AGU PUBLICATIONS

JGR

Journal of Geophysical Research: Space Physics

RESEARCH ARTICLE

10.1002/2015JA022064

Key Points:

- A new layer method is introduced to solve the radiation belt diffusion equation with cross terms
- The new layer method is positivity preserving and efficient
- Bounce resonance with magnetosonic waves is as important as gyroresonance

Correspondence to:
X. Tao,
xtao@ustc.edu.cn

Citation:
Tao, X., L. Zhang, C. Wang, X. Li, J. M. Albert, and A. A. Chan (2016), An efficient and positivity-preserving layer method for modeling radiation belt diffusion processes, *J. Geophys. Res. Space Physics*, *121*, 305–320, doi:10.1002/2015JA022064.

Received 21 OCT 2015
Accepted 24 DEC 2015
Accepted article online 7 JAN 2016
Published online 25 JAN 2016

An efficient and positivity-preserving layer method for modeling radiation belt diffusion processes

X. Tao[1,2], L. Zhang[1,3], C. Wang[1], X. Li[1], J. M. Albert[4], and A. A. Chan[5]

[1] CAS Key Laboratory of Geospace Environment, Department of Geophysics and Planetary Sciences, University of Science Technology of China, Hefei, China, [2] Collaborative Innovation Center of Astronautical Science and Technology, China, [3] Institute of Space Physics and Applied Technology, Peking University, Beijing, China, [4] Air Force Research Laboratory, Space Vehicles Directorate, Kirtland Air Force Base, Albuquerque, New Mexico, USA, [5] Department of Physics and Astronomy, Rice University, Houston, Texas, USA

Abstract An efficient and positivity-preserving layer method is introduced to solve the radiation belt diffusion equation and is applied to study the bounce resonance interaction between relativistic electrons and magnetosonic waves. The layer method with linear interpolation, denoted by LM-L (layer method-linear), requires the use of a large number of grid points to ensure accurate solutions. We introduce a monotonicity- and positivity-preserving cubic interpolation method to be used with the Milstein-Tretyakov layer method. The resulting method, called LM-MC (layer method-monotone cubic), can be used to solve the radiation belt diffusion equation with a much smaller number of grid points than LM-L while still being able to preserve the positivity of the solution. We suggest that LM-MC can be used to study long-term dynamics of radiation belts. We then develop a 2-D LM-MC code and use it to investigate the bounce resonance diffusion of radiation belt electrons by magnetosonic waves. Using a previously published magnetosonic wave model, we demonstrate that bounce resonance with magnetosonic waves is as important as gyroresonance; both can cause several orders of magnitude increase of MeV electron fluxes within 1 day. We conclude that bounce resonance with magnetosonic waves should be taken into consideration together with gyroresonance.

1. Introduction

One widely adopted method to model the global dynamics of radiation belts is by solving a quasilinear diffusion equation. Various numerical codes [*Beutier and Boscher*, 1995; *Varotsou et al.*, 2005; *Li et al.*, 2007; *Tao et al.*, 2008, 2009, 2011a, 2011b, *Shprits et al.*, 2008a, 2008b; *Albert et al.*, 2009; *Xiao et al.*, 2009, 2015; *Su et al.*, 2010; *Subbotin et al.*, 2010; *Tu et al.*, 2013; *Zheng et al.*, 2014; *Glauert et al.*, 2014] based on quasilinear diffusion theory have been developed to evaluate and to understand the dynamic changes of particle fluxes in the radiation belts. *Albert* [2009, 2013] demonstrated that the numerically obtained phase space density might become negative when solving multidimensional quasilinear diffusion equations using standard finite difference methods, due to nonzero cross-diffusion terms. Different numerical techniques have been developed to avoid this problem when solving multidimensional diffusion equations [*Albert and Young*, 2005; *Tao et al.*, 2008, 2009].

The search for a positivity-preserving method for solving the radiation belt diffusion equation starts from *Albert and Young* [2005]. This diffusion equation written in (J_1, J_2, J_3) coordinates has the form

$$\frac{\partial f}{\partial t} = \sum_{i=1}^{3} \frac{\partial}{\partial J_i}\left(D_{ij}\frac{\partial f}{\partial J_j}\right), \tag{1}$$

where D_{ij} is diffusion coefficient, and J_1, J_2, and J_3 are three canonical action variables associated with cyclotron motion, bounce motion, and drift motion of radiation belt particles, respectively [*Schulz and Lanzerotti*, 1974, pp 47]. The negative solutions of f are caused by the presence of cross-diffusion terms ($D_{ij}, i \neq j$) [*Albert*, 2009, 2013]; therefore, *Albert and Young* [2005] made a coordinate transformation to eliminate the cross-diffusion terms and solved the transformed diffusion equation using finite difference methods.

The efficiency of the resulting algorithm is comparable to standard finite difference methods. The main disadvantage of the method, however, is that the success of the method depends on whether the coordinate transformation can be made. Because of this, the method has not been extended to the full 3-D case yet.

A different approach to the problem was taken by *Tao et al.* [2008, 2009]. *Tao et al.* [2008] developed a stochastic differential equation (SDE) code to solve the 2-D bounce-averaged pitch angle and energy diffusion equation. The SDE code is particle based and is very efficient when solutions on a small number of points are needed. Recently, this SDE code has been extended to the 3-D case [*Zheng et al.*, 2014]. However, if solutions are needed on a large computational domain for long times, the SDE code becomes less efficient, because of the need to trace a large number of stochastic trajectories. *Tao et al.* [2009] aimed to address the efficiency problem of the SDE method and introduced the so-called layer method developed by *Milstein and Tretyakov* [2001, 2002]. Although the layer method is based on the SDE theory, it is nevertheless deterministic. One key element of the layer method is the need to use interpolation. The layer method of *Milstein and Tretyakov* [2001, 2002] and *Tao et al.* [2009] used linear interpolation. However, linear interpolation has a low order of accuracy. Correspondingly, the *Tao et al.* [2009] layer method code is not efficient due to the use of a large number of grid points (1400 × 1500 for a 2-D problem). *Tao et al.* [2009] also tried the cubic interpolation method, as suggested by *Milstein and Tretyakov* [2001]; however, they found that the cubic interpolation method can lead to negative solutions of the phase-space density.

Because of the flexibility of the Milstein-Tretyakov layer method framework, it is possible to employ other interpolation methods. In this paper, we introduce the use of a monotonicity-preserving (MP) and positivity-preserving (PP) cubic interpolation method in the layer method; the resulting numerical code does not need to use a large number of grid points to obtain accurate and positive results. Therefore, it is much more efficient than the *Tao et al.* [2009] method. In the text below, we will call this layer method LM-MC (layer method-monotone cubic) to differentiate it from the layer method used by *Tao et al.* [2009] with linear interpolation, which will be called LM-L (layer method-linear).

The remainder of this paper is organized as follows. We introduce the LM-MC using a simple 1-D problem in section 2. We review the layer method in section 2.1 and introduce the MP and PP cubic interpolation method in section 2.2. Then we develop a 2-D LM-MC code to solve the bounce-averaged pitch angle and energy diffusion equation. We validate this 2-D LM-MC code in section 3.1 and then explore effects of bounce resonance diffusion with magnetosonic waves on radiation belt electrons in section 3.2. Our results are then discussed and summarized in section 4.

2. The Layer Method With Monotone Cubic Interpolation (LM-MC)

In this section, we first review the basic framework of Milstein-Tretyakov layer method in 1-D. Details about the derivation of the layer method and the extension of the layer method to higher dimensions can be found in *Milstein and Tretyakov* [2001, 2002]. Then we introduce the monotonicity-preserving and positivity-preserving cubic interpolation method in 1-D and demonstrate its use in the Milstein-Tretyakov layer method. This interpolation method can be easily extended to higher dimensions by performing interpolations dimension by dimension. For example, for a 2-D interpolation in x, y coordinates, the interpolation is first done along x dimension and then along y dimension.

2.1. A Milstein-Tretyakov Layer Method in 1-D

Suppose we want to solve a general 1-D parabolic equation

$$\frac{\partial f}{\partial t} = b(t,x)\frac{\partial f}{\partial x} + \frac{1}{2}a(t,x)\frac{\partial^2 f}{\partial x^2}, \quad x \in [\alpha, \beta], \tag{2}$$

subject to the following general initial and boundary conditions

$$f(t = t_0) = g(x), \tag{3}$$

$$f(x = \alpha) = h(t), \tag{4}$$

$$\frac{\partial f}{\partial x}(x = \beta) = \varphi(t). \tag{5}$$

Here we assume a Dirichlet boundary condition at $x = \alpha$ and a Neumann boundary condition at $x = \beta$ to illustrate how to handle these two kinds of popular boundary conditions in the layer method below.

Like finite difference methods, we discretize t equidistantly into $t_0, t_1, t_2, \cdots, t_n, t_{n+1}, \cdots$ with $t_i = t_0 + i\Delta t$, and x equidistantly into $x_0, x_1, x_2, \cdots, x_{N-1}$ with $x_0 = \alpha$, $x_{N-1} = \beta$, and $x_j = x_0 + j\Delta x$. The time step and the grid size are chosen to be constant here only for simplicity of discussion. For any point x_j with $j = 1, 2, 3, \cdots, N-2$, the Milstein-Tretyakov layer method solution of $f(t_{n+1}, x_j)$ is closely related to the following two points:

$$\tilde{x}_+ = x_j + b(t_{n+1}, x_j)\Delta t + \sigma(t_{n+1}, x_j)\sqrt{\Delta t}, \tag{6}$$

$$\tilde{x}_- = x_j + b(t_{n+1}, x_j)\Delta t - \sigma(t_{n+1}, x_j)\sqrt{\Delta t}, \tag{7}$$

where $\sigma^2 = a$. We can choose either $\sigma = \sqrt{a}$ or $-\sqrt{a}$; different choices of σ do not affect the solution of f. For simplicity of discussion, we choose $\sigma > 0$ and thus $\tilde{x}_- < \tilde{x}_+$. In the simplest case, if $\alpha < \tilde{x}_\pm < \beta$, the layer method solution of f is

$$f(t_{n+1}, x_j) \approx \frac{1}{2}\left[f(t_n, \tilde{x}_-) + f(t_n, \tilde{x}_+)\right]. \tag{8}$$

We can see from equations (6)–(8) that the layer method for the parabolic equation (2) is similar to the method of characteristics for an advection equation. Indeed, if we let $a = 0$, then equation (2) becomes a pure advection equation, and equations (7) and (8) just reduce to the method of characteristics with the Euler method. If $\alpha < \tilde{x}_\pm < \beta$, equation (8) immediately gives $f(t_{n+1}, x_j)$, because $f(t_n, \tilde{x}_\pm)$ can be obtained using interpolation. Clearly, if a small enough time step is used, $\alpha < \tilde{x}_\pm < \beta$ for all x_j with $j = 1, 2, 3, \cdots, N-2$. However, if a larger time step is used and $\tilde{x}$ is outside the computational domain, the value of $f(t_n, \tilde{x})$ in equation (8) is unknown, and we have to find the solution in a slightly different way, depending on the corresponding boundary conditions.

For the Neumann boundary condition at $x = \beta$, if $\tilde{x} > \beta$, we note that using the central finite difference formula, the derivative $\partial f/\partial x$ can be approximated as

$$\frac{\partial f}{\partial x}(t_n, x = \beta) = \varphi(t_n) \approx \frac{f(t_n, \tilde{x}) - f(t_n, \tilde{x}')}{\tilde{x} - \tilde{x}'}, \tag{9}$$

where $\tilde{x}'$ is the symmetric point of $\tilde{x}$ with respect to $x = \beta$; i.e.,

$$\tilde{x}' = 2\beta - \tilde{x}. \tag{10}$$

Equation (9) can be solved to give $f(t_n, \tilde{x})$ as

$$f(t_n, \tilde{x}) = f(t_n, \tilde{x}') + \varphi(t_n)(\tilde{x} - \tilde{x}'). \tag{11}$$

If Δt is small enough that $\tilde{x}'$ is within $[\alpha, \beta)$, the value of $f(t_n, \tilde{x})$ can be obtained from interpolation, and equation (8) can again be used to find $f(t_{n+1}, x_j)$. Note that one of the most common Neumann boundary condition in the radiation belt modeling is of the type $\partial f/\partial x = 0$, in this case $f(t_n, \tilde{x}) = f(t_n, \tilde{x}')$ from equation (11).

For the Dirichlet boundary condition at α, if $\tilde{x}_- < \alpha$ and $\tilde{x}_+ \in [\alpha, \beta]$, *Milstein and Tretyakov* [2001] demonstrated that one method of solving $f(t_{n+1}, x_j)$ is

$$f(t_{n+1}, x_j) \approx \xi_- h(t_{n+1-\lambda}) + \xi_+ f(t_n, \tilde{x}_+), \tag{12}$$

where

$$\xi_- = \frac{1}{1 + \sqrt{\lambda}}, \tag{13}$$

$$\xi_+ = \frac{\sqrt{\lambda}}{1 + \sqrt{\lambda}}, \tag{14}$$

$$t_{n+1-\lambda} = t_{n+1} - \lambda\Delta t. \tag{15}$$

Here $0 < \lambda < 1$, and it is the root of

$$\alpha = x_j + b(t_{n+1}, x_j)\lambda\Delta t - \sigma(t_{n+1}, x_j)\sqrt{\lambda\Delta t}. \tag{16}$$

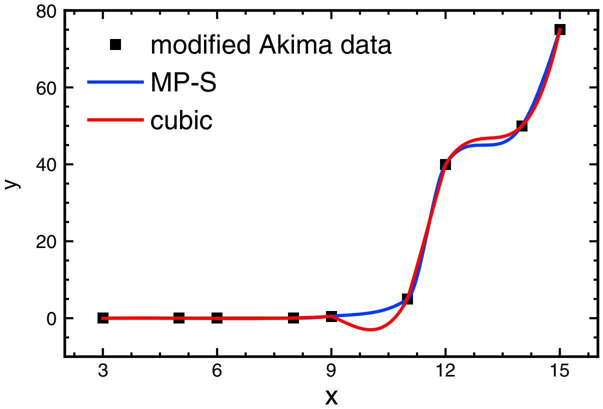

Figure 1. Comparison of the cubic Hermite interpolation method with the MP-S limiter (blue) and the cubic interpolation method (red) using the modified Akima data set (black squares) given in Table 1. The MP-S limiter is defined in section 2.2.1.

In equation (12), ξ_- and ξ_+ are two weights and may be regarded as the probabilities of reaching α and $\tilde{x}_+$, respectively. Note that $\xi_\pm = 1/2$ only in the case $\lambda = 1$. For other ways of handling Dirichlet boundary conditions and more details about the layer method, we refer readers to the *Milstein and Tretyakov* [2001, 2002].

In above equations, the values of $f(t_n, \tilde{x}_-)$ and $f(t_n, \tilde{x}_+)$ are obtained from interpolation. It is clear from equation (8) or (12) that if the interpolation method can guarantee the positivity of $f(t_n, \tilde{x}_\pm)$, the whole layer method is strictly positivity preserving. The layer methods of *Milstein and Tretyakov* [2001, 2002] and *Milstein* [2002] were constructed using linear interpolation, which is positivity preserving and was used by *Tao et al.* [2009].

However, *Tao et al.* [2009] demonstrated that to obtain accurate results, the layer method with linear interpolation requires the use of a large number of grid points. The resulting LM-L code is far less efficient than finite difference methods. *Milstein and Tretyakov* [2002, 2001] suggested the use of cubic interpolation method; however, the cubic interpolation method might give oscillatory interpolant and result in negative solutions of f, as having been reported by *Tao et al.* [2009]. This behavior of cubic interpolation is also demonstrated in Figure 1. We use a monotone data set (Table 1) modified from the Akima data [*Akima*, 1970]. The cubic interpolation leads to a nonmonotone and negative interpolant between $x = 9$ and 11. Therefore, a better interpolation method to be used with the layer method should have a higher accuracy than linear interpolation and be monotonicity preserving and positivity preserving.

2.2. The Monotonicity- and Positivity-Preserving Cubic Interpolation Method

In this section, we give a brief review of *Fritsch and Carlson* [1980], *Carlson and Fritsch* [1985], and *Huynh* [1993] about monotonicity-preserving (MP) or positivity-preserving (PP) cubic interpolation methods. The MP or PP cubic interpolation methods are based on the cubic Hermite interpolations. Let the mesh $\{x_i\}_{i=0}^{N-1}$ be a partition of the interval $[x_0, x_{N-1}]$ with $x_0 < x_1 < \cdots < x_{N-2} < x_{N-1}$. The corresponding data are $\{f_i\}$ with $f_i = f(x_i)$, where f is a piecewise smooth function. The local derivative of f at x_i, $\dot{f}_i$ can be approximated numerically using $\{x_i\}$ and $\{f_i\}$. The cubic Hermite interpolant is defined for $x_0 < x < x_{N-1}$ in terms of $f_i, f_{i+1}, \dot{f}_i, \dot{f}_{i+1}$ by

$$q(x) = c_3(x - x_i)^3 + c_2(x - x_i)^2 + c_1(x - x_i) + c_0, \tag{17}$$

where for $x_i \le x \le x_{i+1}$,

$$c_0 = f_i, \quad c_1 = \dot{f}_i, \tag{18}$$

$$c_2 = \left(3s_{i+1/2} - 2\dot{f}_i - \dot{f}_{i+1}\right)/\Delta x_{i+1/2}, \tag{19}$$

$$c_3 = \left(\dot{f}_i + \dot{f}_{i+1} - 2s_{i+1/2}\right)/(\Delta x_{i+1/2})^2. \tag{20}$$

Table 1. Modified Akima Data

x	3	5	6	8	9	11	12	14	15
y	0.001	0.001	0.001	0.001	0.5	5	40	50	75

Here $\Delta x_{i+1/2} = x_{i+1} - x_i$, and $s_{i+1/2} = \Delta f_{i+1/2}/\Delta x_{i+1/2}$, with $\Delta f_{i+1/2} = f_{i+1} - f_i$, is the linear slope between x_i and x_{i+1}. It is straightforward to verify that $q(x_i) = f_i, q'(x_i) = \dot{f}_i, q(x_{i+1}) = f_{i+1}$, and $q'(x_{i+1}) = \dot{f}_{i+1}$. Note that the derivative $\{\dot{f}_i\}$ is evaluated numerically, e.g., one can use the parabolic formula given below in equation (30). The key idea of MP and PP interpolation methods is then to find in a constructive way how to limit the numerically calculated $\dot{f}_i$ to a certain interval so that the resulting interpolant has the desired properties, such as being monotone or positive. The corresponding ways of modifying $\dot{f}_i$ are called MP limiters or PP limiters. The use of the limited $\dot{f}_i$ in the cubic Hermite interpolant $q(x)$ in equation (17) results in a MP or PP cubic interpolation method.

2.2.1. A Simple MP Limiter

We first review a simple MP limiter from *Fritsch and Carlson* [1980] and *Huynh* [1993]. The purpose of a MP limiter is to limit the numerically calculated $\dot{f}_i$ so that the interpolant is monotone if the data are monotone. The data are monotone at x_i if $(f_i - f_{i-1})(f_{i+1} - f_i) \geq 0$. An interpolant is monotone in $[x_i, x_{i+1}]$ if it is monotone for any x between x_i and x_{i+1}. *Fritsch and Carlson* [1980] and *Huynh* [1993] proved theoretically that if

$$\dot{f}_i, \dot{f}_{i+1} \in I[0, 3s_{i+1/2}], \tag{21}$$

then the resulting interpolant $q(x)$ given by equation (17) is monotone in $[x_i, x_{i+1}]$. In this paper, we adopt the symbols used by *Huynh* [1993], because we find that they are easy to use when developing numerical codes. In equation (21),

$$I[z_0, z_1, \cdots, z_k] = [\min(z_0, z_1, \cdots, z_k), \max(z_0, z_1, \cdots, z_k)] \tag{22}$$

is the interval between minimum and maximum of $z_0, z_1, \cdots, z_k$. To obtain continuous $\dot{f}_i$ for $0 < i < N-1$, one replaces i by $i-1$ in equation (21), which implies

$$\dot{f}_i \in I[0, 3s_{i+1/2}] \text{ and } \dot{f}_i \in I[0, 3s_{i-1/2}]. \tag{23}$$

These two conditions of $\dot{f}_i$ can be combined to give

$$\dot{f}_i \in I[0, 3s_i], \tag{24}$$

with

$$s_i = \text{minmod}(s_{i-1/2}, s_{i+1/2}). \tag{25}$$

Here the minmod function is defined as

$$\text{minmod}(x, y) = \text{median}(x, y, 0), \tag{26}$$

where the median of three numbers is the one between the other two. To preserve monotonicity of data, we only need to bring the numerically estimated $\dot{f}_i$ into the range defined by equation (24) or using the minmod function defined in equation (26),

$$\dot{f}_i \leftarrow \text{minmod}(\dot{f}_i, 3s_i). \tag{27}$$

We call this MP limiter the MP-S (S for Simple) limiter to differentiate it from a more advanced MP limiter discussed below. Equation (27) means that if the numerically evaluated trial $\dot{f}_i \in I[0, 3s_i]$, then we do not modify it; otherwise, $\dot{f}_i = 0$ or $3s_i$ depending on the sign of $\dot{f}_i$ and s_i. For example, if the data are monotone and increasing, then $s_i > 0$. By equation (27), if the numerically evaluated trial $\dot{f}_i < 0$, then we let $\dot{f}_i = 0$; if the trial $\dot{f}_i > 3s_i$, then we let $\dot{f}_i = 3s_i$; otherwise, we do not modify $\dot{f}_i$ and directly use it in the cubic Hermite interpolant equation (17).

The effect of the monotone cubic interpolation with the MP-S limiter is demonstrated in Figure 1 using the modified Akima data shown in Table 1. The trial $\dot{f}_i$ is estimated using the derivative at x_i of a parabola $p_i(x)$ determined using three points f_{i-1}, f_i, and f_{i+1}; i.e.,

$$p_i(x) = f_i + s_{i+1/2}(x - x_i) + d_i(x - x_i)(x - x_{i+1}), \tag{28}$$

with

$$d_i = \frac{s_{i+1/2} - s_{i-1/2}}{x_{i+1} - x_{i-1}}. \tag{29}$$

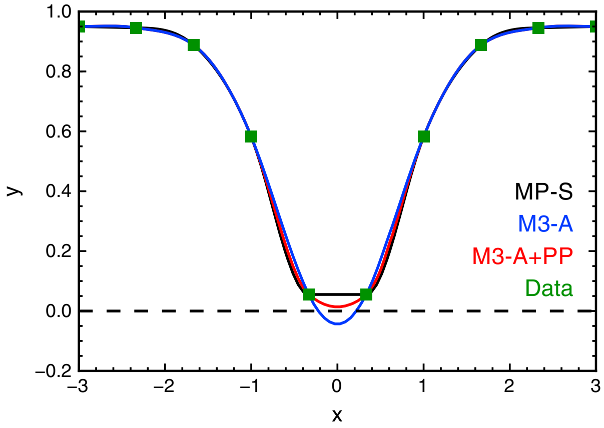

Figure 2. Comparison of the cubic Hermite interpolation with three different limiters: the MP-S limiter (black), the M3-A limiter (blue), and the combined M3-A and PP limiters (red). The data set (green squares) is given in Table 2.

The derivative of $p_i(x)$ at x_i is then simply

$$p_i'(x_i) = \frac{s_{i+1/2}\Delta x_{i-1/2} + s_{i-1/2}\Delta x_{i+1/2}}{\Delta x_{i-1/2} + \Delta x_{i+1/2}}, \tag{30}$$

the weighted average of $s_{i+1/2}$ and $s_{i-1/2}$. It is clear from equation (30) that $p_i'(x_i) \in I[s_{i-1/2}, s_{i+1/2}]$. Note that $p_i'(x_i)$ is a second-order approximation to the exact derivative $f'(x_i)$. In case of uniform grid, $\Delta x_{i-1/2} = \Delta x_{i+1/2}$, equation (30) reduces to the familiar second-order central difference approximation of $f'(x_i)$,

$$p_i'(x_i) = \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}}. \tag{31}$$

After letting $\dot{f}_i = p_i'(x_i)$, we apply the MP-S limiter to $\dot{f}_i$ and use the limited $\dot{f}_i$ in the cubic Hermite interpolant in equation (17); the curve is shown in blue in Figure 1. The monotone cubic interpolation method clearly guarantees that the interpolant is monotone.

2.2.2. The M3 Limiter

While always guaranteeing that the interpolant is monotone, the MP-S limiter described above might clip the interpolant if the data are not monotone, leading to loss of accuracy. For example, if x_i is a local extremum, $s_{i-1/2}s_{i+1/2} < 0$; therefore, $s_i = 0$ from equation (25). The MP-S limiter given by equation (27) will force $\dot{f}_i = 0$. This is demonstrated in Figure 2, using the data set given in Table 2. The MP-S limiter forces $\dot{f}_i = 0$ at $x = \pm 0.33$. The accuracy of the interpolation is reduced and equals that of the linear interpolation between the two points.

A way to increase accuracy is to relax the MP condition (21) near local extrema so that if the data are not monotone, the interpolant is not necessarily monotone. *Huynh* [1993] developed a class of methods, called M3 (monotone third order) methods. To introduce the M3 limiter, we first define $p_{i+1/2}(x)$ by

$$p_{i+1/2}(x) = f_i + s_{i+1/2}(x - x_i) + d_{i+1/2}(x - x_i)(x - x_{i+1}), \tag{32}$$

where $d_{i+1/2} = \text{minmod}(d_i, d_{i+1})$. Then the derivative of $p_{i+1/2}(x)$ at x_i is given by

$$p_{i+1/2}'(x_i) = s_{i+1/2} + d_{i+1/2}(x_i - x_{i+1}). \tag{33}$$

In other words,

$$p_{i+1/2}'(x_i) = \text{median}[s_{i+1/2}, p_i'(x_i), p_{i+1}'(x_i)], \tag{34}$$

Table 2. A Data Set With Local Minima

x	−3.00	−2.33	−1.67	−1.00	−0.33	0.33	1.00	1.67	2.33	3.00
y	0.9499	0.9457	0.8878	0.5821	0.05516	0.05516	0.5821	0.8878	0.9457	0.9499

where $p'_{i+1}(x_i) = s_{i+1/2} + d_{i+1}(x_i - x_{i+1})$. Similarly, we can define $p_{i-1/2}(x)$ through replacing i by $i-1$ in equation (32); the derivative of $p_{i-1/2}(x)$ at x_i is

$$p'_{i-1/2}(x_i) = s_{i-1/2} + d_{i-1/2}(x_i - x_{i-1}), \tag{35}$$

where $d_{i-1/2} = \text{minmod}(d_i, d_{i-1})$. The minmod of $p'_{i-1/2}(x_i)$ and $p'_{i+1/2}(x_i)$ gives t_i; i.e.,

$$t_i = \text{minmod}[p'_{i-1/2}(x_i), p'_{i+1/2}(x_i)]. \tag{36}$$

It can be seen from the above description that the definition of t_i requires five grid points from $i-2$ to $i+2$. Therefore, t_i is not defined for the four points near two boundaries ($i = 0, 1$ and $i = N-2, N-1$). The M3 interval [*Huynh*, 1993] is then defined as

$$\dot{f}_i \in I[0, t_{i,\max}] \tag{37}$$

where

$$t_{i,\max} = \text{sgn}(t_i)\max\left(3|s_i|, \frac{3}{2}|t_i|\right), \tag{38}$$

with $\text{sgn}(t_i)$ the sign of t_i. To enforce that $\dot{f}_i$ belongs to the M3 interval, equation (37), we use

$$\dot{f}_i \leftarrow \text{minmod}(\dot{f}_i, t_{i,\max}). \tag{39}$$

This MP limiter is called the M3 limiter. For details about how the M3 limiter extends the MP S limiter, we refer readers to *Huynh* [1993].

For points near boundaries, we cannot use the M3 limiter. For $i = 1$ or $i = N-2$, we use the MP-S limiter, equation (27), to ensure monotonicity, because the MP-S limiter only requires three points centered at x_i. The monotonicity of the interpolant between $[x_0, x_1]$ and $[x_{N-2}, x_{N-1}]$ is guaranteed by using $\dot{f}_0 \leftarrow$ minmod $(\dot{f}_0, 3s_{1/2})$ and $\dot{f}_{N-1} \leftarrow \text{minmod}(\dot{f}_{N-1}, 3s_{N-3/2})$, respectively, from equation (21).

Because of the relaxation of the monotonicity condition near local extrema, however, the M3 interval in equation (37) does not necessarily guarantee the positivity of f. If x_i is a local minimum, it is possible for the M3 limiter to generate negative values of f. This is illustrated in Figure 2, where the M3 limiter relaxed the monotonicity condition at $x = \pm 0.33$ and results in negative fs between $x = -0.33$ and 0.33. In Figure 2, M3-A refers to the M3 limiter with the trial $\dot{f}_i$ evaluated by equation (43) below, with A for average. To guarantee the positivity of f, we use a positivity-preserving limiter, described by *Carlson and Fritsch* [1985], together with the M3 limiter.

2.2.3. The Combination of MP and PP Limiters

According to *Carlson and Fritsch* [1985], the interpolant will not change sign over $[x_i, x_{i+1}]$, assuming that f_is are positive, if derivatives $\dot{f}_i$ and $\dot{f}_{i+1}$ satisfy

$$\dot{f}_i \geq -\frac{3f_i}{x_{i+1} - x_i} \text{ and } \dot{f}_{i+1} \leq \frac{3f_{i+1}}{x_{i+1} - x_i}. \tag{40}$$

By replacing i by $i+1$ in the above equation, we have

$$-\frac{3f_i}{\Delta x_{i+1/2}} \leq \dot{f}_i \leq \frac{3f_i}{\Delta x_{i-1/2}}. \tag{41}$$

If this condition of $\dot{f}_i$ is satisfied, then equation (17) can be guaranteed to give positive fs. One can prove that if the data are monotone or near local maximum, then if $\dot{f}_i$ is within the M3 interval, it automatically satisfies the PP condition given by equation (41). This proof is given in Appendix A. Using equation (41), a PP limiter can be conveniently defined as

$$\dot{f}_i \leftarrow \text{median}\left(\dot{f}_i, -\frac{3f_i}{\Delta x_{i+1/2}}, \frac{3f_i}{\Delta x_{i-1/2}}\right). \tag{42}$$

Now we combine the M3 limiter and the PP limiter to achieve a MP and PP cubic interpolation method. This involves four steps. First, we estimate a trial value for $\dot{f}_i$. This can be done using

$$\dot{f}_i = \frac{p'_{i-1/2}(x_i) + p'_{i+1/2}(x_i)}{2}, \tag{43}$$

for interior points [*Huynh*, 1993]. For $i = 1$ and $i = N - 2$, we use $\dot{f}_i = p_i'(x_i)$. For $i = 0$ and $i = N - 1$, we use one-sided approximations for $\dot{f}_i$ according to *Huynh* [1993]; i.e.,

$$\dot{f}_0 = p_1'(x_0) = s_{1/2} + d_1(x_0 - x_1), \tag{44}$$

$$\dot{f}_{N-1} = p_{N-2}'(x_{N-1}) = s_{N-3/2} + d_{N-2}(x_{N-1} - x_{N-2}). \tag{45}$$

The resulting algorithm after applying the M3-limiter will called M3-A if equation (43) for interior points is used. One can also use higher-order approximations of $\dot{f}_i$ for interior points. For example, a fourth-order accurate quartic formula for $\dot{f}_i$ is given in equation (4.21) of *Huynh* [1993], resulting in an algorithm called M3-Q, Q for quartic, after applying the M3-limiter. Second, we apply the M3 limiter, equation (39), to $\dot{f}_i$ for interior points and handling $\dot{f}_i$ for points near boundaries as described in the previous section. This will make necessary modifications of the trial $\dot{f}_i$ so that the resulting interpolant is monotonicity preserving. Third, the PP limiter is applied to $\dot{f}_i$, so that the resulting interpolant is both monotonicity preserving and positivity preserving. Note that because of the results in Appendix A the PP limiter will only modify $\dot{f}_i$ at local minima if necessary. Fourth, using the $\dot{f}_i$ from the previous step in the cubic Hermite function, equation (17), results in the desired MP and PP cubic interpolant. The combined use of the M3 limiter and the PP limiter is demonstrated in Figure 2, which leads to a nonoscillatory and positive interpolant throughout the whole data range.

The order of the accuracy of the interpolation depends on the exact algorithm used. The highest order of accuracy of the interpolant $q(x)$ in equation (17) is fourth order, and it requires that the derivatives $\{\dot{f}_i\}$ are exact or third order. Generally, the order of accuracy of $q(x)$ is one order higher than the order of accuracy of derivatives. For example, the M3-A algorithm is uniformly third order accurate, because $\dot{f}_i$ given in equation (43) is a second-order approximation of $f'(x_i)$. The M3-Q algorithm is fourth-order accurate if the data are monotone and third-order accurate near local extrema [*Huynh*, 1993]. The order of accuracy of M3-A and M3-Q methods is demonstrated in Appendix B.

In the remainder of the paper, we always apply the PP limiter together with the M3-A limiter to guarantee the positivity of the phase space density in LM-MC. We use the M3-A interpolation method in this paper because it is relatively simple compared with M3-Q and our test results suggest that it is accurate enough for our purpose. The long-term accuracy of LM-MC is compared with that of LM-L used by *Tao et al.* [2009] in Appendix C for a 1-D pitch angle diffusion equation. This comparison demonstrates that the LM-MC can solve the diffusion equation accurately with a relatively small number of grid points, and therefore can be used for long-term modeling of radiation belt dynamics.

3. Application to Bounce Resonance Diffusion by Magnetosonic Waves

In this section, we first validate a 2-D LM-MC code developed to study pitch angle and momentum diffusion. Then we use the 2-D code to investigate the bounce resonance diffusion of relativistic electrons by magnetosonic waves.

3.1. Validation of a 2-D LM-MC Code

The 2-D pitch angle and momentum diffusion equation is

$$\frac{\partial f}{\partial t} = \frac{1}{Gp}\frac{\partial}{\partial \alpha_0} G\left(D_{\alpha_0\alpha_0}\frac{1}{p}\frac{\partial f}{\partial \alpha_0} + D_{\alpha_0 p}\frac{\partial f}{\partial p}\right) + \frac{1}{G}\frac{\partial}{\partial p} G\left(D_{\alpha_0 p}\frac{1}{p}\frac{\partial f}{\partial \alpha_0} + D_{pp}\frac{\partial f}{\partial p}\right), \tag{46}$$

where α_0 is the equatorial pitch angle, p is the momentum, and $G = p^2 T(\alpha_0)\sin(2\alpha_0)$ with $T \equiv 1.30-0.56\sin\alpha_0$ the normalized bounce period. Here $D_{\alpha_0\alpha_0}, D_{\alpha_0 p}$, and D_{pp} are bounce-averaged pitch angle, mixed, and momentum diffusion coefficients [*Albert*, 2004]. The diffusion coefficients used in this section are calculated using a chorus wave model at $L = 4.5$ [*Horne et al.*, 2005]. Initial and boundary conditions (E in MeV) are

$$f(t=0) = \exp[-(E-0.2)/0.1](\sin\alpha_0 - \sin\alpha_{0L})/p^2, \tag{47}$$

$$f|_{\alpha_0=\alpha_{0L}} = 0, \tag{48}$$

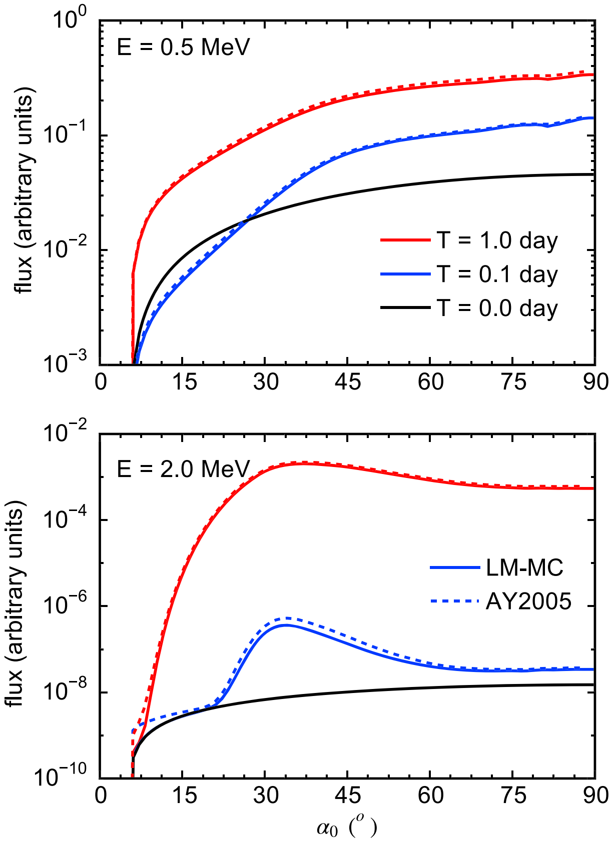

Figure 3. Comparison between the solutions of a 2-D LM-MC code (solid lines) and the *Albert and Young* [2005] method (dashed lines) for (top) $E = 0.5$ MeV and (bottom) 2.0 MeV at $T = 0.1$ day (blue) and $T = 1$ day (red). Black lines indicate initial fluxes.

$$\left.\frac{\partial f}{\partial \alpha_0}\right|_{\alpha_0=90^\circ} = 0, \tag{49}$$

$$f|_{E=E_{\max}} = 0, \tag{50}$$

$$f|_{E=E_{\min}} = \exp[-(E_{\min} - 0.2)/0.1](\sin\alpha_0 - \sin\alpha_{0L})/p^2_{\min}, \tag{51}$$

where the loss cone pitch angle $\alpha_{0L} = 5^\circ$, $E_{\min} = 0.2$ MeV, $E_{\max} = 5$ MeV, and $p_{\min}$ is the momentum corresponding to $E_{\min}$.

We validate the 2-D LM-MC code by comparing the LM-MC results with the *Albert and Young* [2005] solution. The LM-MC solution uses 80 equidistantly spaced grid points in α_0 and 80 equidistantly spaced grid points in log(E). The comparison between LM-MC results and *Albert and Young* [2005] method solution for $E = 0.5$ MeV and 2 MeV at $T = 0.1$ day and 1day is shown in Figure 3. The mean difference between the two results is measured by

$$\varepsilon = \text{mean}(|\log(j_{\text{AY}}) - \log(j_{\text{LM-MC}})|), \tag{52}$$

where $j = p^2 f$ is flux and subscripts denote the corresponding method. We find that $\varepsilon \approx 0.06$ for $E = 0.5$ MeV at $T = 1.0$ day and $\varepsilon \approx 0.14$ for $E = 2.0$ MeV at $T = 1.0$ day. This comparison demonstrates that the results from the two different methods agree very well with each other, especially considering the slight difference in boundary conditions [*Albert and Young*, 2005]. We conclude that our 2-D LM-MC code is capable of solving the bounce-averaged pitch angle and momentum diffusion equation with cross diffusion.

We now briefly comment on the efficiency of LM-MC methods compared with finite difference methods and LM-L. The number of computations used by layer methods in one time step is on the same order as that by finite difference methods; both are of $\mathcal{O}(N_1 N_2)$, with N_1 and N_2 the number of grid points in dimensions 1 and 2,

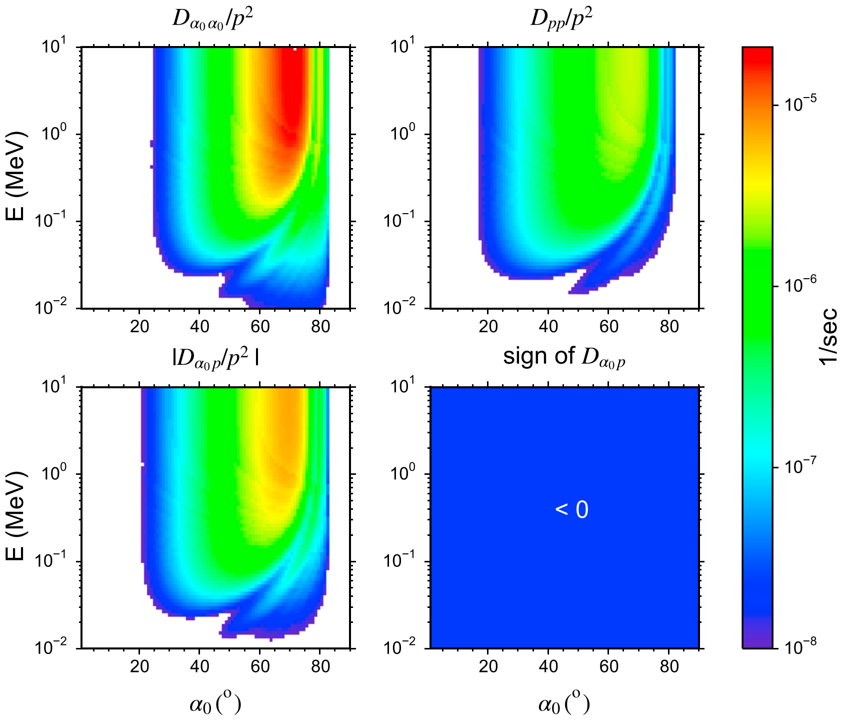

Figure 4. Inverse timescales in units of s^{-1} from bounce resonance diffusion coefficients calculated using the magnetosonic wave model from *Horne et al.* [2007]. The last panel shows the sign of the cross-diffusion coefficients.

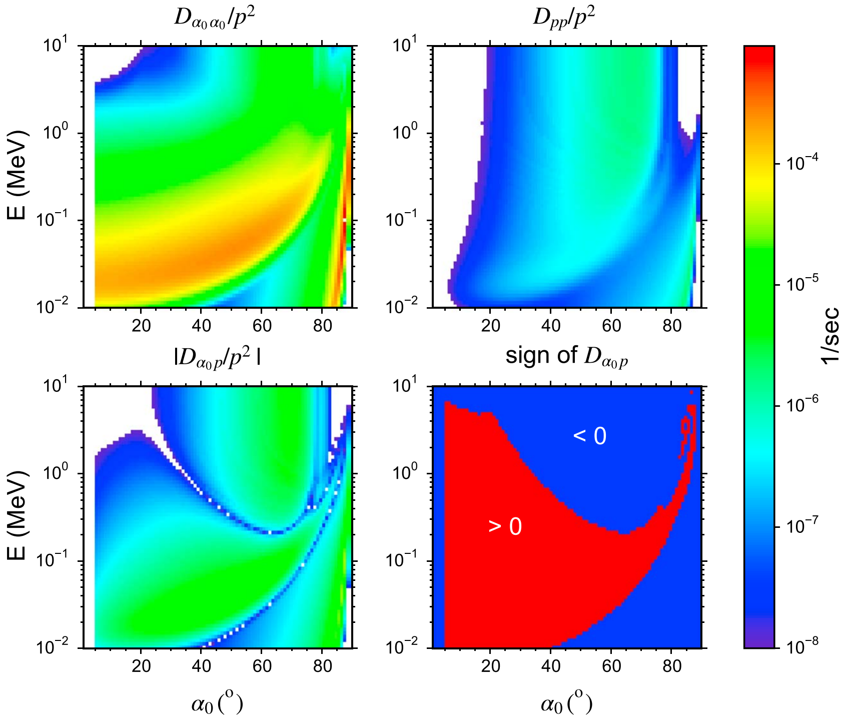

Figure 5. Same as Figure 4 but for the combined diffusion coefficients from bounce resonance with magnetosonic waves and gyroresonance with hiss waves.

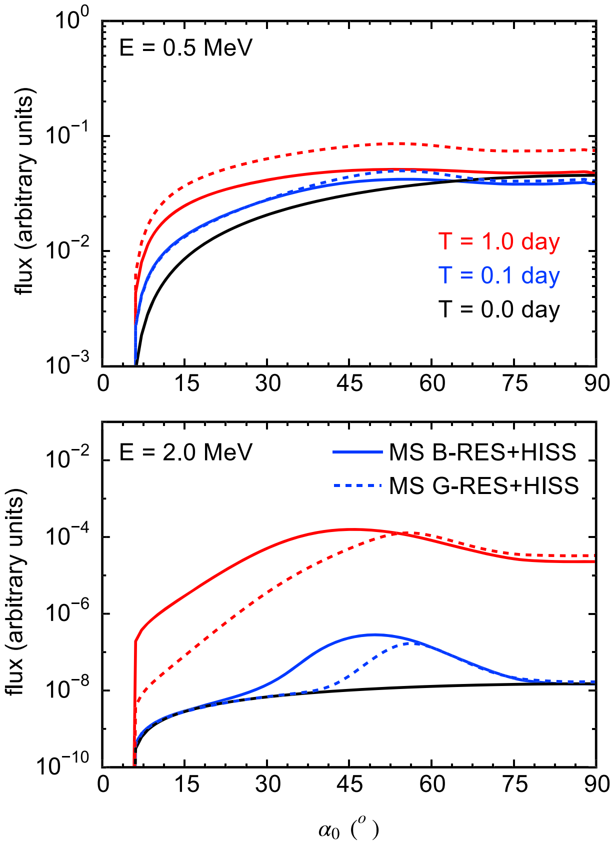

Figure 6. Evolution of electron fluxes using a combined wave model of magnetosonic waves and hiss waves for (top) $E = 0.5$ MeV and (bottom) $E = 2.0$ MeV at $T = 0.1$ day (blue) and 1 day (red). Dashed lines represent gyroresonant interactions with both magnetosonic waves and hiss, while solid lines represent gyroresonance with hiss and bounce resonance with magnetosonic waves. Black lines indicate initial fluxes.

respectively. Because of the extra steps involved in constraining $\dot{f}$, the computation time of LM-MC is longer than finite difference methods if using the same numbers of grid points and time steps. For example, with a CPU of 3.0 GHz, it takes the 2-D LM-MC code about 5 min to finish the calculation in this section, while the computation time for a finite difference code we developed is about 1 min under the same settings. The main disadvantage of finite difference methods is that they do not guarantee positivity of the solution and typically involves trying different settings of grid resolutions to obtain results free of negative values. For example, *Camporeale et al.* [2013] demonstrated that when solving a 2-D problem with a fully implicit finite difference method, negative phase space densities exist even when $N_1 = N_2 = 201$, although their absolute values are small. For time explicit finite difference methods, using a smaller grid size also means a smaller time step, therefore more computation time, because of the Courant-Friedrichs-Lewy condition. On the other hand, one can choose any N_1 and N_2 with LM-MC and completely avoid nonphysical negative solutions. Therefore, the total time used by finite difference methods to achieve positive solutions could be comparable to or even longer than LM-MC, either from extra tries of different grid resolutions or from the use of a large number of grid points. Compared with LM-L, which requires the use of a large number of grid points to obtain accurate solutions of equation (46) [*Tao et al.*, 2009], the LM-MC can significantly save computation time by using a much smaller number of grid points. For example, LM-L would require $N_{\alpha_0} = 1400$ and $N_p = 1500$ for the problem in this section [*Tao et al.*, 2009], and the resulting computation time is a few hours on the same computer. Therefore, we conclude that LM-MC can be used for long-term modeling of radiation belt dynamics.

3.2. Bounce Resonance Diffusion by Magnetosonic Waves

We now use the 2-D LM-MC code to investigate the effect of bounce resonance diffusion of relativistic electrons by magnetosonic waves. Because magnetosonic waves are confined to equatorial regions, the bounce resonance diffusion coefficients derived by *Roberts and Schulz* [1968] cannot be used directly; they assumed that waves cover the whole bounce trajectory of particles. Recently, *Li et al.* [2015] extended the work of

Roberts and Schulz [1968] and derived bounce resonance diffusion coefficients for spatially confined waves such as magnetosonic waves. Using a previously published magnetosonic wave model by *Horne et al.* [2007], they concluded that bounce resonance diffusion is as important as gyroresonance diffusion (from Landau resonance). Note that for simplicity, instead of a spread in wave normal angle direction, *Li et al.* [2015] assumed a single wave normal angle $\psi = 89°$. The calculated bounce resonance diffusion coefficients are shown in Figure 4 assuming $\omega_{pe}/|\Omega_e| = 3$, where ω_{pe} and Ω_e are the plasma frequency and the electron cyclotron frequency at the equator, respectively.

Like gyroresonance diffusion coefficients, the bounce resonance diffusion coefficients of magnetosonic waves are significant only over a limited pitch angle range for a given energy using the model of *Horne et al.* [2007]. Therefore, as in *Tao et al.* [2009], we combine bounce resonance diffusion coefficients of magnetosonic waves outside the plasmasphere with gyroresonance diffusion coefficients by hiss waves inside plasmaspheric plumes from *Li et al.* [2007]. The drift averaged diffusion coefficients from bounce resonance with magnetosonic waves (60%) and gyroresonance with hiss waves (15%) are shown in Figure 5. We adopt the parameter $\epsilon \equiv |D_{\alpha_0 p}|/\sqrt{D_{\alpha_0\alpha_0}D_{pp}}$ from *Albert* [2009] to indicate the relative importance of cross-diffusion terms. The mean value of ϵ is 0.6, and the maximum value of ϵ is about 1.0; these values suggest that cross-diffusion terms are important for this set of diffusion coefficients. We use the same initial and boundary conditions as in section 3.1. The resulting evolution of electron fluxes for $E = 0.5$ MeV and $E = 2.0$ MeV at $T = 0.1$ and 1.0 day is shown in Figure 6 as solid lines together with initial fluxes at these two energies. Bounce resonance with magnetosonic waves can increase the flux of 2.0 MeV electrons by a few orders of magnitudes within a day, and the flux peaks around 45°, producing a butterfly distribution. Also shown by dashed lines in Figure 6 are the corresponding fluxes resulting from gyroresonance with magnetosonic waves and hiss waves calculated by *Tao et al.* [2009]. We see that bounce resonance with magnetosonic waves is as important as gyroresonance, confirming the conclusion of *Li et al.* [2015]. Note that even though there is a difference in wave normal angle distributions used in calculating bounce resonance and gyroresonance diffusion coefficients of magnetosonic waves, adopting a spread in wave normal angle in bounce resonance diffusion calculation should not change our conclusion qualitatively. Our results suggest that bounce resonance should be incorporated into modeling when gyroresonance with magnetosonic waves is considered to be important.

4. Summary

In this work, we introduced the use of monotonicity- and positivity-preserving cubic interpolation methods in the Milstein-Tretyakov layer method. The resulting method, LM-MC, is positivity preserving and can be used to solve the radiation belt diffusion equation with a much smaller number of grid points compared with the layer method with linear interpolation; correspondingly, a significant amount of computation time can be saved. The LM-MC can therefore be used to model long-term radiation belt dynamics.

We then developed a 2-D LM-MC code and used it to investigate the bounce resonance diffusion of radiation belt electrons by magnetosonic waves. Using a previously published magnetosonic wave model [*Horne et al.*, 2007] combined with a plasmaspheric plume hiss wave model [*Li et al.*, 2007], our simulation demonstrated that bounce resonance with magnetosonic waves is as important as gyroresonance, with both leading to significant acceleration of MeV electrons for the wave model used [*Li et al.*, 2015]. Our results suggest that bounce resonance with magnetosonic waves should be taken into consideration when gyroresonance with magnetosonic waves is considered to be important.

Appendix A: The Relationship Between the PP Limiter and the M3 Limiter

In this section, we prove that if $\dot{f}_i$ is within the M3 interval, equation (37), then $\dot{f}_i$ automatically satisfies the PP condition, equation (41), if data are monotone at x_i or if x_i is a local maximum.

First, assume that the data are monotone and increasing; i.e., $f_{i-1} < f_i < f_{i+1}$. In this case, $s_{i-1/2} > 0$ and $s_{i+1/2} > 0$. Assuming that $s_{i+1/2} > s_{i-1/2}$, the M3 interval, $\dot{f}_i \in I[0, 3s_i]$, becomes $0 < \dot{f}_i < 3s_{i-1/2}$. Note that equation (41) can be written as

$$-\frac{3f_i}{\Delta x_{i+1/2}} \le \dot{f}_i \le 3s_{i-1/2} + \frac{3f_{i-1}}{\Delta x_{i-1/2}}. \tag{A1}$$

Because all f_is are positive, $3f_{i-1}/\Delta x_{i-1/2} > 0$ and $3f_i/\Delta x_{i+1/2} > 0$; therefore,

$$-\frac{3f_i}{\Delta x_{i+1/2}} < \underbrace{0 < \dot{f}_i < 3s_{i-1/2}}_{\text{the M3 interval}} < 3s_{i-1/2} + 3f_{i-1}/\Delta x_{i-1/2}. \tag{A2}$$

Hence, the PP condition is automatically satisfied. If, on the other hand, $s_{i+1/2} < s_{i-1/2}$, the M3 limiter leads to $0 < \dot{f}_i < 3s_{i+1/2} < 3s_{i-1/2}$, and the above proof is still correct. The case where data at x_i are monotone and decreasing can be proved similarly.

Second, if x_i is a local maximum, $f_{i-1} < f_i$ and $f_i > f_{i+1}$, $s_{i-1/2} > 0$ and $s_{i+1/2} < 0$. The M3 interval, equation (37), becomes

$$\dot{f}_i \in I\left[0, \frac{3}{2}t_i\right]. \tag{A3}$$

Three cases can be discussed. First, If $t_i = 0$, then $\dot{f}_i = 0$, and the PP condition (41) is satisfied. Second, If $t_i > 0$, from the definition of t_i, equation (36), $p'_{i+1/2}(x_i) > 0$ and $p'_{i-1/2}(x_i) > 0$. Note that from equation (34),

$$p'_{i+1/2}(x_i) \in I\left[s_{i+1/2}, p'_i(x_i)\right]; \tag{A4}$$

similarly,

$$p'_{i-1/2}(x_i) \in I\left[s_{i-1/2}, p'_i(x_i)\right]. \tag{A5}$$

Because $p'_i(x_i) \in I[s_{i-1/2}, s_{i+1/2}]$ from equation (30), equations (A4) and (A5) imply that

$$p'_{i-1/2} \in I[s_{i-1/2}, s_{i+1/2}] \text{ and } p'_{i+1/2} \in I[s_{i-1/2}, s_{i+1/2}]. \tag{A6}$$

Because $p'_{i\pm1/2} > 0$, $s_{i-1/2} > 0$ and $s_{i+1/2} < 0$,

$$0 < p'_{i-1/2} \le s_{i-1/2} \text{ and } 0 < p'_{i+1/2} \le s_{i-1/2}. \tag{A7}$$

From the definition of t_i, $0 < t_i \le s_{i-1/2}$; therefore,

$$-\frac{3f_i}{\Delta x_{i+1/2}} < \underbrace{0 \le \dot{f}_i \le \frac{3}{2}t_i}_{\text{the M3 interval}} \le \frac{3}{2}s_{i-1/2} < 3s_{i-1/2} < 3s_{i-1/2} + \frac{3f_{i-1}}{\Delta x_{i-1/2}}. \tag{A8}$$

Correspondingly, the PP condition is automatically satisfied for $\dot{f}_i$. Third, if $t_i < 0$, similar to second case, we have $s_{i+1/2} \le t_i < 0$. The M3 condition becomes

$$-\frac{3f_i}{\Delta x_{i+1/2}} = 3s_{i+1/2} - \frac{3f_{i+1}}{\Delta x_{i+1/2}} \le 3s_{i+1/2} < \frac{3}{2}s_{i+1/2} \le \underbrace{\frac{3}{2}t_i \le \dot{f}_i \le 0}_{\text{the M3 interval}} < \frac{3f_i}{\Delta x_{i-1/2}}. \tag{A9}$$

Therefore, the PP condition is automatically satisfied for $\dot{f}_i$.

Appendix B: The Order of Accuracy of Different Interpolation Algorithms

In this section, we show the order of accuracy of three different interpolation methods using a fourth-order polynomial

$$y(x) = 5x^4 + 4x^3 + 3x^2 + 2x + 1; x \in [0, 4]. \tag{B1}$$

We divide the x domain into $N - 1$ bins and consider $N = 10, 20, 40, 80, 160, 320$, and 640. For each N, we randomly select 500 points located between $x = 0$ and 4. Approximate values of y, denoted by $\tilde{y}$, are obtained from three interpolation methods: linear, the M3-A method, and the M3-Q method. Both M3-A and M3-Q methods have the PP limiter applied, meaning that positivity of the data is preserved.

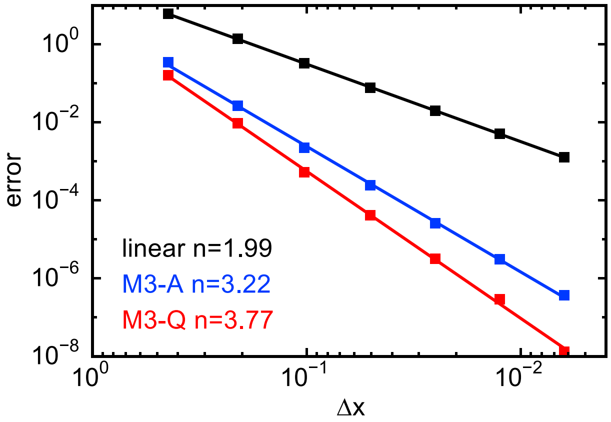

Figure B1. The order of accuracy test for three different interpolation methods: linear interpolation (black), M3-A (blue), and M3-Q (red). Black squares are interpolation errors at different grid sizes.

For a given N, the error of the interpolation is estimated using

$$\text{error} = \left(\sum_{i=1}^{500} |\widetilde{y}_i - y_i| \right) \Big/ 500. \tag{B2}$$

We then plot 7 calculated errors as a function of Δx, the grid size, and perform a least squares fitting to obtain

$$\log(\text{error}) \propto n \log(\Delta x). \tag{B3}$$

Hence, n is the order of accuracy of the corresponding interpolation method.

The results for linear, M3-A, and M3-Q methods are shown in Figure B1. We see that the order of accuracy of linear interpolation is about 2, just as expected. For M3-A, $n \approx 3.2$, and for M3-Q, $n \approx 3.77$. These results are consistent with a theoretical estimate of the order of accuracy of M3-A and M3-Q by *Huynh* [1993]. Other types of functions we have tested but not shown here are x^{-7}, e^x, e^{-x^2}, and $\sin(x)$; conclusions are similar.

Appendix C: The Comparison of Efficiency and Accuracy of the LM-MC and LM-L

To compare the efficiency and accuracy of LM-MC with LM-L used by *Tao et al.* [2009], we solve a 1-D pitch angle diffusion equation for $E = 0.4$ MeV electrons

$$\frac{\partial f}{\partial t} = \frac{1}{G} \frac{\partial}{\partial \alpha_0} \left(\frac{G D_{\alpha_0 \alpha_0}}{p^2} \frac{\partial f}{\partial \alpha_0} \right). \tag{C1}$$

We use an artificial pitch angle diffusion coefficient here for simplicity

$$\frac{D_{\alpha_0 \alpha_0}}{p^2} = 1 + \frac{p}{mc} \sin \alpha_0. \tag{C2}$$

Initial and boundary conditions in α_0 are the same as in equations (47)–(48).

We solve the pitch angle diffusion equation for f at a given time $t = T$ using three different methods: NDSolve from the commercial software Mathematica [*Wolfram Research*, 2014], LM-MC, and LM-L. The function NDSolve solves the diffusion equation using an adaptive solver so that the total error is within a certain range; thus, we do not need to specify the time step and the number of grid points for it. We use the solution from NDSolve as a reference to compare the accuracy of the LM-MC and the LM-L. In both layer methods, we use a time step $\Delta t = 10^{-3} \Delta \alpha_0$, where $\Delta \alpha_0$ is the grid size of α_0. The final time T in all simulations is chosen to correspond to 10000 time steps when we use 100 grid cells; i.e., $T = 1.7\pi/36$. For LM-MC, we use 50 grid cells, while for the LM-L we use a series of number of grid cells (N_{α_0}): 50, 100, 200, 400, 800, 1600, and 3200.

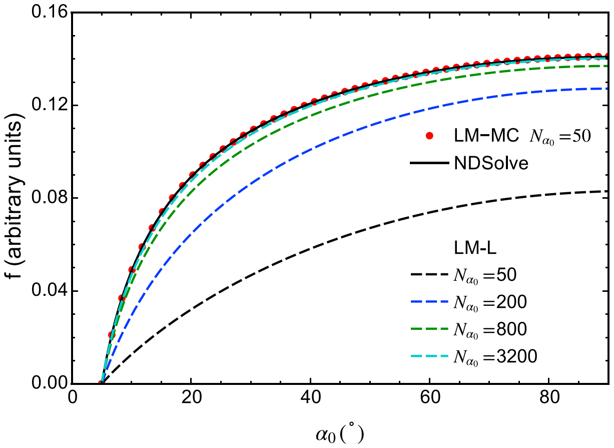

Figure C1. Comparison between solutions at a given time of a 1-D pitch angle diffusion equation using LM-MC (red dots), LM-L (dashed lines), and NDSolve (the solid black line). For LM-L, solutions with different numbers of grid cells (N_{α_0}) are shown to demonstrate the convergence of the solution.

The solutions from three different methods at $t = T$ are shown in Figure C1. Note that for the LM-L, we only show results with $N_{\alpha_0} = 50, 200, 800$, and 3200 for simplicity and clarity of the figure. We use the ε parameter defined by FC1

$$\varepsilon = \text{mean}(|\log(f_{\text{NDSolve}}) - \log(f_{\text{LM}})|), \tag{C3}$$

to measure the error of LM-MC and LM-L. Here the subscript "LM" indicates either LM-MC or LM-L. We find that $\varepsilon \approx 0.0007$ for LM-MC with $N_{\alpha_0} = 50$, indicating almost perfect agreement. On the other hand, the LM-L solution is significantly smaller than the other two solutions when $N_{\alpha_0} = 50$ and the corresponding $\varepsilon \approx 0.8$, indicating that the use of the inaccurate linear interpolation has caused excessive numerical diffusion in this case. To obtain accurate solutions, we have to increase the number of grid cells used. With $\Delta t = 10^{-3}\Delta\alpha_0$, this means that we will also use a smaller time step when $\Delta\alpha_0$ is decreased. For this case, the solution from the LM-L agrees with the NDSolve solution only when $N_{\alpha_0} = 3200$ ($\varepsilon = 0.01$). This comparison demonstrates that compared with LM-L the LM-MC can solve the diffusion equation accurately with a much smaller number of grid points and preserves the positivity of the solution; therefore, the LM-MC can save a significant amount of computation time, and it can be used for long-term simulation of the radiation belt dynamics.

Acknowledgments
This work was supported by NSFC grants 41474142 and 41421063 and CAS Key Research Program KZZD-EW-01-4. The model data will be preserved on a long-term storage system and will be made available upon request to the corresponding author.

References

Akima, H. (1970), A new method of interpolation and smooth curve fitting based on local procedures, *J. ACM*, *17*(4), 589–602.

Albert, J. M. (2004), Using quasi-linear diffusion to model acceleration and loss from wave-particle interactions, *Space Weather*, *2*, S09S03, doi:10.1029/2004SW000069.

Albert, J. M. (2009), The coupling of quasi-linear pitch angle and energy diffusion, *J. Atmos. Sol. Terr. Phys.*, *71*, 1664–1668, doi:10.1016/j.jastp.2008.11.014.

Albert, J. M. (2013), Comment on "On the numerical simulation of particle dynamics in the radiation belt. Part I: Implicit and semi-implicit schemes" and "On the numerical simulation of particle dynamics in the radiation belt. Part II: Procedure based on the diagonalization of the diffusion tensor" by E. Camporeale et al., *J. Geophys. Res. Space Physics*, *118*, 7762–7764, doi:10.1002/2013JA019126.

Albert, J. M., and S. L. Young (2005), Multidimensional quasi-linear diffusion of radiation belt electrons, *Geophys. Res. Lett.*, *32*, L14110, doi:10.1029/2005GL023191.

Albert, J. M., N. P. Meredith, and R. B. Horne (2009), Three-dimensional diffusion simulation of outer radiation belt electrons during the 9 October 1990 magnetic storm, *J. Geophys. Res.*, *114*, A09214, doi:10.1029/2009JA014336.

Beutier, T., and D. Boscher (1995), A three-dimensional analysis of the electron radiation belt by the Salammbô code, *J. Geophys. Res.*, *100*(A8), 14,853–14,861, doi:10.1029/94JA03066.

Camporeale, E., G. L. Delzanno, S. Zaharia, and J. Koller (2013), Reply to comment by J. M. Albert on "On the numerical simulation of particle dynamics in the radiation belt. Part I: Implicit and semi-implicit schemes" and "On the numerical simulation of particle dynamics in the radiation belt. Part II: Procedure based on the diagonalization of the diffusion tensor", *J. Geophys. Res. Space Physics*, *118*, 7765–7767, doi:10.1002/2013JA019389.

Carlson, R. E., and F. N. Fritsch (1985), Monotone piecewise bicubic interpolation, *SIAM J. Numer. Anal.*, *22*(2), 386–400.

Fritsch, F., and R. Carlson (1980), Monotone piecewise cubic interpolation, *SIAM J. Numer. Anal.*, *17*(2), 238–246, doi:10.1137/0717021.

Glauert, S. A., R. B. Horne, and N. P. Meredith (2014), Three-dimensional electron radiation belt simulations using the BAS Radiation Belt Model with new diffusion models for chorus, plasmaspheric hiss, and lightning-generated whistlers, *J. Geophys. Res. Space Physics*, *119*, 268–289, doi:10.1002/2013JA019281.

Horne, R. B., R. M. Thorne, S. A. Glauert, J. M. Albert, N. P. Meredith, and R. R. Anderson (2005), Timescale for radiation belt electron acceleration by whistler mode chorus waves, *J. Geophys. Res.*, *110*, A03225, doi:10.1029/2004JA010811.

Horne, R. B., R. M. Thorne, S. A. Glauert, N. P. Meredith, D. Pokhotelov, and O. Santolík (2007), Electron acceleration in the Van Allen radiation belts by fast magnetosonic waves, *Geophys. Res. Lett.*, *34*, L17107, doi:10.1029/2007GL030267.

Huynh, H. T. (1993), Accurate monotone cubic interpolation, *SIAM J. Numer. Anal.*, *30*(1), 57–100, doi:10.1137/0730004.

Li, W., Y. Y. Shprits, and R. M. Thorne (2007), Dynamic evolution of energetic outer zone electrons due to wave-particle interactions during storms, *J. Geophys. Res.*, *112*, A10220, doi:10.1029/2007JA012368.

Li, X., X. Tao, Q. Lu, and L. Dai (2015), Bounce resonance diffusion coefficients for spatially confined waves, *Geophys. Res. Lett.*, *42*, 9591–9599, doi:10.1002/2015GL066324.

Milstein, G. N. (2002), The probability approach to numerical solution of nonlinear parabolic equations, *Numer. Methods Partial Differential Eq.*, *18*, 490–522, doi:10.1002/num.10020.

Milstein, G. N., and M. V. Tretyakov (2001), Numerical solution of the Dirichlet problem for nonlinear parabolic equations by a probabilistic approach, *IMA J. Numer. Anal.*, *21*, 887–917.

Milstein, G. N., and M. V. Tretyakov (2002), A probabilistic approach to the solution of the Neumann problem for nonlinear parabolic equations, *IMA J. Numer. Anal.*, *22*, 599–622.

Roberts, C. S., and M. Schulz (1968), Bounce resonant scattering of particles trapped in the Earth's magnetic field, *J. Geophys. Res.*, *73*(23), 7361–7376, doi:10.1029/JA073i023p07361.

Schulz, M., and L. J. Lanzerotti (1974), *Particle Diffusion in the Radiation Belts, Physics and Chemistry in Space*, vol. *7*, Springer-Verlag, New York.

Shprits, Y. Y., S. R. Elkington, N. P. Meredith, and D. A. Subbotin (2008a), Review of modeling of losses and sources of relativistic electrons in the outer radiation belt I: Radial transport, *J. Atmos. Sol. Terr. Phys.*, *70*, 1679–1693, doi:10.1016/j.jastp.2008.06.008.

Shprits, Y. Y., D. A. Subbotin, N. P. Meredith, and S. R. Elkington (2008b), Review of modeling of losses and sources of relativistic electrons in the outer radiation belt II: Local acceleration and loss, *J. Atmos. Sol. Terr. Phys.*, *70*, 1694–1713, doi:10.1016/j.jastp.2008.06.014.

Su, Z., F. Xiao, H. Zheng, and S. Wang (2010), STEERB: A three-dimensional code for storm-time evolution of electron radiation belt, *J. Geophys. Res.*, *115*, A09208, doi:10.1029/2009JA015210.

Subbotin, D., Y. Shprits, and B. Ni (2010), Three-dimensional VERB radiation belt simulations including mixed diffusion, *J. Geophys. Res.*, *115*, A03205, doi:10.1029/2009JA015070.

Tao, X., A. A. Chan, J. M. Albert, and J. A. Miller (2008), Stochastic modeling of multidimensional diffusion in the radiation belts, *J. Geophys. Res.*, *113*, A07212, doi:10.1029/2007JA012985.

Tao, X., J. M. Albert, and A. A. Chan (2009), Numerical modeling of multidimensional diffusion in the radiation belts using layer methods, *J. Geophys. Res.*, *114*, A02215, doi:10.1029/2008JA013826.

Tao, X., R. M. Thorne, W. Li, B. Ni, N. P. Meredith, and R. B. Horne (2011a), Evolution of electron pitch-angle distributions following injection from the plasma sheet, *J. Geophys. Res.*, *116*, A04229, doi:10.1029/2010JA016245.

Tao, X., R. M. Thorne, R. B. Horne, B. Ni, J. D. Menietti, Y. Y. Shprits, and D. A. Gurnett (2011b), Importance of plasma injection events for energization of relativistic electrons in the Jovian magnetosphere, *J. Geophys. Res.*, *116*, A01206, doi:10.1029/2010JA016108.

Tu, W., G. S. Cunningham, Y. Chen, M. G. Henderson, E. Camporeale, and G. D. Reeves (2013), Modeling radiation belt electron dynamics during GEM challenge intervals with the DREAM3D diffusion model, *J. Geophys. Res. Space Physics*, *118*, 6197–6211, doi:10.1002/jgra.50560.

Varotsou, A., D. Boscher, S. Bourdarie, R. B. Horne, S. A. Glauert, and N. P. Meredith (2005), Simulation of the outer radiation belt electrons near geosynchronous orbit including both radial diffusion and resonant interaction with whistler-mode chorus waves, *Geophys. Res. Lett.*, *32*, L19106, doi:10.1029/2005GL023282.

Wolfram Research, I. (2014), *Mathematica*, Wolfram Research, Inc., version 10.0 ed., Champaign, Ill.

Xiao, F., Z. Su, H. Zheng, and S. Wang (2009), Modeling of outer radiation belt electrons by multidimensional diffusion process, *J. Geophys. Res.*, *114*, A03201, doi:10.1029/2008JA013580.

Xiao, F., et al. (2015), Wave-driven butterfly distribution of Van Allen belt relativistic electrons, *Nat. Commun.*, *6*, 8590, doi:10.1038/ncomms9590.

Zheng, L., A. A. Chan, J. M. Albert, S. R. Elkington, J. Koller, R. B. Horne, S. A. Glauert, and N. P. Meredith (2014), Three-dimensional stochastic modeling of radiation belts in adiabatic invariant coordinates, *J. Geophys. Res. Space Physics*, *119*, 7615–7635, doi:10.1002/2014JA020127.