ASTR 5463: Accelerated Lambda Iteration

Manuel Barrientos, Anthony Burrow, Adam Moss, Sarah Stangl

1 Introduction

One of the most fundamental problems in stellar atmospheres is being able to solve two equations simultaneously, the radiative transfer equation and the statistical equilibrium equation (e.g., Olson & Kunasz, 1987; Hubeny, 2003). The first one is given for the plane-parallel case as

$$\mu \frac{dI_\nu(\mu,\tau)}{d\tau_\nu} = I_\nu(\mu,\tau_\nu) - S_\nu(\tau_\nu), \tag{1}$$

where $I_\nu(\mu,\tau_\nu)$ is the specific intensity and S_ν the source function. The former depends on the directional cosine μ, the monochromatic optical depth τ_ν, and both quantities depend on the frequency ν. Because of the complicated coupling of the quantities in this equation, the formal solution to this integral-differential equation must be solved numerically. The first-moment equation may be written as

$$J_\nu = \Lambda_\nu S_\nu, \tag{2}$$

with Λ_ν being a wavelength-dependent operator that may act on $S(\tau)$, and

$$J_\nu(\tau) = \frac{1}{2}\int_{-1}^{1} I_\nu(\mu,\tau_\nu) d\mu \tag{3}$$

is the mean intensity. The second one can be expressed as the source function

$$S_\nu = (1-\epsilon)J_\nu + \epsilon B_\nu, \tag{4}$$

where ϵ is the thermalization (also known as the collisional destruction probability), and B_ν is the Planck function. Combining Equations 2 and 4, this may be written as

$$S_\nu = (1-\epsilon)\Lambda_\nu S_\nu + \epsilon B_\nu, \tag{5}$$

or in other words,

$$S_\nu = [1-(1-\epsilon)\Lambda_\nu]^{-1}\epsilon B_\nu, \tag{6}$$

However, because this equation must be solved numerically, and Λ_ν is typically expressible as an $N \times N$ matrix, where N is the number of points on the τ grid used, inverting this matrix quickly becomes computationally expensive. It is therefore important to explore faster and more efficient methods of solving the formal solution.

The formal solution can instead be solved using an iterative method by converging J_ν. By iteratively applying the lambda operator to the source function S_ν, the approximation for J_ν is improved with each iteration. This lambda iteration may be written as

$$S_\nu^{(n+1)} = (1-\epsilon)\Lambda_\nu S_\nu^{(n)} + \epsilon B_\nu. \tag{7}$$

for some n^{th} iteration of calculating S_ν from J_ν. This approach avoids the need to invert Λ_ν. However, for cases of highly non-local thermal equilibrium (non-LTE or NLTE) where $\epsilon \ll 1$, the convergence time is slow (see, e.g., Mihalas, 1978). In this case, the relative changes in the source function become extremely small long before the correct solution is reached (see Hubeny, 2003).

Because of this, a more efficient technique for solving this equation is needed. In particular, accelerated lambda iteration (ALI) has been put into context of the short-characteristics method (see subsection 2.1) by Olson & Kunasz (1987), which will henceforth be referred to as O&K. As explained by O&K, one may consider introducing a perturbation operator Λ_ν^* that contains information about the dominant physics of line-core diffusion. We can then define the exact lambda operator as

$$\Lambda_\nu = \Lambda_\nu^* + (\Lambda_\nu - \Lambda_\nu^*). \tag{8}$$

From here one may create a new iteration method by attributing the first term in this lambda operator to the $(n+1)^{\text{th}}$ iteration, and the other terms to the previous n^{th} iteration; in other words,

$$S_\nu^{(n+1)} = (1-\epsilon)\Lambda_\nu^* S_\nu^{(n+1)} + (1-\epsilon)(\Lambda_\nu - \Lambda_\nu^*) S_\nu^{(n)} + \epsilon B_\nu, \tag{9}$$

which becomes

$$[1-(1-\epsilon)\Lambda_\nu^*] S_\nu^{(n+1)} = (1-\epsilon)(\Lambda_\nu - \Lambda_\nu^*) S_\nu^{(n)} + \epsilon B_\nu. \tag{10}$$

One may also write this in terms of J_ν as

$$J_\nu^{(n+1)} - J_\nu^{(n)} = [1-(1-\epsilon)\Lambda_\nu^*]^{-1} (J_\nu^{FS,(n)} - J_\nu^{(n)}) \tag{11}$$

where $J^{FS,(n)} = \Lambda_\nu S^{(n)}$. This may then be used with equation 4 to complete a full ALI cycle.

The advantage of this new iteration scheme is that, when an appropriate perturbation Λ_ν^* is chosen, such as a diagonal or multi-banded diagonal matrix, we may perform the matrix inversion quickly, with $\mathcal{O}(n)$ time, rather than near $\mathcal{O}(n^3)$ time for typical matrix inversion. At the same time, the convergence procedure is accelerated because introducing the perturbation matrix reduces the eigenvalues of the lambda matrix. Typically this perturbation matrix is chosen to be the diagonals, tri-diagonals, or higher-order banded diagonals of the lambda operator.

In this project, we aim to solve the radiative transfer equation and simultaneously scattering problem for a generally NLTE plane-parallel atmosphere. We specifically perform ALI to do so, and we also add further optimizations to this method, which are described in section 2 along with assumptions we make to conquer this problem. In section 3, results of our work are shown, and we discuss what these results suggest.

2 Methods

2.1 Numerical Approach

To perform ALI, we first use the short-characteristics method of solving the formal solution given by O&K. This method solves the set of equations

$$\begin{aligned} I_\nu^+(\tau_\nu,\mu) &= I_\nu^+(T_\nu,\mu)e^{-(T_\nu-\tau_\nu)/\mu} + \int_{\tau_\nu}^{T_\nu} S_\nu(t)e^{-(t-\tau_\nu)/\mu}dt/\mu, \\ I_\nu^-(\tau_\nu,\mu) &= I_\nu^+(0,\mu)e^{\tau_\nu/\mu} + \int_0^{\tau_\nu} S_\nu(t)e^{-(\tau_\nu-t)/\mu}dt/(-\mu), \end{aligned} \tag{12}$$

where I_ν^+ and I_ν^- are outward-going and inward-going rays, respectively, and T_ν is the monochromatic slab thickness. To solve this, the short-characteristics method performs integration by interpolating $S_\nu(\tau)$ at each set of two to three adjacent points. This can be written as

$$\begin{aligned} I^+(\tau_i,\mu) &= I^+(\tau_{i+1},\mu)e^{-\Delta\tau_i} + \Delta I^+(S,\mu), \\ I^-(\tau_i,\mu) &= I^-(\tau_{i-1},\mu)e^{-\Delta\tau_{i-1}} + \Delta I^-(S,\mu), \end{aligned} \tag{13}$$

where $\Delta I^\pm$ represents the integral evaluations, and $\Delta\tau_i = (\tau_{i+1} - \tau_i)/|\mu|$. We also have left out the ν subscripts for convenience; note that this process must still be performed for each wavelength point on the chosen grid. This can be written in terms of interpolation coefficients $\alpha_i^\pm$, $\beta_i^\pm$, $\gamma_i^\pm$ as

$$\Delta I_i^\pm = \alpha_i^\pm S_{i-1} + \beta_i^\pm S_i + \gamma_i^\pm S_{i+1}. \tag{14}$$

O&K gives parabolic interpolation coefficients as

$$\begin{aligned} \alpha_i^- &= e_{0,i} + \frac{e_{2,i} - (\Delta\tau_i + 2\Delta\tau_{i-1})e_{1,i}}{\Delta\tau_{i-1}(\Delta\tau_i + \Delta\tau_{i-1})}, \\ \beta_i^- &= \frac{(\Delta\tau_i + \Delta\tau_{i-1})e_{1,i} - e_{2,i}}{\Delta\tau_{i-1}\Delta\tau_i}, \\ \gamma_i^- &= \frac{e_{2,i} - \Delta\tau_{i-1}e_{1,i}}{\Delta\tau_i(\Delta\tau_i + \Delta\tau_{i-1})}, \\ \alpha_i^+ &= \frac{e_{2,i+1} - \Delta\tau_i e_{1,i+1}}{\Delta\tau_{i-1}(\Delta\tau_i + \Delta\tau_{i-1})}, \\ \beta_i^+ &= \frac{(\Delta\tau_i + \Delta\tau_{i-1})e_{1,i+1} - e_{2,i+1}}{\Delta\tau_{i-1}\Delta\tau_i}, \\ \gamma_i^+ &= e_{0,i+1} + \frac{e_{2,i+1} - (\Delta\tau_{i-1} + 2\Delta\tau_i)e_{1,i+1}}{\Delta\tau_i(\Delta\tau_i + \Delta\tau_{i-1})}, \end{aligned} \tag{15}$$

where

$$\begin{aligned} e_{0,i} &= 1 - e^{-\Delta\tau_{i-1}}, \\ e_{1,i} &= \Delta\tau_{i-1} - e_{0,i} \\ e_{2,i} &= (\Delta\tau_{i-1})^2 - 2e_{1,i}. \end{aligned} \tag{16}$$

For the boundaries $i = 1$ and $i = N$, linear interpolation must be performed instead; O&K gives the linear interpolation coefficients as

$$\begin{aligned}
\alpha_i^- &= e_{0,i} - \frac{e_{1,i}}{\Delta\tau_{i-1}}, \\
\beta_i^- &= \frac{e_{1,i}}{\Delta\tau_{i-1}}, \\
\gamma_i^- &= 0, \\
\alpha_i^+ &= 0, \\
\beta_i^+ &= \frac{e_{1,i+1}}{\Delta\tau_i}, \\
\gamma_i^+ &= e_{0,i+1} - \frac{e_{1,i+1}}{\Delta\tau_i}.
\end{aligned} \tag{17}$$

At low $\Delta\tau_i < 10^{-2}$, we also enforce the use of linear interpolation as well to avoid numerical problems. It should also be noted that, to also avoid problems with exponentials at very low $\Delta\tau_i < 10^{-7}$, the $e^{-\Delta\tau_i}$ have have been Taylor-expanded to second order, and $e_{0,i}$, $e_{1,i}$, and $e_{2,i}$ were adjusted accordingly to appropriately estimate our results when we want to resolve τ as low as 10^{-8}.

Now, it is helpful to think of the lambda operator as a matrix that acts on a depth-dependent vector $S(\tau_i)$ to receive the formal solution implicitly through $J(\tau_i)$. We may construct this matrix column-wise, such that the elements of each row are given by

$$\Lambda_{ij} = \frac{1}{2}\int_0^1 d\mu \left[\hat{i}_{i,j}^-(\mu) + \hat{i}_{i,j}^+(\mu)\right], \tag{18}$$

where for a given μ,

$$\begin{aligned}
\hat{i}_{i-1,i}^- &= \gamma_{i-1}^-, \\
\hat{i}_{i,i}^- &= \hat{i}_{i-1,i}^- e^{-\Delta\tau_{i-1}} + \beta_i^-, \\
\hat{i}_{i+1,i}^- &= \hat{i}_{i,i}^- e^{-\Delta\tau_i} + \alpha_{i+1}^-, \\
\hat{i}_{k,i}^- &= \hat{i}_{k-1,i}^- e^{-\Delta\tau_{k-1}} \quad \text{for } k = i+2, i+3, \ldots, N, \\
\hat{i}_{k,i}^- &= 0 \quad \text{otherwise,}
\end{aligned} \tag{19}$$

and

$$\begin{aligned}
\hat{i}_{i+1,i}^+ &= \alpha_{i+1}^+, \\
\hat{i}_{i,i}^+ &= \hat{i}_{i+1,i}^+ e^{-\Delta\tau_i} + \beta_i^+, \\
\hat{i}_{i-1,i}^+ &= \hat{i}_{i,i}^+ e^{-\Delta\tau_{i-1}} + \gamma_{i-1}^+, \\
\hat{i}_{k,i}^+ &= \hat{i}_{k+1,i}^+ e^{-\Delta\tau_k} \quad \text{for } k = i-2, i-3, \ldots, 1, \\
\hat{i}_{k,i}^+ &= 0 \quad \text{otherwise.}
\end{aligned} \tag{20}$$

There is a clear symmetry here between the outgoing and incident rays. Using this form of the lambda matrix along with the vector S_i will ensure that equations 13 are satisfied, such that the formal solution is calculated. As for the integrals used to determine the matrix elements of the lambda operator, we choose μ values based on 32-point Gauss-Legendre quadrature, and sum the values of $\hat{i}$ in a weighted sum. In section 3 we discuss how changes to this integration method alter the result.

Once the formal solution is able to be calculated, one may construct Λ^* and perform iterations of the ALI algorithm. For this work we specifically construct Λ^* using the tridiagonals of the lambda operator, as a tridiagonal matrix equation solution is able to be quickly computed. From here, Equations 11 and 4 may be used to iteratively calculate a converged value of J, which solves the scattering problem, allowing the true formal solution to be solved, which may then be used to determine emergent flux or other desired quantities.

2.2 Initial Conditions

To begin ALI formalism from the previous section, certain initial conditions must first be assumed to begin iteration. First, we assume a grey atmosphere by assuming a $\tau_\nu = \tau$ grid that is only a log-space between some τ_{min} and τ_{max}. We also typically set the first element of this grid to $\tau = 0$, and we replace the closest value to $\tau = 1$ with 1. Once the τ grid is established, Λ is able to be calculated for a given ν. Here this only needs to be done once because we assume the grey case, however our code is written to handle wavelength-dependent values of τ.

We initialize our starting value of S_ν to be $S_\nu = B_\nu$, the Planck function. To do so, we require a temperature profile; we choose for simplicity to have an isothermal atmosphere with $T = T_{\text{eff}}$, where T_{eff} is an input parameter to our model. We also assume that there is no incident intensity at the surface $\tau = 0$ such that $\hat{i}^-_{0,0} = 0$, and at depth the outgoing intensity goes as B_ν, and therefore $\hat{i}^+_{N,N} = 1$. One may also insert an technique such as the diffusion approximation here, however for an isothermal atmosphere the B_ν-derivative would be zero and would be irrelevant. From here, the iteration process described in subsection 2.1 may begin and J may be converged.

2.3 Ng Acceleration

Once four iterations of S have been computed, we may also use Ng acceleration (Ng, 1974) to compute the next S vector. This has been shown to further accelerate the convergence of J, leading to fewer required iterations and more computational efficiency. Because this project is specifically aimed at illustrating ALI, most of our results in section 3 do not include Ng acceleration. However, we do include in section 3 an illustration of Ng acceleration at play. Below we summarize this process.

Previous iterations of the vector $\mathbf{S}$: $\mathbf{S}^{n-1}, \mathbf{S}^{n-2}, \mathbf{S}^{n-3}$ occupy a two-dimensional surface in the linear source function, $\mathbf{S}^n$, defined by

$$\mathbf{S}^n = (1 - a - b)\mathbf{S}^{n-1} + a\mathbf{S}^{n-2} + b\mathbf{S}^{n-3} \tag{21}$$

for constants a and b such that the next iteration $\mathbf{S}^{n+1}$ is also

$$\begin{aligned} \mathbf{S}^{n+1} &= (1-\epsilon)\Lambda\mathbf{S}^n + \epsilon\mathbf{B} \\ &= (1 - a - b)\mathbf{S}^n + a\mathbf{S}^{n-1} + b\mathbf{S}^{n-2}, \end{aligned} \tag{22}$$

where the squared magnitude of the difference between the iterations,

$$|\mathbf{S}^{n+1} - \mathbf{S}^n|^2, \tag{23}$$

is minimized. The a, b values minimizing Equation 23 may be found to be

$$\begin{aligned} a &= \frac{C_1B_2 - C_2B_1}{A_1B_2 - A_2B_1} \\ b &= \frac{C_2A_1 - C_1A_2}{A_1B_2 - A_2B_1}, \end{aligned} \tag{24}$$

where

$$\begin{aligned} A_1 &= Q_1 \cdot Q_1, \quad B_1 = Q_1 \cdot Q_2, \quad C_1 = Q_1 \cdot Q_3 \\ A_2 &= Q_2 \cdot Q_1, \quad B_2 = Q_2 \cdot Q_2, \quad C_2 = Q_2 \cdot Q_3 \end{aligned} \tag{25}$$

and

$$\begin{aligned} Q_1 &= \mathbf{S}^n - 2\mathbf{S}^{n-1} + \mathbf{S}^{n-2} \\ Q_2 &= \mathbf{S}^n - \mathbf{S}^{n-1} - \mathbf{S}^{n-2} + \mathbf{S}^{n-3} \\ Q_3 &= \mathbf{S}^n - \mathbf{S}^{n-1}. \end{aligned} \tag{26}$$

The new $\mathbf{S}^{n+1}$ is given as in Equation 22. This five-iteration process is then able to be repeated until convergence.

3 Results and Discussion

3.1 ALI Results

We have successfully implemented the ALI method to achieve convergence across all values of τ. Here we populate our τ grid with $\tau_{\text{min}} = 10^{-8}$, $\tau_{\text{max}} = 10^6$, and a grid size of 256 points. In general we also let $T_{\text{eff}} = 6000$ K. We use a thermalization of $epsilon = 10^{-4}$ to test our results unless otherwise stated.

Figure 1 shows how the ratio S/B improves with each iteration throughout the atmosphere. At $\tau = 0$, S/B is expected to approach $\sqrt{\epsilon}$ (in this case, 10^{-2}). After one iteration, S/B is quite far from this solution but rapidly approaches the expected value near the surface after more iterations. It should also be noted that expanding the exponentials in the formal solution to second order allowed us to numerically resolve the surface all the way to $\tau = 10^{-8}$. At depth, the LTE result $S = B$ is constant for each iteration, as this was the initial condition for all τ.

We also test the effects of varying ϵ and the number of iterations needed to achieve convergence. Our convergence criterion is that J is said to be converged on the n^{th} iteration if

$$\frac{J_n - J_{n-1}}{(J_n + J_{n-1})/2} < 10^{-8} \tag{27}$$

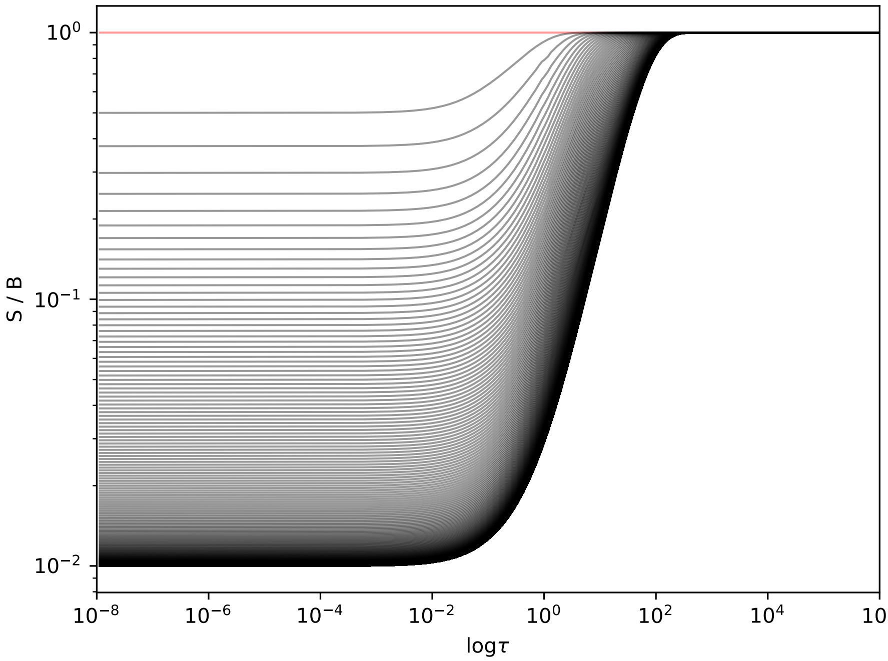

Figure 1: An example convergence plot with our default parameters using ALI. Each gray line represents a new iteration, with 500 iterations in total. At the surface, S/B approaches $\sqrt{\epsilon}$ while at depth, S/B approaches 1.

for all τ. Figure 2 displays these results, with each panel showing a different value of ϵ. The number of iterations required to reach convergence is also noted on each panel. For $\epsilon = 1$, we would expect pure LTE, and it is clear that this holds for every iteration. As ϵ approaches 0 and we deviate from LTE, more iterations are required to achieve convergence.

As a test of the final product of the formal solution, we plot a spectrum generated from our model to ensure we obtain an expected blackbody curve. Figure 3 shows the emergent flux of our model post-convergence at a set of wavelength points between 3000 and 7000 Å. It is clear that we indeed achieve the expected result with no unexpected variation as a function of wavelength.

Since J is the direction-averaged intensity, comparing our J to an analytic value is important as it directly relates to flux from the atmosphere. In Figure 4, we show the percent difference between our calculated J and the J calculated by using the analytic approximation assuming B is linear in τ,

$$J(\tau) = a + b\tau + \frac{(b - \sqrt{3}a)e^{-\sqrt{3\epsilon}}}{(\sqrt{3} + \sqrt{3\epsilon})}, \tag{28}$$

however with $b = 0$ because we assume an isothermal atmosphere. The largest difference between our calculated J and the analytic solution occurs from $10^{-2} < \tau < 1$. This is expected, as this region around $\tau \approx 2/3$ is typically characteristic of the emergent flux of the atmosphere. Elsewhere, there is little divergence between results. We vary ϵ as well for this test, but we do not obtain significant changes as ϵ varies.

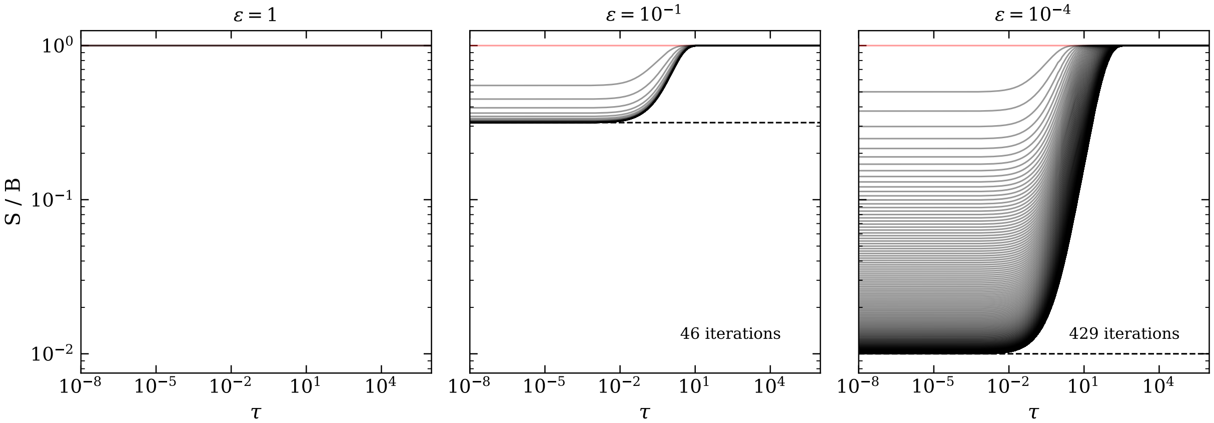

Figure 2: Convergence plots for different values of ϵ. As ϵ approaches 0, more iterations are needed to achieve convergence at the surface.

3.2 Effects of Gauss-Legendre Quadrature

We also investigate the effect of choosing fewer μ values when integrating using Gauss-Legendre quadrature. Ideally one uses as many points as possible for higher degrees of accuracy in the integration to get the formal solution, however this of course increases computation time. In Figure 5 we display the percent difference in S_2, S_4, and S_8, where S_n is the S calculated for n-point quadrature, compared to the 32-point result we typically use, S_{32}. The figure shows that if we use fewer points, our calculated source function deviates significantly in the same $\tau \approx 2/3$ regime.

To test the resulting time it takes to generate a typical model for these different cases, we simply calculate a spectrum similarly to that in subsection 3.1 for 170 wavelength points, which includes the same check for convergence at each step that was described also in subsection 3.1. On average, for the 2-, 4-, 8-, and 32-point methods, these spectra were able to be generated in 3.629, 3.484, 3.583, and 4.238 seconds, respectively. This means that calculations using 2, 4, and 8 points are faster than 32 points by 14%, 18%, and 15%, respectively, but at worst only 9%, 4%, and 2% less accurate. With the same number of iterations, one would expect computation time to increase linearly with the number of points used in integration. With that in mind, the results imply that, when using fewer points, it takes longer to reach the convergence criterion for the same number of iterations when using fewer points.

Although a few seconds for a simple calculation is not much, it is interesting to note this effect for doing radiative transfer on a larger scale, and the possible results that may propagate from the convergence step in the big picture that ALI addresses.

3.3 Effects of Ng Acceleration

Finally, we investigate the effects of Ng acceleration described in subsection 2.3 on the speed of convergence. The left panel in Figure 6 shows the number of iterations required to reach convergence at the surface for $\epsilon = 10^{-4}$ for normal lambda iteration (LI), ALI, and ALI

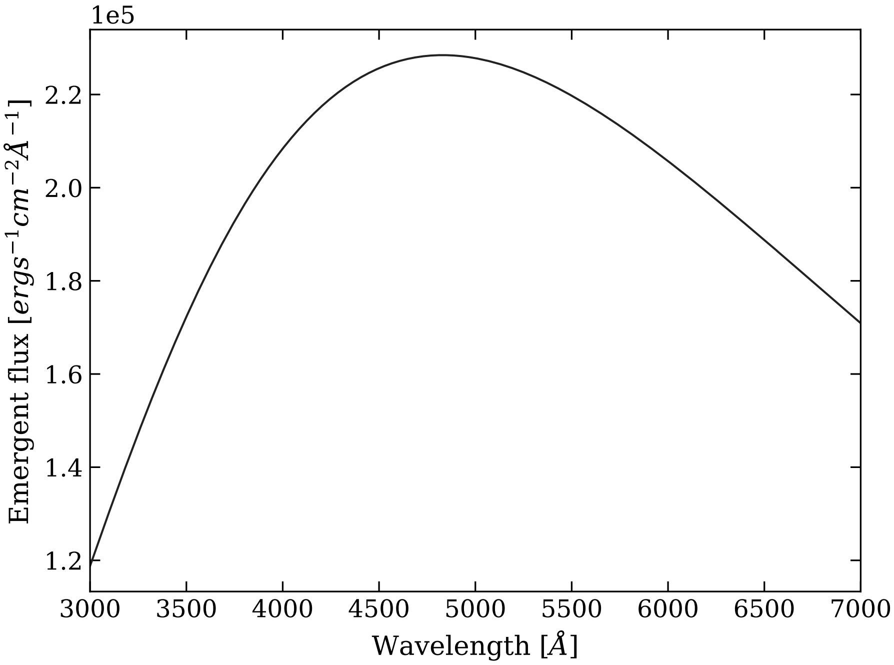

Figure 3: Output spectrum using the default parameters. The line feature at 6564 Å appears, however the shape is incorrect.

with Ng acceleration implemented. As expected, ALI converges much faster than LI, with Ng acceleration further reducing the required number of iterations. The right panel shows the same result in essence, however this is specifically showing the maximum value used to check convergence $\Delta J/\bar{J}$ across all τ. In other words, the dashed line in this right panel at 10^{-8} is our definition of convergence. This gives us an idea of how significantly J itself changes, and it shows again the same sentiment, that Ng acceleration converges in fewer iterations compared to both LI and ALI. LI in particular does not perform well to minimize this quantity, and it would require a drastically higher number iterations to achieve the same result.

In Figure 7, we plot the same convergence test as in Figure 1, however now implementing Ng acceleration. We see that there are large gaps in S/B every time the algorithm described in subsection 2.3 makes a new estimate of S. Clearly this method works to reduce the number of iterations required to fully converge J.

Finally, we perform a similar test as in subsection 3.2, however now to see how quickly J converges and a spectrum may be generated using the different iteration methods. On average, ALI takes 4.222 seconds, and ALI with Ng acceleration takes 1.695 seconds. LI was not included as the number of iterations it would take to converge with $\epsilon = 10^{-4}$ is significantly longer than ALI and frankly is not worth the time spent to compute it.

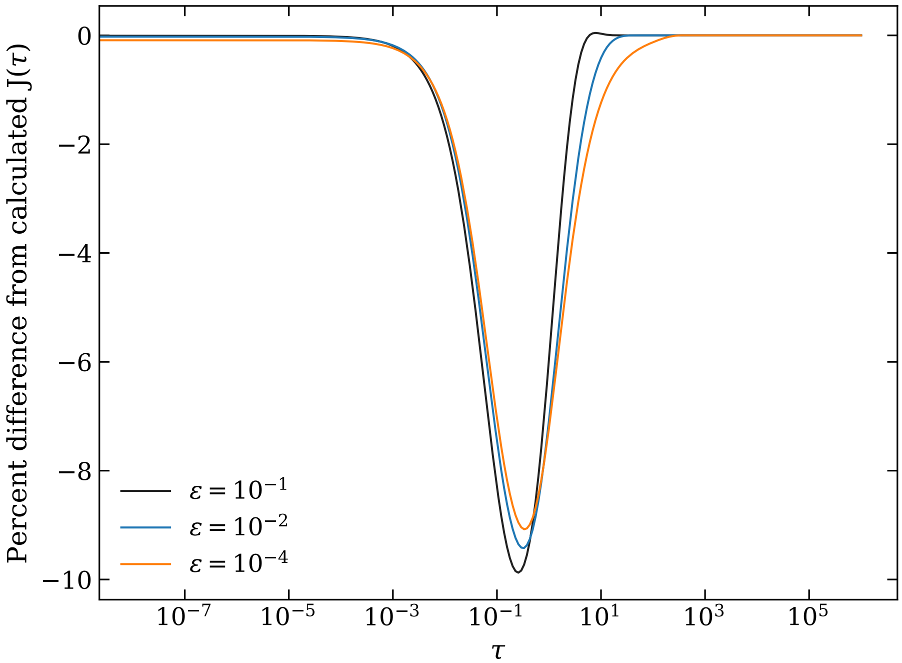

Figure 4: Comparison of our J to the analytic solution. The greatest differences occur around $\tau = 2/3$, where we expect the greatest impact in the emergent flux from the atmosphere. Varying ϵ does not impact J to a significant degree.

4 Conclusion

In this project, we implemented accelerated lambda iteration in order to solve the radiative transfer equation and the NLTE scattering problem. We have tested the results thoroughly for an isothermal, grey atmosphere, which have all had positive results that show our method to be successful in capturing the expected behavior for a solution to the radiative transfer problem. We also show that ALI is able to converge much faster than normal lambda iteration, which is useful for computational efficiency. Finally, we make use of Ng acceleration, and show that such a simple implementation is able to solve such a complicated problem using drastically less time and resources.

Our implementation of this method is available at github.com/anthonyburrow/Rad1D wherein all parameters available for any user to test this method are shown and described.

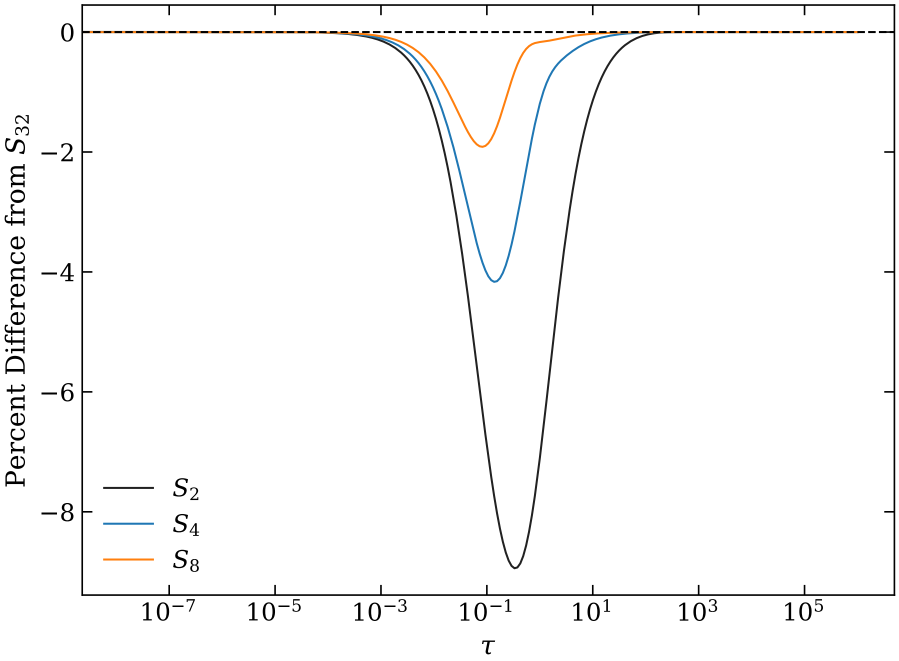

Figure 5: Deviations from 32-point Gaussian quadrature as a function of τ. Similarly to Figure 4, the greatest difference occurs from $0.1 < \tau < 1$.

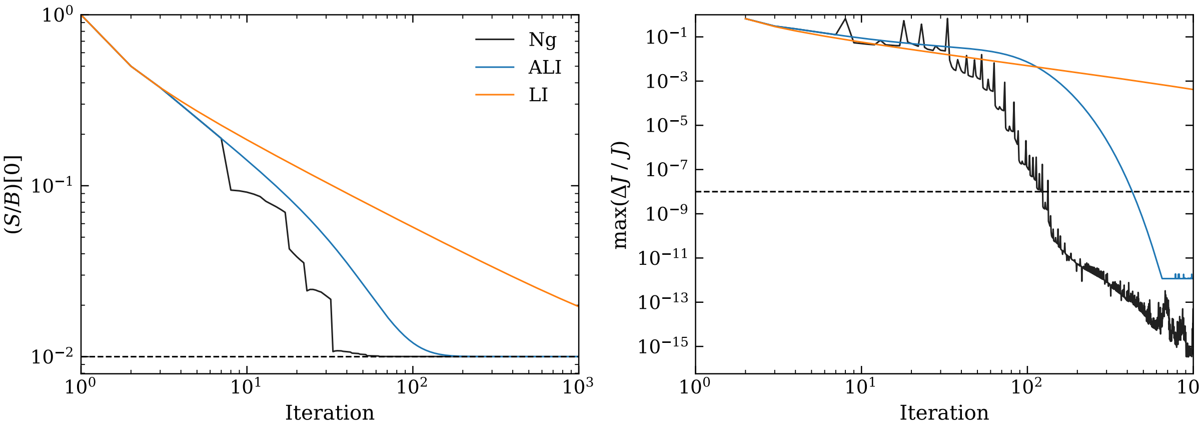

Figure 6: Left: The number of iterations required for S/B to converge at the surface for each method used in our code. Lambda iteration takes significantly longer than ALI as expected. Ng acceleration improves the required number of iterations even further. The jagged pattern is a result of how the algorithm is designed. Right: Variations in J as a function of iteration. Ng acceleration minimizes the change in J faster than ALI, though this does not take effect until about 40 iterations, and then levels off before dropping again at 700 iterations.

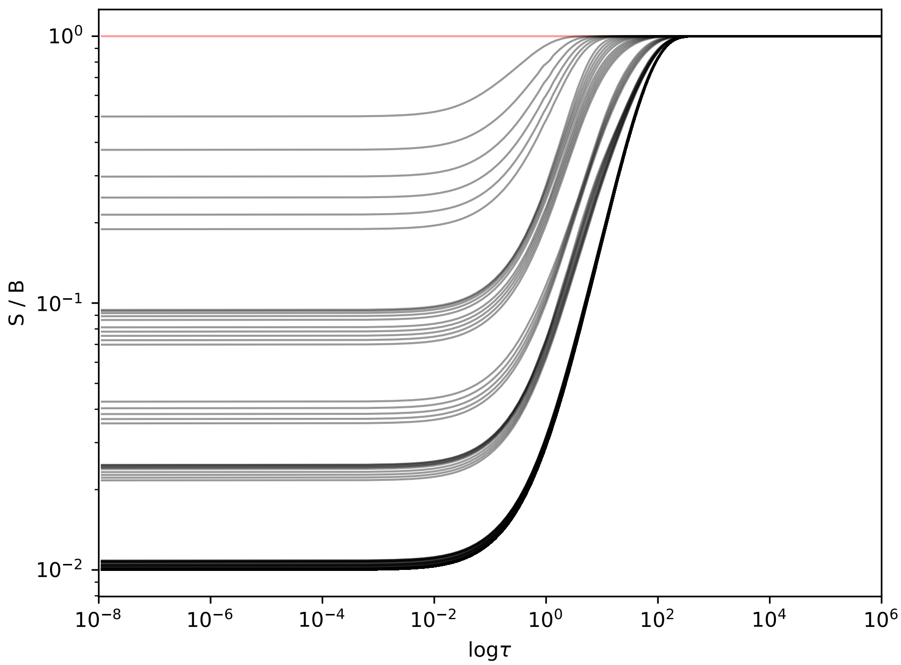

Figure 7: The same convergence plot as Figure 1 but with Ng acceleration implemented. The number of iterations required for convergence decreases significantly.

References

Hubeny, I. 2003, in Astronomical Society of the Pacific Conference Series, Vol. 288, Stellar Atmosphere Modeling, ed. I. Hubeny, D. Mihalas, & K. Werner, 17

Mihalas, D. 1978, Stellar atmospheres

Ng, K. 1974, The Journal of Chemical Physics, 61, 2680–2689, doi: 10.1063/1.1682399

Olson, G. L., & Kunasz, P. B. 1987, , 38, 325, doi: 10.1016/0022-4073(87)90027-6