6th World Congresses of Structural and Multidisciplinary Optimization
Rio de Janeiro, 30 May - 03 June 2005, Brazil

A Cosmic Microwave Background Radiation Map Making Method Using Simulated Annealing

Ricardo Varela Corrêa[1], Thyrso Villela[1], Carlos Alexandre Wuensche[1], Jorge Mejia[1], Airam Jonatas Preto[2], Stephan Stephany[2]

(1) Divisão de Astrofísica – DAS/INPE, e-mail ricardo@cea.inpe.br, [villela,alex,mejia]@das.inpe.br
(2) Laboratório de Computação e Matemática Aplicada – LAC/INPE, e-mail [airam,stephan]@lac.inpe.br
Instituto Nacional de Pesquisas Espaciais - INPE, Av. dos Astronautas,1758, CEP 12227-010, S. José dos Campos, SP, Brasil

Abstract

The making of Cosmic Microwave Background (CMB) maps consists in reducing a large time-ordered data set of temperatures acquired by a telescope into a map of sky temperatures. The CMB map is discretized in pixels according to the resolution of the telescope and contains a set of estimated sky temperature derived from the acquired temperatures. CMB maps allow further analysis in order to estimate cosmological parameters. The standard method for making CMB maps was developed by the team of the COBE satellite. It is based on the least squares approach, and leads to a system of linear equations derived from a vector of acquired temperatures and a pointing matrix determined by the telescope scans. A commonly used approach for reducing the computational complextity of CMB map making is the *bin average* method, that consists in averaging the temperatures successively acquired for each pixel. This work proposes an innovative approach for CMB map making that is formulated as an implicit inverse problem. The associated optimization problem is solved by a simulated annealing algorithm that was parallelized using calls to the MPI (Message Passing Interface) communication library. The proposed method was tested using simulated data that corresponds to a 30-day acquisition by the BEAST experiment corrupted with gaussian noise. The numerical results (CMB map) are compared to those obtained by the *bin average* method for the same simulated data. It is shown that the proposed CMB method yielded a better quality map in a similar processing time.

Keywords: CMB, simulated annealing, MPI, high performance cluster computing.

1. Introduction

The making of Cosmic Microwave Background (CMB) maps consists in reducing a large time-ordered data set temperatures acquired by a telescope into a map of sky temperatures of the observed region of the sky [1]. The CMB map is discretized in $l \times l$ pixels according to the resolution of the telescope and contains a set of estimated sky temperature derived from the acquired temperatures. A pointing matrix correlates the measurements to the points of the sky, being defined by the pointing strategy of the instrument. CMB maps are employed to further analysis in order to estimate cosmological parameters that are of great interest in Cosmology.

The map making process constitutes the inverse problem of estimating temperature values from temporal series of noisy temperature measurements (TOD). The standard method was developed by the team of the COBE satellite. It is based on the least squares approach, and leads to a system of linear equations derived from a vector of acquired temperatures and the pointing matrix. Most other methods are derived from this method, as described by Tegmark [4]. A very common approach to reduce the high computational complexity of this method is the *bin average* method, that averages the temperatures successively acquired for each pixel.

A map is composed of pixels that are valued according to the estimated sky temperatures and stored in a vector m that has dimension $l \times l$ and represents a possible solution. The TOD is stored in a vector of measured temperatures (d) that has dimension $s \times l \times l$. This vector is correlated to m by a known pointing matrix P with dimension $[s \times l \times l] \times [l \times l]$, in a way that $l \times l$ sets of rows of this matrix store the measurement of all pixels of the observed region of the sky. Measured temperatures d can be mapped to the sky map temperatures m by the following equation:

$$d = P\,m + n \tag{1}$$

where n is a $l \times l$ vector that expresses the noise contribution. The BEAST experiment scanning strategy implies in a particular arrangement for the P matrix: in a given i-th row, there is only a unique non zero element in the j-th column with unitary value, that corresponds to the observed j-th pixel, expressed by the Kroenecker delta function (δ):

$$P_{ij} = \delta_{Pi,j} \tag{2}$$

The standard map making technique employs the generalized least square approach and the noise inverse covariance matrix N^{-1} that is estimated from the observed data. The best candidate solution (map) must minimize the difference between the estimated temperature of each pixel and the corresponding measured temperatures of the TOD, as follows:

$$\chi^2 = n^t N^{-1} n = \left(d^t - m^t P^t\right) N^{-1} \left(d - Pm\right) \tag{3}$$

The desired map is then given by the following linear estimator:

$$\tilde{m} = \left(P^t N^{-1} P\right)^{-1} P^t N^{-1} d\,. \tag{4}$$

Given the large amount of data, the inversion of the system in Eq. (4) is computationally heavy, as estimated by Borril [1]. The *bin average* method is used to produce the desired temperature vector without requiring the inversion of the noise covariance matrix [12]. This method take advantage of the noise vector being considered to have zero mean: the estimated temperature for each pixel would be simply given by the average of its measurements. In a first step, one-hour set of data are pixelwise averaged. In the next step, the hourly averages are averaged for each week and, finally, the week averages are averaged for an entire year. The resulting vector is then plotted as a CMB map. However, these averages must be weighted by the standard deviation of the *s* measurements, for each pixel of a particular TOD. The *bin average* of several days of data is performed as a weighted sum given by:

$$\mu_i = \sum_{i=0}^{l} \bar{x}_i \, \sigma_i^{-2} \left(\sum_{i=0}^{l} \sigma_i^{-2} \right)^{-1} \tag{5}$$

where $\bar{x}$ is the average of the temperature vector and σ is the standard deviation for each pixel *i* estimated for each of the 1-hour data set.

This work proposes an innovative approach for CMB map making that is formulated as an implicit inverse problem. The associated optimization problem is solved by a simulated annealing algorithm (SA) [2] that was parallelized using calls to the MPI (Message Passing Interface) communication library. The proposed method was tested using simulated data of the Background Emission Anisotropy Scanning Telescope (BEAST) experiment [3,11]. This data is corrupted with gaussian noise and arranged in 1-hour measurement periods that correspond to the observed region of the sky. During an hour period, each pixel is sampled *20,000* times (*s*) and the region of the sky has *8,000* pixels. The simulated data corresponds to a TOD for a 30-day duration.

The SA is described in the following section, followed by a section of numerical results and a last section with conclusions.

2. The implementation of the simulated annealing method

Simulate Annealing is a heuristic approach for optimization problems. The name derived from a metallurgy technique involving the heating of a material followed by its controlled cooling in order to achieve molecular stability. The heat causes the molecules to move from their position through states of energy. The slow cooling increase the probability of reaching a more stable, lower energy state in comparison to the initial state.

The SA algorithm associates an energy to each point of the search space and tries to reach an energy minimum. This energy is calculated according to the specific optimization goal of the problem. The algorithm starts at an arbitrary point of the search space. At every step, it randomly chooses a neighboring point and moves to that point according to a certain probability. This probability is a function of the energy difference between the two points and a global time-dependent parameter defined as the temperature of the system.

The pseudo-code listed in figure 1 performs the simulated annealing algorithm[14] starting from an initial state S_0 and initial temperature T_0. The *newstate* function provides a move to a new state S_i with a new temperature T_i. The *objective* function then computes the new energy E_i. for the state S_i. The *fitprob* function returns a probability at which a transition to a state of higher energy takes place. The *schedule* function returns a new temperature at each iteration provided by a counter of the annealing schedule.

```
    Initialize(S0,T0)

    E := objective(S0)
    i := 0

    while not stop_condition

Si := newstate(S,T)
Ei := objective(Si)
If [Ei < E] or [random() < fitprob(E , T)]
S := Si
E := Ei
T := schedule(T , i)
i : i + 1

end while
end program
```

Figure 1. Pseudo-code of the simulated annealing algorithm

The above pseudo-code denotes an initial energy state E_0 at temperature T_0 . A random perturbation is applied altering its energy state to E_1. If this energy state is lower the transition is accepted. If it is higher, it might be accepted or not according to a probability function. The change of energy from E_0 to E_1 is performed by the *fitprop* function and the probability of acceptance of the new state is given by the Boltzmann probability distribution[15]:

$$p = \exp\left[-\Delta E / \kappa T\right] \tag{6}$$

A random number in the range [0 < r < 1] is generated by the *random* function and compared to p. If it is greater, the new state is accepted. In the scheme known as fast simulated annealing algorithm, it is employed the Cauchy-Lorentz distribution [16].

SA algorithms may be improved by including a re-annealing phase. This phase occurs when the evaluation of the objective function remains unchanged for a chosen number of iterations. As the algorithm stabilizes, i.e. converges to an optimal solution, a perturbation is applied to the solution. In order to calculate the perturbation, the standard deviation of the last candidate solution is evaluated and used as a correction factor to all elements of this solution. The new solution is denoted by η. The standard deviation of the solution is given by:

$$\alpha_i = \left\langle \left| \ x_i - \mu_i \ \right| \right\rangle \quad \text{where} \quad \mu_i = \left\langle x_i \right\rangle \tag{7}$$

where x represents the temperature vector sampled (candidate solution) for each pixel i.

In the SA of this work, the Boltzmann distribution was replaced by the Cauchy-Lorentz distribution. The SA was tested without and with re-annealing. In the last case, the perturbed solution η is then corrected using an empirical *factor* (in this work, taken as 0.4) for each pixel, as follows:

$$\eta_i = \mu_i * factor * \left(\tilde{m}_i - 1 \right) \tag{8}$$

In general, SA algorithms employ a randomly generated initial guess. The SA of this work was also tested using an initial guess obtained in a different way: sampling values from the TOD. The *stop_condition* in a SA algorithm can be given by a limit number of iterations or when the error decreases below a given threshold. In this case, a limit number of iterations was adopted. In addition, this SA was parallelized by means of calls to the Message Passing Interface (MPI) communication library in order to be executed in a distributed memory parallel machine.

3. Numerical results with simulated data

In order to generate the simulated data, a full sky map was created using the SYNFAST routine of the HEALPIX package [17]. A smaller patch with 8,000 pixels was selected for the tests and corrupted with white Gaussian noise in a way to obtain a signal-to-noise ratio of 0.1. Each simulated TOD series is composed of 20,000 temperature values for each pixel, representing an one hour data acquisition. Figure 1 shows the first 50 values of temperatures of a TOD for a given pixel with and without noise.

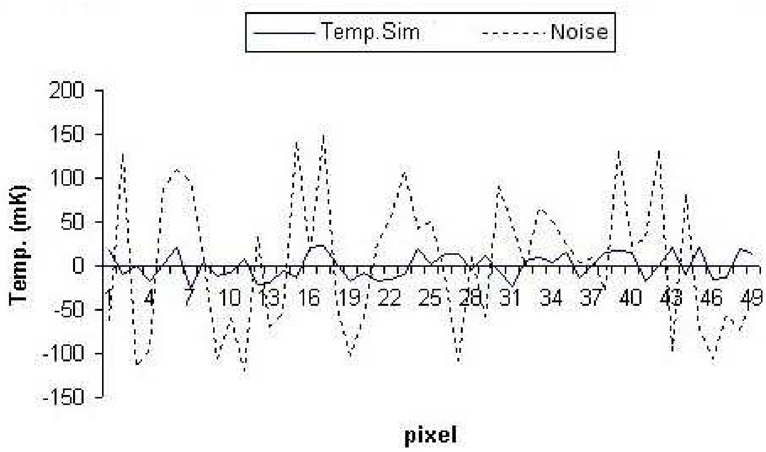

Figure 1. Simulated temperature and noise values.

Initially, the SA method was tested for map making using a random initial guess solution. Figure 2 shows the convergence of the SA for this case using a set of 30 TODs, each with 8,000 pixels and 20,000 values for each one, in order to represent a 30-day acquisition period of a given region of the sky. It can be noted that more than 100 iterations of the SA were required to reach convergence. On the other hand, Figure 3 shows the convergence when the initial guess is obtained by sampling values for each pixel from different measurements in the TOD. The number of required iterations decreased to 40. It is interesting to note that the SA is faster using this scheme to generate the initial candidate solution than in the case of using a randomly generated one.

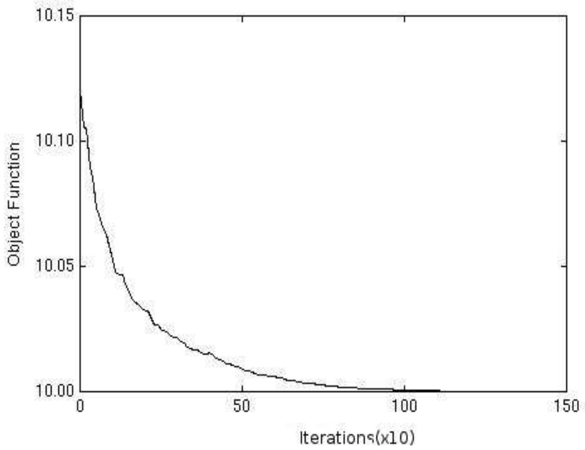

Figure 2. Convergence of the SA using a random vector as initial guess.

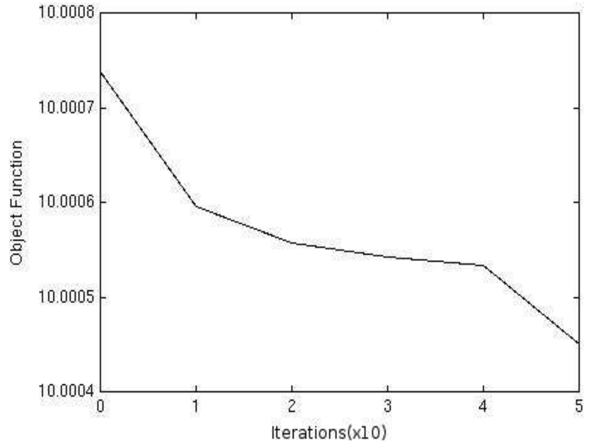

Figure 3. Convergence of the SA using sampled measurements as initial guess.

The inflexion at the 40th iteration shown in Figure 3 is due to the re-annealing phase in the SA process, that is triggered after 10 iterations with no improvement of the solution. After the perturbation of the candidate solution, the objective function start to decrease again, until the iteration limit is reached. Figure 4 shows a comparison of the standard deviations of the solution vector obtained for the two methods (the standard *bin average* method and the proposed SA method), only for the first 100 pixels. However, this comparison is similar if performed for the rest of the 8,000 pixels.

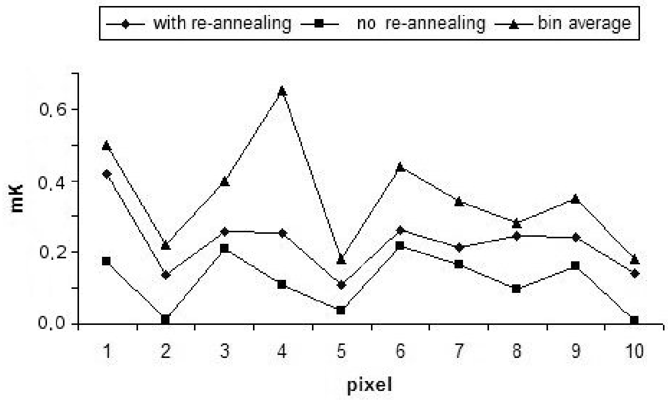

Figure 4. Temperature differences between the exact sand estimated values for the *bin average* and the proposed method employing SA with and without re-annealing.

The quality of the results was evaluated by means of the Pearson (coefficient) correlation and standard deviation from the exact solution for the solutions obtained by the *bin average* and the proposed method, as shown in Table 1.

Table 1. Standard deviation, correlation and execution time for the proposed and the *bin average* method.

Algorithm	Standard Deviation	Correlation	Execution Time (s)
SA Cauchy-Lorentz without re-annealing	0.200	0.999	201
SA Cauchy-Lorentz with re-annealing	0.150	0.999	122
Bin average	0.290	0.995	110

All tests of the proposed method were performed using a version parallel of the SA and executed in a distributed memory parallel machine, a cluster of 16 PC's with Pentium III 800 MHz processor, 256 MB of memory per node and a Fast Ethernet interconnection network.

4. Conclusions

The observations of the CMB are the main experimental evidence that give support to the standard cosmological model. Instruments are employed to measure the temperature associated with the CMB in different regions of the sky. These data are temporal series of temperature measurements contaminated with noise. The CMB maps are a graphical representation of the temperatures estimated for each pixel from the stored measurements. The CMB map making requires to solve the inverse problem of estimating the temperatures for each pixel.

Currently, there are some analytical methods to make CBM maps that are derived from the least squares method. These methods deal with matrices that cannot be easily inverted, which cause a high processing cost due to the high number of pixels and temperature measurements. A common simplification for this problem is the *bin average* method, which is based on computations of arithmetic averages of the temperatures for each pixel. The *bin average* method yields maps with good quality, but it loses some accuracy since it is based on an extensive reduction (summing up) of the temperature measurents.

This work proposes a new method for CMB map production that employs an implicit approach for its associated inverse problem. This inverse problem is formulated as an optimization problem that is solved by a stochastic algorithm, the Simulated Annealing method. One of the key benefits of this approach is the possibility to produce temperature estimations using the complete data set, which improves the map quality substantially.

Numerical results were compared with a standard *bin average* method for simulated data that corresponds to a particular TOD series for a 30-day period. Simulated data corresponds to the BEAST experiment and was corrupted with white (gaussian) noise, presenting signal-to-noise ratio of 0.1. The quality of the results was evaluated by means of the Pearson (coefficient) correlation and standard deviation from the exact solution. The *bin average* method presented correlation of 0.995 and standard deviation of 0.290.

The SA algorithm without re-annealing presented correlation of 0.999 and standard deviation of 0.200, whereas the SA algorithm with re-annealing presented correlation of 0.999 and standard deviation of 0.150. The processing cost of the SA algorithm with re-annealing is also competitive with the *bin average* by a margin in the order of 10% in terms of execution times. It shall be noted that the *bin average* method was executed sequentially, while in the proposed method the SA was parallelized.

The quality of the solution obtained with the new method is superior to the one obtained with the *bin average* at a similar processing cost. Finally, it is important to note that the quality of the solution obtained with the new approach is better than the precision of the measurement instrument. As a result, the proposed method can be employed successfully with data that will be available in a new generation of experiments.

6. References

1. Borrill, J. The challenge of data analysis for future CMBR observations - Phys. Rev. D., 1999, 59
2. Kirpatrick, S.; Gellatt, C.D. & Vecchi, M.P. Optimization by simulated annealing - Science, 1983, 220(4598): 671-680
3. Natoli, P. et al. Imaging the Cosmic Microwave Background: The BEAST Experiment - Cosmology and Particle Physics, CAPP 2000, held 17-28 july, 2000. Verbier – Switzerland. AIP Conference Preceedings, vol.555. Melville, NY: American Institute of Physics, 2001, p.324.
4. Tegmark, M., How to make maps from CMB without losing information - Ap. J., 1997, 480(L87)
5. Cantú-Paz, E. A survey of parallel genetic algorithms - IlliGAL report, 1997,. 97003.
6. Gefenstette, J.J. Parallel adpative algorithms for function optimization. Tech. Report n.C5-81-19. Vanderbilt University, EUA, 1981.
7. Ingber, L. "Very fast simulated reannealing". J.Math. Comp. Modelling, 12:967-973, 1989.
8. Pacheco, P.S. *Parallel programming with MPI*. Morgan Kaufmann Publishers, Inc. São Francisco, 1997.
9. Hwang,K.; Briggs,F.A. *Computer architecture and parallel processing*. McGraw-Hill Book Company, New York, 1984.
10. Spector,D.H. *Building Linux clusters*. O'Reilly & Associates, Sebastopol, 2000.

11. Meinhlod, P.R. et al. *A map of the Cosmic Microwave Background from the BEAST experiment*. Eprint arXiv:astro-ph/0302034 v1, 2003.
12. Dupac, X.; Giard, M. *"Map-making methods for Cosmic Microwave Background experiments".* Mon.Not.Roy.Astron.Soc, vol.330,2002,p. 497.
13. Azencott, R. *Simulated Annealing: Parallelization Techniques*. John Willey & Sons, Inc. Canada. 1992.
14. Ingber, L. *Adaptive simulated annealing(ASA): Lessons learned"*. J. of Control and Cybernetics on Simulated Annealing Applied to Combinatorial Optimization. 1995.
15. Szu, H. *"Fast simulated annealing"*. In J.S. Dendekes, AIP Conf. Proc. 151, Neural Networks for Computing, Snowbird, Utah, AIP, 1986.
16. Gorski, E. K.; Hivon, E., & Wandelt, B. 1998, in Proceedings of the MPA/ESO Garching Conference, ed. A. J. Banday, K. Sheth, & L. Da Costa (Garching:ESO),inpress,http://www.eso.org/science/healpix