The Radiative Transfer Code *STOKES*
Version 1.2

User's manual

René W. Goosmann
Frédéric Marin

web: www.stokes-program.info
contact: admin@stokes-program.info

March 2014

Contents

What is new in version 1.2?

Since it was put online in 2004, the *STOKES* radiative transfer code has evolved in many ways and now offers a much broader range of applications. We decided to upgrade the publicly availble version that is optimized for use in the optical and UV range and that includes a few new features that we hope will be useful for a larger community interested in polarization, spectroscopy and imaging of astrophysical objects in which scattering is a dominant reprocessing mechanism.

With respect to the basic code *STOKES* 1.0 this latest public version 1.2 contains the following extensions:

1. the capacity of polarization and spectroscopic imaging
2. a wider range of possible scattering geometries
3. three-dimensional velocites for all emission and scattering regions
4. a better treatment of the optical depth in multiple scattering regions
5. a faster conversion of the Monte Carlo results

The imporvements are described in more detail in this manual and in Marin et al. (2012) and Goosmann et al. (2013). If this new version of *STOKES* is helpful for your research we kindly ask you to give reference to these articles. The source codes, model examples and this manual can be found on the *STOKES* website:

`www.stokes-program.info`

Should you have questions or problems applying this code do not hesitate to contact us by email:

`admin@stokes-program.info`

Of course, we continue to be very interested in your feed-back on the code and the possible use you may have for it. Also, please report bugs that you eventually find.

Note that more advanced versions of the code are continously developed. They include features like photoionization and absorption effects, line emission and scattering, new morphologies of emission and reprocessing regions, and also extend the simulations into the X-ray range. These versions of *STOKES* are not yet publicly available. Nonetheless, if you want to learn more about them you may contact us.

Finally, we want to thank Paulina Lira (Calán Observatory, Santiago di Chile) and Lorraine Braibant (University of Liège, Belgium) for carefully testing this new version and pointing out some inconsistencies that we were able to repair.

Strasbourg (France) in March 2014,
René Goosmann
Frédéric Marin

Preface of *STOKES* Version 1.0

The computer program *STOKES* was designed to perform three-dimensional radiative transfer simulations for astronomical applications. A crucial requirement was that the code should also consider the polarization properties of the radiation. The program is based on the Monte-Carlo method and treats optical and ultraviolet polarization induced by scattering off free electrons or dust grains. Emission and scattering regions can be arranged in various geometries within the model space, the computed continuum and line spectra can be evaluated at different inclinations and azimuthal viewing angles.

The idea of *STOKES* was given by the observed polarization of Active Galactic Nuclei. Martin Gaskell at the University of Nebraska in Lincoln (USA) suggested to me the project to write a Monte-Carlo radiative transfer code in order to model quasar polarization in the optical and ultraviolet range. Developing the program, we considered existing radiative transfer codes described in the literature. The Monte-Carlo code written by Kartje (1995), the *DIRTY* code (Gordon et al., 2001; Misselt et al., 2001), and the work of Fischer et al. (1994) and Wolf & Henning (1999) were very inspiring. We also would like to mention that Watanabe et al. (2003) present a radiative transfer code being of a similar design as *STOKES*.

We want to make *STOKES* publicly available and therefore publish the first release, e.g. the version 1.0, on the web page

`www.stokes-program.info`

On this website you can also find further support for the program including examples for the input file, precomputed dust models, and the C^{++} source files. The program is available pre-compiled for Linux and Windows systems. We intend to further develop the code and to release follow-up versions.

If you are an astronomer or a physicist doing research on polarization and radiative transfer you might have use for *STOKES*. The program is 100% share ware, so you are invited to use it for your modeling purposes, adjust the source code to your personal needs, and forward the program to interested colleagues.

This manual gives a brief introduction to the concept of *STOKES* and the methods included (chapter 1). We explain the application of the code and the processing of its results (chapter 2) and provide some examples for illustration (chapter 3). As for any computer program, the flexibility and reliability of *STOKES* will rise with the feedback of its users. If you make use of the code and you would like to send feedback, suggestions for improvements, or point out bugs you may send email to:

`admin@stokes-program.info`

If *STOKES* is helpful for your research we kindly ask you to give reference to the article Goosmann & Gaskell (2007) in which we present the program.

Meudon (France) in November 2004,
René Goosmann

Chapter 1

A brief introduction to *STOKES*

In this chapter we introduce the basic methods of the code (section 1.1) and present its main routines (section 1.2). These are then explained in more detail (section 1.3) by following the "life" of a model photon. The dust scattering routines are based on Mie theory (section 1.4) from which extinction cross-sections and the elements of scattering matrices can be obtained.

1.1 Overview

The *STOKES* program is based on the Monte-Carlo method (see e.g. Cashwell & Everett, 1959) and follows single photons from their generation inside the emission regions through various scattering processes until they become absorbed or manage to escape from the model region (see Fig. 1.1). The emission regions can be realized in cylindrical or double-conical geometries. The flux spectrum of the continuum radiation is defined by a power law $F(\nu) \propto \nu^{-\alpha}$ with spectral index α. For the line photons Lorentzian flux spectra are assumed. The relative line intensity with respect to the continuum flux can be defined by the user. Intrinsically, line and continuum photons are assumed to be unpolarized.

Around the source regions various geometries of scattering mirrors can be arranged. At its current state, the program offers toroidal, cylindrical, spherical, or conical shapes. These regions can be filled with free electrons or with dust consisting of silicate and graphite. Three-dimensional bulk motions of the scattering regions can be included, they result in a wavelength shift of the scattered photons. A photon works its way through the model region and generally undergoes several scattering processes. The path length between two scattering events is sampled by the Monte-Carlo method. We use accurate algorithms for Thomson and dust scattering (see section 1.3.4) in which the scattering angle is sampled depending on the polarization state of the incident photon.

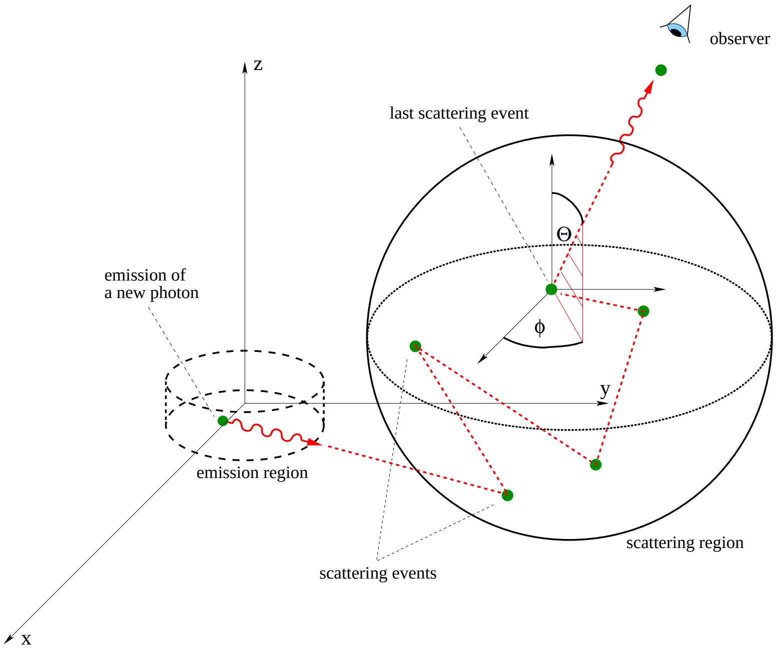

Figure 1.1: A photon working its way through the model space

In case of dust extinction, a certain fraction of the photons is absorbed and never reaches the virtual observer. If a photon escapes from the model region it is registered by a web of virtual detectors arranged in a spherical geometry. The flux and polarization information is obtained by adding up the Stokes parameters of all photons detected at a given wavelength and position of the detector. The object can then be evaluated for total and polarized flux, fraction of polarization, and for the polarization position angle at each viewing direction. It is possible to extract at the same time images and spectra for these values.

The seed for the random number generator is taken from the computers internal clock, hence two runs of the same model setup but started at different points in time do not give exactly the same results. They differ within the Poisson statistics of the Monte-Carlo method. In the current version 1.2 of the code a new random number generator based on a Mersenne Twister algorith was implemented and significantly improves the performance of the code (see Marin et al., 2012, for details). Considering the fact that the Stokes parameters are additive allows the user to conduct simulations with very large photon numbers by running the program on different CPUs simultaneously and with different seeds for the random number generator. The results can later be combined to one set using the auxiliary program *ANALYZE* (see chapter 2).

1.2 Structure of the program

The *STOKES* program is written in C^{++}. The object-oriented capabilities of C^{++} allow us to divide a complex program into organic parts containing all their properties (declarations) and capabilities (routines) by themselves. For the *STOKES* program, these organic parts were designed according to the physical aspects of the model space, of the radiative transfer, and of the virtual detectors.

We use a Cartesian coordinate system to define the geometrical features of the model region: source areas and scattering regions. The geometry of these elements is described by mathematical equations and stored by the object *model*. This object initializes all relevant model parameters and then works as a database for the other objects. The geometry of all emitting and scattering regions, their extinction or scattering densities and systematic velocities, the number of viewing angles at the models space, and the spatial and spectral resolution of the detectors are stored in *model*.

To include dust scattering, an additional object called *mie* supports *model*. This object contains routines to calculate Mie parameters consisting of cross-sections and scattering matrix elements. Their calculation is time consuming and it would slow down the program significantly if the Mie data were calculated during the actual simulation. Instead, if dust scattering is part of the run, *mie* does all the necessary calculations at the beginning of the program or simply reads in precomputed Mie data from a file. It then stores all the relevant parameters in a look-up table.

The radiative transfer is realized by the object *photon*. This object defines all properties and capabilities of a photon being relevant for the modeling. It stores the photon's wavelength, Stokes vector, position inside the model region, and direction of flight. Routines to move a photon inside the model region and to make it interact with matter are also defined in *photon*. The routines of *photon* will be described in more detail in the next section 1.3.

The object *monitor* rules the observational part of the system. It manages a web of polarization spectro-imaging detectors summing up and storing the Stokes parameters of all detected photons as a function of wavelength and projected postion on the plane of the sky. The viewing directions are given by the two angles θ and ϕ defined with respect to the coordinate system of the model space. We measure θ between the line of sight and the z-axis and ϕ between the projected line of sight onto the xy-plane and the x-axis. Each detector is defined as a (sub-)object at a given viewing angle and stores spectra of Stokes parameters. During execution *monitor* regularly saves the data on a disk. It creates four files containing information about the four Stokes parameters I, Q-, U- and V.

1.3 A photon's journey

We describe the major routines of the *STOKES* program and how they work together by following a model photon's journey from its generation to its absorption or detection.

The photon is generated by the *source-routine.* If it is generated inside a scattering region, the *proceed-routine* shifts the photon by a path-length that is sampled using the Monte-Carlo method. If the photon does not leave the cloud, which is checked by the *inside-cloud-routine*, a scattering event is simulated by the *dust-scattering-routine* or by the *electron-scattering-routine.* In case of dust scattering the routine might set up the absorption flag, hence destroy it, and make the cycle start over by creating a new photon inside the source region.

If the photon leaves the scattering cloud or if it was not generated within one, the *shift-routine* detects any other scattering region along its flight direction. If there is another scattering cloud, the photon moves to the cloud's border, and the *proceed-routine* takes over again. If there is no further scattering cloud on the way, the photon escapes from the model region and is detected by the *register-routine.* Its Stokes parameters are saved according to its wavelength and flight direction, wavelength and position on the plane of the sky.

The execution of the program is visualized in the flowchart in Fig. 1.2. The routines ruling a photon's journey are described in more detail in the following sections.

1.3.1 The source routine

To generate a model photon, its initial parameters position, flight direction, wavelength, and polarization have to be set. Some of these values are sampled using the Monte-Carlo method and hence involve random numbers. In the following we denote such random numbers in the range between 0 and 1 by r_{i}.

The sampling of the initial position depends on the nature of the photon. If it is a continuum or a broad line photon it is emitted in a cylindrical region while a narrow line photon originates in a double-conical region. Assuming a constant density of the emitting material a random starting position for the photon is sampled inside the source region declared.

At each position of the source elements we assume isotropic emission. The initial flight direction of the photon is determined by two angles, a polar angle θ and an azimuthal angle ϕ, which are defined with respect to a standard spherical coordinate system. From this, we obtain the following sampling equations for the angles:

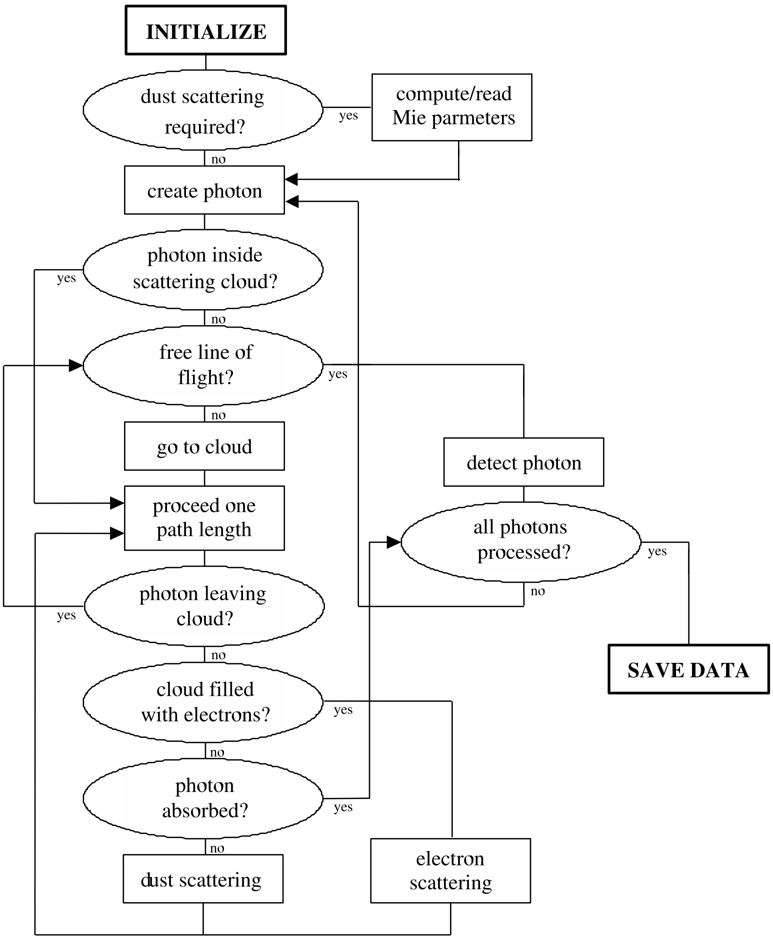

Figure 1.2: Flowchart showing the execution of *STOKES*

$$\theta = \arccos(1-2r_1), \tag{1.1}$$

$$\phi = 2\pi r_2. \tag{1.2}$$

In order to determine the wavelength λ of a continuum photon we assume a power law $F(\nu) \propto \nu^{-\alpha}$ for the intrinsic flux spectrum. Herein, ν denotes the photon frequency and α is the spectral index. The wavelength is sampled over a range $[\lambda_{\min}, \lambda_{\max}]$. This leads to the sampling equations:

$$\lambda = \left\{ \begin{array}{ll} \left[\lambda_{\mathrm{min}}^{\alpha} + r_3 \left(\lambda_{\mathrm{max}}^{\alpha} - \lambda_{\mathrm{min}}^{\alpha}\right)\right]^{\frac{1}{\alpha}}, & \alpha \neq 1, \\ \lambda_{\mathrm{min}}^{\alpha} \left(\frac{\lambda_{\mathrm{max}}^{\alpha}}{\lambda_{\mathrm{min}}^{\alpha}}\right)^{r_3}, & \alpha = 1. \end{array} \right\} \tag{1.3}$$

For line photons we presume a Lorentzian profile with centroid λ_0, and line width Γ. The sampling equation is more complicated and can not be explicitly solved in analytical form. If we define the auxiliary function $\bar{F}(\lambda')$ as

$$\bar{F}(\lambda') = \frac{1}{2} \log \left[(\lambda' - \lambda_0)^2 + \Gamma^2\right] + \frac{\lambda_0}{\Gamma} \arctan \frac{\lambda' - \lambda_0}{\Gamma}, \tag{1.4}$$

the wavelength λ of the photon is numerically computed from the equation

$$\bar{F}(\lambda) - (1 - r_3)\bar{F}(\lambda_{\mathrm{min}}) - r_3\bar{F}(\lambda_{\mathrm{max}}) = 0, \tag{1.5}$$

Note that the above sampling equations include the wavelength-dependence of the photon energy.

The polarization properties of a photon are stored in its Stokes vector consisting of the four Stokes parameters I, Q, U, and V. If we imagine an electromagnetic wave being associated to the photon, then the Stokes parameters can be derived from the position of the electric field vector $\vec{E}$ inside the so-called polarization plane. The polarization plane is oriented perpendicularly to the flight direction of the photon. We assume that newly generated photons are unpolarized. In this case, the Stokes vector has the simple form:

$$\begin{pmatrix} I \\ Q \\ U \\ V \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{1.6}$$

It is possible to generate photons with a different intial polarization state by adapting the source routine accordingly.

1.3.2 The proceed routine

If we ignore scattering back into the beam, the intensity of a photon beam traversing a slab of scattering material with particle number density n and extinction cross-section σ will drop by a factor $e^{n\sigma l}$, with l being the distance traveled inside the scattering region. The same factor therefore determines the probability density of a photon-particle interaction at the distance l. This leads to the sampling function:

$$l = \frac{1}{n\sigma} \ln(1 - r_4). \tag{1.7}$$

The factor $\frac{1}{n\sigma}$ is the mean free path length. Depending on the scattering material, the program will use either a dust extinction cross-section $\sigma_{\rm ext}$ computed from Mie theory or, in case of electron scattering, the Thomson cross section $\sigma_{\rm Th}$.

1.3.3 The inside-cloud routine

After the photon was shifted by a path length and before conducting the next scattering event, it is necessary to check if the photon is still inside the scattering cloud. Each scattering region is mathematically described by a polynomial in-equation, or by a set of coupled polynomial in-equations. For instance, in the case of a spherical cloud with central position $\vec{p}_0 = (x_0, y_0, z_0)^T$ and radius r such an in-equation is given by:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 < r^2, \tag{1.8}$$

with x, y, and z being the photon coordinates. To decide if the photon is inside or outside the region, the program evaluates the associated in-equation(s) for the current photon position.

1.3.4 The scattering routines

Consider a photon being scattered off a spherical particle (see Fig. 1.3). The outgoing electromagnetic wave associated with the photon can be resolved into two components (see e.g. Bohren & Hufman, 1983). These components refer to directions of the electric field vector parallel and perpendicular to the scattering plane. We name the two components $E_\parallel$ and $E_\perp$. For scattering off a spherical particle the following relation between the incoming and scattered electric fields holds:

$$\begin{pmatrix} E_{\parallel,s} \\ E_{\perp,s} \end{pmatrix} = \begin{pmatrix} S_2(\theta) & 0 \\ 0 & S_1(\theta) \end{pmatrix} \begin{pmatrix} E_{\parallel,i} \\ E_{\perp,i} \end{pmatrix}. \tag{1.9}$$

The values $S_1(\theta)$ and $S_2(\theta)$ are the so-called scattering matrix elements. They are independent of the azimuthal angle ϕ. In case of Thomson scattering they are given by simple analytic expressions:

$$S_1(\theta) = \cos^2\theta, \tag{1.10}$$

$$S_1(\theta) = 1. \tag{1.11}$$

For dust scattering the albedo and the matrix elements of a standard dust grain are calculated from Mie theory (see section 1.4). The albedo at the photon wavelength is compared to a random number r_5 in order to decide whether the photon is absorbed or scattered. In case of absorption, the photon is lost, and the cycle starts over with the generation of a new photon by the *source routine*.

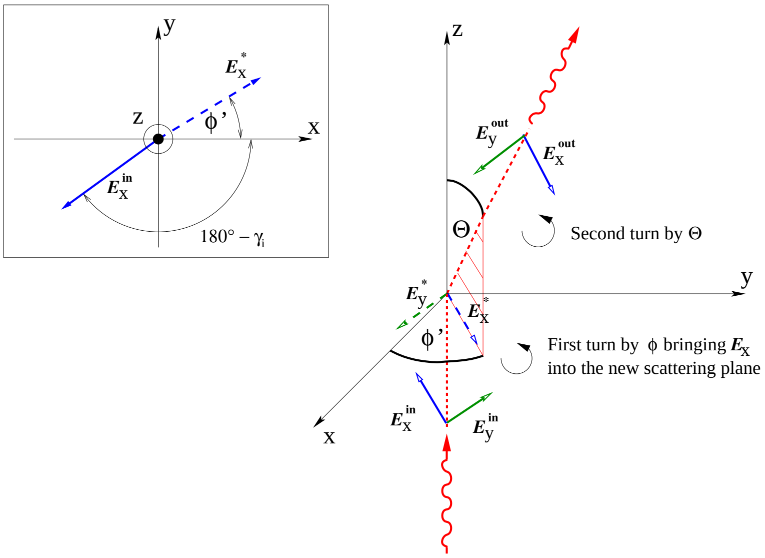

Figure 1.3: Geometry and denotations for a single scattering event. The inset shows the first rotation of the $\vec{E}$-vector by the angle ϕ, the view on the polarization plane is along the negative z-axis.

If the photon is scattered the angle-dependent classical intensity distribution of a scattered electromagnetic wave measures the probability of finding it at a certain direction. Such probability density distributions are derived from equation (1.9).

To also consider the impact of the incident polarization on the scattering angle, we split up the incident intensity I_i into a completely polarized and a completely unpolarized part:

$$I_\mathrm{i} = I_\mathrm{i,pol} + I_\mathrm{i,unpol} = P_\mathrm{i} I_\mathrm{i} + (1 - P_\mathrm{i}) I_\mathrm{i}. \tag{1.12}$$

The incident polarization is denoted by P_i, its position angle by γ_i. Both are calculated from the incident Stokes parameters of the photon. We now rotate the polarization plane around the incident flight direction by the angle $180° - \gamma_\mathrm{i}$ in order to reach a position where the incident electric field vector of the completely polarized component is aligned with the reference axis (position angle $= 0°$ from the x-axis, see the inset of Fig. 1.3). We then compute the angle-dependent intensity I_s behind the scattering center from equation 1.9 together with equation 1.12:

$$I_\mathrm{s}(\theta, \phi') = \frac{I_\mathrm{i}}{k^2 d^2} \left[\begin{array}{c} \frac{1}{2}\left(|S_1(\theta)|^2 + |S_2(\theta)|^2\right)(1 - P_\mathrm{i}) + \\ \left(|S_1(\theta)|^2 \sin^2\phi' + |S_2(\theta)|^2 \cos^2\phi'\right) P_\mathrm{i} \end{array} \right], \tag{1.13}$$

with d being the distance between the scattered photon and the scattering center, and k denoting the wave-number. Considering this intensity as a probability density p, we obtain the probability distribution for the scattering angles θ and ϕ':

$$\begin{aligned} P(\theta) &= C\int_0^{\pi}\left(|S_1(\theta)|^2+|S_2(\theta)|^2\right)\sin\theta' d\theta', && (1.14)\\ P_\theta(\phi') &= \frac{1}{2\pi}\left(\phi'-\frac{|S_1(\theta)|^2-|S_2(\theta)|^2}{|S_1(\theta)|^2+|S_2(\theta)|^2}P\frac{\sin 2\phi'}{2}\right). && (1.15)\end{aligned}$$

The number C is a normalization constant in order to assure that the number (1.14) is in the range from 0 to 1 for scattering angles between 0° and 180°. To sample ϕ' and θ, the right hand-sides of these equations have to be set equal to random numbers r_6, r_7. Then, they are solved for the angles. Note that the sampling of the scattering angle θ is independent of the incident polarization. Since at the beginning we performed a coordinate transformation to make the electric field vector oscillate along the reference axis, the sampling of ϕ' has to be followed by the back-transformation

$$\phi = 180^\circ - \gamma_{\rm i} + \phi'. \qquad (1.16)$$

This procedure of finding the sampling equations for the scattering angles is analogous to the one followed by Fischer et al. (1994).

With each scattering event, the co-moving coordinate system undergoes a double rotation: the first rotation, by the azimuthal angle ϕ, happens around the current flight direction of the photon. It rotates the **E**-vector inside the polarization plane to the position of the new scattering plane (see Fig. 1.3). Physically, it does not affect the polarization state, but the Stokes vector undergoes the following coordinate transformation (see Bohren & Hufman, 1983):

$$\begin{pmatrix} I^* \\ Q^* \\ U^* \\ V^* \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ & \cos 2\phi & -\sin 2\phi & 0 \\ 0 & \sin 2\phi & \cos 2\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} I^{in} \\ Q^{in} \\ U^{in} \\ V^{in} \end{pmatrix}. \qquad (1.17)$$

The second rotation happens inside the scattering plane and by the scattering angle θ. The change of the Stokes vector is determined by the so-called Mueller matrix. For scattering off a spherical particle, the matrix is of the following form:

$$\begin{pmatrix} I^{out} \\ Q^{out} \\ U^{out} \\ V^{out} \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 \\ S_{12} & S_{22} & 0 & 0 \\ 0 & 0 & S_{33} & S_{34} \\ 0 & 0 & -S_{34} & S_{44} \end{pmatrix} \begin{pmatrix} I^* \\ Q^* \\ U^* \\ V^* \end{pmatrix}. \qquad (1.18)$$

The entries of the Mueller matrix are closely related to the elements of the scattering matrix $S_1(\theta)$ and $S_2(\theta)$ as is explained in section 1.4

1.3.5 The shift routine

When a photon escapes from a scattering cloud, the code checks if there is any other scattering region along its flight direction. The photon trajectory is defined by the current photon position $\vec{x}_p$ and by the velocity vector $\vec{v}_p$:

$$\vec{x} = \vec{x}_p + t\vec{v}_p, \tag{1.19}$$

with t denoting the time. The borders of the scattering clouds are described by polynomial equations. In order to find possible clouds along the photon's flight direction, it is hence necessary to combine these polynomial equations with equation 1.19 and to solve for t. For instance, in the case of a spherical scattering cloud with central position $\vec{p}_0 = (x_0, y_0, z_0)^T$ and radius r the border is described by the equation:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2, \tag{1.20}$$

which leads to the following equation for the time parameter t:

$$t^2 - 2t(\vec{x}_p - \vec{p}_0)\vec{v}_p + (\vec{x}_p - \vec{p}_0)^2 - r^2 = 0. \tag{1.21}$$

If a real, non-negative solution t_s for t exists, the photon will be set to the border of the new scattering cloud by the transformation $\vec{x} \to \vec{x} + t_s\vec{v}_p$. Any remaining pathlength of the photon inside the current scattering region is conserved and given to the *proceed routine* applied in the new scattering cloud. If there is no further scattering cloud on the way of the photon it is flagged to escape from the model space and registered.

1.3.6 The register routine

When a photon escapes from the models space it is recorded by one of the virtual detectors. Since the current version 1.2 of the code the detectors also allow for polarization imaging. The photons are thus recorded based on their wavelength and projected position on the plane of the sky.

We project the position of an escaping photon onto a distant plane, D, which is orthogonal to the line of sight (Fig. 1.4). The photon position P is determined by its distance, r_p, from the origin O of the model space and by the two angles θ_p and ϕ_p. The center of the projection plane O' is connected to O by the segment OO', which determines the polar and azimuthal viewing angles ϕ and θ. The distance between O’ and the projected photon position P' is denoted by the vector $\overrightarrow{q}$. By expressing $\overrightarrow{q}$ in the local frame of D, we obtain the projected photon coordinates x' and y' in the plane of the sky.

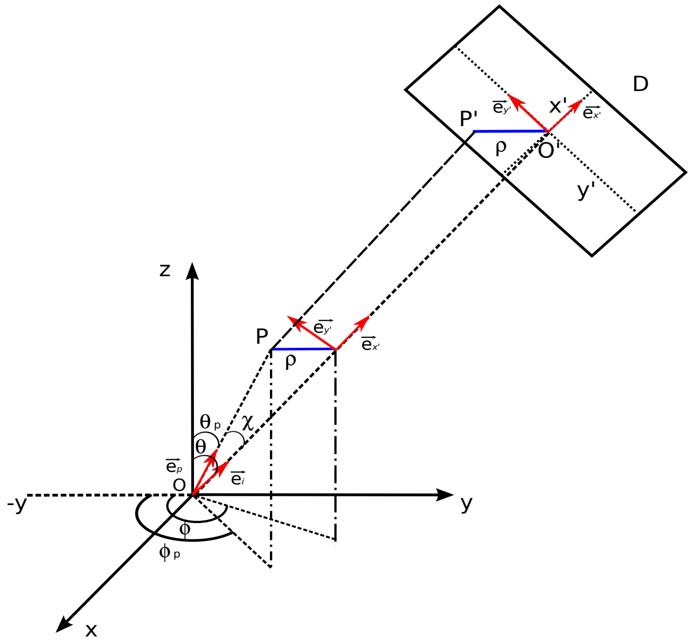
D
P'
$\overrightarrow{e_{y'}}$
x'
$\overrightarrow{e_{x'}}$
ρ
O'
y'
z
P
$\overrightarrow{e_{y'}}$
$\overrightarrow{e_{x'}}$
ρ
θ p
χ
θ
$\overrightarrow{e_p}$
O
$\overrightarrow{e_i}$
-y
y
φ
φ p
x

Figure 1.4: Spatial coordinates that determine the position of a photon P and its projection P' onto the plane of the sky, D.

The unit vectors $\overrightarrow{e_p}$ and $\overrightarrow{e_i}$ are expressed in spherical coordinates:

$$\overrightarrow{e_p} = \begin{pmatrix} \sin\phi_p \sin\theta_p \\ -\cos\phi_p \sin\theta_p \\ \cos\theta_p \end{pmatrix}, \tag{1.22}$$

$$\overrightarrow{e_i} = \begin{pmatrix} \sin\phi \sin i \\ -\cos\phi \sin i \\ \cos i \end{pmatrix}, \tag{1.23}$$

while the unit vectors of the observer's plane of the sky can be written as

$$\overrightarrow{e_{x'}} = \begin{pmatrix} \cos\phi \\ \sin\phi \\ 0 \end{pmatrix}, \tag{1.24}$$

$$\overrightarrow{e_{y'}} = \begin{pmatrix} -\sin\phi \cos i \\ \cos\phi \cos i \\ \sin i \end{pmatrix}. \tag{1.25}$$

We then construct the vector $\overrightarrow{\varrho} = r_p[\overrightarrow{e_p} - \cos\chi \overrightarrow{e_i}]$ and obtain its components in spherical coordinates:

$$\varrho_x = r_p[\sin\phi_p \sin\theta_p - \cos\chi \sin\phi \sin i], \tag{1.26}$$

$$\varrho_y = r_p[\cos\chi\cos\phi\sin i - \cos\phi_p\sin\theta_p], \tag{1.27}$$

$$\varrho_z = r_p[\cos\theta_p - \cos\chi\cos i], \tag{1.28}$$

with χ being the angle between $\overrightarrow{e_p}$ and $\overrightarrow{e_i}$. Using equations (1.24) and (1.25), we finally obtain the projected coordinates $x' = \overrightarrow{\varrho}\,\overrightarrow{e_{x'}}$ and $y' = \overrightarrow{\varrho}\,\overrightarrow{e_{y'}}$ of the photon in the plane of the sky:

$$x' = -r_p\sin\theta_p\sin(\phi-\phi_p), \tag{1.29}$$

$$y' = r_p[\sin i\cos\theta_p - \cos i\sin\theta_p\cos(\phi-\phi_p)]. \tag{1.30}$$

Before the photon is registered, it is necessary to rotate the polarization plane around the flight direction until it matches with the reference axis of the detector. Then, the Stokes vectors of all incoming photons within a given spatial and spectral bin can be added up to the values $\hat{I}$, $\hat{Q}$, $\hat{U}$ and $\hat{V}$. Finally, the net polarization properties as a function of wavelength and position are derived from:

$$\begin{aligned} P &= \frac{\sqrt{\hat{Q}^2+\hat{U}^2+\hat{V}^2}}{\hat{I}}, && (1.31)\\ \gamma &= \frac{1}{2}\arctan\frac{\hat{U}}{\hat{Q}}. && (1.32)\end{aligned}$$

Note that P represents the total polarization combining linear polarization $P_{\text{lin}} = \frac{\sqrt{\hat{Q}^2+\hat{U}^2}}{\hat{I}}$ and circular polarization $P_{\text{circ}} = \hat{V}/\hat{I}$. In the optical/UV domain, circular polarization can most often be neglected compared to linear one, otherwise it is necessary to adapt the output routines of the *ANALYZE* (see Sect. 2.2) program and to print either P_{lin} or P_{circ} instead of P.

1.4 Computation of dust properties

Mathis, Rumpl, & Nordsiek (1977) suggested dust compositions to reproduce extinction curves observed in our Galaxy. The authors assumed spherical dust grains having a size distribution proportional to a^s, with a being the grain radius and s a given index. Weingartner & Draine (2001) further developed the model by using carbonaceous and siliceous dust compositions to verify interstellar extinction in the Milky Way and in the Magellanic clouds. The dust compositions that can be modeled with *STOKES* are based on this work. The user can choose the minimum and maximum radii of the grain size distribution, its index s, and the abundances of graphite and silicate.

The Mie data, i.e. scattering and extinction cross sections, albedo, and entries of the scattering matrix, are computed using the code given by Bohren & Hufman (1983). We imported complex dielectric functions for graphite and silicate measured by Draine & Lee (1984). For graphite, two dielectric functions have to be considered since the optical properties for light polarized parallel and perpendicular to the crystals axis differ from each other. The program therefore works with two different kinds of graphite having abundances in a ratio of 1:2.

The code computes a weighted average of the dust composition and grain size distribution defined. The properties of the resulting "standard dust grain" are used by the *proceed* and the *scattering* routines of *STOKES*. In the following, we briefly summarize the specific equations and describe the routines used to compute the Mie data for user-defined dust compositions and grain size distributions.

Mie theory is based on the solution of the Maxwell equations for electromagnetic scattering off spheres having a homogeneous refractive index and dimensions in the range of the incoming wavelength. A fairly good description of the mathematical procedure was given by Bohren & Hufman (1983).

We show the results connected to the computation of the extinction cross section σ_{ext}, the scattering cross section σ_{sca}, and the scattering matrix elements $S_1(\theta)$ and $S_2(\theta)$ for a given substance, photon wavelength, and grain size radius.

$$
\begin{aligned}
\sigma_{\mathrm{sca}} &= \frac{2\pi}{k^2}\sum_{n=1}^{\infty}(2n+1)\left[|a_n(m,x)|^2+|b_n(m,x)|^2\right], && (1.33)\\
\sigma_{\mathrm{ext}} &= \frac{2\pi}{k^2}\sum_{n=1}^{\infty}(2n+1)\Re\left[a_n(m,x)+b_n(m,x)\right], && (1.34)\\
S_1(\cos\Theta) &= \frac{2\pi}{k^2}\sum_{n=1}^{\infty}\frac{2n+1}{n(n+1)}\left[a_n(m,x)\pi_n+b_n(m,x)\tau_n\right], && (1.35)\\
S_2(\cos\Theta) &= \frac{2\pi}{k^2}\sum_{n=1}^{\infty}\frac{2n+1}{n(n+1)}\left[a_n(m,x)\tau_n+b_n(m,x)\pi_n\right]. && (1.36)
\end{aligned}
$$

The complex functions π_n and τ_n are defined by a recursion that is based on the Legendre polynomials. The functions a_n, b_n take two arguments. The first argument is the relative refractive index m as defined by $m = \frac{N_1}{N}$, where N_1 and N are the complex refractive indexes inside and outside the particle respectively. The value N_1 contains the dielectric function of the grain substance considered. The second argument is the size parameter $x = \frac{2\pi N a}{\lambda}$ with the grain radius a and the wavelength λ. The functions are computed iteratively according to the method described in Wiscombe (1979, 1980).

The set of constants needed for *STOKES* contains the albedo and the entries of the Mueller matrix at each λ and a. They are derived from the data given above:

$$\begin{aligned}
albedo &= \frac{\sigma_{\mathrm{sca}}}{\sigma_{\mathrm{ext}}}, && (1.37)\\
S_{11} &= S_{22} = \frac{1}{2}\left[|S_2(\theta)|^2 + |S_1(\theta)|^2\right], && (1.38)\\
S_{12} &= \frac{1}{2}\left[|S_2(\theta)|^2 - |S_1(\theta)|^2\right], && (1.39)\\
S_{33} &= S_{44} = \frac{1}{2}\left[S_2^*(\theta)S_1(\theta) + S_2(\theta)S_1^*(\theta)\right], && (1.40)\\
S_{34} &= \frac{1}{2}\left[S_2^*(\theta)S_1(\theta) - S_2(\theta)S_1^*(\theta)\right]. && (1.41)
\end{aligned}$$

We denote the abundances of the various dust components by a row of constants g_l, with $l = 1, 2, 3, ...$ and $\sum g_l = 1$. Knowing the extinction cross-section, albedo, and Mueller matrix elements for all components and grain radii, the integrated absorption and scattering properties of the "standard dust grain" are obtained:

$$\begin{aligned}
\overline{\sigma_{\mathrm{ext}}} &= \frac{1}{K}\sum_l \int_{a_{\mathrm{min}}}^{a_{\mathrm{max}}} g_l\ \sigma_{\mathrm{ext}}\ a^s da, && (1.42)\\
\overline{albedo} &= \frac{1}{K}\sum_l \int_{a_{\mathrm{min}}}^{a_{\mathrm{max}}} g_l\ albedo\ a^s da, && (1.43)\\
\overline{S_{11}} &= \overline{S_{22}} = \frac{1}{K}\sum_l \int_{a_{\mathrm{min}}}^{a_{\mathrm{max}}} g_l\ S_{11}\ a^s da, && (1.44)\\
\overline{S_{12}} &= \frac{1}{K}\sum_l \int_{a_{\mathrm{min}}}^{a_{\mathrm{max}}} g_l\ S_{12}\ a^s da, && (1.45)\\
\overline{S_{33}} &= \overline{S_{44}} = \frac{1}{K}\sum_l \int_{a_{\mathrm{min}}}^{a_{\mathrm{max}}} g_l\ S_{33}\ a^s da, && (1.46)\\
\overline{S_{34}} &= \frac{1}{K}\sum_l \int_{a_{\mathrm{min}}}^{a_{\mathrm{max}}} g_l\ S_{34}\ a^s da, && (1.47)
\end{aligned}$$

with a_{min}, a_{max} being the limits and s being the index of the grain size distribution. The normalization constant K is calculated from:

$$K = \int_{a_{\mathrm{min}}}^{a_{\mathrm{max}}} a^s da. \qquad (1.48)$$

Chapter 2

How to run *STOKES*

Compiling the *STOKES* code requires a standard C^{++}-compiler available for Linux, MAC or Windows distributions. You may name the code after comilation including the version number. For a Linux system the comiler command simly is:

```
>> g++ -o lin-stokes-v1.2 *.cpp
```

The code requires a main input file called *input.par*. This file defines all parameters for a model run. If the model includes dust-filled scattering regions, *STOKES* either computes the dust properties or reads them from a file. Files containing dust models have the suffix *.dst*, they must be stored in the same directory as the *STOKES* executable file, and the file *input.par*.

Having your directory prepared with all necessary files, you launch the program. On a Linux system this naturally happens in a command line environment, while for Windows users it is more common to launch programs by clicking on the icon in a window. However, we would like to encourage Windows users to also execute *STOKES* from the command line, as it may print important messages. You make sure that these messages are visible if you run the program from the command line. The Linux command to run the code is simly:

```
>> ./lin-stokes-v1.2
```

When the execution is finished, the program produces four different files with the suffix *.sto*. These files contain the "raw" Stokes data of all virtual detectors. To obtain more convenient results, such as the fraction P of polarization and the position angle γ, you should launch the program *ANALYZE* in the same directory (see section 2.2).

In the following section 2.1 we explain in more detail how to construct the input file *input.par*. Then, in section 2.2, we discuss the output files of *STOKES* and their transformation by the auxiliary program *ANALYZE*. In the final sec-

Table 2.1: Maximum constants for the number of spectral, angular and spatial bins defined in the header file *model-stokes-v1.2.h* of the *model* class.

constant and default value	maximum number of...
`ResMax`= 50	spectral bins
`ScreenNumThetaMax`= 20	angular bins in the polar direction
`ScreenNumPhiMax`= 40	angular bins in the azimuthal direction
`ResXmax`= 30	spatial bins in x
`ResYmax`= 30	spatial bins in y

tion 2.3 we describe how to run *STOKES* simultaneously on several CPUs and how to collect and process the results.

IMPORTANT: The new capacity of *STOKES* to model spectro-imaging results in polarization comes with a significantly higher need in RAM during the execution of the program. The required memory is determined by the maximum number of spectral, angular and spatial bins as they are defined by the list of constants in the header file of the *model* class (see Table 2.1).

The product of all these constants times 8 kilobytes (to store the four Stokes parameters with double precision) estimates the minimum need of RAM required for a given setup of the virtual detectors. If you chose the maximum resolution in all categories indicated in Table 2.1, the model needs a minimum of 300 Megabyte in RAM still excluding the storage of the dust model and/or the various scattering regions. If you want to have a finer precision in any of the categories you may adjust the corresponding constant from Table 2.1 and recompile the code as the resolutions chosen in the *input.par* file can never exceed the limiting values in Table 2.1. Higher limiting constants come with a larger usage of RAM though and may slow down the computation significantly or even freeze your system. Of course, it is possible to optimize the memory usage by lowering some of the constants of Table 2.1. This may be the case when you want to run a model with high spatial but low spectral resolution (see the example 4 in Chapter 3).

2.1 The input file *input.par*

Construction of an input file for *STOKES* is rather simple. All entries in the file are based on keywords declared by the following pattern:

`keyword` *parameter1 parameter2 ...*

The keyword and the parameters are separated by blanks. Each keyword takes a well-defined number of parameters determining the specific model feature. All model parameters have a default setting that will be modified when the associated keyword appears in the input file. You can find a list of all keywords for *input.par* and their default settings in the tables 2.3 and 2.4 at the end of this chapter.

If you want to have comments in the input file you have to put a # at their beginning. The rest of the line will then be ignored. If you put the keyword `end` somewhere, *STOKES* finishes reading the input file at that position and all following entries are ignored. Note that the scanning of keywords in *input.par* is case-sensitive.

2.1.1 Setting general parameters

Before constructing the model space by defining emission and scattering regions, the general model parameters should be defined. It is recommended to change the default file name for the output files:

`OutputFile` *output-file-name CPU-number*

Setting *CPU number* different from 1 only makes sense if you want to perform calculations for the same model on several processors. Then, the input files on the various processors must have successive CPU numbers (see section 2.3). The numbers are later attached to the output file names.

It is further necessary to define how many viewing directions should be considered and which are wavelength and spatial range and resolution of the virtual detectors. If we assume a rectangular coordinate system with x-, y-, and z-axis, a viewing direction is given by its angle θ with the z-axis and by the azimuthal angle ϕ with the xz-plane. The number of viewing directions considered by *STOKES* is declared by the two commands:

`ThetaViewAng` *number-theta*
`PhiViewAng` *number-phi*

The viewing directions in θ are spaced in equal steps of $\cos\theta$, which, for an isotropic source at the origin of the model space, corresponds to equal fluxes at all viewing angles. Having a higher parameter *number-phi* is necessary if the model space is asymmetric with respect to the z-axis and/or if you want to obtain imaging results (see below). If *number-phi* equals 1 the photon flux will be summed over all azimuthal directions.

By default, the model setup is assumed to be symmetric with respect to the xy-plane, and the results of the upper and lower half-space are summed up to enable better statistics. If the model arrangement does not show this symmetry *STOKES* can take into account all viewing directions in θ by setting the *symmetry-flag*, which is a character string, to “no” (meaning *false*):

`PlaneSym` *plane-symmetry flag*

Note that with this new version 1.2 supports not only radial but also 3D velocities (see Sect. 2.1.2). Therefore, the symmetry with respect to the *xy*-plane is not only determined by a given emission and scattering geometry but also by the kinematics. A rotating disk centered on the *xy*-plane of the model space, for example, is not symetric with respect to the equatorial plane and requires the above flag to be set to "no".

At each viewing direction *STOKES* produces spectra over a range of wavelengths and with a spectral resolution declared by:

`LambdaMin` *min-wavelength*
`LambdaMax` *max-wavelength*
`SpectRes` *number-channels*

The spatial resolution in x and y is set by the two following commands:

`ResX` *x-resolution*
`ResY` *y-resolution*

The default value of these resoultions is unity, which suppresses the imaging function. The output files then only contain spectra. The projection of the model space onto the detector plane is sensitive to the number of viewing angles in θ and ϕ that you define. If the number of viewing angles is too low, the images will be averaged over too large of an angular range and thus contain distorsions. We have made good experience with minimum values of 10 and 40 for `ThetaViewAng` and `PhiViewAng` respectively.

For an axis-symmetric model that includes imaging the aforementioned requirement of 40 azimuthal viewing angles significantly slows down computation of sufficiently converged results. Therefore, it is possible to separately set a parameter indicating the axis-symmetry of the model:

`AxisSym` *axis-symmetry flag*

If this flag is set to "yes", the results are summed over all azimuthal angles despite a value of `PhiViewAng` greater than unity.

Finally, the total number of photons to be sampled is determined by the parameter *photon-number* and declared by the command:

`PhotonNum` *photon-number*

2.1.2 Implementation of 3D velocities

With the current version 1.2 of *STOKES* it is possible to include three-dimensional velocities of the emitting and scattering mmaterial. The parameterization of emission and scattering regions contains a new parameter, $v_{\rm type}$ followed by three velocity components. If $v_{\rm type}$ is set to 1 the velocity is given in cylindrical coordinates $(v_\rho, v_\phi, v_{\rm z})$, if it set to 0 spherical coordinates $(v_{\rm r}, v_\phi, v_\theta)$ are expected.

2.1.3 Defining emission regions

At its current state, *STOKES* offers three types of emission regions: a cylindrically shaped continuum source, a cylindrical broad emission line source, and a double-conical narrow emission line source. The difference between broad and narrow line region is rather formal and due to the fact that *STOKES* first was designed for radiative transfer in the unified scheme of Active Galactic Nuclei. In fact, the line width for both emission line regions can be defined in the same way, the difference between them is given by the geometry.

The cylindrical emission region for the continuum source is can have an offset with respect to the *xy*-plane of the coordinate system. The syntax for defining a continuum region is as follows:

`ContSource` $R\ a\ b\ c\ \alpha\ v_{\mathrm{type}}\ v_{\mathrm{r},\rho}\ v_\phi\ v_{\theta,z}$

For the geometrical denotations R, a, b, and c see Fig. 2.1a. The parameter α determines the spectral index of the continuum radiation following a power law $F(\nu) \propto \nu^{-\alpha}$. The continuum will be sampled between the borders *min-wavelength* and *max-wavelength.* You can define a 3D velocity (see Sect. 2.1.2) of the emitting material, which results in a wavelength shift of the emitted photons.

The cylindrical broad emission line region is declared by:

`BLSource` $R\ a\ b\ \lambda_0\ \Gamma$ strength $v_{\mathrm{type}}\ v_{\mathrm{r},\rho}\ v_\phi\ v_{\theta,z}$

The geometrical parameters R, a, and b are again explained in Fig. 2.1a. The line centroid is determined by λ_0, and Γ denotes the line width at the half-maximum. The parameter *strength* measures the percentage of all photons sampled (=*photon-number*) to be reserved for the line. This is a purely phenomenological definition of the line strength and does not involve any physics of the production of the emission lines. With regard to the continuum, *strength* should be adjusted by the user until the required relation between line and continuum is achieved. If you define a 3D velocity (see Sect. 2.1.2) of the emitting material, the photon wavelength will be shifted according to this motion.

The declaration of a double-conically shaped narrow emission line region is rather similar:

`NLSource` $r_1\ r_2\ \theta_0\ \lambda_0\ \Gamma$ strength $v_{\mathrm{type}}\ v_{\mathrm{r},\rho}\ v_\phi\ v_{\theta,z}$

The line characteristics are defined as for the broad line region, only the geometrical parameters are different (see Fig. 2.1b). The 3D velocity (see Sect. 2.1.2) of the emitting material produces a wavelength shift of the photon.

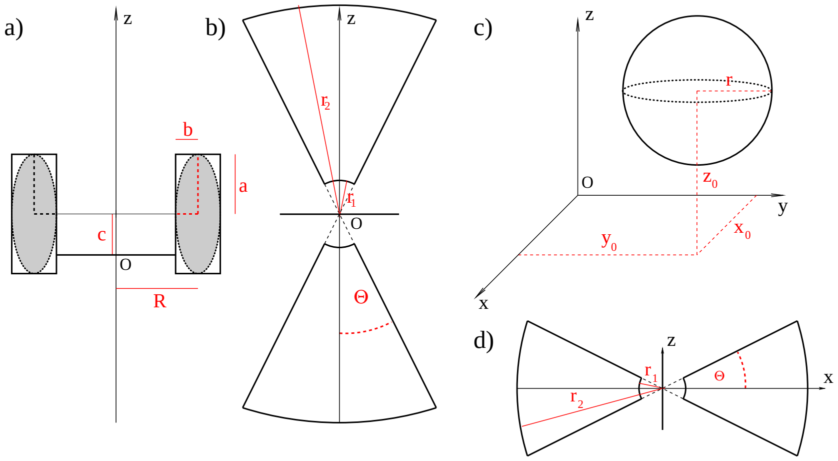

Figure 2.1: Geometries of emission and scattering regions in *STOKES*

2.1.4 Defining scattering regions

With this new version 1.2, *STOKES* supports five types of scattering regions: cylinders, tori, double-cones, flared disks, and single spheres. The double-cone, and the flared disk are centered on the origin of the coordinate system, while a torus or a cylinder can have a vertical offset. A sphere can be located anywhere in the model space. The double-cones have to be defined along the z-axis. All scattering regions can be declared to contain either dust or electrons. The material is denoted by the parameter *dust-flag*, which is a string of characters. If it is equal to the word "dust" the scattering region will by filled with dust, in any other case (like "electrons") it is filled with electrons. The density of the material is given by the parameter ρ. Hereby, ρ is a particle number density. The central-symmetric regions - cylinders, tori, and double-cones - can be attributed with a 3D velocity (see Sect. 2.1.2) of the material simulating an in- or outflow. The scattering events within these regions will consider the projected velocity of the scattering particles by a Doppler-shift of the wavelength. The maximum number of scattering regions allowed is 30.

For a cylindrical region the syntax is (geometrical parameters see Fig. 2.1a):

`Cylinder` R a b c dust-flag ρ $v_{\rm type}$ $v_{{\rm r},\rho}$ v_ϕ $v_{\theta,z}$

For a toroidal region the syntax is (geometrical parameters see Fig. 2.1a):

`Torus` R a b c dust-flag ρ $v_{\rm type}$ $v_{{\rm r},\rho}$ v_ϕ $v_{\theta,z}$

For a double-cone region the syntax is (geometrical parameters see Fig. 2.1b):

`Double-cone` r_1 r_2 θ dust-flag ρ $v_{\rm type}$ $v_{{\rm r},\rho}$ v_ϕ $v_{\theta,z}$

For a spherical region the syntax is (geometrical parameters see Fig. 2.1c):

`Sphere` x_0 y_0 z_0 dust-flag ρ

For a flared-disk region the syntax is (geometrical parameters see Fig. 2.1d):

`Flared-disk` r_1 r_2 θ dust-flag ρ $v_{\rm type}$ $v_{{\rm r},\rho}$ v_ϕ $v_{\theta,z}$

While conducting the radiative transfer, *STOKES* moves photons within the model space. At the borders of scattering regions the program has to decide if a photon is inside the cloud, in which case it possibly undergoes interaction with the scattering material. It turns out that this decision can cause numerical difficulties when the photon happens to be very close to the border of the scattering region. It is therefore necessary to clarify if the photon is inside or outside the scattering region, which is done by shifting it by an additional distance *dx* when the photon enters or leaves a scattering cloud. The parameter *gap* should be small compared to the expansion of the scattering regions and also small compared to their minimum mean free path length. If one obtains inconsistent results, especially if photons seem to penetrate through optically thick dust regions, it is worth trying to increase *dx*. To define the parameter *gap* use the following command:

`Gap` *dx*

A good choice for the gap parameter is 1% of the smallest geometrical dimension and mean free path length occurring in all scattering regions.

2.1.5 Defining a dust model

For the scattering regions filled with dust, a dust model has to be declared. The default dust model is based on a composition reproducing a standard Galactic extinction curve. A new dust model can be defined by changing the default parameters, the new dust properties are then calculated at the beginning of the program. For later use the dust model will be saved in a file on the disk. The declaration of a dust model is as follows:

`GrainRadMin` $r_{\rm min}$
`GrainRadMax` $r_{\rm min}$
`GrainSizeInd` α_s
`DustComp` *graph-orth graph-para silicate*
`GrainRadNum` $N_{\rm rad}$
`GrainLambdaNum` N_λ
`DustModel` *dust-file-name*

The program calculates the extinction properties of dust grains with a number distribution of grain radii r_s following a power law $n(r_s) \propto r^{\alpha_s}$. The limits of the grain radii and the grain size index are given by $r_{\rm min}$, $r_{\rm max}$, and α_s. The fraction of each dust component, is given by the percentages *graph-orth*, *graph-para*, and *silicate*. Since *graph-orth* and *graph-para* both refer to graphite but denote different polarization planes with respect to the crystal symmetry, *graph-para* must be twice the value of *graph-orth*. The sum of all three percentages must equal 100, of course. The dust composition and grain size distribution are integrated to produce an average dust grain used for the radiative transfer. The parameters N_{rad} and N_{λ} determine how many grain radii and wavelengths are considered for the integrations.

When *STOKES* is launched, it will first check if the dust model declared by *dust-file-name* already exists in the current work directory. If it does exist, the program reads in the dust data from the file. If no dust model is found, the Mie data is computed and then saved into the file *dust-file-name*.dst to be used again for other model runs.

The structure of a dust file starts out with the date and time of computation and with a list of the dust parameters as given in the declarations above. After a first delimiter, the wavelength dependence of the extinction, scattering, and absorption cross-section of an integrated dust grain are given in cm^2, followed by the albedo. Then, after a second delimiter, the wavelength dependence of the four Mueller matrix entries $\overline{S_{11}}$, $\overline{S_{12}}$, $\overline{S_{33}}$, and $\overline{S_{34}}$ for each scattering angle are listed.

2.2 Evaluating a simulation with *ANALYZE*

Each run of *STOKES* creates a set of four output files with the suffix *.sto*, one file for each Stokes parameter. The file name is given by the parameter *output-file-name* to which the parameter *CPU-number* is attached. The next character of the file name denotes the Stokes parameter I, Q, U, or V stored in the file. The construction of the output files is rather simple: the first line contains general information of the model run, like the name or the date and time of the simulation. The next line includes all model parameters. Then, following a delimiter, there is a series of data tables containing the Stokes parameters at all wavelengths, spatial pixels and viewing directions.

To be conveniently readable the output files of *STOKES* must be processed with the auxiliary program *ANALYZE*. It extracts various information from the *STOKES* output files and produces four new files with names ending on *...TF.res*, *...PO.res*, *...PA.res*, and *...PF.res*. The header of these files lists all model parameters and general information about the model run. Here you also find the number of photons sampled compared to the total number of photons defined for the run. This might be helpful if you check the intermediate results of a *STOKES* simulation that is still running. Below the header, the files contain the spectral and polarization results as given in Table 2.2.

Table 2.2: Content of the files produced by the auxiliary program *ANALYZE*.

spectral result	file suffix
Total flux F_λ	...TF.res
Fraction of polarization P	...PO.res
Polarization position angle	...PA.res
Polarized flux $P \times F_\lambda$	...PF.res

The viewing direction is denoted at the beginning of each line by a code of the kind "cosT.../P...". Herein "cosT" is followed by the cosine of θ and "P" marks ϕ given in degrees. If there is only one azimuthal viewing direction defined, the denotation of ϕ is suppressed. For the file *...PA.res* the position angle of the polarization vector is measured from a zero axis perpendicular to the z-axis of the model space's coordinate system. If the spatiale resolution in x or y is different from unity, each entry of the result files is preceded by the spatial coordinates of the pixel.

The program *ANALYZE* also produces a new set of result files with the suffix *.sto* but containing no CPU-number. The role of these files is related to a possible expansion of the model run and to running *STOKES* on multiple CPUs, which is explained in the next section 2.3.

IMPORTANT: As for the cas of the *STOKES* code, the new imaging capacity requires a significantly larger amount of RAM also during the application of *ANALYZE*. The limiting constants given in Table 2.1 and explained in the introduction of chapter 2 are valid also for the *ANALYZE* routine and the same considerations for the requited RAM hold true. For *ANALYZE*, you find the five constants limitng the number of spectral, angular and spatial bins in the header files *observe-analyze-v1.2.sto* and *detector-analyze-v1.2.sto*. Of course, you may eventually adjust them to the parameterization given in the *input.par* file and thereby optimize the memory usage.

2.3 Running *STOKES* on multiple CPUs

It is possible to perform simultaneous simulations with *STOKES* on several CPUs. Therefore, the program has to be copied to all machines in use, together with the input file *input.par* of the specified model. Then, the input files have to be attributed with successive CPU-numbers following the declaration of the output file name. As a result, the different copies of *STOKES* will produce output files with successive CPU-numbers that, after all runs are finished, have to be copied into one directory, together with the program *ANALYZE*. Running *ANALYZE* on the multiple output files will add up the Stokes parameters of the various runs and combine them into four new output files with the suffix

.sto. These new output files do not contain any CPU-number. In case you want to improve the Poissonian statistics and extend your model to a higher number of sampled photons, these combined result files can included in a larger set of model runs by attributing a CPU-number to their file names.

In addition to the combined result files *ANALYZE* produces the usual result files with the suffix *.res* (see previous section 2.2). They contain the flux and polarization data extracted from the multiple run.

Table 2.3: Commands to use in *input.par*, their default parameters, and eventual limits (part 1)

command	parameters	default parameters / limits	function
OutputFile	*output-file-name CPU-number*	empty-space 1	set output file name and number of the processor
ThetaViewAng	*number-theta*	10 (max = 80)	set number of viewing angles in i
PhiViewAng	*number-phi*	1 (max = 40)	set number of azimuthal viewing angles
PlaneSym	*symmetry-flag*	“yes”	indicate symmetry with respect to the xy-plane
LambdaMin	*min-wavelength*	1000 (min = 1000)	minimum continuum wavelength in Å
LambdaMax	*max-wavelength*	10000 (max = 10000)	maximum continuum wavelength in Å
SpectRes	*number-channels*	45 (max = 200)	number of channels for each detector
ResX	*x-resolution*	1 (max = 50)	horizontal detector resolution
ResY	*y-resolution*	1 (max = 50)	vertical detector resolution
PhotonNum	*photon-number*	1000000	number of photons to sample
ContSource	R a b c α v_{type} $v_{\mathrm{r},\rho}$ v_ϕ $v_{\theta,z}$	0 0 0 2 0	define continuum emission region (see Fig. 2.1a), set spectral index α and 3D velocity
BLSource	R a b λ_0 Γ strength v_{type} $v_{\mathrm{r},\rho}$ v_ϕ $v_{\theta,z}$	0.007 0.005 0.005 6563 150 0 0	define broad line region (see Fig. 2.1b), set line centroid, width (both in Å), relative strength (percentage), and 3D velocity
NLSource	r_1 r_2 θ_0 λ_0 Γ strength v_{type} $v_{\mathrm{r},\rho}$ v_ϕ $v_{\theta,z}$	10 100 25 0 6563 25 0 0	define narrow line region (see Fig. 2.1c), set line centroid, width (both in Å), relative strength (percentage), and 3D velocity
GrainRadMin	r_{min}	0.005 microns	minimum grain radius of the dust model
GrainRadMax	r_{min}	0.250 microns	maximum grain radius of the dust model
GrainSizeInd	α_s	-3.5	grain size index of the dust model
DustComp	*graph-orth graph-para silicate*	12.5 25.0 62.5	Abundances for the dust components (percentages); Note: *graph-para* = 2 × *graph-orth* and sum = 100
GrainRadNum	N_{rad}	100	number of grain radii considered for the dust model
GrainLambdaNum	N_λ	100	number of grain radii considered for the dust model
DustModel	*dust-file-name*	default	file name of the dust model

Table 2.4: Commands to use in *input.par*, their default parameters, and eventual limits (part 2)

command	parameters	default parameters / limits	function
`Cylinder`	R a b c "dust"/"electrons" ρ $v_{\rm type}$ $v_{{\rm r},\rho}$ v_ϕ $v_{\theta,z}$	not predefined	define cylindrical scattering region (see Fig. 2.1a), with dust/electron density ρ, and 3D velocity
`Torus`	R a b c "dust"/"electrons" ρ $v_{\rm type}$ $v_{{\rm r},\rho}$ v_ϕ $v_{\theta,z}$	not predefined	define toroidal scattering region (see Fig. 2.1a), with dust/electron density ρ, and 3D velocity
`Double-cone`	r_1 r_2 θ "dust"/"electrons" ρ $v_{\rm type}$ $v_{{\rm r},\rho}$ v_ϕ $v_{\theta,z}$	not predefined	define double-conical scattering region (see Fig. 2.1b), with dust/electron density ρ, and 3D velocity
`Flared-disk`	r_1 r_2 θ "dust"/"electrons" ρ $v_{\rm type}$ $v_{{\rm r},\rho}$ v_ϕ $v_{\theta,z}$	not predefined	define flared-disk scattering region (see Fig. 2.1d), with dust/electron density ρ, and 3D velocity
`Sphere`	x_0 y_0 z_0 "dust"/"electrons" ρ $v_{\rm type}$ $v_{{\rm r},\rho}$ v_ϕ $v_{\theta,z}$	not predefined	define spherical scattering region (see Fig. 2.1c), with dust/electron density ρ
`Gap`	dx	10^{-6} pc	computational parameter: photon displacement, must be small compared to dimensions and mean free path lengths of scattering regions

Chapter 3

Exemplary input files and dust models

If you want to familiarize yourself with the program and learn quickly how to run it, you might have a look at the following examples we provide. These examples also show how to parametrize two dust models that are relevant for astronomical applications (see sections 3.1 and 3.2). You can download the input files and the precomputed dust models from the web page featuring *STOKES*.

So far, the program computes all dust models for a wavelength range of 1000 Å to 10000 Å, as the optical data we adopted from Draine & Lee (1984) is given for this range. The accuracy of the model can be changed by choosing the parameters $N_{\rm rad}$ and N_λ denoting the number of wavelengths and grain radii at which the Mie data is computed. According to our experience, that dust models computed for $N_{\rm rad} = 200$ and $N_\lambda = 200$ are quite accurate, so we adopted these parameters for our precomputed models.

3.1 Example 1: electron disk and dusty torus

The first example is taken from an application of *STOKES* to modeling polarization in Active Galactic Nuclei. The input file sets up an emitting and scattering electron disk located within the funnel of an optically thick torus. The system is hence axis-symmetric and will only be evaluated at different polar viewing angles θ. A flat continuum spectrum within a wavelength range of 2000 Å to 8000 Å is considered.

The torus is filled with dust of the model *AGN*. This dust composition is based on modeling extinction curves of quasars as described in Gaskell et al. (2004). It contains graphite and silicate. The abundance for graphite is set to 15% and silicate contributes 85%. The grain size distribution $n(a)$ contains radii a between 0.005 μm and 0.200 μm and follows a power law with $n(a) \propto a^{-2.05}$.

In Fig. 3.1 we show the modeling results for the combined circular and linear degree of polarization P. The diagram can be constructed from the output file *Disk+TorusPO.res*. The wavelength dependence of P varies with the inclination of the object. As one looks at the object from an edge-on direction, e.g. at higher inclinations, the spectra are dominated by radiation scattered off the inner surface of the torus, the polarization rises with wavelength. Observations at a low inclination give a direct view at the electron disk and the polarization is dominated by wavelength independent Thomson scattering.

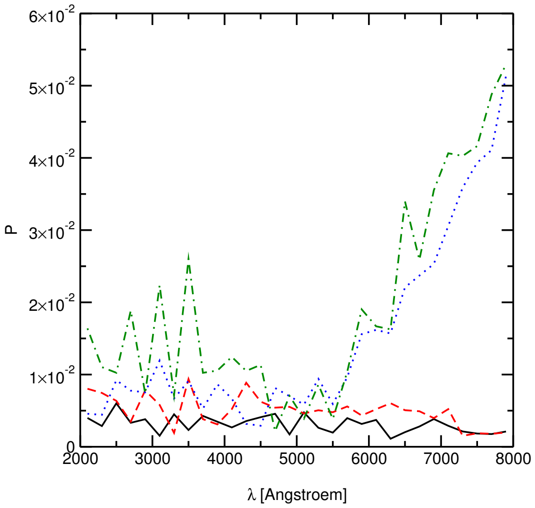

Figure 3.1: Polarization spectra at different inclinations for example 1. The curves denote the following inclinations i: $\cos i = 0.95$ (solid), $\cos i = 0.85$ (dashed), $\cos i = 0.75$ (dotted), and $\cos i = 0.65$ (dashed-dotted).

3.2 Example 2: electron disk and polar cones

In this example we model a system consisting of a disk with polar scattering cones and a surrounding broad line region. The emitting and scattering disk is supposed to be dominated by electrons. Along its symmetry axis we define a scattering double-cone filled with electrons and, further out, a dusty double-cone emitting a narrow Hα emission line at 6563 Å. Both cones have a radial outflow and an azimuthal velocity. A broad Hα emission line at 6563 Å comes from a rotating, cylindrical region around the electron disk. The dusty part of the conical scattering region is filled with dust of the model *MilkyWay*. It describes an average dust composition as observed from extinction measurements in our

Galaxy. The grain size distribution $n(a)$ follows a power law with $n(a) \propto a^{-3.5}$ over a range of 0.005 μm to 0.250 μm. We assume an abundance for silicate of 62.5% and for graphite of 37.5%. This grain composition and size distribution represents the Galactic standard extinction curve fairly well. It was used in the modeling of Fischer et al. (1994).

We show the results for the polarized flux in Fig. 3.2. The diagram can be constructed from the result file *Disk+ConesPF.res*. As the inclination increases toward an edge-on view, the line centroid, seen in polarized flux, shifts toward greater wavelengths. This is due to the radial outflow of the ionization cones. The angle between the velocity vector of the scattering material and the viewing direction increases toward 90°, hence the degree of polarization rises and the Doppler-redshift of the polarized broad line becomes more prominent.

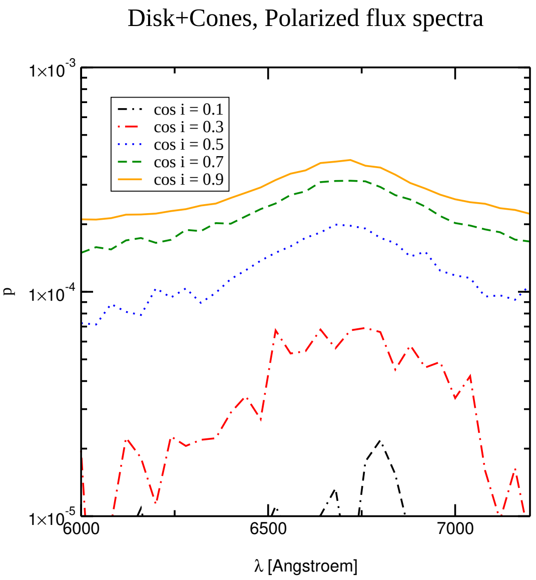

Figure 3.2: Polarized flux spectra at different inclinations for example 2. The curves denote the following inclinations i: $\cos i = 0.9$ (solid), $\cos i = 0.7$ (dashed), $\cos i = 0.5$ (dotted), $\cos i = 0.3$ (dashed-dotted), and $\cos i = 0.1$ (double dashed-dotted).

3.3 Example 3: electron shell and dust clouds

The third example features a setup that is not axis-symmetric. We assume a point-like narrow line region emitting an [OIII]-line at 5007 Å. This source is shielded by a rapidly expanding, highly ionized shell. Some nearby dust-clouds are distributed without a clear pattern around the system. The clouds slowly move along different azimuthal directions. The dust composition is similar to dust in the Milky Way and read from the file *MilkyWay.dst*. In order to conduct this simulation correctly the file *MilkyWay.dst* must first be computed from example 2 and then copied into the current working directory.

In Fig. 3.3 we show the results for the total flux seen at an edge-on inclination. The diagram can be constructed from the output file *Shell+CloudTF.res*. The asymmetry of the line profile is due to the surrounding dust clouds. They absorb a fraction of the light penetrating through the expanding shell or scatter it toward different directions.

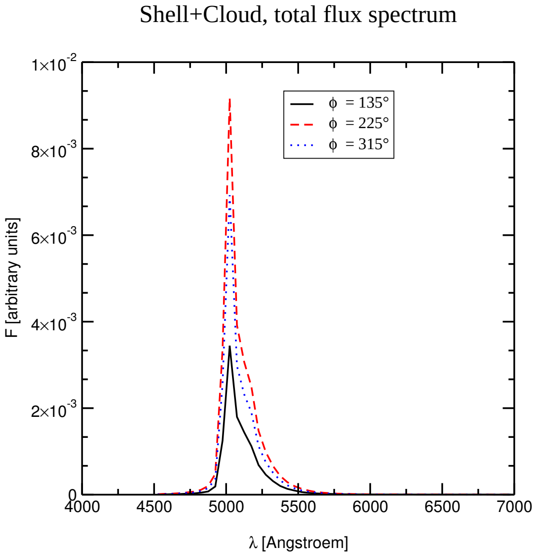

Figure 3.3: Total flux spectra for example 3 at three edge-on viewing directions. The curves denote the following azimuthal viewing angles: $\phi = 135$ (solid), $\phi = 225$ (dashed), and $\phi = 315$ (dotted).

3.4 Example 4: imaging an irradiated flared disk

In the fourth example we illustrate the new imaging capacities of *STOKES* Version 1.2. We chose the simple geometry of a centrally irradiated, equatorial disk with a flared shape. The disk extends between the radii of 0.0003 and 0.0005 pc and subtends an opening angle of 10° with respect to its equatorial plane. The scattering region is filled with electrons having a radial optical depth of unity for Thomson scattering. We do not extract any spectral information for this model and set the number of spectral bins to 1. This reduces the necessary RAM during the run and speeds up the computations.

The resulting polarziation map shown in Fig. 3.4 was obtained with an IDL imaging routine intially written by Michal Dovčiak and further developped by Frédéric Marin (see also Goosmann et al., 2013). The image combines the results for the polarized flux (file *Flared-Disk-ImagingPF.res*) given in color code with the spatial distribution of the polarization degree (file *Flared-Disk-ImagingPO.res*) and position angle (file *Flared-Disk-ImagingPA.res*) . The latter are denominated by the length of and the rotation angle of the black bar set in every pixel of the 30 × 30 resolution image.

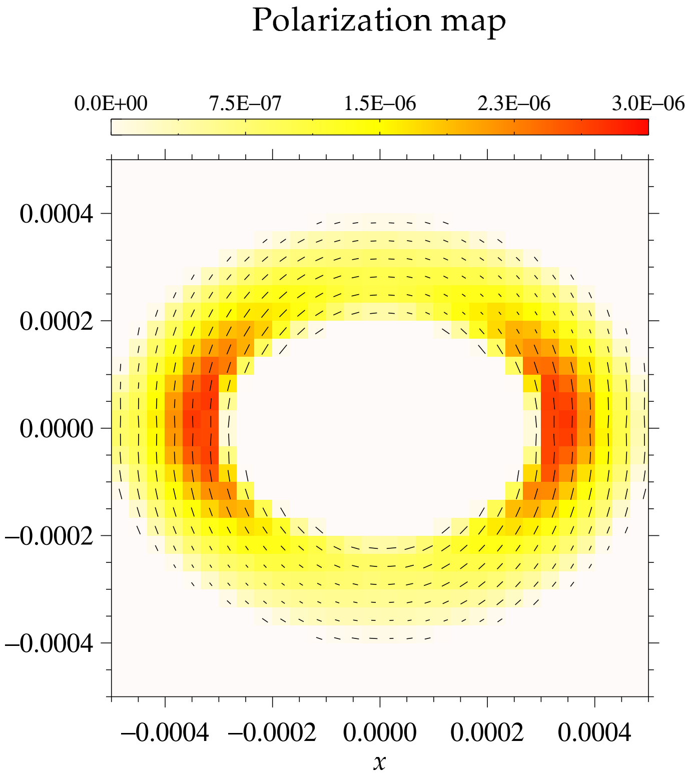

Figure 3.4: Combined image of the polarized flux, polarization information of a centrally irradiated equatorial disk observed at a viewing angle of ∼ 40° with respect to the disk axis. The polarized flux is color coded and for each pixel the position angle and degree of polarization are illustrated by the black bar's rotation angle and length, respectively.

Bibliography

Bohren C. F. & Hufman D. R. 1983, “Absorption and Scattering of light by small particles” (Wiley & Sons inc.)

Cashwell E. D. & Everett C. J. 1959, “Monte Carlo Method for random walk problems”, Pergamon Press

Draine, B. T., & Lee, H. M. 1984, ApJ, 285, 89

Fischer, O., Henning, T., & Yorke, H. W. 1994, A&A, 284, 187

Gaskell, C. M., Goosmann, R. W., Antonucci, R. R. J., & Whysong, D. H. 2004, ApJ, 616, 147

Goosmann, R. W., & Gaskell C. M. 2007, A&A, 465, 129

Goosmann, R. W., Gaskell, C. M., Marin, F., “Offaxis irradiation and the polarization of broad emission lines in active galactic nuclei”, proceedings of the 9th Serbian Conference on Spectral Line Shapes in Astrophysics, held in Banja Koviljaca, Serbia, May 1317, 2013. Accepted for publication in Advances in Space Research, arXiv 1311.2249

Gordon, K. D., Misselt, K. A., Witt, A. N., & Clayton, G. C. 2001, ApJ, 551, 269

Kartje, J. F. 1995, ApJ, 452, 565

Marin, F., Goosmann, R. W., Gaskell, C. M., Porquet, D.; Doviak, M., Modeling optical and UV polarization of AGNs. II. Polarization imaging and complex reprocessing, 548, 121

Mathis, J. S., Rumpl, W., & Nordsiek, K. H. 1977, ApJ, 217, 425

Misselt, K. A., Gordon, K. D., Clayton, G. C., & Wolff, M. J. 2001, ApJ, 551, 277

Watanabe, M., Nagata, T., Sato, S., Nakaya, H., & Hough, J. H. 2003, ApJ, 591, 714

Weingartner J. C., & Draine B. T. 2001, ApJ, 548, 296

Wiscombe, W. J. 1980, Appl. Opt., 19, 1505

Wiscombe, W. J. 1979, “Mie Scattering Calculations: Advances in Technique and Fast, Vector-Speed Computer Codes”, NCAR Technical Note TN-140+STR, 62 pp. Available from NTIS, Springfield, Virginia, 22161

Wolf, S., & Henning, T. 1999, A&A, 341, 675