, we demonstrate that using Pico to compute power spectra and likelihoods produces parameter posteriors that are very similar to those using CAMB and the official WMAP3 code, but in only a fraction of the time. Pico and an interface to CosmoMC are made publicly available on the authors' Web site at http://www.astro.uiuc.edu/~bwandelt/pico/.

Subject headings: cosmic microwave background — cosmology: observations — methods: numerical

Online material: color figures

1. INTRODUCTION

With the *Wilkinson Microwave Anistropy Probe* (*WMAP*) second data release (Hinshaw et al. 2006; Page et al. 2006; Spergel et al. 2006), there is a wealth of new cosmic microwave background (CMB) data available to further constrain cosmological parameters. The major computational burden in parameter estimation remains the calculation of the theoretical power spectrum for a large number of cosmological models, as well as the likelihood based on these spectra. Generally, the power spectrum is computed with codes such as CMBfast (Seljak & Zaldarriaga 1996) or CAMB (Lewis et al. 2000), which evolve the Boltzmann equation using a line-of-sight integration approach. While this provides a 1 or 2 order of magnitude decrease in the computation time over the full Boltzmann codes, power spectrum calculations remain a bottleneck of parameter estimation. Other software such as CMBwarp (Jimenez et al. 2004) and DASh (Kaplinghat et al. 2002) have found ways to improve the efficiency of power spectrum calculations at the cost of a loss of accuracy against the full Boltzmann codes and/or by placing restrictions on the parameters that are available as input. In particular, CMBwarp builds on the method introduced in Kosowsky et al. (2002), where a new set of nearly uncorrelated "physical" parameters were defined that have nearly independent effects on the power spectrum. CMBwarp uses a modified polynomial fit whose coefficients are based on a fiducial model. It allows rapid calculation of the temperature (TT), E-mode polarization (EE), and temperature-polarization (TE) cross-power spectra. CMBwarp, however, requires the use of specific cosmological parameters, and the accuracy of the computed power spectra quickly diminishes as one moves away from the fiducial model in parameter space. Another code, CMBFit (Sandvik et al. 2004), attempts to avoid the need to compute the power spectrum by fitting the likelihood function. This idea is particularly important for the *WMAP* 3 yr data (Hinshaw et al. 2006; Page et al. 2006; Spergel et al. 2006), whose likelihood is time consuming to compute.

In this paper we introduce Pico, a computational technique to accelerate both power spectrum and likelihood computations. This approach removes the two major bottlenecks in parameter estimation. While in a spirit similar to CMBwarp and providing a speed-up similar to CMBfast and CAMB, Pico has several important advantages over CMBwarp and DASh. First, it allows the calculation of power spectra from an arbitrary number of cosmological parameters and in any range of l-values in multipole space. Because of this flexibility, it is easily incorporated into parameter estimation codes. Second, Pico allows the simultaneous computation of all scalar, tensor, and lensed power spectra, as well as the transfer functions. Pico provides more than an order of magnitude increase in accuracy over CMBwarp and about 2 orders of magnitude increase in speed over DASh. Finally, Pico is generic enough to allow the direct fitting of any likelihood functions. Due to the computational expense in computing the likelihood of certain experiments, e.g., WMAP3, any power spectrum acceleration scheme will at most provide a speed-up of order 1–10 in parameter estimation. However, using Pico to also compute the likelihood results in speed-ups of order 10–100. As an additional bonus, using Pico to compute the likelihood directly provides more accurate results than using it to fit the power spectra and computing the likelihood from these approximate spectra. This is important for current and next-generation all-sky CMB data. Meanwhile, the power spectra computed by Pico are more than accurate enough for suborbital experiments with smaller sky coverage and coarser l-resolution.

This paper is organized as follows. We present a brief overview of Pico in § 2 and examine its CPU and memory requirements in § 3. Section 4 presents several tests of the performance of Pico. This includes comparisons of power spectra computed using Pico and CAMB, as well as results of parameter estimation runs using Pico to compute the power spectra and the WMAP3

[1] Department of Physics, University of Illinois at Urbana-Champaign, Urbana, IL; fendt@uiuc.edu.

[2] Department of Astronomy, University of Illinois at Urbana-Champaign, Urbana, IL; bwandelt@uiuc.edu.

[3] Center for Advanced Studies, University of Illinois at Urbana-Champaign, Urbana, IL.

[4] Center for Advanced Studies Beckman Fellow.

likelihood. In § 5 we summarize and discuss the future of Pico. The details of the algorithm used by Pico are presented in the Appendix.

2. OVERVIEW OF THE ALGORITHM

Pico computes CMB power spectra, matter transfer functions, and likelihoods as a function of cosmological parameters by interpolating a precomputed training set. Given this training set, Pico first clusters the points in parameter space into nonoverlaping regions. Within each of these clusters, Pico fits the power spectra, matter transfer function, and likelihood using a multivariate polynomial chosen to minimize the squared error over the training set. In general, Pico does not need to distinguish between the multipole l-values of the power spectra, the k-values of the transfer function, or any likelihood. These values can be combined into a single vector and Pico will try to fit the individual components. Working in this joint space makes it possible to compress the dimension of the space, thereby requiring Pico to fit fewer components, resulting in a smaller memory footprint and faster computations. In practice, it is useful to separate the likelihood computations from the power spectra computations. Since the power spectra and transfer functions have a simple scaling dependence on the initial scalar and tensor amplitudes, it is not necessary to use Pico to interpolate in these parameters. This, however, is not true of the likelihood, which has a complicated dependence on the amplitudes. Similarly, if one wishes to fit the lensed power spectra, it will be necessary to interpolate the amplitude parameters, as there is no longer a simple scaling relation. In both cases, however, the algorithm is *exactly* the same; there is simply a different number of parameters that must be interpolated. A detailed description of our algorithm is presented in the Appendix.

3. CPU AND MEMORY REQUIREMENTS

The quantities that determine the CPU and memory requirements of the algorithm are the number of clusters n, the number of cosmological parameters $\mathcal{N}_x$, the number of l-values and compressed l-values $\mathcal{N}_y$ and $\mathcal{N}_y'$, and the order of the regression polynomial p. Each computation of the power spectra has (approximately) no dependence on the number of clusters, since we only need to determine which cluster the input parameters are in. This is found after n fast distance calculations. The power spectrum is then calculated using the polynomial in the cluster. This takes $2\mathcal{N}\mathcal{N}_y'$ computations where $\mathcal{N}$, the number of polynomial coefficients, is given by

$$\mathcal{N} = \frac{(\mathcal{N}_x + p)!}{\mathcal{N}_x \mathcal{P}} \sim \mathcal{O}\left(\left[\frac{\mathcal{N}_x}{p}\right]^p\right) \quad \text{for } \mathcal{N}_x \gg p \gg 0.$$

After evaluating the polynomial, another $\mathcal{N}_y \mathcal{N}_y'$ calculations are needed to uncompress the spectrum. It is thus possible to calculate the power spectrum with very few computations. For the seven-parameter case we examine in § 4.1, calculation of the power spectra takes approximately 3 ms on a 2 GHz Intel Pentium M processor. This is roughly 3000 times faster than CAMB for flat models and 15,000 times faster for nonflat models. For parameter estimation codes such as CosmoMC (Lewis & Bridle 2002), this speed-up is significantly more than is necessary, since evaluation of the likelihood quickly becomes the new bottleneck. However, as we have noted, Pico removes this bottleneck as well by fitting the (computationally intensive) likelihoods. Furthermore, other techniques such as Gibbs sampling (Wandelt et al. 2004; Chu et al. 2005), which have quicker likelihood evaluations, continue to benefit from a significant speed-up in computing the power spectrum.

The main memory use of the algorithm is in holding the $n\mathcal{N}\mathcal{N}_y'$ regression coefficients. For the example we present in § 4 using fourth-order polynomials in seven parameters ($\mathcal{N} = 330$) over 100 clusters and 60 compressed l-values, this is approximately 15 MB of information. If fitting of the scalar and tensor modes out to $l = 3000$, as well as the transfer function, is included, this number could increase by an order of magnitude. Even this can be accommodated by any modern personal computer.

Given a training set, the two major parameters governing the algorithm are the order of the polynomial p and the number of clusters n. Generally, polynomial regression schemes will eventual run into hurdles as the order of the polynomial is increased. For one, the increased order greatly increases the number of terms in the polynomial, so a sufficiently large training set is needed to ensure the regression problem is well defined. Furthermore, higher order terms may result in large regression coefficients and rely on fine-tuned cancellations between these large terms. In this situation numerical stability becomes an issue. In the cases studied in this paper, we have found that fourth-order polynomials remain stable while providing excellent fits of the power spectra. Given the polynomial order, one would like to maximize the number of clusters while ensuring that within each cluster there is a sufficient number of training set points to provide accurate interpolation. In general, one can generate a test set in the same fashion as the training set and use it to tune these parameters. Lastly, note that the number of compressed l-values $\mathcal{N}_y'$ can be chosen by simply running the compression over the training set, uncompressing the results, and computing the errors for varying values of $\mathcal{N}_y'$. This is done explicitly in § A3 for the test case presented in § 4.1.

4. RESULTS

In this section we provide several tests of the performance of Pico both in its ability to compute power spectra and in the results of parameter estimation runs. The first two tests compare the power spectra computed using Pico with those computed using CAMB for seven- and nine-parameter models. Next we compare the results of a parameter estimation run using Pico and CAMB to compute the power spectrum and the first-year *WMAP* code (Bennett et al. 2003) to compute the likelihood. Finally, we compare parameter estimation runs using Pico to compute the likelihood with runs using CAMB and the official *WMAP* 3 yr likelihood code. In all cases Pico produces parameter posteriors that are in good agreement with CAMB, both for the larger parameter space allowed by WMAP1 and the higher precision required by WMAP3.

4.1. *Power Spectrum Calculation for Seven-Parameter Models*

Here we compare the performance of Pico with CAMB and CMBwarp. To generate our test set we begin with a converged Markov chain Monte Carlo (MCMC) run consisting of ~60,000 cosmological models based on *WMAP* first-year (Bennett et al. 2003) and other CMB data, including CBI (Padin et al. 2001), BOOMERANG (Ruhl et al. 2003), ACBAR (Kuo et al. 2004), VSA (Grainge et al. 2003), MAXIMA (Hanany et al. 2000), DASI (Halverson et al. 2002), and TOCO (Miller et al. 1999). The varied parameters are the baryon density Ω_b, the cold dark matter density $\Omega_{\rm CDM}$, the dark energy density Ω_Λ, Hubble's constant H_0, the scalar spectral index n_s, the optical depth since reionization τ, and the normalization of the power spectra $\mathcal{A}_s$. Next we convert each point in this space to the physical parameters introduced by

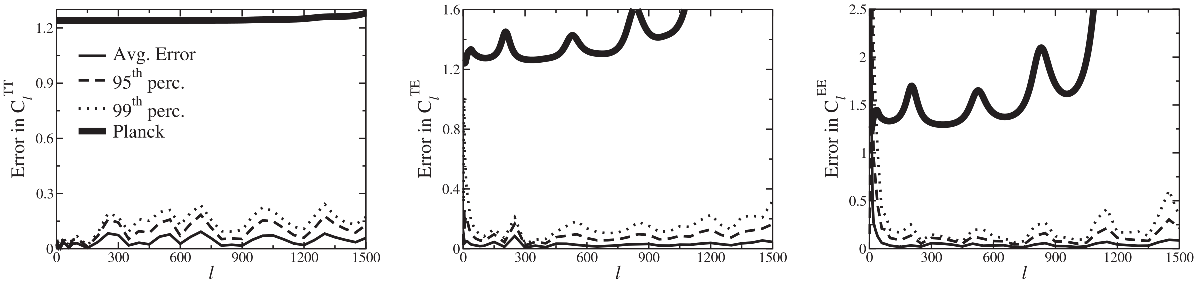

FIG. 1.—Performance of Pico relative to CAMB for seven-parameter models. The three lines denote the average error and the 95th and 99th percentiles over the 10^4 models in the test set. The thick solid line is the expected uncertainty from *Planck* data assuming $f_{\rm sky} = 0.65$. The error is plotted in units of the cosmic standard deviation. [*See the electronic edition of the Journal for a color version of this figure.*]

Jimenez et al. (2004) and Kosowsky et al. (2002). In the physical parameter space there is significantly less correlation in the set of points. We then calculate the mean and variance of this set, and use it to generate an eight-dimensional box in physical parameter space whose sides are of length 3 σ in each direction. Our test set consists of 10^4 models sampled *uniformly* from this box. That is, we in no way bias the test set by weighting points in parameter space based on their likelihoods. The physical parameters are converted back to cosmological parameters and used to run CAMB to give the scalar TT, TE, and EE power spectra. This set of 10^4 models and their corresponding power spectra form our test set.

The performance of Pico is shown in Figure 1. In this example, we ran Pico with fourth-order polynomials over 100 clusters. We have plotted the error in units of the cosmic standard deviation as a function of the multipole l-value for the TT, TE, and EE power spectra. The error in a single computed spectrum is defined as

$$\Delta_l = \frac{|C_l - C_l^{\rm CAMB}|}{\sigma_l^{\rm CV}},$$

where $\sigma_l^{\rm CV}$ is the cosmic standard deviation. The three lines denote the average of this error over the test set and the error that bounds 95% and 99% of the test set (the 95th and 99th percentiles). The thick solid line in each plot denotes the expected uncertainty in data from the *Planck* satellite mission. Here we have assumed 65% of the sky will remain uncontaminated by foregrounds, and we have combined the three frequency bands from the Low-Frequency Instrument (LFI) and the three lowest frequency bands from the High-Frequency Instrument (HFI) according to the method described by Zaldarriaga et al. (1997) and Kinney (1998).

For 99% of the models in our test set, Pico is able to calculate the TT spectrum with an error less than 0.3 $\sigma^{\rm CV}$, the TE spectrum with an error less than 0.4 $\sigma^{\rm CV}$, and the EE spectrum to better than 0.7 $\sigma^{\rm CV}$ for l out to 1500. For the TE and EE spectra this excludes very low l, where the magnitudes of the power spectra and cosmic variance become small. This is better than what will be achievable from even the *Planck* satellite mission. We note that the points with the largest error bars are near the edges of our training set and correspond to models that are highly disfavored even by CMB data alone.

In Figure 2, we have plotted the performance of CMBwarp against CAMB over the same test set. The four lines denote the average and 99th percentile from CMBwarp as well as the average and 99th percentile using our code. We note that Pico is significantly more accurate over all l for the three power spectra. Pico gives more than an order of magnitude increase in accuracy over CMBwarp, while providing a similar decrease in the time required to compute a power spectrum as compared with CAMB.

4.2. *Power Spectrum Calculation for Nine-Parameter Models*

As a second test of Pico, we calculate the scalar TT, TE, and EE spectra as a function of nine parameters. The training set was formed using the seven parameters as described in § 4.1, in addition to the dark energy equation of state and the running of the scalar spectral index. The new parameters were drawn uniformly from the intervals $[-1, -0.78]$ and $[-0.085, 0]$, respectively. Figure 3 shows the performance compared with CAMB. In this example Pico was run with fourth-order polynomials over 100 clusters.

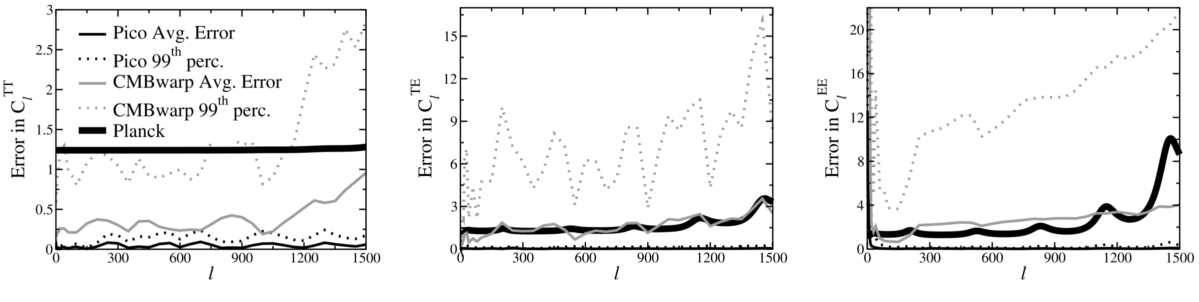

FIG. 2.—Performance of Pico and CMBwarp relative to CAMB. The four lines denote the average error and 99th percentile for CMBwarp and the average error and 99th percentile using Pico. The thick solid line is the expected uncertainty from *Planck* data. The error is plotted in units of the cosmic standard deviation. The Pico errors are so small that at times they do not exceed the thickness of the l-axis. [*See the electronic edition of the Journal for a color version of this figure.*]

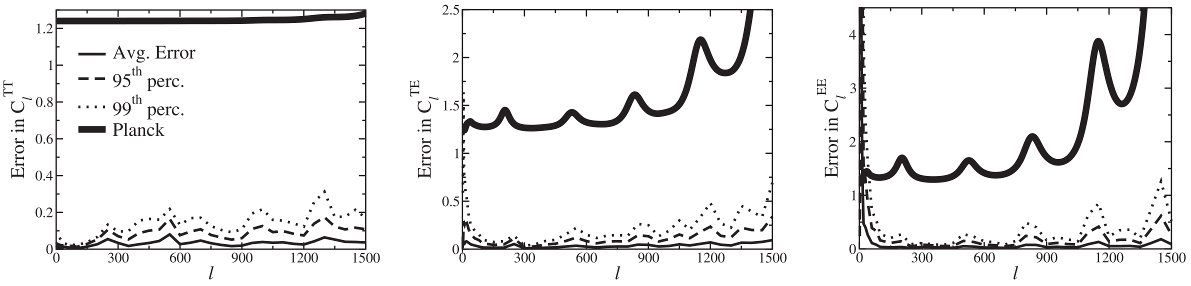

FIG. 3.— Performance of Pico relative to CAMB for nine-parameter models. The three lines denote the average error and the 95th and 99th percentiles. The thick solid line is the expected uncertainty from *Planck* data. The error is plotted in units of the cosmic standard deviation. [*See the electronic edition of the Journal for a color version of this figure.*]

While even at this level the accuracy is better than what will be achievable with *Planck*, one could continue to decrease the error by using a larger training set to allow the use of more clusters.

4.3. *Parameter Posteriors Using Pico to Compute Power Spectra*

We have incorporated Pico into the publicly available parameter estimation code CosmoMC (Lewis & Bridle 2002). The interface allows CosmoMC to use Pico to compute the theoretical power spectrum and transfer function, as well as the WMAP3 likelihood, whenever the parameters are within the range over which Pico's regression coefficients are defined. For parameters outside this range, CosmoMC will continue to use CAMB to compute the power spectrum or the WMAP3 code to compute the likelihood.

In this section we compare the posteriors over the parameters computed by CosmoMC while using CAMB or Pico to compute the theoretical power spectrum. The likelihoods were computed using the *WMAP* first-year data and likelihood function. (Verde et al. 2003; Hinshaw et al. 2003; Kogut et al. 2003). For this test we choose flat models and varied six parameters: $\Omega_b h^2$, $\Omega_{\rm CDM} h^2$, θ, τ, n_s, and the power spectrum amplitude $\mathcal{A}_s$. We ran CosmoMC using Pico for 500,000 steps. CAMB was needed for less than 1% of the models. This took approximately 15 hr. The posterior and mean likelihood over each parameter are shown in Figures 4 and 5, respectively. We have also plotted the posterior and mean likelihood from a 500,000 step run of CosmoMC using only CAMB, which took approximately 160 hr. The posteriors agree quite well, especially near the peaks. In every parameter except τ, the mean of the posteriors differ by less than 0.7%. For τ, which is poorly constrained by this data set, the mean of the posteriors differ by 3.7%. The errors in the likelihood evaluations from Pico are more apparent in Figure 5, as the mean likelihood

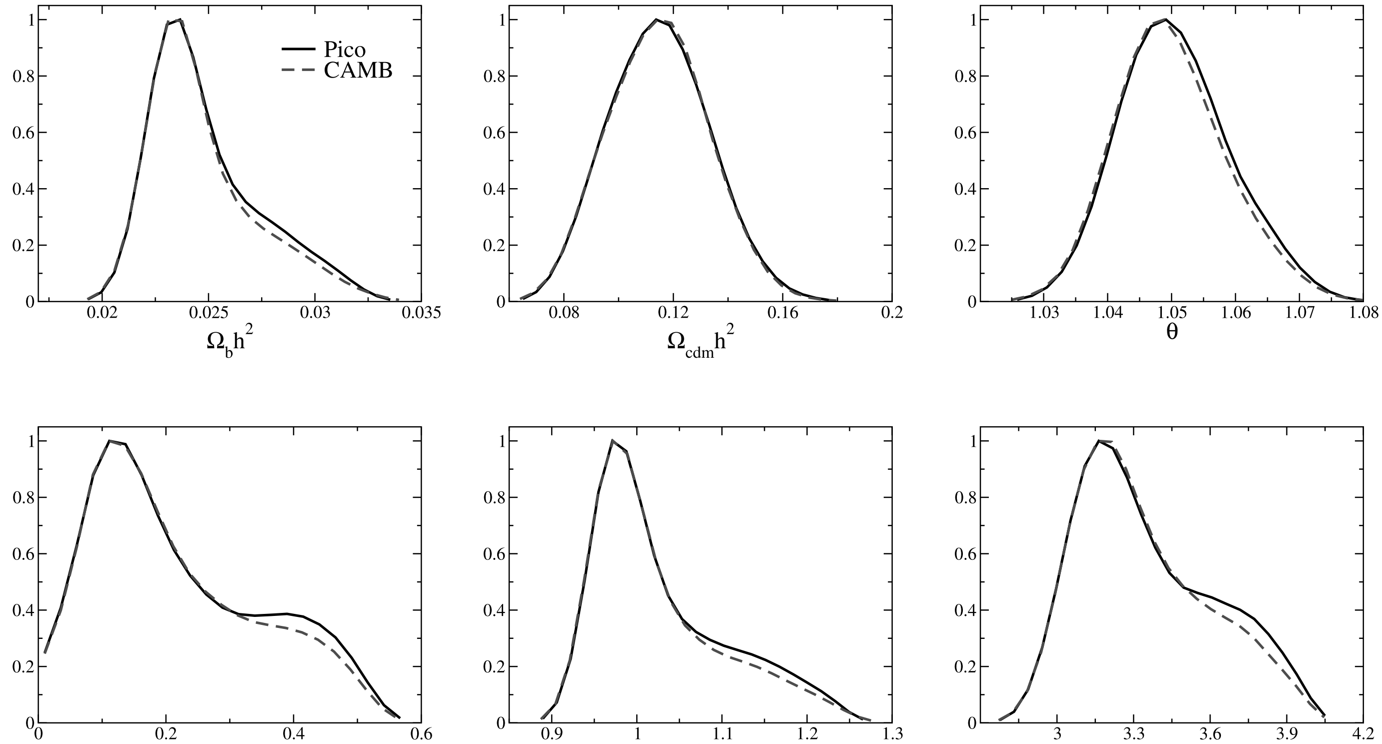

FIG. 4.— One-dimensional posterior constraints on the cosmological parameters using runs of CosmoMC with Pico and CAMB. The likelihoods were computed using the first-year *WMAP* data and likelihood code based on the Pico power spectra (*solid line*) and the CAMB power spectra (*dashed line*). Using Pico decreased the time to compute the power spectra by a factor of 3000, and the overall computational time by a factor of 10, while providing very similar posteriors. [*See the electronic edition of the Journal for a color version of this figure.*]

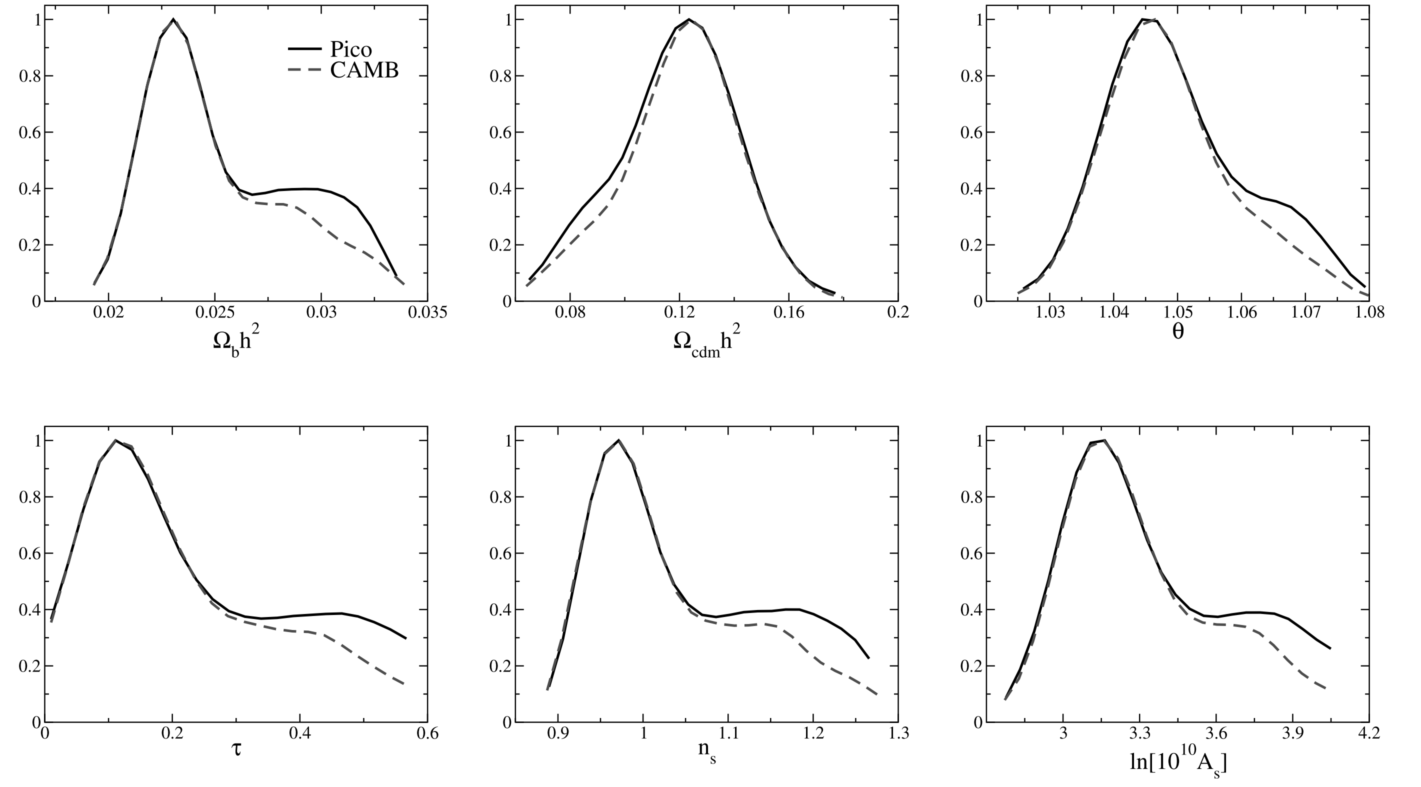

Fig. 5.—Mean likelihoods over the cosmological parameters using runs of CosmoMC with Pico and CAMB. The likelihoods were computed using the first-year *WMAP* data and likelihood code based on the Pico power spectra (*solid line*) and the CAMB power spectra (*dashed line*). Using Pico decreased the overall computational time by a factor of 10, while accurately fitting the peaks of the mean likelihoods. The error in the tails is due to correlated error in the Pico computed power spectra. [*See the electronic edition of the Journal for a color version of this figure.*]

over the posterior depends on the square of the likelihood. Also, the likelihood is very sensitive to any correlated errors in the approximate power spectra computed by Pico. As will be shown in § 4.4, this problem is solved by using Pico to directly compute the likelihood. Even here, however, Pico agrees quite well with CAMB around the peak of the mean likelihood.

In Figure 6, we directly compare the accuracy of the likelihoods computed using power spectra from Pico and CMBwarp with CAMB. Using a uniformly sampled subset of the MCMC chain discussed in the previous paragraph, we first ordered the points by likelihood (*solid line*). Next we recomputed the likelihood using power spectra from Pico (*dark gray circles*) and CMBwarp (*light gray triangles*) at each point. We see that the error in Pico is less than unity over two decades in likelihood. This is a significant improvement over CMBwarp. The dashed lines are plus and minus one of the actual value of the log likelihood.

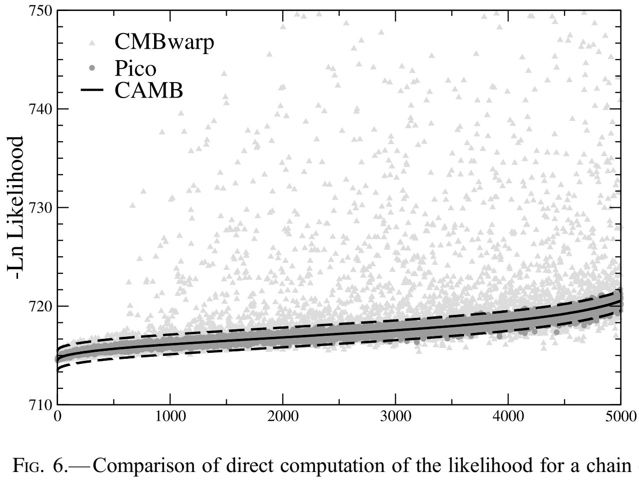

Fig. 6.—Comparison of direct computation of the likelihood for a chain of models. The solid line is the value of the log likelihood using power spectra from CAMB. The dark gray circles and light gray triangles are the values of the log likelihoods computed using power spectra from Pico and CMBwarp, respectively. The dashed lines are ±1 of the log likelihood using CAMB. Pico agrees within 1 log likelihood over 2 decades of likelihood values. [*See the electronic edition of the Journal for a color version of this figure.*]

4.4. *Parameter Posteriors Using Pico to Compute the Likelihood*

As a final test, we ran CosmoMC using Pico to compute the WMAP3 likelihood. Figure 7 compares the posteriors of this run with those using CAMB and the official *WMAP* 3 yr likelihood code. The chains varied seven parameters; these included the six parameters listed in § 4.3, as well as the dark energy equation of state *w*. The dashed line denotes the posterior using Pico to compute the power spectrum *and* the WMAP3 likelihood. The solid line denotes the posterior using CAMB and the official WMAP3 likelihood code. By fitting both the likelihood and the power spectrum Pico provides a factor of 30 increase in speed. Furthermore, with Pico, CosmoMC spends only about 1/5 of its time computing the power spectrum and likelihood, demonstrating that Pico has successfully removed these two bottlenecks from the parameter estimation process. Although it is not needed in this example, Pico also provides the transfer function so that CAMB can compute the matter power spectrum and σ_8.

5. CONCLUSION

This paper provides a fast, accurate, and robust method of calculating CMB power spectra and likelihood functions using local polynomial interpolation. A *k*-means clustering algorithm

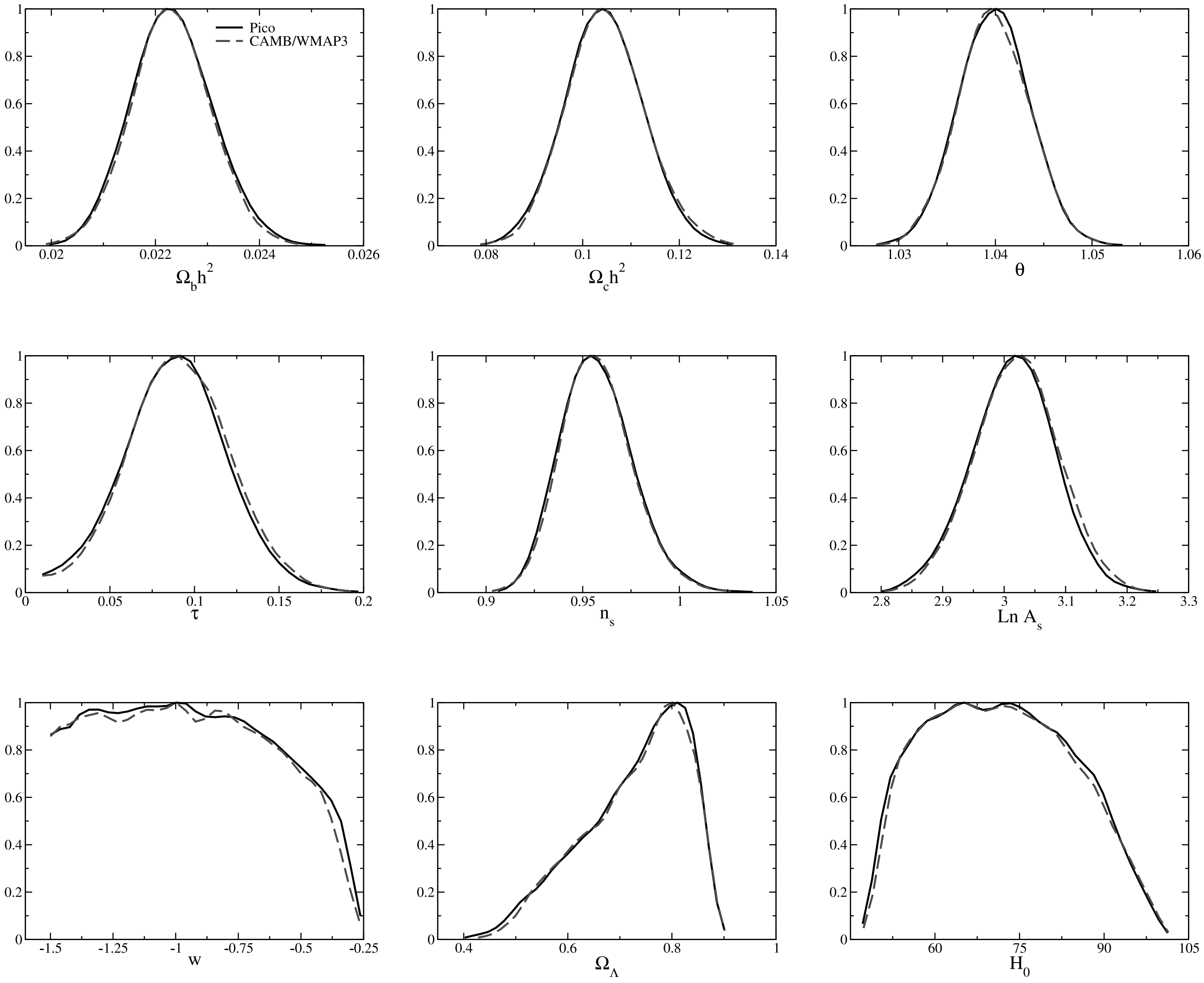

Fig. 7.—Posteriors for seven-parameter flat models using the *WMAP* 3 yr data. The solid line denotes the posterior using Pico to compute both the power spectrum and the likelihood, while the dashed line is the posterior using CAMB and the *WMAP* 3 yr likelihood. The Hubble constant H_0 and the dark energy density Ω_Λ are derived from the other seven parameters. Using Pico to compute the likelihood and power spectra provides a factor of 30 increase in the speed of the parameter estimation code. The improvement over Fig. 4 is the result of fitting the likelihood directly with Pico. [*See the electronic edition of the Journal for a color version of this figure.*]

is used to partition the cosmological parameter space into local regions. Over each region we approximate the CMB power spectra as a polynomial in the cosmological parameters. This method, which we have named Pico, provides several orders of magnitude increase in speed over CAMB and the *WMAP* 3 yr likelihood code, while proving accurate enough for the analysis of data from the current and next generation of cosmic microwave background experiments. The flexibility of our algorithm enables it to handle any reasonable number of cosmological parameters. It has been generalized to allow the fast computation of any observables relevant to a particular data set, e.g., the transfer functions and the power spectrum of *B*-mode polarization anisotropies. Even higher order correlation functions, such as the reduced bispectrum, could be added. Pico is able to compute accurate power spectra over a large volume of parameter space consistent with the *WMAP* data. Furthermore, Pico's performance will only improve as the volume of space it must fit and the uncertainties in the parameters shrink.

Pico is easily inserted into parameter estimation codes such as CosmoMC. It can be used to compute the power spectra, transfer function, and the WMAP3 likelihood, resulting in a significant decrease in computational time. In fact, when Pico is used, CosmoMC spends 80% of its time on tasks other than computing the power spectra and likelihood. This time is spent generating random numbers, evaluating internal and derived parameters, etc. We envision that CAMB will only be needed for nonstandard cosmological models outside the scopes of our training sets. While it is likely possible to further improve the accuracy of our code by using less generic techniques, we have chosen to keep Pico as generic as possible to allow it to grow and adapt to the parameter estimation tasks of the next generation of experiments.

We have made a Fortran 90 implementation of this algorithm publicly available.[5] Here the user will find regression coefficients to use the algorithm for various parameter sets, as well as short and straight forward instructions for incorporating Pico into CosmoMC or using it as a front end for CAMB. The authors also welcome requests for regression coefficients for specific combinations and ranges of parameters. Enabling Pico on a new parameter set simply involves running CAMB to generate a new training set.

[5] See http://www.astro.uiuc.edu/~bwandelt/pico/.

We would like to thank Lloyd Knox, Antony Lewis, and Max Tegmark for useful discussion. This work was partially funded by NSF grant AST 05-07676, by NASA contract JPL1236748, by the National Computational Science Alliance under AST300029N, and by the University of Illinois. We utilized the Teragrid (http://www.teragrid.org) and Xeon cluster tungsten and Itanium 2 cluster mercury at NCSA http://www.ncsa.uiuc.edu.

APPENDIX A

ALGORITHM

This appendix presents the basic algorithm Pico uses to calculate the angular power spectra. It consists of three major pieces, the compression of the training set power spectra, the clustering of the training set cosmological parameters, and the calculation of the local regression polynomials. For clarity, we discuss the latter of these pieces first.

A1. POLYNOMIAL INTERPOLATION

Consider a training set of N vectors of cosmological parameters $\boldsymbol{x}^{(j)}$, each of dimension $\mathcal{N}_x$, and their corresponding power spectrum $\boldsymbol{y}^{(j)}$, each of dimension $\mathcal{N}_y$. The number of cosmological parameters and power spectrum values is arbitrary. In general, $\boldsymbol{y}$ can be constructed by concatenating all the scalar, tensor, and lensed power spectra, as well as the transfer functions into a single vector living in $\mathbb{R}^{\mathcal{N}_y}$.

Our goal is to interpolate the function $\boldsymbol{f}$ that maps the cosmological parameters $\boldsymbol{x}$ into their power spectra $\boldsymbol{y}$, i.e., $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$. This function is an $\mathcal{N}_x$-dimensional manifold that is naturally embedded in an $(\mathcal{N}_x + \mathcal{N}_y)$-dimensional Euclidean space. Our method is to approximate this mapping using a polynomial in the cosmological parameters. The kth component of $\boldsymbol{y}$ is then approximated as a pth order polynomial in the $\mathcal{N}_x$ cosmological parameters as

$$y_k = \sum_{i_1 \geq i_2 \geq \ldots \geq i_p}^{\mathcal{N}_x} \alpha_{i_1 i_2 \ldots i_p} x_{i_1} x_{i_2} \ldots x_{i_p}.$$

The coefficients $\alpha_{i_1 \ldots i_p}$ are chosen to minimize the squared error over the training set

$$R^2 = \sum_{j=1}^{N} \left[\boldsymbol{y}\left(\boldsymbol{x}^{(j)}\right) - \boldsymbol{y}^{(j)}\right]^2,$$

where j indexes the points in the training set.

A straightforward way to solve the regression problem is to define a new vector $\boldsymbol{z}$ whose $\mathcal{N}$ components correspond to the appropriate product of the parameters $\boldsymbol{x}$ for each term in the polynomial. Recall that $\mathcal{N}$ is the number of terms in a pth order polynomial in $\mathcal{N}_x$ variables. Then y_k above can be written as

$$y_k = \boldsymbol{\beta}_k \cdot \boldsymbol{z},$$

where $\boldsymbol{\beta}_k$ is a $\mathcal{N}$-dimensional vector containing the coefficients of the polynomial that evaluates to y_k. These coefficients are found by solving $\partial R^2/\partial \boldsymbol{\beta}_k = 0$ for $\boldsymbol{\beta}_k$. This can be expressed as a matrix equation of the form $A\boldsymbol{\beta}_k = \boldsymbol{b}$ where the components of the matrix A and vector $\boldsymbol{b}$ are given by

$$A_{mn} = \sum_{j=1}^{N} z_m^{(j)} z_n^{(j)}, \qquad b_m = \sum_{j=1}^{N} z_m^{(j)} y_k^{(j)},$$

where the sums are over the points in the training set. Assuming the training set is sufficiently large, A will be a nonsingular matrix that can be inverted to give the regression coefficients $\boldsymbol{\beta}_k$. This procedure is repeated for each index of $\boldsymbol{y}$.

We have generalized this algorithm to include arbitrary fitting functions, for example Chebyshev or Legendre polynomials. Our tests show that Pico performs at a similar level using these functions as using standard polynomials.

A2. CLUSTERING

The interpolation method described above fails to accurately model the power spectra over the entire parameter space. To remedy this, we would like to fit polynomials on disjoint local regions of the full parameter space, limiting the variation in the power spectra over the individual regions. While naively gridding this large-dimensional space would be computationally prohibitive, clustering avoids the "curse of dimensionality" by using the points in the training set to naturally divide the parameter space into smaller regions. A polynomial is used within each cluster to provide a local approximation of the power spectra within the cluster. It is then only necessary to ensure that each cluster has a sufficient number of training set points to accurately calculate the regression coefficients.

One of the simplest methods for hard clustering is the k-means algorithm (MacQueen 1967; Kirby 2001). The k-means clustering algorithm begins by choosing n points to represent the centers of the n clusters. The original n points can simply be chosen as the first n members of the full training set. Each member of the training set is grouped with its nearest center according to a predefined distance

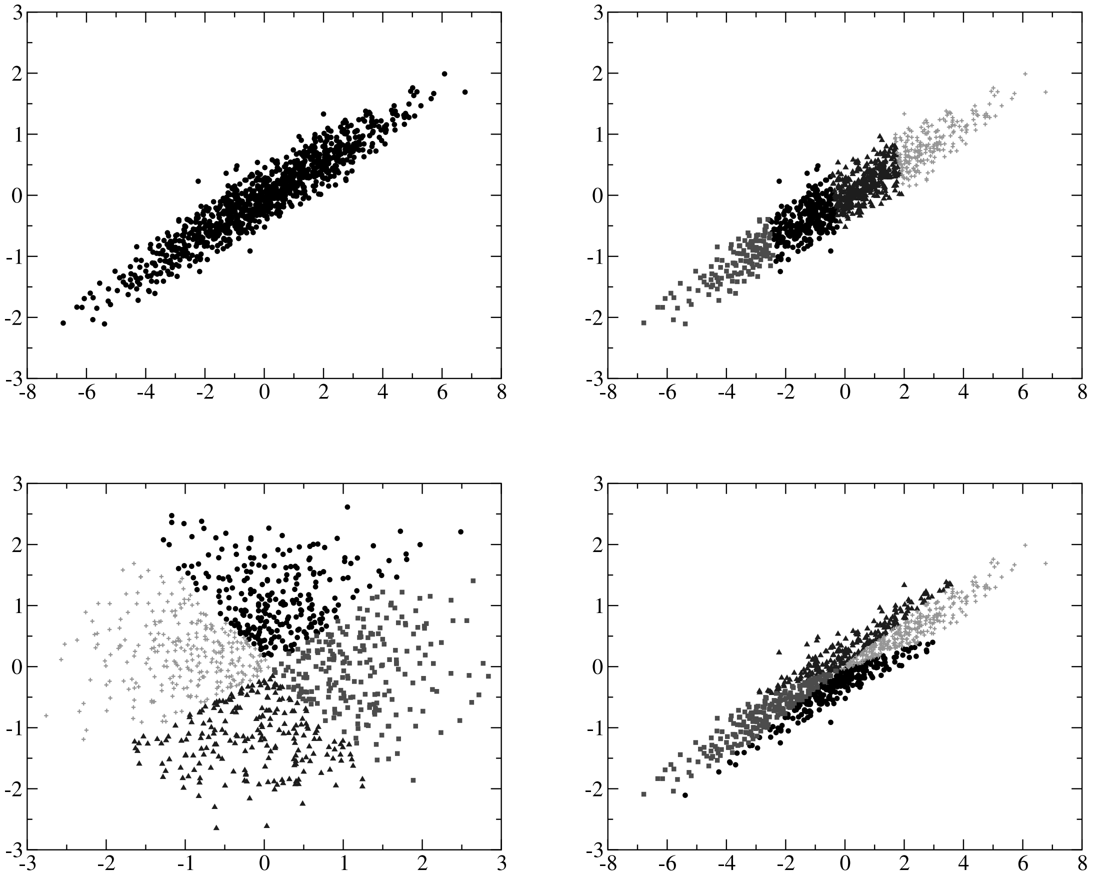

FIG. 8.—Example of sphering and k-means clustering using a two-parameter training set. The top left figure is the original data set. In the top right figure the training set has been clustered into 4 regions. Since the variation of the parameters is significantly larger in the horizontal direction, the clustering simply divides along this axis. However, the power spectrum will vary equally in both directions of parameter space, so there is a much larger variation in the power spectrum along the vertical direction of each cluster. This property can be avoided by sphering the parameter space prior to clustering. The bottom left figure shows the data set after sphering and then clustering. In this basis the parameters and the power spectrum vary equally in all directions. The bottom right figure shows the data set back in the original basis. Now there is no bias in the clusters based on the scale or correlations of the parameters. The clusters retain the property that the power spectrum will vary equally across each cluster. [*See the electronic edition of the Journal for a color version of this figure.*]

function. The centers of each cluster are recalculated using the mean position of the points in each cluster. Using the new cluster centers the training set is reindexed by again finding the nearest center to each point. This process is iterated until the points no longer change clusters. The algorithm returns an array denoting the cluster membership of each point in the training set.

Ideally, all clusters encompass volumes of parameter space over which the power spectra vary roughly equally. For example, we would like to take into account the fact that there is a roughly equal variation of the power spectra from a change in the baryon density of $\sim$0.01 as from a change in the cold dark matter density of $\sim$0.1. This is achieved by sphering the training set prior to clustering. A sphered data set is defined to have a covariance matrix equal to the identity. If C_{xx} denotes the covariance matrix of the cosmological parameters that make up the training set, then the set is sphered by constructing the matrix MU such that

$$MUC_{xx}U^TM^T = ME_{xx}M = I.$$

Here U is the orthogonal matrix that diagonalizes C_{xx}, M is a diagonal matrix whose entries are the inverse of the square root of the eigenvalues of C_{xx}, and the diagonal matrix E_{xx} contains the eigenvalues of C_{xx}. The matrix MU is used to map the training set into a new sphered basis. In this basis the parameter space is clustered using the k-means algorithm and the standard Euclidean distance. Since in the sphered space, the power spectra corresponding to the parameters will vary equally in all directions, the clusters will retain this desired property when mapped back to the unsphered basis.

In Figure 8 we demonstrate the results of using the k-means clustering algorithm on a two-dimensional parameter space, $\mathcal{N}_x = 2$. The different point types distinguish the members of each of the four clusters. Note the difference in the arrangement of the clusters when the data is sphered prior to clustering. The sphered data ignores the scale and correlations of the parameters, giving clusters over which the power spectra vary roughly equally.

A3. POWER SPECTRUM COMPRESSION

The efficiency of the algorithm can be improved by using Karhunen-Loève compression (Karhunen 1947; Loève 1955; Tegmark & Bunn 1995) to transform the power spectra subspace of the training set to a new, lower dimensional space. We begin with a training set of 10^4 power spectra generated as in § 4.1. The training set consists of vectors $\boldsymbol{y}^{(j)}$ formed from concatenating the scalar TT, TE, and EE

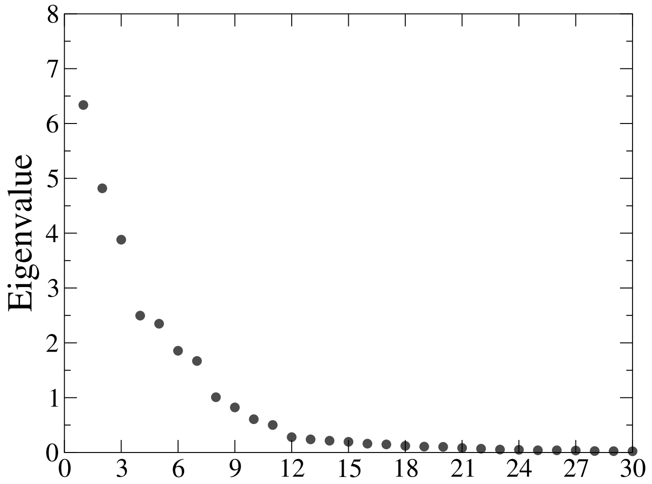

FIG. 9.—Plot of the 30 largest eigenvalues of the covariance matrix of the training set power spectra. There are 135 total eigenvalues. The fact that only a few of the eigenvalues dominate indicates that the power spectra can be compressed significantly by rotating into a new basis and projecting out directions corresponding to small eigenvalues. In the examples below we keep the directions corresponding to the 60 largest eigenvalues. [*See the electronic edition of the Journal for a color version of this figure.*]

power spectra evaluated at the 45 "usual" l-values used by CMBfast out to $l = 1500$. The "usual" l-values are the ones actually evolved by CMBfast or CAMB; the power spectrum is interpolated at the intermediary l's. After constructing the covariance matrix of the power spectra C_{yy}, an eigen decomposition gives a transformation matrix V having the property

$$VC_{yy}V^T = E_{yy},$$

where E_{yy} is a diagonal matrix containing the eigenvalues of C_{yy}. In Figure 9, we have plotted the 30 largest eigenvalues of C_{yy}. The fact that these eigenvalues vary over a large range indicates that some redundancy remains in the components of $\boldsymbol{y}$. By choosing a new basis nearly all of the information in the three power spectra can be stored in significantly fewer coefficients. The compression matrix is formed by dropping the rows of V, which are the eigenvectors of C_{yy}, that have small eigenvalues (relative to the largest). Then V is a mapping from a 135 dimensional space to a much smaller (~60 dimensional) space. Since a set of polynomial regression coefficients is needed for each component of $\boldsymbol{y}$, this compression algorithm provides a significant reduction in the computation time and memory requirements of the algorithm.

In Figure 10 we plot the error accrued due to the compression of the power spectra. That is, we computed V, truncated the specified number of rows, and calculated

$$\boldsymbol{y}' = V^T V \boldsymbol{y}$$

over the 10^4 models, computed using CAMB, that we will use as our test set in § 4. The dimensionless average error in the plots is defined as the mean absolute deviation,

$$\text{Error} = \frac{1}{\mathcal{N}_y} \sum_l \left\langle \frac{\left| C_l - C_l^{\text{CAMB}} \right|}{\sigma_l^{\text{CV}}} \right\rangle,$$

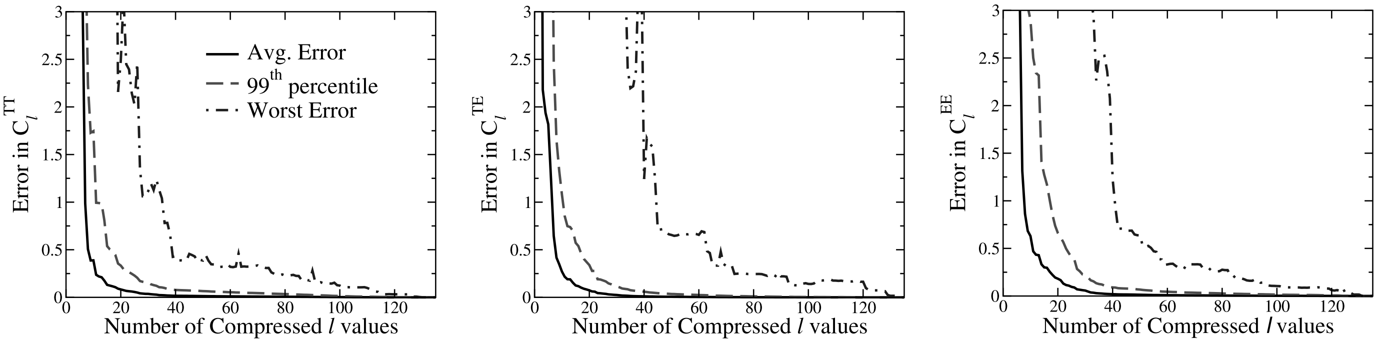

FIG. 10.—Plots of the error accrued due to compressing the power spectra as a function of the number of compressed l's. The three lines denote the average error, the 99th percentile error bar, and the worst error over the 10^4 models in the test set. The worst error is the largest deviation from CAMB at any l in any member of the test set. For nearly all of the models, only ~60 compressed l's are needed of the 135 possible to maintain sufficient accuracy in the compressed power spectra. [*See the electronic edition of the Journal for a color version of this figure.*]

where C_l and $C_l^{\rm CAMB}$denote the individual power spectra that make up $\boldsymbol{y}'$ and $\boldsymbol{y}$ respectively. The brackets denote averaging over the 10^4 models, and $\sigma_l^{\rm CV}$ is the cosmic standard deviation computed using $C_l^{\rm CAMB}$. Recall that the cosmic standard deviation of the TT, TE, and EE spectra are given by

$$\sigma_l^{\rm CV,TT} = \sqrt{\frac{2}{2l+1}C_l^{\rm TT}},$$

$$\sigma_l^{\rm CV,\,TE} = \sqrt{\frac{1}{2l+1}\left[C_l^{\rm TT}C_l^{\rm EE} + \left(C_l^{\rm TE}\right)^2\right]},$$

$$\sigma_l^{\rm CV,\,EE} = \sqrt{\frac{2}{2l+1}C_l^{\rm EE}}.$$

REFERENCES

Bennett, C., et al. 2003, ApJS, 148, 1

Chu, M., Eriksen, H. K., Knox, L., Gorski, K. M., Jewell, J. B., Larson, D. L., O'Dwyer, I. J., & Wandelt, B. D. 2005, Phys. Rev. D, 71, 103002

Grainge, K., et al. 2003, MNRAS, 341, L23

Halverson, N. W., et al. 2002, ApJ, 568, 38

Hanany, S., et al. 2000, ApJ, 545, L5

Hinshaw, G., et al. 2003, ApJS, 148, 135

———. 2006, ApJ, submitted (astro-ph/0603451)

Jimenez, R., Verde, L., Peiris, H., & Kosowsky, A. 2004, Phys. Rev. D, 70, 023005

Kaplinghat, M., Knox, L., & Skordis, C. 2002, ApJ, 578, 665

Karhunen, K. 1947, Ann. Acad. Sci. Fennicae, 37

Kinney, W. H. 1998, Phys. Rev. D, 58, 123506

Kirby, M. 2001, Geometric Data Analysis: An Empirical Approach to Dimensionality Reduction and the Study of Patterns (New York: John Wiley & Sons)

Kogut, A., et al. 2003, ApJS, 148, 161

Kosowsky, A., Milosavljevic, M., & Jimenez, R. 2002, Phys. Rev. D, 66, 063007

Kuo, C., et al. 2004, ApJ, 600, 32

Lewis, A., & Bridle, S. 2002, Phys. Rev. D, 66, 103511

Lewis, A., Challinor, A., & Lasenby, A. 2000, ApJ, 538, 473

Loève, M. 1955, Probability Theory (Princeton: Van Nostrand)

MacQueen, J. 1967, Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability, 1, 281

Miller, A. D., et al. 1999, ApJ, 524, L1

Padin, S., et al. 2001, ApJ, 549, L1

Page, L., et al. 2006, ApJ, submitted (astro-ph/0603450)

Ruhl, J. E., et al. 2003, ApJ, 599, 786

Sandvik, H. B., Tegmark, M., Wang, X., & Zaldarriaga, M. 2004, Phys. Rev. D, 69, 063005

Seljak, U., & Zaldarriaga, M. 1996, ApJ, 469, 437

Spergel, D. N., et al. 2006, ApJ, submitted (astro-ph/0603449)

Tegmark, M., & Bunn, E. F. 1995, ApJ, 455, 1

Verde, L., et al. 2003, ApJS, 148, 195

Wandelt, B. D., Larson, D. L., & Lakshminarayanan, A. 2004, Phys. Rev. D, 70, 083511

Zaldarriaga, M., Spergel, D. N., & Seljak, U. 1997, ApJ, 488, 1