Stellar Atmospheres
Radiative Transfer Computer Exercises
Solutions

Mandy Hagenaar, Thijs Krijger and Alfred de Wijn

June 11, 2003

Contents

Chapter 1

Getting Started

1.1 VALIII Atmosphere

1.1 `VALIII_READ` restores height and temperature data from `valiii.txt`.

1.2 See figure 1.1.

1.3 The minimum temperature is 4179.08 K.

1.4 The height at the minimum temperature is 491.721 km.

1.5 The temperature of the corona is between 2 kK and 2 MK.

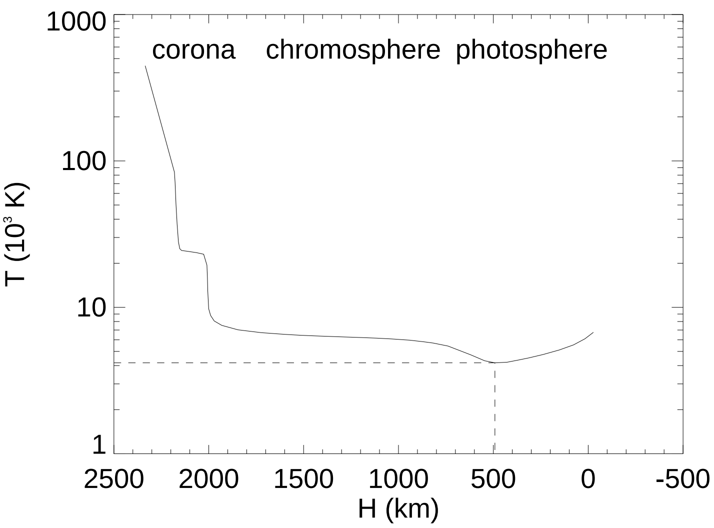

Figure 1.1: Temperature structure of the solar atmosphere.

1.2 Ca II K line

1.6 —

1.7 The dimensions of the image are 250 by 250 pixels.

1.8 —

1.9 The pixelsize is $\tan^{-1}\left(\frac{1550}{150\times10^6}\right) 3600 = 2.13$”.

1.10 In kilometers, the dimensions of the image are $1550 \times 250 = 3.875 \times 10^5$ kilometers square. This equals $3.875 \times 10^5 \frac{360}{2\pi r} = 31.7°$ square.

1.11 See figure 1.2. `n_elements(y0)` yields 1381.

1.12 `n_elements(y1)` yields 2297, because the minimal value has been set to zero.

1.13 The most occoring value is 1991.

1.14 See figure 1.3.

1.15 Supergranules have typical sizes of 20 to 40 Mm.

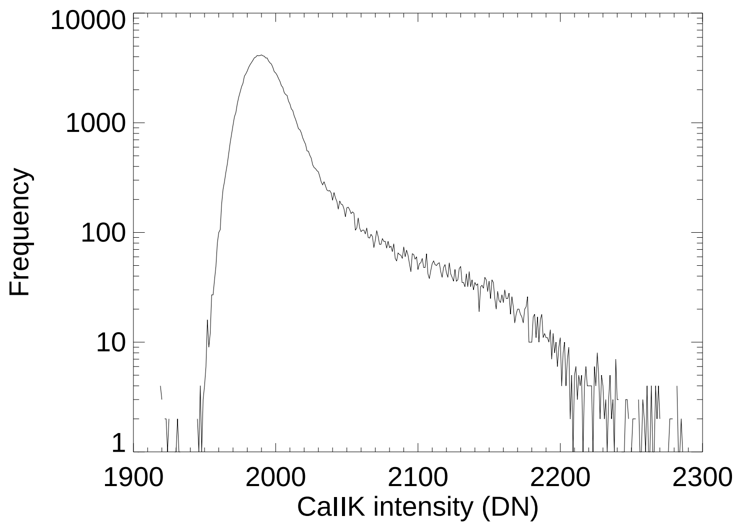

Figure 1.2: Intensity distribution of Ca II K image.

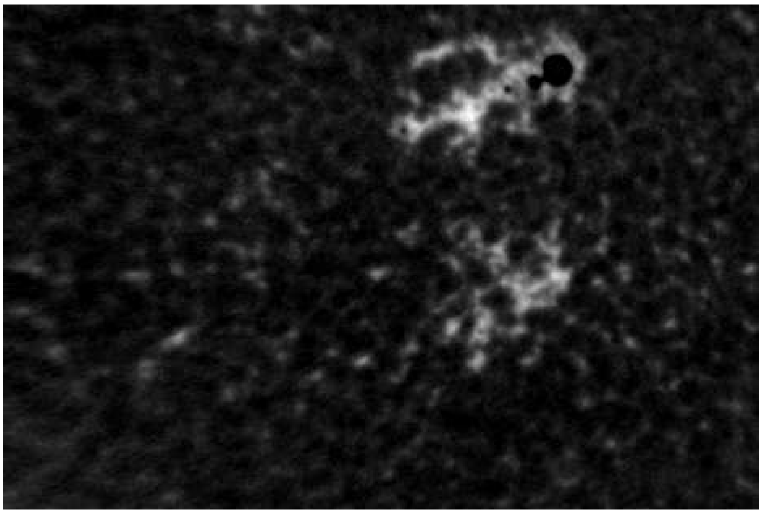

Figure 1.3: The Ca II K image thresholded above 1970.

Chapter 2

The Inversion Problem

2.1 Gray Atmosphere

2.1 `x=findgen(51)/10.-4`, `tau=10^x`.

2.2 `sigma=5.67e5`, `Teff=5770`, and `T=(3./4.*(tau+2./3.))^0.25*Teff`. Be careful that `Teff^4` may be bigger than what can fit in a float. `S=T^4*sigma/!pi`.

2.3 See figure 2.1.

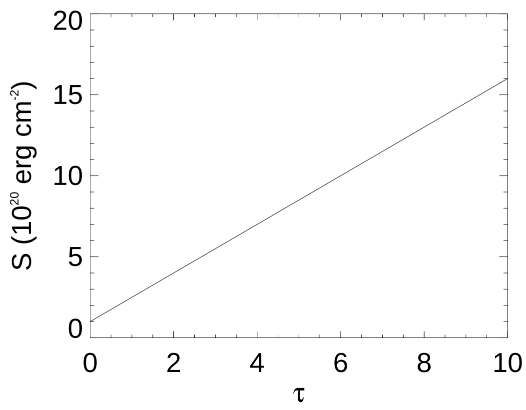

Figure 2.1: Source function as a function of optical depth: $\mathbf{S} = \frac{3\sigma}{4\pi}\left(\tau + \frac{2}{3}\right)$.

2.2 Line Profile

2.4 `y=findgen(21)/2.-5`

2.5 The simple, gaussian profile function plot is omitted.

2.6 You can create `tau2` with loops or matrix-multiplication Matrix multiplication in IDL is done by giving the command `result=A##B`. Define the matrix `tau2=fltarr(51,21)`

so that IDL knows in what type of array it has to put the variable. Fill the array with two loops from 0 to 50 and from 0 to 20, respectively,
`for i=0,50 do for j=0,20 do tau2(i,j)=(10*phi(j)+1)*tau(i)` or
`tau2=(10*phi+1)##tau` (or `tau2=tau#(10*phi+1)`).
See Figure 2.2.

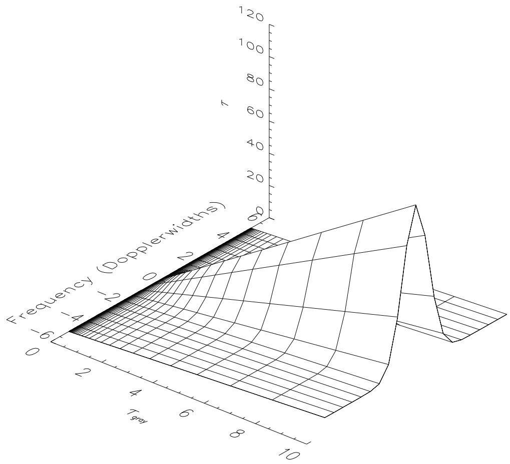

Figure 2.2: Nongray optical depth, as a function of gray optical depth (continuum) and frequency.

2.7 The emerging intensity has the integral form

$$I_\nu(\tau_\nu = 0) = \int_0^\infty S_\nu(t_\nu)\, \mathrm{e}^{-t_\nu}\, \mathrm{d}t_\nu \; . \tag{2.1}$$

2.8 In our example we write this convolution as

$$I_\nu(y) = \int_0^\infty S(\tau(x))\, \mathrm{e}^{-\tau_2(x,y)} \mathrm{d}\tau_2 \; . \tag{2.2}$$

Writing

$$\sum_{\tau_2(x=0,y)}^{\tau_2(x=50,y)} S(\tau(x))\, \mathrm{e}^{\tau_2(x,y)\,\Delta\tau_2(x,y)} = \sum_{x=0}^{x=50} S(\tau(x)) \mathrm{e}^{-\tau_2(x,y)} \log_\mathrm{e} 10\, \tau_2(x,y)\, \delta \; , \tag{2.3}$$

we find

$$\begin{aligned} f(x) &= S(\tau(x)) \; , && (2.4) \\ g(x) &= \mathrm{e}^{-\tau_2(x,y)} \log_\mathrm{e} 10\, \tau_2(x,y)\, \delta \; , && (2.5) \end{aligned}$$

so that

$$I(y) = \sum_0^{50} f(x)\, g(x,y) \; , \tag{2.6}$$

or

$$
\begin{aligned}
I(0) &= g(0,0)f(0) + g(1,0)f(1) + \cdots + g(50,0)f(50) \\
\cdots \\
I(20) &= g(0,20)f(0) + g(1,20)f(1) + \cdots + g(50,20)f(50) \ .
\end{aligned}
$$

2.9 This can be written in a matrix equation

$$
\mathbf{I} = \begin{pmatrix} g(0,0) & \ldots & g(50,0) \\ \vdots & \ddots & \vdots \\ g(0,20) & \ldots & g(50,20) \end{pmatrix} \mathbf{f} = \mathbf{K}\,\mathbf{S} \ .
$$

2.10 See figure 2.3.

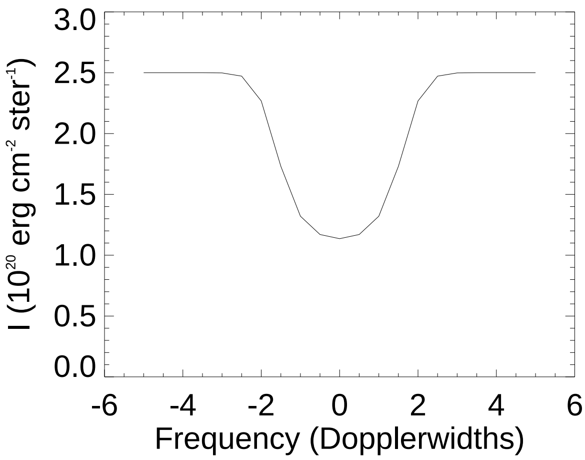

Figure 2.3: Emerging intensity as a function of frequency. The frequencies are in Doppler widths $\nu_{\mathrm{D}} = \frac{\nu_0}{c}\sqrt{\frac{2kT}{m}}$.

2.3 The Eddington-Barbier Approximation

2.11 —

2.12 See figure 2.4

2.4 Inversion

2.13 $(\mathbf{K}^{\mathrm{T}}\,\mathbf{K})^{-1}\,\mathbf{K}^{\mathrm{T}}\,\mathbf{I} = \mathbf{S}$

2.14 Any matrix with the same dimensions as $\mathbf{K}^{\mathrm{T}}\,\mathbf{K}$ will work.

2.15 See figure 2.5

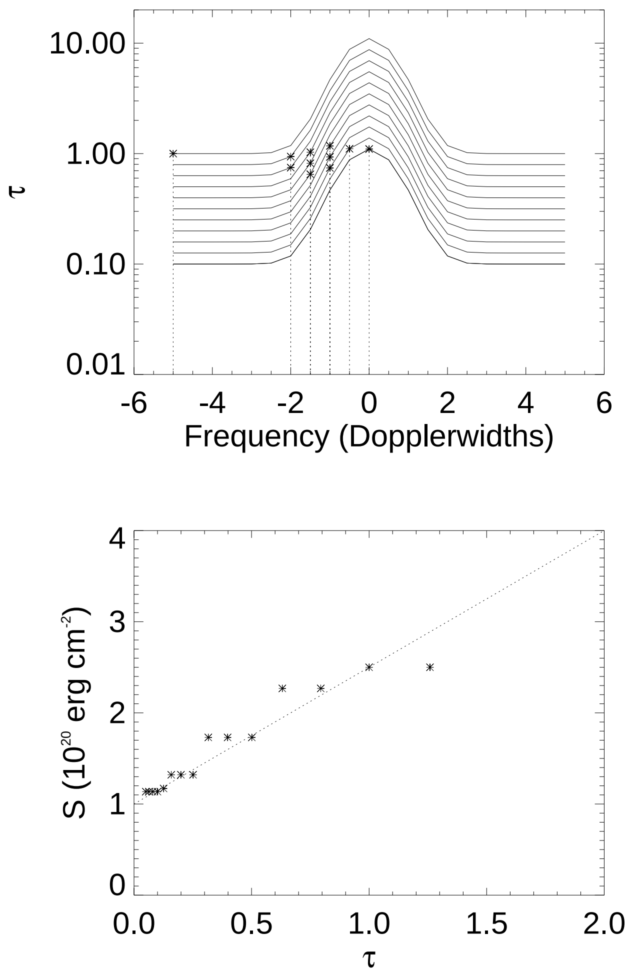

Figure 2.4: Upper panel: Determination of frequencies for which the line optical depth $\tau \simeq 1$. Lower panel: Source function as a function of gray optical depth, using the frequencies found in the upper panel and the Eddington-Barbier relation $S_\nu(\tau_\nu = \mu) \simeq I_\nu^+(\tau_\nu = 0, \mu)$, where $\mu = 1$.

2.5 Regularization

2.16 The regularization matrix $\mathbf{H}$ has the form

$$\mathbf{H} = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & & & & & & & & \\ -2 & 5 & -4 & 1 & 0 & \dots & & & & & & & \\ 1 & -4 & 6 & -4 & 1 & 0 & \dots & & & & & & \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & \dots & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & & & \dots & 1 & -4 & 6 & -4 & 1 \\ & & & & & & & & \dots & 1 & -4 & 5 & -2 \\ & & & & & & & & & \dots & 1 & -2 & 1 \end{pmatrix}$$

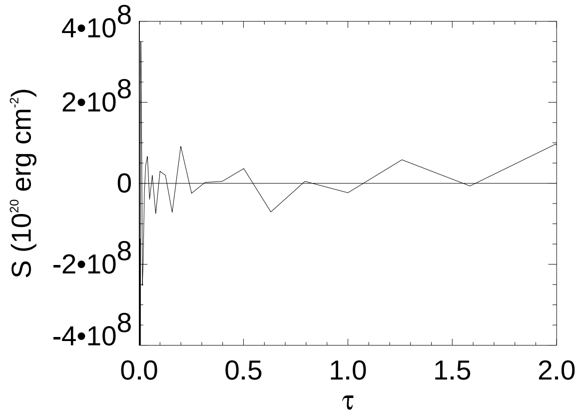

Figure 2.5: Source function determined by straightforward inversion of $\mathbf{I} = \mathbf{K}\,\mathbf{S}$ with the help of the relation $(\mathbf{A}\,\mathbf{K})^{-1}\mathbf{A}\,\mathbf{I} = \mathbf{S}$. $\mathbf{A}$ is non-unique, and the source function determined by this inversion is highly unstable. In this plot we used $\mathbf{A} = \mathbf{K}^{\mathrm{T}}$.

2.17 $\lambda = 1.2 \times 10^{-4}$, determined by calculating $\sum((S - S_{new})^2)$.

2.18 See figure 2.6.

2.19 See figure 2.6. Enough noise spoils the results.

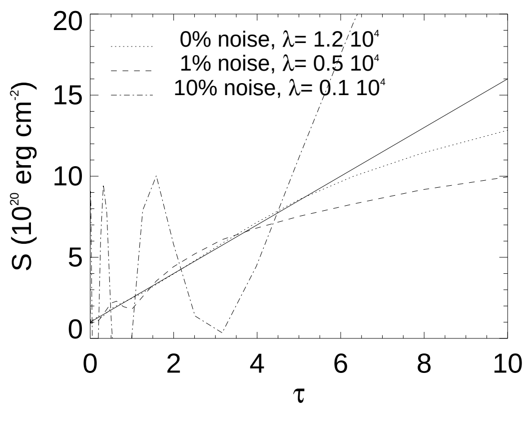

Figure 2.6: Source function determined by inversion of the regularized equation $(\mathbf{K}^{\mathrm{T}}\,\mathbf{K} + \lambda\mathbf{H})^{-1}\mathbf{K}^{\mathrm{T}}\,\mathbf{I} = \mathbf{S}$, where we have added several percentages of random noise to the emerging intensity.

Chapter 3

The Newton-Raphson Method

3.1 Four iteraties are required to find $\sqrt{2}$ up to th 5th decimal with starting estimate x=1.

3.2 Ten iteraties are required to find $\sqrt{2}$ up to th 5th decimal with starting estimate x=100.

3.3 We now find $-\sqrt{2}$ in same number of iterations. This is also a solution of the equation.

3.4 There are four solutions, 1, -1, i, and $-i$. The equation becomes $\delta x = (1 - x^4)/(4x^3)$.

3.5 See figure 3.1.

3.6 Some programming is omitted.

3.7 With a finer grid the result looks the same. We have a fractal.

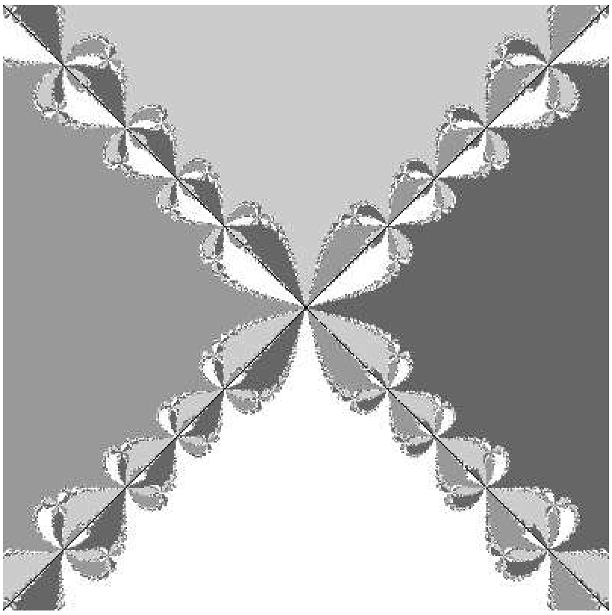

Figure 3.1: Application of the Newton-Raphson method for $x^4 = 1$ in the complex plane. The colors correspond to the solutions $(1, i, -1, -i)$ reached after 99 iterations.

Chapter 4

The Feautrier Method

4.1 Feautrier Method by hand

4.1 Approximate $\mathrm{d}P = P_{i+1} - P_i$, $\mathrm{d}\tau = \delta$,

$$\frac{P_{i+1} - P_i}{\delta^2} - \frac{P_i - P_{i-1}}{\delta^2} = P_i - S_i \ , \tag{4.1}$$

$$P_{i+1} - (2 + \delta^2)P_i + P_{i-1} = \delta^2 S_i \ . \tag{4.2}$$

4.2 The first order boundary conditions (Lecture Notes page 119) are

$$\tau \ll 1 : \quad \frac{\mathrm{d}P}{\mathrm{d}\tau} = P \ , \tag{4.3}$$

$$\frac{P_1 - P_0}{\delta} = P_0 \ , \tag{4.4}$$

$$P_0 = \frac{P_1}{1 + \delta} \ , \tag{4.5}$$

$$\tau \gg 1 : \quad \frac{\mathrm{d}P}{\mathrm{d}\tau} = B + \frac{\mathrm{d}B}{\mathrm{d}\tau} - P \ , \tag{4.6}$$

$$\frac{P_n - P_{n-1}}{\delta} = B_{n-1} + \left(\frac{\mathrm{d}B}{\mathrm{d}\tau}\right)_{n-1} - P_{n-1} \ , \tag{4.7}$$

$$P_{n-1} = \frac{P_n - \delta B_{n-1} - \delta \left(\frac{\mathrm{d}B}{\mathrm{d}\tau}\right)_{n-1}}{1 - \delta} \ . \tag{4.8}$$

4.3 For $i = 1$ and $i = 2$ respectively we have

$$P_2 - (2 + \delta^2)P_1 + P_0 = -\delta^2 S_1 \ , \tag{4.9}$$

$$P_2 - (2 + \delta^2)P_1 + \frac{P_1}{1 + \delta} = -\delta^2 S_1 \ , \tag{4.10}$$

$$\frac{(1 + \delta)(P_2 + \delta^2 S_1)}{(1 + \delta)(2 + \delta^2) - 1} = P_1 \ , \tag{4.11}$$

$$
\begin{aligned}
-\delta^2 S_2 &= P_3 - (2+\delta^2)P_2 + P_1 \ , && (4.12)\\
-\delta^2 S_2 &= P_3 - (2+\delta^2)P_2 + \frac{(1+\delta)(P_2+\delta^2 S_1)}{(1+\delta)(2+\delta^2)-1} \ , && (4.13)\\
P_2 &= \frac{\left[(1+\delta)(2+\delta^2)-1\right]\left[P_3+\delta^2\left(\frac{1+\delta}{(1+\delta)(2+\delta^2)-1}S_1+S_2\right)\right]}{(2+\delta^2)\left[(1+\delta)(2+\delta^2)-1\right]-(1+\delta)} \ . && (4.14)
\end{aligned}
$$

4.2 Feautrier Program

4.4 See Lecture Notes eqns. 5.16 and 5.17,

$$
\begin{aligned}
P_1 &\approx P_0 + P_0'\Delta\tau_0 + \frac{1}{2}P_0''\Delta\tau_0^2 \ , && (4.15)\\
P_0' &= \frac{1}{2}(I_0^+ - I_0^-) = P_0 - I_0^- \ , && (4.16)\\
P_0'' &= P_0 - S_0 \ , && (4.17)
\end{aligned}
$$

so that

$$
P_1 \approx P_0 + \Delta\tau_0(P_0 - I_0^-) + \frac{1}{2}\Delta\tau^2(P_0 - S_0) \ . \qquad (4.18)
$$

4.5 Use

$$
\begin{aligned}
I_{n-1}^+ &= R_{n-1}I_{n-1}^- + H_{n-1} \ , && (4.19)\\
I_{n-1}^+ &= 2P_{n-1} - I_{n-1}^- \ , && (4.20)
\end{aligned}
$$

to find

$$
I_{n-1}^- = \frac{2P_{n-1} - H_{n-1}}{1 + R_{n-1}} \ . \qquad (4.21)
$$

4.6 In the deepest point of the atmosphere in an half-infinite space $I^+ = I^-$, so $H_n = 0$ and $R_n = 1$.

4.7 For an illuminated medium $I^- = I_{\rm ill}$, so that $H_0 = I_{\rm ill}$ and $R_0 = 0$.

4.8 For a symmetrical slab $I^+ = I^-$, so $H_n = 0$ and $R_n = 1$.

4.9 —

4.10 `tau(0)` is not equal 0, so the radiation still has to go through very small part of the atmosphere. $I' = I - S$, $(I\mathrm{e}^{-\tau})' = -S\mathrm{e}^{-\tau}$, $I_0 = I_\tau \mathrm{e}^{-\tau} + (1-\mathrm{e}^{-\tau})S_\tau$, if you assume that S_τ equals S_0.

Chapter 5

Λ-Iteration

5.1 Angle-Quadrature

5.1 The Feautrier transport equation can be rewritten as

$$\frac{\mathrm{d}^2 P(\tau/\mu)}{\mathrm{d}(\tau/\mu)^2} = P(\tau/\mu) - S(\tau) \ , \tag{5.1}$$

so that the transformation is $\tau \to \tau/\mu$, while S stays the same.

5.2

$$J(\tau) = \frac{2\ \mathrm{h}}{45}(7P(\tau,\mu_1) + 32P(\tau,\mu_2) + 12P(\tau,\mu_3) + 32P(\tau,\mu_4) + 7P(\tau,\mu_5)) \ . \tag{5.2}$$

5.3 The values of μ in the procedure are 0.05, 0.275, 0.5, 0.725, 0.95. The value $\mu = 0$ would lead to infinity in τ/μ, so this is avoided. The summation is normalized to 0.5.

5.2 Λ-Matrix

5.4 We know $Z_0 = \frac{S_0}{B_0}$, so $X_{00} = \frac{1}{B_0}$. For Z_1, we find

$$\begin{aligned} Z_1 &= \frac{S_1 + A_1\, Z_0}{C_1(1+F_1)} \ , & (5.3)\\ &= \frac{S_1 + A_1\, X_{00}\, S_0}{C_1(1+F_1)} \ , & (5.4)\\ &= X_{10} S_1 + X_{10} S_0 \ , & (5.5) \end{aligned}$$

so that

$$\begin{aligned} X_{10} &= \frac{A_1\, X_{00}}{C_1(1+F_1)} \ , & (5.6)\\ X_{11} &= \frac{1}{C_1(1+F_1)} \ . & (5.7) \end{aligned}$$

The expression for Z_2 are

$$Z_2 = \frac{S_2 + A_2\, Z_1}{C_2(1+F_2)} \; , \tag{5.8}$$

$$= \frac{(S_2 + A_2\, X_{11}\, S_1 + A_2\, X_{10}\, S_0)}{C_2(1+F_2)} \; . \tag{5.9}$$

Hence,

$$X_{20} = \frac{A_2\, X_{10}}{C_2(1+F_2)} \; , \tag{5.10}$$

$$X_{21} = \frac{A_2\, X_{11}}{C_2(1+F_2)} \; , \tag{5.11}$$

$$X_{22} = \frac{1}{C_2(1+F_2)} \; . \tag{5.12}$$

5.5 With $P_n = 0$ and the bidiagonal system

$$P_i = (1+F_i)^{-1} P_{i+1} + Z_i \; , \tag{5.13}$$

we find

$$P_{n-1} = Z_{n-1} \tag{5.14}$$

$$= [\mathbf{X}\,\mathbf{S}]_{n-1} \; , \tag{5.15}$$

thus, writing $\mathbf{P} = \mathbf{T}\,\mathbf{S}$, we find $T_{n-1,j} = X_{n-1,j}$. For P_{n-2} we find

$$P_{n-2} = (1+F_{n-2})^{-1} [\mathbf{X}\,\mathbf{S}]_{n-1} + [\mathbf{X}\,\mathbf{S}]_{n-2} \; , \tag{5.16}$$

so that $T_{n-2,j} = (1+F_{n-2})^{-1}\, X_{n-1,j} + X_{n-2,j}$.

5.6 The IDL program should look like the following.

```
FUNCTION LMBD,TAU
 nd=n_elements(tau)
 quadrature,a,mu
 ms=n_elements(mu)
 Lambda=fltarr(nd,nd)
 FOR i=0,ms -1 DO Lambda=Lambda+a(i)*lmbd_matrix(tau,mu(i))
RETURN,Lambda
END
```

5.7 See figure 5.1.

5.3 Inversion

5.8 $\mathbf{S} = (1-\epsilon)\mathbf{J} + \epsilon\,\mathbf{B}$, $\mathbf{S} = (1-\epsilon)\mathbf{\Lambda}\,\mathbf{S} + \epsilon\,\mathbf{B}$, $\mathbf{S} = (1-(1-\epsilon)\mathbf{\Lambda}^{-1}(\epsilon\,\mathbf{B})$

5.9 See figure 5.2.

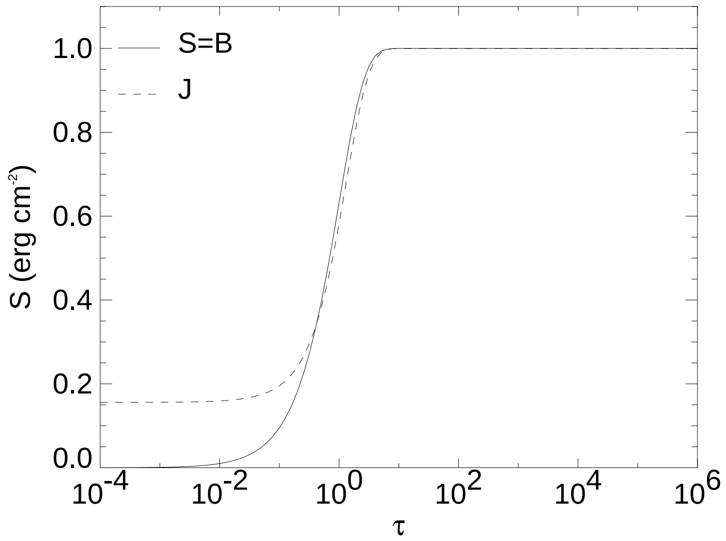

Figure 5.1: $S(\tau) = B(\tau) = 1 - \mathrm{e}^{-\tau}$, $\mathbf{J} = \mathbf{\Lambda}\,\mathbf{S}$.

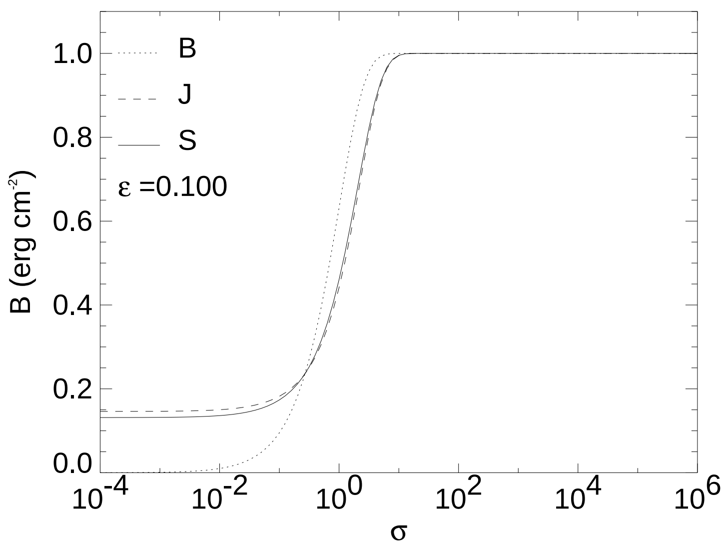

Figure 5.2: $B(\tau) = 1 - \mathrm{e}^{-\tau}$, $\epsilon = 0.1$. Source function derived by straightforward inversion.

5.4 Λ-Iteration

5.10 The iteration is stopped when the difference between steps becomes smaller then $\epsilon/100$.

5.11 The number of steps varies from 2434 for $\epsilon = 10^{-3}$ to 2 for $\epsilon = 0.9$.

5.12 See figure 5.3.

5.13 See figure 5.4.

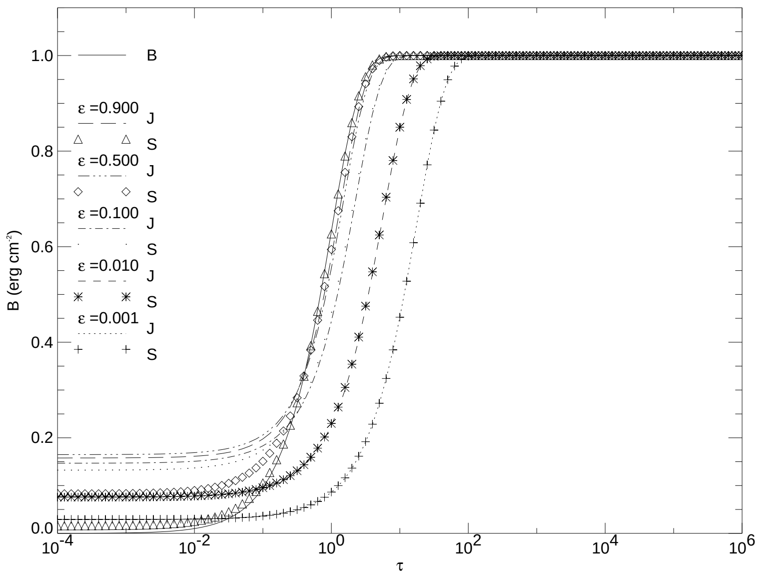

Figure 5.3: Source function derived by lambda iteration for different ϵ.

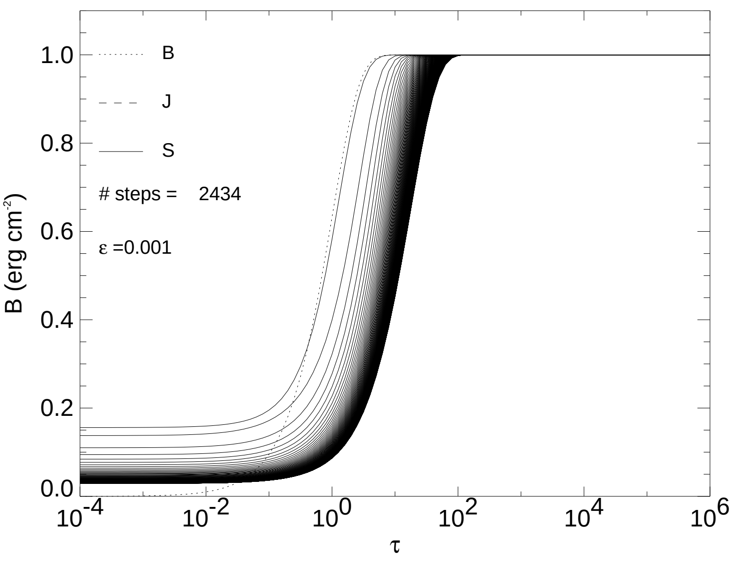

Figure 5.4: The source function derived by **Λ**-iteration.

5.5 Accelerated Λ-Iteration

5.14 The inverse of a diagonal matrix is the matrix with the inverse of each element on the diagonal.

5.15 —

5.16 —

5.17 —

5.18 —

5.19 —

5.20 —

5.21 —

5.22 Some relatively easy IDL code is omitted.

5.23 See figure 5.5.

5.24 See figure 5.6.

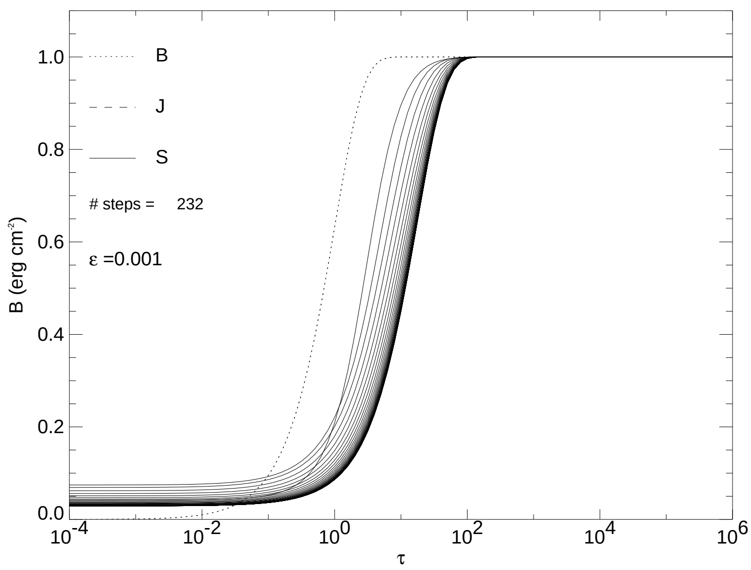

Figure 5.5: The source function derived by accelerated **Λ**-iteration.

5.6 Optional

5.25 —

5.26 —

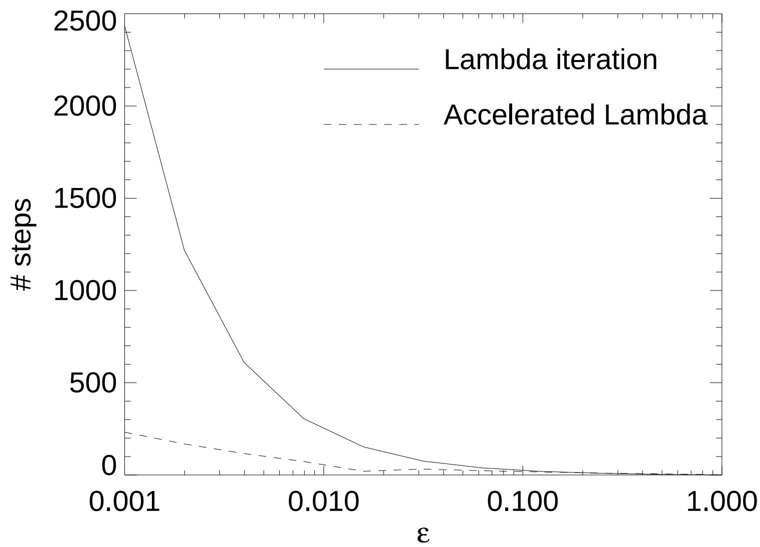

Figure 5.6: Number of steps needed for convergence of lambda iteration and accelerated lambda iteration, as a function of destruction probability ϵ.