A&A 501, 813–820 (2009)
DOI: 10.1051/0004-6361/200911793

Astronomy & Astrophysics

Comoving-frame radiative transfer in arbitrary velocity fields

II. Large scale applications

S. Knop[1], P. H. Hauschildt[1], and E. Baron[1,2,3]

[1] Hamburger Sternwarte, Gojenbergsweg 112, 21029 Hamburg, Germany
e-mail: [sknop;yeti]@hs.uni-hamburg.de
[2] Homer L. Dodge Department of Physics and Astronomy, University of Oklahoma, 440 W Brooks, Rm 100, Norman, OK 73019, USA
e-mail: baron@ou.edu
[3] Computational Research Division, Lawrence Berkeley National Laboratory, MS 50F-1650, 1 Cyclotron Rd, Berkeley, CA 94720, USA

Received 4 February 2009 / Accepted 18 March 2009

ABSTRACT

Aims. A solution of the radiative-transfer problem in arbitrary velocity fields introduced in a previous paper, has limitations in its applicability. For large-scale applications, the methods described also require large memory sets that are commonly not available to state-of-the-art computing hardware. In this work, we modify the algorithm to allow the computation of large-scale problems.

Methods. We reduce the memory footprint via a domain decomposition. By introducing iterative Gauss-Seidel type solvers, we improve the speed of the overall computation. Because of the domain decomposition, the new algorithm requires the use of parallel-computing systems.

Results. The algorithm that we present permits large-scale solutions of radiative-transfer problems that include arbitrary wavelength couplings. In addition, we discover a quasi-analytic formal solution of the radiative transfer that significantly improves the overall computation speed. More importantly, this method ensures that our algorithm can be applied to multi-dimensional Lagrangian radiative-transfer calculations. In multi-dimensional atmospheres, velocity fields are in general chaotic ensuring that the inclusion of arbitrary wavelength couplings are mandatory.

Key words. radiative transfer

1. Introduction

Radiative transfer in spherical symmetry and moving media has been solved with different methods. Operator-splitting techniques (Cannon 1973) remain the state-of-the-art methods. An operator-splitting method that uses an approximate Λ-operator (ALO) (Scharmer 1981) – called accelerated Λ-iteration (ALI) – for the solution of the special relativistic Lagrangian equation of radiative transfer was described, for example, in Hauschildt (1992). A Lagrangian description is necessary for the treatment of line transfer because of the complicated form of the emissivities and opacities in the Eulerian frame and also the large number of wavelength points needed to sample the spectral line at all points in the atmosphere. It follows that the velocity field must then appear explicitly in the equation of radiative transfer (Mihalas 1980). For monotonic velocity fields, the solution of the radiative-transfer problem becomes an initial value problem in wavelength that can be solved with the method described in Hauschildt (1992). The inclusion of non-monotonic velocity fields changes the problem into a boundary-value problem in wavelength. A solution method and its algorithm for this case was introduced in Baron & Hauschildt (2004, Paper I).

However, the applicability of the methods in Paper I depends on the size of the numerical system, which is determined mainly by the number of wavelength points as well as the number of layers in the 1-D model atmosphere. The number of wavelength points is the more critical factor, since it is commonly much higher than the number of layers in 1-D. The number of layers is however important, because the number of entries in the Λ-operator described in Paper I is proportional to the number of layers squared.

On average hardware, the algorithm from Paper I is only suited to solving problems with 64 layers and about 1000 wavelength points because of the limited memory available. Configurations with more than 1000 wavelength points are called in the following, large-scale applications.

The work from Paper I must be regarded as a proof of concept. In this work, we present a new algorithm that is capable of efficiently solving large-scale applications. Besides being immediately useful for 1-D applications, the results are also applicable to the development of 3-D radiative transfer (Hauschildt & Baron 2006; Baron & Hauschildt 2007; Hauschildt & Baron 2008).

In Sect. 2, we summarise the formalism again for completeness. The approach for the new algorithm is described in Sect. 3. In Sect. 4, we present the results for test calculations and scalability tests before we conclude in Sect. 5

2. The formal framework in general

The framework presented in Baron & Hauschildt (2004) was developed for just one possible wavelength discretisation. During the course of this work, it became clear that a fully implicit discretisation (Hauschildt & Baron 2004) is necessary to solve

Article published by EDP Sciences

the radiative-transfer problem in all generality. This is because for strong wavelength couplings the interpretation of the entire wavelength discretisation as a source term is not valid for all optical depths. Furthermore, the use of a new formal solution that avoids negative generalised opacities must be used (Knop et al. 2008). Therefore, we discuss here the basic framework in those parts that are different from Baron & Hauschildt (2004).

The equation of radiative transfer in its characteristic form for the specific intensity I along a path s reads:

$$\frac{\mathrm{d}I_l}{\mathrm{d}s} = \eta_l - \chi_l I_l - 4a_l I_l - a_l \frac{\partial(\lambda I_l)}{\partial\lambda}, \tag{1}$$

where η is the emissivity, χ the opacity, and the subscript l indicates the dependence of the intensity on wavelength. The a_l term is the coupling term between the wavelengths that depends on both the structure of the atmosphere and the mechanism of the coupling (Mihalas 1980).

The wavelength derivative can be discretised in two ways as described in Hauschildt & Baron (2004). The different discretisations can be mixed via a Crank-Nicholson scheme with a mixing parameter $\xi \in [0, 1]$. The wavelength-discretised equation of radiative transfer can then be written as:

$$\begin{aligned}\frac{\mathrm{d}I_l}{\mathrm{d}s} &= \eta_l - \chi_l I_l - a_l \left(4 + \xi\, p_l^{|}\right) I_l \\ &\quad -\xi\, a_l \left(p_l^{-} I_{l-1} + p_l^{+} I_{l+1}\right) \\ &\quad -[1-\xi]\, a_l \left(p_l^{-} I_{l-1} + p_l^{|} I_l + p_l^{+} I_{l+1}\right) \end{aligned} \tag{2}$$

where the $p_l^{\bullet}$ coefficients in an ordered wavelength grid $\lambda_{l-1} < \lambda_l < \lambda_{l+1}$ are defined as:

$$\left.\begin{aligned} p_l^{-} &= -\frac{\lambda_{l-1}}{\lambda_l - \lambda_{l-1}} \\ p_l^{|} &= \frac{\lambda_l}{\lambda_l - \lambda_{l-1}} \\ p_l^{+} &= 0 \end{aligned}\right\} \text{for:}\quad a_{\lambda_l} \geq 0 \tag{3}$$

$$\left.\begin{aligned} p_l^{-} &= 0 \\ p_l^{|} &= \frac{\lambda_l}{\lambda_l - \lambda_{l+1}} \\ p_l^{+} &= -\frac{\lambda_{l+1}}{\lambda_l - \lambda_{l+1}} \end{aligned}\right\} \text{for:}\quad a_{\lambda_l} < 0. \tag{4}$$

The dependence on the sign of a_λ is introduced to define local upwind schemes (see Baron & Hauschildt 2004).

After introducing a generalised opacity (see Knop et al. 2008)

$$\hat{\chi}_l = \chi_l + \xi\, a_l p_l^{|}, \tag{5}$$

as well as defining the source functions:

$$S_l = \frac{\eta_l}{\chi_l} \tag{6}$$

$$\hat{S}_l = \frac{\chi_l}{\hat{\chi}_l}\left(S_l - \xi\,\frac{a_l}{\chi_l}\left(p_l^{-} I_{l-1} + p_l^{+} I_{l+1}\right)\right) \tag{7}$$

$$\tilde{S}_l = -\frac{a_l}{\hat{\chi}_l}\left([1-\xi]\,\left(p_l^{-} I_{l-1} + p_l^{+} I_{l+1}\right) + \left[4 + [1-\xi]\,p_l^{|}\right] I_l\right) \tag{8}$$

a formal solution of the radiative-transfer problem can be formulated. In our work, we use a characteristic method along different photon paths throughout the atmosphere. The spatial position of a characteristic is then discretised on a spatial grid. In the following, a pair of subscript indices indicate the position in both the spatial grid and the wavelength grid. Commonly the spatial grid is mapped onto an optical depth grid via the relation $\mathrm{d}\tau_l = \hat{\chi}_l \mathrm{d}s$. The formal solution of the equation of radiative transfer in Eq. (2) between two points s_{i-1} and s_i on a spatial grid along the photon path can be written in terms of the optical depth as follows:

$$I_{i,l} = I_{i-1,l}\mathrm{e}^{-\Delta\tau} + \delta\hat{I}_{i,l} + \delta\tilde{I}_{i,l} \tag{9}$$

$$\delta\hat{I}_{i,l} = \int_{\tau_{i-1}}^{\tau_i} \hat{S}_l \mathrm{e}^{\tau-\tau_i}\mathrm{d}\tau = \alpha_{i,l}\hat{S}_{i-1,l} + \beta_{i,l}\hat{S}_{i,l} + \gamma_{i,l}\hat{S}_{i+1,l} \tag{10}$$

$$\delta\tilde{I}_{i,l} = \int_{\tau_{i-1}}^{\tau_i} \tilde{S}_l \mathrm{e}^{\tau-\tau_i}\mathrm{d}\tau = \tilde{\alpha}_{i,l}\tilde{S}_{i-1,l} + \tilde{\beta}_{i,l}\tilde{S}_{i,l} \tag{11}$$

with $\Delta\tau = \tau_{i+1,l} - \tau_{i,l}$ and $\tau_{i,l} = \int^{s_i} \chi_l(s)\mathrm{d}s$. The α-β-γ coefficients are described in Olson & Kunasz (1987) and Hauschildt (1992). In Eq. (11) $\delta\tilde{I}_l$ is interpolated linearly and the coefficients differ in general from the coefficients in Eq. (10), and is therefore marked with a tilde.

Equation (9) can be written in matrix notation for any given characteristic:

$$\boldsymbol{I} = \boldsymbol{A} \cdot \boldsymbol{I} + \boldsymbol{\Delta I}. \tag{12}$$

Here $\boldsymbol{I}$ is a vector containing all intensities, $\boldsymbol{A}$ is a square matrix that describes the influence of the different intensities upon each other, and $\boldsymbol{\Delta I}$ is a vector with the thermal emission and scattering contribution of the source function. For a characteristic with n_i spatial points and n_l points in the wavelength grid, the intensity vector $\boldsymbol{I}$ has $n_i \times n_l$ entries. In the following, a superscript of k labels the characteristic being described. The components of the matrix $\boldsymbol{A}$ from Eq. (12) at the spatial point i and the wavelength point l are given by:

$$A_{i,l}^{-,k} = -\left(\xi\alpha_{i,l}^{k} + [1-\xi]\,\tilde{\alpha}_{i,l}^{k}\right)\frac{a_{i-1,l}^{k}}{\hat{\chi}_{i-1,l}^{k}} p_{i-1,l}^{-,k} \tag{13}$$

$$B_{i,l}^{-,k} = -\left(\xi\beta_{i,l}^{k} + [1-\xi]\,\tilde{\beta}_{i,l}^{k}\right)\frac{a_{i,l}^{k}}{\hat{\chi}_{i,l}^{k}} p_{i,l}^{-,k} \tag{14}$$

$$C_{i,l}^{-,k} = -\xi\gamma_{i,l}^{k}\frac{a_{i+1,l}^{k}}{\hat{\chi}_{i+1,l}^{k}} p_{i+1,l}^{-,k} \tag{15}$$

$$A_{i,l}^{\searrow,k} = \exp\left(-\Delta\tau_{i-1,l}^{k}\right) - \tilde{\alpha}_{i,l}^{k}\frac{a_{i-1,l}^{k}}{\hat{\chi}_{i-1,l}^{k}}\left[4 + [1-\xi] p_{i-1,l}^{|,k}\right] \tag{16}$$

$$B_{i,l}^{\searrow,k} = -\tilde{\beta}_{i,l}^{k}\frac{a_{i,l}^{k}}{\hat{\chi}_{i,l}^{k}}\left[4 + [1-\xi]\, p_{i,l}^{|,k}\right] \tag{17}$$

$$C_{i,l}^{\searrow,k} = -\hat{\gamma}_{i,l}^{k}\frac{a_{i+1,l}^{k}}{\hat{\chi}_{i+1,l}^{k}} p_{i+1,l}^{|,k} \tag{18}$$

$$C_{i,l}^{\searrow,k} = 0 \tag{19}$$

$$A_{i,l}^{+,k} = -\left(\xi\alpha_{i,l}^{k} + [1-\xi]\,\tilde{\alpha}_{i,l}^{k}\right)\frac{a_{i-1,l}^{k}}{\hat{\chi}_{i-1,l}^{k}} p_{i-1,l}^{+,k} \tag{20}$$

$$B_{i,l}^{+,k} = -\left(\xi\beta_{i,l}^{k} + [1-\xi]\,\tilde{\beta}_{i,l}^{k}\right)\frac{a_{i,l}^{k}}{\hat{\chi}_{i,l}^{k}} p_{i,l}^{+,k} \tag{21}$$

$$C_{i,l}^{+,k} = -\xi\gamma_{i,l}^{k}\frac{a_{i+1,l}^{k}}{\hat{\chi}_{i+1,l}^{k}} . p_{i+1,l}^{+,k}. \tag{22}$$

The naming scheme of the quantities defined in Eqs. (13) to (22) indicates the specific intensity element with which they are associated. For an index pair i and l, a $\bullet^{-}$ superscript refers to an

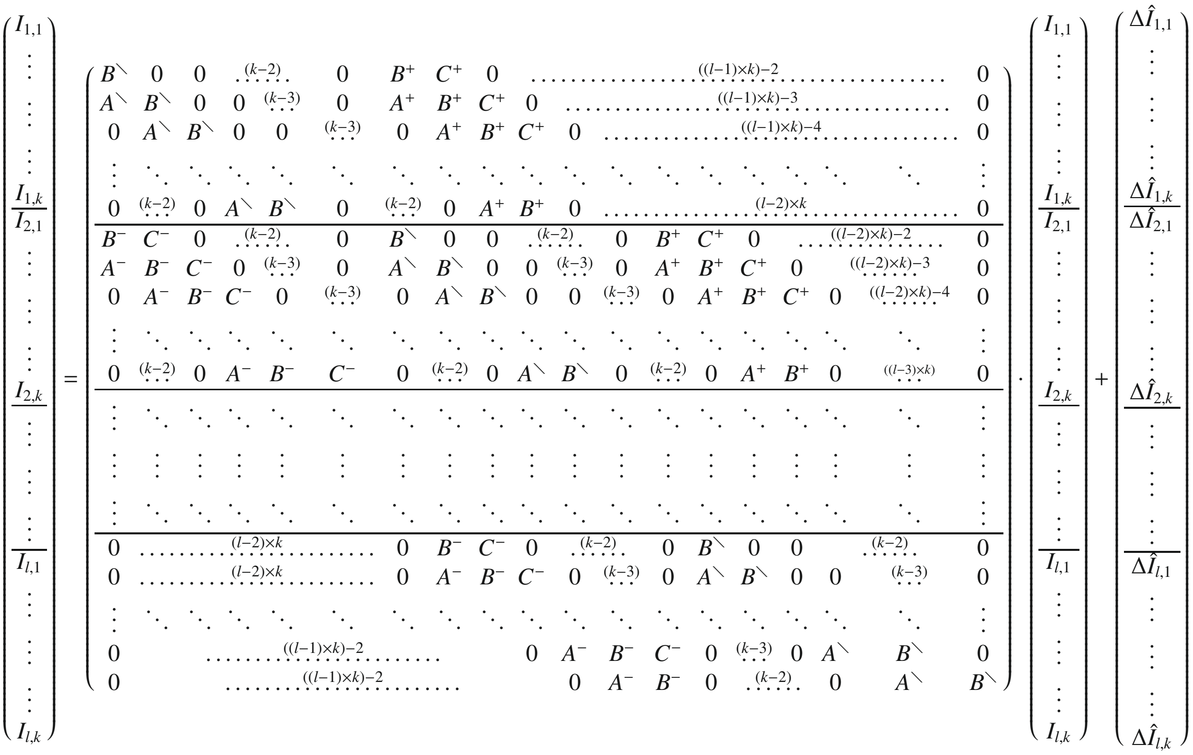

Fig. 1. Explicit matrix form of the formal solution for a characteristic of length k and l wavelength points. The horizontal lines mark block borders of different wavelengths to clarify the structure.
The matrix has three tridiagonal bands. The one on the main diagonal is called *diag*$(=\diagdown)$ and the lower and upper accordingly *sub*$(=-)$ and *super*$(=+)$. The diagonals of these bands are called A, B, and C.

intensity at wavelength $l-1$, a $\bullet^{\diagdown}$ superscript to the same wavelength, and $\bullet^{+}$ to the next wavelength point $l+1$. The A, B, and C terms refer to the spatial points $i-1$, i, and $i+1$ respectively. For clarity, the structure of the matrix of the formal solution is shown schematically in Fig. 1.

An element of the source function vector $\boldsymbol{\Delta I}$ is given by:

$$\Delta I^k_{i,l} = \alpha^k_{i,l} S^k_{i-1,l} + \beta^k_{i,l} S^k_{i,l} + \gamma^k_{i,l} S^k_{i+1,l}. \tag{23}$$

From Eq. (12), the solution for the specific intensity at a given spatial point and wavelength reads:

$$\begin{aligned} I^k_{i,l} = {} & \left(1 - B^{\mathrm{diag},k}_{i,l}\right)^{-1} \cdot \left(\Delta I^k_{i,l} + B^{\mathrm{sub},k}_{i,l} I^k_{i,l-1} + B^{\mathrm{super},k}_{i,l} I^k_{i,l+1}\right. \\ & + A^{\mathrm{sub},k}_{i,l} I^k_{i-1,l-1} + A^{\mathrm{diag},k}_{i,l} I^k_{i-1,l} + A^{\mathrm{super},k}_{i,l} I^k_{i-1,l+1} \\ & \left. + C^{\mathrm{sub},k}_{i,l} I^k_{i+1,l-1} + C^{\mathrm{super},k}_{i,l} I^k_{i+1,l+1}\right). \end{aligned} \tag{24}$$

Given the form of Eq. (24) for the formal solution, the construction of the Λ^*-operator can proceed exactly as described in Baron & Hauschildt (2004) and will not be discussed further here.

3. Optimisation of the algorithm

In the following sections, we outline the changes made to the algorithm from Paper I to improve the performance and usability of the solution to the radiative-transfer problem in the case of large-scale applications.

3.1. Smaller amount of memory requirements

The solution of the equation of transfer for arbitrary wavelength couplings is a boundary-value problem in wavelength (Baron & Hauschildt 2004), where the wavelength derivative sense changes throughout the atmosphere. This implies that the radiative transfer must be solved for all discrete wavelength points at the same time, and that all wavelength dependent quantities such as the opacities, interpolation coefficients, wavelength-derivative discretisation, and the Λ-operator must be kept in memory at the same time. For large-scale applications, these requirements easily exceed the memory of commonly available computer hardware. Therefore, the key method of solution is a domain decomposition of the data. Ideally, every process stores only the data that it works on. This immediately implies parallelisation of code execution as well.

The formal solution can be parallelised. The formal solutions for different characteristics are independent of each other and accordingly a computing node in a parallel setup must store only the data for those characteristics upon which it works.

The Λ^*-operator is the largest data object that must be retained in memory, and also offers the most hopeful possibility for optimisation. The operator has the full spatial bandwidth but is only tridiagonal in wavelength. Therefore, the number of entries of the Λ^*-operator is $n_{\mathrm{layer}} \times n_{\mathrm{layer}} \times n_\lambda \times 3$, where n_{layer} is the number of discrete radial points in a spherically symmetric atmosphere and n_λ is the number of discrete wavelength points.

For a model atmosphere with 100 layers and 20 000 wavelength points, the operator takes up ≈4.5 GB of memory. This easily exceeds the average memory per processor available in modern computing systems. The complete operator must be kept in memory for the solution of the ALI step if direct solvers such as the LAPACK package are used. The need to store the factorisation actually doubles the memory requirements. However, this is not the case if an iterative method of solution is used for the ALI step. Then different tasks can work on different parts of the index field that are iterated. This in turn means that only those parts of the operator that are needed for the local iteration of the new mean intensities need to exist in each task. The storage requirements are then reduced by a factor equal to the number of tasks involved in the computation.

A drawback of this strategy of decomposing the operator is that it greatly increases the need for communication between tasks. The formal solution and its accompanying data are parallelised over the characteristics of the radiation field (see Sect. 3.3). A Λ^*-operator element is influenced by all characteristics and therefore contributions to the element that is stored for just one process must be calculated by all processes and be communicated. A parallelised iterative solution of the mean intensity in the ALO step is also enforced, increasing the need for communication even more (see Sect. 3.4).

3.2. Optimisation of the speed of the formal solution

To improve the overall computational speed, the time take to complete a formal solution must be decreased because it is performed most often in an ALI solution. In small-scale applications, the SuperLU package (Demmel et al. 1999) provided an efficient solver for the matrix equation of the formal solution along a characteristic. However, there was room for improvement, e.g., in terms of memory footprint, so we developed an iterative Gauss-Seidel (GS) type solver (Golub & Van Loan 1989), which proved to have a minimal memory footprint as well as to be very fast. The main advantage of this new solver is, however, that for a linearly interpolated source function $\hat{S}$, the formal solution, becomes quasi-analytic and there is no need at all to solve a matrix equation.

In principle, our method is a standard GS type iterative solver that uses a physically guided index field stepping scheme. That means that we use knowledge of the physics along the characteristic to increase the convergence of the iteration by a huge factor. The GS method does not prescribe the order in which the elements of the system are iterated. Therefore, we can freely choose the order of the steps in the solution of the linear system. We choose it so that we follow the motion of a pulse along the characteristic, hence follow the physical flow of information in the system. Because we know that the information will always be propagated along the characteristic, we are left only with the task of determining whether the information flows to longer or shorter wavelengths at any given spatial point. This problem has been already solved in the construction of the approximate Λ-operator and its solution can be reused here.

To clarify the process of stepping, the flow of information along a characteristic for alternating wavelength couplings is shown schematically in Fig. 2. The arrows mark the direction – in space and in wavelength – in which the information flows along a characteristic at wavelength λ. It is obvious that the information flows only along the characteristic and along the wavelength derivative sense determined by the sign of a_l.

In the following, the intensities of the nth iteration are written with an additional superscript (n). The iteration step for a specific intensity at a point i and wavelength l of a characteristic k can then be written for $a_{i,l}^k \geq 0$ as

$$I_{i,l}^{k,(n+1)} = \left(1 - B_{i,l}^{\diagdown,k}\right)^{-1} \cdot \left(\Delta I_{i,l}^k + B_{i,l}^{-,k} I_{i,l-1}^{k,(n+1)} + A_{i,l}^{-,k} I_{i-1,l-1}^{k,(n+1)} + A_{i,l}^{\diagdown,k} I_{i-1,l}^{k,(n+1)} + C_{i,l}^{-,k} I_{i+1,l-1}^{k,(n)}\right) \quad (25)$$

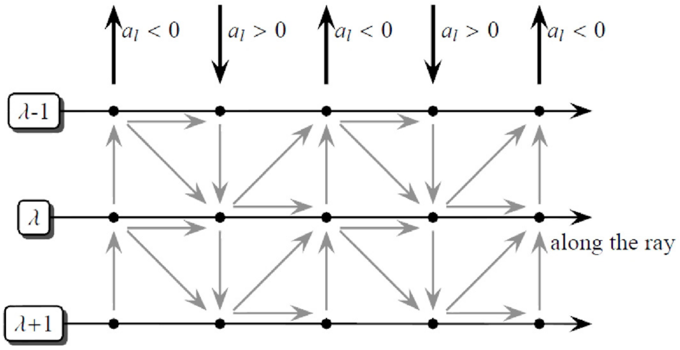

Fig. 2. Flow of information along a characteristic and wavelength for alternating wavelength couplings.

and for $a_{i,l}^k < 0$ as

$$I_{i,l}^{k,(n+1)} = \left(1 - B_{i,l}^{\diagdown,k}\right)^{-1} \cdot \left(\Delta I_{i,l}^k + B_{i,l}^{+,k} I_{i,l+1}^{k,(n+1)} + A_{i,l}^{\diagdown,k} I_{i-1,l}^{k,(n+1)} + A_{i,l}^{+,k} I_{i-1,l+1}^{k,(n+1)} + C_{i,l}^{+,k} I_{i+1,l+1}^{k,(n)}\right), \quad (26)$$

where all coefficients that vanish have been omitted for the given sign of a_l.

From Eqs. (25)–(26), it can be seen that in the case of linear interpolation of the $\hat{S}_l$ source function, the scheme becomes independent of elements with the iteration order (n) and is therefore quasi-analytic since it depends only on elements with the same iteration order.

In explicit form, the formal solution is given for $a_{i,l}^k \geq 0$ by:

$$\left(1 - B_{i,l}^{\diagdown,k}\right) I_{i,l}^{k,(n+1)} = \left(\Delta I_{i,l}^k + B_{i,l}^{-,k} I_{i,l-1}^{k,(n+1)} + A_{i,l}^{-,k} I_{i-1,l-1}^{k,(n+1)} + A_{i,l}^{\diagdown,k} I_{i-1,l}^{k,(n+1)}\right) \quad (27)$$

and for $a_{i,l}^k < 0$ by

$$\left(1 - B_{i,l}^{\diagdown,k}\right) I_{i,l}^{k,(n+1)} = \left(\Delta I_{i,l}^k + B_{i,l}^{+,k} I_{i,l+1}^{k,(n+1)} + A_{i,l}^{\diagdown,k} I_{i-1,l}^{k,(n+1)} + A_{i,l}^{+,k} I_{i-1,l+1}^{k,(n+1)}\right). \quad (28)$$

The formal solution remains a boundary condition problem, but it can be solved as an initial value problem at every spatial point. The solution is then direct and its speed is optimal.

3.3. Calculation of the formal solution matrix

All non-zero entries in the Matrix A – also called characteristic data in the following – from Eq. (12) must be known before the formal solution can be calculated. In the previous algorithm, the calculation of this characteristic data was parallelised over wavelength to ensure optimal scaling with the number of processors (see Fig. 3 in Baron & Hauschildt 2004). The problem with this approach is that every process also calculates data it does not need. Furthermore, the data was written to disk to allow other processes to access the data in case it was needed. This was developed to allow for small memory demands since the data for a characteristic could be loaded just before the calculation and deleted afterwards. This strategy proved to be troublesome for large numbers of wavelength points because the I/O performance was the limiting factor in the calculations. In non-parallel file systems, the simultaneous writing and reading of the data files also proved to be a severe bottle neck. The severity of this problem can be reduced if a server process does all the I/O and distributes/receives the data to/from the client processes.

Optimal performance was achieved if the setup for the calculation of characteristic data was parallelised over characteristics instead. Every process then calculates the data it will need and stores it directly in memory. This completely removes the need for I/O and increases the speed by a large factor. Concerns about load balancing proved to be unfounded. The calculation of the necessary data is so fast that the inbalance in the load is not a factor. This setup has the disadvantage that for large-scale applications the calculations cannot be performed on a small number of processors. Because the data files on disk are absent, the data must be kept in memory at all times and thus a larger number of processors is needed to perform the domain decomposition effectively. In practice, this is not a problem since the number of processors is normally chosen to be large to reduce the overall computing time anyway.

3.4. Iterative solution of the ALO step

As described in Sect. 3.1, the domain decomposition of the ALO reduces the memory requirements. However, one then must use an iterative solution for the ALO system. The convergence of iterative solvers is also likely to be very good because the source function in the later iterations will already be close to the final solution, whereas direct solutions cannot take advantage of this. Further iterative solvers are well suited to keeping the memory footprint small because they do not have to keep additional data in memory besides the linear system.

We implemented GS and Jacobi type solvers to solve the linear system of the ALI step:

$$(1 - \epsilon\,\Lambda^*)\,J = J^{\mathrm{FS}} - \epsilon\,\Lambda^*\,J^{\mathrm{old}} \tag{29}$$

where J is the mean intensity and $\epsilon = \frac{\sigma}{\chi}$, where σ is the scattering part of the opacity χ. The mean intensity from the formal solution is J^{FS} and J^{old} is the result from the previous ALO step. The Λ^*-operator is tridiagonal in wavelength, but has the full spatial bandwidth and its elements are identical to the corresponding Λ-operator elements. The three different bands in wavelength are called Λ^*_-,$\Lambda^*_\diagdown$, and Λ^*_+. An example of Λ^* in Eq. (29) at the wavelength l can then be written as:

$$\Lambda^*\,[J_l] = \Lambda^*_- J_{l+1} + \Lambda^*_\diagdown J_l + \Lambda^*_+ J_{l-1}. \tag{30}$$

Equation (29) can then be rearranged into the following form

$$\begin{aligned} J_{m,l} \;=\; & \left(1 - \epsilon_{m,l}\Lambda^*_{\diagdown,m,m,l}\right)^{-1} \cdot \Bigg(J^{\mathrm{FS}}_{m,l} - \sum_n \epsilon_{n,l} J^{\mathrm{old}}_{n,l} \Lambda^*_{\diagdown,m,n,l} \\ & - \sum_n \epsilon_{n,l+1} J^{\mathrm{old}}_{n,l+1} \Lambda^*_{-,m,n,l+1} - \sum_n \epsilon_{n,l-1} J^{\mathrm{old}}_{n,l-1} \Lambda^*_{+,m,n,l-1} \\ & + \sum_{n\neq m} \epsilon_{n,l} J_{n,l} \Lambda^*_{\diagdown,m,n,l} + \sum_n \epsilon_{n,l-1} J_{n,l-1} \Lambda^*_{+,m,n,l-1} \\ & + \sum_n \epsilon_{n,l+1} J_{n,l+1} \Lambda^*_{-,m,n,l+1} \Bigg), \end{aligned} \tag{31}$$

where n and m are indices for the number of layers. Equation (31) can be readily used in the GS and Jacobi iteration schemes.

Because the mean intensity at a wavelength l depends on quantities at the wavelengths $l-1$, l, and $l+1$, the domain decomposition of the ALO in wavelength must be performed blockwise for the GS method to be applicable. These blocks must overlap with one wavelength point at the boundaries to minimise communication.

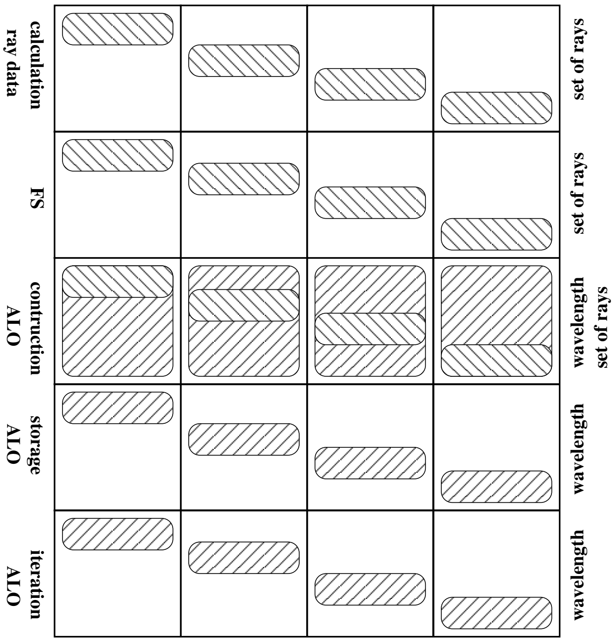

Fig. 3. Schematic overview of the parallelisation strategy for four processes. Backward-shaded areas (\) indicate characteristic-dependent data and forward-shaded areas (/) wavelength-dependent data.

3.5. Summary of the optimal parallelisation

The parallelisation strategy is the key element for the computation of large-scale applications in the framework of radiative transfer with arbitrary wavelength couplings. As a summary, the most important aspects are schematically shown in Fig. 3.

4. Test calculations

We present the results of test calculations that we performed to test the new algorithm in terms of speed, scalability, and by regression testing.

In the following, model calculations have about 20 000 wavelength points and 64 layers unless noted otherwise. This setup would have been impossible with the old algorithm for radiative transfer as the memory of an average computing node would have been by far exceeded. Unless noted otherwise, the computations were performed on 1800 MHz AMD Opteron 244 CPUs with 4 GB RAM per CPU.

4.1. Comparison to old algorithm

We compare the new radiative-transfer algorithm with the methods described in Baron & Hauschildt (2004). We show that the improvements to the formal solution and the ALO step solver described in Sect. 3, are significant.

4.1.1. Formal solution

We compare the speed of two solvers of the formal solution: the SuperLU solver package and our quasi-analytic solution (see Sect. 3.2). The comparison is unfair, because the LU decomposition cannot take advantage of the special character of the matrix. However, the comparison demonstrates clearly that the quasi-analytic solution is needed to calculate large-scale applications in the given framework, since even fast and sophisticated solvers

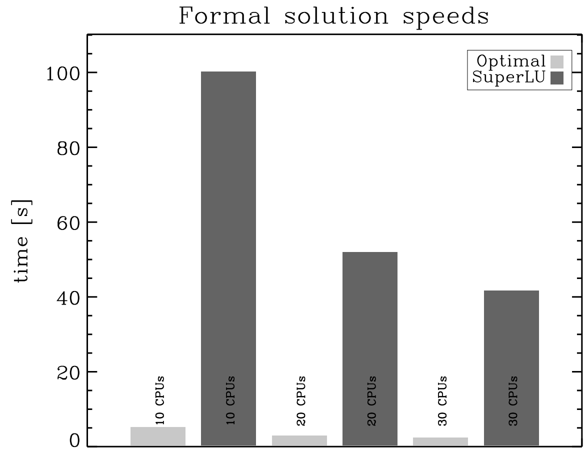

Fig. 4. Comparison of the mean times needed for the complete formal solution using the optimal algorithm and SuperLU for different numbers of processors.

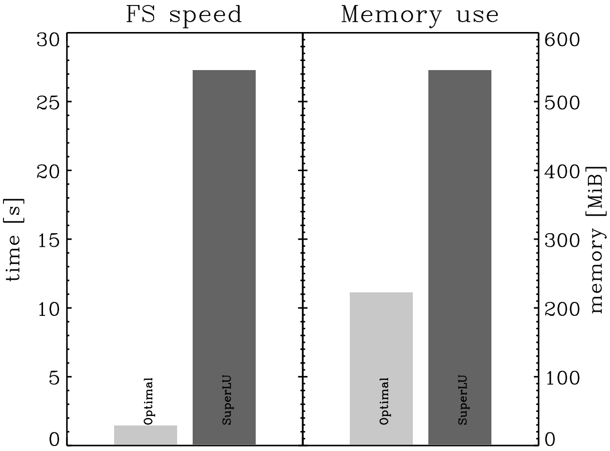

Fig. 5. Comparison of the mean time (*left*) as well as the memory footprint (*right*) needed for the formal solution of the longest characteristic using the optimal algorithm and SuperLU.

such as SuperLU are not fast enough to ensure that the calculation is practically feasible.

In Fig. 4, the mean time needed for a formal solution is shown for the two solvers and different numbers of processors. It is obvious that the optimal solution outperforms the SuperLU package. The SuperLU solution benefits well from an increase in the number of processors. This is also true for the optimal solution, but the effect is not as dramatic because the times are already very short (see Sect. 4.2.3 for the scalability of the formal solution).

In Fig. 5, the time and memory consumption comparison for the different solvers is shown for the formal solution along the longest characteristic in the atmosphere. This allows a more direct comparison of the solutions. The speed advantage of the optimal solution is again obvious. The memory footprint of the optimal solution is also less than half as large as for SuperLU.

The possibility of an analytic solution and the resulting fast formal solution open up the possibility of calculating radiative transfer in multiple, spatial dimensions with a characteristic method in the Lagrangian frame, while allowing non-monotonic velocity fields or other arbitrary wavelength couplings. If the speed of the formal solution were not optimal, the vast number of characteristics would increase the computation time beyond feasibility.

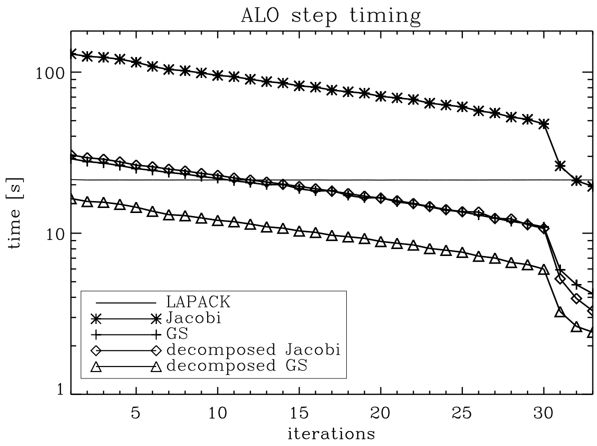

Fig. 6. Comparison of the iteration times of the approximate Λ-iteration for different algorithms.

4.1.2. Iterative ALO step

In Fig. 6, we compare the iteration speeds of the ALI for different algorithms[1]. The LAPACK solver has a special role here, since it was the method of choice in the previous algorithm. As a direct solver the speed is always the same because it cannot take advantage of the benefits provided by a source function that is already close to the solution.

The opposite is true for the iterative solvers. Here we show the results for Jacobi and GS solvers in serial mode as well as in parallelised versions. The serial version of the Jacobi solver is significantly slower than the direct solution. It becomes comparable in efficiency with the LAPACK solver in the last few iterations because it makes use of the convergence of the source function.

The serial GS and the parallelised Jacobi solver have similar speeds in the given example. For the first iterations, they are slower but after about one third of the iterations they become faster than the LAPACK solver. This results in an better overall superior performance for the complete ALI.

The parallelised GS solver is even faster than the LAPACK solver from the beginning and provides the best performance of all solvers, while keeping the memory requirements to a minimum.

4.2. Scalability of the new algorithm

By considering scalability, we describe the increased speed and reduced memory footprint of the algorithm when the strategies from Sect. 3 are applied to different numbers of processors.

In the following, we present the results for the domain decomposition of the ALO, the ALO construction times, the formal solution speeds, and the ALO iteration speeds.

[1] The number of wavelength points had to be reduced to about 10 000 in this example due to the increased memory footprint of the LAPACK solver. Otherwise the calculation would not have been possible on the available hardware. Furthermore, this test was performed on a different computer (HLRN-I) than the other calculations. Hence, direct speed comparisons with other results in this work are invalid.

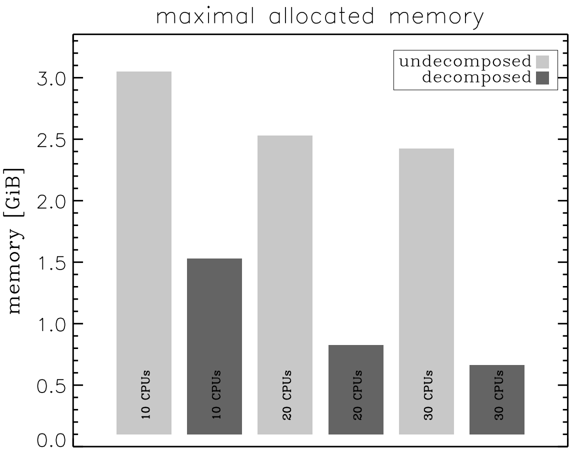

Fig. 7. Comparison of the maximum memory used between decomposed and non-decomposed algorithms of the radiative transfer for 10, 20, and 30 processors.

4.2.1. Domain decomposition of the ALO

Here we show the drastic effect that the domain decomposition of the ALO has on the overall memory consumption.

In Fig. 7, the maximal allocated memory is shown for algorithms with and without a decomposed ALO for 10, 20, and 30 processors. The maximum memory allocated not only includes the ALO but also all other data, such as the formal solution data and the opacities.

For the non-decomposed as well as decomposed algorithms, the memory footprint reduces when the number of processors is increased. In the non-decomposed case, this is only caused by the reduction in the characteristics data that must be kept in memory. In the decomposed case, the smaller amount of ALO data that must be stored further decreases the memory usage. It is obvious that the decomposed algorithm can reduce the memory requirements sufficiently for it to be used on average current hardware.

4.2.2. Construction of the ALO

One of the drawbacks of a decomposed ALO is its distributed construction (see Sect. 3.1). Figure 8 shows how the construction time of the decomposed ALO compares with the construction time of the non-decomposed ALO for different numbers of processors.

Because the decomposed ALO parts are not calculated by one task alone but from all formal solution tasks, the construction time increases. However, the more formal solution tasks present, the faster the construction becomes. This can also be seen in Fig. 8. For a highly parallelised calculation with 30 processors, the decomposed construction is only ≈1.4 times slower than in the non-decomposed case. Since the construction of the ALO must be performed only once during a full ALI, this drawback is not significant in the light of the smaller memory footprint that the decomposition provides.

4.2.3. Formal solution

The formal solution is the routine that is called the most often during a full ALI solution. Hence, it is especially important that it be as fast as possible. In Fig. 9, the mean times for a full optimal formal solution are shown for different numbers of processors.

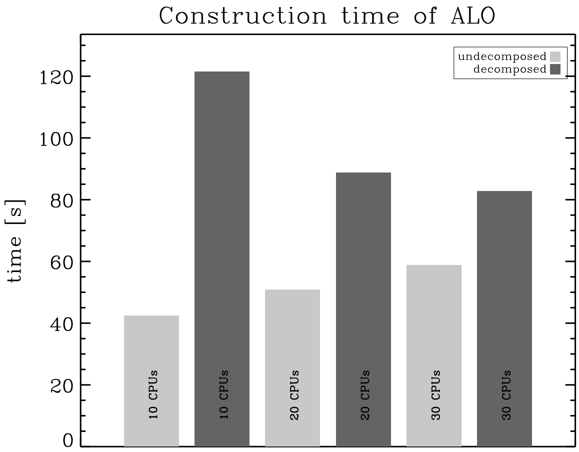

Fig. 8. Comparison of the construction times of the approximate Λ-operator between non-decomposed and decomposed setups.

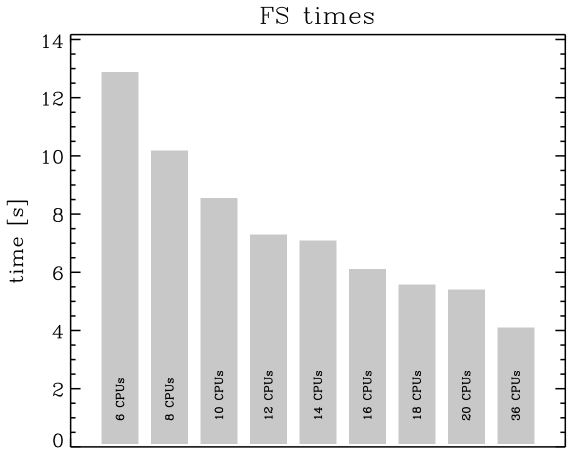

Fig. 9. Overview of the timings for the formal solution for different numbers of processors.

The formal solution speed scales with the number of processors. However, the scaling is not linear. The total time needed for a complete formal solution is limited by the time of the process with the largest characteristics set involved in the computation.

The overall speed of the local formal solutions is determined by the number of characteristics and the total number of spatial points along characteristics that a task must handle. It follows that an increase in the number of processors produces a significant increase in speed only if the number of characteristics on the slowest task is reduced.

The formal solution achieves its optimal speed when there are at least as many tasks as there are characteristics in the system.

4.2.4. ALO iteration speeds

After the formal solution, the ALO step is the next most time-consuming part of the calculation. In Fig. 10, the iteration times for a full ALI are shown for 6, 12, 18, and 36 processors. In these calculations, the parallelised GS-type solver has been used.

As the source function comes closer to its final value, the iteration speed increases for all calculations. Furthermore, it is clear from Fig. 10 that the overall speed increases with the

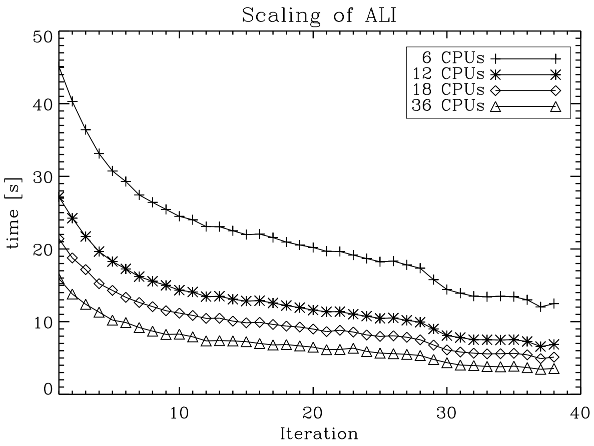

Fig. 10. Comparison of the iteration times of the decomposed parallelised GS type approximate Λ-iteration for a different number of processors.

number of processors, although the scaling is clearly non-linear. Nonetheless, the overall computing time of the ALI increases as the number of processors working on it increases. The asymptotic scaling results from the need for communication between all ALI tasks after each GS step to update the iterated solution.

4.3. Regression test for supernova atmosphere

The formalism and the old algorithm were compared with another solution for the case of monotonic velocity fields in Paper I, although the tests were limited to a toy model. With the new algorithm, we are now in the position to calculate full atmosphere models. Hence, it is instructive to repeat the regression tests in the new framework. As an example of a large-scale application, we compare the spectra from a supernova atmosphere calculation with 100 layers and about 20 000 wavelength points. As a reference for the solution of the radiative transfer in a monotonic velocity-field we used the well tested and established algorithm described in Hauschildt (1992).

The resulting comoving spectra from the old and new algorithm differ for only about 20 wavelength points in the first ten leading digits and the maximal relative differences are of the order of 10^{-6}.

This magnitude of difference is at first unexpected, since both radiative-transfer algorithms use an internal relative convergence criterion of 10^{-8}. To understand the remaining differences, we recall that the iteration procedure differs substantially between the two algorithms. In the recursive solution, the transfer is solved wavelength by wavelength and the ALI step will stop as soon as the internal criterion is reached. This is not true for the matrix-based solution, in which all wavelength points are iterated at the same time. This means that the ALI will continue to iterate the solution also at those wavelength points, which are already internally converged.

More importantly, the solution from the previous wavelength point is fixed in the recursive scheme and accordingly the influence of the wavelength coupling does not change in the solution of the radiative transfer at any given wavelength point. The opposite is again true for the matrix-based solution, since the solution is obtained for all wavelength points simultaneously. This means that the solution converges more successfully at most wavelength points in the matrix-based solution, since the solution is consistent with the convergence criterion for all wavelengths at the same time.

Taking the points above into account, the overall match of the two solutions is excellent and no residuals were found in our test calculations.

5. Conclusion

We have presented algorithm strategies and details of the solution of the radiative-transfer problem in atmospheres with arbitrary wavelength couplings that are suited to the treatment of large-scale applications. The main aim of the optimisations of the existing framework has been the reduction of the memory usage to make the calculations feasible on currently available hardware. This has been achieved by domain decomposition of the data and parallelised code execution. In addition, the speed of the formal solution, the calculation of its matrix elements, and the ALI have all been vastly improved. The speed of all new algorithms scales with the number of processors used in the calculations. Although the scaling is non-linear, the overall computation time is still significantly reduced by an increase in the number of processors.

We are now in a position to calculate large-scale model atmospheres that include alternating wavelength couplings – as from non-monotonic velocity fields or general-relativistic wavelength shifts.

Future possible applications are the velocity profiles of cool stellar winds, the treatment of partial redistribution, and the calculation of radiative transfer in shock fronts as in accretion shocks. However, most promising are the prospects for the transition of radiative transfer to multiple spatial dimensions. Because of the good scaling of the formal solution and the ALO step with the number of processors, the algorithm strategy can be reused for 3D calculations.

Acknowledgements. This work was supported in part by SFB 676 from the DFG, NASA grant NAG5-12127, NSF grant AST-0707704, and US DOE Grant DE-FG02-07ER41517. This research used resources of the National Energy Research Scientific Computing Center (NERSC), which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231; and the Höchstleistungs Rechenzentrum Nord (HLRN). We thank all these institutions for a generous allocation of computer time.

References

Baron, E., & Hauschildt, P. H. 2004, A&A, 427, 987
Baron, E., & Hauschildt, P. H. 2007, A&A, 468, 255
Cannon, C. J. 1973, ApJ, 185, 621
Demmel, J. W., Eisenstat, S. C., Gilbert, J. R., Li, X. S., & Liu, J. W. H. 1999, SIAM J. Matrix Analysis and Applications, 20, 720
Golub, G. H., & Van Loan, C. F. 1989, Matrix computations (Baltimore: Johns Hopkins University Press)
Hauschildt, P. H. 1992, J. Quant. Spectr. Rad. Trans., 47, 433
Hauschildt, P. H., & Baron, E. 2004, A&A, 417, 317
Hauschildt, P. H., & Baron, E. 2006, A&A, 451, 273
Hauschildt, P. H., & Baron, E. 2008, A&A, 490, 873
Knop, S., Hauschildt, P. H., & Baron, E. 2008, A&A, in press
Mihalas, D. 1980, ApJ, 237, 574
Olson, G., & Kunasz, P. 1987, J. Quant. Spectr. Rad. Trans., 38, 325
Scharmer, G. B. 1981, ApJ, 249, 720