Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

Mon. Not. R. Astron. Soc. **406,** 1177–1189 (2010) doi:10.1111/j.1365-2966.2010.16788.x

Efficient cosmological parameter sampling using sparse grids

M. Frommert,[1⋆†] D. Pflüger,[2⋆†] T. Riller,[1] M. Reinecke,[1] H.-J. Bungartz[2] and T. A. Enßlin[1]

[1] *Max-Planck-Institut für Astrophysik, D-85740 Garching, Germany*
[2] *Institut für Informatik, TU München, 85748 Garching, Germany*

Accepted 2010 March 25. Received 2010 March 3; in original form 2009 November 12

ABSTRACT

We present a novel method to significantly speed up cosmological parameter sampling. The method relies on constructing an interpolation of the cosmic microwave background log-likelihood based on sparse grids, which is used as a shortcut for the likelihood evaluation. We obtain excellent results over a large region in parameter space, comprising about 25 log-likelihoods around the peak, and we reproduce the one-dimensional projections of the likelihood almost perfectly. In speed and accuracy, our technique is competitive to existing approaches to accelerate parameter estimation based on polynomial interpolation or neural networks, while having some advantages over them. In our method, there is no danger of creating unphysical wiggles as it can be the case for polynomial fits of a high degree. Furthermore, we do not require a long training time as for neural networks, but the construction of the interpolation is determined by the time it takes to evaluate the likelihood at the sampling points, which can be parallelized to an arbitrary degree. Our approach is completely general, and it can adaptively exploit the properties of the underlying function. We can thus apply it to any problem where an accurate interpolation of a function is needed.

Key words: methods: data analysis – cosmological parameters.

1 INTRODUCTION

The main two bottlenecks in cosmological parameter estimation using the power spectrum of the cosmic microwave background (CMB) are the calculation of the theoretical C_l spectrum using Boltzmann codes such as CMBFAST (Seljak & Zaldarriaga 1996), CAMB (Lewis, Challinor & Lasenby 2000) or CMBEASY (Doran 2005) and the evaluation of the likelihood using the official *Wilkinson Microwave Anisotropy Probe* (*WMAP*) likelihood code.[1] There exist several methods to speed up the calculation of the power spectrum (Kaplinghat, Knox & Skordis 2002; Jimenez et al. 2004; Habib et al. 2007) or the *WMAP* likelihood function $\mathcal{L}$ (Sandvik et al. 2004; Fendt & Wandelt 2007; Auld, Bridges & Hobson 2008). These methods are based on different techniques, such as analytic approximations, polynomial fits and neural networks, which are all trained using a set of training points, for which the real power spectra and likelihood values have to be calculated. Once the codes are trained for a particular cosmological model, they can be used to evaluate the power spectrum or the likelihood function in every subsequent parameter estimation run, which significantly speeds up the Markov Chain Monte Carlo (MCMC) simulations used for parameter estimation. Due to the ever-growing amount of available data, a fast evaluation of the likelihood is becoming of increasing importance, especially when combining CMB data with data sets whose likelihood is less expensive to evaluate. The *Planck Surveyor* mission (Tauber 2000) will be the upcoming challenge in this respect.

In this work, we approximate the *WMAP* log-likelihood function $\ln \mathcal{L}$ in the spirit of CMBfit (Sandvik et al. 2004) and Pico (Fendt & Wandelt 2007), which work with polynomial fits, and CosmoNet (Auld et al. 2008), an approach based on neural networks. In contrast to the fitting functions constructed therein, we introduce the technique of *sparse grids* in this context to construct an interpolation of $\ln \mathcal{L}$, returning the exact function values at a set of sampling points.

Most straightforward interpolation techniques are based on sets of sampling points in each dimension, typically based on (uniform) grid structures – consider, for example, piecewise d-linear or piecewise d-polynomial interpolation schemes. Unfortunately, grid-based methods are only feasible in low-dimensional settings, as they suffer from the so-called *curse of dimensionality*: spending $\tilde{N}$ function evaluations or grid points in one dimension leads to $\tilde{N}^d$ grid points in d dimensions. The exponential dependency on the dimensionality imposes severe restrictions on the number of dimensions that can be handled. Sparse grids, as introduced by Zenger (1991), allow to overcome the curse of dimensionality to some extent, at least for sufficiently smooth functions as it is the case in our setting.

⋆E-mail: mona@mpa-garching.mpg.de (MF); pflueged@in.tum.de (DP)
†The authors contributed equally to the article.
[1] http://lambda.gsfc.nasa.gov/product/map/dr3/likelihood_get.cfm

Sparse grid interpolation is based on an a priori selection of grid points, requiring significantly fewer grid points than conventional interpolation on a full grid, while preserving the asymptotic error decay of a full grid interpolation with increasing grid resolution up to a logarithmic factor. This permits us to compute higher dimensional interpolations and approximations than before. A very good overview about sparse grids, discussing general properties, can be found in Bungartz & Griebel (2004).

The sparse grid technique is a completely general approach, not tailored to a single application, and can therefore be used to interpolate any function which is sufficiently smooth. Additionally, as it allows for arbitrary adaptive refinement schemes, the general, fast convergence rates can be improved even further, by adapting to the special characteristics of the underlying target function.

We obtain excellent results, which are competitive to fitting procedures using polynomials (Sandvik et al. 2004; Fendt & Wandelt 2007) or neural networks (Auld et al. 2008) in speed and accuracy. Furthermore, we believe that the interpolation based on sparse grids has several advantages over these approaches. First of all, we can use the results of sparse grid approximation quality (Bungartz & Griebel 2004), guaranteeing the convergence of the interpolation towards the original function with increasing number of grid points.

Secondly, once we have chosen the volume in which we want to interpolate the function in question, the sparse grid structure itself determines a priori the location of potential sampling points (which can additionally be refined in an adaptive manner a posteriori). This makes it unnecessary to assemble a set of training points beforehand (e.g. by running MCMCs as it is done by Fendt & Wandelt 2007). The generation of the sampling points and the construction of the interpolant can be strongly parallelized, which makes the sparse grid approach an ideal candidate for computational grid projects such as the AstroGrid.[2] The time needed to construct the interpolant is determined almost only by the time it takes to evaluate the likelihood at the sampling points. We do not need additional training time as in the case of Auld et al. (2008).

Furthermore, polynomial fits to a set of training points run the risk of creating unphysical wiggles if the polynomial degree of the fitting function is chosen too high with respect to the amount of training points available. Using the sparse grids approach, piecewise polynomials of low degree are sufficient, as we are not fitting certain evaluation points but rather interpolating a function. Increasing the accuracy is therefore equivalent to evaluating at more sampling points.

Sparse grids are based on a hierarchical formulation of the underlying basis functions, which can be used to obtain a generic estimate of the current approximation error while evaluating more and more sampling points. This can be directly used as a criterion for adaptive refinement as well as to stop further refinement.

Another advantage is that the projection of sparse grid interpolations can be done in a very fast and simple way. This would make sparse grids in principle a good candidate for sampling posteriors and projecting them directly, without having to use a Markov Chain approach in order to marginalize the posterior. Given that MCMCs need to determine the points sequentially and can therefore not be parallelized (apart from running several chains at the same time), it would be highly desirable to find alternatives that can be run in parallel.

We have attempted to use sparse grids in order to substitute the MCMCs in cosmological parameter estimations. In order to directly project the posterior distribution, we would need to sample the posterior rather than its logarithm. Since, in general, the logarithm of a probability density function is considerably more well behaved than the function itself, Sandvik et al. (2004), Fendt & Wandelt (2007) and Auld et al. (2008) all operate in log-space to speed up the generation of MCMCs instead. As the convergent phase of the interpolation with sparse grids sets in rather late when interpolating Gaussian functions (and thus the *WMAP* likelihood, which is close to a d-dimensional Gaussian), we restricted ourselves to the log-likelihood and thus to accelerating MCMCs too.

The article is organized as follows. First, we describe the basics of sparse grids in Section 2, introducing a modification of the standard sparse grid approach, thus adapting the latter to our problem. In Section 3, we then present the interpolation of the *WMAP* likelihood for two different sets of parameters in both six and seven dimensions. We show that the results obtained for regular (non-adaptive) sparse grids are already competitive to other approaches and demonstrate how adaptive refinement can further improve the results. Section 4 finally concludes our work.

2 BASICS OF SPARSE GRIDS

Standard grid-based approaches of interpolating a function f exhibit the curse of dimensionality, a term going back to Bellman (1961): any straightforward discretization scheme which employs $\tilde{N}$ grid points (or, equivalently, degrees of freedom) in one dimension leads to $\tilde{N}^d$ grid points in d dimensions. For reasonable $\tilde{N}$, the exponential dependency on the number of dimensions typically does not allow to handle more than four-dimensional problems.

Sparse grids are able to overcome this hurdle to some extent, requiring significantly fewer grid points than a full grid, while preserving the asymptotic error decay of full grid interpolation with increasing grid resolution up to a logarithmic factor. Sparse grids have originally been developed for the solution of partial differential equations (Zenger 1991) and have meanwhile been applied to various problems, see Bungartz & Griebel (2004) and the references cited therein. Recent work on sparse grids includes stochastic and non-stochastic partial differential equations in various settings (von Petersdorff & Schwab 2006; Ganapathysubramanian & Zabaras 2007; Widmer, Hiptmair & Schwab 2008), as well as applications in economics (Reisinger & Wittum 2007; Holtz 2008), regression (Garcke 2006; Garcke & Hegland 2009), classification (Garcke, Griebel & Thess 2001; Bungartz, Pflüger & Zimmer 2008), fuzzy modelling (Klimke, Nunes & Wohlmuth 2006) and more. Note that (non-adaptive) sparse grids are closely related to the technique of hyperbolic crosses (Temlyakov 1993).

In this section, we provide a brief overview of sparse grids for interpolation. For a detailed derivation of the characteristics of sparse grids, we refer to Bungartz & Griebel (2004). We start by formulating the interpolation on a conventional full grid using hierarchical basis functions, from which we then derive the interpolation on a sparse grid by omitting the basis functions contributing least to the interpolation.

2.1 General idea of interpolation on a full grid

We consider the piecewise d-linear interpolation of a function $f : \Omega \to \mathbb{R}$ which is given only algorithmically, i.e. we have no closed form of f but we can only evaluate f at arbitrary points using a numerical code. As we want to discretize our domain of interest Ω, we restrict Ω to a compact subvolume of $\mathbb{R}^d$; here, $\Omega \equiv [0, 1]^d$, the d-dimensional unit hypercube. (For the standard approach of

[2] http://www.d-grid.de/index.php?id=45&L=1

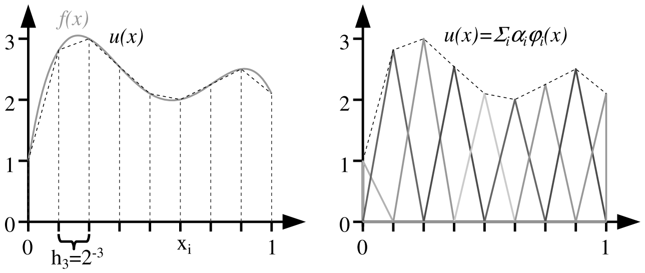

Figure 1. One-dimensional piecewise linear interpolation $u(x)$, dashed, of a function $f(x)$, solid, (left-hand panel) by a linear combination of hat basis functions (right-hand panel).

sparse grids techniques, we only consider functions that are zero on the boundary of the volume on which they are defined. This assumption will be dropped when we come to the interpolation of the log-likelihood of *WMAP*.)

To construct an interpolant u of f, we discretize Ω via a regular grid, obtaining equidistant grid points $\boldsymbol{x}_i$, with mesh width $h_n = 2^{-n}$ for some discretization or refinement level n, at which we evaluate and interpolate f. If we define a suitable set of piecewise d-linear basis functions $\varphi_i(\boldsymbol{x})$, we can obtain $u(\boldsymbol{x})$ from the space of continuous, piecewise d-linear functions V_n by combining them adequately as a weighted sum of basis functions, i.e.

$$f(\boldsymbol{x}) \approx u(\boldsymbol{x}) \equiv \sum_i \alpha_i \varphi_i(\boldsymbol{x})$$

with coefficients α_i. Fig. 1 sketches the idea for a one-dimensional example, using the standard nodal basis.

The curse of dimensionality, encountered when using a full grid, can be circumvented by a suitable choice of basis functions: we need a basis where the relevant information is represented by as few basis functions as possible. Most basis functions can then be omitted as they contribute only little to the interpolation of f, reducing a full grid to a sparse grid and allowing us to handle higher dimensional functions than before. A suitable basis can be found by a hierarchical construction as introduced in the following section.

2.2 Hierarchical basis functions in one dimension

Sparse grids depend on a hierarchical decomposition of the underlying approximation spaces. Therefore, and first considering only the one-dimensional case which we will later extend to d dimensions, we use the standard hat function,

$$\varphi(x) = \max(1 - |x|, 0)\,, \tag{1}$$

from which we derive one-dimensional hat basis functions by dilatation and translation,

$$\varphi_{l,i}(x) \equiv \varphi(2^l x - i)\,, \tag{2}$$

which depend on a level l and an index i, $0 < i < 2^l$. The basis functions have local support and are centred at grid points $x_{l,i} = 2^{-l} i$, at which we will interpolate f. Introducing the hierarchical index sets

$$I_l \equiv \left\{ i \in \mathbb{N} : 1 \le i \le 2^l - 1, i \text{ odd} \right\}\,, \tag{3}$$

we obtain a set of hierarchical subspaces W_l,

$$W_l \equiv \operatorname{span}\left\{ \varphi_{l,i}(x) : i \in I_l \right\}. \tag{4}$$

We can then formulate the space of piecewise linear functions V_n on a full grid with mesh width h_n for a given level n as a direct sum of W_l,

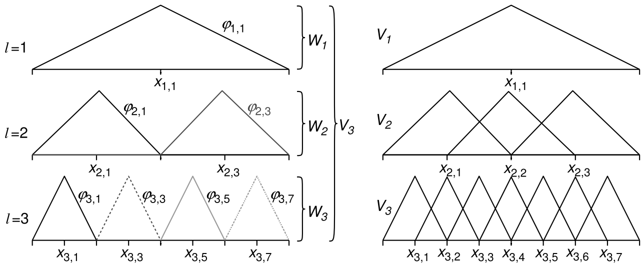

Figure 2. One-dimensional basis functions $\varphi_{l,i}$ and corresponding grid points $x_{l,i}$ up to level $n = 3$ in the hierarchical basis (left-hand panel) and the common nodal point basis (right-hand panel).

$$V_n = \bigoplus_{l \le n} W_l\,, \tag{5}$$

see Fig. 2. Note that all basis functions of the same subspace W_l have the same size, shape and compact support, that their supports are non-overlapping, and that together they cover the whole domain.

The interpolation $u(x) \in V_n$ can then be written as a finite sum,

$$u(x) = \sum_{l \le n, i \in I_l} \alpha_{l,i} \varphi_{l,i}(x)\,, \tag{6}$$

where the so-called (hierarchical) surpluses $\alpha_{l,i}$ are uniquely indexed by the same level and index as the corresponding basis functions.

2.3 Higher dimensional interpolation on a full grid

The basis functions are extended to the d-dimensional case via a tensor product approach,

$$\varphi_{\boldsymbol{l},\boldsymbol{i}}(\boldsymbol{x}) \equiv \prod_{j=1}^{d} \varphi_{l_j,i_j}(x_j)\,, \tag{7}$$

with the d-dimensional multi-indices $\boldsymbol{l}$ and $\boldsymbol{i}$ indicating level and index for each dimension. The other one-dimensional notations can be transferred to the arbitrary-dimensional case as well, consider, for example, the index set $\boldsymbol{I_l}$,

$$\boldsymbol{I_l} \equiv \left\{ \boldsymbol{i} : 1 \le i_j \le 2^{l_j} - 1, i_j \text{ odd}, 1 \le j \le d \right\}\,, \tag{8}$$

the subspaces $W_{\boldsymbol{l}}$, the space V_n of piecewise d-linear functions with mesh width h_n in each dimension,

$$V_n = \bigoplus_{|\boldsymbol{l}|_\infty \le n} W_{\boldsymbol{l}}\,, \tag{9}$$

leading to a full grid with $(2^n - 1)^d$ grid points and to the interpolant $u(\boldsymbol{x}) \in V_n$,

$$u(\boldsymbol{x}) = \sum_{|\boldsymbol{l}|_\infty \le n, \boldsymbol{i} \in \boldsymbol{I_l}} \alpha_{\boldsymbol{l},\boldsymbol{i}} \varphi_{\boldsymbol{l},\boldsymbol{i}}(\boldsymbol{x})\,. \tag{10}$$

Here and later on, we need the l_1-norm $|\boldsymbol{l}|_1 = \sum_{j=1}^{d} l_j$ and the maximum-norm $|\boldsymbol{l}|_\infty = \max_{1 \le j \le d} l_j$ of multi-indices $\boldsymbol{l}$. Fig. 3 shows some two-dimensional examples for the basis functions of the subspaces $\boldsymbol{W_l}$, which correspond to anisotropic subgrids with mesh-width h_{l_j} in dimension j characterized by the multi-index $\boldsymbol{l}$.

2.4 Sparse grids

Starting from the hierarchical representation of V_n by the subspaces $W_{\boldsymbol{l}}$, we can now select those subspaces that contribute most to the

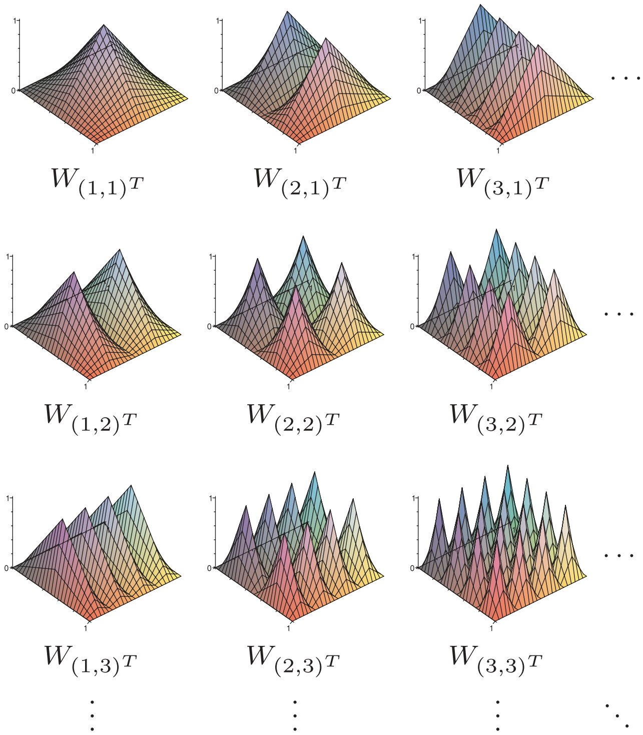

Figure 3. Basis functions of the subspaces $W_{\mathbf{l}}$ for $|\mathbf{l}|_\infty \le 3$ in two dimensions.

overall solution of the full grid interpolation (10). If the function we want to approximate meets certain smoothness conditions – the mixed second derivatives have to be bounded – this can be done a priori as we can derive bounds for the contributions of the different subspaces. We then obtain the sparse grid space

$$V_n^{(1)} \equiv \bigoplus_{|\mathbf{l}|_1 \le n+d-1} W_{\mathbf{l}}\,, \tag{11}$$

leaving out those subspaces from the full grid space V_n with many basis functions of small support. (The exact choice of subspaces depends on the norm in which we measure the error; the result above is optimal for both the L_2 norm and the maximum norm.) Note that in the one-dimensional case the sparse grid space equals the full grid space.

Fig. 4 shows the selection of subspaces and the resulting sparse grid for $n = 3$, i.e. the sparse grid space $V_3^{(1)}$. Compared to the full grid for the same discretization level n (the full grid space V_3 would also comprise the grey subspaces in Fig. 4), this reduces the number of grid points (and therefore function evaluations and unknowns) significantly from $\mathcal{O}(h_n^{-d}) = \mathcal{O}(2^{nd})$ to $\mathcal{O}[h_n^{-1}(\log h_n^{-1})^{d-1}]$ – whereas the asymptotic accuracy deteriorates only slightly from $\mathcal{O}(h_n^2)$ to $\mathcal{O}[h_n^2(\log h_n^{-1})^{d-1}]$, see Bungartz & Griebel (2004) for detailed derivations. Fig. 5 shows sparse grids in two and three dimensions for level $n = 6$ each.

Functions which do not meet the smoothness requirements or which show significantly differing characteristics (e.g. comprising steep regions as well as flat ones) can be tackled as well, if adaptive refinement is used. The sparse grid structure (11) defines an a priori selection of grid points which is optimal if certain smoothness conditions are met and no further knowledge about the function in question is known or used. An adaptive (a posteriori) refinement can additionally select which grid points in a sparse grid structure should be refined next, due to local error estimation, for example. To refine a grid point, often all $2d$ children in the hierarchical structure are added to the current grid, if they have not been created yet. Note that it usually has to be ensured that all missing parents have to be created, as algorithms working on sparse grids depend on traversals of the hierarchical tree of basis functions. If additional knowledge about the problem at hand is available, it can be used in the criterion for adaptive refinement, allowing to better adapt to problem specific characteristics.

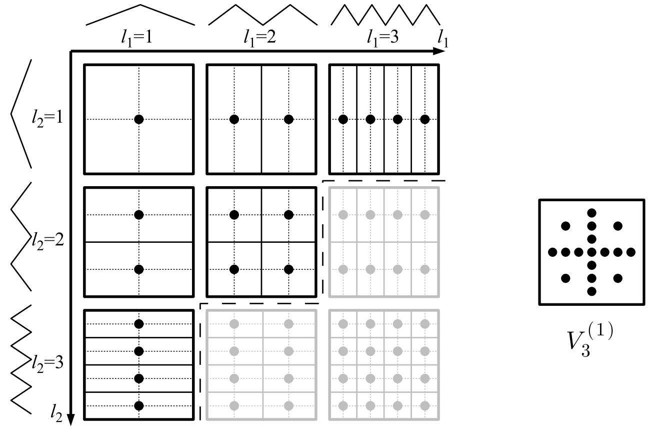

Figure 4. The two-dimensional subspaces $W_{\mathbf{l}}$ up to $l = 3\,(h_3 = 1/8)$ in each dimension. The optimal selection of subspaces (black) and the corresponding sparse grid on level $n = 3$ for the sparse grid space $V_3^{(1)}$. The corresponding full grid of level 3 corresponds to the direct sum of all subspaces that are shown.

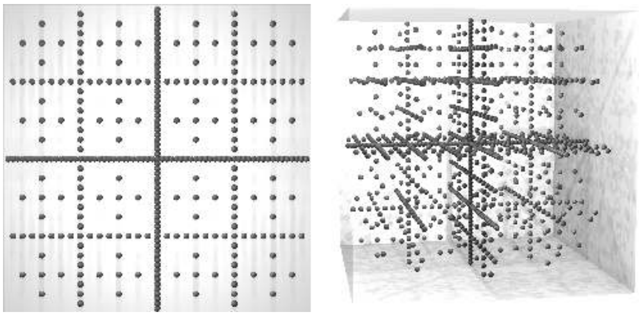

Figure 5. Sparse grids in two and three dimensions for level $n = 6$.

2.5 Extension to functions that are non-zero on the boundary

Up until now, we have only considered functions that are zero on the domain's boundary $\delta\Omega$. To allow for non-zero values on the boundary, usually additional grid points located directly on $\delta\Omega$ are introduced. For example, the one-dimensional basis on level 1, containing only $\varphi_{1,1}(x)$, is extended by two basis functions with level 0 and indices 0 and 1 restricted to Ω, namely $\varphi_{0,0}(x)$ and $\varphi_{0,1}(x)$. They are then extended to the d-dimensional case as before, with the exception that the new basis now contains basis functions on the modified level one with overlapping support.

Apparently, this approach results in many more grid points (and therefore expensive function evaluations) than before. This shows quite nicely that it is not sufficient to just consider the asymptotic behaviour: asymptotically, nothing changes, but for very high dimensionalities we are not able to even start to interpolate any more. In d dimensions, the basis for the subspace $\boldsymbol{W}_{\boldsymbol{1}}$ for example contains already 3^d basis functions, rather than a single one. Especially in

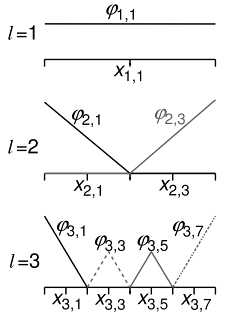

Figure 6. Modified one-dimensional basis functions $\varphi_{l,i}$: constant on level 1 and 'folded up' if adjacent to the boundary on all other levels.

settings where a very high accuracy close to the boundary is not required – which holds in our case – (or where an adaptive selection of grid points is used in any case), it can be advantageous to omit the grid points on the boundary, and instead modify the basis functions to extrapolate towards the boundary of the domain.

We modify the one-dimensional basis functions as follows: on level 1, we have only one degree of freedom, the best guess towards the boundary is to assume the same value, leading to a constant basis function. On all other levels, we extrapolate linearly towards the boundary, 'folding up' the uttermost basis functions. All other basis functions remain unchanged, yielding

$$\varphi_{l,i}(x) \equiv \begin{cases} 1 & \text{if } l = 1 \wedge i = 1\,, \\ \varphi_{l,i}^{\text{left}}(x) & \text{if } l > 1 \wedge i = 1\,, \\ \varphi_{l,i}^{\text{right}}(x) & \text{if } l > 1 \wedge i = 2^l - 1\,, \\ \varphi\left(x\, 2^l - i\right) & \text{else,} \end{cases} \tag{12}$$

with

$$\varphi_{l,i}^{\text{left}}(x) \equiv \left\{ \begin{matrix} 2 - 2^l x & \text{if } x \in \left[0, \frac{1}{2^{l-1}}\right] \\ 0 & \text{else} \end{matrix} \right\}, \tag{13}$$

$$\varphi_{l,i}^{\text{right}}(x) \equiv \left\{ \begin{matrix} 2^l x + 1 - i & \text{if } x \in \left[1 - \frac{1}{2^{l-1}}, 1\right] \\ 0 & \text{else} \end{matrix} \right\}. \tag{14}$$

Examples of the modified one-dimensional basis functions are shown in Fig. 6. The d-dimensional basis functions are obtained as before via a tensor product of the one-dimensional ones.

3 INTERPOLATION OF THE *WMAP* LIKELIHOOD SURFACE

We now construct an interpolation of the *WMAP* log-likelihood, $\ln \mathcal{L}$, using sparse grids. In order to adapt the problem to our interpolation approach, we first use a six-dimensional set of so-called normal parameters introduced in Sandvik et al. (2004), which are a transformation of the usual cosmological parameters such that the major axes of the Gaussian align with the coordinate axes. The logarithm of the likelihood is then well approximated by a sum of one-dimensional parabolas in the different parameters, a fact that we will take advantage of by using the modified basis functions described in (12). For this set of normal parameters, we obtain an accurate interpolation already for a comparably low refinement level. This is shown for the six-dimensional model as well as for a seven-dimensional extension, using the running of the spectral index as an additional parameter.

However, as a subsequent step we demonstrate that the parameter transformation is not essential for obtaining a good interpolation. By investing more grid points, we obtain an accurate interpolation as well when using directly the six- and seven-dimensional standard parameter set, which is usually used in cosmological parameter sampling. This approach shows the advantage of sparse grids of being rather generic. Furthermore, we are not restricted to the parameter range in which the transformation to normal parameters can be inverted.

3.1 Choice of basis functions

We use the modified basis functions as introduced in (12), which are well suited for our problem. First, and as already indicated in Section 2, the region close to the domain's boundary is less important in our setting than the centre of Ω: we will centre the domain of interest roughly at the maximum of the log-likelihood function $\ln \mathcal{L}$ and determine the boundary such that it includes the region with $(\ln \mathcal{L}_{\max} - \ln \mathcal{L}) \lesssim 25$, which we will refer to as the 25 log-likelihood region (see Section 3.3).

Towards the boundaries of our intervals, the likelihood is then effectively zero and thus no great accuracy is needed in these regions. Therefore, we do not want to spend too much work on $\delta\Omega$. Using the modified boundary functions, we extrapolate (d-linearly) towards $\delta\Omega$, see the discussion of the modified basis functions above.

Secondly, the modifications are especially well suited if the function to interpolate can be separated into a sum of one-dimensional functions. Assume that the likelihood $\mathcal{L}$ was a perfect product of one-dimensional Gaussians,

$$\mathcal{L}(\boldsymbol{x}) = c\mathrm{e}^{-a_1(x_1-\mu_1)^2 - \cdots - a_d(x_d-\mu_d)^2}, \tag{15}$$

centred at $(\mu_1, \ldots, \mu_d)^T$. Then, the interpolation of the log-likelihood $\ln \mathcal{L}$ reduces to d one-dimensional problems,

$$\ln \mathcal{L}(\boldsymbol{x}) = \ln c + \sum_{k=1}^{d} f_k(x_k)\,, \tag{16}$$

with

$$f_k(x_k) = -a_k(x_k - \mu_k)^2\,, \tag{17}$$

separating into a constant term plus a sum of functions that are constant in all directions but one.

Keeping in mind that the one-dimensional basis function on level 1, $\varphi_{1,1}(x)$, is constant (cf. Fig. 6), this simplifies the interpolation task. The d-dimensional basis function on level **1**, $\boldsymbol{\varphi}_{\mathbf{1,1}}(\boldsymbol{x})$, serves as an offset. [Only if $(\mu_1, \ldots, \mu_d)^T$ is the centre of Ω, $\boldsymbol{\alpha}_{\mathbf{1,1}}\boldsymbol{\varphi}_{\mathbf{1,1}}(\boldsymbol{x})$ exactly expresses $\ln c$.] Additionally, it is sufficient to spend only grid points on the main axes of the sparse grid (level 1 in all dimensions but one) to approximate the remaining one-dimensional contributions $f_k(x_k)$ arbitrarily well:

$$u(\boldsymbol{x}) = \underbrace{\boldsymbol{\alpha}_{\mathbf{1,1}}\boldsymbol{\varphi}_{\mathbf{1,1}}(\boldsymbol{x})}_{\ln c} + \sum_{k=1}^{d} \underbrace{\left[\sum_{l_k,i_k} \boldsymbol{\alpha}_{\boldsymbol{l},\boldsymbol{i}}\; \varphi_{l_k,i_k}(x_k) \prod_{1\le j\le d,\, j\ne k} \varphi_{1,1}(x_j)\right]}_{f_k(x_k)}. \tag{18}$$

Of course, $\mathcal{L}$ is not a perfect product of one-dimensional Gaussians; grid points that do not lie on the sparse grid's main axes account for the additional mixed (correlated) terms of $\ln \mathcal{L}$. Given that in sparse grids a large amount of points lie on the main axes, this mechanism works very well – the better the less correlation between the different parameters exists.

In order to take as much advantage as possible of the effects described above, we introduce a parameter transformation in the following section, for which the new parameters are less correlated. However, the fact that the interpolation using the standard parameters – which have much stronger correlations – works as well, spending just more grid points, will show that the sparse grid approach does not depend on this argumentation: sparse grids can make use of such properties but do not rely on them.

3.2 Normal parameters

The set of cosmological parameters describing the Λ cold dark matter model consists of the Hubble constant, $h \equiv \frac{H_0}{100\,\mathrm{km\,(s\ Mpc)^{-1}}}$, the density parameter of vacuum energy, Ω_Λ, the ones of baryons, Ω_b, and of matter (baryonic + dark), Ω_m, the optical depth to the last scattering surface, τ, the scalar spectral index of the primordial power spectrum, n_s, and the scalar initial amplitude, A_s. We will refer to these parameters as cosmological parameters. For a more detailed description of the cosmological parameters, we refer to Coles & Lucchin (2002). In the literature, there have been several attempts to transforming these parameters into a set of parameters that mirror the various physical effects on the CMB power spectrum (Hu et al. 2001; Kosowsky, Milosavljevic & Jimenez 2002). In Chu, Kaplinghat & Knox (2003), a set of parameters is provided in which the likelihood surface of the CMB is well approximated by a multivariate Gaussian with the major axes aligned with the coordinate axes. In this work, we use the parameters given by Sandvik et al. (2004), where the parameter set of Chu et al. (2003) is combined with the other parameter sets mentioned, in order to bring the major axes of the likelihood surface even closer to the coordinate axes. The new parameters are then $\{\Theta_\mathrm{s}, h_2, h_3, t, A_*, Z\}$, which we refer to as normal parameters. When working with the latter, the logarithm of the likelihood is well approximated by a sum of one-dimensional parabolas in the different parameters. The basis functions introduced above are therefore ideally adapted to this problem. In the following, we repeat the definitions of the normal parameters for convenience.

The first parameter of our set is the angle subtended by the acoustic scale

$$\Theta_\mathrm{s} \equiv \frac{r_\mathrm{s}(a_\mathrm{ls})}{D_\mathrm{A}(a_\mathrm{ls})}\frac{180}{\pi}\,, \tag{19}$$

where the index 'ls' denotes the time of last scattering, $D_\mathrm{A}(a_\mathrm{ls})$ stands for the comoving angular diameter distance to the surface of last scattering (which we will come back to later) and $r_\mathrm{s}(a_\mathrm{ls})$ is the comoving sound horizon at last scattering,

$$r_\mathrm{s}(a_\mathrm{ls}) \equiv \int_0^{t_\mathrm{ls}} \frac{c_\mathrm{s}(t)}{a(t)}\mathrm{d}t\,. \tag{20}$$

Here, $c_\mathrm{s}(t)$ denotes the sound speed for the baryon–photon fluid at time t, which is well approximated by

$$c_\mathrm{s}(t)^2 \approx \frac{1}{3}\left(1 + 3\frac{\rho_\mathrm{b}}{\rho_\gamma}\right)^{-1}\,, \tag{21}$$

with the index 'b' standing for baryons and the index γ for photons. Using the Friedmann equations and ignoring the vacuum energy at last scattering, $r_\mathrm{s}(a_\mathrm{ls})$ can be shown to be (Kosowsky et al. 2002; Sandvik et al. 2004)

$$r_\mathrm{s}(a_\mathrm{ls}) = \frac{2\sqrt{3}}{3H_0\sqrt{\Omega_\mathrm{m}}}\sqrt{\frac{a_\mathrm{ls}}{R_\mathrm{ls}}}\ln\frac{\sqrt{1+R_\mathrm{ls}}+\sqrt{R_\mathrm{ls}+r_\mathrm{ls}R_\mathrm{ls}}}{1+\sqrt{r_\mathrm{ls}R_\mathrm{ls}}}\,, \tag{22}$$

where

$$R_\mathrm{ls} \equiv \frac{3\rho_\mathrm{b}(a_\mathrm{ls})}{4\rho_\gamma(a_\mathrm{ls})} = 30\,w_\mathrm{b}\left(\frac{z_\mathrm{ls}}{10^3}\right)^{-1}\,, \tag{23}$$

$$r_\mathrm{ls} \equiv \frac{\rho_\mathrm{r}(a_\mathrm{ls})}{\rho_\mathrm{m}(A_*)} = 0.042\,w_\mathrm{m}^{-1}\left(\frac{z_\mathrm{ls}}{10^3}\right)\,. \tag{24}$$

The index 'r' stands for radiation, i.e. ρ_r consists of the sum of photon and neutrino energy densities, and the index 'm' is used for matter (baryons + dark matter). We define $w_\mathrm{m} \equiv \Omega_\mathrm{m}h^2$ in the same way as $w_\mathrm{b} \equiv \Omega_\mathrm{b}h^2$. The redshift at last scattering, z_ls, is well approximated by (Hu et al. 2001)

$$z_\mathrm{ls} = 1048\,(1 + 0.001\,24\,w_\mathrm{b}^{-0.738})(1 + g_1 w_\mathrm{m}^{g_2})\,, \tag{25}$$

$$g_1 \equiv 0.0783\,w_\mathrm{b}^{-0.238}(1 + 39.5\,w_\mathrm{b}^{0.763})^{-1}\,, \tag{26}$$

$$g_2 \equiv 0.560\,(1 + 21.1w_\mathrm{b}^{1.81})^{-1}\,. \tag{27}$$

As already mentioned, $D_\mathrm{A}(a_\mathrm{ls})$ in (19) denotes the comoving angular diameter distance to the surface of last scattering and is given by

$$D_\mathrm{A}(a_\mathrm{ls}) = \frac{c}{H_0}\int_{a_\mathrm{ls}}^1 \frac{1}{\sqrt{\Omega_\Lambda\tilde{a}^4 + \Omega_\mathrm{m}\,\tilde{a} + \Omega_\mathrm{r}}}\mathrm{d}\tilde{a}\,. \tag{28}$$

The second and third parameters in our set are the ratios of the second and the third peak to the first peak in the C_l^T spectrum of the CMB (Hu et al. 2001), where the tilt dependence is factored out (Page et al. 2003),

$$h_2 \equiv 0.0264\,w_\mathrm{b}^{-0.762} \exp\left\{-0.476\,[\ln(25.5\,w_\mathrm{b} + 1.84\,w_\mathrm{m})]^2\right\}\,, \tag{29}$$

$$h_3 \equiv 2.17\left[1 + \left(\frac{w_\mathrm{b}}{0.044}\right)^2\right]^{-1} w_\mathrm{m}^{0.59}\left[1 + 1.63\left(1 - \frac{w_\mathrm{b}}{0.071}\right)w_\mathrm{m}\right]^{-1}\,. \tag{30}$$

We use the tilt parameter given by Sandvik et al. (2004), which is a slightly modified version of the one in Chu et al. (2003) in order to minimize the correlation with w_b:

$$t \equiv \left(\frac{w_\mathrm{b}}{0.024}\right)^{-0.5233} 2^{n_\mathrm{s}-1}\,. \tag{31}$$

The amplitude parameter is

$$A_* \equiv \frac{\tilde{A}_\mathrm{s}}{2.95 \times 10^{-9}}\mathrm{e}^{-2\tau}\left(\frac{k}{k_\mathrm{p}}\right)^{n_\mathrm{s}-1} w_\mathrm{m}^{-0.568}\,, \tag{32}$$

where $k_\mathrm{p} = 0.05\,\mathrm{Mpc}^{-1}$ denotes the pivot point. The normalization factor of 2.95×10^{-9} comes in because we use the scalar amplitude $\tilde{A}_\mathrm{s}$ of CMBEASY, which is defined as the primordial power of the curvature fluctuations evaluated at the pivot point, $\tilde{A}_\mathrm{s} \equiv \Delta^2_\mathcal{R}(k_\mathrm{p})$. It is related to the scalar amplitude A_s of CMBFAST, which is used in Sandvik et al. (2004), by $A_\mathrm{s} = \frac{\tilde{A}_\mathrm{s}}{2.95\times10^{-9}}$ (cf. Verde et al. 2003). Finally, we use the physical damping due to the optical depth to last scattering as our last parameter:

$$Z \equiv \mathrm{e}^{-\tau}\,. \tag{33}$$

In order to construct the interpolation of the likelihood surface, we need the transformation that maps the normal parameters back to cosmological parameters. The reason for this is the way we construct the interpolation: our sparse grid algorithm chooses the normal parameters where it wants to refine the grid, which we then need to transform into cosmological parameters to run CMBEASY and the *WMAP* likelihood code. Our technique of inverting the parameter transformation is presented in Appendix A.

3.3 Generation of test set and choice of interpolation range

For choosing the parameter range in which to construct the interpolation, we have run MCMCs containing about 50 000 points at a temperature of $T = 3$. That is, in the Metropolis algorithm we choose the transition probability $a(\boldsymbol{x}, \boldsymbol{y})$ from a point $\boldsymbol{x}$ in the chain to a new point $\boldsymbol{y}$ to be $a(\boldsymbol{x}, \boldsymbol{y}) \equiv \min\{[\mathcal{L}(\boldsymbol{y})/\mathcal{L}(\boldsymbol{x})]^{\frac{1}{T}}, 1\}$. Using this transition probability with $T = 1$ results in the usual Metropolis algorithm, whereas choosing $T = 3$ allows us to explore a larger parameter range than with the regular algorithm. These chains covered a region reaching out to about 25 log-likelihoods around the peak.

The optical depth to the last scattering surface, τ, which can be determined from the CMB polarization, is not well constrained by the *WMAP* polarization data due to their low signal-to-noise ratio. Therefore, when running the MCMCs at $T = 3$, we had to restrict τ to the physically meaningful range $\tau \geq 0$. This restriction corresponds to $Z \leq 1$ for the normal parameters. In the case of the normal parameters in seven dimensions, we had to additionally restrict the intervals to $h_2 \leq 0.52$ and $h_3 \geq 0.38$, which is the range in which the parameter transformation is invertible. Furthermore, we chose to restrict our set of points to be within the 25 log-likelihood region around the peak.

In order to roughly centre our intervals at the maximum of the log-likelihood function, we have determined the latter using a few runs of a simple simplex search.[3] The interval boundaries were then defined as the box centred at the maximum which contains all points of the above-described chains. Note that it is not important for the accuracy of the interpolation that the intervals are well-centred at the maximum. Note further that we have used this set of points as a test set for comparing our interpolation with the real log-likelihood.

3.4 Results

We have interpolated the log-likelihood of the *WMAP* 5 yr data in the six-dimensional normal parameter space described in Section 3.2. The same has been done for a seven-dimensional model, in which we have chosen the running of the spectral index of the primordial power spectrum, α, as an additional parameter. Constructing the interpolation can be parallelized to an arbitrary degree, according to the available computational resources.

For the six-dimensional model, we plot the absolute error of the log-likelihood, $(u - \ln \mathcal{L})$, against the negative *WMAP* log-likelihood, $(-\ln \mathcal{L})$, for the points in the test set in Fig. 7. We have used an interpolation on a sparse grid of level $n = 5$ (consisting of 2561 grid points) in the top panel, and of level $n = 6$ (consisting of 10 625 grid points) in the bottom panel. One clearly sees the improvement in accuracy when increasing the grid level from $n = 5$ to 6. Fig. 8 shows the same plot for the seven-dimensional model, for grid level $n = 6$ (18 943 grid points) in the top panel and $n = 7$ (78 079 grid points) in the bottom panel. We again see the improvement in accuracy with increasing refinement level. However, the additional parameter α is quite strongly correlated with many of the other parameters, whereas the correlations between the normal parameters in six dimensions are reduced to a minimum. We therefore have to increase the grid level by one in seven dimensions, in order to obtain results comparable to the six-dimensional ones. In both figures, we note a systematic negative offset of the interpolation with respect to the real function, which becomes less severe for the higher refinement levels. This offset is due to the fact that we construct a d-linear interpolant of a convex function, which systematically lies below the function. This could be easily coped with by adding a small offset to $\alpha_{1,1}$ after the interpolation or, even better, by using piecewise polynomial instead of the piecewise linear basis functions. We leave the usage of piecewise polynomial basis functions, which promise to be well adapted to the log-likelihood, for future work.

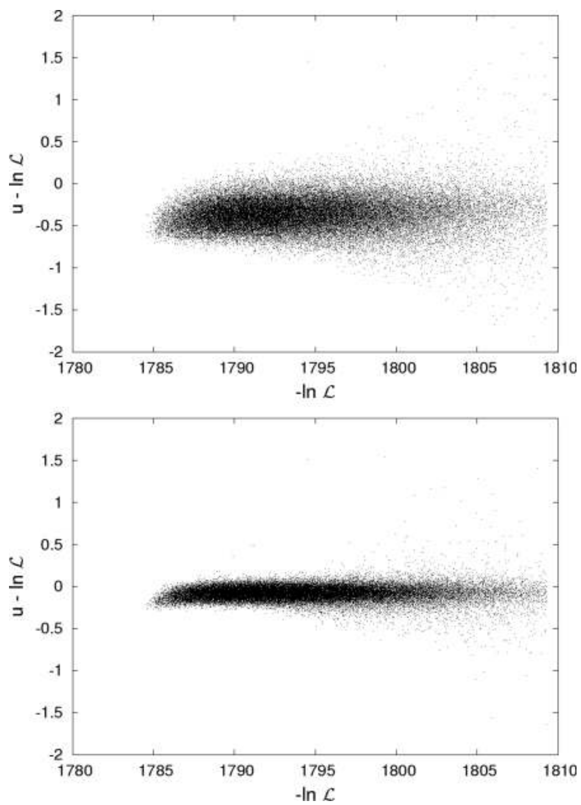

Figure 7. Absolute error of the interpolation with respect to the real log-likelihood in six dimensions for an interpolation with a sparse grid of level 5 (top panel) and of level 6 (bottom panel) for normal parameters.

Note that we have restricted the plot range to $[-2, 2]$, because only 0.1 per cent or less of the points lie outside this range.[4] Almost all of these points lie in the corners of Ω due to relatively strongly correlated parameters. These are the regions in parameter space where the 25 log-likelihood range around the peak extends to the interval boundaries. Due to the extrapolation we use close to the boundaries (cf. the end of Section 2), we obtain relatively large uncertainties in those regions, which do not affect the one-dimensional projections of the likelihood function, though. The uncertainties can be further reduced, spending (adaptively) more grid points in those regions, see also the discussion about adaptivity in Section 3.5.

For the six-dimensional interpolation with a sparse grid of level 6, 2.5 per cent of the test points have an absolute error >0.25 in the log-likelihood and 0.03 per cent of the test points have an absolute error >1. In seven dimensions and for refinement level 7,

[3] We were running several simplex searches and chose the result with the highest value of the log-likelihood. The runs did not all converge to exactly the same point, which we think was due to numerical issues (the log-likelihood was presumably not completely convex, which could be due to the dips we will mention in Section 3.5).

[4] In six dimensions, the number of points outside this range is 0.02 per cent (0.003 per cent) for $n = 5$ $(n = 6)$; in seven dimensions, it is about 0.1 per cent for both grid levels.

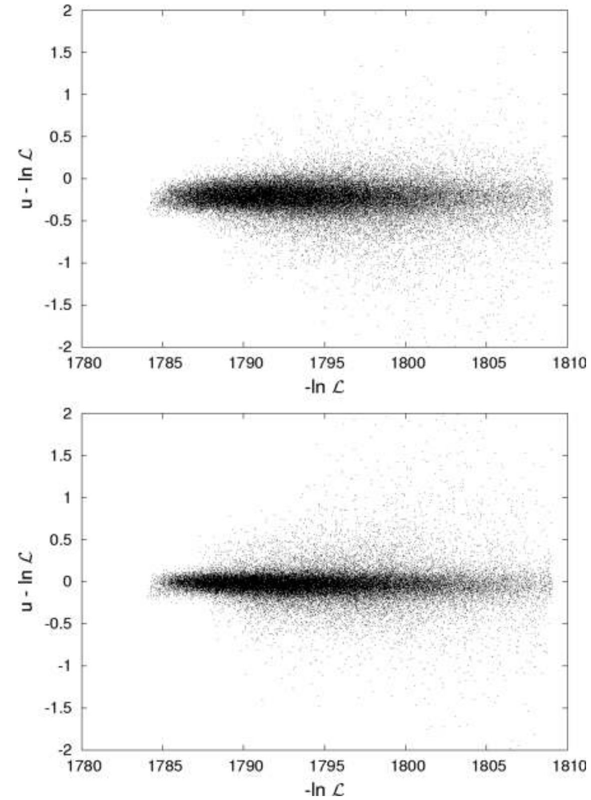

Figure 8. Absolute error of the interpolation with respect to the real log-likelihood in seven dimensions for an interpolation with a sparse grid of level 6 (top panel) and of level 7 (bottom panel) for normal parameters.

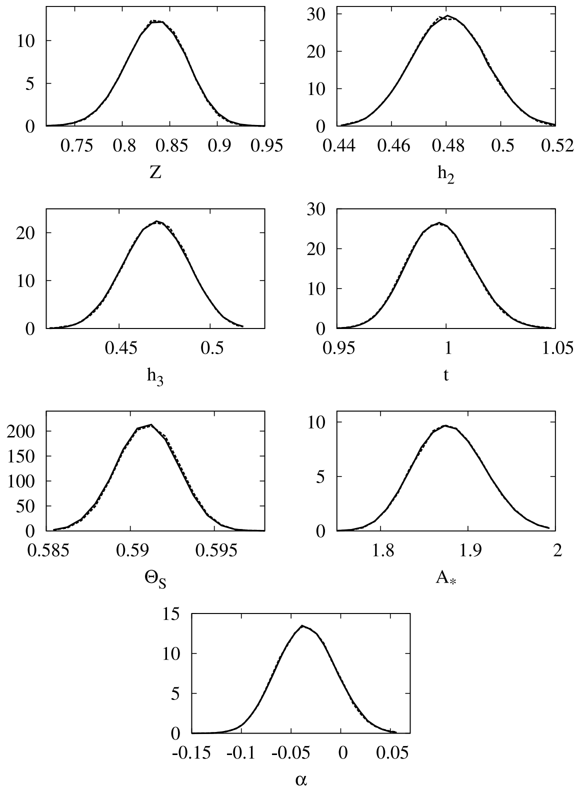

Figure 9. Comparison of the one-dimensional projections of the seven-dimensional *WMAP* 5 yr likelihood (solid) and its interpolation (dashed) using a sparse grid of level $n = 6$ (consisting of 18 943 grid points) for normal parameters. The curves match almost perfectly.

the corresponding numbers are 9 and 0.5 per cent, respectively. This is a higher level of accuracy as reached by Pico (Fendt & Wandelt 2007), for which about 90 per cent of the points in a region reaching out to 25 log-likelihoods around the peak have been calculated with an absolute error below 0.25. However, we note that these numbers for Pico are valid for a nine-dimensional parameter space, whereas we work in six- and seven-dimensional spaces and leave the extension to higher dimensional models to future work. But, we also note that in all settings where a systematic offset in the interpolation error can be observed it is sufficient to reduce the offset to improve our results significantly, in particular for interpolations on lower levels (see e.g. the scatterplot for the six-dimensional model and grid level 5; Fig. 7).

We have projected both the interpolation and the *WMAP* likelihood function using MCMCs of about 150 000 points, and compare the results for the seven-dimensional model for grid level $n = 6$ in Fig. 9. We reproduce the projected one-dimensional likelihood curves almost perfectly. The results for the six-dimensional model for grid level $n = 5$ are very similar to the seven-dimensional ones. Note that we obtain these excellent results for the six-dimensional model using only 2561 grid points. The visual comparison of our results with the projected one-dimensional likelihoods obtained by CosmoNet (Auld et al. 2007, 2008) shows that we reproduce the original curves with a higher accuracy than the latter. Note also that our interpolation is constructed in a rather wide region, encompassing about 25 log-likelihoods around the peak, whereas in Auld et al. (2008) the region in which $\ln \mathcal{L}$ was fitted covers only 4σ around the peak for the combined likelihood of CMB and large-scale structure (LSS). This corresponds to a region of about eight log-likelihoods around the peak for the combined likelihood, and even less when using only the CMB likelihood.

Consider now the interpolation of the *WMAP* likelihood surface using directly the standard parameters, which are used by default when doing cosmological parameter sampling with the MCMC driver from CMBEASY (Doran & Müller 2004): $\{w_\mathrm{m}, w_\mathrm{b}, h, \tau, n_\mathrm{s}, \ln(10^{10} A_\mathrm{s}) - 2\tau\}$, to which we again add α as an additional parameter in the seven-dimensional case. Working with these parameters has the advantage that we do not have to restrict ourselves to the parameter range in which the parameter transformation is invertible. However, the problem is now less adapted to our choice of basis functions, due to the stronger correlations between the different parameters. We therefore pay the price of having to increase the grid level by one in this case in order to reach an accuracy as good as before. We show the absolute error of the log-likelihood, $(u - \ln \mathcal{L})$, against the negative *WMAP* log-likelihood, $(-\ln \mathcal{L})$, for the six-dimensional model for grid level $n = 6$ (10 625 points) and $n = 7$ (40 193 points) in Fig. 10[5] and for the seven-dimensional one for $n = 7$ (87 079 points) and $n = 8$ (297 727 points) in Fig. 11.[6] For the six-dimensional (seven-dimensional) interpolation with a sparse grid of level $n = 7 (n = 8)$, the fraction of test points with absolute error >0.25 in the log-likelihood is 6 per cent (20 per cent), and 0.5 per cent (2.5 per cent) for an absolute error >1. Again, the one-dimensional projections of the six-dimensional case for level

[5] Here, about 0.3 per cent (0.2 per cent) of the points in the test set lie outside the chosen plot range for the grid of level $n = 6$ ($n = 7$).

[6] About 1 per cent of the points in the test set lie outside the chosen plot range for the grid of both level $n = 7$ and level $n = 8$.

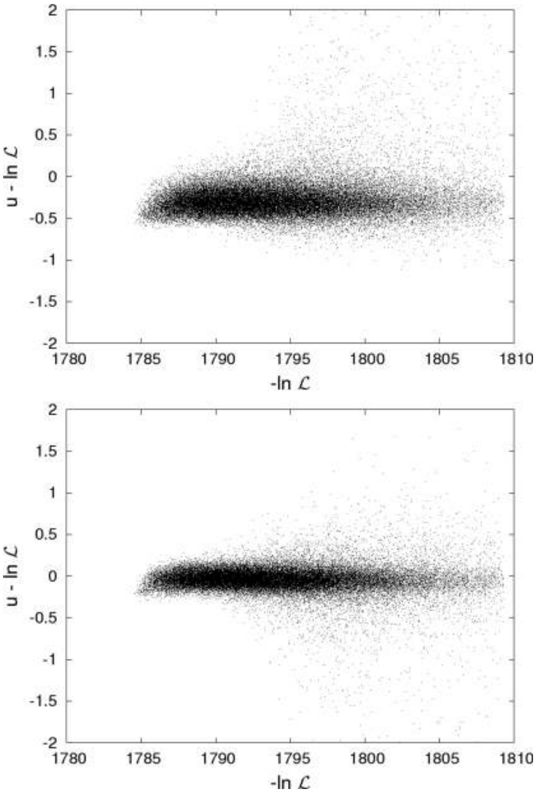

Figure 10. Absolute error of the interpolation with respect to the real log-likelihood in six dimensions for an interpolation with a sparse grid of level 6 (top panel) and level 7 (bottom panel) for standard parameters.

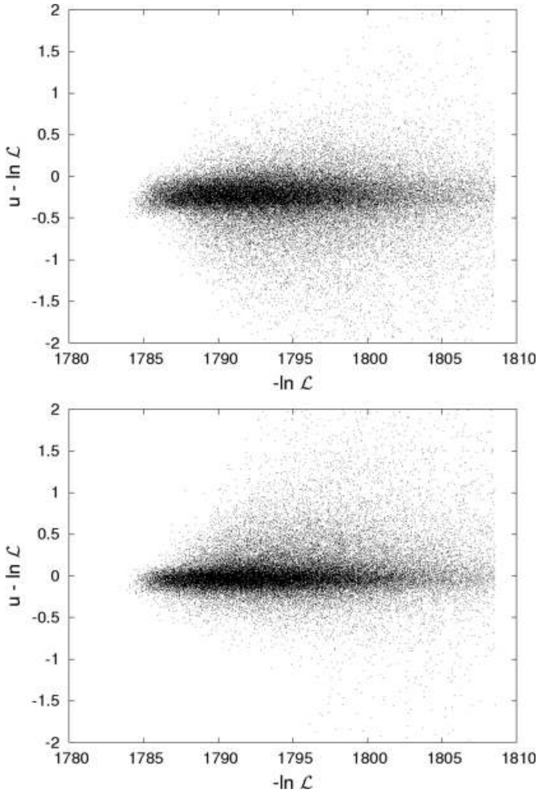

Figure 11. Absolute error of the interpolation with respect to the real log-likelihood in seven dimensions for an interpolation with a sparse grid of level 7 (top panel) and level 8 (bottom panel) for standard parameters.

$n = 6$ and of the seven-dimensional case for level $n = 7$ are very similar to the ones in Fig. 9, and are thus not shown.

We have tested the evaluation time of our interpolation by evaluating a sparse grid interpolant of level 6 in six dimensions for 2 000 000 points randomly chosen from within Ω. On a conventional desktop computer (Intel chipset, 2.8 GHz), this took about 92 μs per point, including the random generation of the point. In seven dimensions on the same level we have twice as many grid points and one dimension more, which doubles the evaluation time to 189 μs. For CosmoNet and Pico, the evaluation of a six-dimensional model is specified to take about 10 and 250 μs, respectively (Auld et al. 2008). Note that we do not know on which hardware the evaluation times of CosmoNet and Pico have been measured, which makes a comparison hardly possible. Note further that our code to evaluate a sparse grid function is not optimized for fast evaluation times and that there is still room for improvement. In any case, for all of these codes the bottleneck in cosmological parameter sampling is now the MCMC algorithm itself rather than the evaluation of the likelihood, at least with the MCMC driver used in this work (Doran & Müller 2004).

Note that in seven dimensions we need significantly more grid points than in six dimensions, since the additional parameter α strongly correlates with the other parameters, and we thus need to increase the grid level by one to obtain good results. As the storage requirements are rather low, this mainly increases the number of evaluations that are needed for constructing the sparse grid interpolation. As already stated before, though, the construction of the interpolation can be parallelized to an arbitrary degree, according to available computational resources, so that this should not be an issue. To store the interpolant for a regular sparse grid in d dimensions for level l with N grid points, we would only need N doubles for the coefficients and two integers to remember both d and l, leading to $(N + 1)8$ Bytes. For adaptively refined sparse grids, we additionally have to store at least which grid points have been refined, requiring slightly more storage. For current hardware architectures, the size of the memory is therefore not a limiting factor for our application.

We have shown in this section that the interpolation of the log-likelihood with regular sparse grids provides excellent and competitive results in both six and seven dimensions, even when interpolating in the rather wide range covering about 25 log-likelihoods around the peak. The memory requirements to store the sparse grid are rather low and the evaluation times are very fast. We have demonstrated that the one-dimensional projections of the interpolants match the original ones almost perfectly, and that the scatterplots of the errors exhibit only a small fraction of grid points with higher errors. For standard parameters, more grid points have to be spent to obtain the same accuracy than for normal parameters, as the latter ones correlate less. In the next section, we will therefore focus on an adaptive extension of the method used so far to further reduce the number of grid points that are required.

3.5 Improvements with adaptive sparse grids

As it has already been mentioned, the log-likelihood is not a perfect sum of one-dimensional functions. The different parameters contribute differently to $\mathcal{L}$ and correlate more or less with each other. It is therefore reasonable, especially when using the standard parameters which correlate more, to employ adaptivity, spending more grid points in critical regions and less grid points elsewhere. In this section, we demonstrate the utility of adaptivity by showing some first results as a proof of concept. As they can clearly be improved further, we leave a thorough study of adaptive sparse grids for the interpolation of $\ln \mathcal{L}$ to future work. We start by specifying how to refine, we formally derive a suitable criterion that can be used to specify where to refine as well as to measure the quality of an interpolation, and we finally provide results that show how adaptivity can improve the interpolation obtained for regular sparse grids.

Employing adaptivity, one can attempt to either obtain better results fixing roughly the number of grid points used or achieve a similar accuracy using less grid points. In the following, we show results for the former, tackling the seven-dimensional example using the standard parameters on level 7 with 78 079 grid points presented above. We start with a regular sparse grid of some low level and refine grid points, creating all $2d$ children in the hierarchical structure (if possible) each, until the grid size exceeds 78 000 grid points. In settings where the contributions of the dimensionalities differ significantly, it can be useful to start with level 2 to allow dimensional adaptivity, neglecting unimportant dimensions; here, the grid points on low levels will be created in any case, so we can start with a sparse grid on level 5, e.g. to save on the number of adaptive steps.

Choosing a suitable refinement criterion, it can be determined whether to refine in a broad way (close to regular sparse grids) or in a more greedy way in the sparse grid's hierarchical tree structure. It is reasonable to take the surpluses of the grid points into account as they contain the local information about the functions, i.e. if the function has a high gradient locally. Furthermore, they decay quickly with increasing level sum in the convergent phase. The mere surplus-based criterion, refining the grid points with the highest absolute value of the surplus first, is known to tend to minimize the L_2 norm of the error. As we do not spend grid points on the domain's boundary, and as we are extrapolating towards the boundary, the biggest surpluses per level can be found for the modified basis functions which are adjacent to the boundary. A mere surplus-based criterion will therefore only refine towards the boundary. This reduces the error especially for sampling points with a high error in the scatterplots, as they are located towards the boundary of the domain.

In the following, we theoretically derive a refinement criterion which is better suited to our problem than the purely surplus-based one. In order to maximize the information our interpolation contains about the real likelihood, we attempt to minimize the Kullback–Leibler distance d_{KL} between the interpolation and the likelihood function,

$$\begin{aligned} d_{\mathrm{KL}} &\equiv \int \mathrm{d}^d x\, \mathcal{L}(\boldsymbol{x}) \ln \frac{\mathcal{L}(\boldsymbol{x})}{\exp[u(\boldsymbol{x})]} \\ &= \int \mathrm{d}^d x\, \mathcal{L}(\boldsymbol{x}) \left[\ln \mathcal{L}(\boldsymbol{x}) - u(\boldsymbol{x})\right], \end{aligned} \tag{34}$$

which is defined for two normalized probability distributions $\mathcal{L}$ and $\exp(u)$. Let us now derive the refinement criterion we obtain from minimizing d_{KL}. Assume that we have already computed an interpolation $u(\boldsymbol{x})$ with N grid points, then the Kullback–Leibler distance d_{KL} when evaluating the function at an additional point $\boldsymbol{x}_{N+1}$ is changed by

$$\begin{aligned} \left|d_{\mathrm{KL}}^{\mathrm{new}} - d_{\mathrm{KL}}^{\mathrm{old}}\right| &= \left| \int \mathrm{d}^d x\, \mathcal{L}(\boldsymbol{x}) \Bigg[\ln \mathcal{L}(\boldsymbol{x}) - u^{\mathrm{new}}(\boldsymbol{x}) \right. \\ &\qquad \left. - \ln \mathcal{L}(\boldsymbol{x}) + u^{\mathrm{old}}(\boldsymbol{x}) \Bigg] \right| \\ &= \left| \int \mathrm{d}^d x\, \mathcal{L}(\boldsymbol{x}) \left[u^{\mathrm{old}}(\boldsymbol{x}) - u^{\mathrm{new}}(\boldsymbol{x})\right] \right| \\ &= \left| \int \mathrm{d}^d x\, \mathcal{L}(\boldsymbol{x}) \left[\sum_{i=1}^{N} \alpha_i \varphi_i(\boldsymbol{x}) - \sum_{i=1}^{N+1} \alpha_i \varphi_i(\boldsymbol{x}) \right] \right| \\ &= \left| \int \mathrm{d}^d x\, \mathcal{L}(\boldsymbol{x}) \left[\alpha_{N+1} \varphi_{N+1}(\boldsymbol{x})\right] \right| . \end{aligned} \tag{35}$$

If we refine the interpolation around the grid point that contributed most to the Kullback–Leibler distance, we can hope to converge towards the minimum of d_{KL} fastest. In order to obtain a suitable refinement criterion, we have to simplify the formula in (35) considerably. We thus assume the likelihood $\mathcal{L}(\boldsymbol{x})$ as well as the basis function $\varphi_{N+1}(\boldsymbol{x})$ to be locally constant on φ_{N+1}'s support, obtaining

$$\left|d_{\mathrm{KL}}^{\mathrm{new}} - d_{\mathrm{KL}}^{\mathrm{old}}\right| \sim V_{N+1} \mathcal{L}(\boldsymbol{x}_{N+1}) \left|\alpha_{N+1}\right| , \tag{36}$$

where V_{N+1} is the volume covered by the basis function φ_{N+1} (i.e. its support), and we have used $\varphi_{N+1}(\boldsymbol{x}_{N+1}) = 1$.

With (36), we have derived an estimation of the contribution of a basis function to d_{KL}, which is a reasonable refinement criterion in our setting. In addition to the surplus of the grid point, $|\alpha_{N+1}|$, it takes into account the value of the likelihood $\mathcal{L}(\boldsymbol{x}_{N+1})$ at the grid point and the volume of the basis function V_{N+1}. The likelihood takes care of the fact that we would like to be more accurate where the likelihood is higher. The regions of very low likelihood are less interesting for us – the likelihood being already very close to zero beyond a difference of about 20 log-likelihoods. The volume factor, on the other hand, prevents the interpolation to refine too deeply (to very high grid levels) locally in the parameter space. However, since this usually only takes effect after several refinement steps, and as we have restricted the number of grid points, we choose not to include the volume factor but rather to refine several points at the same time, which addresses this issue in an alternative way and which will be discussed later on. We further choose to introduce a temperature T again, which allows us to weight the likelihood with respect to the surplus and thus to influence how much to refine close to the maximum. The refinement criterion we used in this study is thus

$$\left[\frac{\mathcal{L}(\boldsymbol{x}_{l,i})}{\mathcal{L}_{\max}}\right]^{1/T} \left|\alpha_{l,i}\right| , \tag{37}$$

where we have divided the likelihood by its peak value, $\mathcal{L}_{\max}$ (which we have already obtained determining the interpolation domain), because the *WMAP* code returns the log-likelihood only up to a constant offset, so that we do not know the correct normalization of $\mathcal{L}$. For $T = 1$, refinement takes place only very close to the maximum as $\mathcal{L}$ decays quickly; a temperature of $T = 6$ showed to provide good results within the whole domain of interest.

Refining only one grid point per refinement step often causes adaptivity to get stuck in a single, special characteristic of the function. Interpolating $\ln \mathcal{L}$ with our choice of basis functions, all grid points are likely to be created only in the direction where the log-likelihood decays fastest, or around one of the local dips we will address later on. Refining more than one grid point at the same time helps to circumvent such effects, resulting in a broader refinement scheme.

The Kullback–Leibler distance d_{KL} can also be used to measure the quality of our interpolation: the distance between the real likelihood and the interpolation should be as small as possible. However, as we have already mentioned above, we do not know the normalization of the *WMAP* likelihood function. Therefore, d_{KL} is not necessarily positive and thus looses its property of being a useful

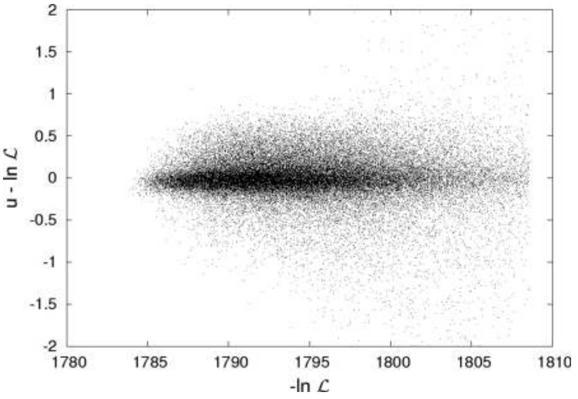

Figure 12. Absolute error of the interpolation with respect to the real log-likelihood in seven dimensions for an interpolation with an adaptively refined sparse grid for standard parameters.

measure of the 'closeness' of the two functions. We thus use a slightly modified version,

$$\widehat{d}_{\mathrm{KL}} \equiv \int \mathrm{d}^d x\, \mathcal{L}(\boldsymbol{x}) | \ln \mathcal{L}(\boldsymbol{x}) - u(\boldsymbol{x}) | \,, \qquad (38)$$

as a measure of the quality of our interpolation, instead of the actual Kullback–Leibler distance. It can be easily calculated from an MCMC with $T = 1$ obtained for $\mathcal{L}$, by simply averaging the absolute errors $|\ln \mathcal{L}(\boldsymbol{x}_i) - u(\boldsymbol{x}_i)|$ over all points in the chains. Furthermore, we quote this value averaged over a chain of $T = 3$ (exploiting the interpolation domain better), which corresponds to $\int \mathrm{d}^d x\, \mathcal{L}(\boldsymbol{x})^{\frac{1}{T}} |\ln \mathcal{L}(\boldsymbol{x}) - u(\boldsymbol{x})|$.

Fig. 12 shows the scatterplot for an adaptively refined sparse grid in seven dimensions. Starting from a regular grid of level 5, we refined 100 grid points each according to the refinement criterion (37) with $T = 6$. Needing only about as much grid points (78 551) as for the regular sparse grid of level 7, Table 1 shows that we obtain results which are close to those of a regular sparse grid of level 8 with almost four times as many grid points. We provide the mean squared error (MSE) as well as $\widehat{d}_{\mathrm{KL}}$ for both $T = 1$ and 3 chains for regular sparse grids of levels 7 and 8 and for the adaptively refined case. We also quote how many points exhibit an absolute error larger than 1 or 0.25 for the $T = 3$ chains. We do not show the histograms of the adaptively refined model, as the histograms for both the regular grid on level 8 and the adaptively refined one are very close to the histograms for normal parameters in seven dimensions shown in Fig. 9.

We would like to mention that, due to numerical problems, the current version of CMBEASY produces local, unphysical and sometimes rather high dips. This problem is already known and will be corrected in the next release. For stochastic approaches, this is not a big problem though: The dips are local and just cause some noisy evaluations. But, it poses a problem for numerical approaches if a grid point hits a dip. Then, it can happen that spending more grid points can even deteriorate the results. For example with our regular grid in six dimensions using the standard parameters, increasing the level from 7 to 8 caused a higher overall error in the chain data used for the histograms. In particular two new basis functions close to the peak caused an error of up to 12 of the log-likelihood for all evaluations affected by those basis functions.

Fortunately, dips can be detected automatically due to the hierarchical structure of the sparse grid and the smoothness of ln $\mathcal{L}$, using a criterion that is once more based on the surpluses. Furthermore, it is not a severe problem when using adaptivity, as adaptivity localizes the effects of the dips automatically. One just has to take care not to spend too much grid points to compensate for the dips.

The first adaptive results are promising, but there is still room for a lot of improvement. Even better refinement criteria than those used so far could be employed. Using not only piecewise linear functions, but rather piecewise polynomials, and applying adaptivity in both the mesh width and the polynomial degree are very promising; especially, the extrapolation properties towards the boundary would be improved, and less grid points would be needed to obtain the same accuracies.

4 CONCLUSIONS

In this work, we have explored the utility of interpolating the *WMAP* log-likelihood surface using sparse grids. We demonstrated that the results are excellent and competitive to other approaches regarding speed and accuracy, and we discussed advantages over fitting the likelihood surface with polynomials (Sandvik et al. 2004; Fendt & Wandelt 2007) or neural networks (Auld et al. 2008):

The interpolation based on sparse grids converges towards the exact function in the limit of the grid level going to infinity. We can therefore reach an arbitrary accuracy by simply increasing the amount of work we spend. In the case of a polynomial fit, this is not guaranteed since increasing the polynomial degree runs the risk of becoming unstable.

In order to construct the sparse grid interpolation, we do not need to sample a set of training points using MCMCs beforehand, since the sampling points are determined by the sparse grid structure which is given a priori. Once we have chosen the volume of interest, the time for constructing the interpolation is dominated by the evaluation of the likelihood function at the grid points. We do not need additional training time as for neural networks (Auld et al. 2008), for example. Constructing the interpolation can thus be done almost arbitrarily in parallel, only limited by the computational resources that are available.

The sparse grid technique is rather general and not restricted to certain classes of functions. In particular, the choice of sampling

Table 1. Comparison of errors of regular sparse grids of levels 7 and 8, respectively, and an adaptively refined sparse grid using approximately as many grid points as contained in the regular grid of level 7. Shown are the number of points with an absolute error larger than 1 or 0.25 in the $T = 3$ chains, the MSE for chains of $T = 1$ and 3 and $\widehat{d}_{\mathrm{KL}}$, which denotes the absolute value of the error averaged over chains of $T = 1$ and 3.

	Err > 1, $T = 3$	Err > 0.25, $T = 3$	MSE, $T = 1$	MSE, $T = 3$	$\widehat{d}_{\mathrm{KL}}$, T = 1	$\widehat{d}_{\mathrm{KL}}$, $T = 3$
	(per cent)	(per cent)				
Level 7	4.2	50.5	0.087	0.532	0.256	0.354
Level 8	2.3	19.3	0.017	0.210	0.091	0.193
Adaptive	1.8	23.6	0.027	0.202	0.110	0.204

points and basis functions is not tailored to a single problem as for neural networks, where the topology of the network has to be chosen problem specifically (often in a heuristic way). The sparse grid interpolation technique as well as our extensions can therefore be readily applied to other problems in astrophysics and cosmology, and will be useful in further tasks, where an accurate interpolation of a function is needed.

The excellent performance of the sparse grid interpolation can be further improved, leaving future research to do: it can be applied to models with more than seven parameters by spending more computational effort. Further modification of the basis functions, for example allowing for a piecewise polynomial interpolation, promises better convergence rates and higher accuracies. Adaptive refinement schemes, which take into account the characteristics of the interpolated function, can be used to further increase the accuracy of the interpolation, as we have already demonstrated for a first example in this work.

ACKNOWLEDGMENTS

The authors would like to thank Georg Robbers for his help with CMBEASY and Stefan Zimmer and the anonymous referee for helpful discussions and comments. We acknowledge the use of CMBEASY and the *WMAP* likelihood code.

REFERENCES

Auld T., Bridges M., Hobson M. P., Gull S. F., 2007, MNRAS, 376, L11
Auld T., Bridges M., Hobson M. P., 2008, MNRAS, 387, 1575
Bellman R., 1961, Adaptive Control Processes: A Guided Tour. Princeton Univ. Press, Princeton, NJ
Bungartz H.-J., Griebel M., 2004, Acta Numer., 13, 147
Bungartz H.-J., Pflüger D., Zimmer S., 2008, in Bock H., Kostina E., Hoang X., Rannacher R., eds, Proc. High Performance Scientific Computing 2006, Modelling, Simulation and Optimization of Complex Processes, Hanoi, Vietnam Adaptive Sparse Grid Techniques for Data Mining. Springer, New York, p. 121
Chu M., Kaplinghat M., Knox L., 2003, ApJ, 596, 725
Coles P., Lucchin F., 2002, Cosmology: The Origin and Evolution of Cosmic Structure, 2nd edn. John Wiley & Sons, New York
Doran M., 2005, J. Cosmol. Astropart. Phys., 10, 11
Doran M., Müller C. M., 2004, J. Cosmol. Astropart. Phys., 9, 3
Fendt W. A., Wandelt B. D., 2007, preprint (arXiv:0712.0194)
Ganapathysubramanian B., Zabaras N., 2007, J. Comput. Phys., 225, 652
Garcke J., 2006, in Cohen W. W., Moore A., eds, ICML '06: Proc. 23rd International Conference on Machine Learning, Regression with the Optimised Combination Technique. ACM Press, New York, p. 321
Garcke J., Hegland M., 2009, Computing, 84, 1
Garcke J., Griebel M., Thess M., 2001, Computing, 67, 225
Habib S., Heitmann K., Higdon D., Nakhleh C., Williams B., 2007, Phys. Rev. D, 76, 083503
Holtz M., 2008, Dissertation, Institut für Numerische Simulation, Universität Bonn
Hu W., Fukugita M., Zaldarriaga M., Tegmark M., 2001, ApJ, 549, 669
Jimenez R., Verde L., Peiris H., Kosowsky A., 2004, Phys. Rev. D, 70, 023005
Kaplinghat M., Knox L., Skordis C., 2002, ApJ, 578, 665
Klimke A., Nunes R., Wohlmuth B., 2006, Int. J. Uncertainty Fuzziness Knowledge-Based Syst., 14, 561
Kosowsky A., Milosavljevic M., Jimenez R., 2002, Phys. Rev. D, 66, 063007
Lewis A., Challinor A., Lasenby A., 2000, ApJ, 538, 473
Page L. et al., 2003, ApJS, 148, 233
Reisinger C., Wittum G., 2007, SIAM J. Sci. Comput., 29, 440
Sandvik H. B., Tegmark M., Wang X., Zaldarriaga M., 2004, Phys. Rev. D, 69, 063005
Seljak U., Zaldarriaga M., 1996, ApJ, 469, 437
Tauber J. A., 2000, Astrophys. Lett. Communications, 37, 145
Temlyakov V. N., 1993, Approximation of Periodic Functions. Nova Science Publishers, Hauppauge
Verde L. et al., 2003, ApJS, 148, 195
von Petersdorff T., Schwab C., 2006, Appl. Math., 51, 145
Widmer G., Hiptmair R., Schwab C., 2008, J. Comput. Phys., 227, 6071
Zenger C., 1991, in Hackbusch W., ed., Notes on Numerical Fluid Mechanics, Vol. 31, Parallel Algorithms for Partial Differential Equations, Sparse Grids. Vieweg, Wiesbaden, p. 241

APPENDIX A: INVERSION OF THE PARAMETER TRANSFORMATION

In the following, we present a technique of inverting the parameter transformation of Section 3.2 to compute the cosmological parameters given the normal parameters. The normal parameter h_2 in terms of cosmological parameters is given by

$$h_2(w_\mathrm{m}, w_\mathrm{b}) = 0.0264\, w_\mathrm{b}^{-0.762} \exp\left\{-0.476\, [\ln(25.5\, w_\mathrm{b} + 1.84\, w_\mathrm{m})]^2\right\} . \tag{A1}$$

We solve this equation for w_m as a first step:

$$w_\mathrm{m}(h_2, w_\mathrm{b}) = \left\{ \exp\left\{ \pm \left[-\frac{1}{0.476} \ln\left(\frac{h_2}{0.0264}\, w_\mathrm{b}^{0.762}\right)\right]^{1/2} \right\} - 25.5\, w_\mathrm{b} \right\} \frac{1}{1.84} . \tag{A2}$$

Inconveniently, there exist two different solutions for $w_\mathrm{m}(h_2, w_\mathrm{b})$, which complicates the inversion. We now substitute w_m in $h_3(w_\mathrm{m}, w_\mathrm{b})$ (30) for (A2) and thus obtain $h_3(h_2, w_\mathrm{b})$, which, of course, has two solutions as well. An example of the two branches of $h_3(h_2, w_\mathrm{b})$ for $h_2 = 0.45$ is depicted in Fig. A1. We can compute the critical point where only one solution exists using the condition

$$-\frac{1}{0.476} \ln\left(\frac{h_2}{0.0264}\, w_\mathrm{b}^{0.762}\right) = 0 , \tag{A3}$$

as can be seen from (A2). This condition gives us the following formulae for the parameter values at the critical point:

$$w_\mathrm{b,crit}(h_2) = \left(\frac{0.0264}{h_2}\right)^{1/0.762} , \tag{A4}$$

$$w_\mathrm{m,crit}(h_2) = \left(1 - 25.5\, w_\mathrm{b,crit}\right) \frac{1}{1.84} , \tag{A5}$$

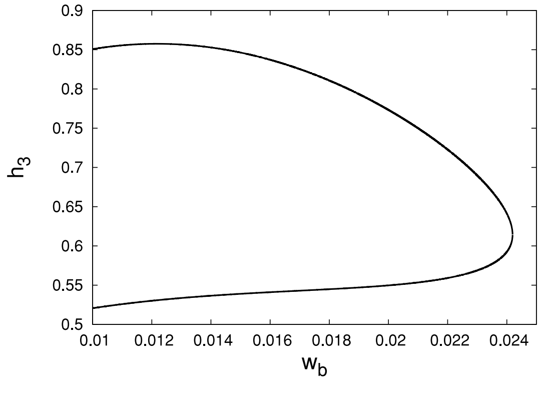

Figure A1. The two branches of h_3 versus w_b for $h_2 = 0.45$.

$$h_{3,\mathrm{crit}}(h_2) = 2.17\left[1+\left(\frac{w_{\mathrm{b,crit}}}{0.044}\right)^2\right]^{-1} w_{\mathrm{m,crit}}^{0.59}$$
$$\left[1+1.63\left(1-\frac{w_{\mathrm{b,crit}}}{0.071}\right) w_{\mathrm{m,crit}}\right]^{-1}. \qquad \text{(A6)}$$

The two parameters h_2 and h_3 can now be inverted to w_m and w_b. For a given h_2, we express h_3 in terms of h_2 and w_b, as described above. We then use $h_{3,\mathrm{crit}}(h_2)$ to choose the upper branch of $h_3(h_2,\, w_\mathrm{b})$ if our given h_3 is bigger than $h_{3,\mathrm{crit}}(h_2)$ and the lower branch if it is smaller. Using the respective branch of $h_3(h_2,\, w_\mathrm{b})$, we search numerically in w_b until $h_3(h_2,\, w_\mathrm{b})$ matches the given h_3. Substituting that value of w_b into equation (A2), we readily obtain the value for w_m.

Now, it is straightforward to compute the values for n_s and A_s from t and A_*. To obtain h from Θ_s, we follow the procedure suggested by Kosowsky et al. (2002), expressing Θ_s in terms of h and then searching in h numerically until $\Theta_\mathrm{s}(h)$ matches the given value of Θ_s.

This paper has been typeset from a TEX/LATEX file prepared by the author.