

1989

Parameter estimation using VLA data

Willem C. Venter
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Astrophysics and Astronomy Commons](#), [Electrical and Computer Engineering Commons](#), and the [Statistics and Probability Commons](#)

Recommended Citation

Venter, Willem C., "Parameter estimation using VLA data " (1989). *Retrospective Theses and Dissertations*. 11171.
<https://lib.dr.iastate.edu/rtd/11171>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

90

14966

UMI

MICROFILMED 1990

INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600



Order Number 9014966

Parameter estimation using VLA data

Venter, Willem C., Ph.D.

Iowa State University, 1989

U·M·I

**300 N. Zeeb Rd.
Ann Arbor, MI 48106**



Parameter estimation using VLA data

by

Willem C. Venter

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department: Electrical Engineering and Computer Engineering
Major: Electrical Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1989

Copyright © Willem C. Venter, 1989. All rights reserved.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	xvi
ABSTRACT	xvii
1. BACKGROUND	1
1.1 Introduction	1
1.2 Astronomical Background	1
1.2.1 VLA (Very Large Array)	3
1.2.2 Data Generation	4
1.2.3 Data Observation	6
1.2.4 Problems in Observed Data	6
1.3 Estimation Background	7
1.3.1 Estimation Techniques	7
1.3.2 Parameter Estimation from Digitized Image Data	8
2. PROBLEM STATEMENT	10
2.1 Introduction	10
2.2 Flux Density Equation	12
2.2.1 Concept of Optical Depth	14

2.2.2	Temperature Model	16
2.3	Direct Solution	20
2.3.1	Description of Approach	25
2.4	Results of Direct Approach	30
2.5	Alternative Approach	31
3.	DOUBLE SMOOTHING APPROACH	33
3.1	Simulation Study	33
3.2	Simulation Results	39
3.3	Double Smoothing Technique	43
3.4	Application of the Double Smoothing Technique to the Observed Flux Density Values	52
3.4.1	Optical Depth τ_6 -map	52
3.4.2	Temperature Maps	56
3.5	Effect of a Different Smoothing Window	56
4.	STATISTICAL ANALYSIS	62
4.1	Introduction	62
4.2	Variance of Estimated Parameters	63
4.2.1	The Transformation of Variables	64
4.2.2	Application of the Central Limit Theorem for Linear Systems	65
4.2.3	Generation of System Moments	66
4.2.4	Monte Carlo Simulation	68
4.3	Double Smoothing Statistical Analysis	74
4.3.1	Flux Smoothing	74

4.3.2	Parameter Smoothing	74
4.4	Conclusion	75
5.	IMAGE RESTORATION	77
5.1	Introduction	77
5.2	Spatial Interaction Models	78
5.3	Image Description using SAR Models Defined on a Toroidal Lattice	79
5.4	Image Restoration using SAR Models	85
5.4.1	Image Restoration using an SAR model for the Degraded Image x	87
5.5	Estimation of SAR Model	92
5.5.1	Estimation of Neighbor Coefficient Parameters	92
5.5.2	Choice of Neighbor Set	93
5.6	Synthetic Image Generation	94
6.	RANDOM FIELD RESTORATION RESULTS	99
6.1	Synthetic Image Generation	99
6.2	SAR Model Parameter Estimation	102
6.3	Calculation of Test Statistics	102
6.4	Restoration of Noisy Image	106
6.5	Restoration of Noisy Simulated Flux Density Maps	109
7.	AN APPROXIMATE SOLUTION FOR NGC 7027 TEMPER- ATURE AND OPTICAL DEPTH PARAMETERS	121
7.1	Introduction	121
7.2	Approximate Solution for NGC 7027 Parameters	122

7.2.1	Determination of Emission Measure Map	122
7.2.2	Determination of Temperature T_1 -map	124
7.2.3	Determination of τ_6 -map	128
7.2.4	Determination of Temperature T_0 -map	130
7.3	Conclusion	132
7.4	Temperature Model 3	133
8.	AUTOMATIC MEASUREMENT OF RAINDROP DIAMETERS	134
8.1	Problem Statement	134
8.1.1	Approach	136
8.2	Distinguish between the Raindrops and the Background	136
8.2.1	Thresholding	136
8.2.2	Smoothing	137
8.2.3	Morphology	138
8.3	Distinguish between Individual Raindrops and Find their Actual Sizes	139
8.3.1	The modified Hough Transform	139
8.3.2	A Robust Method	146
9.	GENERAL PARAMETER ESTIMATION USING DIGITIZED	
	IMAGES	154
9.1	Introduction	154
9.2	Logarithmic Equations	155
9.2.1	Problem Statement	155
9.2.2	Parameter Estimation in Logarithmic Equations	156
9.2.3	Statistical Analysis	158

9.2.4 Conclusion	168
9.3 Polynomial Equations	168
9.3.1 Problem Statement	168
9.3.2 Parameter Estimation in Polynomial Equations	169
9.4 Conclusion	171
10. CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK	177
10.1 Conclusions	177
10.2 Suggestions for Future Work	178
11. BIBLIOGRAPHY	180
12. APPENDIX A	183
13. APPENDIX B	190

LIST OF TABLES

Table 6.1:	Synthetic generated image parameters	100
Table 6.2:	Maximum likelihood estimated parameters	103
Table 6.3:	Estimated parameters for image (4,1)	105
Table 6.4:	Evaluated parameters and statistics for image in Figure 6.2 .	108

LIST OF FIGURES

Figure 1.1:	Very Large Array	4
Figure 1.2:	Free-free emission	5
Figure 1.3:	A spectrum of the brightness distribution for free-free emission	5
Figure 2.1:	Postulated temperature models	18
Figure 2.2:	Observed flux density map at 2 cm	22
Figure 2.3:	Observed flux density map at 6 cm	23
Figure 2.4:	Observed flux density map at 20 cm	24
Figure 2.5:	Spectrum of flux density variation approach	32
Figure 3.1:	2-D derived temperature map for NGC 7027 by Basart and Daub (1987), temperature contours start at 9 000 K and increment by 500 K	34
Figure 3.2:	2-D derived optical depth map for NGC 7027 by Basart and Daub (1987), optical depth contours start at 0.2 and increment by 0.1	34
Figure 3.3:	T_0 simulated map	35
Figure 3.4:	T_1 simulated map	35
Figure 3.5:	τ_6 simulated map	35

Figure 3.6:	Simulated flux density map at 2 cm	37
Figure 3.7:	Simulated flux density map at 6 cm	37
Figure 3.8:	Simulated flux density map at 20 cm	37
Figure 3.9:	Noisy flux density image at 2 cm	38
Figure 3.10:	Noisy flux density image at 6 cm	38
Figure 3.11:	Noisy flux density image at 20 cm	38
Figure 3.12:	Cut through row 2 of T_0 maps with different amounts of noise	40
Figure 3.13:	Cut through row 16 of T_0 maps with different amounts of noise	40
Figure 3.14:	Cut through row 2 of T_1 maps with different amounts of noise	41
Figure 3.15:	Cut through row 16 of T_1 maps with different amounts of noise	41
Figure 3.16:	Cut through row 2 of τ_6 maps with different amounts of noise	42
Figure 3.17:	Cut through row 16 of τ_6 maps with different amounts of noise	42
Figure 3.18:	Cut through row 2 of T_0 maps with smoothed estimated pa- rameters	44
Figure 3.19:	Cut through row 16 of T_0 maps with smoothed estimated pa- rameters	44
Figure 3.20:	Cut through row 2 of T_1 maps with smoothed estimated pa- rameters	45
Figure 3.21:	Cut through row 16 of T_1 maps with smoothed estimated pa- rameters	45
Figure 3.22:	Cut through row 2 of τ_6 maps with smoothed estimated pa- rameters	46

Figure 3.23: Cut through row 16 of τ_6 maps with smoothed estimated parameters	46
Figure 3.24: Cut through row 2 of T_0 maps with smoothed noise and smoothed estimated parameters	49
Figure 3.25: Cut through row 16 of T_0 maps with smoothed noise and smoothed estimated parameters	49
Figure 3.26: Cut through row 2 of T_1 maps with smoothed noise and smoothed estimated parameters	50
Figure 3.27: Cut through row 16 of T_1 maps with smoothed noise and smoothed estimated parameters	50
Figure 3.28: Cut through row 2 of τ_6 maps with smoothed noise and smoothed estimated parameters	51
Figure 3.29: Cut through row 16 of τ_6 maps with smoothed noise and smoothed estimated parameters	51
Figure 3.30: T_0 estimated parameter 3-D map using observed data	53
Figure 3.31: T_0 estimated parameter contour map using observed data . .	53
Figure 3.32: T_1 estimated parameter 3-D map using observed data	54
Figure 3.33: T_1 estimated parameter contour map using observed data . .	54
Figure 3.34: τ_6 estimated parameter 3-D map using observed data	55
Figure 3.35: τ_6 estimated parameter contour map using observed data . .	55
Figure 3.36: T_0 estimated parameter 3-D map using observed data, 5×5 smoothing window used	58

Figure 3.37: T_0 estimated parameter contour map using observed data, 5×5 smoothing window used	58
Figure 3.38: T_1 estimated parameter 3-D map using observed data, 5×5 smoothing window used	59
Figure 3.39: T_1 estimated parameter contour map using observed data, 5×5 smoothing window used	59
Figure 3.40: τ_6 estimated parameter 3-D map using observed data, 5×5 smoothing window used	60
Figure 3.41: τ_6 estimated parameter contour map using observed data, 5×5 smoothing window used	60
Figure 4.1: Standard deviation of parameter T_0	72
Figure 4.2: Standard deviation of parameter T_1	72
Figure 4.3: Standard deviation of parameter τ_6	73
Figure 5.1: Degradation model	86
Figure 6.1: Synthetic generated images	99
Figure 6.2: Restoration of Noisy Image	107
Figure 6.3: 2 cm flux density maps	110
Figure 6.4: 6 cm flux density maps	111
Figure 6.5: 20 cm flux density maps	112
Figure 6.6: Cut through row 16 of T_0 maps showing the effect of random field restoration	114

Figure 6.7:	Cut through row 16 of T_1 maps showing the effect of random field restoration	114
Figure 6.8:	Cut through row 16 of T_0 maps showing the effect of random field restoration	115
Figure 6.9:	Cut through row 2 of T_0 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models	118
Figure 6.10:	Cut through row 16 of T_0 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models	118
Figure 6.11:	Cut through row 2 of T_1 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models	119
Figure 6.12:	Cut through row 16 of T_1 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models	119
Figure 6.13:	Cut through row 2 of τ_6 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models	120
Figure 6.14:	Cut through row 16 of τ_6 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models	120

Figure 7.1:	Emission measure contour map using approximate method and observed data	125
Figure 7.2:	Emission measure contour map obtained by Basart and Daub (1987)	125
Figure 7.3:	T_1 estimated parameter 3-D map using approximate method and observed data	127
Figure 7.4:	T_1 estimated parameter contour map using approximate method and observed data	127
Figure 7.5:	τ_6 estimated parameter 3-D map using approximate method and observed data	129
Figure 7.6:	τ_6 estimated parameter contour map using approximate method and observed data	129
Figure 7.7:	T_0 estimated parameter 3-D map using approximate method and observed data	131
Figure 7.8:	T_0 estimated parameter contour map using approximate method and observed data	131
Figure 8.1:	Raindrop image	135
Figure 8.2:	“Ghost” raindrops. The left two circles represent raindrops. The circle C is an artifact	142
Figure 8.3:	Simulated image	145
Figure 8.4:	Observed image	149
Figure 8.5:	Background removed by morphology	149
Figure 8.6:	Binary image after thresholding	150

Figure 8.7:	Generated image	150
Figure 8.8:	Raindrop with fitted circle	151
Figure 8.9:	Raindrop with false fitted circle	152
Figure 8.10:	Modified robust method	153
Figure 9.1:	Logarithmic X_1 simulated map	157
Figure 9.2:	Logarithmic X_2 simulated map	157
Figure 9.3:	Logarithmic X_3 simulated map	157
Figure 9.4:	Logarithmic Y_1 generated map	159
Figure 9.5:	Logarithmic Y_2 generated map	159
Figure 9.6:	Logarithmic Y_3 generated map	159
Figure 9.7:	Logarithmic noisy Y_1 map	160
Figure 9.8:	Logarithmic noisy Y_2 map	160
Figure 9.9:	Logarithmic noisy Y_3 map	160
Figure 9.10:	Cut through row 16 of X_1 estimated parameter maps, logarithmic equations used	161
Figure 9.11:	Cut through row 16 of X_2 estimated parameter maps, logarithmic equations used	161
Figure 9.12:	Cut through row 16 of X_3 estimated parameter maps, logarithmic equations used	162
Figure 9.13:	Standard deviation of X_4 estimated parameters using logarithmic equations	165
Figure 9.14:	Standard deviation of X_2 estimated parameters using logarithmic equations	165

Figure 9.15: Standard deviation of X_3 estimated parameters using logarithmic equations	166
Figure 9.16: Standard deviation of X_4 estimated parameters using logarithmic equations determined by double smoothing technique	166
Figure 9.17: Standard deviation of X_2 estimated parameters using logarithmic equations determined by double smoothing technique	167
Figure 9.18: Standard deviation of X_3 estimated parameters using logarithmic equations determined by double smoothing technique	167
Figure 9.19: Polynomial X_1 simulated map	172
Figure 9.20: Polynomial X_2 simulated map	172
Figure 9.21: Polynomial X_3 simulated map	172
Figure 9.22: Polynomial Y_1 generated map	173
Figure 9.23: Polynomial Y_2 generated map	173
Figure 9.24: Polynomial Y_3 generated map	173
Figure 9.25: Polynomial noisy Y_1 map	174
Figure 9.26: Polynomial noisy Y_2 map	174
Figure 9.27: Polynomial noisy Y_3 map	174
Figure 9.28: Cut through row 16 of X_1 estimated parameter maps, polynomial equations used	175
Figure 9.29: Cut through row 16 of X_2 estimated parameter maps, polynomial equations used	175
Figure 9.30: Cut through row 16 of X_3 estimated parameter maps, polynomial equations used	176

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my major professor, Dr. J. P. Basart, for giving me the opportunity to contribute this work. I would also like to thank my other committee members, Prof. W. Q. Meeker, Prof. D. W. Jacobson, Prof. S. F. Russell and Prof. H. S. Hung, for their helpful suggestions and comments to improve the quality of the work done in this dissertation.

It would not have been possible for me to undertake my studies in the U.S.A. without the moral and financial support from a number of people and institutions who believed in me. Special thanks go to the C.S.I.R. for their financial support as well as to Prof. H. F. J. Strydom and Mr. C. P. Bester who helped to make my studies financially possible. I am greatly in debt to Mr. G. J. Kahl for every thing he has done for me during the past three years. Gideon, I will never forget it.

I would like to use this opportunity to thank my parents and sister for their moral support, not only during the last three years, but since this whole journey started in 1976. Last, but definitely not least, I would like to express my love and gratitude to my wife for her contributions. There are too many to name, but staying awake with me through the many sleepless nights, meant more to me than anything else.

If there is any glory to give, let it be given to the Lord of heaven and earth who make everything possible and in whom I believe.

ABSTRACT

The main objective of this dissertation is to extract parameters from multiple wavelength images, on a pixel-to-pixel basis, when the images are corrupted with noise and a point spread function. The data used are from the field of radio astronomy. The very large array (VLA) at Socorro in New Mexico was used to observe planetary nebula NGC 7027 at three different wave lengths, 2 cm, 6 cm and 20 cm. A temperature model, describing the temperature variation in the nebula as a function of optical depth, is postulated. Mathematical expressions for the brightness distribution (flux density) of the nebula, at the three observed wavelengths, are obtained. Using these three equations and the three data values available, one from the observed flux density map at each wavelength, it is possible to solve for two temperature parameters and one optical depth parameter at each pixel location.

Due to the fact that the number of unknowns equal the number of equations available, estimation theory cannot be used to smooth any noise present in the data values. It was found that a direct solution of the three highly nonlinear flux density equations is very sensitive to noise in the data. Results obtained from solving for the three unknown parameters directly, as discussed above, were not physical realizable. This was partly due to the effect of incomplete sampling at the time when the data were gathered and to noise in the system.

The application of rigorous digital parameter estimation techniques result in estimated parameters that are also not physically realizable. The estimated values for the temperature parameters are for example either too high or negative, which is not physical possible. Simulation studies have shown that a "double smoothing" technique improves the results by a large margin. This technique consists of two parts: in the first part the original observed data are smoothed using a running window and in the second part a similar smoothing of the estimated parameters are done. This method provides an improvement over the previous method of directly solving the three nonlinear flux density equations when no adjacent pixel information was taken into account. When using the double smoothing technique, results were obtained that were not only physical realizable, but also compared well with previous results obtained from a two dimensional solution of the problem, assuming a constant temperature along the line of sight.

To conclude the investigation, an approximate solution was found for the same temperature and optical depth parameters. This solution takes into account approximations that can be made as a result of the physical characteristics of the nebula as well as the results obtained from the previous 2-D study.

1. BACKGROUND

1.1 Introduction

The objective of the research in this dissertation is to develop a method to estimate parameters from digitized images. Two different cases will be considered:

1. Estimation of parameters describing a physical object using only one digitized image.
2. Estimation of parameters describing a physical characteristic using multiple digitized images.

The second problem will be investigated first. This problem will be approached by trying to estimate the parameters describing the temperature variation in Planetary Nebula NGC 7027. This statement deserves a bit more attention.

1.2 Astronomical Background

Most of our information about the universe has been obtained from electromagnetic radiation. Before 1930 this information was restricted to the observation of visible light with wavelengths reaching from the near ultraviolet to the near infrared, $0.35 \mu m \leq \lambda \leq 1 \mu m$. Astronomers used optical telescopes to investigate these data

so this science was known as optical astronomy.

Since 1930 additional “windows” transparent to electromagnetic radiation of widely different wavelength ranges were opened in the blanket of the earth’s atmosphere (Rohlf, 1986). The radio window spanning from $\lambda \simeq 10 \text{ m}$ to $\lambda \simeq 1 \text{ mm}$ was the first new spectral range to become available to astronomy. Astronomers used radio telescopes to investigate the observations in the radio window so this science is known as Radio Astronomy.

A very important point must be emphasized here: The objects that emit the radiation in the different windows are the same. There is only one single world behind the radiation in the different windows although the aspect of the universe may differ from window to window. This is one of the fundamental axioms of science (Rohlf, 1986).

The angular resolution of a radio telescope is given by (Rohlf, 1986)

$$\delta \sim \frac{\lambda}{D} \quad (1.1)$$

where

δ = smallest angular separation between two point sources that are still recognizable as separate objects,

λ = radiation wavelength, and

D = diameter of telescope.

It is clear that the only way to improve the angular resolution at a specific wavelength is to increase the diameter D of the telescope. This solution is constrained by practical limits for the telescope size. These limits are in general in the range

25 – 100 *m*.

As an alternative, radio astronomers have improved resolving power by linking two or more telescopes into an interferometer arrangement (Rohlfs, 1986). In this case, D in equation (1.1) is the separation between the phase centers of the antennas when projected in the direction of the source.

As the science of astronomy progresses, it is important to evolve the methods of extraction of information from the data. In this dissertation we focus on parameter estimation from multiple data sets.

1.2.1 VLA (Very Large Array)

The data used in this investigation were observed with the Very Large Array (VLA) of the National Radio Astronomy Observation (NRAO) in New Mexico. The VLA consists of twenty-seven antennas arranged in a three-armed, equiangular Y-shaped array with nine antennas in each arm as shown in Figure 1.1. All twenty-seven antennas are paired with each other providing 351 combinations (Thompson et al., 1986). Variable antenna locations are possible with this setup and the four main configurations are known as the A, B, C and D configurations.

These configurations differ in terms of the length spanned by the antennas in the different arms and were designed to give almost the same resolution for observations at 1.35 cm, 2 cm, 6 cm and 20 cm for the A, B, C and D arrays, respectively. This arrangement is called a scaled array.

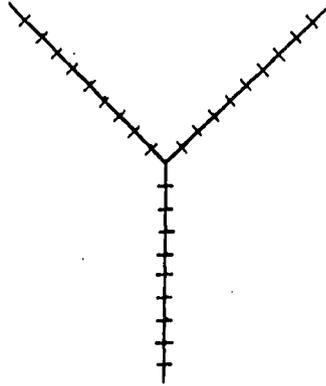


Figure 1.1: Very Large Array

1.2.2 Data Generation

The VLA was used to observe radio waves emitted from the nebula under investigation. These radio waves are generated in the nebula by a process called free-free emission (or bremsstrahlung), graphically depicted in Figure 1.2 (Rohlfs, 1986).

What is meant by free-free emission is the following. In the nebula there are lots of free electrons and free protons from hydrogen. As a free electron moves past a free proton, the electron de-accelerates and loses energy in the form of a radio wave. It is these radio waves that are being observed with the VLA at different frequencies. A spectrum of the brightness distribution for free-free emission is shown in Figure 1.3.

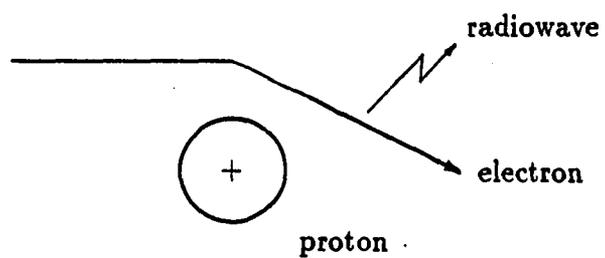


Figure 1.2: Free-free emission

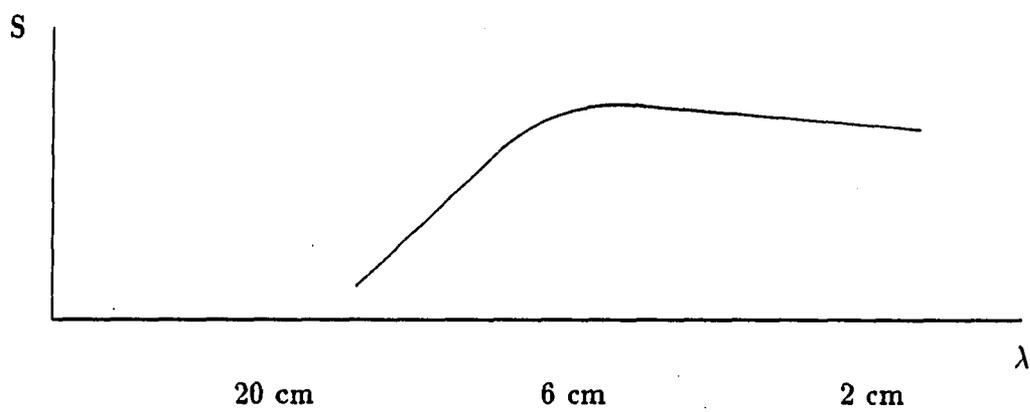


Figure 1.3: A spectrum of the brightness distribution for free-free emission

1.2.3 Data Observation

The data are observed with the VLA in the spatial frequency (u-v) domain. The principal mathematical procedure is to transform the data in the u-v domain to the x-y (spatial) domain by means of an inverse Fourier transform. The transformed data in the x-y domain gives an image of the nebula being observed.

1.2.4 Problems in Observed Data

There are certain problems inherent in the method used to observe the data which creates errors in the map in the spatial domain. These problems are:

1. The data measurements are noisy. There are two sources contributing to the noise in the observed data values:
 - (a) Noise due to the electronics used in the VLA used to observe the data.
 - (b) Atmospheric noise due to the interaction of the radio waves with the water vapor in the atmosphere.
2. Missing data values in the u-v domain cause large sidelobes in the spatial domain.
3. The time span of the observations are limited.
4. Synthesized beamwidths are not exactly the same.
5. The data values are finite in length.

1.3 Estimation Background

It would be wonderful if we could live in a perfect world where all data measurements were exact. However, in reality, all data measurements encountered in engineering and most branches of science are noisy. The need to obtain the best possible results for unknown parameters from these noisy data led to the development of what is today known as estimation theory. It is generally accepted that the progenitor of estimation theory was Karl Friederich Gauss, but R.A. Fisher, Norbert Wiener, Rudolph Kalman and many others also made valuable contributions (Mendel, 1987, p. xiii.)

1.3.1 Estimation Techniques

Many different estimation techniques have been developed through the years. Most of these techniques can be classified as either parameter estimation or signal (state) estimation, although there is an overlap in some situations. Two of the most important papers published in the area of state estimation were by Kalman (1960), and Kalman and Bucy (1961). The area of state estimation can be divided into mean-squared prediction, mean-squared filtering and mean-squared smoothing. All three of these operations can be implemented using a Kalman filter or modified Kalman filter. The interested reader is referred to Brown (1983) for a thorough discussion of these implementations. No wonder that the Kalman filter is even today the most popular way to do state estimation.

This dissertation, however, will primarily be concerned with the estimation of parameters from noisy data. The discipline of parameter estimation can be divided

into the estimation of deterministic parameters and the estimation of random parameters. Different techniques have been developed for each situation. Techniques used for deterministic parameter estimation include least squares (LS) estimation, best linear unbiased estimation (BLUE) and maximum likelihood (ML) estimation. Random parameter estimation is handled by techniques such as mean-squared estimation, maximum a posteriori estimation, BLUE and weighted least squares. All these different techniques and their interrelationships were covered in detail by Mendel (1987).

1.3.2 . Parameter Estimation from Digitized Image Data

In an overdetermined system (in which the number of data measurements exceeds the number of unknown parameters), all these parameter estimation techniques use the extra data values (the difference between the number of observations and the number of unknown parameters), under different criteria, to smooth the noise in the observation values to obtain more accurate estimates of the unknown parameters. However, when the number of equations equals the number of unknown parameters, there is no extra data available to smooth the noise in the observed values. All that can be done in this case is to solve the system of equations uniquely to fit the noisy observed values exactly.

A possible exception to the situation discussed in the previous paragraph, where the number of data values available equals the number of unknown parameters, is the solution of a system of equations using data values when these values are obtained from a digitized image. Due to the statistical relationship between data values at adjacent pixels in an image, the idea would be to use adjacent pixel data values to

smooth the data value at a specific pixel. The smoothed data values can then be used in the parameter estimation problem.

In this dissertation, the emphasis will be to estimate three parameters in three highly nonlinear equations, where the noisy data values in each equation are obtained from digitized images. Attempts have already been made in the literature to estimate parameters from digitized image data. An example is the estimation of 3-D skeletons and transverse areas of coronary arteries from biplane angiograms by Kitamura, Tobis and Sklansky (1988). However, a literature study did not produce any previous work done on the estimation of n parameters in n nonlinear equations (where n is small) on a pixel-to-pixel basis in an image. This was the motivation for the study undertaken.

A question that immediately comes to mind and must be addressed before the investigation is: "Why not obtain more data measurements and use the established theory to estimate the parameters?" The answer is that this is not always possible. In radio astronomy an objective often is to obtain the maximum information from the available data. This necessarily leads to the situation discussed above where the number of unknown parameters in a system of equations equals the number of data values available.

2. PROBLEM STATEMENT

2.1 Introduction

The objective of the study undertaken in the first investigation is to find a model describing the temperature variation in Planetary Nebula NGC 7027. This model will be used to estimate the surface temperature of the nebula as well as the variation in temperature along the line of sight. To be able to draw maps of the estimated temperatures as they vary in the nebula, these temperatures must be estimated on a pixel-to-pixel basis.

The way the study was done is as follows: Three flux density maps of the nebula were observed at 2 cm, 6 cm and 20 cm (see Basart and Daub, 1987 for discussion of 6 cm and 20 cm data). Each of these observed flux density maps is of size 32×28 pixels. A temperature model describing the temperature variation in the nebula along the line of sight (as a function of optical depth) was postulated. This temperature model was used to obtain theoretical flux density expressions at the three wavelengths used to observe the nebula.

The next step was to use the observed flux density values in the observed flux density maps to solve for the unknown parameters in the theoretical flux density equations on a pixel-to-pixel basis. In the end, estimated values for all the unknown

parameters in the postulated temperature model was calculated at every pixel in the 32×28 pixel maps based on the flux density value from the same pixel location in each of the three flux density maps.

It must be pointed out that the actual observed flux density maps were of size 512×512 pixels, but most of the nebula information was concentrated in only a small section of these observed maps. Contour plots drawn from the observed flux density maps showed that all the contours, down to a level of 0.5%, were contained in a section of the maps of size no larger than 32×28 pixels. These small sections were cut from the actual observed maps and will be referred to as the observed flux density maps from here on without any misunderstanding.

A similar study was previously done by Basart and Daub (1987) for the two-dimensional case. They used observations at 6 cm and 20 cm, obtained with the VLA, to derive two-dimensional distributions of temperature, emission measure and optical depth for the nebula. This were done by assuming a constant line of sight-temperature.

In the present study an extra observation of the nebula at 2 cm is available. The idea is to use the extra observation to expand on the work done by Basart and Daub (1987) to obtain information about the temperature variation in Planetary Nebula NGC 7027 along the line of sight.

To summarize, at this point, three flux density maps, observed at 2 cm, 6 cm and 20 cm with the VLA, are available. Each of these flux density maps are of size 32×28 pixels. The goal is to use three flux density values, one from each observation frequency, at the same pixel location, to find the parameters describing the temper-

ature variation and optical depth in the nebula at that specific pixel location. This must be done for all the pixels in the 32×28 pixel maps to obtain the estimated parameters as they vary over the whole nebula.

2.2 Flux Density Equation

A theoretical flux density equation, at each observation wavelength, can be obtained by starting with the general equation for flux density and deriving an expression for the observed flux density per beam for a model of the source.

The general observed radio flux density is given by

$$S = \int I_\nu(\theta, \phi) P_n(\theta, \phi) d\Omega \quad (2.1)$$

where

$I_\nu(\theta, \phi)$ = frequency dependent specific intensity of radiation,

$P_n(\theta, \phi)$ = normalized antenna power pattern,

$d\Omega = \sin\theta d\theta d\phi$ = incremental solid angle, and

θ and ϕ are the standard angular spherical coordinate variables.

Basart and Daub (1987) assumed an elliptical Gaussian form for the antenna beam with half power beam widths (HPBW's) of θ_x and θ_y along the two axes of symmetry. They used a normalized expression for this beam given by

$$P_n(\theta, \phi) = \exp[-4(\ln 2)\theta^2(\frac{\cos^2\phi}{\theta_x^2} + \frac{\sin^2\phi}{\theta_y^2})] \quad (2.2)$$

In addition, they assumed that θ_x and $\theta_y \ll$ the nebular angular diameter and due to the lack of a detailed nebular model, they also assumed that $I_\nu(\theta, \phi) = I_\nu$, a

constant over angle, within the beam of the antenna. (However, I_ν was not assumed to be constant over the source.) By taking into account that θ_x and θ_y are only a few arc seconds at most, the approximation $\sin\theta \approx \theta$ can be used.

By using all these approximations, the flux density within the synthesized beam is given by

$$\begin{aligned} S &= I_\nu \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \theta \exp[-4(\ln 2)\theta^2 (\frac{\cos^2\phi}{\theta_x^2} + \frac{\sin^2\phi}{\theta_y^2})] d\theta d\phi \\ &= \frac{\pi I_\nu \theta_x \theta_y}{4 \ln(2)} \end{aligned} \quad (2.3)$$

By using the usual expression, for radio frequencies, for the thermal emission I_ν , given by

$$I_\nu = \int_0^{\tau_\nu} B_\nu e^{-t_\nu} dt_\nu = \frac{2k\nu^2 e^{-\tau_\nu}}{c^2} \int_0^{\tau_\nu} T e^{t_\nu} dt_\nu \quad (2.4)$$

the expression for the flux is given by

$$S = \frac{\pi k \theta_x \theta_y e^{-\tau_\nu}}{2 \ln(2) \lambda^2} \int_0^{\tau_\nu} T e^{t_\nu} dt_\nu \quad (2.5)$$

where

S = the observed flux density,

k = Boltzman's constant,

θ_x = half power beam width in minor axis direction,

θ_y = half power beam width in major axis direction,

λ = observation wavelength,

t_ν = absorption from the back of the nebula to any point x along the line of sight,

τ_ν = optical depth, the total absorption from the back of the nebula to the distance s , and

T = temperature in the nebula.

In the derivation of equation (2.5) it was assumed that the optical depth is 0 at the back of the nebula. k , θ_x , θ_y and λ are known for each observed wavelength. The unknown parameters in the flux density equation are the optical depth of the nebula, τ_ν , and the parameters describing the temperature variation in the nebula, as a function of optical depth, and represented by the symbol T .

2.2.1 Concept of Optical Depth

The parameter optical depth used in the description of the nebula, is a measure of the absorption of the radio waves as it moves through the plasma. The more dense the plasma, the higher the absorption and consequently the higher the optical depth. The converse is also true.

A mathematical equation for optical depth is given by

$$\tau_\nu = \int_0^s \kappa_\nu dx \quad (2.6)$$

where

κ_ν = absorption coefficient (at a specific wavelength), and

x = distance variable

where s is the distance from the back of the nebula to any point within the nebula along the line-of-sight.

This equation therefore gives the relationship between the physical depth and the

optical depth of the nebula. If, for instance, the absorption coefficient is a constant throughout the nebula at a specific wavelength, then

$$\tau_{\nu} = \kappa_{\nu} \int_0^s dx = \kappa_{\nu} s \quad (2.7)$$

This indicates that if the absorption coefficient is a constant at a specific wavelength there exists a linear relationship between optical depth, τ_{ν} , and physical depth, s , in the nebula at the specific wavelength.

A physical interpretation of optical depth is as follows. When the optical depth is zero, $\tau_{\nu} = 0$, it is possible to see through the nebula; when the optical depth is infinite, $\tau_{\nu} = \infty$, it is not possible to see anything inside the nebula, only the front surface can be seen. Values of τ_{ν} in between allow different observation depths of the nebula.

The optical depth is a function of the wavelength used in the observation of the nebula. Osterbrock (1974) has shown that the optical depth at one wavelength is proportional to the optical depth at another wavelength. The relationship is given by

$$\tau_{\nu 1} = a_{\nu 1} \tau_{\nu 2} \quad (2.8)$$

where

$$a_{\nu 1} = \frac{\nu_{GHZ}^{2.1}(\tau_{\nu 2})}{\nu_{GHZ}^{2.1}(\tau_{\nu 1})} \quad (2.9)$$

and ν_{GHZ} is the observation frequency in GigaHertz.

The optical depth at 6 cm is chosen to be the standard from here on. All other optical depths will be referenced to this optical depth and the symbol used for the optical depth at 6 cm is τ_6 . Therefore

$$\tau_\nu = a_\nu \tau_6 \quad (2.10)$$

where

τ_ν = optical depth at any frequency, and

a_ν = constant.

The constants relating the optical depths at 2 cm and 20 cm to the optical depth at 6 cm are, respectively:

$$a_{02} = 0.09528$$

$$a_{20} = 12.544.$$

The observed flux density maps in this study are of size 32×28 pixels. The optical depth in the nebula will differ from pixel location to pixel location. Therefore, the optical depth, at each pixel location, is one of the unknown parameters in the flux density equations. As already mentioned, the optical depths at different wavelengths are related by a known constant. It is therefore only necessary to solve for the optical depth at 6 cm from the observed flux density values to have all the knowledge available about the optical depth behavior in the nebula.

2.2.2 Temperature Model

The second unknown in the flux density equations is the temperature model used to describe the temperature variation in the nebula as a function of optical depth (along the line of sight). Many different temperature models can be postulated. At least two parameters are necessary to define the temperature model, one to

describe the temperature of the nebula across the surface and the other to describe the temperature variation along the line of sight. The three models making most sense from a physical viewpoint are given in Figure 2.1. The equations describing the different models are also given.

The first model assumes a temperature distribution with highest temperature value at the back of the nebula and an exponential decrease in temperature with a linear increase in optical depth. The second model has an almost similar form, but the decrease in temperature is linear instead of exponential with a linear increase in optical depth. The third model assumes that the highest temperature is in the center of the nebula with a linear decrease in temperature to both sides.

Equation (2.5) can be used to obtain flux density expressions for the nebula using each temperature model. The flux densities, at an arbitrary wavelength, for all three models are given below:

Model 1:

$$S = K_1 \left\{ T_0 (1 - e^{-\tau\nu}) + \frac{T_1}{1-b} (e^{-b\tau\nu} - e^{-\tau\nu}) \right\} \quad (2.11)$$

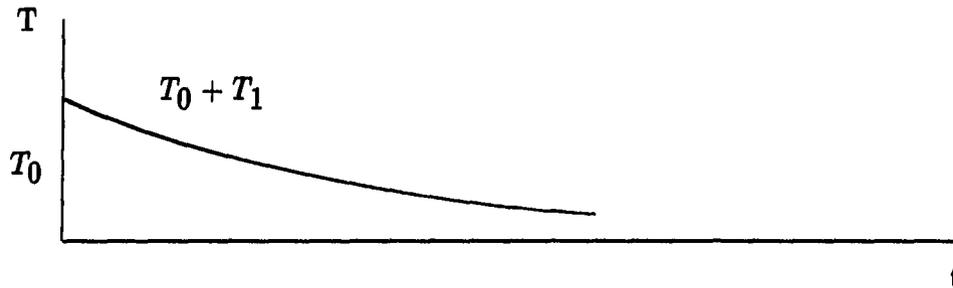
Model 2:

$$S = K_1 \left\{ T_0 (1 - e^{-\tau\nu}) - \frac{(T_0 - T_1)}{\tau\nu} (\tau\nu - 1 + e^{-\tau\nu}) \right\} \quad (2.12)$$

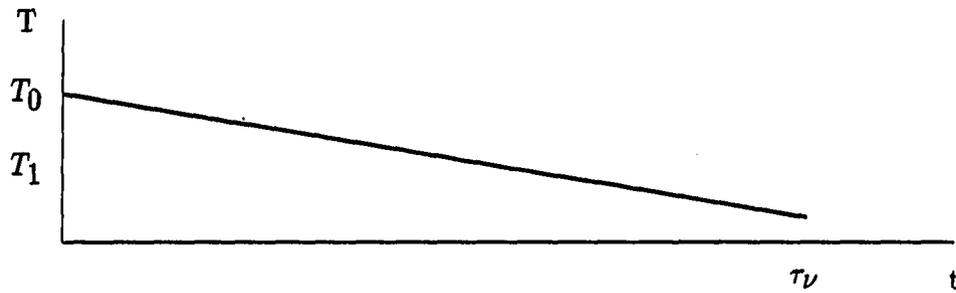
Model 3:

$$S = K_1 \left\{ T_1 \left[e^{-\tau\nu} \left(-1 - \frac{2}{\tau\nu} \right) + e^{\frac{-\tau\nu}{2}} \left(\frac{4}{\tau\nu} \right) + 1 - \frac{2}{\tau\nu} \right] + T_0 \left[\frac{2}{\tau\nu} e^{-\tau\nu} - \frac{4}{\tau\nu} e^{\frac{-\tau\nu}{2}} + \frac{2}{\tau\nu} \right] \right\} \quad (2.13)$$

Temperature Model 1: $T = T_0 + T_1 e^{-bt}$



Temperature Model 2: $T = T_0 - \left(\frac{T_0 - T_1}{\tau_\nu}\right)t$



Temperature Model 3: $T = T_0 - \left[\frac{2(T_0 - T_1)}{\tau_\nu}\right] \left|t - \frac{\tau_\nu}{2}\right|$

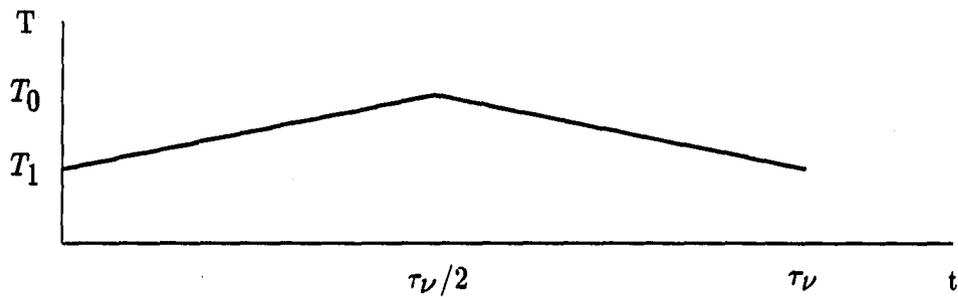


Figure 2.1: Postulated temperature models

where

$$K_1 = \frac{\pi k \theta_x \theta_y}{2(\ln 2)\lambda^2} \quad (2.14)$$

in all three cases as before. To apply any one of the three temperature models to the case under investigation, the equation for the flux density using that model must be evaluated at all three wavelengths, 2 cm, 6 cm and 20 cm, used to observe the nebula. This can be done by substituting the correct value for λ in each case and using the correct constant to relate the optical depth at the specific wavelength to the optical depth at 6 cm.

As can be seen, equations (2.11), (2.12) and (2.13) are all highly nonlinear in the unknown parameters, τ_ν , T_0 and T_1 . In other words, at each pixel in the 32×28 -pixel observed flux density maps, three highly nonlinear equations, evaluated at the three wavelengths used as already described, are available to be solved. In the case of model 1, there are four unknown parameters in the three equations and in the case of the other two models, there are three unknown parameters. At this stage, it must be pointed out that there exists no physical evidence to assume that one of the possible temperature models is more correct than any of the others. Actually, it is hoped that such information can be obtained from this investigation.

It is clear that there are four unknown parameters in the flux density equations obtained by using the first temperature model. Due to the fact that only three flux density maps, and therefore three flux density equations, are available at each pixel location, one of the four unknown parameters cannot be solved for in this case. Because not enough knowledge about the physical characteristics of the nebula are available to make an educated guess about one of the unknown parameters, it would

be best to use one of the other two models in the present investigation. It can also be seen that the flux density equation obtained by using temperature model 3 is much more complicated than the equivalent equation obtained by using temperature model 2. This is one of the reasons why temperature model 2 will first be investigated as a possible model describing the temperature variation in the nebula. Temperature model 3 will be investigated later.

2.3 Direct Solution

The contour plots of the three observed flux density maps made with the VLA are shown in Figures 2.2, 2.3 and 2.4. The 20, 6, and 2 cm observations were made with the A, B and C configurations, respectively. The double peak structure, often observed in radio maps of planetary nebulae, with reduced brightness in the center can be seen in the 6 cm map of NGC 7027, but the 20 cm map has a peak slightly northeast of the center which has no counterpart in the 6 cm map (Basart and Daub, 1987). The 2 cm map has the same structure as the 6 cm map with only a slightly higher peak flux density value of 0.263 Jy/beam instead of 0.232 Jy/beam. For more information on the structure of planetary nebulae, see Pottasch (1984), Aller (1984), the proceedings of the 131st symposium of the International Astronomical Union held in Mexico City (1987) and the Planetary Nebula Symposium no. 103 held in London, 1982 (Flower 1983).

The actual observed values for the flux densities at each pixel location, are given in Appendix A. By using temperature model 2 to describe the temperature variation in the nebula as a function of optical depth, the following three flux density equations

at the observed wavelengths are available to solve for the three unknown parameters:

$$S_{02} = K_{102} \{ T_0 (1 - e^{-a_{02}\tau_6}) - \frac{(T_0 - T_1)}{a_{02}\tau_6} (a_{02}\tau_6 - 1 + e^{-a_{02}\tau_6}) \}, \quad (2.15)$$

$$S_{06} = K_{106} \{ T_0 (1 - e^{-a_{06}\tau_6}) - \frac{(T_0 - T_1)}{a_{06}\tau_6} (a_{06}\tau_6 - 1 + e^{-a_{06}\tau_6}) \}, \quad (2.16)$$

and

$$S_{20} = K_{120} \{ T_0 (1 - e^{-a_{20}\tau_6}) - \frac{(T_0 - T_1)}{a_{20}\tau_6} (a_{20}\tau_6 - 1 + e^{-a_{20}\tau_6}) \}. \quad (2.17)$$

The subscripts 02, 06 and 20 in these three equations refer to the wavelengths for which the specific equations are valid. The definitions of the rest of the parameters are the same as before and the meaning of the constant K_{102} for instance is that it is the value of the constant K_1 , defined before, evaluated at 2 cm.

The most obvious approach to solve for the three unknown parameters in the three highly nonlinear equations, is to algebraically solve the three equations simultaneously at each pixel in the observed flux density maps. Due to the nonlinearity of the equations, ordinary linear algebra cannot be used to solve for the three unknown parameters directly. The approach followed was to linearize the nonlinear equations and to use iterative least squares estimation to solve for the three unknown parameters.

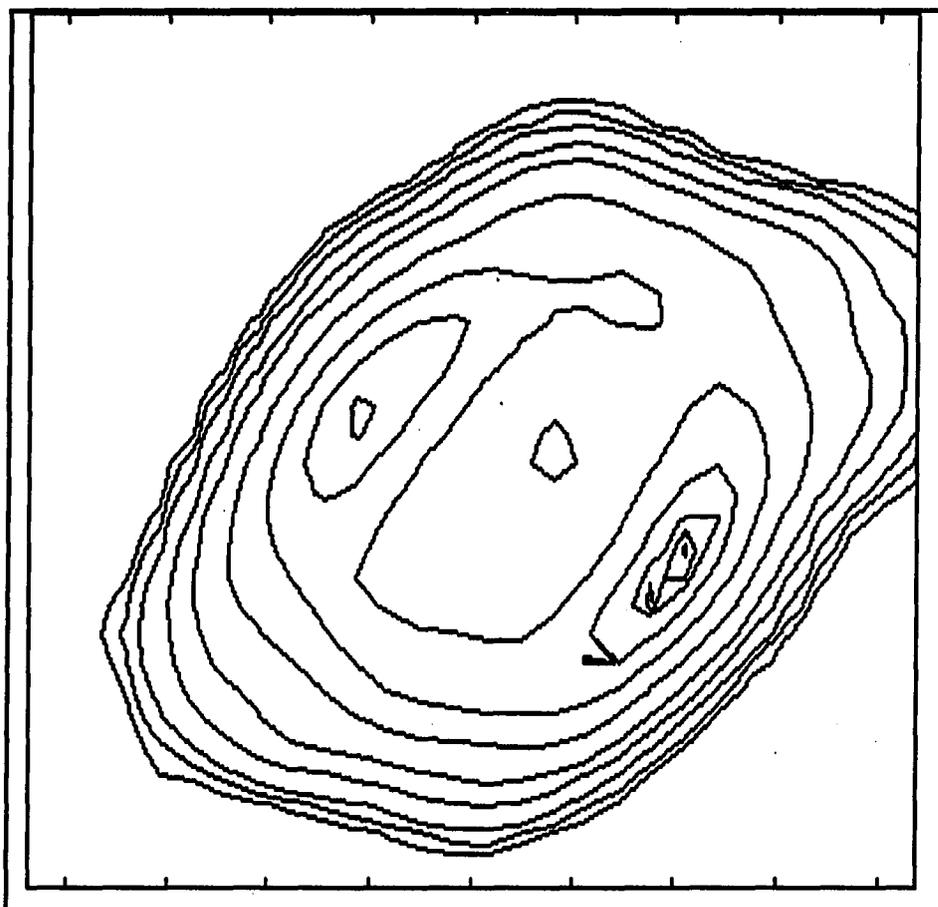


Figure 2.2: Observed flux density map at 2 cm

Peak value = 0.26316 Jy/beam

Contour levels at 99%, 95%, 90%, 75%, 50%, 25%, 10%, 5%, 2%, 1%, 0.5%

Scale: Each tick mark represents 0.25 seconds



Figure 2.3: Observed flux density map at 6 cm
Peak value = 0.23244 Jy/beam
Contour levels = 99%, 95%, 90%, 75%, 50%, 25%, 10%, 5%, 2%, 1%, 0.5%
Scale: Each tick mark represents 0.25 seconds

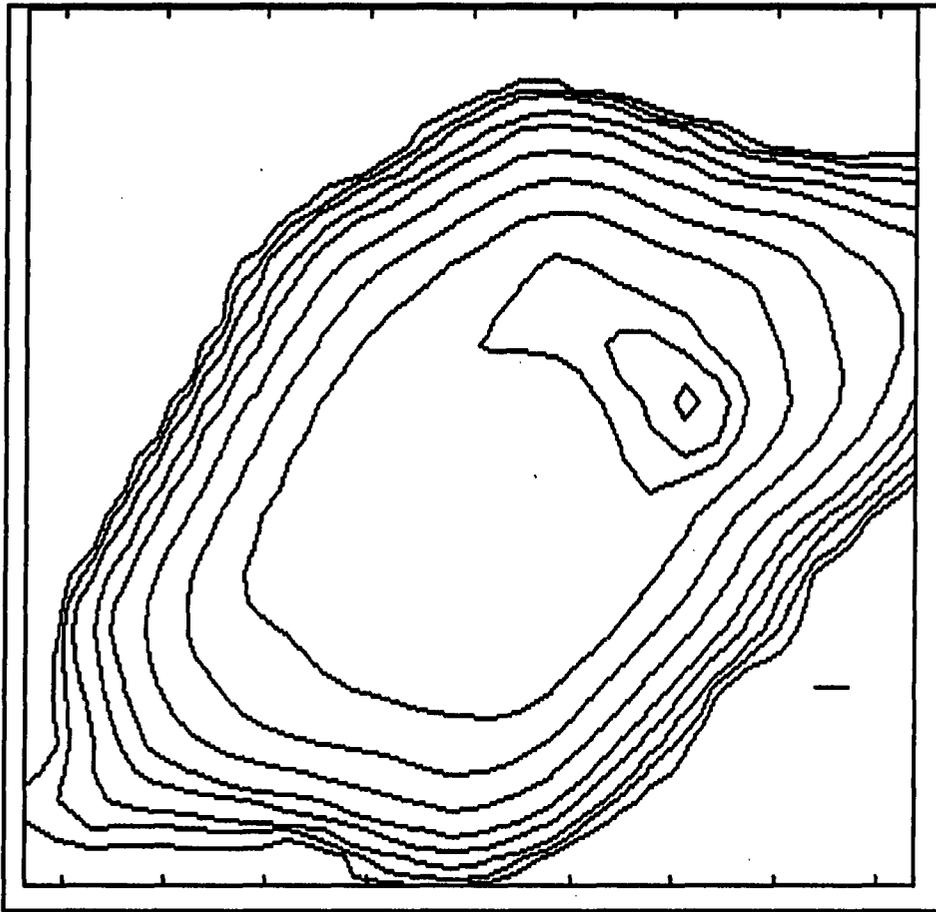


Figure 2.4: Observed flux density map at 20 cm
Peak value = 0.03984 Jy/beam
Contour levels at 99%, 95%, 90%, 75%, 50%, 25%, 10%, 5%, 2%, 1%, 0.5%
Scale: Each tick mark represents 0.25 seconds

2.3.1 Description of Approach

The basic idea of this approach is to linearize the three nonlinear flux density equations with respect to the three unknown parameters. These equations must be linearized around a normalization point. As an example, the original flux density equation at 2 cm is repeated here for clarity,

$$S_{02} = K_{102} \left\{ T_0 (1 - e^{-a_{02}\tau_6}) - \frac{(T_0 - T_1)}{a_{02}\tau_6} (a_{02}\tau_6 - 1 + e^{-a_{02}\tau_6}) \right\} \quad (2.18)$$

or

$$S_{02} = K_{102} \left\{ T_0 (1 - e^{-a_{02}\tau_6}) - (T_0 - T_1) \left(1 - \frac{1}{a_{02}\tau_6} + \frac{1}{a_{02}\tau_6} e^{-a_{02}\tau_6} \right) \right\} \quad (2.19)$$

Linearizing this equation with respect to the three unknown parameters will give

$$S_{02} \simeq S_{02}^* + s_{02} \quad (2.20)$$

where

S_{02}^* = flux density evaluated at the normalization point, and

$$s_{02} = \frac{\partial S_{02}}{\partial T_0} \Big|_* t_0 + \frac{\partial S_{02}}{\partial T_1} \Big|_* t_1 + \frac{\partial S_{02}}{\partial \tau_6} \Big|_* t_6 = L_1(\tau_2)t_0 + L_2(\tau_2)t_1 + L_3(\tau_2)t_6 \quad (2.21)$$

where * means linearization around the normalization point and the lowercase letters represent variation from the reference condition. Also

$$\frac{\partial S_{02}}{\partial T_0} \Big|_* = K_{102} \left(\frac{1}{a_{02}\tau_6^*} - \frac{1}{a_{02}\tau_6^*} e^{-a_{02}\tau_6^*} - e^{-a_{02}\tau_6^*} \right) = L_1(\tau_2) \quad (2.22)$$

$$\frac{\partial S_{02}}{\partial T_1} \Big|_* = K_{102} \left(1 - \frac{1}{a_{02}\tau_6^*} + \frac{1}{a_{02}\tau_6^*} e^{-a_{02}\tau_6^*} \right) = L_2(\tau_2) \quad (2.23)$$

$$\begin{aligned} \frac{\partial S_{02}}{\partial \tau_6} \Big|_* &= K_{102} T_0^* \left\{ -\frac{1}{a_{02}(\tau_6^*)^2} + \frac{1}{a_{02}(\tau_6^*)^2} e^{-a_{02}\tau_6^*} + \frac{1}{\tau_6^*} e^{-a_{02}\tau_6^*} + a_{02} e^{-a_{02}\tau_6^*} \right\} \\ &+ K_{102} T_1^* \left\{ \frac{1}{a_{02}(\tau_6^*)^2} - \frac{1}{a_{02}(\tau_6^*)^2} e^{-a_{02}\tau_6^*} - \frac{1}{\tau_6^*} e^{-a_{02}\tau_6^*} \right\} = L_3(\tau_2) \end{aligned} \quad (2.24)$$

Similar expressions can be obtained for the other two wavelengths of observation.

The problem can now be formulated in terms of the familiar linear model

$$\mathbf{Z}(k) = \mathbf{H}(k)\Theta + \mathbf{V}(k) \quad (2.25)$$

where

$\mathbf{Z}(k) = N \times 1$ measurement vector,

$\mathbf{H}(k) = N \times n$ observation matrix,

$\Theta = n \times 1$ parameters vector, and

$\mathbf{V}(k) = N \times 1$ measurement noise vector, as follows

$$\begin{bmatrix} s(\tau_{20}) \\ s(\tau_6) \\ s(\tau_2) \end{bmatrix} = \begin{bmatrix} L_1(\tau_{20}) & L_2(\tau_{20}) & L_3(\tau_{20}) \\ L_1(\tau_6) & L_2(\tau_6) & L_3(\tau_6) \\ L_1(\tau_2) & L_2(\tau_2) & L_3(\tau_2) \end{bmatrix} \begin{bmatrix} t_0 \\ t_1 \\ t_6 \end{bmatrix} + \begin{bmatrix} v(\tau_{20}) \\ v(\tau_6) \\ v(\tau_2) \end{bmatrix} \quad (2.26)$$

The least squares solution to this problem is given by Mendel (1987, p.19) as

$$\hat{\Theta}_{LS}(k) = [\mathbf{H}'(k)\mathbf{H}(k)]^{-1}\mathbf{H}(k)\mathbf{Z}(k) \quad (2.27)$$

Because $V(k)$ cannot be measured, it is neglected in the least squares solution.

It must be pointed out that the solution of this equation is fraught with numerical difficulties, in particular the evaluation of

$$[\mathbf{H}'(k)\mathbf{H}(k)]^{-1} \quad (2.28)$$

An example is in order.

Example:

The following \mathbf{H} matrix was obtained at a specific pixel location:

$$\mathbf{H} = \begin{bmatrix} 0.2759203E - 005 & 0.2199522E - 006 & 0.2468661E - 005 \\ 0.1937950E - 004 & 0.8101084E - 005 & 0.2255684E + 000 \\ 0.2507618E - 004 & 0.1233902E - 004 & 0.4780105E + 000 \end{bmatrix} \quad (2.29)$$

By using the standard procedure (single precision arithmetic)

$$[\mathbf{H}'(k)\mathbf{H}(k)]^{-1} = \frac{1}{\det([\mathbf{H}'(k)\mathbf{H}(k)])} \text{adj}(\mathbf{H}'(k)\mathbf{H}(k)) \quad (2.30)$$

to calculate the inversion, the following results were obtained:

$$\mathbf{H}^{-1} = \begin{bmatrix} 0.4917572E + 006 & -0.4745371E + 005 & 0.2239052E + 005 \\ -0.1628728E + 007 & 0.5954964E + 006 & -0.2810008E + 006 \\ 0.1624548E + 002 & -0.1288233E + 002 & 0.8168208E + 001 \end{bmatrix} \quad (2.31)$$

The correct solution as calculated by using an HP 41 C calculator equipped with a math-pack, is given by

$$\mathbf{H}^{-1} = \begin{bmatrix} 0.4924199E + 006 & -0.4752298E + 005 & 0.2242308E + 005 \\ -0.1630930E + 007 & 0.5962995E + 006 & -0.2813794E + 006 \\ 0.1626756E + 002 & -0.1289942E + 002 & 0.8179027E + 001 \end{bmatrix} \quad (2.32)$$

This solution was verified by calculating $\mathbf{H}\mathbf{H}^{-1}$, and to five significant digits the result agreed with the identity matrix.

It can be seen that the standard procedure results in values that are on the average 0.13% too low in this example.

A much better approach to invert the given \mathbf{H} matrix is to use the method of LU Decomposition. This method, also using single precision arithmetic, results in the following inverse matrix:

$$\mathbf{H}^{-1} = \begin{bmatrix} 0.4923947E + 006 & -0.4751427E + 005 & 0.2242076E + 005 \\ -0.1630840E + 007 & 0.5962703E + 006 & -0.2813663E + 006 \\ 0.1626654E + 002 & -0.1289907E + 002 & 0.8178872E + 001 \end{bmatrix} \quad (2.33)$$

These values are on the average 0.0066% lower than the correct values, quite an improvement over the standard procedure. The LU Decomposition method, using double precision arithmetic, results in the following inverse matrix:

$$\mathbf{H}^{-1} = \begin{bmatrix} 0.4924198E + 006 & -0.4752294E + 005 & 0.2242306E + 005 \\ -0.1630929E + 007 & 0.5962992E + 006 & -0.2813793E + 006 \\ 0.1626754E + 002 & -0.1289941E + 002 & 0.8179023E + 001 \end{bmatrix} \quad (2.34)$$

These values are on the average 0.0007% lower than the correct values.

One of the reasons for the numerical difficulties is because the data values in the original matrix differ by as much as 6 orders of magnitude. The interested reader is referred to an article by Laub (1985) on this topic.

It was decided to use the method of LU Decomposition as given in Numerical Recipes (Press et al., 1986) and double precision arithmetic to handle all the numerical work in this dissertation.

The normalization point for the first iteration can be calculated by using knowledge about the physical characteristics of the nebula to make certain approximations in the different observed flux density equations. It is known that the nebula is optically thick at 20 cm and optically thin at 2 cm. What is meant by this is that at 20 cm

$$\tau_\nu \gg 1 \quad (2.35)$$

while at 2 cm

$$\tau_\nu \ll 1 \quad (2.36)$$

It was found that the iterative least squares approach is very sensitive in how close the first normalization point is to the actual solution. When both the above approximations were used to find this point, it turned out that the approach is unstable at many pixel locations because the normalized values were not close enough to the actual solutions. In the end only the approximation in equation (2.35) was used.

Taking this approximation into account, the flux density equation at this wavelength can be simplified as follows

$$S_{20} \simeq K_{120} \left\{ T_1 \left(1 - \frac{1}{a_{20} \tau_6} \right) + T_0 \left(\frac{1}{a_{20} \tau_6} - e^{-a_{20} \tau_6} \right) \right\} \quad (2.37)$$

Substituting this expression in the other two expressions, the unknown parameters can be solved for uniquely.

These approximate values were used as the normalization values in the first step of the iterative least squares approach. The new solutions were used as the normalization values in the second step and so forth. It was found that the approach converged after the third or fourth iteration to the correct solution at all the pixels in the 32×28 pixel maps.

2.4 Results of Direct Approach

The numerical solutions for the three unknown parameters in the three nonlinear equations could be found very accurately at all the pixel locations in the 32×28 observed maps by using the above approach. This was verified by using the estimated values for T_0 , T_1 and τ_6 in the three flux density equations at the three observed wavelengths at each pixel location to calculate the original values of the observed flux density values. However, one problem with the estimated parameters was that they were not acceptable at many of the pixel locations. The reason why they could not be accepted was because the estimated parameters were not physically possible. It is known from general radio astronomy theory and previous studies done on this nebula, that the total temperature (as given by the postulated temperature model), as well as the optical depth for this whole nebula, must be within certain bounds. For example, it is obvious that the total temperature at no pixel location can be negative. It is also known that the optical depth over the whole nebula varies between 0.2 and 5.0. These constraints were broken by the solutions obtained at many of the pixels using

the direct approach.

The reason the solutions are not physically acceptable at many of the pixels in the observed flux density maps can be contributed to inaccuracies in the observed flux density values. These inaccuracies were already discussed in detail in Chapter 1.

2.5 Alternative Approach

One possible approach to take the inaccuracies in the observed flux density values into account and to find an acceptable solution at each pixel in the observed flux density maps for the unknown parameters, is to let the observed flux density values at each pixel vary with an amount that seems to be a reasonable estimate for the inaccuracies. This amount was chosen to be $\pm 10\%$ of the observed flux density values at each pixel in each of the three observed flux density maps at the three different wavelengths. The new approach was then to solve for the unknown parameters, using the direct solution discussed above, at each pixel using all the possible flux density values in all three flux density maps at the specific pixel location. This situation is depicted graphically in Figure 2.5.

The idea behind this approach was the following: as the flux density values at each pixel location in all three flux density maps, approach their "true" values, it was expected that acceptable solutions would be obtained that were the actual correct solutions. Further it was expected that as the flux density values diverge from their "true" values, the estimated parameters would vary only a small amount around their correct values while large variations from the "true" values of the flux density would result in nonphysical solutions. If this assumption was true, it would be easy to find

the “correct” estimated parameters at each pixel location.

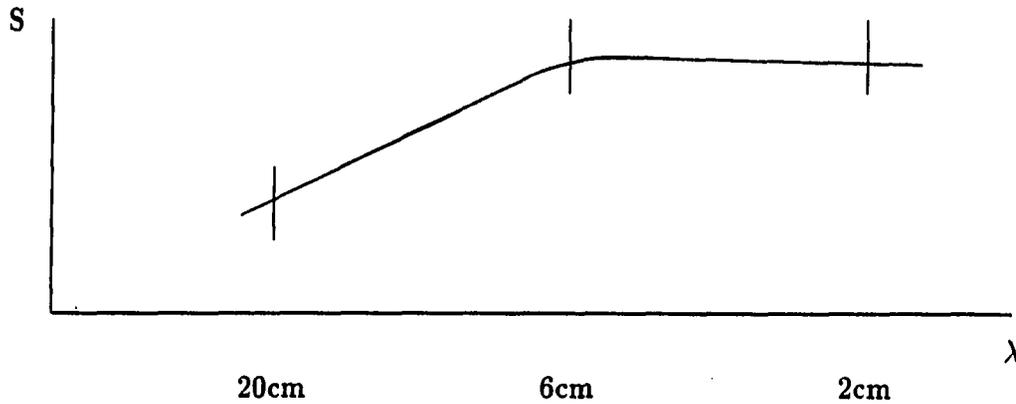


Figure 2.5: Spectrum of flux density variation approach

It turned out that this approach was not the solution to the problem. When the flux density values at each pixel location in all three flux density maps were varied in steps of 2%, there were 11^3 solutions possible at each pixel location. Although many of the solutions could be rejected because they were not physically acceptable, the rest were all physically acceptable but they did not vary around the “correct” value as expected. They spanned the whole range of acceptable values at each pixel location and there was no way to choose the “correct” solution from all the possibilities.

Although this method did not provide a solution to the main problem, it provides important information about the behavior of the estimated parameters. It was found that the estimated parameters were very sensitive to small changes in the flux density values used to solve for these parameters. This sensitivity will be further investigated in the next chapter.

3. DOUBLE SMOOTHING APPROACH

3.1 Simulation Study

In the previous chapter it was discovered that the estimated parameters are very sensitive to inaccuracies in the observed flux density values. To investigate exactly how sensitive, a thorough simulation study was undertaken. The approach was as follows:

1. Two temperature maps and one optical depth map, each of size 32×28 pixels, were simulated. The values for the unknown parameters in these maps were chosen according to the expected values for these parameters as obtained by the 2-D study by Basart and Daub (1987). The derived maps for temperature and optical depth, obtained by Basart and Daub (1987) for the 2-D case, are shown in Figures 3.1 and 3.2. The simulated maps are shown in Figures 3.3, 3.4 and 3.5.
2. The simulated temperature maps and optical depth map were used to obtain three simulated flux density maps, at 2 cm, 6 cm and 20 cm. The values of the flux densities at each pixel in the 32×28 pixel maps were calculated using the same formula as derived in Chapter 2.

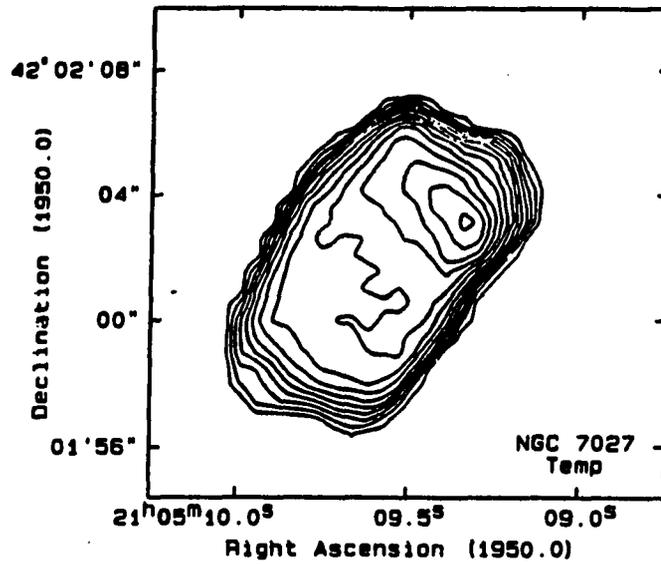


Figure 3.1: 2-D derived temperature map for NGC 7027 by Basart and Daub (1987), temperature contours start at 9 000 K and increment by 500 K

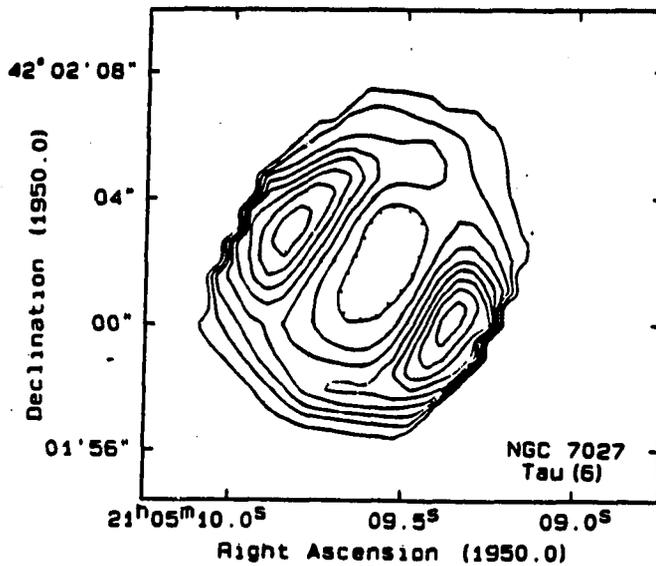
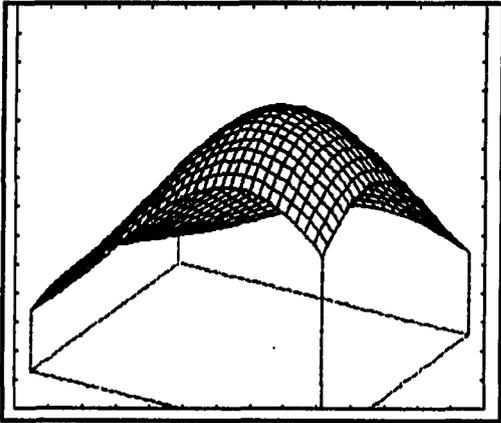
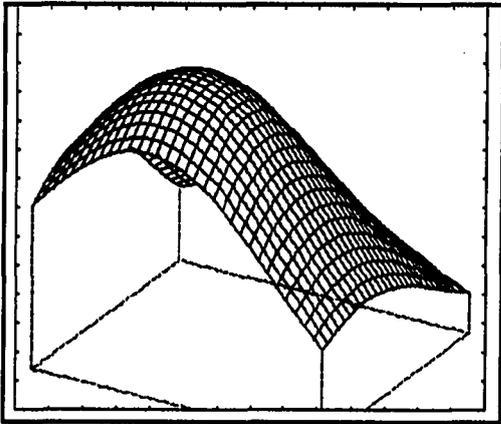


Figure 3.2: 2-D derived optical depth map for NGC 7027 by Basart and Daub (1987), optical depth contours start at 0.2 and increment by 0.1



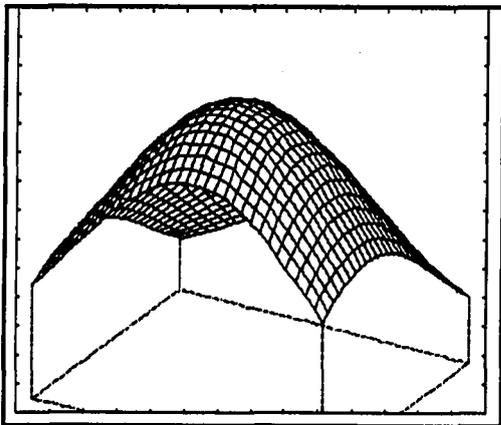
Maximum value = 15 000 K
Ground level value = 0 K

Figure 3.3: T_0 simulated map



Maximum value = 5 000 K
Ground level value = 0 K

Figure 3.4: T_1 simulated map



Maximum value = 1.4
Ground level value = 0.0

Figure 3.5: τ_6 simulated map

$$S = K_1 e^{-\tau\nu} \int_0^{\tau\nu} T e^t dt \quad (3.1)$$

where

S is the flux density,

$K_1 = \text{constant}$,

$\tau\nu = \text{optical depth}$, and

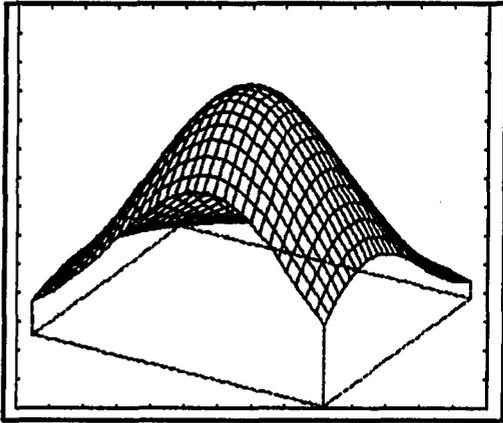
T is the temperature model used.

The temperature model used in the simulation study was the same model postulated in Chapter 2 namely

$$T = T_0 - \left(\frac{T_0 - T_1}{\tau\nu} \right) t \quad (3.2)$$

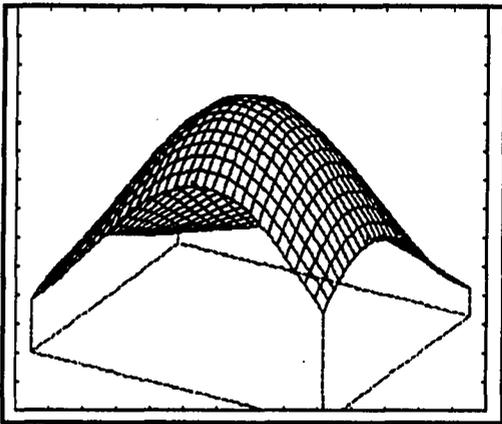
The simulated flux density maps are shown in Figures 3.6, 3.7 and 3.8.

3. Zero mean white Gaussian noise with a standard deviation of 10% of the flux density value at the individual pixels was added to each pixel in all three flux density maps. The new flux density maps with the noise added will be referred to as the "noisy" flux density maps and are shown in Figures 3.9, 3.10 and 3.11.
4. The "noisy" flux density maps were used to estimate the original two temperature parameters and the optical depth parameter. These estimated parameters were then compared with the actual temperature and optical depth parameters used in the simulation to find exactly how sensitive the estimated parameters were to inaccuracies in the flux density values.



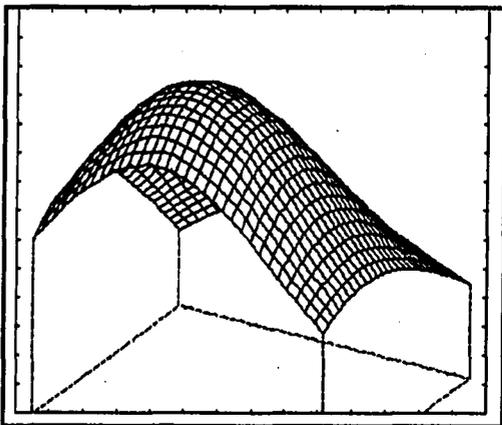
Maximum value = 0.3210 Jy/beam
Ground level value = 0.0 Jy/beam

Figure 3.6: Simulated flux density map at 2 cm



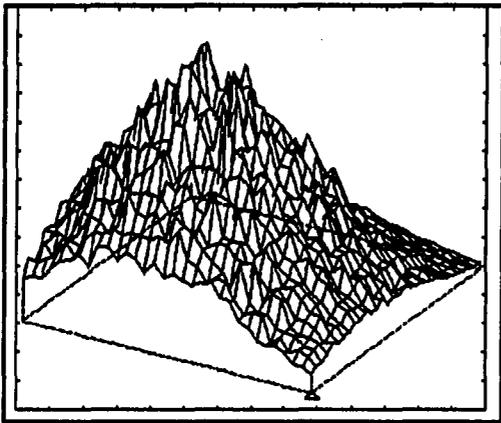
Maximum value = 0.1920 Jy/beam
Ground level value = 0.0 Jy/beam

Figure 3.7: Simulated flux density map at 6 cm



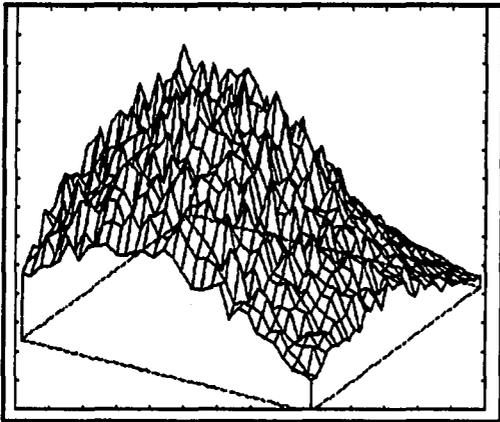
Maximum value = 0.0149 Jy/beam
Ground level value = 0.0 Jy/beam

Figure 3.8: Simulated flux density map at 20 cm



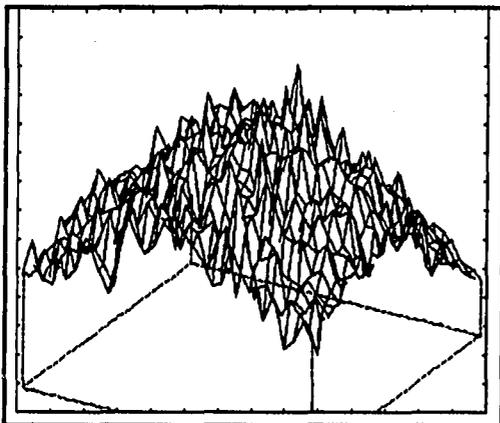
Maximum value = 0.3950 Jy/beam
Ground level value = 0.0 Jy/beam

Figure 3.9: Noisy flux density image at 2 cm



Maximum value = 0.2290 Jy/beam
Ground level value = 0.0 Jy/beam

Figure 3.10: Noisy flux density image at 6 cm



Maximum value = 0.0183 Jy/beam
Ground level value = 0.0 Jy/beam

Figure 3.11: Noisy flux density image at 20 cm

3.2 Simulation Results

Simulated noisy flux density maps were generated for three different situations. In the first case, 0% noise was added to each pixel in the three generated flux density maps, in the second case zero mean white Gaussian noise with a standard deviation of 5% of the individual pixel values was added to each pixel of the three flux density maps and in the third case zero mean white Gaussian noise with a standard deviation of 10% of the individual pixel values was added to each pixel in the three flux density maps. Each of the three sets of noisy flux density maps was used to estimate the parameters T_0 , T_1 and τ_6 at each pixel. The results are displayed as follows: Cuts were made through the estimated parameter maps at different rows for each of the noise situations discussed before. The results at the same row for all three cases were superimposed on one figure. The results for row 2 and row 16 were typical of the results for all the rows and were arbitrarily chosen to be displayed here for all three estimated parameters. These results are shown in Figures 3.12, 3.13, 3.14, 3.15, 3.16 and 3.17.

From these figures it can be seen that the estimated parameters T_0 , T_1 and τ_6 have similar behavior in each of the situations discussed above. In the case of 0% noise added to the original simulated flux density maps, the estimated parameters at each pixel location are exactly the same as the original parameters used to generate the flux density maps. Both these curves are labelled "0% noise" in Figures 3.12, 3.13, 3.14, 3.15, 3.16 and 3.17 and can be used as a reference curve.

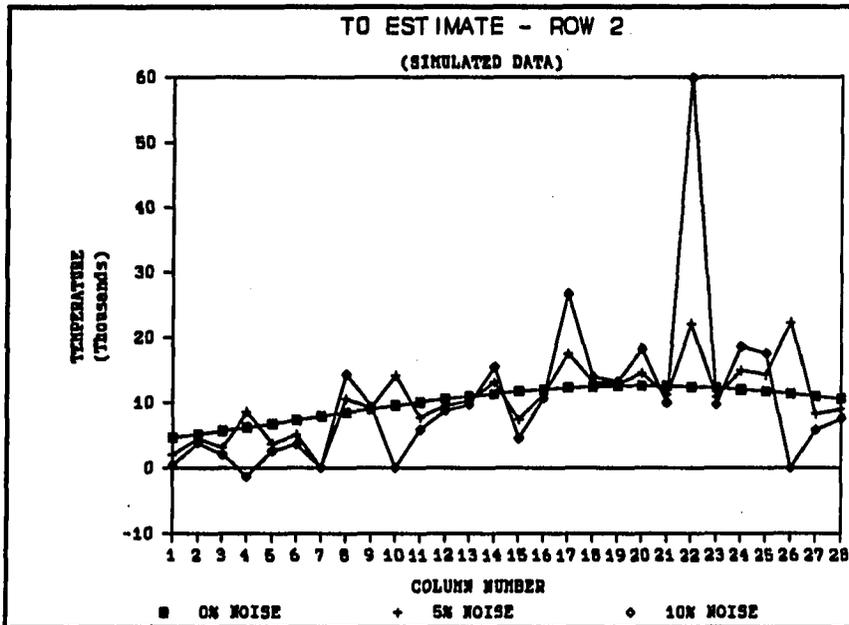


Figure 3.12: Cut through row 2 of T_0 maps with different amounts of noise

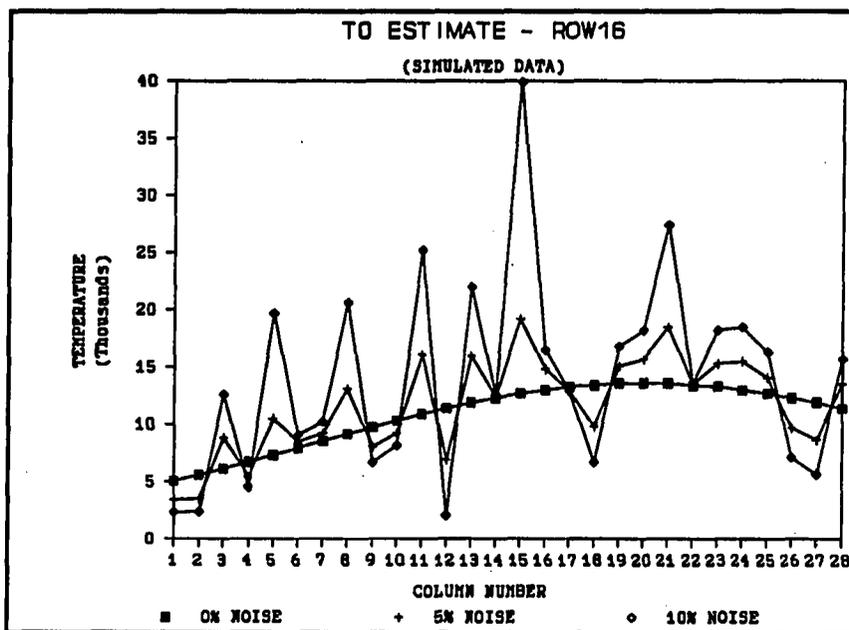


Figure 3.13: Cut through row 16 of T_0 maps with different amounts of noise

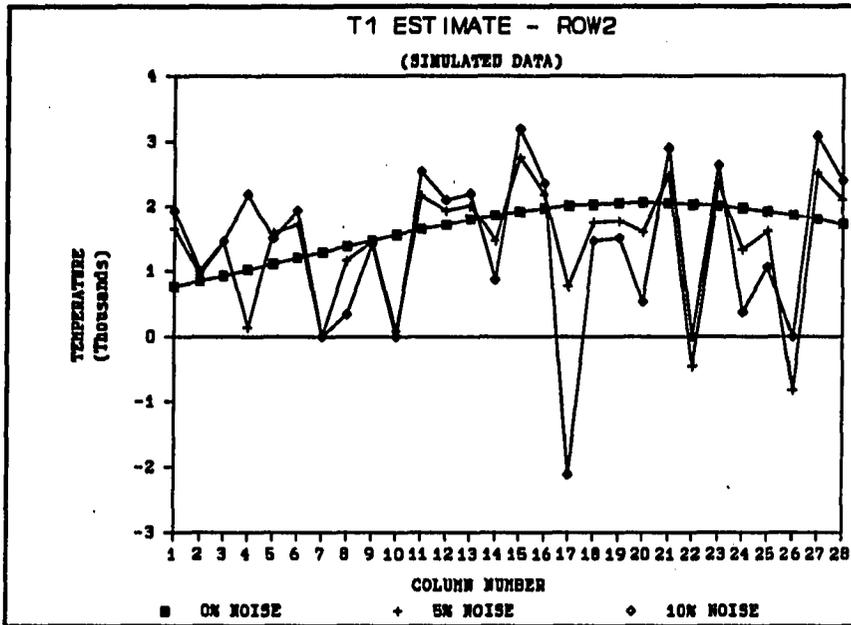


Figure 3.14: Cut through row 2 of T_1 maps with different amounts of noise

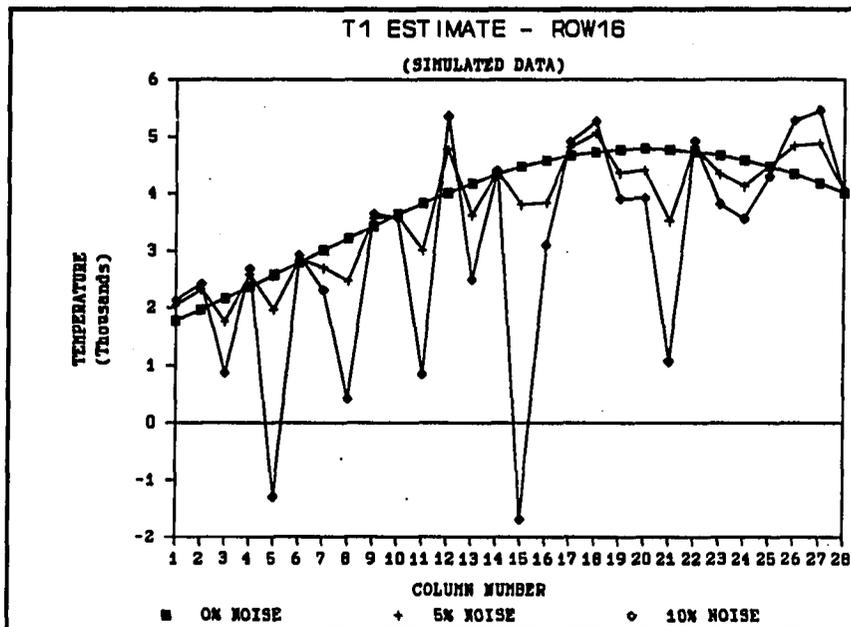


Figure 3.15: Cut through row 16 of T_1 maps with different amounts of noise

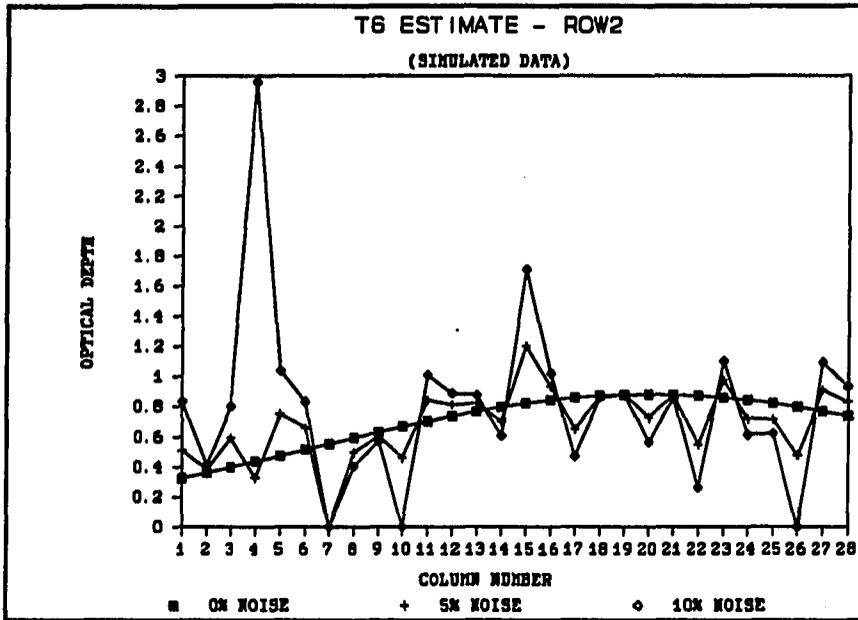


Figure 3.16: Cut through row 2 of τ_6 maps with different amounts of noise

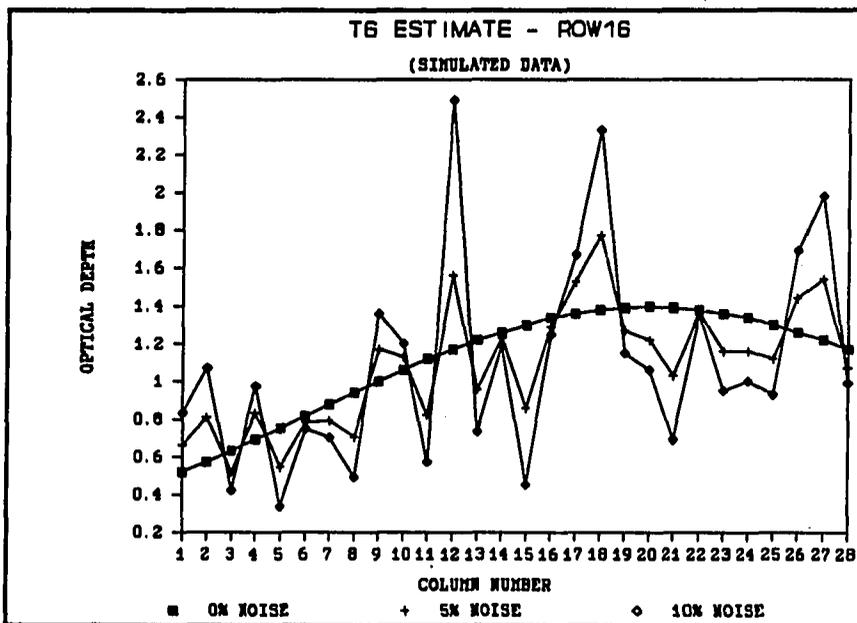


Figure 3.17: Cut through row 16 of τ_6 maps with different amounts of noise

These three figures also show that even a small amount of noise can result in estimated parameters that differ significantly from their true values as displayed by the curves labelled "5% noise". These curves show the estimated parameters in the situation when 5% zero mean white Gaussian noise, as discussed before, was added to the simulated flux density maps. The curves labelled "10% noise" show the results obtained when 10% zero mean white Gaussian noise, as discussed before, was added to the original flux density maps. These curves deviate further from the reference curve compared to the 5% noise curves, as expected, because the flux density values were more noisy in this case.

The true amount of inaccuracy in the *observed* flux densities is unknown, but it can be as high as 30%. With such large errors, it is obvious from the simulation study that something must be done to reduce these inaccuracies or the estimated parameter values, using the observed flux density values, will be almost worthless.

3.3 Double Smoothing Technique

In an attempt to improve the estimated parameters, a 3×3 smoothing running window was used to smooth the estimated parameter maps. The weight assigned to each cell in the 3×3 running window was exactly the same, $1/9$. The boundary cells were handled by assigning 0 values to all of them. Cuts were again made through the estimated parameter maps at the same rows as before. Only the results for the 10% noise case are shown in Figures 3.18, 3.19, 3.20, 3.21, 3.22 and 3.23. It can be seen that the smoothed estimated parameters, labelled by the curves "10% Noise Smoothed", approach the original values much more closely.

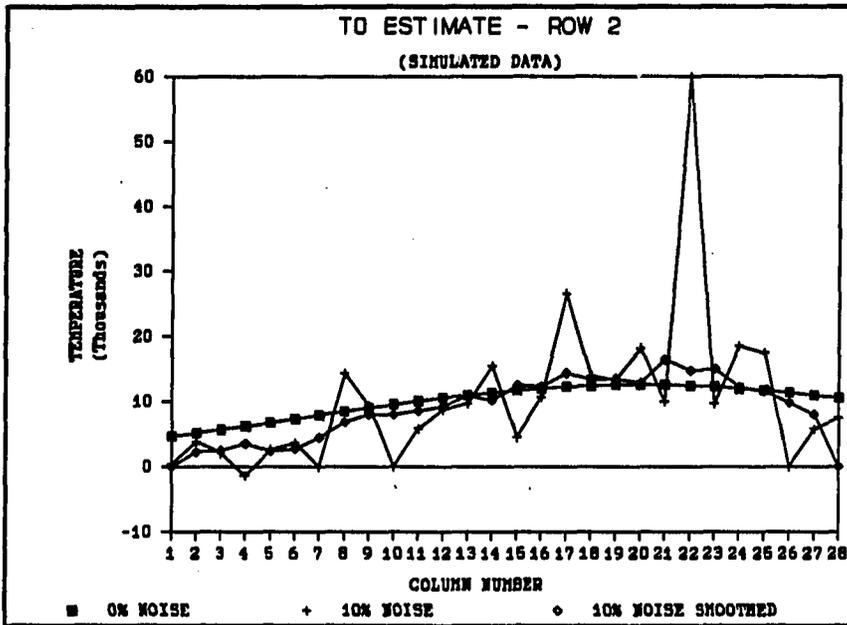


Figure 3.18: Cut through row 2 of T_0 maps with smoothed estimated parameters

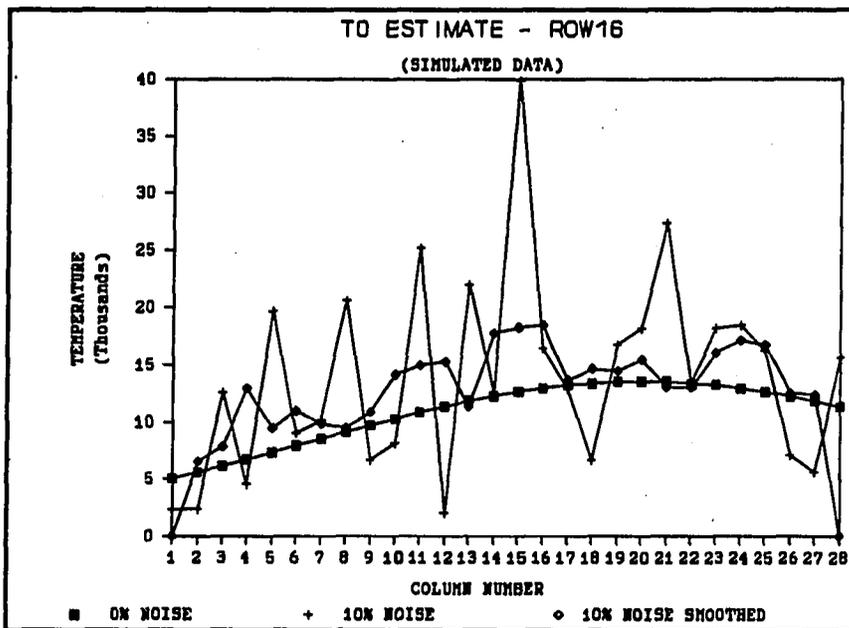


Figure 3.19: Cut through row 16 of T_0 maps with smoothed estimated parameters

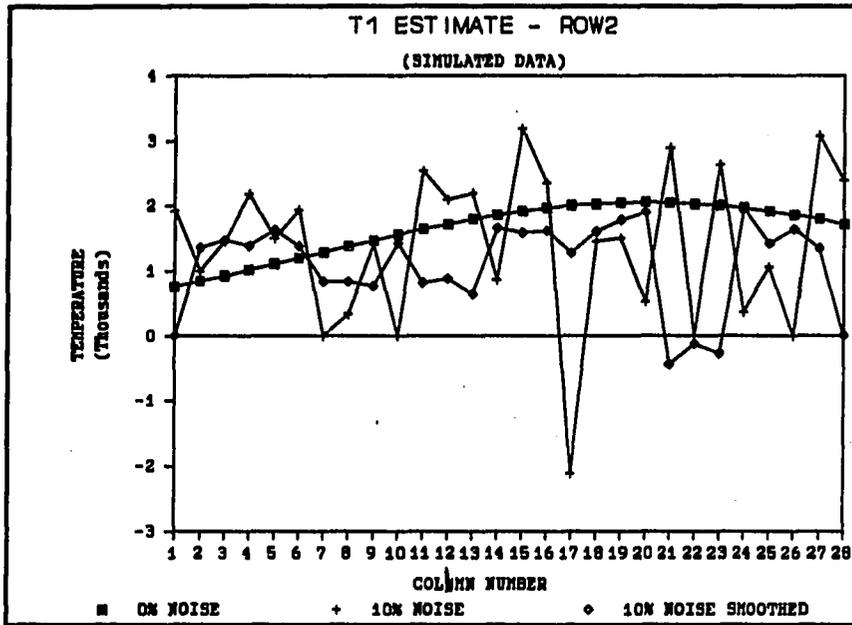


Figure 3.20: Cut through row 2 of T_1 maps with smoothed estimated parameters

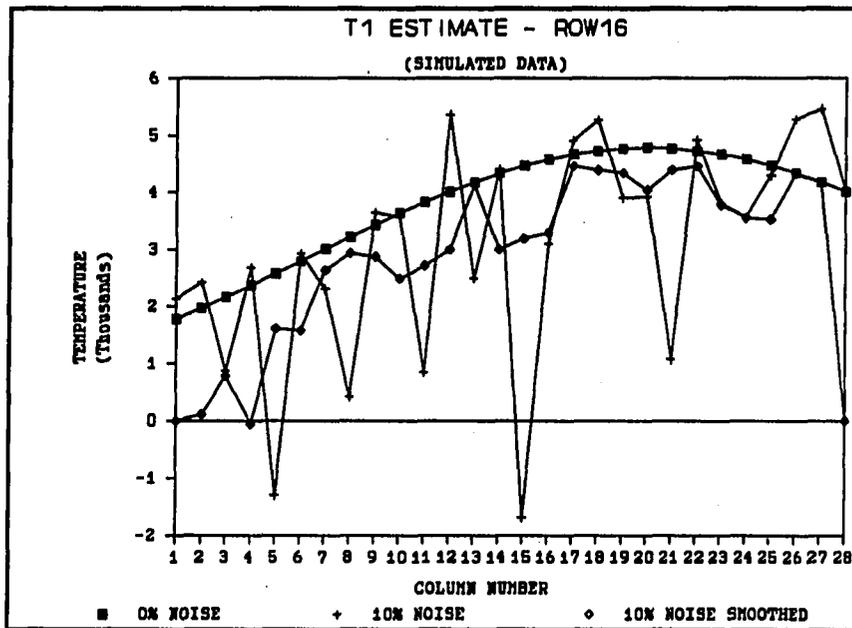


Figure 3.21: Cut through row 16 of T_1 maps with smoothed estimated parameters

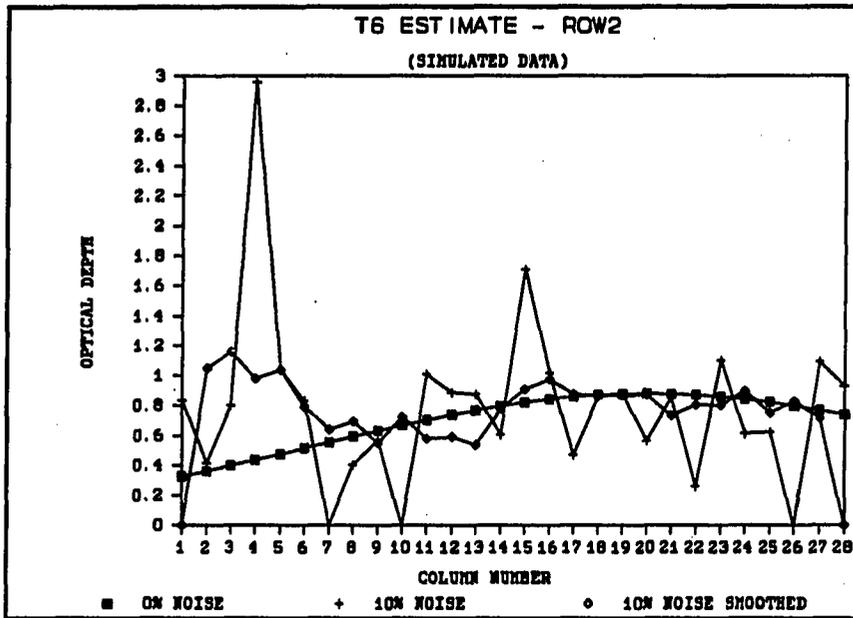


Figure 3.22: Cut through row 2 of τ_0 maps with smoothed estimated parameters

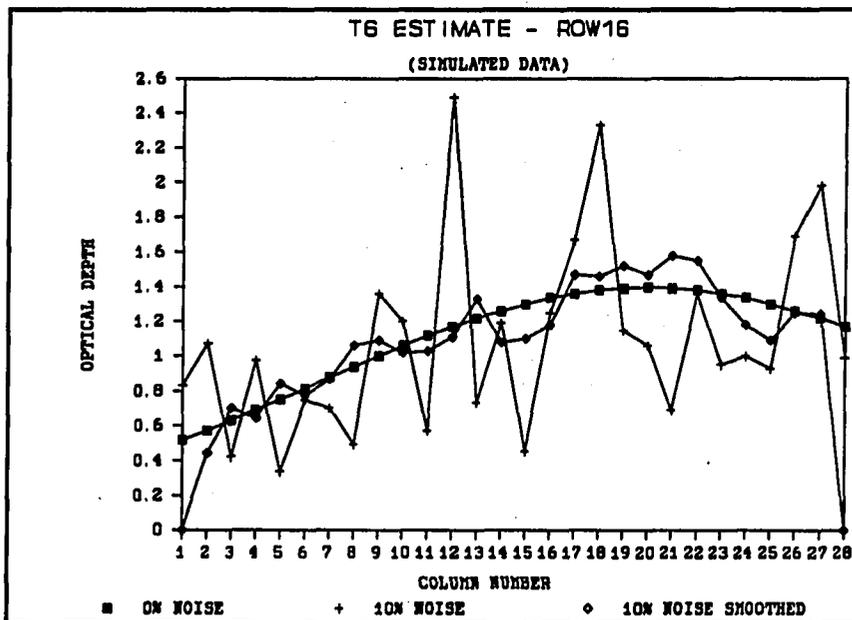


Figure 3.23: Cut through row 16 of τ_0 maps with smoothed estimated parameters

The other two curves have the same definitions as before. It is difficult to assign an error measure for each parameter in this case, because the nominal value of each parameter varies from pixel-to-pixel. Error measures as a function of different nominal values for each parameter will be calculated in Chapter 4 using a Monte Carlo analysis.

In an attempt to find the best possible estimated parameters, using the noisy flux density values, three different methods were tried (in each of the following situations, the same weight was assigned to each pixel in the 3×3 smoothing running window, namely $1/9$; boundary cells were assigned 0 values):

1. The first approach was to use a 3×3 smoothing running window to smooth the estimated parameter maps as already discussed. In this situation it was assumed that the estimated parameters were laying on a smooth surface and neighboring pixel values were of the same order. The effect of the smoothing window would be to reduce any noise effects in the estimated parameters.
2. The second approach was to use a 3×3 smoothing running window to smooth the "noisy" flux density maps, after the 10% zero mean white Gaussian noise was added. These smoothed "noisy" flux density maps were then used to estimate the unknown parameters. The same reasoning was followed in this case. It was assumed that the flux density values at neighbor pixels were of the same order and that the smoothing window would reduce any noise effects in the flux density values. This is a valid assumption to make in the case of the *observed* flux density maps as can be seen from the individual observed flux density values at all pixels given in the appendix as well as the observed contour maps shown in Figures 2.2, 2.3 and 2.4.

3. The third approach was to combine approaches 1 and 2. The flux density maps were smoothed before estimating the unknown parameters and this was followed by a smoothing of the estimated parameter maps themselves.

Although smoothing reduces the angular resolution of the maps, it is more important to reduce errors in the estimation than to retain the original resolution.

Cuts were again made through all the estimated parameter maps at the same rows as before, after applying the different approaches, and the results are shown in Figures 3.24, 3.25, 3.26, 3.27, 3.28 and 3.29. The curves labelled "0% noise" are again the reference curves, the curves labelled "Smoothed Temp" show the estimated parameters when the 3×3 running window was used to smooth the parameters estimated from the noisy flux density maps. The curves labelled "Smoothed Flux" show the parameters estimated when only the noisy flux density values were smoothed. The results of smoothing both the noisy flux density values and the estimated parameters are shown by the curves labelled "Smoothed Flux & Temp".

The calculation of the amount of noise in the estimated parameters is referred to the next chapter, where it will be done using a Monte Carlo analysis. However, it is clear from Figures 3.24 to 3.29 that smoothing of the estimated parameter maps improve the results in the sense that the estimated parameters are closer to their original values. Smoothing of the "noisy" flux density maps results in estimated parameters that are even better while the most accurate results were obtained by smoothing both the "noisy" flux density maps as well as the estimated parameter maps. This last technique will be referred to as the double smoothing technique from here on.

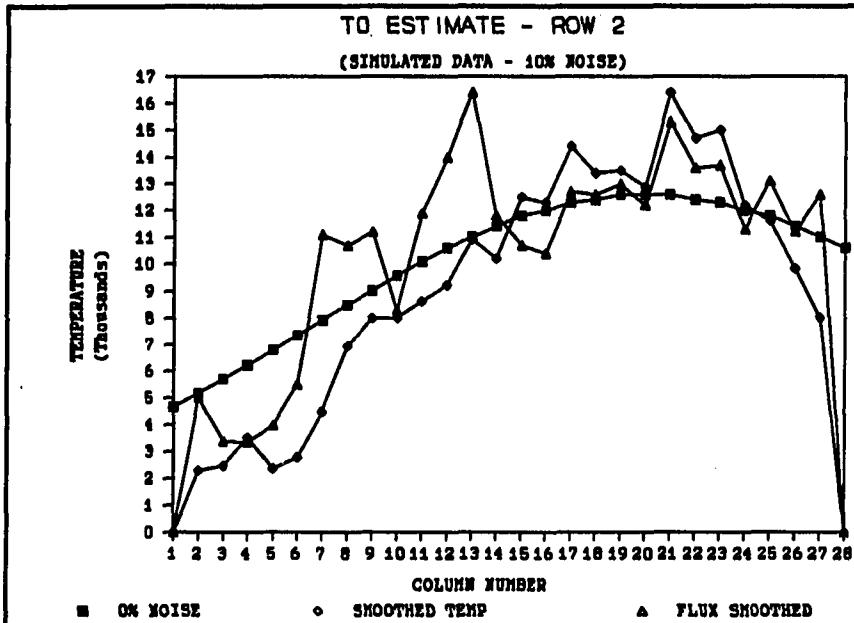


Figure 3.24: Cut through row 2 of T_0 maps with smoothed noise and smoothed estimated parameters

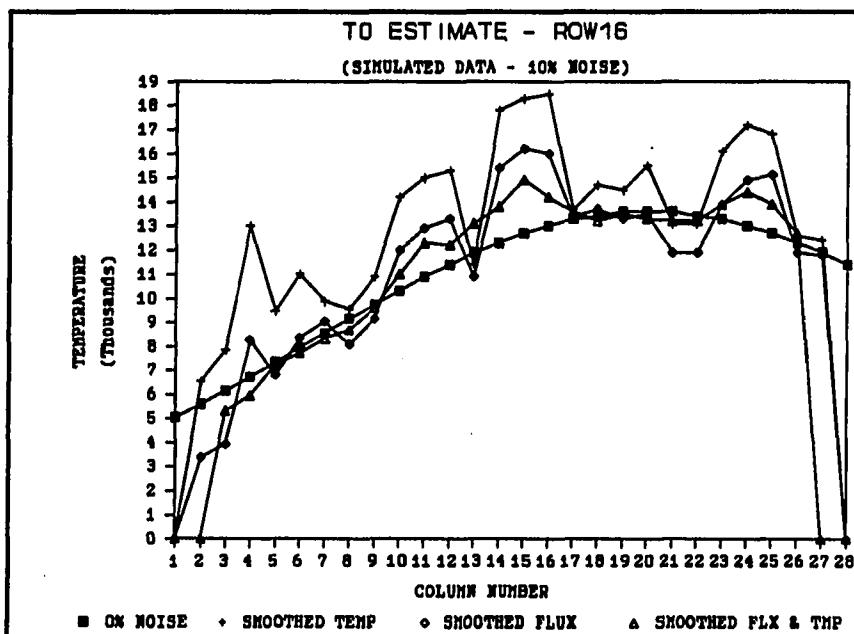


Figure 3.25: Cut through row 16 of T_0 maps with smoothed noise and smoothed estimated parameters

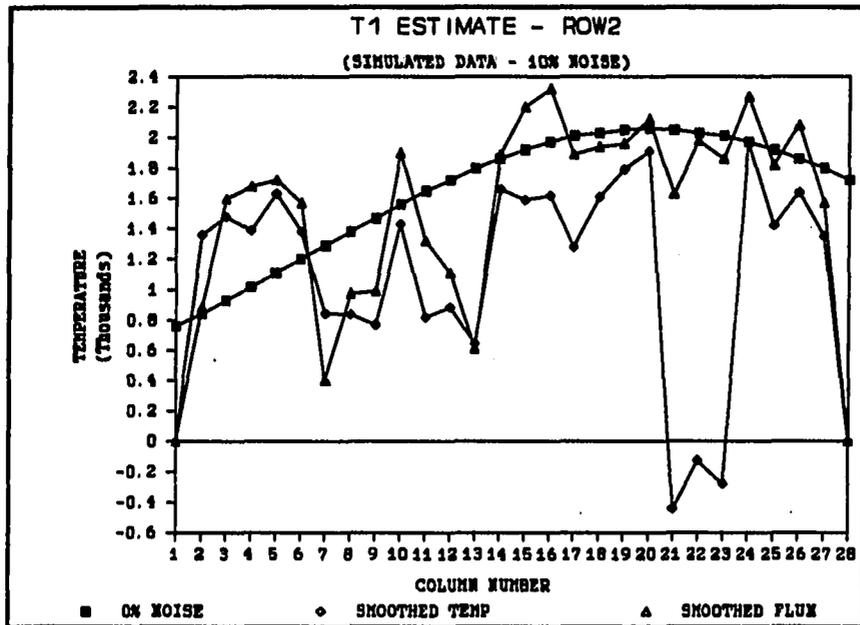


Figure 3.26: Cut through row 2 of T_1 maps with smoothed noise and smoothed estimated parameters

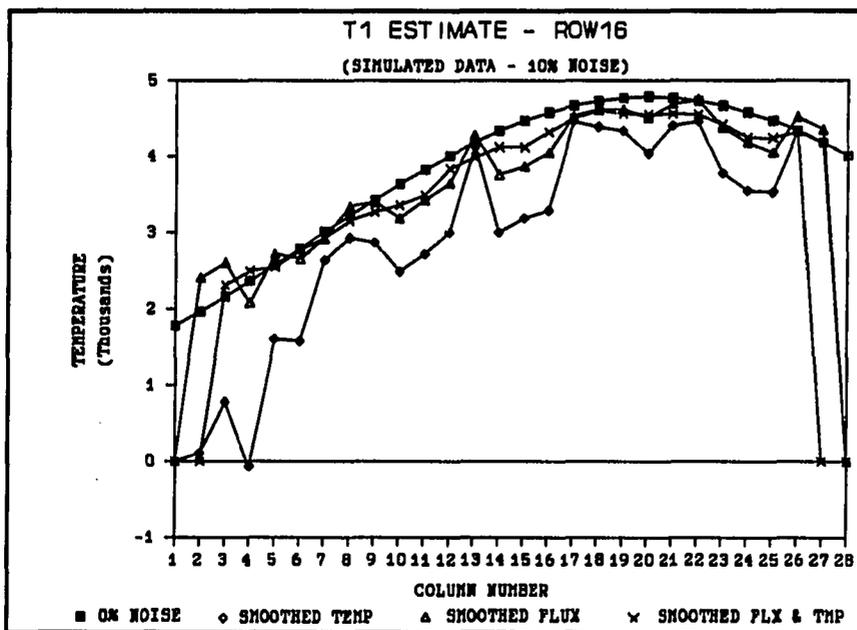


Figure 3.27: Cut through row 16 of T_1 maps with smoothed noise and smoothed estimated parameters

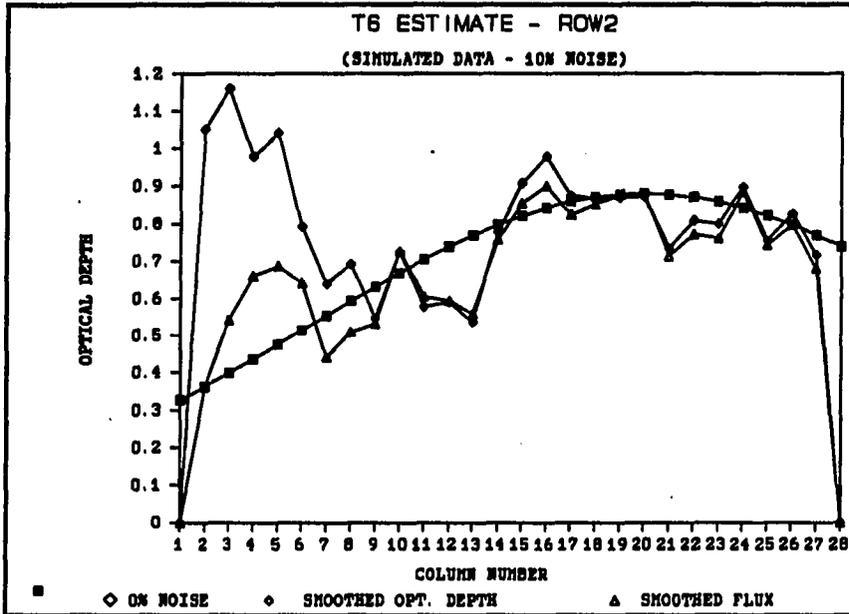


Figure 3.28: Cut through row 2 of τ_0 maps with smoothed noise and smoothed estimated parameters

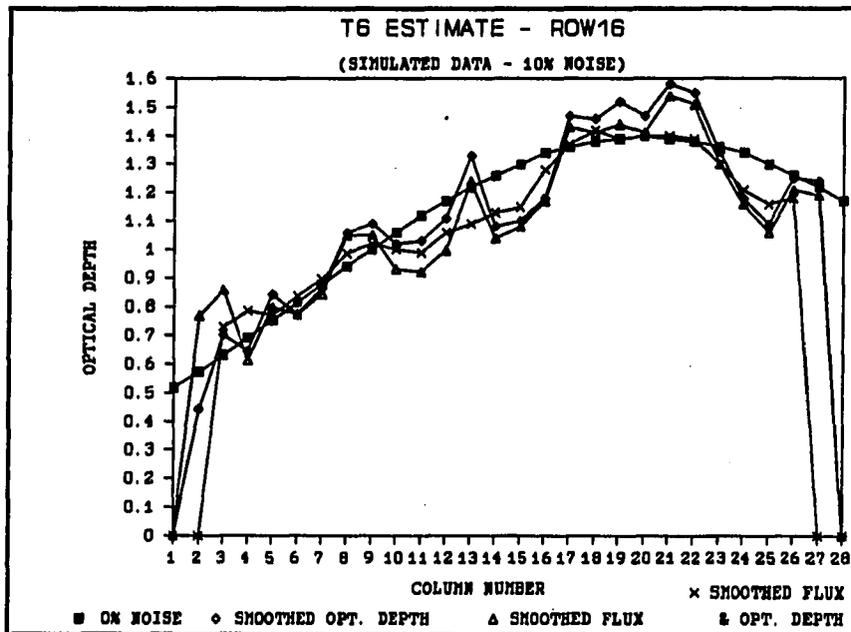


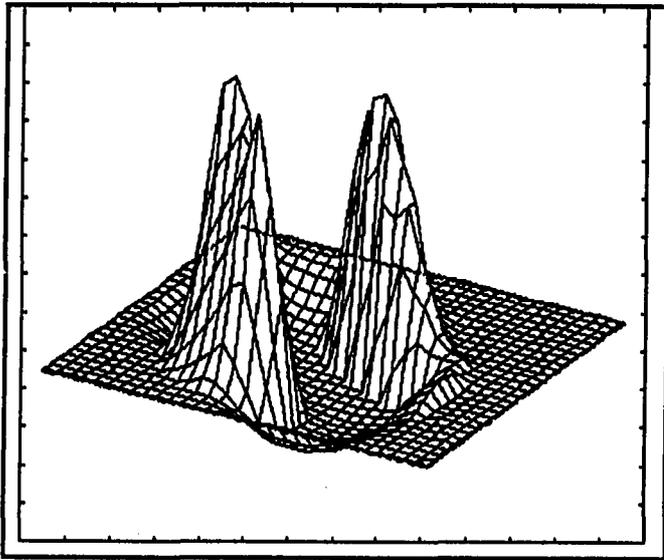
Figure 3.29: Cut through row 16 of τ_0 maps with smoothed noise and smoothed estimated parameters

3.4 Application of the Double Smoothing Technique to the Observed Flux Density Values

In the development of the double smoothing technique to estimate parameters in the noisy simulated flux density maps, it was assumed that both the flux density values and the estimated parameters lay on a smooth 2-D surface. When interpreting the results of the double smoothing technique as applied to the observed flux density values, these assumptions must be kept in mind. As a general rule, it can be expected that the edges in the estimated parameter maps are smoother than they are in reality. The double smoothing technique was then applied to the actual observed flux density maps. Three dimensional and contour plots of the estimated parameters are shown in Figures 3.30, 3.31, 3.32, 3.33, 3.34 and 3.35.

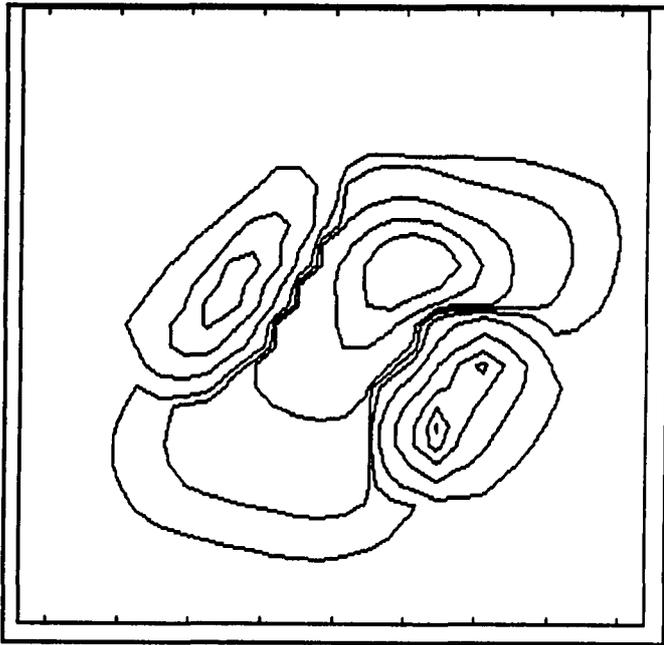
3.4.1 Optical Depth τ_6 -map

Comparing the estimated map for the optical depth at 6 cm, Figures 3.34 and 3.35, with the map for the same parameter obtained by Basart and Daub (1987), Figure 3.2, it can be seen that both have the same basic double peak structure with reduced optical depth in the center. However, the maximum optical depth found by the present investigation is 2.39 compared with a value of 1.0 obtained by Basart and Daub (1987). The estimated parameters in the present situation are very sensitive to errors in the flux density values, as already established. Therefore, differences in the obtained numerical values can be expected, but the fact that the basic structures in the two maps for τ_6 are the same, suggests that the temperature model used, seems to apply to the nebula under investigation.



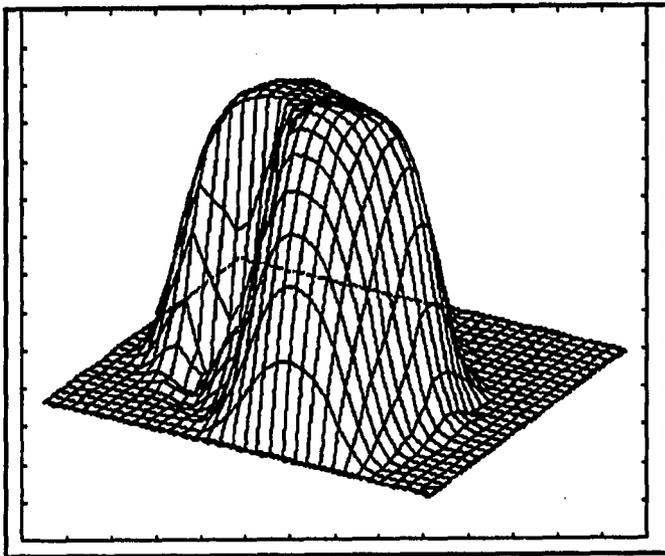
Maximum value = 45300 K
Minimum value = -8870 K

Figure 3.30: T_0 estimated parameter 3-D map using observed data



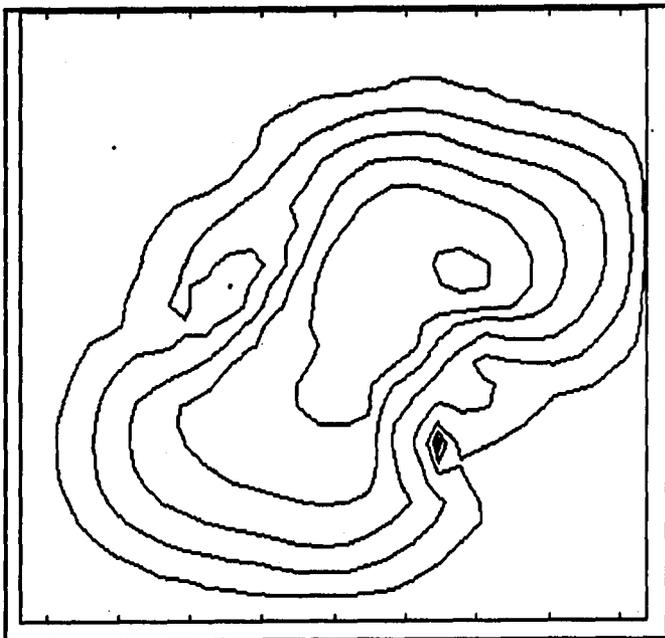
Peak value = 45300 K
Contour levels at 99%, 90%,
75%, 50%, 25%, 10%, 5%,
2%, 1%, 0.5%

Figure 3.31: T_0 estimated parameter contour map using observed data



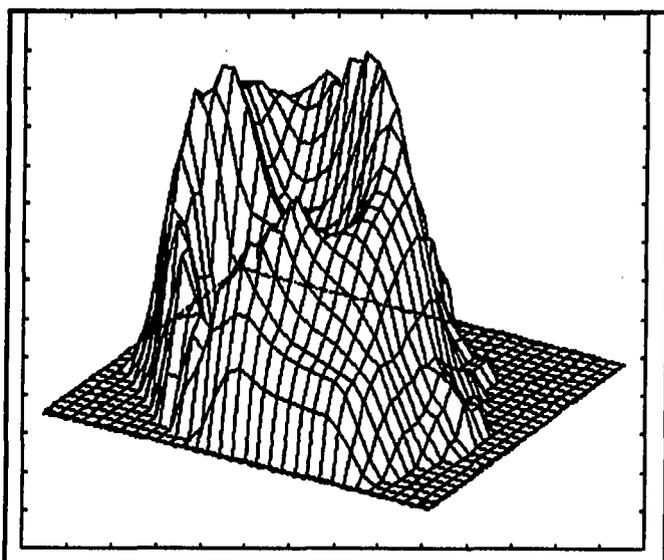
Maximum value = 15000 K
Ground level value = 0 K

Figure 3.32: T_1 estimated parameter 3-D map using observed data



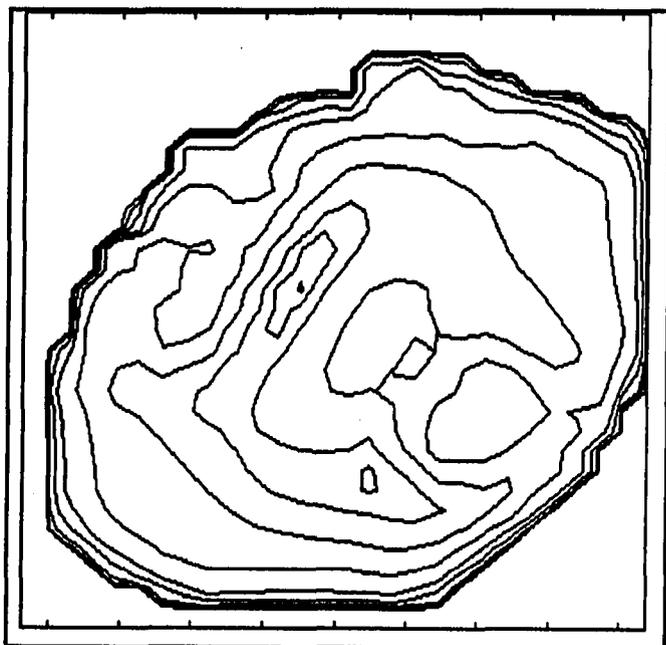
Peak value = 15000 K,
contour levels at 99%, 90%,
75%, 50%, 25%, 10%, 5%,
2%, 1%, 0.5%

Figure 3.33: T_1 estimated parameter contour map using observed data



Maximum value = 2.39
Ground level value = 0.0

Figure 3.34: τ_6 estimated parameter 3-D map using observed data



Peak value = 2.39
contour levels at 99%, 90%,
75%, 50%, 25%, 10%, 5%,
2%, 1%, 0.5%

Figure 3.35: τ_6 estimated parameter contour map using observed data

3.4.2 Temperature Maps

Basart and Daub (1987) assumed a constant temperature along the line of sight and therefore derived only one temperature map, shown in Figure 3.1. A comparison of this temperature map with the temperature map obtained in the present study for the temperature at the front of the nebula, T_1 , Figures 3.32 and 3.33, shows a correspondence that is remarkable. Not only do both have the same structure, the maximum temperature obtained by Basart and Daub (1987) was 15 000 K, exactly the same as the maximum temperature obtained by the present investigation, 15 000 K. The structures of the two temperature maps obtained using the different approaches are also comparable. Both have a high temperature spot in the north east corner of the maps with a relative flat temperature profile for the rest of the surface. The temperature at the back of the nebula, T_0 , Figures 3.30 and 3.31, can obviously not be compared with the previous results. However, the symmetrical structure is encouraging because it has the same form as the flux density maps obtained for the nebula at low values of optical depth. The regions of negative temperature, in the center of the map, are not physical possible. The most logical explanation for this characteristic would be to blame it on the uncertainties in flux density values and assume zero temperatures for this region.

3.5 Effect of a Different Smoothing Window

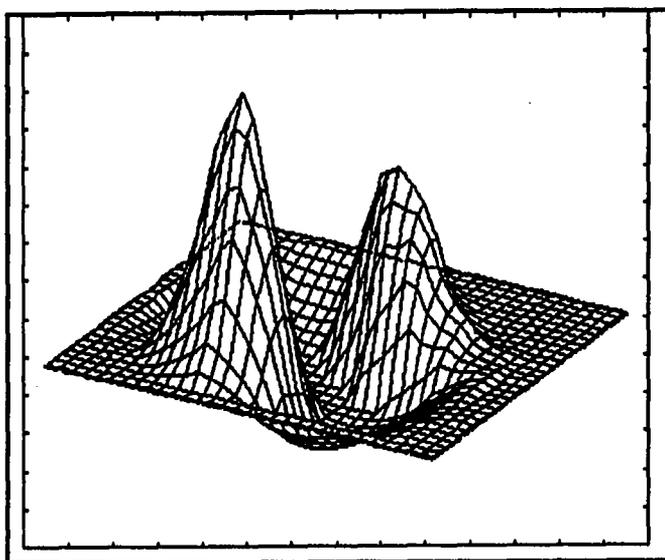
The same parameters were also estimated using a different window size, 5×5 , in the double smoothing approach. In this case the weight assigned to each cell in the running window was $1/25$.

The resulting temperature and optical depth maps are shown in Figures 3.36, 3.37, 3.38, 3.39, 3.40 and 3.41. These maps have basically similar structures as the maps obtained using the 3×3 running window, shown in Figures 3.30 to 3.35. However, there are some small differences between the two sets of maps. The most important difference is that the τ_6 -map obtained using the 5×5 running window, shown in Figures 3.40 and 3.41, is less symmetric than the τ_6 -map obtained using the 3×3 running window shown in Figures 3.34 and 3.35, and the region of lower optical depth in the center of the τ_6 -map is now absent. Another difference is that the two peaks in the T_0 temperature map in Figures 3.30 and 3.31, now have different maximum values, as shown in Figures 3.36 and 3.37.

The numerical values obtained using the two different methods, however, differ by large margins. These differences are the largest for the maximum temperature value at the back of the nebula which is now 33 600 K instead of 45 300 K and the optical depth which now has a maximum value of 1.91 instead of 2.39 before.

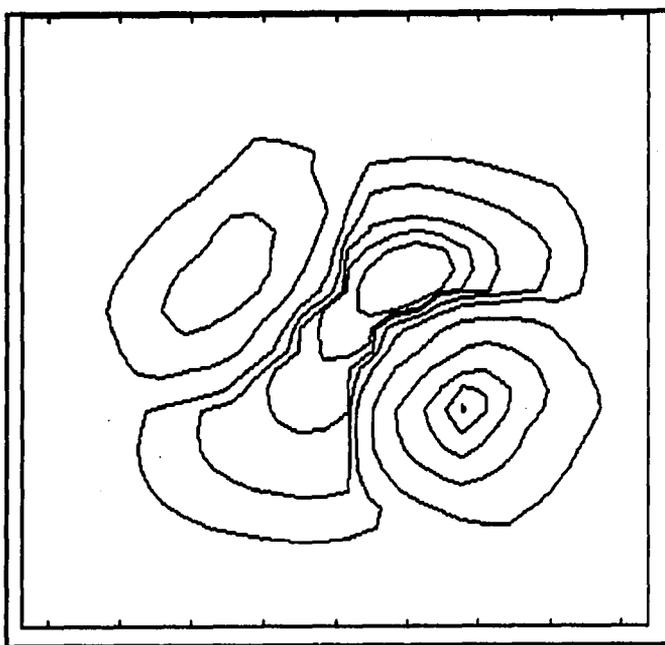
All these differences can be explained by the fact that more smoothing was applied to the original observed flux density values using the 5×5 smoothing window than is the case using the 3×3 smoothing window, keeping in mind the sensitivity of the estimated parameters on the values of the flux densities used in the estimation procedure.

There is obviously no hard and fast rule in choosing one of the two sets of maps as more correct than the other. However, by using the 5×5 smoothing window, more detailed information about the fine structure in the nebula is lost than using the 3×3 window.



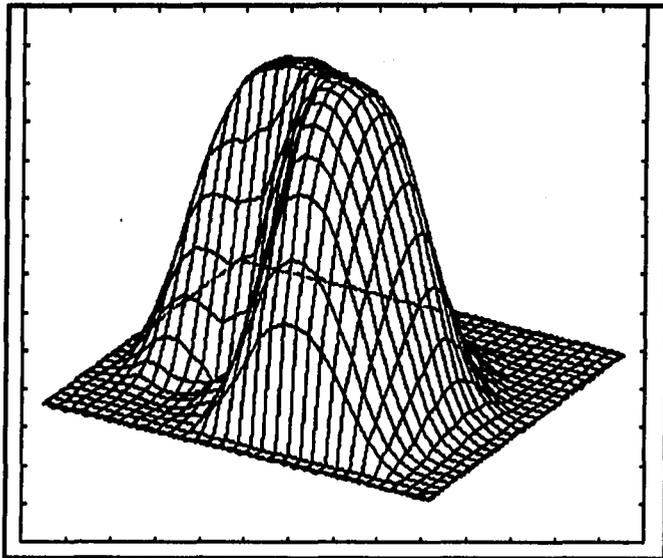
Maximum value = 33600 K
Minimum value = -8360 K

Figure 3.36: T_0 estimated parameter 3-D map using observed data, 5x5 smoothing window used



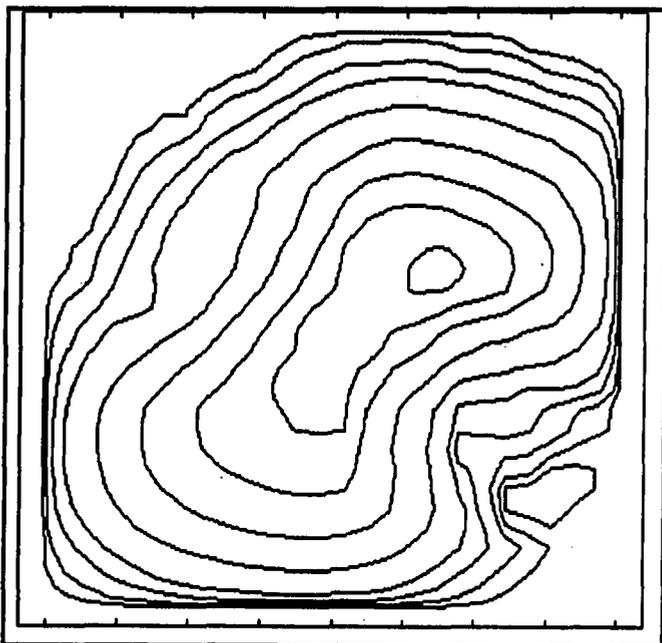
Peak value = 33600 K
contour levels at 99%, 90%,
75%, 50%, 25%, 10%, 5%,
2%, 1%, 0.5%

Figure 3.37: T_0 estimated parameter contour map using observed data, 5x5 smoothing window used



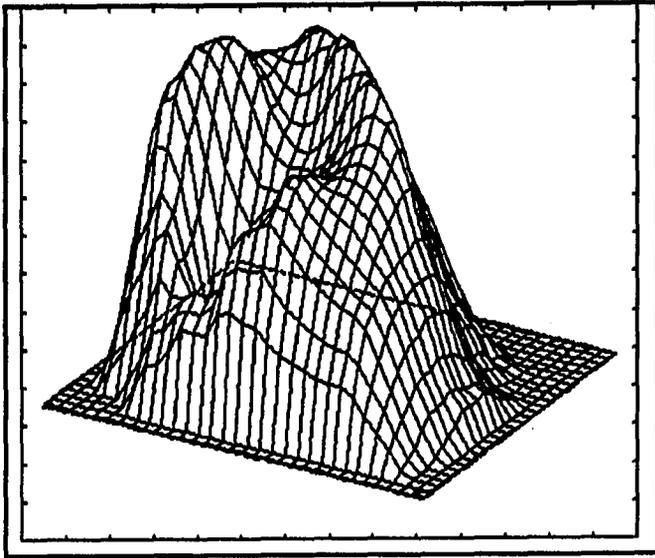
Maximum value = 14400 K
Ground level value = 0 K

Figure 3.38: T_1 estimated parameter 3-D map using observed data, 5x5 smoothing window used



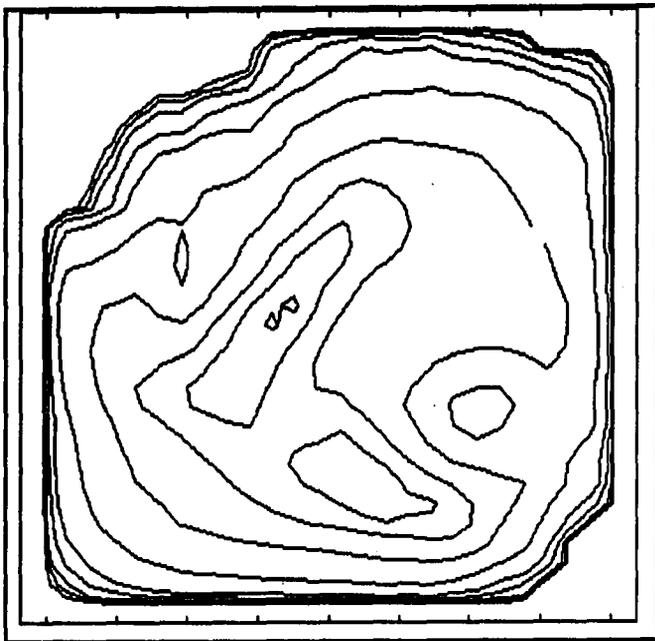
Peak value = 14400 K
Contour levels at 99%, 90%,
75%, 50%, 25%, 10%, 5%,
2%, 1%, 0.5%

Figure 3.39: T_1 estimated parameter contour map using observed data, 5x5 smoothing window used



Maximum value = 1.91
Ground level value = 0.0

Figure 3.40: τ_6 estimated parameter 3-D map using observed data, 5x5 smoothing window used



Peak value = 1.91
Contour levels at 99%, 90%,
75%, 50%, 25%, 10%, 5%,
2%, 1%, 0.5%

Figure 3.41: τ_6 estimated parameter contour map using observed data, 5x5 smoothing window used

Also, the assumption that the actual flux density values observed from the nebula stays constant over a 5×5 window, is too extreme. Therefore, the results of the double smoothing technique, using the 3×3 smoothing running window will be used in future references, because no physical insight was gained by using the larger smoothing window.

4. STATISTICAL ANALYSIS

4.1 Introduction

A summary of the double smoothing technique, as discussed in Chapter 3, to estimate the two temperature parameters, T_0 and T_1 , and the optical depth parameter, τ_6 , from the observed flux density maps, is as follows (the 3×3 smoothing running window used, from here on, assigns the same weight, $1/9$, to each cell):

1. Smooth each of the three original observed flux density maps using the 3×3 running window.
2. Use linear estimation theory to estimate the three unknown parameters in the three highly nonlinear equations describing the flux densities at the three observed wavelengths as discussed in Section 2.3.1. These three nonlinear equations are given by equations (2.15), (2.16) and (2.17). The smoothed flux density values are used in the estimation of the unknown parameters and the estimation is done on a pixel-to-pixel basis.
3. Smooth the estimated parameters T_0 , T_1 and τ_6 using the 3×3 running window.

Two statistical aspects of the above procedure deserves further attention:

1. The determination of the variance of the estimated parameters from a knowledge of the variance of the observed flux densities.
2. How much the smoothing of the observed flux density values and the smoothing of the estimated parameters reduce the variance of the estimated parameters as compared to the case when no smoothing is used.

4.2 Variance of Estimated Parameters

The flux density equation, using temperature model 2, is given by

$$S = K_1 \left\{ T_0 (1 - e^{-a\tau_6}) - \frac{(T_0 - T_1)}{a\tau_6} (a\tau_6 - 1 + e^{-a\tau_6}) \right\} \quad (4.1)$$

where

$a = 0.09528$ at 2 cm,

$a = 1.0000$ at 6 cm, and

$a = 12.544$ at 20 cm.

Theoretically there are four methods to determine the probability distribution of a system given the probability distributions of the system components.

1. The Transformation of Variables.
2. Application of the Central Limit Theorem for Linear Systems.
3. Generation of System Moments.
4. Monte Carlo Simulation.

4.2.1 The Transformation of Variables

4.2.1.1 Method Description Only the case of three equations in three unknowns (as applicable to the problem under consideration) is discussed.

Let the three equations be given by

$$y_1 = u_1(x_1, x_2, x_3) \quad (4.2)$$

$$y_2 = u_2(x_1, x_2, x_3) \quad (4.3)$$

$$y_3 = u_3(x_1, x_2, x_3) \quad (4.4)$$

and let $f(x_1, x_2, x_3)$ be the value of the joint probability density of the continuous random variables x_1 , x_2 and x_3 at (x_1, x_2, x_3) . Under the constraint that the functions given by y_1 , y_2 and y_3 are partially differentiable with respect to x_1 , x_2 and x_3 and the constraint that there exists a one-to-one transformation for all values within the range of x_1 , x_2 and x_3 for which $f(x_1, x_2, x_3) \neq 0$, it is possible for these values of x_1 , x_2 and x_3 to solve for x_1 , x_2 and x_3 uniquely from the equations for y_1 , y_2 and y_3 to get

$$x_1 = w_1(y_1, y_2, y_3) \quad (4.5)$$

$$x_2 = w_2(y_1, y_2, y_3) \quad (4.6)$$

$$x_3 = w_3(y_1, y_2, y_3) \quad (4.7)$$

The joint probability density for $y_1 = u_1(x_1, x_2, x_3)$, $y_2 = u_2(x_1, x_2, x_3)$ and $y_3 = u_3(x_1, x_2, x_3)$ is then given by (Freund and Walpole, 1987, p. 257)

$$g(y_1, y_2, y_3) = f[w_1(y_1, y_2, y_3), w_2(y_1, y_2, y_3), w_3(y_1, y_2, y_3)]|J| \quad (4.8)$$

where J is called the Jacobian of the transformation and is given by

$$J = \begin{vmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} & \frac{\partial x_1}{\partial y_3} \\ \frac{\partial x_2}{\partial y_1} & \frac{\partial x_2}{\partial y_2} & \frac{\partial x_2}{\partial y_3} \\ \frac{\partial x_3}{\partial y_1} & \frac{\partial x_3}{\partial y_2} & \frac{\partial x_3}{\partial y_3} \end{vmatrix} \quad (4.9)$$

4.2.1.2 Application of Method In the problem under consideration it is assumed that the probability distributions of the observed flux densities (y_1 , y_2 and y_3) are known and it required to obtain the probability distributions of the unknown parameters (x_1 , x_2 and x_3).

To be able to apply this method it is necessary to solve for x_1 , x_2 and x_3 (T_0 , T_1 and τ_6). This requires that x_1 , x_2 and x_3 be uniquely solved for in terms of y_1 , y_2 and y_3 (S_{02} , S_{06} and S_{20}). Due to the nonlinearity of the equations involved this is not possible, especially for τ_6 , and therefore this method is not applicable to the problem under consideration.

4.2.2 Application of the Central Limit Theorem for Linear Systems

4.2.2.1 Method Description Given that x_1, x_2, \dots, x_n are independent observations from distributions with finite mean and variance, according to the central limit theorem, the average of these observations approaches a normal distribution as the number of observations increase. This is also true for the sum of independent variables.

Let x_1, x_2, \dots, x_n be random variables. The expected value of the sum of these random variables is given by

$$E(x_1 + x_2 + x_3 + \dots + x_n) = E(x_1) + E(x_2) + \dots + E(x_n) = \mu. \quad (4.10)$$

If the variables are also uncorrelated, their variances are given by

$$Var(x_1 + x_2 + \dots + x_n) = Var(x_1) + Var(x_2) + \dots + Var(x_n) = \sigma^2. \quad (4.11)$$

Knowing the mean μ and the variance σ^2 , the normal distribution is completely known.

4.2.2.2 Application of Method This approach is also not applicable to the case under consideration because none of the unknown parameters is a linear sum of the observed flux densities. Even if the unknown parameters were a linear sum of the observed flux densities, it would not be correct to assume that the central limit theorem applied to the case when only three random variables are used.

4.2.3 Generation of System Moments

4.2.3.1 Method Description The following notation will be used in this section (Hahn and Shapiro, 1967):

$E(x_i)$ = mean of i the component variable,

$u_k(x_i)$ = k the central moment of i the component variable,

$E(y)$ = mean of system output, and

$u_k(y)$ = k the central moment of system output.

It will also be assumed that the equation describing the system behavior is given by

$$y = h(x_1, x_2, \dots, x_n) \quad (4.12)$$

where x_1, x_2, \dots, x_n are the components contributing to the system output and are called component variables. The objective of this method is to find equations describing the mean of the system output ($E(y)$) and the second, third and fourth central moments of the system output ($u_k(y)$ for $k = 2, 3, 4$) given:

1. The system transfer function $h(x_1, x_2, \dots, x_n)$.
2. The system component data. These data must then be used to calculate the mean ($E(x_i)$) and central moments ($u_k(x_i)$) of each of the component variables.

Hahn and Shapiro (1967, p. 229) had shown that, assuming that the component variables are uncorrelated, the expression for the mean of the system output, retaining terms up to second order, is given by

$$E(y) = h[E(x_1), E(x_2), \dots, E(x_n)] + \frac{1}{2} \sum_{i=1}^n \frac{\partial^2 \bar{h}}{\partial x_i^2} Var(x_i) \quad (4.13)$$

where $\partial^2 \bar{h} / \partial x_i^2 = \partial^2 h / \partial x_i^2$ evaluated at the mean value $E(x_i)$, and

$$Var(y) = \sum_{i=1}^n \left(\frac{\partial h}{\partial x_i} \right)^2 Var(x_i) + \sum_{i=1}^n \left(\frac{\partial h}{\partial x_i} \right) \left(\frac{\partial^2 h}{\partial x_i^2} \right) u_3(x_i) \simeq \sum_{i=1}^n \left(\frac{\partial h}{\partial x_i} \right)^2 Var(x_i). \quad (4.14)$$

4.2.3.2 Application of Method To be able to apply this method to the problem under consideration, the unknown parameters, T_0 , T_1 and τ_6 , must each be solved for in terms of the observed flux densities at 2 cm, 6 cm and 20 cm. As already seen, this is not possible. Therefore this method is not applicable to the problem under consideration.

4.2.4 Monte Carlo Simulation

4.2.4.1 Method Description This method requires only that the distributions of all the system component variables be known. If this is the case, the distribution of the system output can be calculated by computer simulations of many systems using values for the system components obtained from the component distributions. The steps in this procedure would be as follows:

1. Find and mathematically describe the statistical distribution for each component variable.
2. For each system component, select a random value from the corresponding component distributions.
3. Use these values of the system components to calculate a value of the system output.

Repeat steps 2 and 3 as many times as necessary.

4. Use the resulting values of the system output to get an approximation of the system distribution.

4.2.4.2 Confidence Level The Monte Carlo simulation is a statistical procedure. Therefore, any estimate of the system output calculated using this procedure, will have a certain confidence level. In general, by increasing the number of trials, the errors in the estimates can be reduced and the confidence level be increased.

The number of trials required to obtain a specified confidence level can be calculated using the following expression under certain restrictions¹

$$n = \frac{p'(1-p')}{E^2} z_{1-\alpha/2}^2 \quad (4.15)$$

where

n is the number of trials required,

E is the maximum allowable error in estimating p ,

p is the parameter of a binomial distribution,

p' is the initial estimate of p , and

$z_{1-\alpha/2} = 1 - \alpha/2$ percent point of standard normal distribution.

A problem in using this equation is that it requires an initial estimate of p , the quantity to be determined by the Monte Carlo simulation. This problem can be solved by using a worst case scenario and assume $p = 0.5$. This will result in the largest possible number of trials needed to obtain a specified confidence level.

To be able to estimate the variance of the estimated parameters with a 95% confidence level, the number of trials needed is given by

¹This equation is based on the normal approximation to the binomial distribution. The restrictions are that neither np or $n(1-p)$ is less than 5. [from Hahn and Shapiro, 1967].

$$n = \frac{(0.5)(0.5)}{(0.05)^2} (1.96)^2 = 384 \quad (4.16)$$

It is interesting to note that a 99% confidence level, under the worst case scenario, would require 16641 trials.

4.2.4.3 Application of Method This method is directly applicable to the problem under consideration. (It is assumed that the corrupting noise in the observed flux densities is white Gaussian noise.) To find the relationship between the noise in the observed flux densities and the resulting errors in the estimated parameters, use the following method:

1. Choose values for T_0 , T_1 and τ_6 in the expected ranges of these variables, as already discussed. These values will also be referred to as the nominal values.
2. Calculate the flux densities at the three wavelengths used, 2 cm, 6 cm and 20 cm using the nominal values of T_0 , T_1 and τ_6 and the given flux density equations.
3. Add zero mean white Gaussian noise with a specified variance to the simulated flux densities.

Repeat this 384 times to get 384 sets of noise corrupted flux densities for each set of nominal parameters chosen.

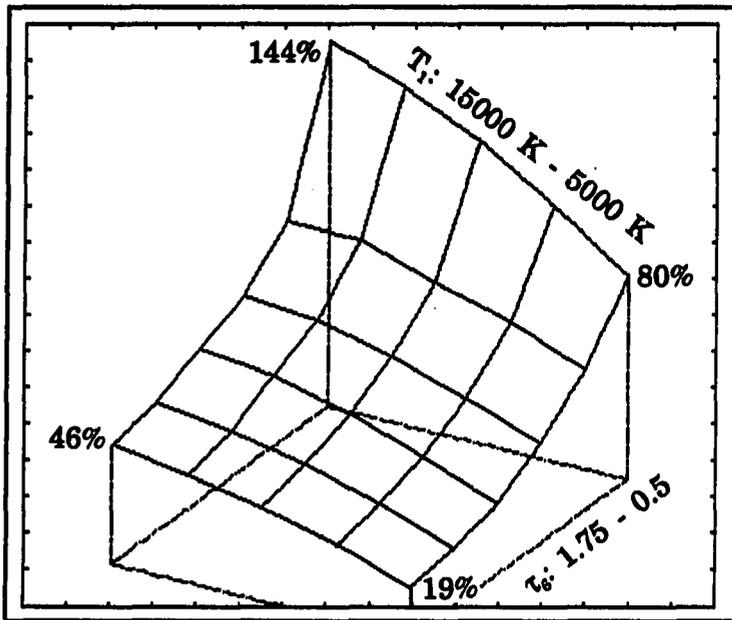
4. Solve 384 times for the values of T_0 , T_1 , and τ_6 from the 384 sets of noise corrupted flux densities.
5. Calculate the variance of T_0 , T_1 and τ_6 from all the solutions obtained in steps 3 and 4.

6. Repeat this procedure for different sets of nominal values.

4.2.4.4 Experimental Results The above procedure was followed to obtain the standard deviation of the estimated parameters for a nominal value of $T_0 = 10000$ K and varying nominal values of T_1 and τ_6 . It was assumed that zero mean white Gaussian noise with a standard deviation of 5% of the nominal value of the flux densities was added to the calculated flux densities in each case.

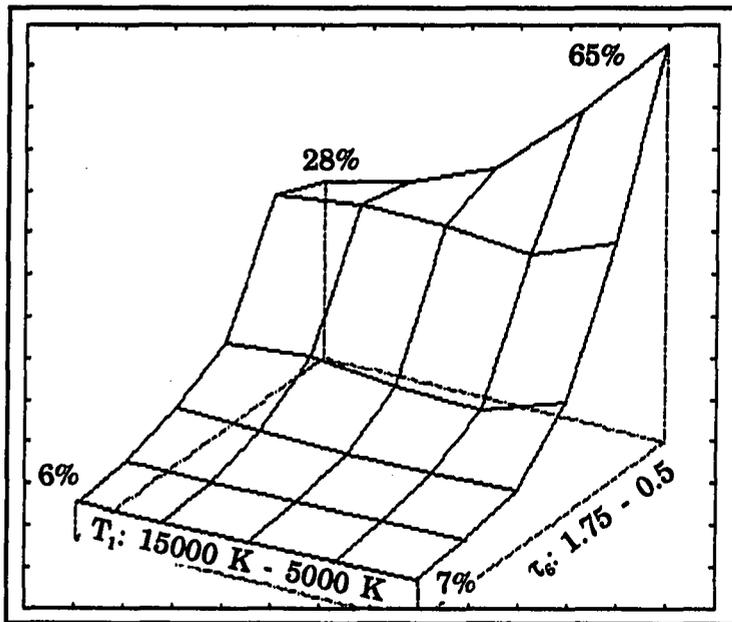
Figure 4.1 shows the standard deviation of parameter T_0 as a function of the nominal values of parameters T_1 and τ_6 , $T_0 = 10000$ K. These values are expressed as percentages of the nominal values used for the different parameters. It is quite clear that even a 5% standard deviation in the values of the observed flux densities can result in enormous errors in the value of estimated parameter T_0 . It can further be seen that the standard deviation of the estimated T_0 parameter increases as the nominal value of parameter τ_6 decrease and as the nominal value of parameter T_1 increases.

Figure 4.2 shows the standard deviation of parameter T_1 as a function of the nominal values of parameters T_1 and τ_6 , $T_0 = 10000$ K. These values are expressed as percentages of the nominal values. It is again clear that a 5% standard deviation in the values of the observed flux densities can result in large errors in the values of estimated parameter T_1 , although these errors are smaller than the errors for estimated parameter T_0 . The standard deviation of the estimated parameter T_1 is not very sensitive to the nominal value of parameter T_1 , but increases as the nominal value of parameter τ_6 decreases.



T_0 nominal = 10000 K

Figure 4.1: Standard deviation of parameter T_0 estimated directly from noisy simulated flux density values



T_0 nominal = 10000 K

Figure 4.2: Standard deviation of parameter T_1 estimated directly from noisy simulated flux density values

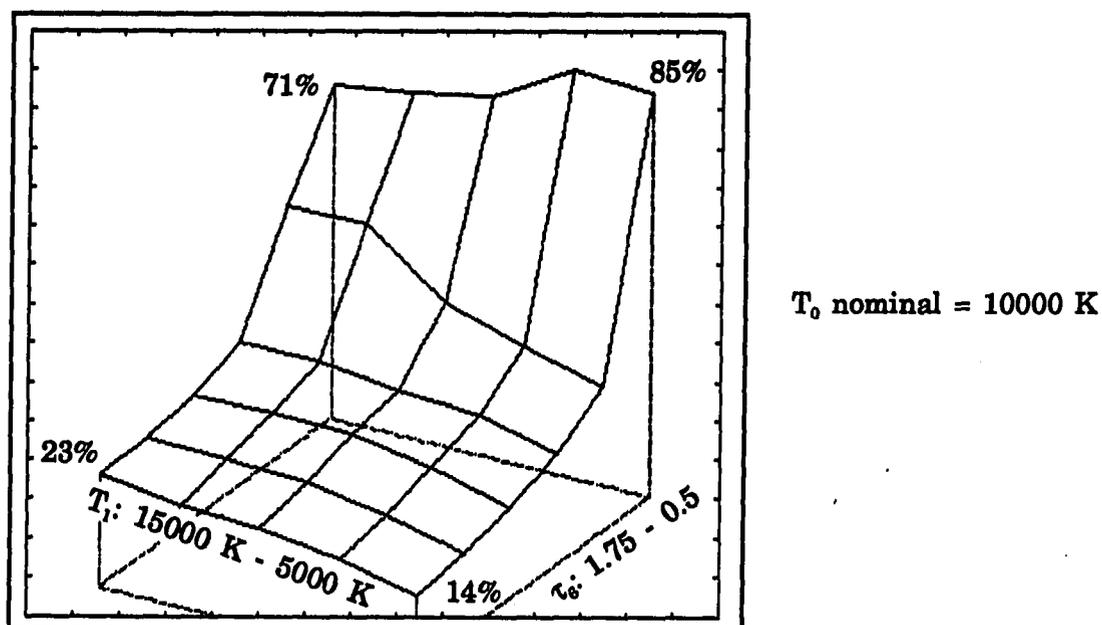


Figure 4.3: Standard deviation of parameter τ_6 estimated directly from noisy simulated flux density values

Figure 4.3 shows the standard deviation of parameter τ_6 as a function of the nominal values for parameters T_1 and τ_6 , $T_0 = 10000$ K. These values are also expressed as percentages of the nominal values. It is again clear that a 5% standard deviation in the values of the observed flux densities can result in large errors in the values of estimated parameter τ_6 . The standard deviation of parameter τ_6 is less sensitive to changes in the values of parameter T_1 than it is to changes in the value of parameter τ_6 ; it increases quite rapidly as parameter τ_6 decreases.

4.3 Double Smoothing Statistical Analysis

4.3.1 Flux Smoothing

A mathematical description of the flux smoothing (3×3 window) operation is given by

$$\hat{S}(i, j) = \frac{\sum_{l=-1}^1 \sum_{k=-1}^1 S(i+l, j+k)}{9.0} \quad (4.17)$$

where $S(i, j)$ is the observed flux density at pixel (i, j) and $\hat{S}(i, j)$ is the smoothed flux density at pixel (i, j) .

4.3.1.1 Assumption By investigation of the observed flux density maps it is clear that the variation in flux density over the entire map, at all three frequencies, are slow enough that the smoothing operation provides a good approximation of the average value of the flux density at each pixel, (i, j) . By assuming that the observed flux density at each pixel consists of signal plus zero mean white Gaussian noise with a variance of σ^2 , the variance of the noise at each pixel after smoothing operation will be given by $\sigma^2/9$.

4.3.2 Parameter Smoothing

A mathematical description of the parameter smoothing (3×3 window) operation is given by

$$\hat{T}(i, j) = \frac{\sum_{l=-1}^1 \sum_{k=-1}^1 T(i+l, j+k)}{9.0} \quad (4.18)$$

where

$T(i, j)$ = estimated parameter at pixel (i, j) , and

$\hat{T}(i, j)$ = smoothed estimated parameter at pixel (i, j) .

Due to the operation of the flux smoothing, the estimated parameters at the individual pixels are no longer statistically independent. The Monte Carlo method was again chosen to find the variances. It was found that the standard deviation of the estimated parameters using the double smoothing technique have exactly the same behavior shown in Figures 4.1, 4.2 and 4.3. The only exception was the actual numerical values of the standard deviations, which were reduced by a factor of ± 4 . The reason for the similar behavior can be contributed to the fact that the same white noise sequence was used to obtain the noisy flux density values used in the double smoothing technique as the white noise sequence used to obtain the noisy flux density values in the original parameter estimation problem. Although the double smoothing technique reduces the standard deviation of all three estimated parameters by a factor of ± 4 , the constraint that neighbor pixel values have flux density values of the same order, must be kept in mind. The double smoothing technique cannot be applied when there is no or low statistical dependency between neighbor pixel values.

4.4 Conclusion

The statistical analysis in this chapter reveals that the standard deviation of the estimated parameters using the double smoothing technique are lower by a factor of ± 4 than the standard deviation of the parameters estimated directly from the noisy flux density values. However, the standard deviations of the double smoothing

estimated parameters are still very large, keeping in mind that only 5% noise was added to the simulated flux density values. These errors must be kept in mind when interpreting the parameters estimated from the observed flux density values.

5. IMAGE RESTORATION

5.1 Introduction

Although the double smoothing technique, as defined in Chapter 3, gives acceptable results for the parameter estimation problem discussed in this dissertation, it is not clear if this method is optimal in any sense. What is meant by this is the following: in the double smoothing technique each pixel value in the image is replaced by a weighted sum of the eight neighbor pixel values and the value of the pixel under consideration. Also, the weight assigned to each contributing pixel value in the summation is exactly the same, $1/9$. The question that remains unanswered is if it is possible to do better by using only a subset of the eight neighbor pixel values, or use a larger set of neighbor pixel values, and by assigning different weights to the chosen neighbor pixel values when restoring an image. This problem was investigated in detail by Kashyap and Chellappa (1981, 1983). They use two-dimensional random field (RF) models to describe the image to be restored and the minimum mean square error (MMSE) criterion in the actual restoration. A summary of this method will be given in the following sections.

5.2 Spatial Interaction Models

Any $M \times M$ image can be described by the familiar two-dimensional array of finite gray levels as follows

$$\{y(s), s = (i, j) \in \Omega\}, \Omega = \{s = (i, j), 1 \leq i, j, \leq M\} \quad (5.1)$$

An important characteristic of an image described this way is the statistical dependency of the gray-level value at each pixel location on the gray-level values of all the neighbor pixels in all directions. One possible way to describe this dependency is by using spatial - interaction models, also known as random field models. These models describe the image gray level, $y(s)$, at pixel location s , as a linear combination of the neighbor pixel gray levels $\{y(s+r), r \in N\}$, where N is the neighbor set and does not include $(0, 0)$, and an additive noise. The spatial interaction models place no constraint on the neighbor sets used, they can be causal or noncausal and include any number of neighbors.

The spatial interaction models can be divided into two classes of models known as the simultaneous models and the conditional Markov (CM) models. The class of simultaneous models can be further divided into the simultaneous autoregressive (SAR) models, the simultaneous moving average (SMA) models and the simultaneous autoregressive and moving average (SARMA) models. The relationship between the simultaneous and CM models is as follows: for each SAR model there exists a CM model with the same spectral density function, but the converse is not true.

The reason why simultaneous models are still important although SAR models are a subset of CM models, is because SAR models are more parsimonious than CM

models and because a study of SAR models leads to SMA and SARMA models which are not a subset of CM models. The emphasis will be on SAR models from here on.

The image restoration problem always involves a noisy image to be restored. First of all, an appropriate SAR model must be fitted to this image. This will include a method to estimate the coefficients of the neighbor pixel values in a specific neighbor set used as well as a method to find the best neighbor set, from all possible neighbor sets, to be used for the image under consideration. After this is done, the image will be restored using a MMSE criterion and the estimated SAR model.

It must be pointed out that the restoration can be done treating the image either as a finite block defined over an infinite lattice or as an image defined over a finite lattice. The latter approach will be followed due to the simplification in the mathematical procedures involved (Kashyap and Chellappa, 1983). To be more specific, the image will be defined on a toroidal lattice. This has the advantage that the transformation matrix will be block circulant and therefore this matrix's eigenvalues can be written down exactly, as will be seen.

5.3 Image Description using SAR Models Defined on a Toroidal Lattice

As can be seen from equation (5.1), the image is defined over the finite lattice Ω , where

$$\Omega = \{s = (i, j), 1 \leq i, j \leq M\} \quad (5.2)$$

Partition the lattice Ω into two subsets, the interior set Ω_I and the boundary set Ω_B , which are mutually exclusive and totally inclusive. This will give

$$\Omega_B = \{s = (i, j); s \in \Omega \text{ and } (s+r) \notin \Omega \text{ for at least one } r \in N\} \quad (5.3)$$

$$\Omega_I = \Omega - \Omega_B \quad (5.4)$$

Given this partition, the toroidal lattice SAR model for a finite image is defined by (Chellappa and Kashyap, 1982) as

$$y(s) = \sum_{r \in N} \theta_r y(s+r) + \sqrt{\rho} w(s) \quad s \in \Omega_I \quad (5.5)$$

$$y_1(s) = \sum_{r \in N} \theta_r y_1(s+r) + \sqrt{\rho} w(s) \quad s \in \Omega_B \quad (5.6)$$

where

$$y_1((i, j) + (k, l)) = y((i, j) + (k, l)) \quad \text{if } ((i, j) + (k, l)) \in \Omega \quad (5.7)$$

$$y_1((i, j) + (k, l)) = y[\{(k+i-1)_{\text{mod } M}\} + 1, \{(j+l-1)_{\text{mod } M}\} + 1]$$

$$\text{if } ((i, j) + (k, l)) \notin \Omega$$

In these equations it was assumed that $y(s)$ is stationary over the whole image and that $w(s)$ is an i.i.d. noise sequence with zero mean and unity variance. The unknown parameters are $(\theta_r, r \in N)$ and ρ .

The two equations (5.5) and (5.6) used to describe the spatial relationship in the image, result in M^2 equations giving the image gray-level values at each pixel location in terms of the neighbor pixel gray-level values and the i.i.d. noise sequence $\{w(s)\}$. By using a lexicographic row ordered array format for $\{y(s)\}$ and $\{w(s)\}$, two vectors \mathbf{y} and \mathbf{w} , dimension $M^2 \times 1$, result which are related by

$$\mathbf{B}(\Theta)\mathbf{y} = \sqrt{\rho}\mathbf{w} \quad (5.8)$$

where $\mathbf{B}(\Theta)$ is a block circulant matrix:

$$\mathbf{B}(\Theta) = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \cdots & \mathbf{B}_{1,M} \\ \mathbf{B}_{1,M} & \mathbf{B}_{1,1} & \cdots & \mathbf{B}_{1,M-1} \\ \vdots & & & \\ \mathbf{B}_{1,2} & \mathbf{B}_{1,3} & \cdots & \mathbf{B}_{1,1} \end{bmatrix} \quad (5.9)$$

and $\Theta = \text{col}(\theta_r, r \in N)$, a vector of dimension $m \times 1$, where m is the number of neighbors used in the specific SAR model. An example will clarify this notation.

Example:

Suppose the following neighbor set about pixel $(0, 0)$ is used to describe the spatial relationship in an image

$$N = \{(-1, 0), (0, 1), (1, 0), (0, -1)\} \quad (5.10)$$

Graphically this neighbor set looks as follows:

$$\begin{array}{ccccc}
 & & (-1,0) & & \\
 & \cdot & \bullet & \cdot & \\
 & & (0,0) & & \\
 (0,-1) & \bullet & \times & \bullet & (0,1) \\
 & & (1,0) & & \\
 & \cdot & \bullet & \cdot &
 \end{array}$$

Using the toroidal SAR model for the image, a general equation for the gray-level value at each pixel location will be as follows

$$\begin{aligned}
 y(s) = & \theta_{-1,0}y(s + (-1,0)) + \theta_{0,1}y(s + (0,1)) + \theta_{1,0}y(s + (1,0)) + \\
 & \theta_{0,-1}y(s + (0,-1)) + \sqrt{\rho}w(s)
 \end{aligned} \tag{5.11}$$

Substituting the value for $s = (i, j)$ at each pixel location and taking the toroidal lattice assumption into account, the set of equations (5.13) results.

This matrix consists of a set of blocks $\mathbf{B}_{i,j}$. To understand the composition of these blocks, consider block $\mathbf{B}_{1,1}$, which is

$$\mathbf{B}_{1,1} = \begin{bmatrix} 1 & -\theta_{0,1} & 0 & \dots & 0 & -\theta_{0,-1} \\ -\theta_{0,-1} & 1 & -\theta_{0,1} & \dots & 0 & 0 \\ \vdots & & & & & \\ -\theta_{0,1} & 0 & \dots & 0 & -\theta_{0,-1} & 1 \end{bmatrix} \tag{5.12}$$

This block (and every other block) is circulant. To fill in the elements we can start by centering a 3×3 window on the (1, 1) element of the image.

$$\begin{bmatrix}
 1 & -\theta_{0,1} & 0 & \dots & -\theta_{0,-1} & -\theta_{1,0} & 0 & \dots & 0 & \phi & -\theta_{-1,0} & 0 & \dots & 0 \\
 -\theta_{0,-1} & 1 & -\theta_{0,1} & \dots & 0 & -\theta_{1,0} & \dots & 0 & \phi & 0 & -\theta_{-1,0} & \dots & 0 & 0 \\
 \vdots & \vdots \\
 -\theta_{0,1} & 0 & \dots & 1 & 0 & 0 & \dots & -\theta_{1,0} & \phi & 0 & 0 & \dots & -\theta_{-1,0} & 0 \\
 -\theta_{-1,0} & 0 & 0 & \dots & 0 & 1 & -\theta_{0,1} & \dots & -\theta_{0,-1} & 0 & 0 & 0 & 0 & 0 \\
 \vdots & \vdots
 \end{bmatrix}
 \begin{bmatrix}
 y(1,1) \\
 y(1,2) \\
 y(1,3) \\
 \vdots \\
 y(1,M) \\
 y(2,1) \\
 \vdots \\
 y(M,M-1) \\
 y(M,M)
 \end{bmatrix}
 = \sqrt{\rho} w(s)$$

(5.13)

Then the (1, 1) element of $\mathbf{B}_{1,1}$ is 1 because no weighting applied to $y(1, 1)$. To get the (1, 2) element of $\mathbf{B}_{1,1}$ we go to the east neighbor of $y(1, 1)$, which is $y(1, 2)$, and use the coefficient $\theta_{0,1}$. The element is $-\theta_{0,1}$ because the $y(s+r)$ terms in equation (5.11) have been moved to the left hand side in equation (5.3). The (1, 3) element of $\mathbf{B}_{1,1}$ is zero because it lies outside the neighbor set used. The following elements in the first row are zero for the same reason. The last element in the first row is $-\theta_{0,-1}$ because $\mathbf{B}_{1,1}$ is constructed as block circulant. The remaining elements of $\mathbf{B}_{1,1}$ can be realized from its block circulant characteristics.

It is clear from this example that the block circulant matrix satisfies the general form given in equation (5.9) with

$$\mathbf{B}_{1,1} = \text{circulant}(1, -\theta_{0,1}, 0, \dots, -\theta_{0,-1}),$$

$$\mathbf{B}_{1,2} = \text{circulant}(-\theta_{1,0}, 0, 0, \dots, 0),$$

$$\mathbf{B}_{1,M} = \text{circulant}(-\theta_{-1,0}, 0, 0, \dots, 0), \text{ and}$$

$$\mathbf{B}_{i,j} = \mathbf{0} \quad j \neq 1, 2, M.$$

It follows from equation (5.8) that

$$\mathbf{y} = \sqrt{\rho} \mathbf{B}(\Theta)^{-1} \mathbf{w} \quad (5.14)$$

The image covariance matrix of \mathbf{y} can be computed as

$$\begin{aligned} \mathbf{Q}_y &= \text{cov}(\mathbf{y}) \\ &= E(\mathbf{y} \mathbf{y}^T) \\ &= E(\sqrt{\rho} \mathbf{B}(\Theta)^{-1} \mathbf{w} \mathbf{w}^T [\mathbf{B}(\Theta)^{-1}]^T \sqrt{\rho}) \\ &= \rho (\mathbf{B}(\Theta)^{-1} E(\mathbf{w} \mathbf{w}^T) [\mathbf{B}(\Theta)^{-1}]^T) \\ &= \rho (\mathbf{B}(\Theta)^{-1} [\mathbf{B}(\Theta)^{-1}]^T) \end{aligned}$$

$$\begin{aligned}
&= \rho(\mathbf{B}(\Theta)^{-1}[\mathbf{B}(\Theta)^T]^{-1}) \\
&= \rho(\mathbf{B}^T(\Theta)\mathbf{B}(\Theta))^{-1}
\end{aligned} \tag{5.15}$$

This derivation depends on the fact that the matrix $\mathbf{B}(\Theta)$ has an inverse. Chellappa and Kashyap (1982) have shown that a necessary and sufficient condition for $\mathbf{B}(\Theta)^{-1}$ to exist is that all the eigenvalues of the block circulant transformation matrix $\mathbf{B}(\Theta)$, namely u_s , $s \in \Omega$, not be zero. Therefore

$$u_s = (1 - \Theta^T \Psi_s) \neq 0 \tag{5.16}$$

where

$$\begin{aligned}
\Psi_s &= \text{col}[\exp\sqrt{-1}(\frac{2\pi}{M})(s^T - (1, 1)r), r \in N] \\
&= \text{col}[\exp\sqrt{-1}(\frac{2\pi}{M})((i-1)k + (j-1)l), (k, l) \in N]
\end{aligned}$$

The deviation leading to equation (5.16) can be found in the appendix of Chellappa and Kashyap (1982).

5.4 Image Restoration using SAR Models

The image restoration method discussed from here on is based on the MMSE principle. First of all the degradation model must be defined. This model is shown in Figure 5.1.

The qualities are defined by:

H is a non-separable, spatially invariant, periodic point spread function (PSF),

$\eta(i, j)$ is a zero mean, signal independent, additive noise of variance γ .

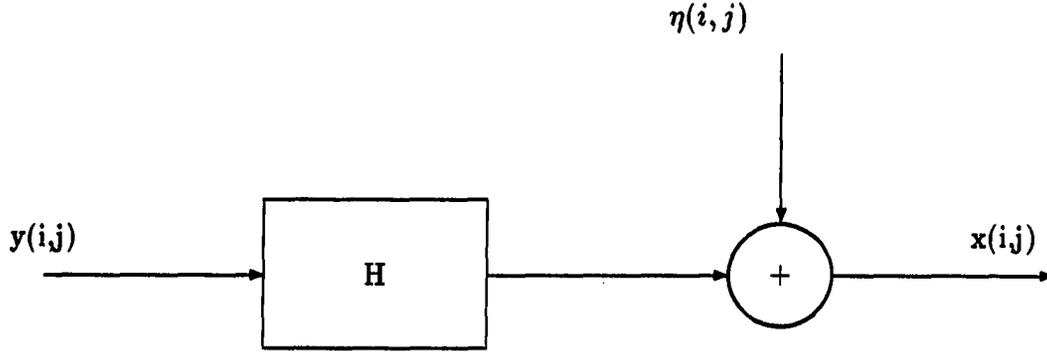


Figure 5.1: Degradation model

Let \mathbf{x} , \mathbf{y} and η represent lexicographic ordered arrays of the degraded image, original image and additive noise, respectively. It then follows that the degradation model is given by the following mathematical equation

$$\mathbf{x} = \mathbf{H}\mathbf{y} + \eta \quad (5.17)$$

where \mathbf{H} is the block circulant matrix. The fact that \mathbf{H} is a block circulant matrix follows from the requirement that the PSF is periodic and is shown as such in Gonzalez and Wintz (1987, p. 212).

Andrews and Hunt (1977) have shown that, using this degradation model, the MMSE estimate of \mathbf{y} , is given by the following equation

$$\hat{\mathbf{y}} = \mathbf{Q}_y \mathbf{H}^T (\mathbf{H} \mathbf{Q}_y \mathbf{H}^T + \gamma \mathbf{I})^{-1} \mathbf{x} \quad (5.18)$$

where \mathbf{Q}_y is the covariance matrix of the original image \mathbf{y} as already defined.

Chellappa and Kashyap (1982) suggested the following distinction between three

different cases in the image restoration problem:

1. A prototype of the original image is available and an SAR model with parameters (Θ, ρ) is assumed for the undegraded image y . These parameters can be estimated from the prototype image.
2. A prototype of the original image is not available and an SAR model with parameters (Θ', ρ') is assumed for the degraded image x . These parameters can be estimated from the given degraded image x .
3. A prototype of the original image is not available and an SAR model with parameters (Θ, ρ) is assumed for the original image. These parameters can be estimated from the given degraded image x .

Only cases 2 and 3 are of interest in this study, because a prototype of the original image is never available in radio astronomy data.

5.4.1 Image Restoration using an SAR model for the Degraded Image x

The SAR model for the degraded image is given by the following equations

$$x(s) = \sum_{r \in N} \theta'_r x(s+r) + \sqrt{\rho'} w(s) \quad s \in \Omega_I \quad (5.19)$$

$$x(s) = \sum_{r \in N} \theta'_r x_1(s+r) + \sqrt{\rho'} w(s) \quad s \in \Omega_B \quad (5.20)$$

where $x_1(s+r)$ has the same definition as $y_1(s+r)$ in equation (5.6).

The covariance matrix for the degraded image is given by

$$\begin{aligned}
\mathbf{Q}_x &= E\{[\mathbf{H}\mathbf{y} + \eta][\mathbf{H}\mathbf{y} + \eta]^T\} \\
&= E\{[\mathbf{H}\mathbf{y} + \eta][\mathbf{y}^T \mathbf{H}^T + \eta^T]\} \\
&= E\{\mathbf{H}\mathbf{y}\mathbf{y}^T \mathbf{H}^T + \eta\mathbf{y}^T \mathbf{H}^T + \mathbf{H}\mathbf{y}\eta^T + \eta\eta^T\} \\
&= \mathbf{H}\mathbf{Q}_y \mathbf{H}^T + \gamma \mathbf{I}
\end{aligned} \tag{5.21}$$

where $\gamma = E\{\eta\eta^T\}$. It then follows that the covariance matrix of the original image is given by

$$\mathbf{Q}_y = \mathbf{H}^{-1}(\mathbf{Q}_x - \gamma \mathbf{I})(\mathbf{H}^T)^{-1} \tag{5.22}$$

Substituting this expression back in the equation for the MMSE estimate of \mathbf{y} gives

$$\hat{\mathbf{y}} = \mathbf{H}^{-1}(\mathbf{Q}_x - \gamma \mathbf{I})(\mathbf{H}^T)^{-1} \mathbf{H}^T [\mathbf{H}\mathbf{H}^{-1}(\mathbf{Q}_x - \gamma \mathbf{I})(\mathbf{H}^T)^{-1} \mathbf{H}^T + \gamma \mathbf{I}]^{-1} \mathbf{x}$$

or

$$\hat{\mathbf{y}} = \mathbf{H}^{-1}(\mathbf{Q}_x - \gamma \mathbf{I})\mathbf{Q}_x^{-1} \mathbf{x} \tag{5.23}$$

Two important matrix equations concerning block circulant matrixes are in order here to simplify the expression above. Gonzalez and Wintz (1987, p. 214) have shown that if \mathbf{H} is a block circulant matrix for an $N \times N$ image, then

$$\mathbf{H} = \mathbf{W}\mathbf{D}\mathbf{W}^{-1} \tag{5.24}$$

and

$$\mathbf{H}^T = \mathbf{W}\mathbf{D}^* \mathbf{W}^{-1} \tag{5.25}$$

where \mathbf{D} is a diagonal matrix whose elements $D(k, k)$ are related to the eigenvalues of \mathbf{H} , and

$$\mathbf{W}(i, m) = w_M(i, m)\mathbf{W}_N \quad i, m = 0, 1, \dots, M-1 \quad (5.26)$$

where

$$W_N(k, n) = w_N(k, n) \quad k, n = 0, 1, \dots, M-1, \quad (5.27)$$

$$w_M(i, m) = \exp[j\frac{2\pi}{M}im] \quad i, m = 0, 1, \dots, M-1, \quad (5.28)$$

$$w_N(k, n) = \exp[j\frac{2\pi}{M}kn] \quad k, n = 0, 1, \dots, M-1 \quad (5.29)$$

and $j = \sqrt{-1}$. This formulation implies that \mathbf{W} is a $M^2 \times M^2$ matrix which was broken down into M^2 partitions of size $M \times M$. \mathbf{W}^{-1} is defined analogous by

$$\mathbf{W}^{-1}(i, m) = \frac{1}{M}w_M(i, m)\mathbf{W}_N^{-1} \quad i, m = 0, 1, \dots, M-1 \quad (5.30)$$

where

$$W_N^{-1}(k, n) = \frac{1}{M}w_N^{-1}(k, n) \quad k, n = 0, 1, \dots, M-1 \quad (5.31)$$

$$w_M^{-1}(i, m) = \exp[-j\frac{2\pi}{M}im] \quad i, m = 0, 1, \dots, M-1 \quad (5.32)$$

and

$$w_N^{-1}(k, n) = \exp[-j\frac{2\pi}{M}kn] \quad k, n = 0, 1, \dots, M-1 \quad (5.33)$$

Substituting all these values back, the matrix in equation (5.34) is obtained.

$$\mathbf{W} = \begin{bmatrix}
 1 & 1 & \dots & 1 & & 1 & 1 & \dots & 1 & \dots \\
 1 & e^{\frac{j2\pi}{M}} & & e^{\frac{j2\pi}{M}(M-1)} & & 1 & & & & \dots \\
 \vdots & & & \vdots & & & & & & \\
 1 & e^{\frac{j2\pi}{M}(M-1)} & & e^{\frac{j2\pi}{M}(M-1)^2} & & 1 & & e^{\frac{j2\pi}{M}(M-1)^2} & & \dots \\
 1 & 1 & \dots & 1 & & & & & & \dots \\
 1 & e^{\frac{j2\pi}{M}} & & e^{\frac{j2\pi}{M}(M-1)} & & & & & & \dots \\
 \vdots & & & \vdots & & & & & & \\
 1 & e^{\frac{j2\pi}{M}(M-1)} & & e^{\frac{j2\pi}{M}(M-1)^2} & & & & & & \\
 \vdots & & & \vdots & & & & & & \\
 1 & 1 & \dots & 1 & & & e^{\frac{j2\pi}{M}(M-1)^2} & \dots & e^{\frac{j2\pi}{M}(M-1)^2} & \\
 1 & e^{\frac{j2\pi}{M}} & & e^{\frac{j2\pi}{M}(M-1)} & & \dots & e^{\frac{j2\pi}{M}(M-1)^2} & \dots & e^{\frac{j2\pi}{M}[(M-1)^2+(M-1)]} & \\
 \vdots & & & \vdots & & & & & & \\
 1 & e^{\frac{j2\pi}{M}(M-1)} & & e^{\frac{j2\pi}{M}(M-1)^2} & & \dots & e^{\frac{j2\pi}{M}(M-1)^2} & & e^{\frac{j2\pi}{M}[(M-1)^2+(M-1)^2]} &
 \end{bmatrix}$$

(5.34)

Because \mathbf{H} , and therefore \mathbf{Q}_x , are both block circulant matrixes, equation (5.24) can be substituted into equation (5.23) to give

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{W}\mathbf{D}_H^{-1}\mathbf{W}^{-1}(\mathbf{W}\mathbf{D}_{Q_x}\mathbf{W}^{-1} - \gamma\mathbf{I})\mathbf{W}(\mathbf{D}_{Q_x})^{-1}\mathbf{W}^{-1}\mathbf{x} \\ &= \mathbf{W}\mathbf{D}_H^{-1}(\mathbf{D}_{Q_x} - \gamma\mathbf{I})(\mathbf{D}_{Q_x})^{-1}\mathbf{W}^{-1}\mathbf{x}\end{aligned}\quad (5.35)$$

By writing out this expression, using the values for \mathbf{W} in equation (5.34), it follows that

$$\hat{\mathbf{y}} = \frac{1}{M^2} \sum_{i,j} \left\{ \mathbf{f}_{ij} \left[\frac{\rho' - \gamma \|u_{ij}\|^2}{\rho' \bar{h}_{ij}} \right] \mathbf{f}_{ij}^{*T} \mathbf{x} \right\} \quad (5.36)$$

where

$$\mathbf{f}_{ij} = \text{col}[\mathbf{t}_j, \lambda_i \mathbf{t}_j, \dots, \lambda_i^{M-1} \mathbf{t}_j] \quad M^2 \times 1 \text{ vector,}$$

$$\mathbf{t}_j = \text{col}[1, \lambda_j, \lambda_j^2, \dots, \lambda_j^{M-1}] \quad M \times 1 \text{ vector,}$$

$$\lambda_i = \exp[\sqrt{-1} 2\pi(i-1)/M], \text{ and}$$

\bar{h}_{ij} and $\rho' / \|u_{ij}\|^2$ are the eigenvalues of the block circulant matrixes \mathbf{H} and \mathbf{Q}_x , respectively.

Equation (5.36) is an optimal estimate for $\hat{\mathbf{y}}$ if Θ' and ρ' are known exactly (Chellappa and Kashyap, 1982). Because these parameters must be estimated from the degraded image in practice, deviations from the optimal can be expected.

An estimate for γ can also be obtained from the degraded image. The usual procedure is to use the high frequency components of the spectral density function of the degraded image as an estimate of γ . In the case of a 128×128 image a possible estimate for γ would be given by

$$\gamma = \frac{1}{16} \sum_{s \in \Omega_s} \frac{\rho'}{\|u_s\|^2} \quad (5.37)$$

where

$\Omega_s = [s = (i, j), 61 \leq i, j \leq 64]$. This equation is valid because the eigenvalues of the covariance matrix Q_x are the same as the values of the discrete spectral density defined as the discrete 2-D Fourier Transform of the covariance function (Kashyap, 1980b).

5.5 Estimation of SAR Model

To be able to apply equation (5.36) to restore an image, it is first of all necessary to determine the SAR model needed to describe the image. This problem can be broken down into two parts: Given a specific neighbor set, it is necessary to determine the coefficients of these neighbors to be used in an SAR model. In the second part it is necessary to find the best neighbor set to be used for a specific image giving all possible neighbor sets.

5.5.1 Estimation of Neighbor Coefficient Parameters

One possible way to estimate the coefficients of the neighbors in an SAR model is by using ordinary least squares (LS). However, this approach has the disadvantage that the LS estimates are not consistent for toroidal lattice SAR models using nonunilateral neighbor sets (Ord, 1975 and Whittle, 1954). An alternative method is to use maximum likelihood (ML) estimates. This method requires computationally expensive algorithms because for Gaussian SAR models with nonunilateral neighbor

sets, the loglikelihood function is nonquadratic. Kashyap and Chellappa (1983) have derived the following approximate iterative estimation scheme which gives estimates close to the ML estimates and is computationally attractive

$$\Theta_{t+1} = [\mathbf{R} - \frac{1}{\rho_t} \mathbf{S}]^{-1} [\mathbf{V} - \frac{1}{\rho_t} \mathbf{U}] \quad t = 0, 1, \dots, \text{ and}$$

$$\rho_t = \frac{1}{M^2} \sum_{\Omega} [\mathbf{x}(s) - \Theta_t^T \mathbf{z}(s)]^2 \quad t = 0, 1, 2, 3, \dots$$

where

$$\mathbf{z}(s) = \text{col}[\mathbf{x}(s+r), r \in N],$$

$$\mathbf{S} = \sum_{\Omega} \mathbf{z}(s) \mathbf{z}^T(s),$$

$$\mathbf{U} = \sum_{\Omega} \mathbf{z}(s) \mathbf{x}(s),$$

$$\mathbf{V} = \sum_{\Omega} \mathbf{C}_s,$$

$$\mathbf{R} = \sum_{\Omega} \mathbf{S}_s \mathbf{S}_s^T - \mathbf{C}_s \mathbf{C}_s^T,$$

$$\mathbf{C}_s = \text{col}[\text{Cos} \frac{2\pi}{M} \{s^T - (1, 1)\}r, r \in N],$$

$$\mathbf{S}_s = \text{col}[\text{Sin} \frac{2\pi}{M} \{s^T - (1, 1)\}r, r \in N], \text{ and}$$

$$\Theta_0 = \mathbf{S}^{-1} \mathbf{U}.$$

This iterative procedure is started by using the initial value of $\Theta_0 = \mathbf{S}^{-1} \mathbf{U}$. It was found that this scheme normally converges in 3-4 iterations.

5.5.2 Choice of Neighbor Set

After the ML estimates of the neighbor coefficients have been determined for a number of different neighbor sets, it is necessary to find the neighbor set which best describes the image under consideration. There are three possible approaches to choose between different neighbor sets:

1. Pairwise Hypothesis Testing.

2. Akaike's Information Criterion (AIC).

3. Bayes Approach.

The pairwise hypothesis testing method suffers from the fact that it is not transitive nor consistent (Kashyap, 1977). The AIC method is in general transitive, but not consistent (Kashyap, 1980a). Kashyap and Chellappa (1983) have shown that using the Bayes decision rules, the following test statistic results

$$C_k = \left\{ - \sum_{s \in \Omega} \ln(1 - 2\Theta_k^T \mathbf{C}_{ks} + \Theta_k^T \mathbf{Q}_{ks} \Theta_k) + M^2 \ln \rho_k + m_k \ln(M^2) \right\} \quad (5.38)$$

where

$$\mathbf{Q}_{ks} = \sum_{\Omega} \mathbf{S}_{ks} \mathbf{S}_{ks}^T + \mathbf{C}_{ks} \mathbf{C}_{ks}^T,$$

$$\mathbf{C}_{ks} = \text{col} \left[\text{Cos} \frac{2\pi}{M} \{s^T - (1, 1)\} r, r \in N_k \right],$$

$$\mathbf{S}_{ks} = \text{col} \left[\text{Sin} \frac{2\pi}{M} \{s^T - (1, 1)\} r, r \in N_k \right],$$

$\Theta_k = \Theta$ estimated for the k-the neighbor set,

$m_k =$ number of neighbors of k-the neighbor set, and

$\rho_k = \rho$ estimated for the k-the neighbor set.

The neighbor set N_k with the lowest value of C_k is then chosen as the best neighbor set for the specific image under consideration.

5.6 Synthetic Image Generation

To be able to test how close the estimated parameters, for an image obeying a specific SAR model, are to their correct values, such an image must first be generated.

This can be done using equation (5.14) as follows

$$\begin{aligned} \mathbf{y} &= \mathbf{B}(\Theta)^{-1} \sqrt{\rho} \mathbf{w} \\ &= \mathbf{W} \mathbf{D}_\theta^{-1} \mathbf{W}^{-1} \sqrt{\rho} \mathbf{w} \end{aligned} \quad (5.39)$$

The second step is true because $\mathbf{B}(\Theta)^{-1}$ is block circulant and we can use equation (5.24). \mathbf{W} , \mathbf{D}_θ and \mathbf{W}^{-1} were defined before. By evaluating the matrix multiplications above, keeping in mind that \mathbf{D}_θ is a diagonal matrix consisting of the eigenvalues of the block circulant matrix $\mathbf{B}(\Theta)$, it can be shown that

$$\mathbf{y} = \sum_{i,j} \frac{\mathbf{f}_{ij} \sqrt{\rho} (\mathbf{f}_{ij}^*)^T}{M^2 \mu_{ij}} \mathbf{w} = \frac{\sqrt{\rho}}{M^2} \sum_{i,j} \frac{\mathbf{f}_{ij} s_{ij}}{\mu_{ij}} \quad (5.40)$$

where

$$s_{ij} = (\mathbf{f}_{ij}^*)^T \mathbf{w},$$

μ_{ij} = eigenvalues of block circulant matrix $\mathbf{B}(\Theta)^{-1}$, and

\mathbf{f}_{ij} = Fourier vectors defined before.

The image defined by equation (5.40) can be generated using two dimensional FFT algorithms. This is best explained by an example.

Example:

Consider the generation of a 2×2 synthetic image. Using the definitions following equations (5.36) and (5.40) the following expressions are obtained

$$\lambda_1 = e^{j2\pi \frac{(0)}{2}} \quad \lambda_2 = e^{j2\pi \frac{(1)}{2}}$$

$$t_1 = \begin{bmatrix} 1 \\ e^{j2\pi\frac{(0)}{2}} \end{bmatrix} \quad t_2 = \begin{bmatrix} 1 \\ e^{j2\pi\frac{(1)}{2}} \end{bmatrix}$$

$$f_{11} = \begin{bmatrix} 1 \\ e^{j2\pi\frac{(0)}{2}} \\ e^{j2\pi\frac{(0)}{2}} \\ e^{j2\pi\frac{(0)}{2}} e^{j2\pi\frac{(0)}{2}} \end{bmatrix} \quad f_{12} = \begin{bmatrix} 1 \\ e^{j2\pi\frac{(1)}{2}} \\ e^{j2\pi\frac{(0)}{2}} \\ e^{j2\pi\frac{(0)}{2}} e^{j2\pi\frac{(1)}{2}} \end{bmatrix}$$

$$f_{21} = \begin{bmatrix} 1 \\ e^{j2\pi\frac{(0)}{2}} \\ e^{j2\pi\frac{(1)}{2}} \\ e^{j2\pi\frac{(1)}{2}} e^{j2\pi\frac{(0)}{2}} \end{bmatrix} \quad f_{22} = \begin{bmatrix} 1 \\ e^{j2\pi\frac{(1)}{2}} \\ e^{j2\pi\frac{(1)}{2}} \\ e^{j2\pi\frac{(1)}{2}} e^{j2\pi\frac{(1)}{2}} \end{bmatrix}$$

Therefore

$$s_{11} = (1 \cdot w(1) + e^{-j2\pi\frac{(0)}{2}} w(2) + e^{-j2\pi\frac{(0)}{2}} w(3) + e^{-j2\pi\frac{(0)}{2}} e^{-j2\pi\frac{(0)}{2}} w(4))$$

$$s_{12} = (1 \cdot w(1) + e^{-j2\pi\frac{(1)}{2}} w(2) + e^{-j2\pi\frac{(0)}{2}} w(3) + e^{-j2\pi\frac{(0)}{2}} e^{-j2\pi\frac{(1)}{2}} w(4))$$

$$s_{21} = (1.w(1) + e^{-j2\pi\frac{(0)}{2}} w(2) + e^{-j2\pi\frac{(1)}{2}} w(3) + e^{-j2\pi\frac{(1)}{2}} e^{-j2\pi\frac{(0)}{2}} w(4))$$

$$s_{22} = (1.w(1) + e^{-j2\pi\frac{(1)}{2}} w(2) + e^{-j2\pi\frac{(1)}{2}} w(3) + e^{-j2\pi\frac{(1)}{2}} e^{-j2\pi\frac{(1)}{2}} w(4))$$

It is clear that these Fourier components of a white noise lexicographically ordered image, $w(s)$, can be computed using a 2-D FFT directly. Next, we evaluate equation (5.40). Set $p_{ij} = \frac{\sqrt{\rho}}{M^2} \frac{s_{ij}}{\mu_{ij}}$.

It then follows that $y = f_{11}p_{11} + f_{12}p_{12} + f_{21}p_{21} + f_{22}p_{22}$,

or

$$\begin{bmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \end{bmatrix} = \begin{bmatrix} p_{11} \\ p_{12} \\ p_{21} \\ p_{22} \end{bmatrix} + \begin{bmatrix} p_{12} \\ e^{j2\pi\frac{(1)}{2}} p_{12} \\ p_{12} \\ e^{j2\pi\frac{(1)}{2}} p_{12} \end{bmatrix} + \begin{bmatrix} p_{21} \\ p_{21} \\ e^{j2\pi\frac{(1)}{2}} p_{21} \\ e^{j2\pi\frac{(1)}{2}} p_{21} \end{bmatrix} + \begin{bmatrix} p_{22} \\ e^{j2\pi\frac{(1)}{2}} p_{22} \\ e^{j2\pi\frac{(1)}{2}} p_{22} \\ e^{j2\pi\frac{(2)}{2}} p_{22} \end{bmatrix}$$

It can be seen by inspection that this can be computed using an inverse 2-D FFT directly. Therefore the synthetic image can be generated using 2-D FFTs.

The unknowns in equation (5.40) are the eigenvalues, μ_{ij} , of the block circulant matrix $\mathbf{B}(\Theta)^{-1}$. Chellappa and Kashyap (1982) have shown that these eigenvalues are given by

$$\mu_{ij} = 1 + \sum_{k, l \in N} \theta_{k, l} \lambda_i^k \lambda_j^l \quad (5.41)$$

By arbitrarily choosing values for $\theta_{k, l}$ and ρ , synthetic images can be generated by using equation (5.40); $w(s)$ is a white noise sequence with zero mean and unit variance as before. Results of computer experiments using the above theory of image generation are shown in the next chapter.

6. RANDOM FIELD RESTORATION RESULTS

6.1 Synthetic Image Generation

The theory in Section 5.6 was used to numerically generate synthetic images obeying different SAR models. The parameters used for the different SAR models were the same as those used by Chellappa and Kashyap (1981) and are given in Table 6.1. The same white noise sequence with zero mean and unity variance was used in the generation of all the images. The image gray-level values for each image were transformed to lie between 0 and 255. The sixteen synthetic generated images are displayed in Figure 6.1.

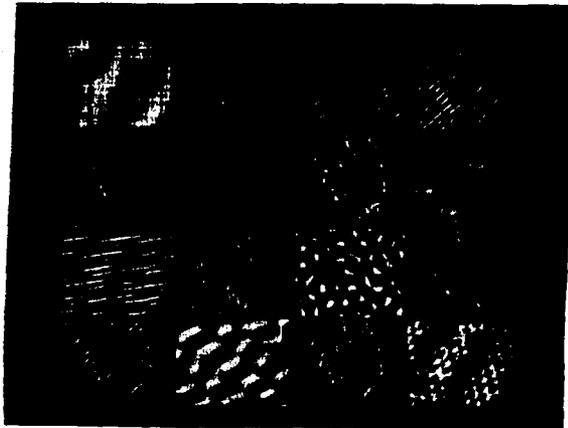


Figure 6.1: Synthetic generated images

Table 6.1: Synthetic generated image parameters

Image	Neighbor Set and Neighbor Coefficients
1,1	$N = \{(-1,0); (0,-1); (-1,-1)\}$ $\theta_{-1,0} = 0.9704; \theta_{0,-1} = 0.9735; \theta_{-1,-1} = -0.9686$
1,2	$N = \{(-1,0); (1,0); (0,-1)\}$ $\theta_{-1,0} = 0.18; \theta_{1,0} = 0.18; \theta_{0,-1} = -0.12$
1,3	$N = \{(-1,0); (0,-1); (-1,-1); (1,-1)\}$ $\theta_{-1,0} = 0.18; \theta_{0,-1} = 1.1011; \theta_{-1,-1} = -1.039; \theta_{1,-1} = -0.1806$
1,4	$N = \{(-1,0); (0,1); (1,1); (-1,1)\}$ $\theta_{-1,0} = 0.18; \theta_{0,1} = 1.1011; \theta_{1,1} = -1.039; \theta_{-1,1} = -0.1806$
2,1	$N = \{(0,1); (1,0)\}$ symmetric $\theta_{0,1} = \theta_{0,-1} = 0.26; \theta_{1,0} = \theta_{-1,0} = -0.12$
2,2	$N = \{(1,-1); (1,1)\}$ symmetric $\theta_{1,-1} = \theta_{-1,1} = -0.14; \theta_{1,1} = \theta_{-1,-1} = 0.28$
2,3	$N = \{(0,1); (0,-1); (-1,0); (1,0); (1,1)\}$ $\theta_{0,1} = 0.12; \theta_{0,-1} = 0.15; \theta_{-1,0} = 0.18; \theta_{1,0} = 0.1; \theta_{1,1} = 0.11$
2,4	$N = \{(1,-1); (0,1); (1,1)\}$ symmetric $\theta_{1,-1} = \theta_{-1,1} = 0.23; \theta_{0,1} = \theta_{0,-1} = -0.14; \theta_{1,1} = \theta_{-1,-1} = 0.22$
3,1	$N = \{(1,-1); (0,1); (1,1); (1,0)\}$ symmetric $\theta_{1,-1} = \theta_{-1,1} = -0.2480; \theta_{0,1} = \theta_{0,-1} = 0.5081$ $\theta_{1,1} = \theta_{-1,-1} = -0.2874; \theta_{1,0} = \theta_{-1,0} = 0.5256$
3,2	$N = \{(0,1); (1,0); (0,2); (2,0)\}$ symmetric $\theta_{0,1} = \theta_{0,-1} = 0.20; \theta_{1,0} = \theta_{-1,0} = -0.10$ $\theta_{0,2} = \theta_{0,-2} = -0.15; \theta_{2,0} = \theta_{-2,0} = 0.20$
3,3	$N = \{(0,1); (1,0)\}$ symmetric $\theta_{0,1} = \theta_{0,-1} = 0.1825; \theta_{1,0} = \theta_{-1,0} = 0.3794$

Table 6.1 (continued)

Image	Neighbor Set and Neighbor Coefficients
3,4	$N = \{(1,0); (1,1); (0,1)\}$ symmetric $\theta_{1,0} = \theta_{-1,0} = 0.18; \theta_{1,1} = \theta_{-1,-1} = 0.22; \theta_{0,1} = \theta_{0,-1} = -0.10$
4,1	$N = \{(1,0); (1,-1); (1,1)\}$ symmetric $\theta_{1,0} = \theta_{-1,0} = 0.12; \theta_{1,-1} = \theta_{-1,1} = 0.28; \theta_{1,1} = \theta_{-1,-1} = -0.14$
4,2	$N = \{(-1,0); (-2,0); (0,-1); (0,-2); (-1,-1)\}$ $\theta_{-1,0} = 1.0388; \theta_{-2,0} = -0.1814; \theta_{0,-1} = 0.9046; \theta_{0,-2} = -0.1088;$ $\theta_{-1,-1} = -0.7288$
4,3	$N = \{(1,0); (1,-1); (0,1)\}$ symmetric $\theta_{1,0} = \theta_{-1,0} = 0.28; \theta_{1,-1} = \theta_{-1,1} = 0.22; \theta_{0,1} = \theta_{0,-1} = -0.14$
4,4	$N = \{(0,1); (1,0); (1,1); (1,-1)\}$ symmetric $\theta_{0,1} = \theta_{0,-1} = 0.5246; \theta_{1,0} = \theta_{-1,0} = 0.5357$ $\theta_{1,1} = \theta_{-1,-1} = -0.3126; \theta_{1,-1} = \theta_{-1,1} = -0.25$

It is obvious from Figure 6.1 that different SAR models give rise to different characteristics in the generated images. Different white noise sequences were used with the same SAR models to generate more synthetic image sets. It was found that the different white noise sequences do not change the characteristics of the images because the characteristics are determined by the SAR model used.

6.2 SAR Model Parameter Estimation

The approximate maximum likelihood parameter estimation technique, as discussed in Section 5.5.1, was used to estimate the SAR model parameters from the last four synthetic generated images. These estimated parameters were then compared with the original SAR model parameters used to generate the images. The results are given in Table 6.2.

It can be seen that the estimated parameters, using the approximate maximum likelihood estimation technique, are very close to the original parameters used in the image generation. The difference between the estimated parameters and the original parameters is a function of the SAR model used to generate the synthetic image and varies from 14% for image (4,1) to 1.18% for image (4,2) (matrix notation is used to refer to the different images in Figure 6.1). The minimum mean square error restoration of images is a function of the SAR model used to describe the images (see Section 5.4). The best restoration is obtained when the underlying SAR model describing an image is known exactly. However, in practice the model is estimated from the noisy image to be restored because the original image is seldom available. (What is the sense of restoring an image when the original is available?) This fact must always be kept in mind and in this sense an error of 14% in the underlying model is not too extreme.

6.3 Calculation of Test Statistics

To test the effectiveness of the test statistic given in Section 5.5.2, image (4,1) was used as an example. To find the SAR model that best describes this image, different

Table 6.2: Maximum likelihood estimated parameters

Image	Neighbor Coordinate	Original Parameter	Estimated Parameter
4,1	1,0	$\theta_{1,0} = 0.1200$	$\hat{\theta}_{1,0} = 0.1388$
	-1,0	$\theta_{-1,0} = 0.1200$	$\hat{\theta}_{-1,0} = 0.1388$
	1,-1	$\theta_{1,-1} = 0.2800$	$\hat{\theta}_{1,-1} = 0.3168$
	-1,1	$\theta_{-1,1} = 0.2800$	$\hat{\theta}_{-1,1} = 0.3168$
	1,1	$\theta_{1,1} = -0.1400$	$\hat{\theta}_{1,1} = -0.1587$
	-1,-1	$\theta_{-1,-1} = -0.1400$	$\hat{\theta}_{-1,-1} = -0.1587$
		$\rho = 1.111$	$\hat{\rho} = 1.003$
4,2	-1,0	$\theta_{-1,0} = 1.0388$	$\hat{\theta}_{-1,0} = 1.0301$
	-2,0	$\theta_{-2,0} = -0.1814$	$\hat{\theta}_{-2,0} = -0.1804$
	0,-1	$\theta_{0,-1} = 0.9046$	$\hat{\theta}_{0,-1} = 0.8996$
	0,-2	$\theta_{0,-2} = -0.1088$	$\hat{\theta}_{0,-2} = -0.1055$
	-1,-1	$\theta_{-1,-1} = -0.7288$	$\hat{\theta}_{-1,-1} = -0.7219$
		$\rho = 1.111$	$\hat{\rho} = 1.103$
4,3	1,0	$\theta_{1,0} = 0.2800$	$\hat{\theta}_{1,0} = 0.3141$
	-1,0	$\theta_{-1,0} = 0.2800$	$\hat{\theta}_{-1,0} = 0.3141$
	1,-1	$\theta_{1,-1} = 0.2200$	$\hat{\theta}_{1,-1} = 0.2308$
	-1,1	$\theta_{-1,1} = 0.2200$	$\hat{\theta}_{-1,1} = 0.2308$
	0,1	$\theta_{0,1} = -0.1400$	$\hat{\theta}_{0,1} = -0.1242$
	0,-1	$\theta_{0,-1} = -0.1400$	$\hat{\theta}_{0,-1} = -0.1242$
		$\rho = 1.111$	$\hat{\rho} = 1.045$
4,4	0,1	$\theta_{0,1} = 0.5246$	$\hat{\theta}_{0,1} = 0.5241$
	0,-1	$\theta_{0,-1} = 0.5246$	$\hat{\theta}_{0,-1} = 0.5242$
	1,0	$\theta_{1,0} = 0.5357$	$\hat{\theta}_{1,0} = 0.5353$
	-1,0	$\theta_{-1,0} = 0.5357$	$\hat{\theta}_{-1,0} = 0.5353$
	1,1	$\theta_{1,1} = -0.3126$	$\hat{\theta}_{1,1} = -0.3120$
	-1,-1	$\theta_{-1,-1} = -0.3126$	$\hat{\theta}_{-1,-1} = -0.3120$
	1,-1	$\theta_{1,-1} = -0.2500$	$\hat{\theta}_{1,-1} = -0.2494$
	-1,1	$\theta_{-1,1} = -0.2500$	$\hat{\theta}_{-1,1} = -0.2494$
		$\rho = 1.111$	$\hat{\rho} = 1.102$

neighbor sets were chosen as candidate SAR models. Although infinite different neighbor sets are possible, it must be kept in mind that the statistical relationship between pixels far apart in an image is very low. It is therefore only necessary to compare the test statistics of neighbor sets using the closest neighbors to a pixel. Keeping this fact in mind, seven different neighbors sets were chosen as candidate SAR models using only the 4-neighbors (neighbor set 4), the 8-neighbors (neighbor sets 1, 2, 5 and 6), a causal neighbor set (neighbor set 3) as well as one larger neighbor set (set number 7).

The parameters of each neighbor set were estimated (using the approximate maximum likelihood technique in Section 5.5.1) for the image under consideration. These estimated parameters were used to calculate a test statistic for each neighbor set using equation (5.38). The SAR model with the lowest test statistic, C_k , is chosen as the SAR model that best describes the image under consideration. The results are given in Table 6.3.

As can be seen from Table 6.3, neighbor set number 1 has the lowest value for the test statistic. This indicates that the decision rule given in Section 5.5.2, which chooses the neighbor set with the lowest test statistic, is powerful enough to pick the correct neighbor set used to generate the synthetic image.

This is true even in the situation when the correct neighbor set is part of a larger possible neighbor set (neighbor set number 1 is included in both neighbor set number 6 and neighbor set number 7). It can also be seen from Table 6.3 that the closer a neighbor set is chosen to the correct neighbor set, the lower the value of the test statistic calculated for that neighbor set.

Table 6.3: Estimated parameters for image (4,1)

Number	Neighbor set used	Estimated Parameters	$\hat{\rho}$	Test Statistic
1	-1,0 1,0 -1,1 1,-1 -1,-1 1,1	$\hat{\theta}_{-1,0} = 0.1388$ $\hat{\theta}_{1,0} = 0.1388$ $\hat{\theta}_{-1,1} = 0.3168$ $\hat{\theta}_{1,-1} = 0.3168$ $\hat{\theta}_{-1,-1} = -0.1587$ $\hat{\theta}_{1,1} = -0.1587$	1.003	2088
2	0,-1 0,1 -1,1 1,-1 -1,-1 1,1	$\hat{\theta}_{0,-1} = 0.0398$ $\hat{\theta}_{0,1} = 0.0398$ $\hat{\theta}_{-1,1} = 0.3328$ $\hat{\theta}_{1,-1} = 0.3328$ $\hat{\theta}_{-1,-1} = -0.1637$ $\hat{\theta}_{1,1} = -0.1637$	1.131	2490
3	-1,0 0,-1 -1,-1	$\hat{\theta}_{-1,0} = 0.2910$ $\hat{\theta}_{0,-1} = 0.0472$ $\hat{\theta}_{-1,-1} = -0.5376$	2.601	3940
4	0,-1 0,1 -1,0 1,0	$\hat{\theta}_{0,-1} = 0.0513$ $\hat{\theta}_{0,1} = 0.0513$ $\hat{\theta}_{-1,0} = 0.1719$ $\hat{\theta}_{1,0} = 0.1719$	3.362	5280
5	-1,1 1,-1 -1,-1 1,1	$\hat{\theta}_{-1,1} = 0.3418$ $\hat{\theta}_{1,-1} = 0.3418$ $\hat{\theta}_{-1,-1} = -0.1567$ $\hat{\theta}_{1,1} = -0.1567$	1.135	2524
6	0,-1 0,1 -1,0 1,0 -1,1 1,-1 -1,-1 1,1	$\hat{\theta}_{0,-1} = 0.0124$ $\hat{\theta}_{0,1} = 0.0124$ $\hat{\theta}_{-1,0} = 0.1365$ $\hat{\theta}_{1,0} = 0.1365$ $\hat{\theta}_{-1,1} = 0.3143$ $\hat{\theta}_{1,-1} = 0.3143$ $\hat{\theta}_{-1,-1} = -0.1609$ $\hat{\theta}_{1,1} = -0.1609$	1.005	2114

Table 6.3 (continued)

Number	Neighbor set used	Estimated Parameters	$\hat{\rho}$	Test Statistic
7	-1,0	$\hat{\theta}_{-1,0} = 0.1420$	1.010	2163
	1,0	$\hat{\theta}_{1,0} = 0.1420$		
	-1,1	$\hat{\theta}_{-1,1} = 0.3125$		
	1,-1	$\hat{\theta}_{1,-1} = 0.3125$		
	-1,-1	$\hat{\theta}_{-1,-1} = -0.1515$		
	1,1	$\hat{\theta}_{1,1} = -0.1515$		
	-2,0	$\hat{\theta}_{-2,0} = -0.0160$		
	2,0	$\hat{\theta}_{2,0} = -0.0160$		

For example, neighbor set number 6 and neighbor set number 7 have much lower test statistics than neighbor set number 4, which has little in common with the original neighbor set used to generate the synthetic image.

6.4 Restoration of Noisy Image

The random field restoration technique was applied to restore an image contaminated with zero mean white Gaussian noise. The original image, noisy image and all seven restored images (using the seven possible neighbor sets given in the Table 6.3) are shown in the Figure 6.2. The signal-to-noise ratio for the noisy image was 0 dB.

The estimated parameters as well as the test statistic for each neighbor set were evaluated as discussed in Section 6.3. The noisy image was restored using each estimated SAR model. The next step was to calculate the MSE between the original image and the different restored images. The results are given in Table 6.4. The test

statistic for SAR model number 6 has the lowest value, with SAR model number 3 second. The MSE for SAR model number 3 was found to be the lowest but SAR model number 6 followed very closely. The apparent discrepancy can be contributed to round off errors in the numerical calculations as well as approximations made in the theory when the random field restoration technique was developed. Figure 6.4 also visually suggests that the restored images number 3 and number 6 are the best. The test statistics for models 4 and 5 were very similar. A visual examination of restored images 4 and 5 shows quite different results. This difference is also reflected in the MSE of models 4 and 5. Model 5 has an MSE of ten times more than model 4. This result indicates that one must be very cautious when using only the test statistic as a guideline for the restoration of an image.

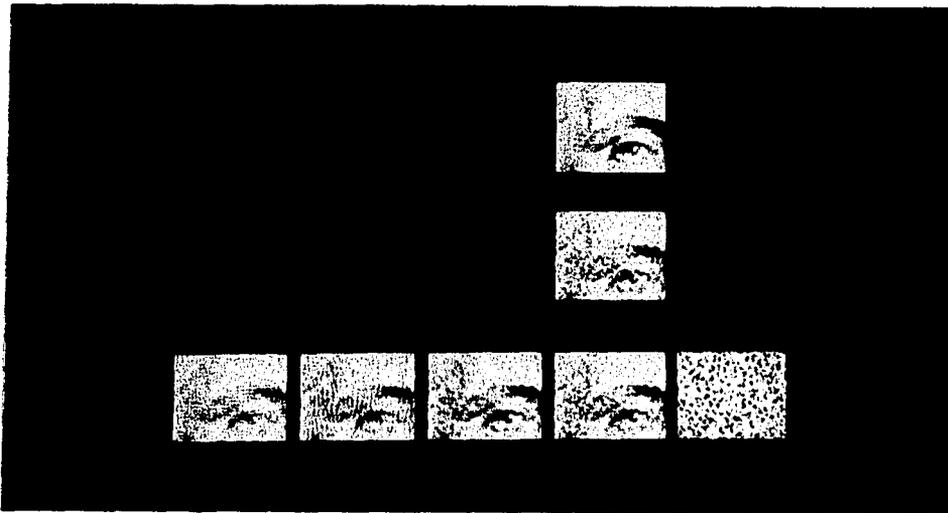


Figure 6.2: Restoration of Noisy Image

Table 6.4: Evaluated parameters and statistics for image in Figure 6.2

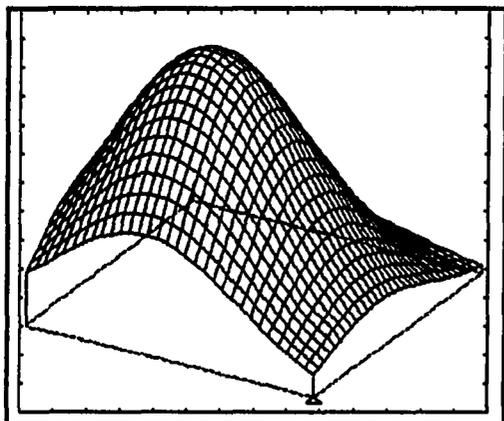
No.	Neighbor Set used	Estimated Parameters	$\hat{\rho}$	Test Statistic	MSE
1	-1,0 1,0 -1,1 1,-1 -1,-1 1,1	$\hat{\theta}_{-1,0} = 0.1207$ $\hat{\theta}_{1,0} = 0.1207$ $\hat{\theta}_{-1,1} = 0.1308$ $\hat{\theta}_{1,-1} = 0.1308$ $\hat{\theta}_{-1,-1} = 0.1039$ $\hat{\theta}_{1,1} = 0.1039$	820.89	27945	1174.9
2	0,-1 0,1 -1,1 1,-1 -1,-1 1,1	$\hat{\theta}_{0,-1} = 0.1421$ $\hat{\theta}_{0,1} = 0.1421$ $\hat{\theta}_{-1,1} = 0.1250$ $\hat{\theta}_{1,-1} = 0.1250$ $\hat{\theta}_{-1,-1} = 0.0998$ $\hat{\theta}_{1,1} = -0.0998$	792.17	27840	705.08
3	-1,0 0,-1 -1,-1	$\hat{\theta}_{-1,0} = 0.2233$ $\hat{\theta}_{0,-1} = 0.2974$ $\hat{\theta}_{-1,-1} = 0.1666$	889.41	27839	163.08
4	0,-1 0,1 -1,0 1,0	$\hat{\theta}_{0,-1} = 0.1688$ $\hat{\theta}_{0,1} = 0.1688$ $\hat{\theta}_{-1,0} = 0.1444$ $\hat{\theta}_{1,0} = 0.1444$	832.09	28034	257.81
5	-1,1 1,-1 -1,-1 1,1	$\hat{\theta}_{-1,1} = 0.1642$ $\hat{\theta}_{1,-1} = 0.1642$ $\hat{\theta}_{-1,-1} = 0.1376$ $\hat{\theta}_{1,1} = 0.1376$	871.64	28187	2403
6	0,-1 0,1 -1,0 1,0 -1,1 1,-1	$\hat{\theta}_{0,-1} = 0.1215$ $\hat{\theta}_{0,1} = 0.1215$ $\hat{\theta}_{-1,0} = 0.0929$ $\hat{\theta}_{1,0} = 0.0929$ $\hat{\theta}_{-1,1} = 0.1041$ $\hat{\theta}_{1,-1} = 0.1041$	769.95	27783	164.92

Table 6.4 (continued)

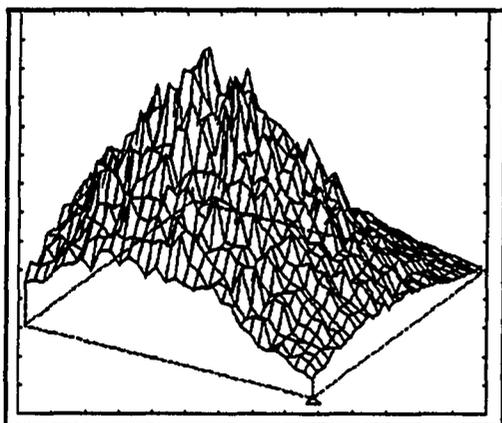
No.	Neighbor Set used	Estimated Parameters	$\hat{\rho}$	Test Statistic	MSE
	-1,-1 1,1	$\hat{\theta}_{-1,-1} = 0.0786$ $\hat{\theta}_{1,1} = 0.0786$			
7	-1,0 1,0 -1,1 1,-1 -1,-1 1,1 -2,0 2,0	$\hat{\theta}_{-1,0} = 0.1074$ $\hat{\theta}_{1,0} = 0.1074$ $\hat{\theta}_{-1,1} = 0.1195$ $\hat{\theta}_{1,-1} = 0.1195$ $\hat{\theta}_{-1,-1} = 0.0901$ $\hat{\theta}_{1,1} = 0.0901$ $\hat{\theta}_{-2,0} = 0.0642$ $\hat{\theta}_{2,0} = 0.0642$	812.62	27923	2036.6

6.5 Restoration of Noisy Simulated Flux Density Maps

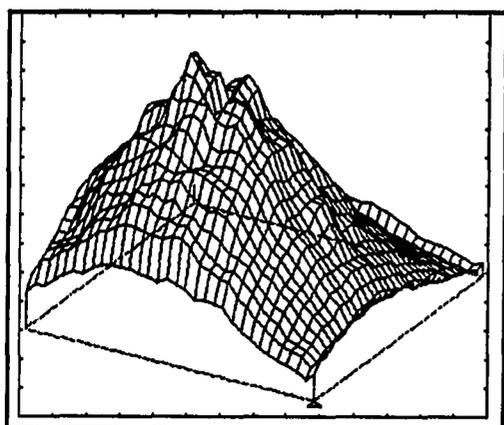
To determine the effectiveness of the random field restoration procedure with respect to the estimation of parameters using noisy image data, this procedure was used to restore the same three noisy simulated flux density maps discussed in detail in Section 3.1. The same seven possible neighbor sets as in the previous section were used as candidate SAR models for each flux density map. The parameters for each neighbor set were estimated using the approximate maximum likelihood estimation procedure as before, and the best set was chosen by calculating the test statistic for each neighbor set as already discussed.



Original image
Maximum value = 0.3210 Jy/beam
Ground level value = 0.0 Jy/beam

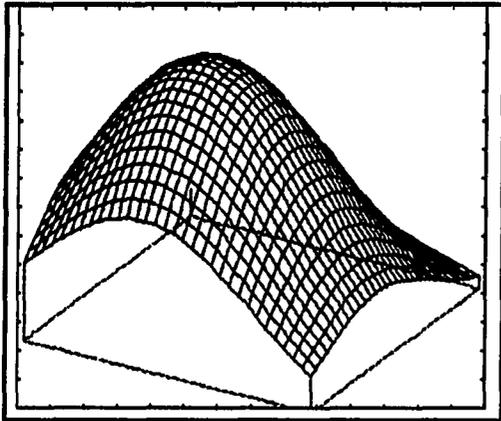


Noisy image
Maximum value = 0.3950 Jy/beam
Ground level value = 0.0 Jy/beam

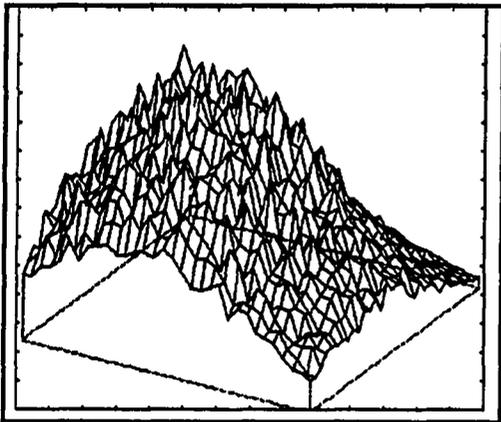


Restored image
Maximum value = 0.3660 Jy/beam
Ground level value = 0.0 Jy/beam

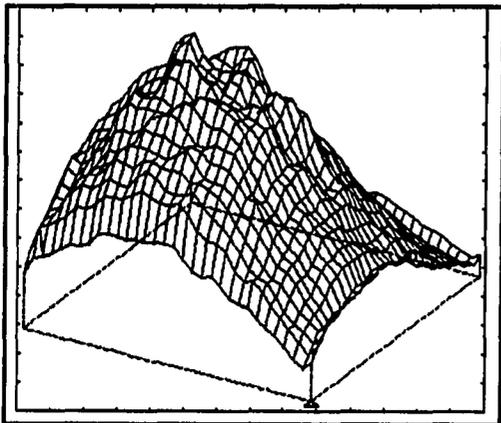
Figure 6.3: 2 cm flux density maps



Original image
Maximum value = 0.1920 Jy/beam
Ground level value = 0.0 Jy/beam

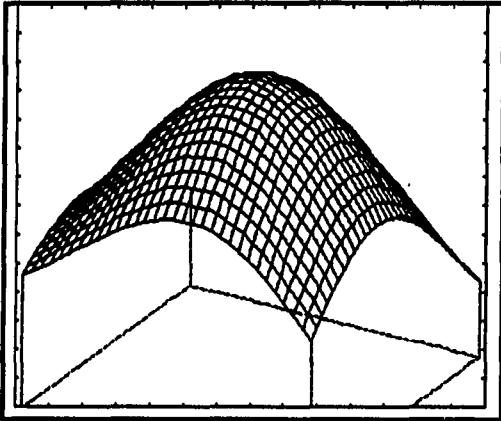


Noisy image
Maximum value = 0.2290 Jy/beam
Ground level value = 0.0 Jy/beam

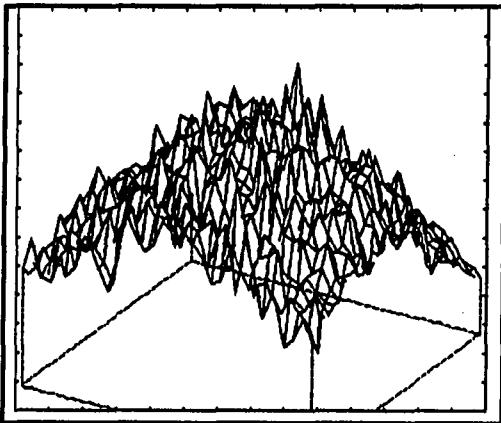


Restored image
Maximum value = 0.2090 Jy/beam
Ground level value = 0.0 Jy/beam

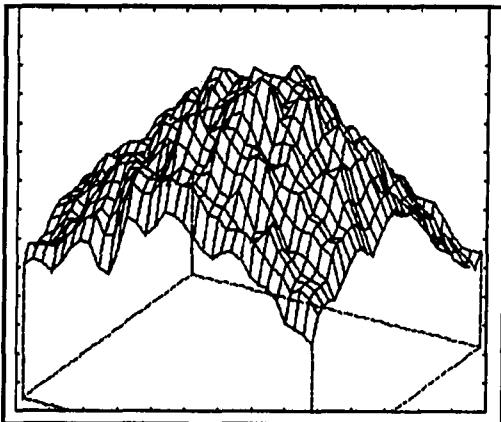
Figure 6.4: 6 cm flux density maps



Original image
Maximum value = 0.0149 Jy/beam
Ground level value = 0.0 Jy/beam



Noisy image
Maximum value = 0.0183 Jy/beam
Ground level value = 0.0 Jy/beam



Restored image
Maximum value = 0.0160 Jy/beam
Ground level value = 0.0 Jy/beam

Figure 6.5: 20 cm flux density maps

Each flux density map was restored using the SAR model with the lowest test statistic for that map. Figures 6.3, 6.4 and 6.5 show the results for each flux density map obtained by using the random field restoration.

An important observation from these three figures is that in all three cases the restored images are much smoother than the noisy images. The only areas that do not appear to be very smooth are the peak areas. This is true for all three restored images. There exists a simple explanation for this phenomena. From Section 3.1 we know that zero mean white Gaussian noise with a standard deviation of 10% of a specific pixel value was added to all pixels. Because the peak areas have larger values than the rest of the image, more noise was added to the pixels in these areas. This is a reason why the random field restoration procedure did a better smoothing job for the pixels not in the peak areas, compared to the pixels in the peak areas.

The restored images in Figures 6.3, 6.4 and 6.5 were then used to estimate the parameters in the flux density equations (2.15), (2.16) and (2.17). This was done in exactly the same way as discussed in Section 2.3. Only in this case the random field restored images were used.

The results for row 16 for all three estimated parameters are shown in Figures 6.6, 6.7 and 6.8. These curves show a comparison of the estimated parameters, for a specific cut through the estimated parameters maps, for different situations. The different curves are labelled on the figures. The curve labelled "0% noise" shows the estimated parameters when 0% noise was added to the simulated flux density maps (resulting in the original temperature and optical depth parameters used in the simulation as discussed in Chapter 3) and is used as a reference condition.

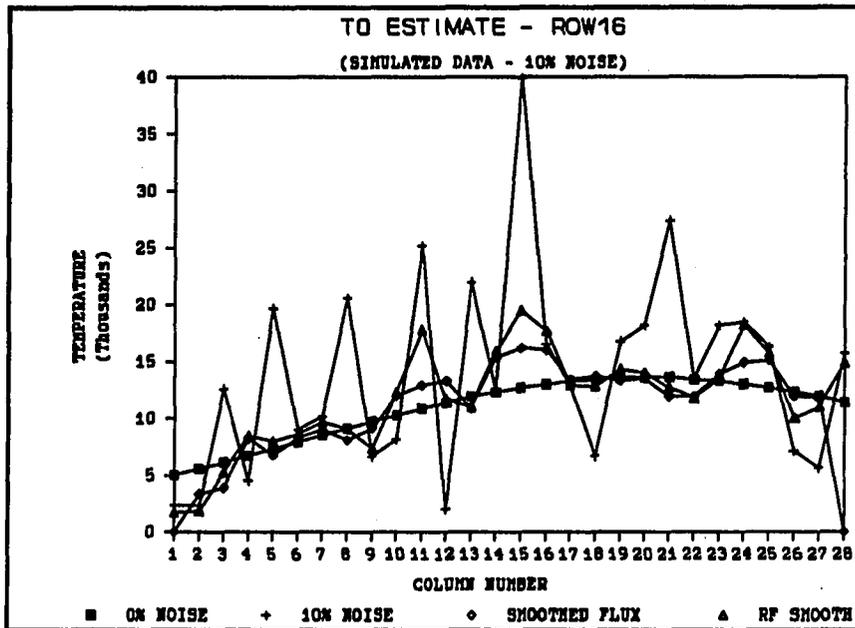


Figure 6.6: Cut through row 16 of T_0 maps showing the effect of random field restoration

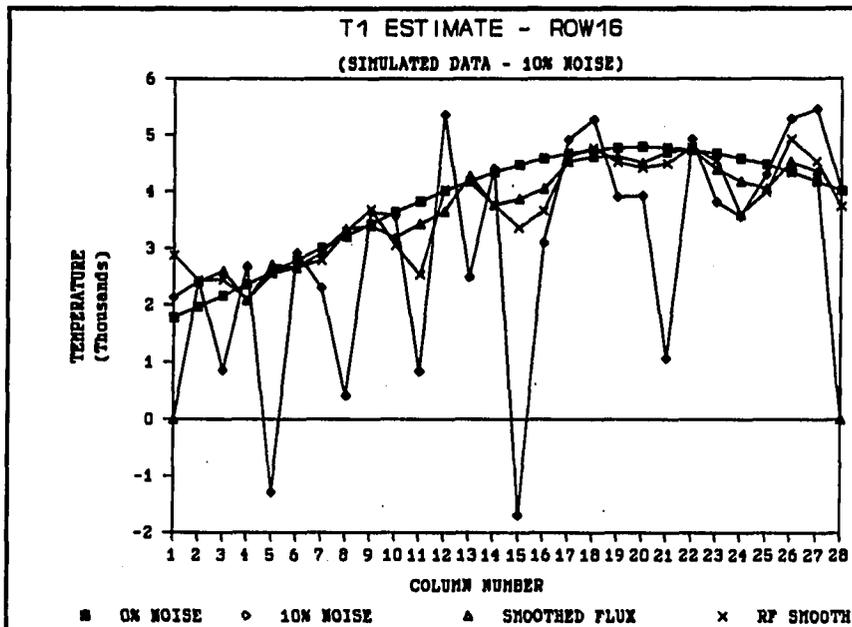


Figure 6.7: Cut through row 16 of T_1 maps showing the effect of random field restoration

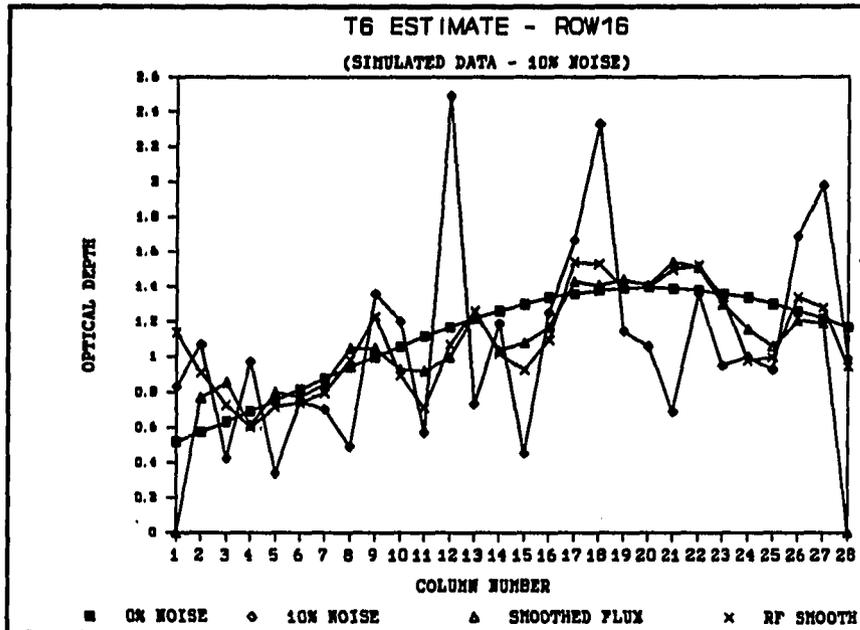


Figure 6.8: Cut through row 16 of τ_0 maps showing the effect of random field restoration

The curve labeled "10% noise" shows the estimated parameters when 10% zero mean white Gaussian noise was added to the simulated flux density maps. The curve labeled "Smoothed Flux" shows the estimated parameters when 10% zero mean white Gaussian noise was added to the simulated flux density maps and the noisy flux density maps were smoothed using a 3×3 running window before estimating the parameters. All these cases were discussed thoroughly in Chapter 3. The fourth curve, labeled "RF Smooth" shows the estimated parameters when again 10% zero mean white Gaussian noise was added to the simulated flux density maps, but in this case the random field restoration procedure was applied before estimating the

parameters.

The cut through row 16 of the estimated parameter maps was typical of similar cuts through other rows. A statistical analysis reveals that the MSE between the estimated parameters and their original values using the random field restoration are a small amount, less than 20%, larger than the MSE between the estimated parameters and the original parameters when the noisy flux density values were first smoothed as already discussed. This is visually shown in Figures 6.6, 6.7 and 6.8 where the estimated parameter curves using the smoothing technique first, is almost at every pixel closer to the original curve than the estimated parameters using the random field restoration. These results are shown on a larger scale for row 2 and row 16 for all three flux density maps in Figures 6.9, 6.10, 6.11, 6.12, 6.13 and 6.14.

A question that comes to mind is why the results of the double smoothing technique were not compared to the results of the random field restoration, but instead the results of the smoothed flux images were compared with the results obtained using the random field restoration. The reason for this is as follows: Using the random field restoration, only the noisy flux density maps were smoothed and nothing was done to the estimated parameter maps. To compare the results of the random field restoration with the results of the double smoothing technique, it would be necessary to smooth the estimated parameter maps obtained by using the random field restoration technique also. Because the estimated parameters using the smoothed flux maps were closer to the original values than the estimated parameter maps using the random field technique, this was not necessary.

In this chapter it was shown that, for the specific simulated temperature and

optical depth maps, adding zero mean white Gaussian noise to the resulting flux density maps, the double smoothing technique performs better in estimating parameters from the images than the random field restoration technique. Much more investigation is needed to make a general statement of comparison between the two techniques when white Gaussian noise is added to other digitized images. Even then, the area of colored noise has not been touched.

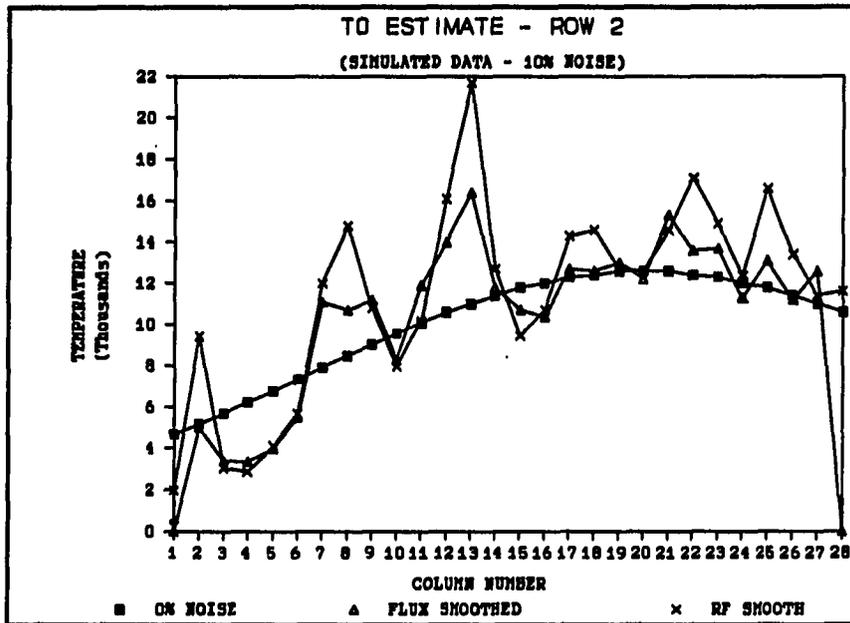


Figure 6.9: Cut through row 2 of T_0 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models

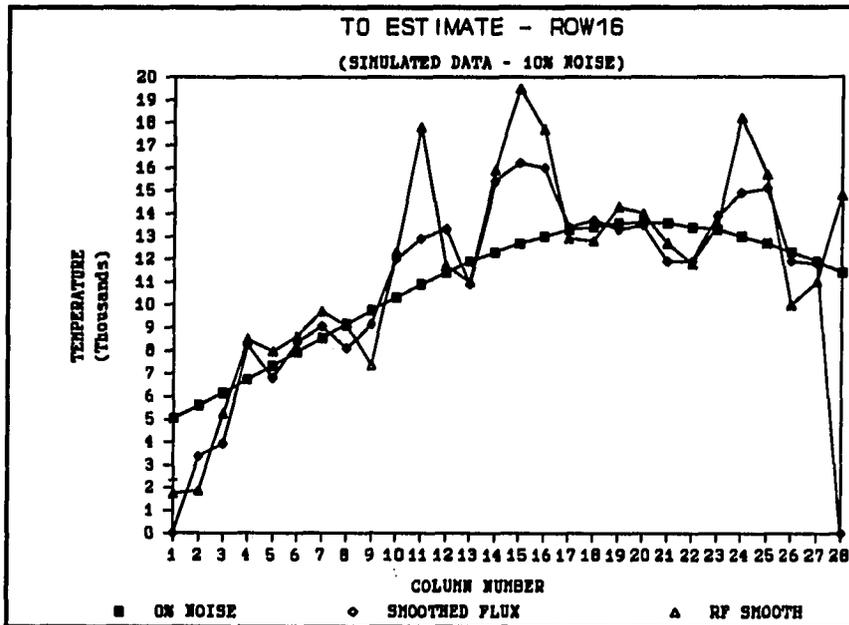


Figure 6.10: Cut through row 16 of T_0 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models

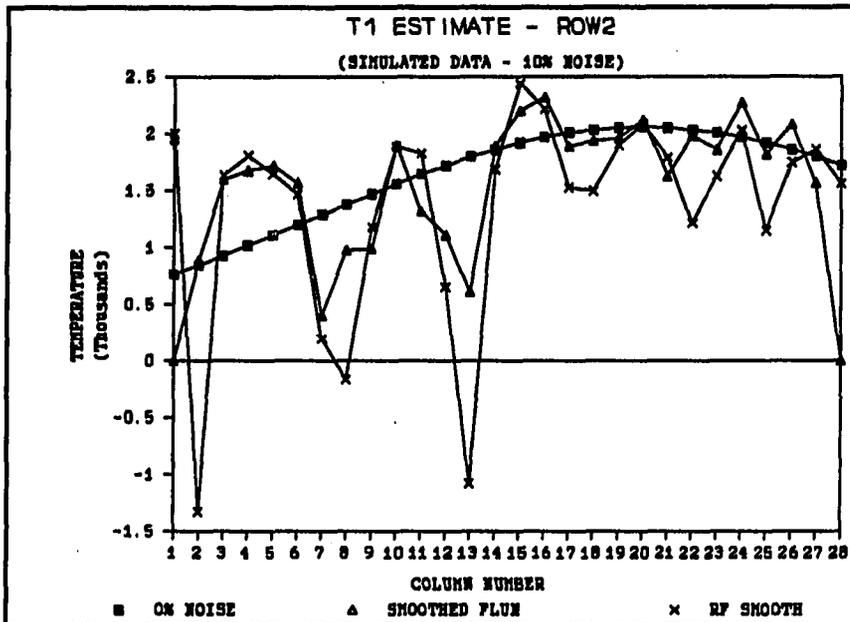


Figure 6.11: Cut through row 2 of T_1 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models

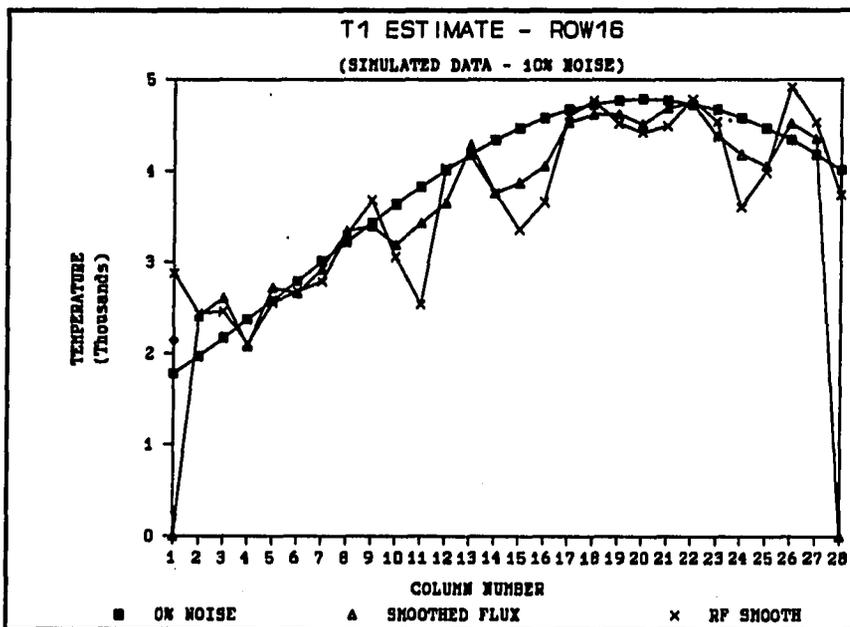


Figure 6.12: Cut through row 16 of T_1 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models

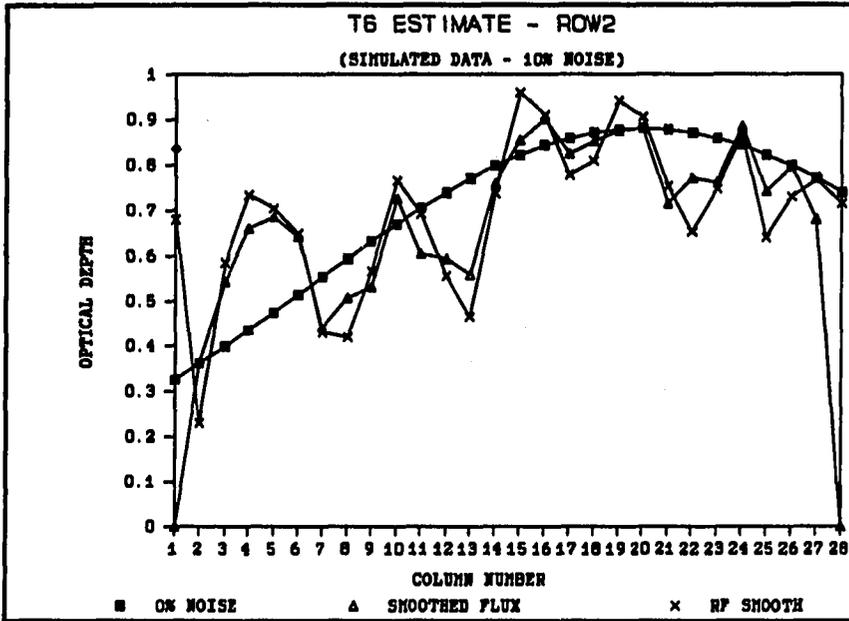


Figure 6.13: Cut through row 2 of τ_6 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models

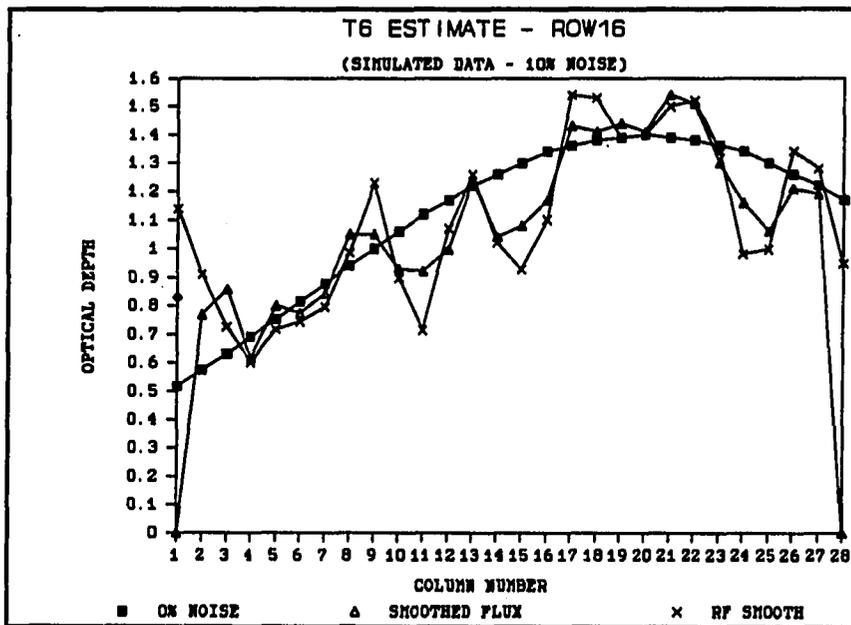


Figure 6.14: Cut through row 16 of τ_6 maps comparing the effect of smoothing the flux densities and restoring the flux densities using random field models

7. AN APPROXIMATE SOLUTION FOR NGC 7027 TEMPERATURE AND OPTICAL DEPTH PARAMETERS

7.1 Introduction

It is appropriate to review what has been done so far. Using the observed flux density values, it was found that a direct solution of the derived flux density equations, using temperature model 2, results in temperature and optical depth parameters that are not physically realizable (Section 2.4). An attempt to obtain temperature and optical depth parameters that are physically acceptable by varying the observed flux density values by $\pm 10\%$ (Section 2.5), was also unsuccessful, because it was not possible to choose the correct solution from all the possible solutions.

The double smoothing technique, applied to the observed flux density values, results in estimated parameter maps that correspond, in structure at least, quite well with previously obtained maps for the same parameters (Section 3.4). However, as indicated by simulation studies (Chapter 4), the actual numerical values of the estimated parameters using this technique, can be wrong by large margins.

In all of the previous methods the approach was to solve the derived flux density equations exactly. The only approximations made were the smoothing of the flux density values and the smoothing of the estimated parameters in the double smoothing

technique. Even in this technique, using the smoothed flux density values, the derived flux density equations were solved exactly.

An alternative approach would be to solve for the unknown parameters approximately. In this solution approximations would be made based on the physical characteristics of the nebula as well as knowledge of the estimated parameters obtained by previous studies. A discussion of such an approach follows.

7.2 Approximate Solution for NGC 7027 Parameters

7.2.1 Determination of Emission Measure Map

As a first approximation, use the physical characteristic that the nebula is optically thin at 2 cm. From a practical viewpoint, this means that at 2 cm wavelength, it is possible to see through the nebula. If, for example it was possible to add more gas at the back of the nebula, this would be reflected in the 2 cm observed flux density map. As a result, the 2 cm observed flux density map is a good approximation of the emission measure of the nebula. This is verified by using the same approximation in the derived mathematical equation for observed flux density at 2 cm. The optically thin characteristic at 2 cm is mathematically given by

$$a_{02}\tau_6 \ll 1 \tag{7.1}$$

and a Taylor series expansion can be used for $e^{-a_{02}\tau_6}$ to simplify the flux density equation at 2 cm (see equation (2.15)) as follows

$$\begin{aligned}
S_{02} &= K_{102} \left\{ T_0 (1 - e^{-a_{02}\tau_6}) - \frac{(T_0 - T_1)}{a_{02}\tau_6} (a_{02}\tau_6 - 1 + e^{-a_{02}\tau_6}) \right\} \\
&\simeq K_{102} \frac{a_{02}\tau_6}{2} [T_0 + T_1] \quad (7.2)
\end{aligned}$$

where all the symbols have the same meaning as discussed in Chapter 2. As a second approximation, assume that the observed flux density at 2 cm is not very sensitive to the temperature variation in the nebula. Physically this can be explained from the fact that it is possible to see through the nebula at 2 cm. As a result, assume that the temperature at the back of the nebula, T_0 , is the same as the temperature at the front of the nebula, T_1 . The resulting flux density equation at 2 cm is then given by

$$S_{02} = K_{102} a_{02} \tau_6 T_1. \quad (7.3)$$

Osterbrock (1974) has shown that, assuming T to be constant with optical depth, the relationship between optical depth and emission measure, E , is given by

$$\tau_\nu = \frac{0.0824 E}{\nu_{\text{GHz}}^{2.1} T^{1.35}}. \quad (7.4)$$

Substituting equation (7.4) in equation (7.3) gives

$$\begin{aligned}
S_{02} &= \frac{K_{102} a_{02} 0.0824 E T_1}{\nu_{\text{GHz}}^{2.1} T_1^{1.35}} \\
&= \frac{K_{102} a_{02} 0.0824 E}{(4.8851)^{2.1} T_1^{0.35}}. \quad (7.5)
\end{aligned}$$

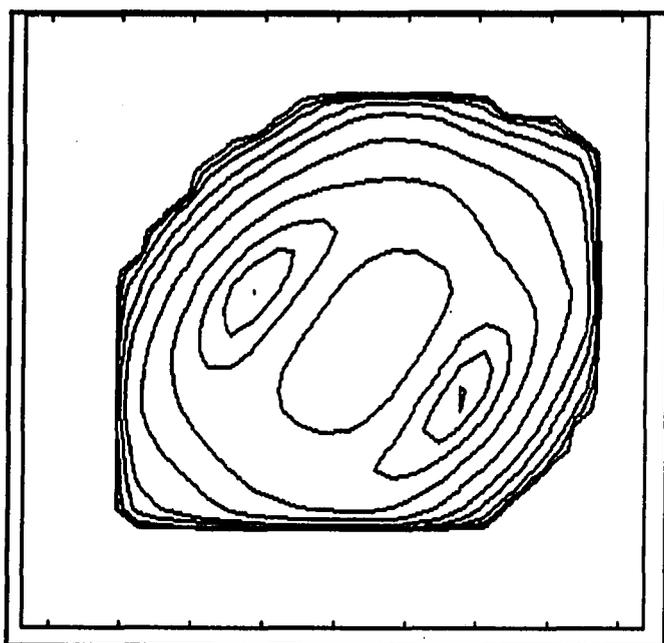
This equation supports the assumption that the observed flux density at 2 cm is not very sensitive to changes in temperature through the nebula. A change in T_1 from

10 000 K to 20 000 K results in only a 27% change in the value of S_{02} . By choosing a typical value for T_1 obtained from the 2-D analysis by Basart and Daub (1987) of 13 000 K, the emission measure of the nebula can be obtained from the 2 cm flux density map. This map is shown in Figure 7.1.

Using the approximations above, this map is only a scaled version of the 2 cm observed flux density map. Figure 7.1 has the same double peak structure as the 2 cm flux density map. This emission measure map corresponds almost 100% with the emission measure map obtained in the 2-D analysis of the nebula by Basart and Daub (1987) shown in Figure 7.2. They both have the same structure and the peak values are almost identical, 8.0 pc cm^{-6} obtained by Basart and Daub (1987) as compared to 8.35 pc cm^{-6} obtained by the approximate analysis. Because the observed flux density values are very low on the edges of the nebula, the signal to noise ratios for these values are also very low. Most of these low flux density values on the edges were ignored in the approximate solution and this give rise to the unnatural looking edges in the estimated parameter maps.

7.2.2 Determination of Temperature T_1 -map

To obtain a map of the nebula temperature at the front surface, the physical characteristic that the nebula is optically thick at 20 cm is used. From a practical viewpoint, this means that at 20 cm wavelength it is not possible to see through the nebula, all that can be seen is the front surface.



Peak value = 8.35 E07
 Contour levels at 99%
 90%, 75%, 50%, 25%, 10%,
 5%, 2%, 1%, 0.5%

Figure 7.1: Emission measure contour map using approximate method and observed data

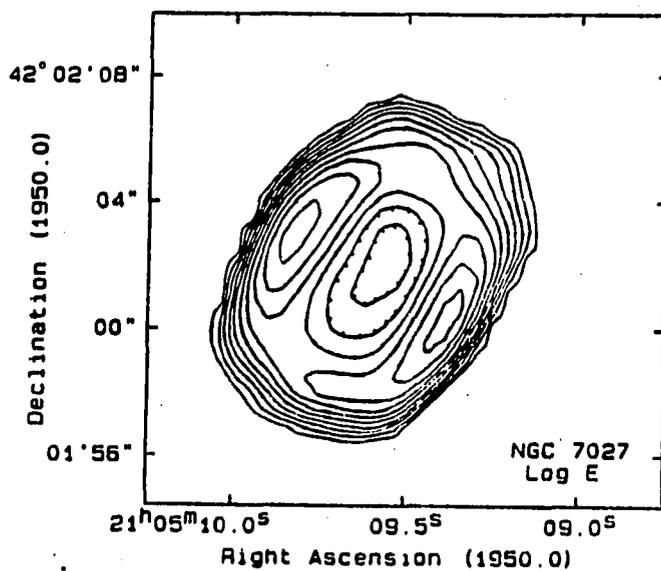


Figure 7.2: Emission measure contour map obtained by Basart and Daub (1987)

Mathematically, the optically thick characteristic at 20 cm is given by

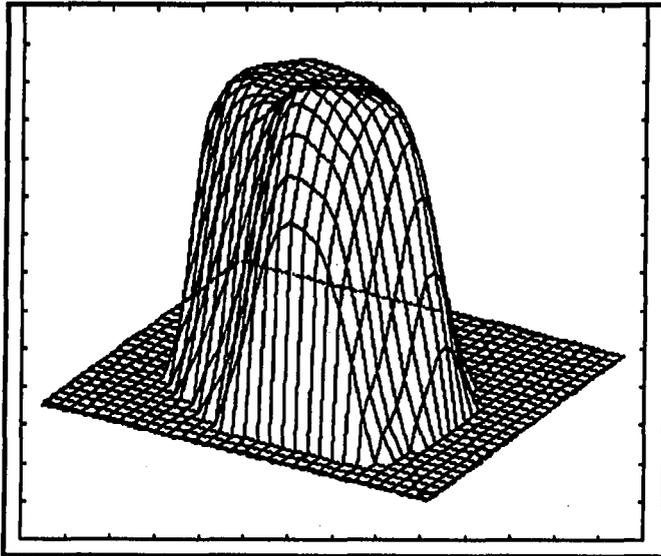
$$a_{20}\tau_6 \gg 1. \quad (7.6)$$

Still assuming that the temperature at the back of the nebula is the same as the temperature at the front of the nebula, $T_0 = T_1$, the flux density equation at 20 cm (see equation (2.17)) can now be approximated as

$$\begin{aligned} S_{20} &= K_{120} \left\{ T_0 (1 - e^{-a_{20}\tau_6}) - \frac{(T_0 - T_1)}{a_{20}\tau_6} (a_{20}\tau_6 - 1 + e^{-a_{20}\tau_6}) \right\} \\ &\simeq K_{120} T_1 (1 - e^{-a_{20}\tau_6}). \end{aligned} \quad (7.7)$$

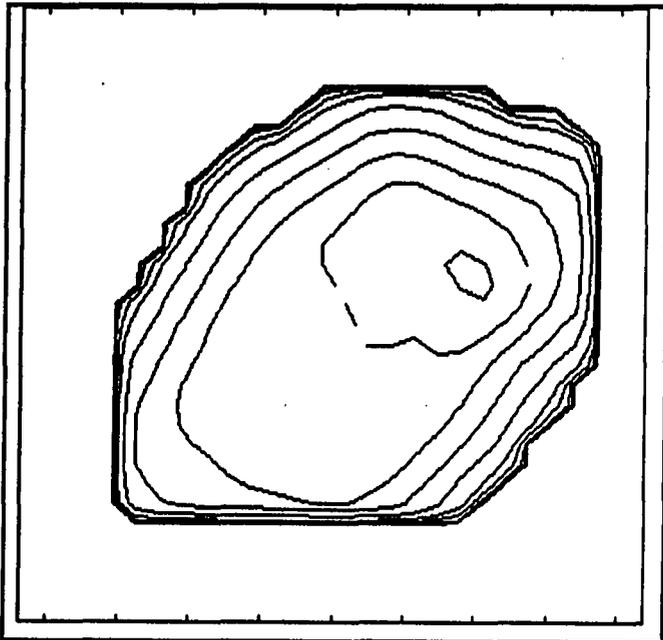
The expression for τ_6 given by equation (7.4) can be used in equation (7.7) to solve for T_1 . The emission measure, E , solved for above, was also used and the resulting map for temperature T_1 is shown in Figures 7.3 and 7.4. This map for the temperature at the front surface of the nebula, T_1 , has the same structure as the map obtained for temperature distribution by Basart and Daub (1987) which is shown in Figure 3.1. (Remember Basart and Daub (1987) assumed that the temperature is constant along the line of sight.) As a matter of fact, the map obtained for T_1 using the double smoothing technique, shown in Figures 3.32 and 3.33, has an identical structure. All three maps show a maximum temperature region in the north east corner.

The only difference between the three maps is the values of the peak temperature, 14 000 K obtained by the approximate technique, 15 000 K obtained by the double smoothing technique and by Basart and Daub (1987).



Maximum value = 14000 K
Ground level value = 0 K

Figure 7.3: T_1 estimated parameter 3-D map using approximate method and observed data



Peak value = 14000 K
Contour levels at 99%, 90%,
75%, 50%, 25%, 10%, 5%,
2%, 1%, 0.5%

Figure 7.4: T_1 estimated parameter contour map using approximate method and observed data

This small difference is almost negligible. Also, the value of $T_1 \simeq 13\,000$ K over a large portion of the map, supports the value used for T_1 used in the evaluation of the emission measure map, E , previously.

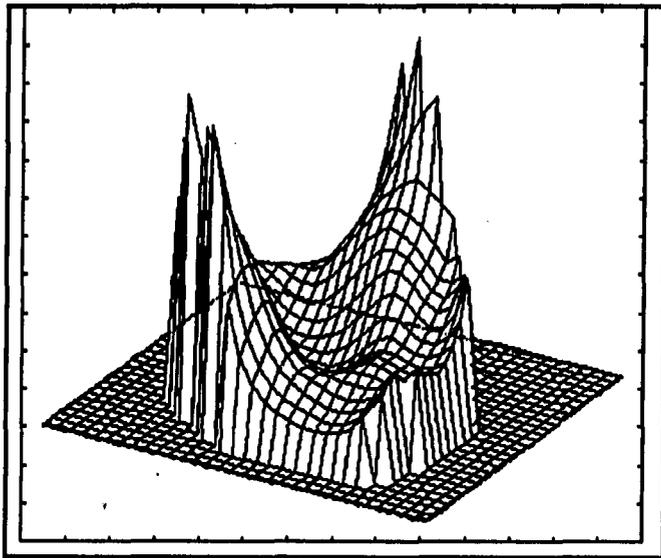
7.2.3 Determination of τ_6 -map

All the information necessary to obtain a map of the optical depth at 6 cm, τ_6 , is available. The equation used to find the τ_6 values is equation (7.4), evaluated at 6 cm, and is given by

$$\tau_6 = \frac{0.0824 E}{(4.8551)^{2.1} T_1^{1.35}} \quad (7.8)$$

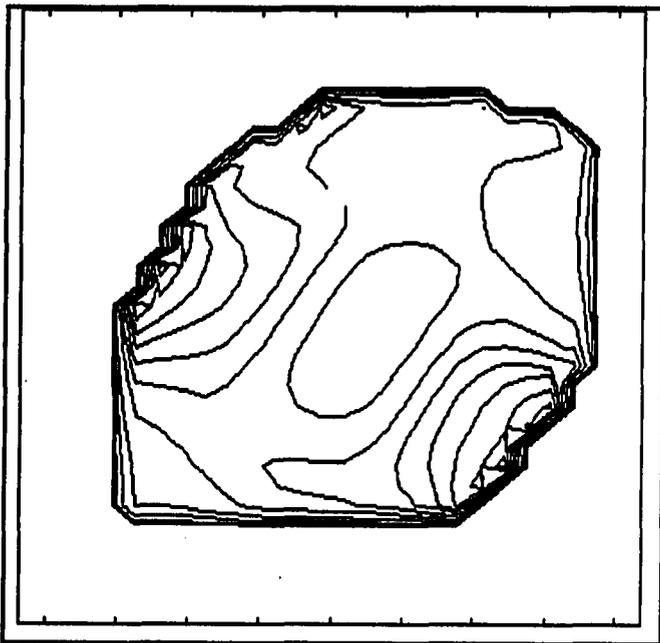
The values used for E and T_1 in this evaluation, were the values previously solved for. The map for τ_6 is shown in Figures 7.5 and 7.6. This map for τ_6 , in both structure and maximum value, compares better with the map for the same parameter obtained by Basart and Daub (1987), shown in Figure 3.2, than the τ_6 -map obtained by using the double smoothing technique, shown in Figures 3.34 and 3.35. This statement can be made because the map for τ_6 obtained using the approximate method is more symmetric than the τ_6 -map obtained using the double smoothing technique.

Also the maximum value for τ_6 obtained using the double smoothing technique, 2.39, is a bit higher than what is physically expected. The maximum value of 1.69 for τ_6 , obtained using the approximate method, is closer to the physically expected value of $\simeq 1.4$. However, even in this case, all three maps have the same *basic* structure. An investigation of the three maps show that they all three have the same double peaks in the northwest and southeast regions with a region of lower optical depth in the



Maximum value = 1.69
Ground level value = 0.0

Figure 7.5: τ_0 estimated parameter 3-D map using approximate method and observed data



Peak value = 1.69
Contour levels at 99%, 90%,
75%, 60%, 50%, 40%, 30%,
20%, 10%, 5%, 2%, 1%,
0.5%

Figure 7.6: τ_0 estimated parameter contour map using approximate method and observed data

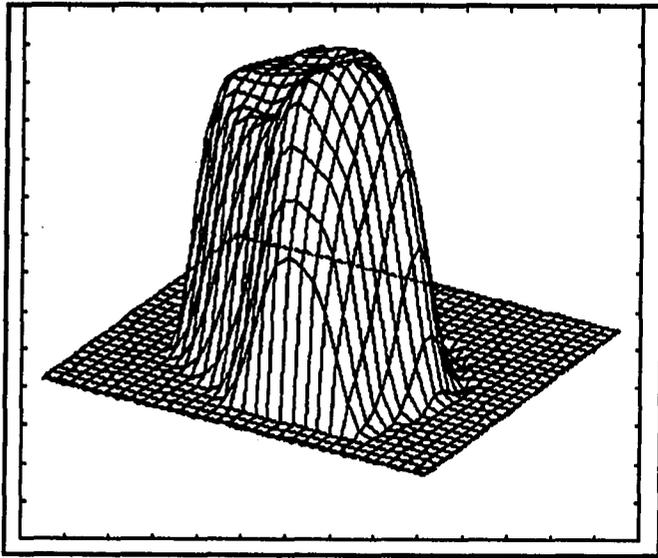
center of the maps.

7.2.4 Determination of Temperature T_0 -map

The temperature at the back of the nebula, T_0 , can be obtained from the flux density equation at 6 cm, using the values for τ_6 and T_1 previously solved for. The 6 cm flux density equation is repeated here for clarity

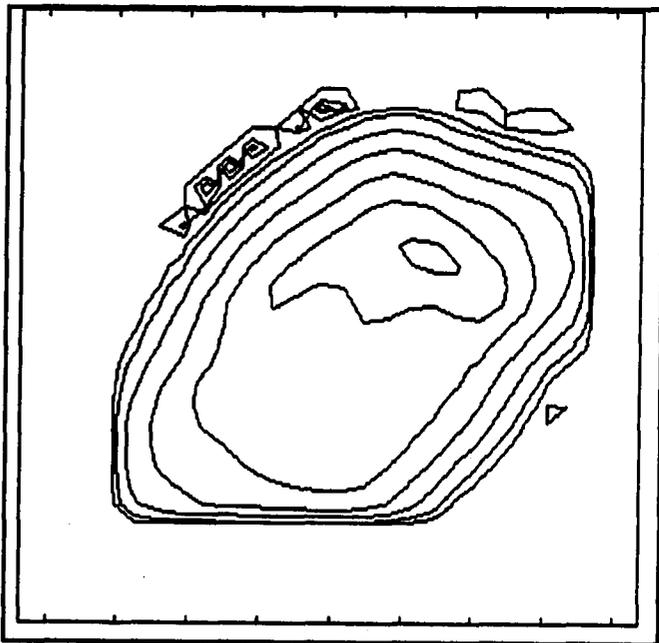
$$S_{06} = K_{106} \left\{ T_0 (1 - e^{-a_{06}\tau_6}) - \frac{(T_0 - T_1)}{a_{06}\tau_6} (a_{06}\tau_6 - 1 + e^{-a_{06}\tau_6}) \right\} \quad (7.9)$$

The resulting T_0 -map is shown in Figures 7.7 and 7.8. This map for the temperature at the back of the nebula, differs significantly from the map obtained by the double smoothing technique for the same parameter, shown in Figures 3.30 and 3.31. The map for T_0 obtained using the approximate technique, has nearly the same structure as the map for T_1 obtained by using the same technique, shown in Figures 7.3 and 7.4. The only significant difference is between the maximum values of the two individual maps, T_0 has a maximum value of 16 000 K and T_1 has a maximum value of 14 000 K. There is obviously no map from Basart and Daub (1987) to compare these maps with. Physical information to choose one of the maps instead of the other as the "true" temperature map at the back of the nebula does not exist.



Maximum value = 16000 K
Ground level value = 0 K

Figure 7.7: T_0 estimated parameter 3-D map using approximate method and observed data



Peak value = 16000 K
Contour levels at 99%, 90%,
75%, 50%, 25%, 10%, 5%,
2%, 1%, 0.5%

Figure 7.8: T_0 estimated parameter contour map using approximate method and observed data

However, a discussion with an astronomer, Dr. C. T. Daub (San Diego State University, private communication) led to a conclusion that the T_0 -map obtained by the double smoothing technique seems more logical, due to its symmetric structure. This structure is almost the same as the double peak structure in the 2 cm observed flux density map. Remember that the nebula is optically thin at 2 cm wavelength and that the 2 cm observed flux density map gives an indication of the back surface of the nebula. The T_0 -map obtained by the approximate technique, lacks this symmetry.

7.3 Conclusion

As a summary, the following conclusions can be drawn from the present analysis:

1. The temperature maps made by three different techniques for the front surface temperature of the nebula, T_1 , are almost identical and suggests that they represent the "true" temperature at the front surface of the nebula.
2. The optical depth maps have a similar structure, but the one obtained with the approximate method is more believable due to the maximum value of 1.69 for optical depth which corresponds with previous studies.
3. As already discussed, the T_0 -map obtained by the double smoothing technique seems to be more acceptable.

To find more accurate estimates of these unknown parameters, more observations of the nebula, at different wavelengths, are necessary. The extra data can then be used to smooth the noise in the data values, as already discussed.

7.4 Temperature Model 3

Before stepping off the subject of estimating the nebula NGC 7207 temperature and optical depth parameters, the question must be answered: "What happens when the temperature model 3 (Figure 2.1) is used to derive flux density equations?" The answer is real short. For many attempted calculations no physical allowable parameters could be obtained using this model. This fact further supports the results obtained so far that temperature model 2 is a valid temperature model for the nebula.

8. AUTOMATIC MEASUREMENT OF RAINDROP DIAMETERS

8.1 Problem Statement

In the previous chapters multiple digitized images were used to estimate the parameters in an equation describing some characteristic in the images. To extend this work, another parameter estimation problem using digitized image data were also investigated. In this case only one digitized image was used to find the unknown parameters.

The problem is as follows: Given an image of raindrops (lighter spots) on a darker background (see Figure 8.1), determine the number of raindrops and their individual sizes from this image automatically by using image processing techniques. It is important to note that some of the raindrops may overlap. The approach used must be able to detect these overlapping raindrops and give the actual sizes of the individual raindrops.

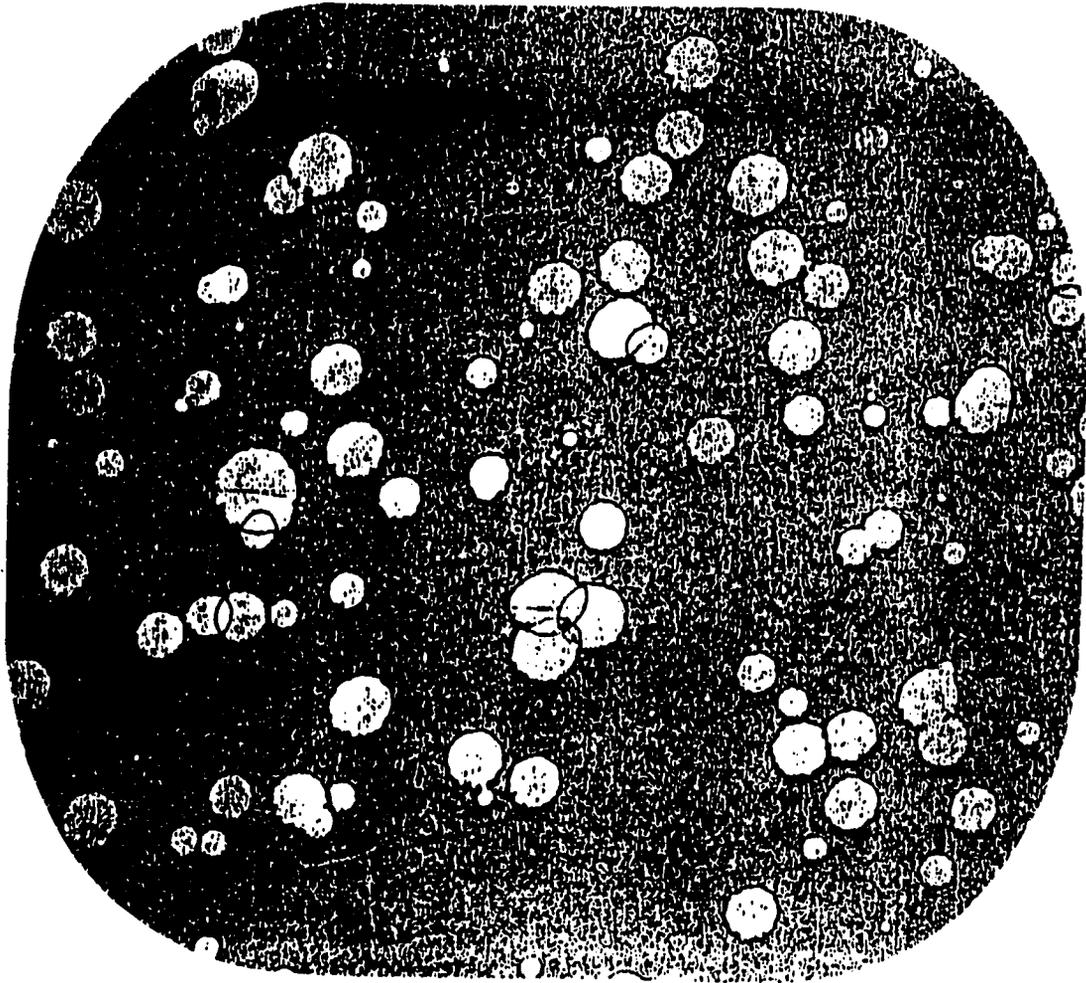


Figure 8.1: Raindrop image

8.1.1 Approach

This problem can be broken down into two parts:

1. The image processing system must be able to distinguish between the raindrops and the background.
2. The image processing system must be able to find the individual raindrops in the image after the background has been removed, and determine their actual sizes. As already mentioned, this includes that the image processing system must be able to detect overlapping raindrops and determine the sizes of the individual raindrops in a larger overlapping raindrop.

8.2 Distinguish between the Raindrops and the Background

8.2.1 Thresholding

As can be seen from Figure 8.1, the raindrops and the background have different gray levels. Therefore, a bimodal image gray-level histogram is expected so the most obvious approach to distinguish between the raindrops and the background would be to use thresholding.

In practice it is found that this approach does not work. There are a number of reasons:

1. When looking at the original image, it can be seen that there are dark pixels inside the lighter raindrops as well as light pixels in the darker background.

When using a thresholding technique, the dark pixels inside the raindrops will

remain dark and the light pixels in the background will still be light. This will result in distorted raindrop areas and false raindrops respectively.

2. It is also clear from Figure 8.1 that the background gray level is not uniform. This is due to the process by which the background coloring was sprayed onto the original paper. As a result it is possible that the average gray level in a portion of the background can be lighter than the average gray level inside a raindrop at another position. This contributes to the fact that the image histogram is not bimodal and a threshold cannot be found as discussed to separate the raindrops from the background.

As an alternative, several different thresholds were chosen by trial and error, all giving the same general result. When the threshold is chosen to include all the raindrop information in the image, some background information is also included. When choosing a threshold to get rid of all the background, some raindrop information is also lost. In both cases, the actual raindrop areas in the original image are distorted. Therefore, this technique cannot be applied if it is necessary to find accurate estimates of the raindrop positions and sizes in the original image.

8.2.2 Smoothing

In an attempt to improve the thresholding technique, the original image was first smoothed using a 3×3 running window with equal weight assigned to all 9 cells in the window. The idea behind this approach was to get rid of the light pixels in the background and the dark pixels inside the raindrops. However, it was found that the image gray-level histogram was still not bimodal. Thresholding using different chosen

threshold values resulted in the same general problems as for the original unsmoothed image.

8.2.3 Morphology

To distinguish between the raindrops and the background, it would be necessary to use a technique that depends on local properties of an image instead of global properties, like the thresholding technique. Such a technique is provided by applying mathematical morphology to the raindrop image. The specific operation that was used from the field of mathematical morphology was the operation of opening of a gray scale image. Gray scale opening of an image is defined as the dilation of the eroded image using the same structuring element. For a definition of these basic terms in mathematical morphology, consult, for example, Serra (1982).

The opening of the raindrop image was used to estimate the background in the image. The background is defined as those parts of the image that do not include any raindrops. In this specific case, the raindrop image was opened using a hemispherical structuring element which was bigger than the largest raindrop in the image. The opened image, which was an estimate of the background, was then subtracted from the original image. The resulting image was the image containing only the raindrop information.

By applying this technique, it was possible to get rid of all the background, but still keeping all the raindrop information almost undisturbed.

8.3 Distinguish between Individual Raindrops and Find their Actual Sizes

8.3.1 The modified Hough Transform

An investigation of the images to be analyzed leads to the following conclusion: The individual raindrop areas can be treated as circular without much loss in accuracy. To determine exactly how much, is a function of the specific image used. For the image shown in Figure 8.1, the loss in accuracy is less than 5% in area. This was determined by comparing the areas of possible circles to be used with the area of the raindrop under consideration. By using this approximation, the first method used for finding the individual raindrops and their sizes was the modified Hough Transform.

8.3.1.1 Method Description A detailed description of the Hough Transform can be found in Gonzalez and Wintz (1987, Section 3.7), as well as a modified version to detect circles in the $x - y$ plane. Only a summary of the last method will be given here.

Consider the case of only one circle in the $x - y$ plane. The general equation for this circle is given by

$$(x - c_1)^2 + (y - c_2)^2 = c_3^2 \quad (8.1)$$

where c_1 and c_2 are the x and y coordinates of the circle respectively and c_3 is the radius of the circle. Form a three dimensional parameter space, c_1, c_2, c_3 space, where the dimensions of c_1 and c_2 are the same as for the original image and the dimension of c_3 is equal to the maximum possible radius of the circle to be detected.

Divide the c_1, c_2, c_3 space into cubelike cells of the form $A(i, j, k)$, also known as accumulator cells. For each pixel in the x, y plane that is an element of the circle whose position and size is to be found, perform the following calculation: Use all possible values for c_1 and c_2 that lie within the image to calculate the value of c_3 that satisfies equation (8.1). After each calculation, increment the corresponding accumulator cell, that is: if values of $c_1 = t$ and $c_2 = r$ results in a value of $c_3 = s$ for a specific pixel (x, y) then set

$$A(t, r, s) = A(t, r, s) + 1 \quad (8.2)$$

At the end of the procedure the cell c'_1, c'_2, c'_3 with the highest count is an estimate of the position and radius of the circle to be found: c'_1 and c'_2 will be the x and y coordinates of the circle and c'_3 will be the radius of the circle.

8.3.1.2 Application of Method To be able to use the modified Hough Transform to find the individual raindrops and their sizes, the edges of the raindrops must first be found. This will result in the circles in the x, y plane to be used. The edges were found by applying the Sobel edge detector operators (Gonzalez and Wintz, 1987, Section 7.1.3.1) to the image after the background was removed by morphology techniques. The background removal created a bimodal histogram of the gray-levels. A threshold value was determined by choosing a gray-level value midway between the two peak gray-level values. Thresholding was applied to the image obtained by the Sobel operators using this threshold to get rid of any artifacts, due to the Sobel operators.

The next step was the actual application of the modified Hough Transform to

the image containing the edges. The resulting c_1, c_2, c_3 space was used to find the individual raindrop positions and their sizes as follows:

At each pixel in the image we are concerned about the size of the circle most likely to be centered at that pixel. This means for each c_1 and c_2 parameter, we are interested in the value of c_3 that contains the highest count. Store this value of c_3 for each (c_1, c_2) pixel as well as the count inside the corresponding accumulator cell.

The problem can now be considered as a three dimensional problem: Each pixel (c_1, c_2) has a count that gives the number of pixels in the original image that will lie on a circle whose center is at c_1, c_2 . The corresponding value of c_3 gives the radius of that circle.

The raindrop patterns can be found by a process of elimination:

1. Find the pixel (c_1, c_2) in the image that has the highest count.
2. Remove the raindrop by setting the count in all cells within a radius c_3 from pixel (c_1, c_2) to zero.
3. Store the value of c_1, c_2 and c_3 of the removed raindrop.

Continue this process until a threshold value is reached for the number of counts in the accumulator cells.

8.3.1.3 Special Considerations Although the modified Hough Transform works quite well when the position and size of only one raindrop is to be found, problems can arise when the image has two or more raindrops that are located near each other or overlap. In this case the method can result in "ghost" raindrops.

This means the direct application of the modified Hough Transform can result in accumulator cells with a high count that is not situated in a raindrop in the original image. An example is given in Figure 8.2.

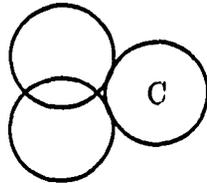


Figure 8.2: "Ghost" raindrops. The left two circles represent raindrops. The circle C is an artifact

The edges of the two overlapping circles result in a cell with a high count at position C. This will result in a "ghost" raindrop as indicated.

I suggest the following change in the detection procedure in Section 8.3.1.2: Consider only pixels inside raindrops in the original image as candidates for the center of raindrops. This is a logical step and gets rid of all "ghost" raindrops. The complete detection procedure is then as follows:

1. Find the pixel (c_1, c_2) in the image that has the highest count.
2. Make sure this pixel is inside a raindrop in the original image.
3. Remove the raindrop by setting the count in all cells within a radius c_3 from pixel (c_1, c_2) to zero.

4. Store the value of c_1 , c_2 and c_3 of the removed raindrop.

The application of this technique to practical situations results in another problem: When applying the Sobel edge detectors to the original image, the resulting edges are not "thin". What is meant by this is that the edges can be 2 or more pixels wide. Although this is satisfactory when the only goal is to enhance the edges visually in an image, it is not satisfactory when trying to estimate parameters from an image. The problem with the wide edges is that it does not describe a unique circle in the x, y plane.

By using all the pixels in the "wide" edges, a number of circles can be fitted through the detected edges, and these circles are not concentric and do not all have the same radius. This is an unsatisfactory situation.

It was necessary to use a technique to find "thin" edges of the raindrops. Such a technique was developed by Bergholm (1987). It is known as edge focussing. By applying this technique, edges of the raindrops were found that were only one pixel thick.

A summary of the edge focussing technique will be given next. In the first step the original image is blurred using a Gaussian mask of size $8\sigma_0$ where σ is generally called the resolution parameter. A value of $\sigma_0 = 4.0$ was used in most images and resulted in a window of size 31×31 (must be an odd number for symmetry). The function of this blurring is to filter out noise in the image and to get rid of unnecessary detail. The next step is to use the Sobel edge detector operators to find the gradient at each pixel in the image. This information is used to calculate the magnitude and angle of the gradient at all pixels. The resulting angle at each pixel is then

grouped into one of six classes depending on which one it is closest too: 0, 30, 60, 90, 120 or 150 degrees. This angle is used to find neighbor pixels to investigate. The magnitude of the neighbor pixels lying on both sides of a specific pixel on the angle line are compared with the magnitude of the pixel under consideration. All pixels for which the neighbor pixels have either both higher or both lower magnitude values, are marked with a one, all other pixels in the image are marked with a zero. This is done because the pixels found in the previous step are lying either on an edge or in a valley. This image of one's and zero's is called the initial edge image.

In the next step the original image is blurred with a smaller mask of size $\sigma = \sigma_0 - 0.8$. All the previous steps are then repeated with the exception that only the edge pixels (marked one) and their 8-neighbors are used to look for new maxima and minima. Old edges are discarded and only the new ones registered.

This process is repeated for decreasing values of σ down to $\sigma = 0.8$ (7×7 window size). The resulting image gives the thin edges. As a result, the edges describe only one circle in the x, y plane and these edges were suitably used by the modified Hough Transform.

8.3.1.4 Simulation Results The whole technique was first applied to a simulated image that contains 3 overlapping raindrops as well as a couple of other raindrops. The simulated image was computer generated. All raindrops in the simulated image were 100% circular. The image background gray level was set equal to a value of 20 on a 0 - 255 scale and the raindrop gray levels were set to 255.

In the simulated case the modified Hough Transform technique was able to find

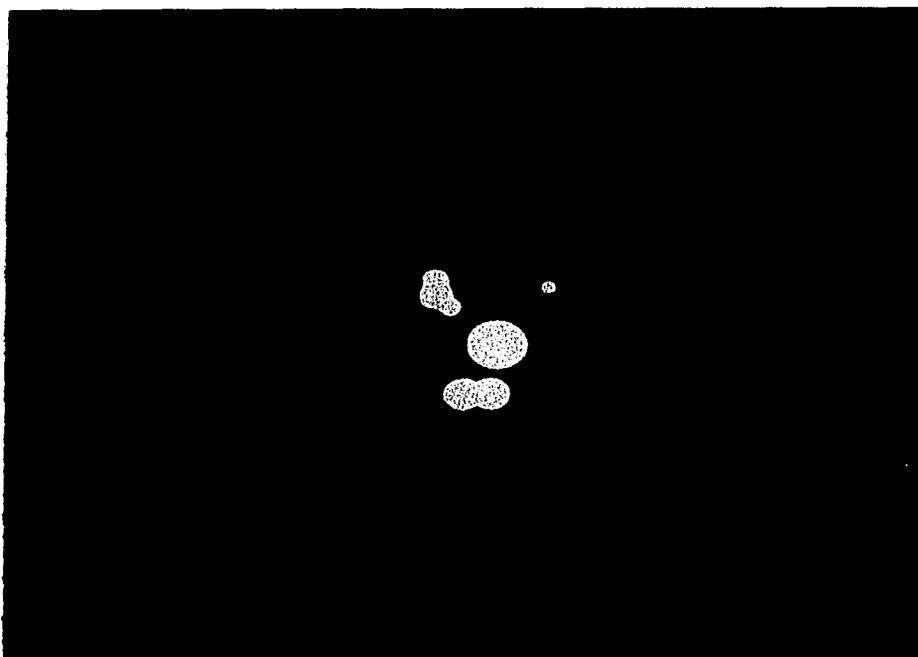


Figure 8.3: Simulated image

the position and size of the individual raindrops to within 0.01% so that for all practical purposes the image generated from these data corresponds 100% with the simulated image in Figure 8.3. This is true because the positions of the simulated raindrop centers, (c_1, c_2) , and the radii of the simulated raindrops, c_3 , found by the modified Hough Transform, corresponded exactly with the values used to generate the simulated image.

8.3.1.5 Practical Results Application of the modified Hough Transform, as discussed in the previous sections, to an actual raindrop image, gave unsatisfactory results. The detected raindrop areas do not correspond well to the actual raindrop areas. Further investigation led to the conclusion that the correct functioning of the

modified Hough Transform requires that the edges to be used must describe a near perfect circle in the x, y plane. Even a small deviation from this constraint results in false positions and areas detected by the modified Hough Transform. It is clear from Figure 8.1 that although the actual raindrops are almost circular in form, they are not perfectly circular. The small deviation is enough to lead to wrong results as found by the application of the modified Hough Transform.

8.3.2 A Robust Method

It is clear that it is necessary to find an alternative method that is more robust with respect to the raindrop geometry to determine the position and sizes of the raindrops in the image accurately. I suggest the following method:

As a first step, remove the background in the original image by using mathematical morphology techniques as already described. The gray-level histogram of the resulting image is bimodal and thresholding can be used to get a binary image as follows

$$\begin{aligned} g(x, y) &= 0 && \text{if } f(x, y) < Th \\ &= 255 && \text{if } f(x, y) \geq Th \end{aligned} \quad (8.3)$$

where

$g(x, y)$ = binary image,

$f(x, y)$ = original image after mathematical morphology, and

Th is the threshold.

As a result, all the pixels in the original image that are inside raindrops will have gray-level values equal to 255 and all other pixels will have a gray-level value of 0.

The individual raindrop positions and their sizes can be found by using the following iterative technique which will be called the robust method:

1. (a) At each pixel in the binary image, find the number of pixels inside a circle with a radius of 1 pixel, centered at the pixel considered, that as a gray level of 255. If this number of pixels is more than 95% of the total number of pixels inside a circle of radius of 1, increase the radius size by 1 and repeat.
 - (b) Continue until the percentage of pixels inside the circle as already described is less than 95%.
 - (c) Reduce the value of the radius used last by 1. This will give the size of the largest possible raindrop centered at the specific pixel.
2. The next step is to find the raindrop positions and their sizes. This can also be done by using an iterative procedure:
 - (a) Find the radius of the largest possible circle in the image as determined in step 1. Using this radius, determining the area of the largest raindrop in the image. The origin of this raindrop will be at the same pixel location.
 - (b) Remove this raindrop by setting the sizes of all possible circles at pixel locations inside the radius of the raindrop being removed to zero.
 - (c) Repeat for all possible raindrops in the image down to raindrops with a radius of 1 pixel.

This method is much more robust and approximates all raindrops in the original image as circular areas even though the actual raindrop areas may deviate from circular. The closer the raindrops in the original image are to being circular, the more accurate this approach is.

8.3.2.1 Practical Application Figure 8.4 shows a cut of an observed image, digitized using gray levels in the 0 - 255 range. This image includes many overlapping raindrops and we can see that the individual raindrop areas differ significantly. By investigation of this image it is clear that the gray levels inside raindrops at certain portions of the image are darker than the gray levels of the background at other parts of the image. This last observation suggests that mathematical morphological techniques must be used to get rid of the background.

Figure 8.5 shows the same image after mathematical morphology has been applied. It can be seen from this image that the morphological technique was able to separate the raindrops from the background and the resulting image is clearly bimodal. It is also clear from Figure 8.5 that the mathematical morphological technique did not distort the raindrop areas in the original image, Figure 8.4. A gray-level histogram was drawn for the image in Figure 8.5. This histogram was bimodal, as expected, and by using the gray-level value between the two peaks in the histogram as a threshold level, the binary image in Figure 8.6 was obtained. This image contains the same information as Figure 8.5, but all the raindrop areas were now set equal to gray-level value 255 and the background was set equal to gray level 0. The binary image in Figure 8.6 was used with the robust method to find the raindrop areas and their sizes.

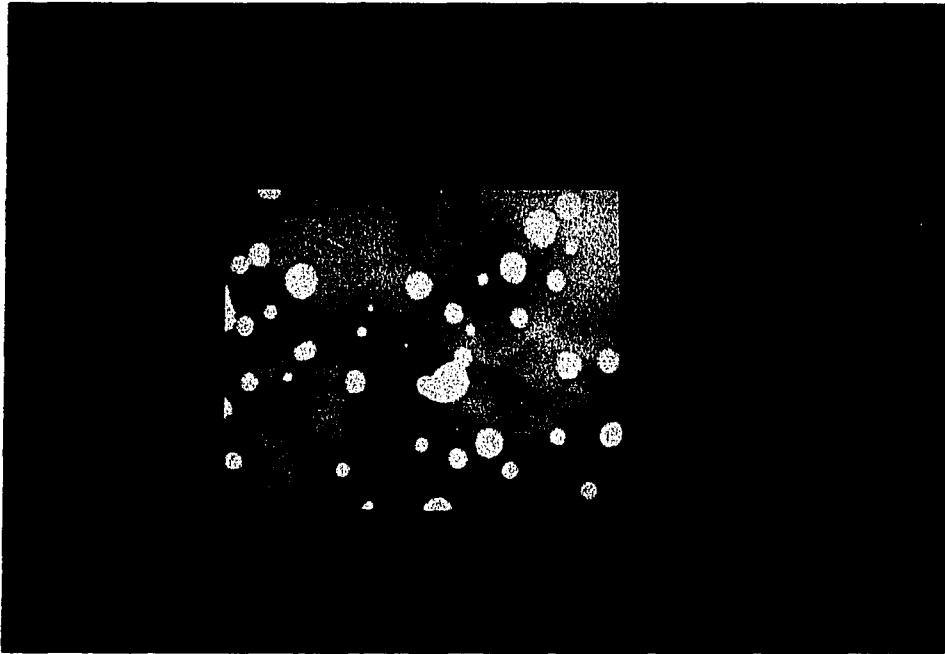


Figure 8.4: Observed image

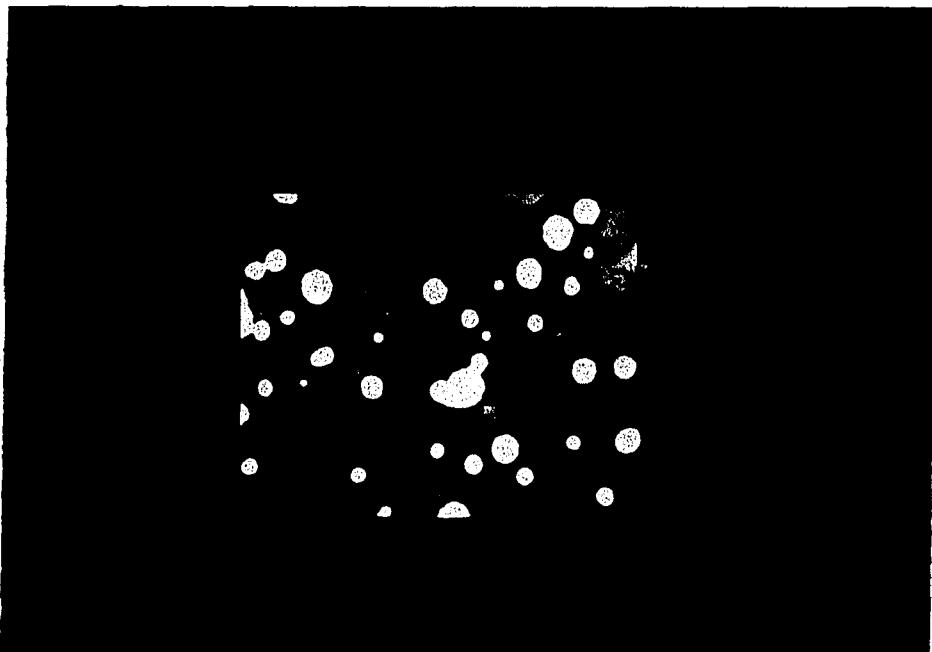


Figure 8.5: Background removed by morphology

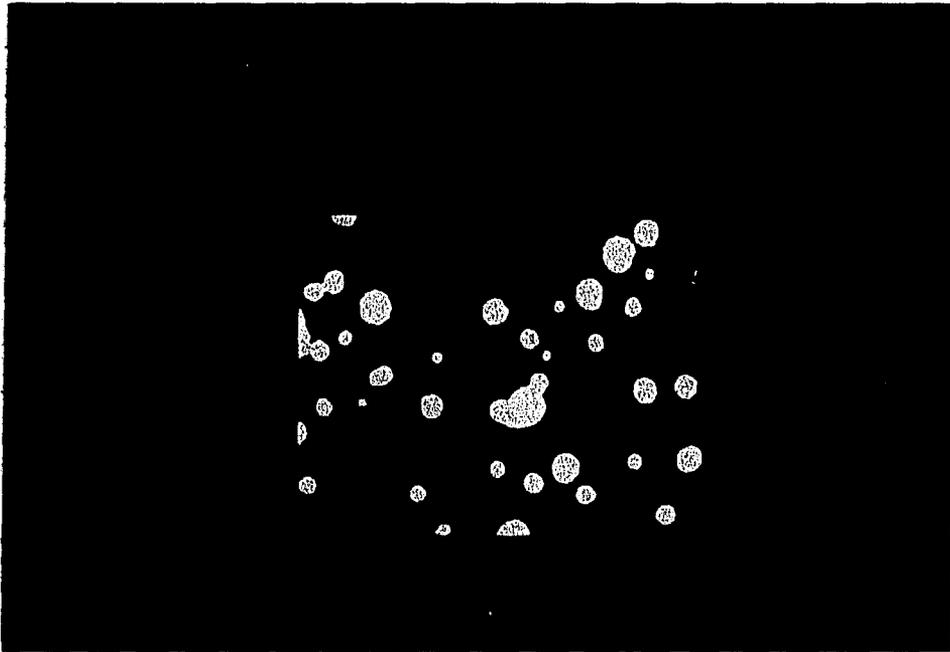


Figure 8.6: Binary image after thresholding

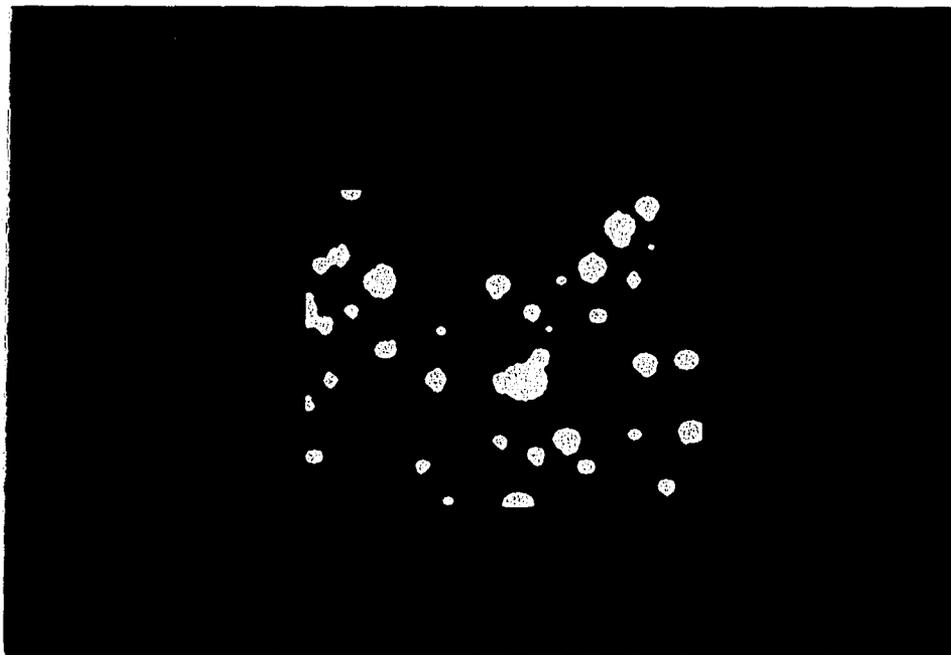


Figure 8.7: Generated image

Figure 8.7 shows the resulting image, obtained by computer generation, using the results of the robust method to determine raindrop locations and sizes. By comparing Figure 8.7 with Figure 8.4, the effectiveness of the robust method can be seen. Most of the raindrops in the original image were detected, the only exceptions were the raindrops that were smaller than a circle with radius of 1 pixel. However, Figure 8.7 shows that the raindrop edges were not smooth, due to a number of small raindrops attached to the edges of larger raindrops. This problem will be discussed further in the next section.

8.3.2.2 Improved Robust Method The method discussed in Section 8.3.2 works perfectly when all the raindrops are 100% circular and there are no overlapping raindrops. When a raindrop is not 100% circular, the method will fit the largest possible circle in a given raindrop area. This technique will result in errors if certain precautions are not taken. This is best explained by an example: Look at the raindrop and the largest circle fitted inside this raindrop, shown in Figure 8.8.



Figure 8.8: Raindrop with fitted circle

After fitting the largest possible circle inside the given raindrop, there still re-

mains some raindrop area not accounted for. By applying the technique as already discussed, another small circle is fitted in this remaining raindrop area as shown in Figure 8.9.

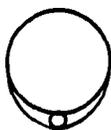


Figure 8.9: Raindrop with false fitted circle

One way to get around this problem is to set a limit on the size of the smallest circle to be fitted in the entire image. Although this technique solves the problem above, it has the disadvantage of not being able to detect the small raindrops that are present in some images.

It seems that the best way to solve this problem is to detect all possible circles in an image down to a radius of 1 pixel, but use only those small drops that do not touch larger drops. Application of this modification to the same image as in Section 8.3.2.1, gives the results shown in Figure 8.10. It can be seen from this image that the modification gets rid of the problem of small drops attached to larger drops in an image, but that it does not reduce the effectiveness of the robust method to detect all the raindrops in this specific image.

By comparing Figure 8.7 with Figure 8.10, it can be seen that in Figure 8.10 the edges of the larger raindrops are smooth, without small drops attached to them. This

was not true in Figure 8.7 where a number of small (false) raindrops were attached to the larger raindrops. Using the modified robust method it is not possible to detect very small raindrops that are very close to, or touching, other raindrops. However, the tradeoff between the detection of very small raindrops and the improved accuracy in measuring remaining raindrops is worthwhile.

8.3.2.3 Conclusions Although this method doesn't suffer from the same shortcomings of the modified Hough Transform (and does not require that the raindrop edges be perfect circles) it is very sensitive to the percentage of pixels inside a circle that is actually part of the original raindrop. When this percentage is too low, it can result in wrong detected raindrops when 2 raindrops are overlapping. The 95% value used in this project was found to be a good predicted value to be used and gives satisfactory results, as already shown.

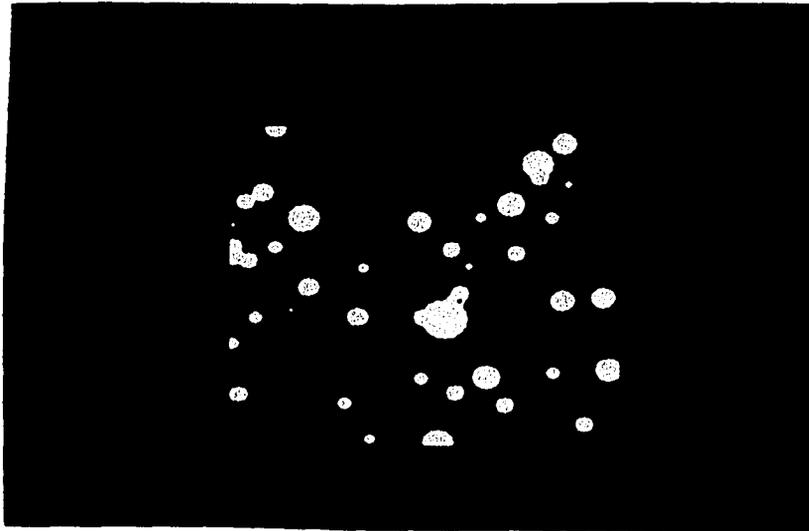


Figure 8.10: Modified robust method

9. GENERAL PARAMETER ESTIMATION USING DIGITIZED IMAGES

9.1 Introduction

From the analysis of the radio astronomy problem in the previous chapters, it was found that the double smoothing technique gives the best parameter estimates when using simulated flux density values that were contaminated with zero mean white Gaussian noise. Also, it was found that using the double smoothing technique, estimated parameter maps could be obtained from the *observed* flux density values of nebula NGC 7027 that compare reasonably well with parameter maps determined using other techniques (Chapter 3).

In this chapter an investigation will be done to find the effectiveness of the double smoothing technique to estimate parameters from digitized images in which the characteristics of interest are described by two general classes of equations often encountered in engineering problems, namely logarithmic equations and polynomial equations. It will be assumed that these equations describe a physical characteristic in the images, as before, and the parameter estimation will be done on a pixel- to-pixel basis. The investigation will be constrained to the situation of three digitized images available to estimate three unknown parameters in three equations. The equations

used will be chosen to be as general as possible. The results obtained using the noisy data values directly in the parameter estimation problem will be compared with the results obtained using the double smoothing technique.

9.2 Logarithmic Equations

9.2.1 Problem Statement

Problems that obey a logarithmic relationship were investigated by using the following set of equations

$$y_1 = \ln(x_1)^{t_1} + t_2 x_2 + x_3 = t_1 \ln(x_1) + t_2 x_2 + x_3 \quad (9.1)$$

$$y_2 = \ln(x_1)^{t_3} + t_4 x_2 + x_3 = t_3 \ln(x_1) + t_4 x_2 + x_3 \quad (9.2)$$

$$y_3 = \ln(x_1)^{t_5} + t_6 x_2 + x_3 = t_5 \ln(x_1) + t_6 x_2 + x_3 \quad (9.3)$$

where

x_1, x_2, x_3 are variables, and

$t_1, t_2, t_3, \dots, t_6$ are constants (known).

By substituting x_4 for $\ln(x_1)$, three linear equations in three unknowns result and a unique solution for x_4, x_2 and x_3 can be obtained as long as the three equations are linearly independent. The resulting solution for x_1 will then be given by

$$x_1 = e^{x_4}. \quad (9.4)$$

9.2.2 Parameter Estimation in Logarithmic Equations

To evaluate the effect of the double smoothing technique on the estimation of parameters from digitized images using the defined logarithmic equations, a simulation study was undertaken. The way the study was done, is as follows. First of all, three x-parameter images were generated for the x_4 , x_2 and x_3 parameters. These images were each of size 32×28 pixels and in each case it was assumed that the values in the individual images lay on a 2-D Gaussian curve. The equation used to generate the images was

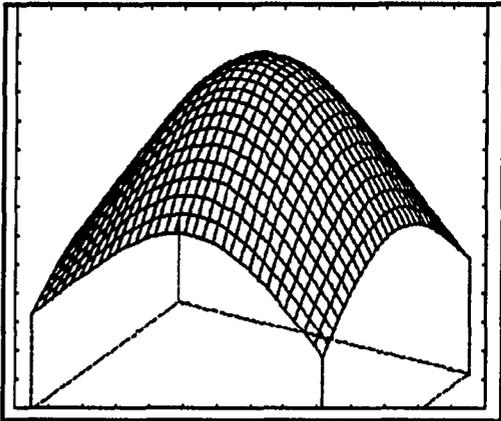
$$x(i, j) = K e^{-2[(i-x_0)^2 + (j-y_0)^2]/729} \quad (9.5)$$

were the constants for the individual images were arbitrarily chosen as

Image	K	x_0	y_0
x_4	1 500	15	15
x_2	1 000	20	20
x_3	2 000	10	25

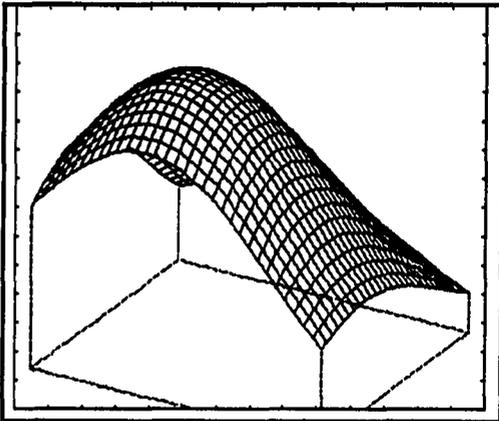
The simulated x-images are shown in Figures 9.1, 9.2 and 9.3. These simulated x-images were then used to generate three y-images by using equations (9.1), (9.2) and (9.3) to find y-values at each pixel location.

The constants $t_1, t_2, t_3, \dots, t_6$ were arbitrarily chosen, the only constraint was that the resulting three equations must not be linearly dependent. The values chosen for these constants were $t_1 = 0.7$, $t_2 = 1.7$, $t_3 = 0.8$, $t_4 = 1.4$, $t_5 = 0.9$ and $t_6 = 2.3$.



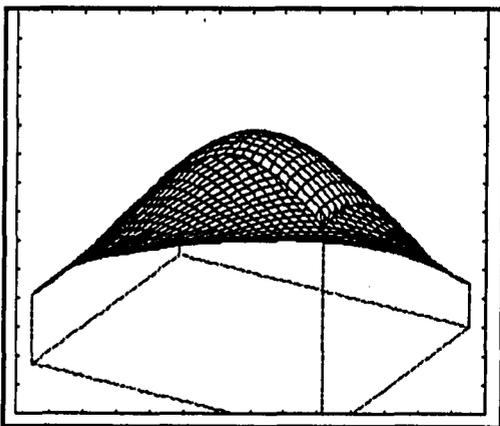
Maximum value = 1500
Ground level value = 0

Figure 9.1: Logarithmic X_1 simulated map



Maximum value = 1000
Ground level value = 0

Figure 9.2: Logarithmic X_2 simulated map



Maximum value = 2000
Ground level value = 0

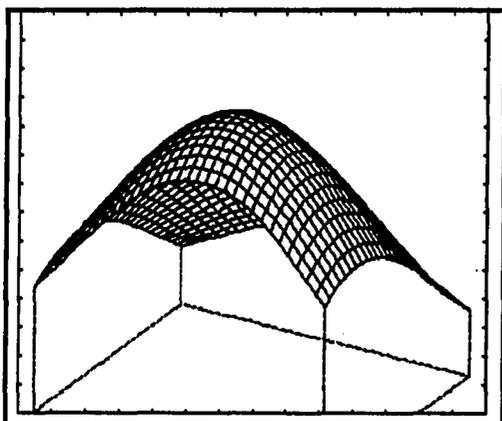
Figure 9.3: Logarithmic X_3 simulated map

The resulting y-images are shown in Figures 9.4, 9.5 and 9.6. These y-images were then contaminated by adding zero mean white Gaussian noise, having a standard deviation of 10% of the individual pixel values, to each pixel in all three y-images. The noisy y-images are shown in Figures 9.7, 9.8 and 9.9.

The noisy y-images were used to estimate the x-parameters in the original simulated x-images. First, the x-parameters were estimated using the noisy y-values directly. The double smoothing technique was then used to estimate the same x-parameters. A cut was arbitrarily made through row 16 of the resulting estimated parameter images and the result is shown in Figures 9.10, 9.11 and 9.12. This cut was typical of similar cuts made through the estimated parameter images at different rows. The effect of the double smoothing technique can clearly be seen. The curves labelled "0% noise" are the reference curves. The curves labelled "10% noise" show the results of estimating the x-parameters from the noisy y-images directly and the curves labelled "10% DS" show the results of estimating the x-parameters from the noisy y-images using the double smoothing technique.

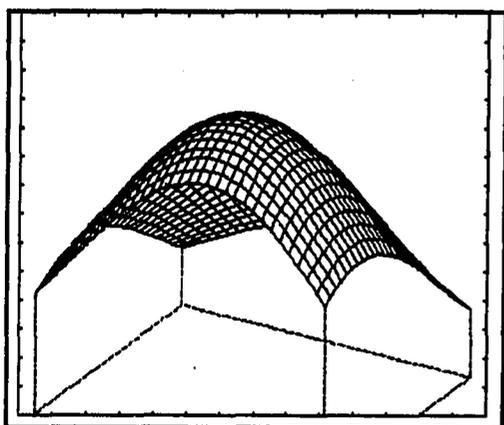
9.2.3 Statistical Analysis

To investigate the effectiveness of the double smoothing technique qualitatively, the variances of the estimated parameters x_4 , x_2 and x_3 , using the noisy image data, were compared with the variances of the same estimated parameters using the double smoothing technique.



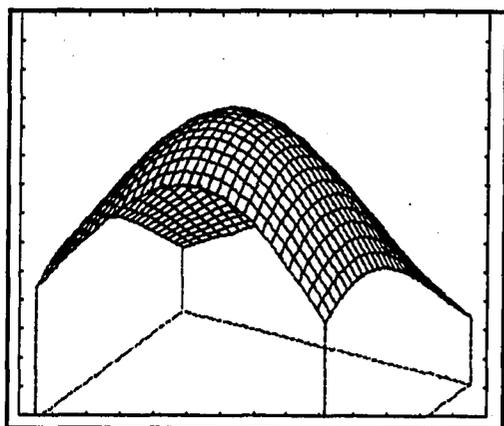
Maximum value = 4320
Ground level value = 0

Figure 9.4: Logarithmic Y_1 generated map



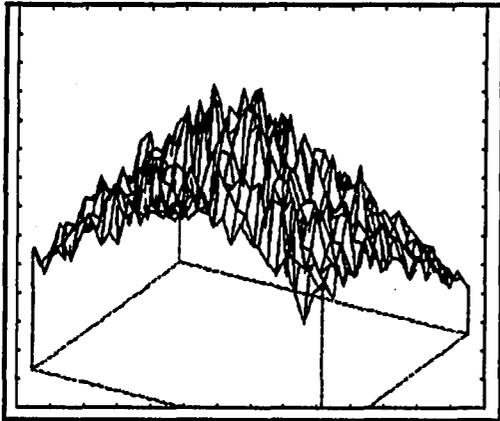
Maximum value = 4180
Ground level value = 0

Figure 9.5: Logarithmic Y_2 generated map



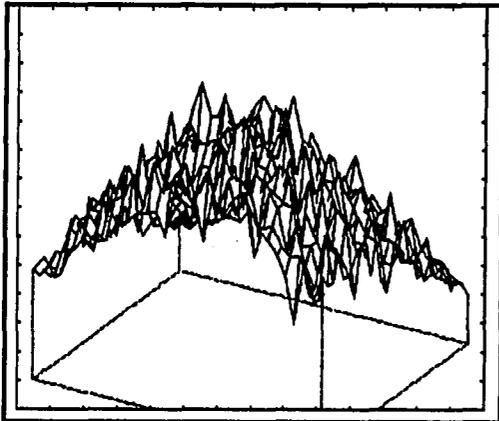
Maximum value = 5150
Ground level value = 0

Figure 9.6: Logarithmic Y_3 generated map



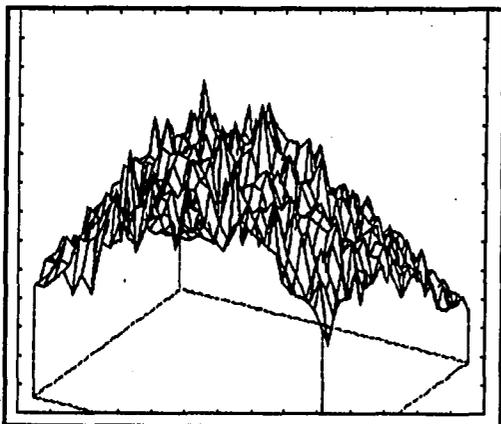
Maximum value = 5290
Ground level value = 0

Figure 9.7: Logarithmic noisy Y_1 map



Maximum value = 4920
Ground level value = 0

Figure 9.8: Logarithmic noisy Y_2 map



Maximum value = 6000
Ground level value = 0

Figure 9.9: Logarithmic noisy Y_3 map

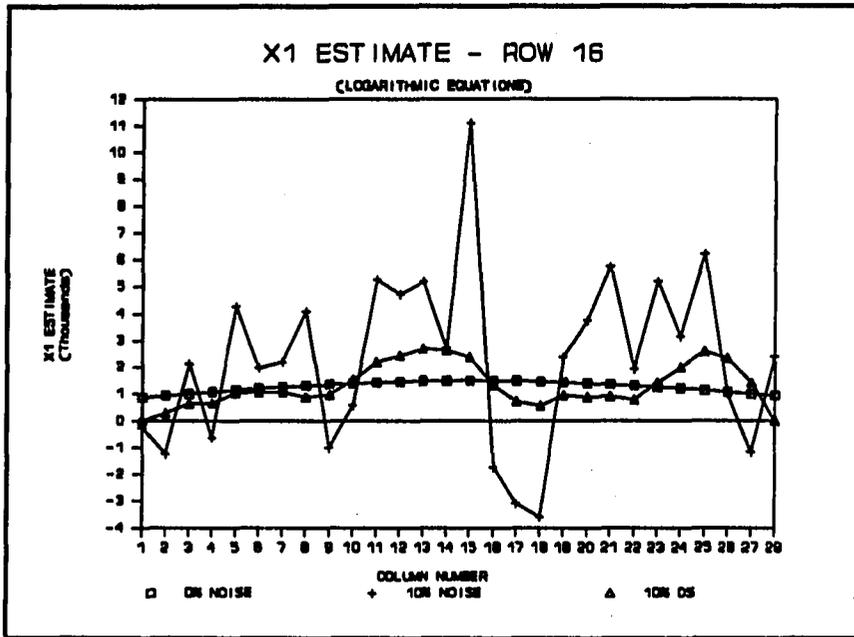


Figure 9.10: Cut through row 16 of X_1 estimated parameter maps, logarithmic equations used

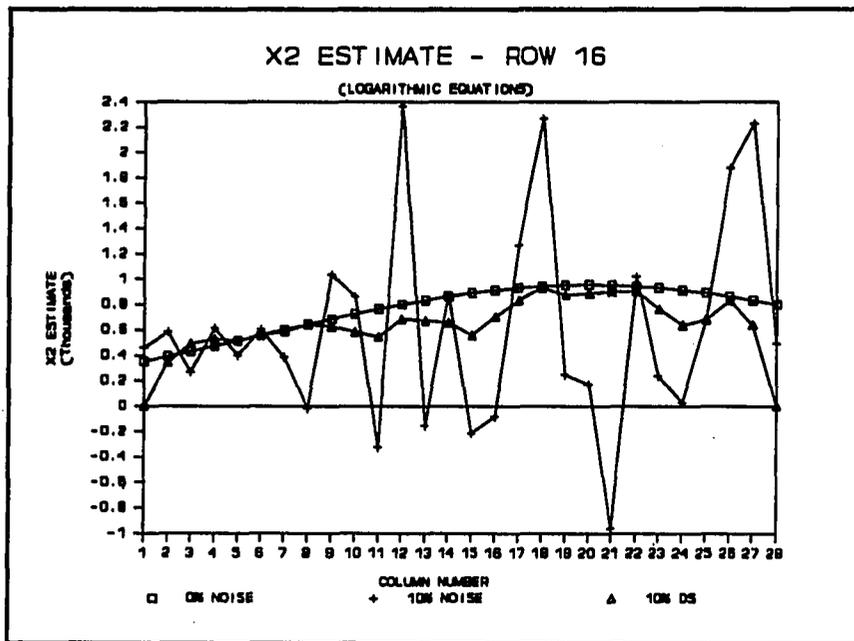


Figure 9.11: Cut through row 16 of X_2 estimated parameter maps, logarithmic equations used

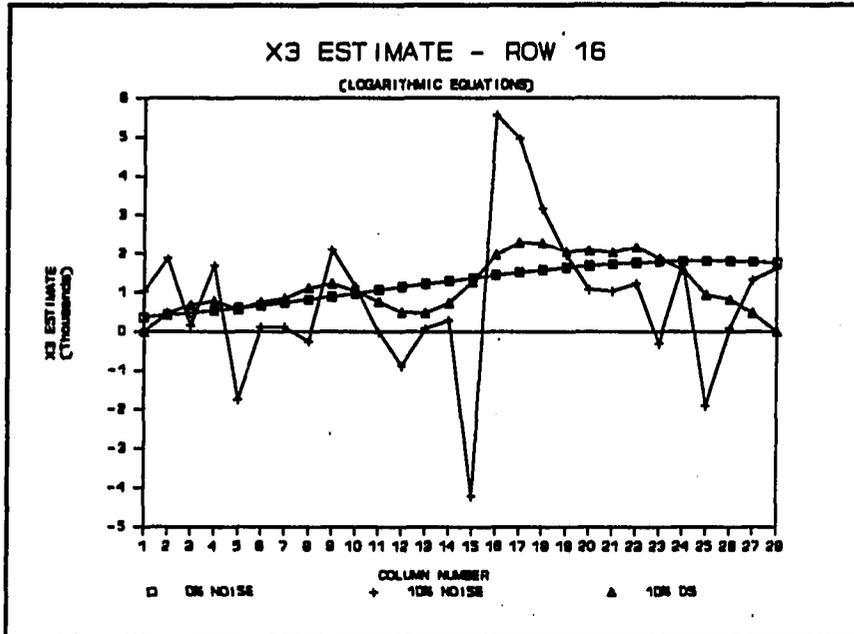


Figure 9.12: Cut through row 16 of X_3 estimated parameter maps, logarithmic equations used

The variance of the estimated parameters x_4 , x_2 and x_3 , for the direct approach, were first determined by using the following approximate formula given by Hahn and Shapiro (1967)

$$Var(x_j) \approx \sum_{i=1}^n \left(\frac{\partial h}{\partial y_i} \right)^2 Var(y_i) \quad (9.6)$$

where $y_i = h(x_1, x_2, \dots, x_n)$.

A numerical example for a specific situation is given

Parameter	t_1	t_2	t_3	t_4	t_5	t_6	x_4	x_2	x_3
Value Used	0.7	1.7	0.8	1.4	0.9	2.3	1500	1000	2000

The resulting values for y_1 , y_2 and y_3 are $y_1 = 4750$; $y_2 = 4600$ and $y_3 = 5650$. Assume that zero mean white Gaussian noise with a standard deviation (STD) of 10% of the individual values of y_1 , y_2 and y_3 was added to the values of y_1 , y_2 and y_3 at each pixel location respectively. By using the approximate formula in equation (9.6), the following values for the standard deviations of x_4 , x_2 and x_3 were obtained

Variable	STD	STD(%)
x_4	4 469.51	297.97
x_2	982.53	98.25
x_3	2 926.39	146.32

The column "STD(%)" gives the standard deviation of the x-parameters as a percentage of the nominal values used for these parameters.

In an attempt to verify these results for the standard deviation of the x-parameters independently, a Monte Carlo analysis of the same situation was done. The results of the Monte Carlo analysis, at the 5% confidence level, was as follows

Variable	STD	STD(%)
x_4	4 460	297.33
x_2	933	93.30
x_3	3 030	151.50

It is clear that in this specific case the standard deviation of variables x_4 , x_2 and x_3 evaluated using the two different methods are almost identical. It is clear from these analyses that zero mean white Gaussian noise with a standard deviation of 10% of the individual y-parameter values results in enormous errors in the estimated x-parameters.

In order to find the standard deviation of x_4 , x_2 and x_3 at different values of these variables, the approximate formula was used to generate the graphs in Figures 9.13, 9.14 and 9.15. From these graphs it can be seen that the standard deviation of the estimated parameters are very sensitive to the nominal data values of the x-parameters used. In general, however, as a percentage of the nominal values used, the standard deviation of parameter x_4 increases with an increase in the nominal values of parameters x_2 and x_3 . The standard deviation of parameter x_2 decreases with an increase in the nominal value of parameter x_2 and increases with a decrease in the nominal value of parameter x_3 . The standard deviation of parameter x_3 increases with an increase in the nominal value of parameter x_2 and decreases with an increase in the nominal value of parameter x_3 . The same situation was also addressed when the parameters were estimated using the double smoothing technique. The resulting graphs are shown in Figures 9.16, 9.17 and 9.18. These graphs were generated using a Monte Carlo analysis. It can be seen that after the application of the double smoothing technique the standard deviation of the estimated parameters are on the average only 1/4 compared to the case before. Therefore, the double smoothing technique succeeds to find better estimates from noisy image data also in the case of logarithmic equations used to describe image characteristics.

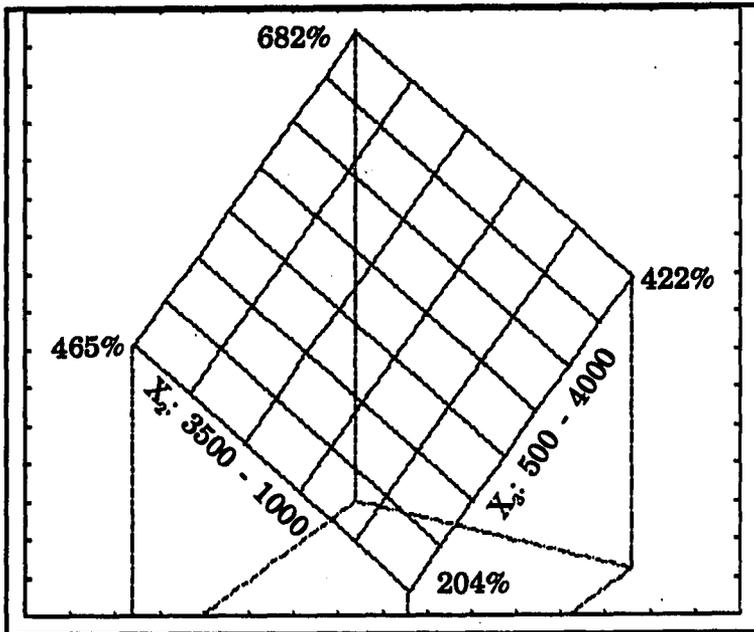


Figure 9.13: Standard deviation of X_1 estimated parameters using logarithmic equations

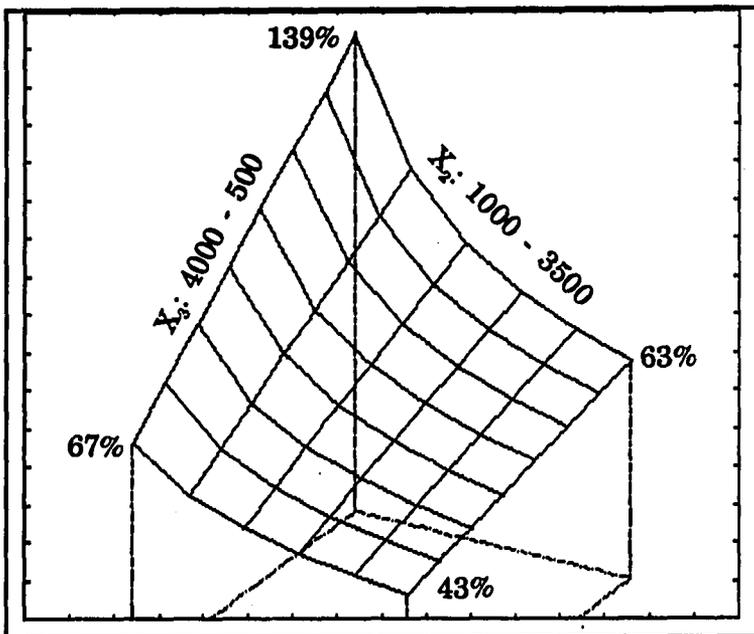


Figure 9.14: Standard deviation of X_2 estimated parameters using logarithmic equations

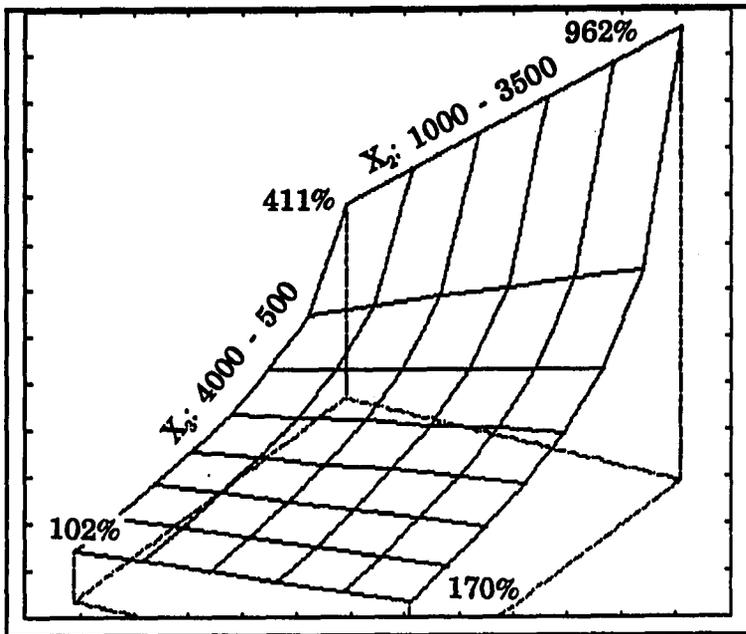


Figure 9.15: Standard deviation of X_2 estimated parameters using logarithmic equations

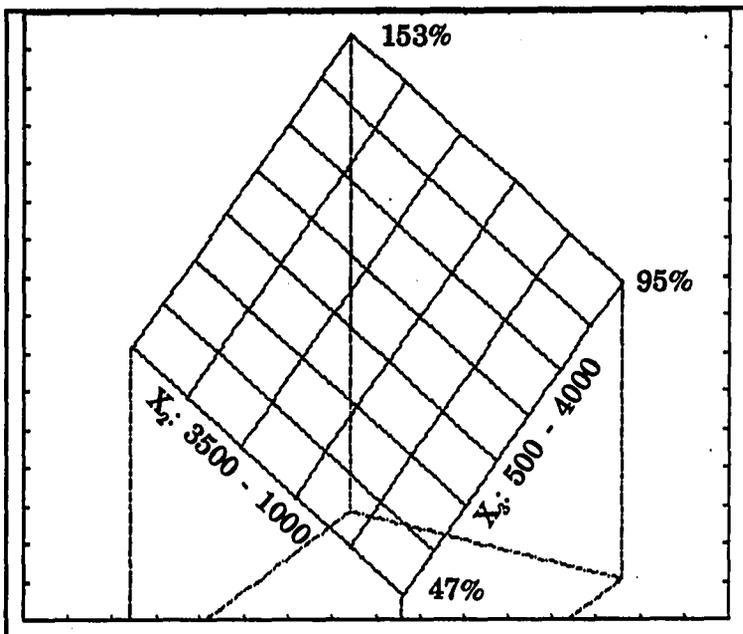


Figure 9.16: Standard deviation of X_1 estimated parameters using logarithmic equations determined by double smoothing technique

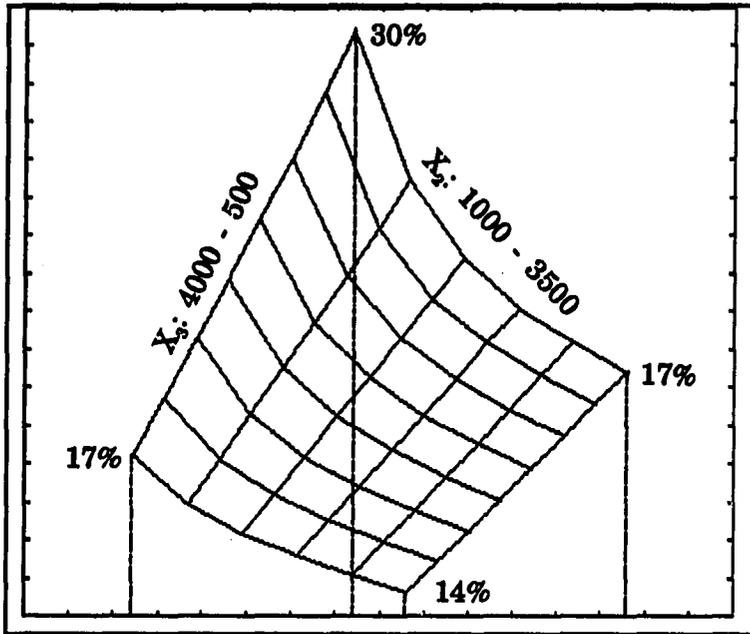


Figure 9.17: Standard deviation of X_2 estimated parameters using logarithmic equations determined by double smoothing technique

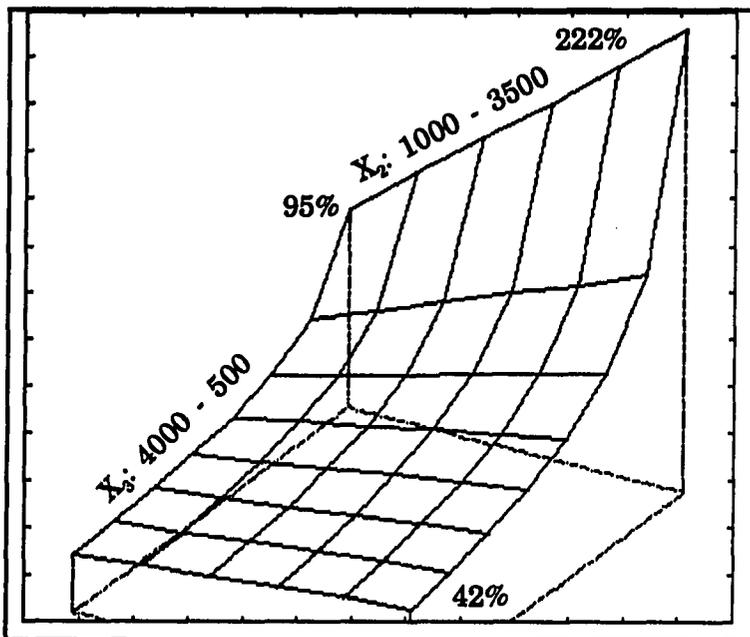


Figure 9.18: Standard deviation of X_3 estimated parameters using logarithmic equations determined by double smoothing technique

9.2.4 Conclusion

The analysis in this section concentrated on the standard deviations of the parameters x_4 , x_2 and x_3 . However, the standard deviation of the parameter x_1 in the original equations used, must still be determined.

Remember that

$$x_1 = e^{x_4} \quad (9.7)$$

and therefore

$$Var(x_1) = e^{Var(x_4)} \quad (9.8)$$

It has already been shown that the standard deviation of the estimated parameter x_4 is very large, even when the double smoothing technique is used. Keeping the result of equation (9.8) in mind, it can be seen that the estimation of parameter x_1 in the logarithmic equations defined in this analysis, from digitized image data, is worthless when the data values are noisy, even after the application of the double smoothing technique. The best solution would be to obtain more data measurements.

9.3 Polynomial Equations

9.3.1 Problem Statement

Problems that obey a polynomial relationship were investigated by using the following set of equations:

$$y_1 = x_1^{t_1} + t_2 x_2 + x_3 \quad x_1 > 0 \quad (9.9)$$

$$y_2 = x_1^{t_3} + t_4 x_2 + x_3 \quad x_1 > 0 \quad (9.10)$$

$$y_3 = x_1^{t_5} + t_6 x_2 + x_3 \quad x_1 > 0 \quad (9.11)$$

where

x_1, x_2, x_3 are variables, and

t_1, t_2, \dots, t_6 are constants (known).

The way these three equations, in three unknowns, were solved exactly, was to find the first order Taylor Series expansion of each equation in x_1 and solve the resulting equations for the three unknown parameters. These approximate solutions were then used in a Newton type procedure to solve for the unknown parameters exactly.

9.3.2 Parameter Estimation in Polynomial Equations

A similar study as the one for the logarithmic equations was done to find the effect of the double smoothing technique on the estimation of parameters, from digitized images, using polynomial equations. Three x-parameter images were generated for the x_1, x_2 , and x_3 parameters. Equation (9.5) was again used in this simulation and the constants chosen in this case were

Image	K	x_0	y_0
x_1	25 000	15	15
x_2	5 000	20	20
x_3	19 000	10	25

The x_1 -image was then shifted up by adding 40 000 to the values of the individual pixels. This was done to ensure that the estimated x_1 -parameters from the noisy data were still positive, because equations (9.1), (9.2) and (9.3) are not defined for negative x_1 -parameters.

The simulated x-images are shown in Figures 9.19, 9.20 and 9.21. The y-images obtained from these x-images, using equations (9.9), (9.10) and (9.11), are shown in Figures 9.22, 9.23 and 9.24.

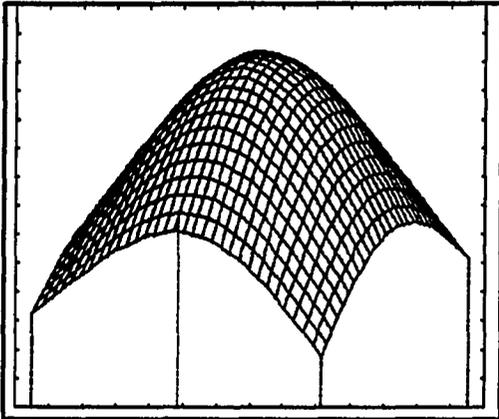
The same values were chosen for the constants t_1, t_2, \dots, t_6 as before. Zero mean white Gaussian noise with the same characteristics as in Section 9.2 was added to these images. The resulting noisy images are shown in Figures 9.25, 9.26 and 9.27.

The results of the different estimation procedures are displayed the same way as before in Figures 9.28, 9.29 and 9.30 for a similar cut through the estimated parameter images at row 16. The symbols used in these figures also have the same definitions as in Section 9.2. Again it can be seen that the double smoothing technique improve the accuracy of the estimated parameters dramatically.

9.4 Conclusion

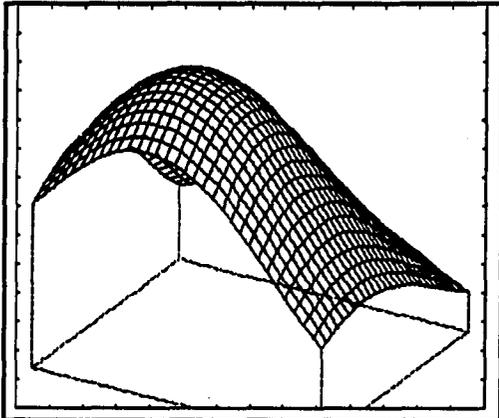
Although only two specific situations were addressed, both these situations support the results obtained before in the radio astronomy problem.

The accuracy of the estimated parameters, using the double smoothing technique, is much better than the accuracy of the same parameters using the noisy image. When estimating parameters from noisy image data, the double smoothing technique seems a good place to start.



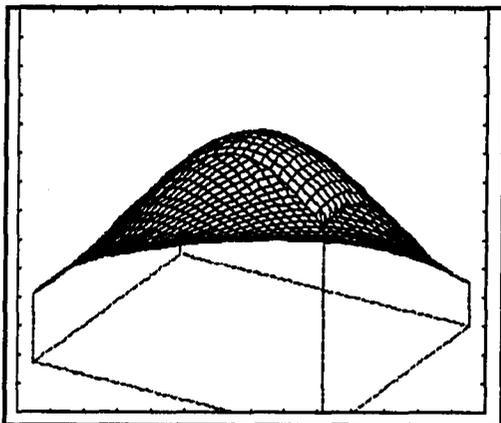
Maximum value = 65000
Minimum value = 0

Figure 9.19: Polynomial X_1 simulated map



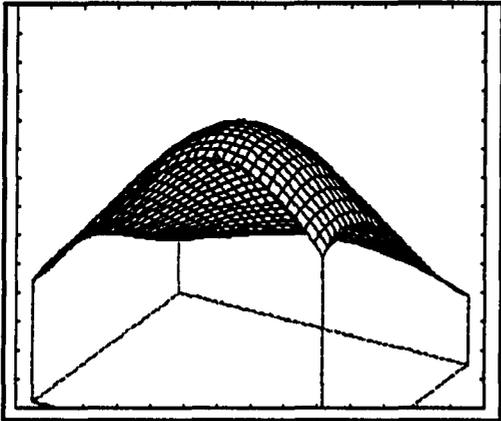
Maximum value = 5000
Ground level value = 0

Figure 9.20: Polynomial X_2 simulated map



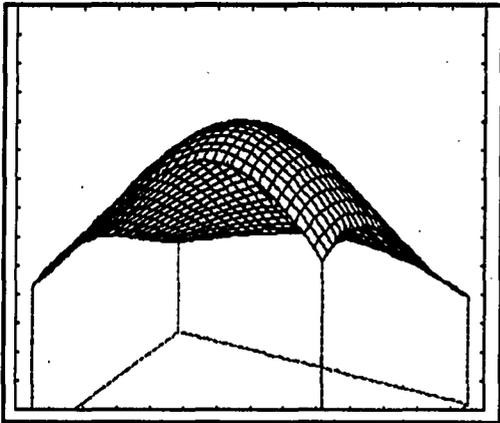
Maximum value = 19000
Ground level value = 0

Figure 9.21: Polynomial X_3 simulated map



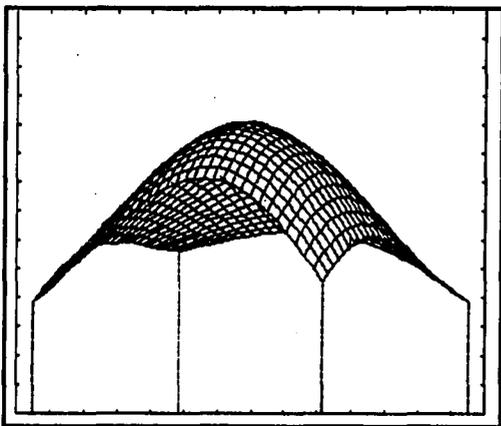
Maximum value = 26200
Ground level value = 0

Figure 9.22: Polynomial Y_1 generated map



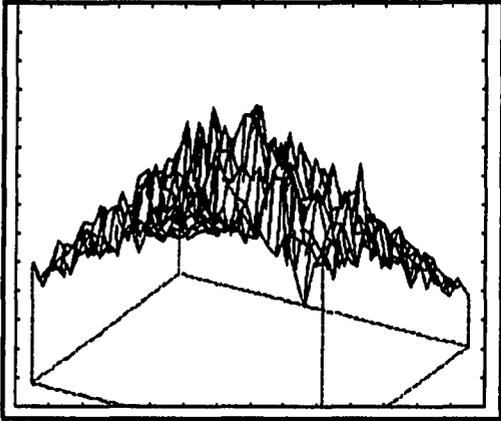
Maximum value = 30200
Ground level value = 0

Figure 9.23: Polynomial Y_2 generated map



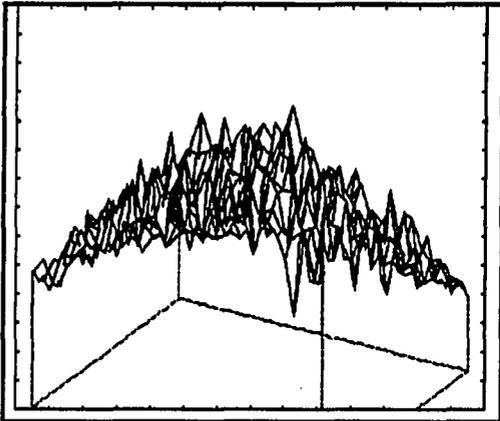
Maximum value = 43300
Minimum value = 18300

Figure 9.24: Polynomial Y_3 generated map



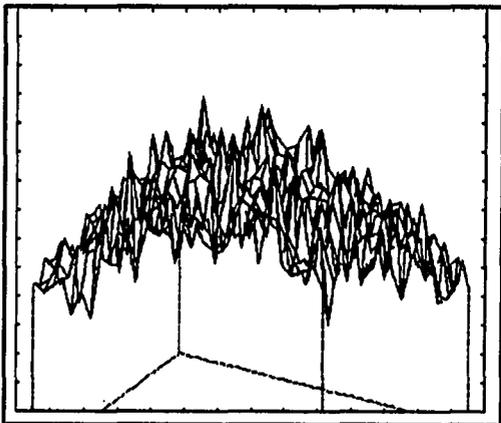
Maximum value = 31300
Ground level value = 0

Figure 9.25: Polynomial noisy Y_1 map



Maximum value = 35400
Ground level value = 0

Figure 9.26: Polynomial noisy Y_2 map



Maximum value = 50700
Ground level value = 0

Figure 9.27: Polynomial noisy Y_3 map

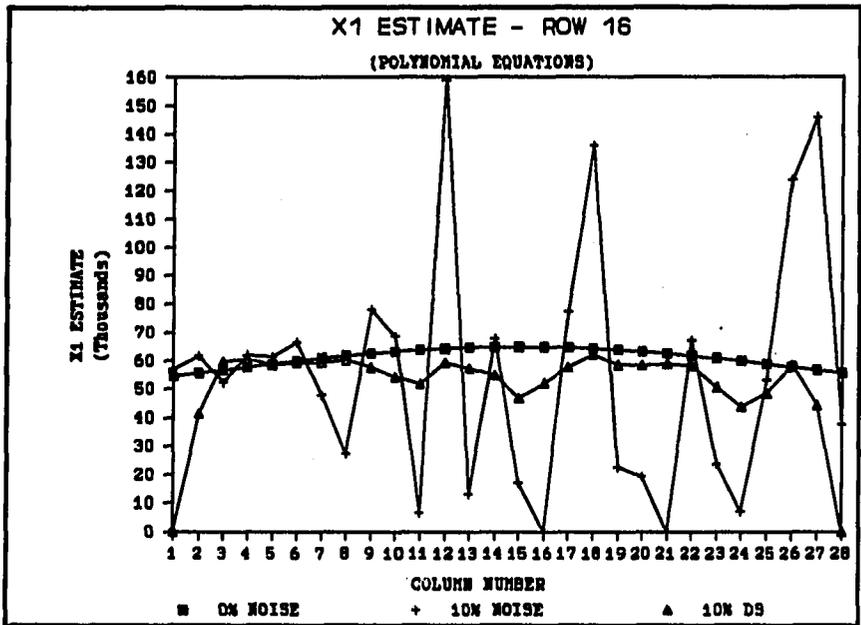


Figure 9:28: Cut through row 16 of X_1 estimated parameter maps, polynomial equations used

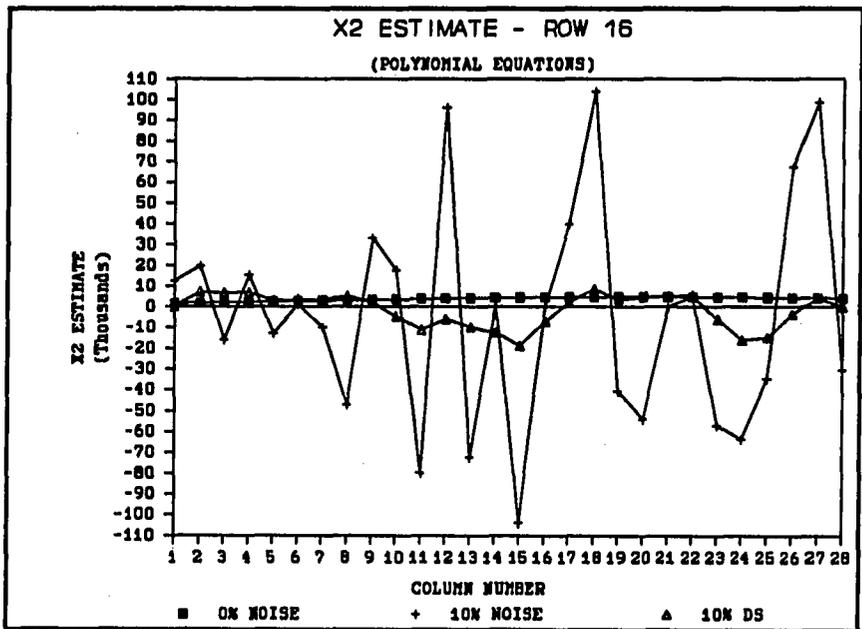


Figure 9:29: Cut through row 16 of X_2 estimated parameter maps, polynomial equations used

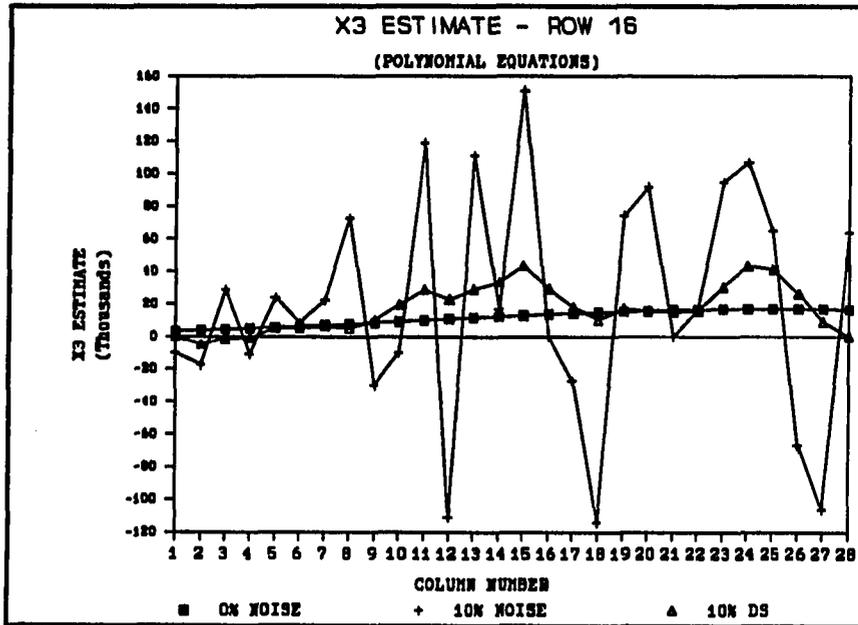


Figure 9:30: Cut through row 16 of X_3 estimated parameter maps, polynomial equations used

10. CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

10.1 Conclusions

A number of conclusions can be drawn from the work done in this dissertation:

1. From the parameter estimation problem using the VLA data it was found that it is very difficult to estimate n parameters in n nonlinear equations when using noisy observed data values. The direct solution of the n parameters led to results that were not physically possible. The best estimated parameters were obtained by using the double smoothing technique defined in this dissertation. These results were even better than the results obtained when the technique of random field restoration was used to restore the noisy simulated flux density maps as was verified by simulation studies.
2. The technique developed to estimate the locations and sizes of raindrops, as collected on a sheet of paper, gave very satisfactory results for the images analyzed. However, the accuracy of this technique depends on how close the raindrop areas are from being circular.
3. As a general result it was concluded that the estimation of n parameters from n noisy digitized images have large errors, even when the double smoothing

technique is used. This was verified for two specific types of equations often encountered in engineering, namely polynomial equations and logarithmic equations. The actual estimation errors, however, are a function of the specific equations used.

10.2 Suggestions for Future Work

1. To find the "true" temperature at the back of nebula NGC 7027, it would be necessary to obtain more observations of the nebula at different frequencies. This would help to reduce the errors in the estimated parameters. At this point in time it is not possible to verify if the temperature map obtained using the double smoothing technique is more correct than the temperature map obtained using the approximate technique.
2. To add more credibility to the work done on the estimation of the raindrop locations and sizes, it would be necessary to analyze many more raindrop images using the robust method to draw statistical conclusions about the accuracy of this method. It is also necessary to determine if this method can be used to detect areas that are not circular.
3. Concerning the general estimation of n parameters from n digitized images, much work still needs to be done. The work in Chapter 9 must be extended to include many more basic equations generally encountered in engineering. In the situation when more images are available than unknowns, work also needs to be done to find the accuracy of estimated parameters using the double smoothing technique together with the well established techniques of parameter estimation

compared to the situation when the parameters are estimated without using the double smoothing technique.

11. BIBLIOGRAPHY

- Aller, L. H. 1984. *Physics of Thermal Gaseous Nebulae: Physical Processes in Gaseous Nebula*. D. Reidel Pub. Co., Boston.
- Andrews, H. C. and B. R. Hunt. 1977. *Digital Image Restoration*. Prentice-Hall Inc., Englewood Cliffs, New Jersey.
- Basart, J. P. and C. T. Daub. 1987. *Temperature and Emission - Measure Distributions for Several Planetary Nebulae*. *The Astrophysical Journal* 317:412-422.
- Bergholm, F. 1987. *Edge Focussing*. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9(6):726-741.
- Brown, R. G. 1983. *Introduction to Random Signal Analysis and Kalman Filtering*. John Wiley & Sons, New York.
- Chellappa, R. and R. L. Kashyap. 1981. *Synthetic Generation and Estimation in Random Field Models of Images*. Pp. 577-582. Proceedings of the IEEE Computer Society Conference on Pattern Recognition, Image Processing, Dallas, Texas. IEEE Computer Society Press, Los Angeles.
- Chellappa, R. and R. L. Kashyap. 1982. *Digital Image Restoration Using Spatial-Interaction Models*. *IEEE Transactions on Acoustics, Speech and Signal Processing* ASSP-30(3):461-472.
- Flower, D. R. 1983. *Planetary Nebulae: Symposium no. 103, held at University College, London, U.K., August 9-13, 1982*. D. Reidel Pub. Co., Boston.
- Freund, J. E. and R. E. Walpole. 1987. *Mathematical Statistics*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

- Gonzalez, R. C. and P. Wintz. 1987. *Digital Image Processing*. Addison-Wesley Publishing Company, Cambridge, Mass.
- Hahn, G. J. and S. S. Shapiro. 1967. *Statistical Models in Engineering*. John Wiley & Sons, New York.
- International Astronomical Union. 1987. *Planetary Nebulae: Proceedings of the 191st Symposium of the International Astronomical Union held in Mexico City, Mexico, October 5-9, 1987*. Edited by Silvia Torres-Peimbert. Kluwer Academic Publishers, Boston.
- Kalman, R. E. 1960. *A New Approach to Linear Filtering and Prediction Problems*. Trans. of the ASME, J. of Basic Engr. 82:35-45.
- Kalman, R. E. and R. S. Bucy. 1961. *New Results in Linear Filtering and Prediction*. Trans. of the ASME, J. of Basic Engr. 83:95-108.
- Kashyap, R. L. 1977. *A Bayesian Comparison of Different Classes of Models using Empirical Data*. IEEE Transactions on Automatic Control AC-22:715-727.
- Kashyap, R. L. 1980a. *Inconsistency of the AIC Rule for Estimating the Order of Autoregressive Models*. IEEE Transactions on Automatic Control AC-25:996-998.
- Kashyap, R. L. 1980b. *Random Field Models on Torus Lattices for Finite Images*. Pp. 1103-1105. Proceedings on the 5th International Conference on Pattern Recognition, Miami Florida. Los Alamitos, California.
- Kashyap, R. L. and R. Chellappa. 1981. *Decision Rules for Choice of Neighbors in Random Field Models of Images*. Computer Graphics and Image Processing 15(15):301-317.
- Kashyap, R. L. and R. Chellappa. 1983. *Estimation and Choice of Neighbors in Spatial-Interaction Models of Images*. IEEE Transactions On Information Theory IT-29(1):60-72.
- Kitamura, K., J. M. Tobis and J. Sklansky. 1988. *Estimating the 3-D Skeleton and Transverse areas of Coronary Arteries form Biplane Angiograms*. IEEE Transactions on Medical Imaging. 7(3):173-187.

- Laub, A. J. 1985. *Numerical Linear Algebra Aspects of Control Design Computations*. IEEE Transactions on Automatic Control AC-30(2):97-108.
- Mendel, J. M. 1987. *Lessons in Digital Estimation Theory*. Prentice-Hall Inc., Englewood Cliffs, New Jersey.
- Ord, K. 1975. *Estimation Methods for Models of Spatial Interaction*. J. Am. Statist. Assoc. 70:120-126.
- Osterbrock, D. E. 1974. *Astrophysics of Gaseous Nebulae*. Freeman, San Francisco.
- Pottasch, S. R. 1984. *Planetary Nebulae: A Study of Late Stages of Stellar Evolution*. Kluwer Academic Publishers, Hingham, Massachusetts.
- Press, W. H., B. P. Flannery, S. A. Teukolsky and W. T. Vetterling. 1986. *Numerical Recipes - The Art of Scientific Computing*. Cambridge University Press, Cambridge.
- Rohlfs, K. 1986. *Tools of Radio Astronomy*. Springer-Verlag, Berlin.
- Serra, J. 1982. *Image Analysis and Mathematical Morphology*. Academic Press, London.
- Thompson, A. R., J. M. Moran and G. W. Swenson. 1986. *Interferometry and Synthesis in Radio Astronomy*. John Wiley and Sons, New York.
- Whittle, P. 1954. *On Stationary Processes in the Plane*. Biometrika 41:434-449.

12. APPENDIX A

Observed Flux Density Values at 2 cm.

		Column number													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
2	-1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
5	0	0	0	0	0	1	0	0	0	-1	0	0	0	0	2
6	0	0	0	0	1	0	0	0	0	0	0	0	0	3	15
7	0	0	0	1	1	0	0	1	2	1	1	5	22	60	
8	1	0	0	1	1	0	0	1	1	3	11	32	78	152	
9	0	-1	0	2	1	1	2	0	3	17	49	107	193	287	
10	-1	0	1	1	1	2	2	2	16	58	138	255	380	458	
11	0	0	0	0	2	3	1	8	48	146	307	486	611	644	
12	1	1	0	1	2	1	4	33	127	312	539	706	765	734	
13	1	0	0	1	1	1	15	86	268	537	755	833	787	673	
14	0	0	0	1	0	4	42	178	449	733	870	856	734	557	
15	1	1	1	0	0	15	90	302	622	858	891	781	613	448	
16	1	1	0	0	3	34	158	429	735	881	841	682	510	381	
17	0	0	-1	0	9	64	238	534	779	829	720	555	422	333	
18	0	-1	-1	1	21	106	312	574	735	737	610	465	362	293	
19	0	0	0	4	38	148	357	563	651	619	519	422	342	282	
20	0	0	0	11	58	174	352	510	574	545	478	406	336	300	
21	1	0	3	19	71	179	326	455	515	508	477	417	350	333	
22	0	1	6	24	72	158	274	399	495	521	500	454	409	383	
23	0	1	8	27	67	127	203	305	433	524	534	518	508	481	
24	0	0	5	24	58	100	144	205	315	448	539	583	596	590	
25	-1	0	3	18	46	76	106	145	214	324	452	549	595	620	
26	-1	0	1	11	30	55	83	117	154	206	280	356	409	448	
27	0	0	1	5	15	31	55	80	99	116	138	170	201	228	
28	0	1	3	4	5	10	21	33	42	51	62	79	98	112	
29	0	0	2	2	1	1	2	5	10	15	22	32	45	53	
30	0	0	0	0	0	0	-1	-1	0	2	4	8	13	16	
31	0	0	-1	0	0	0	0	-1	-1	-1	0	0	1	2	
32	0	0	0	0	0	-1	0	0	-1	-1	-1	0	0	1	

		Column number													
		15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	0	0	0	0	0	0	0	-1	0	0	-1	0	0	0	0
2	0	0	-1	-1	-1	-1	0	0	0	0	0	0	0	0	0
3	1	0	0	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	0
4	2	2	3	2	0	0	-1	-1	-1	-1	-1	-1	-1	-1	0
5	7	12	16	13	7	3	0	0	0	0	0	0	0	0	0
6	35	57	67	58	38	20	8	2	0	0	0	0	0	0	0
7	115	164	181	159	116	72	41	22	12	6	3	1	0	0	0
8	234	294	313	289	237	177	124	84	58	39	22	11	5	0	0
9	351	387	406	402	366	307	243	185	139	97	57	30	16	0	0
10	473	472	477	484	465	405	330	266	213	154	93	50	28	0	0
11	601	542	516	526	519	458	370	295	242	185	117	64	39	0	0
12	631	520	480	495	505	482	421	332	257	196	129	75	49	0	0
13	527	417	390	412	445	473	461	392	299	213	137	84	51	0	0
14	404	316	292	323	391	456	484	449	354	242	155	96	52	0	0
15	333	272	255	286	354	447	528	518	406	271	169	97	47	0	0
16	291	248	253	287	356	483	597	588	454	288	160	77	31	0	0
17	273	243	246	289	402	580	704	667	480	262	116	45	15	0	0
18	264	259	263	320	487	705	831	732	451	193	63	18	4	0	0
19	264	275	301	394	605	837	907	696	351	115	26	4	1	0	0
20	292	299	353	496	726	928	890	574	235	59	8	1	1	0	0
21	335	347	436	631	855	946	759	403	134	26	3	1	1	0	0
22	380	426	545	749	912	837	535	227	61	9	2	1	0	0	0
23	471	531	659	815	827	602	300	100	21	2	1	1	1	0	0
24	584	621	708	746	601	336	129	32	5	1	1	0	0	0	0
25	625	637	636	532	328	141	40	6	1	1	0	0	0	1	0
26	471	472	416	278	131	43	9	1	1	0	0	0	0	1	0
27	249	245	191	106	40	11	3	2	2	0	0	0	0	0	0
28	115	101	67	31	10	2	1	2	1	0	0	0	0	0	0
29	48	34	17	7	2	0	0	1	1	0	0	0	0	0	0
30	13	7	3	0	0	0	0	0	0	0	0	0	0	0	0
31	0	-1	-1	-1	-1	0	0	0	0	0	0	0	0	1	0
32	0	0	0	-1	0	0	0	0	0	-1	-1	0	0	0	0

Beam width in minor axis direction = 1.18 arcsec

Beam width in major axis direction = 1.18 arcsec

Map scale: 1000 = 0.26316 Jy/beam.

Observed Flux Density Values at 6 cm.

		Column number													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Row number	4	3	3	3	3	3	2	2	2	1	2	2	2	3
2		3	3	3	4	3	3	2	2	2	2	1	1	2	3
3		3	4	4	3	2	2	2	1	2	2	1	1	2	3
4		4	4	3	3	3	3	2	1	1	1	1	2	3	3
5		4	4	4	4	3	3	2	2	1	1	2	2	3	5
6		3	4	4	5	4	3	3	2	1	2	3	3	6	16
7		4	4	3	4	4	3	2	1	1	3	4	9	26	64
8		4	4	3	3	4	3	1	1	2	5	13	37	89	175
9		4	4	4	4	4	3	2	3	6	19	51	116	217	339
10		3	4	4	3	3	3	4	6	17	57	145	274	421	534
11		4	4	4	3	3	4	4	11	48	148	319	515	671	739
12		4	4	3	3	4	5	7	31	124	314	562	769	863	858
13		4	4	4	5	5	6	18	84	263	543	801	931	921	825
14		3	4	5	5	5	10	45	176	448	770	959	969	877	715
15		3	4	4	4	6	21	94	303	634	915	999	922	774	593
16		4	5	4	4	10	42	164	442	782	973	957	831	662	506
17		4	5	4	6	17	74	253	563	848	947	869	710	557	456
18		4	5	5	8	30	120	345	646	646	862	758	608	489	417
19		4	5	6	14	53	178	411	658	786	766	663	550	461	392
20		5	6	8	24	82	224	440	628	704	684	619	539	458	404
21		5	6	12	36	104	239	421	576	646	643	604	540	474	444
22		5	6	16	44	110	228	372	511	613	651	627	575	532	504
23		5	8	18	45	103	191	294	417	548	641	663	642	619	597
24		5	7	15	40	92	154	214	299	425	565	663	704	718	713
25		5	6	12	32	72	120	161	212	306	441	579	672	722	744
26		5	7	11	23	49	89	130	173	230	300	390	477	532	568
27		5	6	9	16	31	56	88	125	156	177	211	253	288	322
28		5	7	8	11	16	24	39	59	74	86	104	126	154	174
29		6	7	7	7	8	8	11	16	22	32	45	60	80	89
30		6	6	5	5	4	3	3	4	6	10	14	19	27	31
31		4	4	4	3	3	3	3	3	3	4	5	6	7	7
32		3	3	3	2	3	4	3	3	3	4	4	4	4	4

		Column number													
		15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	3	4	4	3	3	3	3	2	3	2	2	2	2	2	2
2	3	4	4	3	3	3	3	3	3	2	2	2	3	2	
3	3	4	4	4	4	4	3	4	3	2	2	3	3	3	
4	4	5	6	6	5	4	4	4	3	3	3	3	4	3	
5	9	15	20	21	14	9	6	5	5	4	3	4	3	2	
6	35	58	75	69	46	27	15	9	7	5	5	5	3	2	
7	124	183	209	186	137	88	50	29	19	13	10	7	5	4	
8	281	360	381	350	285	209	146	100	68	47	31	19	12	9	
9	438	492	507	495	447	369	296	231	173	124	77	44	28	19	
10	576	587	598	602	577	506	423	347	281	207	129	73	44	29	
11	717	672	654	654	640	577	481	394	330	252	163	96	60	39	
12	777	674	632	644	647	607	523	431	352	270	181	110	74	47	
13	690	565	527	555	590	606	578	496	391	291	194	120	77	43	
14	548	443	420	457	523	590	615	563	450	325	215	137	78	34	
15	452	377	360	400	479	577	657	640	518	360	232	141	70	25	
16	408	357	352	392	478	610	721	709	566	376	223	116	47	14	
17	383	346	353	391	508	687	809	774	583	343	167	68	22	6	
18	364	351	368	428	590	796	910	806	524	248	91	28	8	3	
19	368	381	401	498	700	906	960	742	388	137	36	9	3	2	
20	402	413	460	600	810	979	919	602	255	67	12	3	2	2	
21	443	461	551	722	913	989	788	425	147	31	6	4	3	2	
22	502	545	656	830	963	881	571	248	69	12	4	4	4	3	
23	599	649	761	899	886	642	327	114	26	6	4	3	3	3	
24	707	740	817	833	659	371	146	39	9	5	4	3	3	3	
25	746	760	746	615	378	164	50	12	5	5	4	4	4	4	
26	593	594	517	348	166	55	15	7	6	4	4	4	4	3	
27	347	337	263	149	58	17	8	7	6	5	4	4	3	3	
28	178	155	105	51	18	8	7	6	6	6	4	3	3	2	
29	83	62	36	17	9	7	6	6	6	5	4	4	3	2	
30	28	19	11	7	7	6	6	7	6	4	4	4	3	3	
31	7	5	5	5	6	6	7	7	5	4	4	4	3	3	
32	4	5	5	6	6	6	6	5	5	4	4	3	3	3	

Beam Width in minor axis direction = 1.34 arcsec

Beam Width in major axis direction = 1.15 arcsec

Map scale: 1000=0.23244 Jy/beam

Observed Flux Density Values at 20 cm.

		Column number													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Row number	-6	-8	-7	-5	-4	-3	-3	-2	-2	-3	-5	-6	-8	-5
2		-7	-7	-5	-5	-3	0	0	-1	-1	-3	-4	-5	-5	-3
3		-5	-3	0	-2	-1	1	1	0	-1	-3	-2	-1	-1	-3
4		-3	1	3	0	-4	-5	-4	-2	-3	-6	-3	0	0	-1
5		-1	1	0	-3	-8	-12	-9	-5	-4	-4	-2	0	3	9
6		-1	-2	-4	-5	-6	-9	-9	-7	-4	-1	0	0	11	40
7		-1	-2	-3	-3	-4	-4	-6	-6	-2	1	4	16	57	143
8		0	-1	0	-2	-4	-2	-3	-2	1	10	35	88	186	343
9		0	-1	-1	-4	-5	-3	-3	0	13	54	136	254	408	580
10		1	-1	-1	-3	-4	-2	1	9	45	145	304	471	625	746
11		-1	-2	-2	-1	0	1	4	22	99	273	488	655	754	814
12		-4	-6	-4	1	4	2	5	42	174	411	635	759	820	862
13		-6	-5	-4	0	3	1	15	96	291	553	739	809	849	891
14		-4	-2	-3	-2	0	3	43	187	445	686	811	850	871	894
15		-3	-2	-2	-2	0	15	99	306	580	761	828	857	868	879
16		-3	-2	1	1	3	43	188	446	688	805	848	871	861	850
17		-3	-3	-1	1	15	90	292	564	750	828	853	856	852	846
18		-2	-3	-3	4	42	168	415	665	789	839	853	850	847	826
19		-1	-1	-2	16	91	274	539	737	814	843	848	846	842	824
20		0	-1	3	43	166	397	638	766	822	846	850	843	842	830
21		-2	-1	15	84	246	487	686	786	829	838	850	847	850	846
22		-4	1	33	126	305	523	681	768	803	819	846	848	843	851
23		-3	5	41	142	320	520	658	721	763	806	826	842	853	858
24		-2	6	34	122	289	474	590	641	704	775	802	828	837	848
25		-3	6	26	94	244	414	520	586	650	717	760	790	805	816
26		-2	6	21	70	185	323	439	539	598	643	670	696	733	746
27		0	5	19	55	124	205	297	396	451	497	535	570	620	647
28		3	7	19	41	68	96	138	192	230	279	342	403	477	521
29		7	10	17	23	26	30	40	55	70	101	152	209	275	316
30		4	7	10	9	9	10	11	11	10	18	41	69	102	122
31		-1	2	3	2	2	2	2	2	-2	-2	7	15	23	28
32		-4	-1	-1	0	0	-2	-3	-2	-1	1	4	4	4	5

		Column number													
		15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	Row number	-5	-7	-6	-4	-6	-5	-3	-3	-2	-2	-2	-4	-3	2
2		-4	-6	-5	-3	-5	-4	-4	-4	-1	0	-2	-3	-2	2
3		-5	-5	-4	-5	-6	-5	-6	-5	-1	1	2	2	0	0
4		3	10	9	3	-2	-4	-4	-3	-1	1	3	2	-1	-2
5		27	53	66	58	34	14	6	3	1	2	1	-2	-1	1
6		97	172	220	208	145	83	43	19	7	4	1	-2	1	3
7		271	400	473	453	358	247	150	83	50	36	26	19	15	13
8		521	648	714	703	613	486	355	259	195	155	118	79	54	43
9		720	805	856	846	786	707	608	528	443	362	270	177	121	93
10		822	875	908	895	863	833	775	737	672	565	419	283	204	144
11		865	901	912	923	911	886	843	810	766	666	509	366	276	184
12		895	909	918	945	943	935	906	848	794	702	549	413	320	205
13		902	905	914	937	957	964	937	887	835	736	583	443	323	192
14		884	875	877	902	950	980	982	940	867	759	617	464	296	146
15		867	852	848	879	923	957	999	978	893	783	631	446	247	95
16		840	835	842	863	904	947	981	966	875	741	555	339	155	47
17		835	825	837	846	888	944	946	917	804	613	392	189	63	14
18		818	820	837	856	878	903	874	796	627	398	202	73	15	2
19		830	834	835	866	881	877	810	647	408	192	70	18	1	1
20		824	834	840	866	868	828	726	503	242	78	18	5	2	1
21		838	852	868	873	858	787	632	372	136	29	4	3	2	0
22		857	860	868	855	816	701	491	240	71	14	1	0	1	0
23		859	863	870	835	736	544	307	119	30	8	2	0	1	0
24		855	848	834	755	587	354	151	41	7	4	3	3	3	0
25		828	816	769	626	403	189	56	7	0	3	5	5	2	-1
26		762	741	646	451	231	84	19	2	0	2	3	2	-1	-2
27		658	610	465	264	104	29	6	2	1	0	0	-2	-2	0
28		500	416	271	123	37	8	4	3	1	-1	-2	-2	-1	-1
29		288	209	114	41	9	3	3	1	-2	-5	-4	-2	-1	-3
30		107	69	31	8	1	0	0	1	-3	-6	-4	-1	-2	-4
31		23	12	3	-1	-1	-1	-1	0	-3	-7	-4	-3	-4	-5
32		3	0	-3	-3	-1	0	0	-2	-5	-6	-4	-4	-4	-3

Beam Width in minor axis direction = 1.24 arcsec

Beam Width in major axis direction = 1.21 arcsec

Map scale: 1000 = 0.039684 Jy/beam.

13. APPENDIX B

Following is a listing of some of the most important programs used in this dissertation.

```

C*****
C
C NAME: BLUE.FOR
C
C PURPOSE: PARAMETER ESTIMATION FROM SIMULATED DATA OR
C          OBSERVED DATA FOR NGC 7027 USING LINEAR
C          ESTIMATION THEORY.
C
C REFERENCES:
C
C
C AUTHOR: W C VENTER
C
C DATE WRITTEN: 2/13/88
C
C LAST REVISION: 10/17/88
C
C*****
C
C          PROGRAM BLUE
C
C*****
C          *
C SYMBOL DECLARATIONS *
C          *
C*****
C
C          INTEGER*4 I, J, II, JJ, KK, IN
C          REAL*8 DAT02(32,28), DAT06(32,28), DAT20(32,28)
C          REAL*8 DD02, DD06, DD20, DUM, DUM1, DUM2, DUM3, DUM4
C          REAL*8 PI, BOLTZ, THETAX, THETAY, A02, A06, A20
C          REAL*8 K102, K106, K120, K202, K206, K220
C          REAL*8 TONOM, T1NOM, T6NOM
C          REAL*8 S02, S02NOM, S06, S06NOM, S20, S20NOM
C          REAL*8 L102, L202, L302, L106, L206, L306
C          REAL*8 L120, L220, L320, L302A, L306A, L320A
C          REAL*8 H(3,3), HT(3,3), TEMP1(3,3), TEMP2(3,3), K(3,3)
C          REAL*8 R(3,3), RINV(3,3)
C          REAL*8 TOEST(32,28), T1EST(32,28), T6EST(32,28),
TEMP(32,28)
C          REAL*8 CF02, CF06, CF20, B
C          REAL*8 A02T6, A06T6, A20T6, E02, E06, E20

```

```

C
C*****
C          *
C OPEN FILES *
C          *
C*****
C
C#####
C
C SIMULATED VALUES FOR FLUX DENSITY AT 2 CM IS STORED      #
C SEQUENTIALLY IN FILE CM02SI.SEQ                            #
C SIMULATED VALUES FOR FLUX DENSITY AT 6 CM IS STORED      #
C SEQUENTIALLY IN FILE CM06SI.SEQ                            #
C SIMULATED VALUES FOR FLUX DENSITY AT 20 CM IS STORED     #
C SEQUENTIALLY IN FILE CM20SI.SEQ                            #
C
C OBSERVED VALUES FOR FLUX DENSITY AT 2 CM IS STORED      #
C SEQUENTIALLY IN FILE CM02SQ.SEQ                            #
C OBSERVED VALUES FOR FLUX DENSITY AT 6 CM IS STORED      #
C SEQUENTIALLY IN FILE CM06SQ.SEQ                            #
C OBSERVED VALUES FOR FLUX DENSITY AT 20 CM IS STORED     #
C SEQUENTIALLY IN FILE CM20SQ.SEQ                            #
C
C ESTIMATED VALUES FOR PARAMETER T0 IS STORED IN FILE      #
C T0EST.MAP IN MAP FORMAT                                    #
C ESTIMATED VALUES FOR PARAMETER T1 IS STORED IN FILE      #
C T1EST.MAP IN MAP FORMAT                                    #
C ESTIMATED VALUES FOR PARAMETER T6 IS STORED IN FILE      #
C T6EST.MAP IN MAP FORMAT                                    #
C
C ESTIMATED VALUES FOR PARAMETER T0 IS STORED IN FILE      #
C T0EST.AGR IN AGRAPH FORMAT                                  #
C ESTIMATED VALUES FOR PARAMETER T1 IS STORED IN FILE      #
C T1EST.AGR IN AGRAPH FORMAT                                  #
C ESTIMATED VALUES FOR PARAMETER T6 IS STORED IN FILE      #
C T6EST.AGR IN AGRAPH FORMAT                                  #
C
C#####
C
      WRITE(6,110)
      110 FORMAT(1X,'USE OBSERVED DATA? - 1 FOR YES; 0 FOR NO')
      READ(6,*) IN
C

```

```

OPEN(13, FILE='NOISE', ACCESS='SEQUENTIAL', STATUS='OLD')
OPEN(14, FILE='DATOUT', STATUS='UNKNOWN')
OPEN(15, FILE='TOEST.MAP', STATUS='UNKNOWN')
OPEN(16, FILE='T1EST.MAP', STATUS='UNKNOWN')
OPEN(17, FILE='T6EST.MAP', STATUS='UNKNOWN')
OPEN(19, FILE='TOEST.AGR', STATUS='UNKNOWN')
OPEN(20, FILE='T1EST.AGR', STATUS='UNKNOWN')
OPEN(21, FILE='T6EST.AGR', STATUS='UNKNOWN')
IF(IN.EQ.1)GO TO 120
  OPEN(10, FILE='CM02SI.SEQ', ACCESS='SEQUENTIAL',
STATUS='OLD')
  OPEN(11, FILE='CM06SI.SEQ', ACCESS='SEQUENTIAL',
STATUS='OLD')
  OPEN(12, FILE='CM20SI.SEQ', ACCESS='SEQUENTIAL',
STATUS='OLD')
  GO TO 130
  120 OPEN(10, FILE='CM02SQ', ACCESS='SEQUENTIAL', STATUS='OLD')
  OPEN(11, FILE='CM06SQ', ACCESS='SEQUENTIAL', STATUS='OLD')
  OPEN(12, FILE='CM20SQ', ACCESS='SEQUENTIAL', STATUS='OLD')
C
C*****
C      *
C PROGRAM *
C      *
C*****
C
C#####
C      #
C READ DATA IN #
C      #
C#####
C
  130 READ(10,*) DAT02
      READ(11,*) DAT06
      READ(12,*) DAT20
      READ(13,*) R
C
C#####
C      #
C CONVERTS INPUT DATA TO JENSKYS #
C      #
C#####
C

```

```

      IF(IN.NE.1)GO TO 140
      CF02=0.26316*1E-26/1000.0
      CF06=0.23244*1E-26/1000.0
      CF20=0.039684*1E-26/1000.0
      GO TO 150
C
140  CF02=1E-26
      CF06=1E-26
      CF20=1E-26
C
150  DO 710 I=1,32
      DO 720 J=1,28
          DAT02(I,J)=DAT02(I,J)*CF02
          DAT06(I,J)=DAT06(I,J)*CF06
720   DAT20(I,J)=DAT20(I,J)*CF20
710   CONTINUE
C
C#####
C      #
C  CONSTANTS #
C      #
C#####
C
      PI=3.14159265358979
      BOLTZ=1.381E-23
      THETAX=6.01169E-6
      THETAY=5.86625E-6
      A02=0.09528
      A06=1.0
      A20=12.544
C
C      WRITE(6,700)
C 700  FORMAT(1X,'VALUE OF B=?')
C      READ(6,*)B
C
      K102=(PI*BOLTZ*THETAX*THETAY)/(2.0*LOG(2.0)*(0.02)**2)
      K106=(PI*BOLTZ*THETAX*THETAY)/(2.0*LOG(2.0)*(0.06)**2)
      K120=(PI*BOLTZ*THETAX*THETAY)/(2.0*LOG(2.0)*(0.20)**2)
C      K202=K102/(1.0-(B/A02))
C      K206=K106/(1.0-(B/A06))
C      K220=K120/(1.0-(B/A20))
C
      WRITE(14,101)

```

```

101 FORMAT(1X,' K102 K106 K120 K202',6X,
Q'K206 K220')
WRITE(14,100)K102, K106, K120
100 FORMAT(1X,3(E10.3E3),/)

```

C

C#####

C

C CALCULATE NOMINAL VALUE FOR EACH DATA POINT BY #

C SOLVING THREE SIMULTANEOUS EQUATIONS DIRECTLY. #

C

C#####

C

```

DO 210 I=6,26
DO 220 J=6,25
T6EST(I,J)=0.2
820 ET6=EXP(-1.0*T6EST(I,J))
A02T6=A02*T6EST(I,J)
A20T6=A20*T6EST(I,J)
E02=EXP(-1.0*A02T6)
E20=EXP(-1.0*A20T6)

C
DUM=(1.0/A20T6)-E20
DUM1=(1.0/A02T6)-((1.0/A02T6)*E02)-E02
DUM2=(DAT02(I,J)/K102)*DUM/DUM1
DUM2=(DAT20(I,J)/K120)-DUM2
DUM3=1.0-(1.0/A02T6)+(1.0/A02T6)*E02
DUM4=1.0-(1.0/A20T6)-((DUM3/DUM1)*DUM)
T1EST(I,J)=DUM2/DUM4
TOEST(I,J)=((DAT02(I,J)/K102)-(T1EST(I,J)*DUM3))/DUM1

C
DUM=TOEST(I,J)*(1.0-ET6)
DUM1=(TOEST(I,J)-T1EST(I,J))/T6EST(I,J)
DUM=DUM-DUM1*(T6EST(I,J)-1.0+ET6)
DUM=DUM-(DAT06(I,J)/K106)
IF(DUM.LT.-10.0)GO TO 830
IF(DUM.GT.10.0)GO TO 830
GO TO 810
830 T6EST(I,J)=T6EST(I,J)+0.001
IF(T6EST(I,J).GT.5.0)GO TO 810
GO TO 820
810 WRITE(6,850)TOEST(I,J),T1EST(I,J),T6EST(I,J),I,J
850 FORMAT(1X,3F15.3,2I4)

```

C

```

C#####
C                                     #
C TAKE OUT VALUES WITH T6EST>2.99 #
C                                     #
C#####
C
      IF (T6EST(I,J).LT.4.99) GO TO 870
      T6EST(I,J)=0.0
      T0EST(I,J)=0.0
      T1EST(I,J)=0.0
      TEMP(I,J)=0.0
      GO TO 220

C
C#####
C                                     #
C PARAMETER ESTIMATION              #
C                                     #
C#####
C
      870      DO 560 KK=1,6
C
      TONOM=T0EST(I,J)
      T1NOM=T1EST(I,J)
      T6NOM=T6EST(I,J)

C
      A02T6=A02*T6NOM
      A06T6=A06*T6NOM
      A20T6=A20*T6NOM
      E02=EXP(-A02T6)
      E06=EXP(-A06T6)
      E20=EXP(-A20T6)

C
      550      S02=K102*TONOM*((1.0/A02T6)*(1.0-E02)-E02)
      S02NOM=K102*T1NOM*(1.0-((1.0/A02T6)*(1.0-E02)))
      S02NOM=S02NOM+S02
      S06=K106*TONOM*((1.0/A06T6)*(1.0-E06)-E06)
      S06NOM=K106*T1NOM*(1.0-((1.0/A06T6)*(1.0-E06)))
      S06NOM=S06NOM+S06
      S20=K120*TONOM*((1.0/A20T6)*(1.0-E20)-E20)
      S20NOM=K120*T1NOM*(1.0-((1.0/A20T6)*(1.0-E20)))
      S20NOM=S20NOM+S20

C
      S02=((S02NOM-DAT02(I,J))/DAT02(I,J))*100.0

```

S06=((S06NOM-DAT06(I,J))/DAT06(I,J))*100.0
 S20=((S20NOM-DAT20(I,J))/DAT20(I,J))*100.0

C

S02NOM=S02NOM/1E-26
 S06NOM=S06NOM/1E-26
 S20NOM=S20NOM/1E-26

C

C WRITE(14,110)TONOM, T1NOM, T6NOM, I, J
 C 110 FORMAT(1X,3(E14.7E3),2I5)
 WRITE(6,112)S02, S06, S20
 112 FORMAT(1X,3F9.2)

C

C#####
 C #
 C CORRECT DATA UNITS #
 C #
 C#####

C

DD02=DAT02(I,J)/1E-26
 DD06=DAT06(I,J)/1E-26
 DD20=DAT20(I,J)/1E-26

C

DD02=DD02-S02NOM
 DD06=DD06-S06NOM
 DD20=DD20-S20NOM

C

C#####
 C #
 C H MATRIX #
 C #
 C#####

C

L102=K102*((1.0/A02T6)*(1.0-E02)-E02)
 L202=K102*(1.0-((1.0/A02T6)*(1.0-E02)))
 L302=(1.0/(A02T6*T6NOM))*(E02-1.0)
 L302=(L302+(E02*((1.0/T6NOM)+A02)))*TONOM
 L302A=(1.0/(A02T6*T6NOM))*(1.0-E02)
 L302A=(L302A-(1.0/T6NOM)*E02)*T1NOM
 L302=(L302+L302A)*K102
 L106=K106*((1.0/A06T6)*(1.0-E06)-E06)
 L206=K106*(1.0-((1.0/A06T6)*(1.0-E06)))
 L306=(1.0/(A06T6*T6NOM))*(E06-1.0)
 L306=(L306+(E06*((1.0/T6NOM)+A06)))*TONOM

```

L306A=(1.0/(A06T6*T6NOM))* (1.0-E06)
L306A=(L306A-(1.0/T6NOM)*E06)*T1NOM
L306=(L306+L306A)*K106
L120=K120*((1.0/A20T6)*(1.0-E20)-E20)
L220=K120*(1.0-((1.0/A20T6)*(1.0-E20)))
L320=(1.0/(A20T6*T6NOM))*(E20-1.0)
L320=(L320+(E20*((1.0/T6NOM)+A20)))*TONOM
L320A=(1.0/(A20T6*T6NOM))*(1.0-E20)
L320A=(L320A-(1.0/T6NOM)*E20)*T1NOM
L320=(L320+L320A)*K120

```

C

```

H(1,1) = L120/1E-26
H(1,2) = L220/1E-26
H(1,3) = L320/1E-26
H(2,1) = L106/1E-26
H(2,2) = L206/1E-26
H(2,3) = L306/1E-26
H(3,1) = L102/1E-26
H(3,2) = L202/1E-26
H(3,3) = L302/1E-26

```

C

```

C WRITE(14,151)
C 151 FORMAT(1X,'H-MATRIX')
C DO 160 NI=1,3
C WRITE(14,170) (H(NI,NJ),NJ=1,3)
C 170 FORMAT(1X,3E16.7E3)
C 160 CONTINUE

```

C

```

CALL TRANSP(H, 3, 3, HT)
CALL MATINV(R, RINV, 3)
CALL MUL33(TEMP1, RINV, H)
CALL MUL33(TEMP2, HT, TEMP1)
CALL MATINV(TEMP2, TEMP1, 3)
CALL MUL33(TEMP2, HT, RINV)
CALL MUL33(K, TEMP1, TEMP2)

```

C

```

C WRITE(14,180)
C 180 FORMAT(1X,'K-MATRIX')
C DO 190 NI=1,3
C WRITE(14,200) (K(NI,NJ),NJ=1,3)
C 200 FORMAT(1X,3E16.7E3)
C 190 CONTINUE

```

C

```

T0EST(I, J) = K(1, 1) * DD20 + K(1, 2) * DD06 + K(1, 3) * DD02
T1EST(I, J) = K(2, 1) * DD20 + K(2, 2) * DD06 + K(2, 3) * DD02
T6EST(I, J) = K(3, 1) * DD20 + K(3, 2) * DD06 + K(3, 3) * DD02
T0EST(I, J) = T0EST(I, J) + T0NOM
T1EST(I, J) = T1EST(I, J) + T1NOM
T6EST(I, J) = T6EST(I, J) + T6NOM

```

C

```

GO TO 570
IF (T6EST(I, J) .LT. 0.0) GO TO 530
IF (T6EST(I, J) .GT. 5.0) GO TO 530
GO TO 570
530 IF (T6NOM .GT. 0.25) GO TO 540
T6NOM = 0.6
GO TO 570
540 T6NOM = 0.2
GO TO 550
570 WRITE(6, 1110) T0EST(I, J), T1EST(I, J), T6EST(I, J), I, J
1110 FORMAT(1X, 3(E10.3E3), 2I5)
560 CONTINUE

```

C

```

IF (T6EST(I, J) .GE. 0.2 .AND. T6EST(I, J) .LT. 3.99) GO TO 220
T6EST(I, J) = 0.0
T0EST(I, J) = 0.0
T1EST(I, J) = 0.0
TEMP(I, J) = 0.0
220 CONTINUE
210 CONTINUE

```

C

```

C#####
C                                     #
C WRITE OUT FINAL ESTIMATED VALUES #
C                                     #
C#####

```

C

```

WRITE(15, 230)
230 FORMAT(1X, 'T0EST', /)
WRITE(16, 240)
240 FORMAT(1X, 'T1EST', /)
WRITE(17, 250)
250 FORMAT(1X, 'T6EST', /)
DO 260 I=1, 32
WRITE(15, 270) (T0EST(I, J), J=1, 7)
WRITE(16, 270) (T1EST(I, J), J=1, 7)

```

```

        WRITE(17,270) (T6EST(I,J),J=1,7)
270    FORMAT(1X,7E10.3E2)
260    CONTINUE
        WRITE(15,280)
        WRITE(16,280)
        WRITE(17,280)
280    FORMAT(1X)
        DO 290 I=1,32
            WRITE(15,270) (TOEST(I,J),J=8,14)
            WRITE(16,270) (T1EST(I,J),J=8,14)
290    WRITE(17,270) (T6EST(I,J),J=8,14)
        WRITE(15,280)
        WRITE(16,280)
        WRITE(17,280)
        DO 300 I=1,32
            WRITE(15,270) (TOEST(I,J),J=15,21)
            WRITE(16,270) (T1EST(I,J),J=15,21)
300    WRITE(17,270) (T6EST(I,J),J=15,21)
        WRITE(15,280)
        WRITE(16,280)
        WRITE(17,280)
        DO 310 I=1,32
            WRITE(15,270) (TOEST(I,J),J=22,28)
            WRITE(16,270) (T1EST(I,J),J=22,28)
310    WRITE(17,270) (T6EST(I,J),J=22,28)
C
C#####
C                                     #
C WRITE OUT FINAL ESTIMATED VALUES #
C IN AGRAF FORMAT.                   #
C                                     #
C#####
C
        WRITE(19,2800) 28,32,0,0
        WRITE(20,2800) 28,32,0,0
        WRITE(21,2800) 28,32,0,0
2800  FORMAT(1X,4I4)
        DO 800 I=32,1,-1
            WRITE(19,270) (TOEST(I,J),J=1,7)
            WRITE(19,270) (TOEST(I,J),J=8,14)
            WRITE(19,270) (TOEST(I,J),J=15,21)
            WRITE(19,270) (TOEST(I,J),J=22,28)
            WRITE(20,270) (T1EST(I,J),J=1,7)

```

```

      WRITE(20,270) (T1EST(I,J),J=8,14)
      WRITE(20,270) (T1EST(I,J),J=15,21)
      WRITE(20,270) (T1EST(I,J),J=22,28)
      WRITE(21,270) (T6EST(I,J),J=1,7)
      WRITE(21,270) (T6EST(I,J),J=8,14)
      WRITE(21,270) (T6EST(I,J),J=15,21)
800   WRITE(21,270) (T6EST(I,J),J=22,28)
C
      STOP
      END
C
C*****
C
C NAME: SUBROUTINE - MUL33.FOR
C
C PURPOSE: MULTIPLIES TWO 3X3 MATRICES
C
C
C
C REFERENCES:
C
C
C
C WRITER: W C VENTER
C
C DATE WRITTEN: 2/25/88
C
C LAST REVISION:
C
C*****
C
      SUBROUTINE MUL33 (MUL, A, B)
C
C*****
C
C SYMBOL DECLARATIONS *
C
C*****
C
      REAL*8 MUL(3,3), A(3,3), B(3,3)
C
C*****
C
      *
```

```

C PROGRAM *
C *
C*****
C
  MUL(1,1)=-A(1,1)*B(1,1)+A(1,2)*B(2,1)+A(1,3)*B(3,1)
  MUL(1,2)=-A(1,1)*B(1,2)+A(1,2)*B(2,2)+A(1,3)*B(3,2)
  MUL(1,3)=-A(1,1)*B(1,3)+A(1,2)*B(2,3)+A(1,3)*B(3,3)
  MUL(2,1)=-A(2,1)*B(1,1)+A(2,2)*B(2,1)+A(2,3)*B(3,1)
  MUL(2,2)=-A(2,1)*B(1,2)+A(2,2)*B(2,2)+A(2,3)*B(3,2)
  MUL(2,3)=-A(2,1)*B(1,3)+A(2,2)*B(2,3)+A(2,3)*B(3,3)
  MUL(3,1)=-A(3,1)*B(1,1)+A(3,2)*B(2,1)+A(3,3)*B(3,1)
  MUL(3,2)=-A(3,1)*B(1,2)+A(3,2)*B(2,2)+A(3,3)*B(3,2)
  MUL(3,3)=-A(3,1)*B(1,3)+A(3,2)*B(2,3)+A(3,3)*B(3,3)
C
  RETURN
  END
C
C*****
C NAME: SUBROUTINE - TRANSP.FOR
C
C PURPOSE: CALCULATES THE TRANSPOSE OF A MATRIX - MAXIMUM
C           SIZE (100,100)
C
C REFERENCES:
C
C WRITER: W C VENTER
C
C DATE WRITTEN: 2/24/88
C
C LAST REVISION:
C
C*****
C           SUBROUTINE TRANSP (MIN, N1, M1, MTP)
C*****
C *
C SYMBOL DECLARATIONS *
C *

```



```

C                                     *
C*****
C
  INTEGER*4 NMAX
  REAL*8 TINY, D
  PARAMETER (NMAX=100, TINY=1.0E-20)
  REAL*8 A(NP, NP), AAMAX, VV(NMAX), SUM, DUM
  INTEGER*4 INDX(N)
  INTEGER*4 I, J, K, IMAX

C
  D=1.
  DO 12 I=1, N
    AAMAX=0.
    DO 11 J=1, N
      IF (ABS(A(I, J)).GT.AAMAX) AAMAX=ABS(A(I, J))
11    CONTINUE
      IF (AAMAX.EQ.0.) PAUSE 'Singular matrix.'
      VV(I)=1./AAMAX
12    CONTINUE
    DO 19 J=1, N
      IF (J.GT.1) THEN
        DO 14 I=1, J-1
          SUM=A(I, J)
          IF (I.GT.1) THEN
            DO 13 K=1, I-1
              SUM=SUM-A(I, K)*A(K, J)
13            CONTINUE
              A(I, J)=SUM
            ENDIF
          CONTINUE
14          CONTINUE
        ENDIF
        AAMAX=0.
        DO 16 I=J, N
          SUM=A(I, J)
          IF (J.GT.1) THEN
            DO 15 K=1, J-1
              SUM=SUM-A(I, K)*A(K, J)
15            CONTINUE
              A(I, J)=SUM
            ENDIF
          DUM=VV(I)*ABS(SUM)
          IF (DUM.GE.AAMAX) THEN
            IMAX=I

```



```

C PURPOSE: CALCULATES THE INVERSE OF A MATRIX.          *
C                                                         *
C                                                         *
C                                                         *
C REFERENCES:                                             *
C                                                         *
C                                                         *
C WRITER: W C VENTER                                     *
C                                                         *
C DATE WRITTEN: 3/22/88                                  *
C                                                         *
C LAST REVISION: 3/24/88                                 *
C                                                         *
C*****
C
C      SUBROUTINE MATINV(A, Y, N)
C
C*****
C          *
C SYMBOL DECLARATIONS *
C          *
C*****
C
C
C      REAL*8 A(3,3), Y(3,3), D
C      INTEGER*4 INDX(3)
C      INTEGER*4 I, J
C
C#####
C      #
C DATA #
C      #
C#####
C
C      NP=3
C
C#####
C      #
C PROGRAM #
C      #
C#####
C

```

```
      DO 1000 I=1,N
        DO 1001 J=1,N
          Y(I,J)=0.
1001      CONTINUE
          Y(I,I)=1.
1000 CONTINUE
      CALL LUDCMP (A,N, NP, INDX, D)
      DO 1002 J=1,N
        CALL LUBKSB (A,N, NP, INDX, Y(1,J))
1002 CONTINUE
      RETURN
      END
```

```

C*****
C
C NAME: RNDPLOC.FOR
C
C PURPOSE: FIND THE LOCATION AND SIZE OF THE RAINDROPS
C
C
C
C
C PROGRAMMER : W C VENTER
C
C DATE WRITTEN: 12/17/88
C
C LAST REVISION: 1/2/89
C
C*****
C
C          PROGRAM RNDPLOC
C
C*****
C          *
C SYMBOL DECLARATIONS *
C          *
C*****
C
C          INTEGER*2 IHR1, IMIN1, ISEC1, IHUN1
C          INTEGER*2 IHR2, IMIN2, ISEC2, IHUN2
C          INTEGER*2 IHRD, IMIND, ISEC2, IHUND
C
C          INTEGER*2 I, J, K, L, IT, JT, KT, CELLX, RAD
C          INTEGER*2 IM, IP, JM, JP, M, ISIZE, JSIZE
C          INTEGER*2 CELL(256, 256)
C          INTEGER*2 NUM1, NUM2, NUM3, NUM4, XOFF, YOFF, K1, L1
C          REAL*4 C3, DUM
C
C*****
C          *
C PROGRAM          *
C          *
C*****
C
C#####

```

```

C          #
C START TIMER #
C          #
C#####
C
      CALL GETTIM(IHR1, IMIN1, ISEC1, IHUN1)
      WRITE(6,900)IHR1, IMIN1, ISEC1, IHUN1
      900 FORMAT(1X,'STARTING TIME :: ',I2,':'I2,':'I2,','I2)
C
C#####
C          #
C OPEN INPUT FILE #
C          #
C#####
C
C#####
C          #
C CELL INFORMATION MUST BE STORED IN FILE 'CELL' #
C IN AGRAF FORMAT #
C RAINDROP LOCATIONS WILL BE STORED IN FILE 'DROP' #
C          #
C#####
C
      OPEN(9, FILE='DROP', STATUS='UNKNOWN')
      OPEN(10, FILE='CELL', STATUS='OLD')
C
C#####
C          #
C READ INPUT DATA #
C          #
C#####
C
      WRITE(6,80)
      80 FORMAT(1X,'MINIMUM RAINDROP RADIUS IN PIXELS?')
      READ(6,*)RAD
C
      READ(10,*)ISIZE, JSIZE
      READ(10,*)((CELL(I,J),J=1,JSIZE),I=1,ISIZE)
C
C#####
C          #
C FIND POSITION AND SIZE OF FIRST CELL WITH #
C HIGHEST COUNT #

```

```

C                                                                 #
C#####
C
    WRITE(9,100) ISIZE, JSIZE
    100 FORMAT(1X,2I5)
C
    DO 200 M=1,500
C
    CELLX=0
    IT=0
    JT=0
    KT=0
    DO 120 I=1, ISIZE
        DO 120 J=1, JSIZE
            IF (CELL(I,J).LE.CELLX) GO TO 120
            CELLX=CELL(I,J)
            IT=I
            JT=J
            KT=CELL(I,J)
    120    CONTINUE
        IF (CELLX.LT.RAD) GO TO 210
C
C#####
C                                                                 #
C FIND RAINDROP POSITION    #
C                                                                 #
C#####
C
C#####
C                                                                 #
C FIND ROW OFFSET        #
C                                                                 #
C#####
C
    NUM2=0
    DO 300 K=1,KT
        NUM1=0
        DO 310 L=-KT,KT
            K1=IT+K
            L1=JT+L
            IF (K1.LT.1) GO TO 310
            IF (K1.GT.ISIZE) GO TO 310
            IF (L1.LT.1) GO TO 310

```

```

        IF (L1.GT.JSIZE)GO TO 310
        IF (CELL(K1,L1).LT.KT)GO TO 310
        NUM1=NUM1+1
310     CONTINUE
        IF (NUM1.EQ.0)GO TO 315
        NUM2=NUM2+1
300     CONTINUE
315 DUM=FLOAT(NUM2)/2.0
        XOFF=INT(DUM)
C
C#####
C                                     #
C FIND COLUMN OFFSET                 #
C                                     #
C#####
C
        NUM2=0
        NUM3=0
        NUM4=0
        DO 320 K--KT,KT
            NUM1=0
            DO 330 L=0,KT
                K1=IT+L
                L1=JT+K
                IF (K1.LT.1)GO TO 330
                IF (K1.GT.ISIZE)GO TO 330
                IF (L1.LT.1)GO TO 330
                IF (L1.GT.JSIZE)GO TO 330
                IF (CELL(K1,L1).LT.KT)GO TO 330
                NUM1=NUM1+1
330     CONTINUE
            IF (NUM1.EQ.0)GO TO 320
            IF (NUM4.NE.0)GO TO 340
            NUM4=NUM4+1
            NUM2=K
340     NUM3=K
320     CONTINUE
        YOFF=NUM3-NUM2
        DUM=FLOAT(YOFF)/2.0
        YOFF=INT(DUM)
C
C NUM2 IS ALWAYS LESS THAN OR EQUAL TO 0
C

```

```

      IT=IT+XOFF
      JT=JT+YOFF+NUM2
C
C#####
C                                     #
C STORE RAINDROP POSITION           #
C                                     #
C#####
C
      WRITE(6,150) IT, JT, KT
      150 FORMAT(1X,3I6)
      WRITE(9,150) IT, JT, KT
C
C#####
C                                     #
C REMOVE RAINDROPS                 #
C                                     #
C#####
C
      IM=IT-KT
      IP=IT+KT
      JM=JT-KT
      JP=JT+KT
      IF (IM.LE.1) IM=1
      IF (IP.GE.ISIZE) IP=ISIZE
      IF (JM.LE.1) JM=1
      IF (JP.GE.JSIZE) JP=JSIZE
C
      DO 160 I=IM, IP
        DO 160 J=JM, JP
          C3=(I-IT)**2.0+(J-JT)**2.0
          C3=SQRT(C3)
          IF(C3.GT.KT)GO TO 160
          CELL(I, J)=0
160      CONTINUE
C
200 CONTINUE
C
210 WRITE(9,150) 999, 999, 999
C
      CLOSE(9)
      CLOSE(10)
C

```

```

C#####
C                                     #
C STOP TIMER                         #
C CALCULATE TIME USED                #
C                                     #
C#####
C
      CALL GETTIM(IHR2, IMIN2, ISEC2, IHUN2)
      WRITE(6,901) IHR2, IMIN2, ISEC2, IHUN2
901  FORMAT(1X,'FINISHING TIME :: ',I2,':'I2,':'I2,','I2,','I2)
C
      IHRD=IHR2-IHR1
      IMIND=IMIN2-IMIN1
      ISECD=ISEC2-ISEC1
      IHUND=IHUN2-IHUN1
      IF(IHUND.GE.0)GO TO 903
      IHUND=IHUND+100
      ISECD=ISECD-1
903  IF(ISECD.GE.0)GO TO 904
      ISECD=ISECD+60
      IMIND=IMIND-1
904  IF(IMIND.GE.0)GO TO 905
      IMIND=IMIND+60
      IHRD=IHRD-1
905  WRITE(6,902) IHRD, IMIND, ISECD, IHUND
902  FORMAT(1X,'TIME USED :: ',I2,':'I2,':'I2,','I2,','I2)
C
      STOP
      END

```

```

C*****
C
C NAME: CIRITR.FOR
C
C PURPOSE: ITERATIVE CIRCLE FIT PROCEDURE FOR RAINDROP
C PROJECT
C
C
C
C
C PROGRAMMER : W C VENTER
C
C DATE WRITTEN: 12/17/88
C
C LAST REVISION: 4/21/89
C
C*****
C
C PROGRAM CIRITR
C
C*****
C
C SYMBOL DECLARATIONS
C
C*****
C
C INTEGER*2 IHR1, IMIN1, ISEC1, IHUN1
C INTEGER*2 IHR2, IMIN2, ISEC2, IHUN2
C INTEGER*2 IHRD, IMIND, ISECD, IHUND
C
C INTEGER*2 I, J, K, L, K2, L2, RAD, ISIZE, JSIZE
C INTEGER*2 KM, KP, LM, LP, TOT, NUM
C INTEGER*2 IM(256,256), CELL(256,256)
C REAL*4 C3, RRAD, DUM1, PER
C CHARACTER*20 FNAME
C
C#####
C #
C OPEN FILES #
C #
C#####
C
C WRITE(6,80)

```

```

80 FORMAT(1X,'NAME OF IMAGE FILE?')
   READ(6,90)FNAME
90 FORMAT(A20)
C
   WRITE(6,200)
200 FORMAT(1X,'PERCENTAGE OF PIXELS INSIDE ORIGINAL RAINDROP?')
   READ(6,*)PER
   PER=PER/100.0
C
   OPEN(10, FILE=FNAME, STATUS='OLD')
   OPEN(11, FILE='CELL', STATUS='UNKNOWN')
   OPEN(12, FILE='CELL.MAP', STATUS='UNKNOWN')

C
C#####
C      #
C START TIMER #
C      #
C#####
C
   CALL GETTIM(IHR1, IMIN1, ISEC1, IHUN1)
   WRITE(6,900)IHR1, IMIN1, ISEC1, IHUN1
900 FORMAT(1X,'STARTING TIME :: ',I2,':'I2,':'I2,','I2)
C
C#####
C      #
C PROGRAM      #
C      #
C#####
C
   READ(10,*)ISIZE, JSIZE
   READ(10,*)((IM(I,J),J=1,JSIZE),I=1,ISIZE)
C
C#####
C      #
C INITIALIZE CELLS #
C      #
C#####
C
   DO 95 I=1,ISIZE
     DO 95 J=1,JSIZE
95      CELL(I,J)=0
C

```

```

DO 100 I=2, ISIZE-1
  WRITE(6,160) I
  DO 100 J=2, JSIZE-1
C
C#####
C
C FIT LARGEST POSSIBLE RAINDROP CENTERED AT EACH PIXEL #
C
C#####
C
      RAD=1
150      KM=I-RAD
          KP=I+RAD
          LM=J-RAD
          LP=J+RAD
          IF (KM.LT.1) KM=1
          IF (KP.GT.ISIZE) KP=ISIZE
          IF (LM.LT.1) LM=1
          IF (LP.GT.JSIZE) LP=JSIZE
          NUM=0
          TOT=0
          DO 130 K=KM, KP
            DO 130 L=LM, LP
              C3=(K-I)**2+(L-J)**2
              C3=SQRT(C3)
              RRAD=RAD
              IF (C3.GT.RRAD) GO TO 130
              NUM=NUM+1
              IF (IM(K, L).EQ.0) GO TO 130
              TOT=TOT+1
130          CONTINUE
          DUM1=FLOAT(TOT)/FLOAT(NUM)
C
C#####
C
C DETERMINE IF FITTED CIRCLE CONTAINS ENOUGH RAINDROP #
C
C#####
C
      IF (DUM1.LE.PER) GO TO 140
      RAD=RAD+1
      GO TO 150
140      CELL(I, J)=RAD-1

```

```

100      CONTINUE
C
C#####
C          #
C WRITE OUT DATA  #
C          #
C#####
C
      WRITE(11,160) ISIZE, JSIZE
160  FORMAT(1X,2I5)
      WRITE(11,170) ((CELL(I,J),J=1,JSIZE),I=1,ISIZE)
170  FORMAT(1X,16I5)
C
C#####
C          #
C WRITE OUT DATA IN MAP FORMAT  #
C          #
C#####
C
C      KM=JSIZE/16
C      DO 300 K=1,KM
C          K2=(K-1)*16+1
C          L2=K*16
C          DO 310 I=1, ISIZE
C 310      WRITE(12,400) (CELL(I,J),J=K2,L2)
C          WRITE(12,410)
C 300      CONTINUE
C
C 400  FORMAT(1X,16I4)
C 410  FORMAT(1X)
C
C#####
C          #
C STOP TIMER          #
C CALCULATE TIME USED  #
C          #
C#####
C
      CALL GETTIM(IHR2, IMIN2, ISEC2, IHUN2)
      WRITE(6,901) IHR2, IMIN2, ISEC2, IHUN2
901  FORMAT(1X,'FINISHING TIME :: ',I2,':'I2,':'I2,','I2)
C
      IHRD=IHR2-IHR1

```

```
IMIND=IMIN2-IMIN1
ISECD=ISEC2-ISEC1
IHUND=IHUN2-IHUN1
IF (IHUND.GE.0) GO TO 903
IHUND=IHUND+100
ISECD=ISECD-1
903 IF (ISECD.GE.0) GO TO 904
ISECD=ISECD+60
IMIND=IMIND-1
904 IF (IMIND.GE.0) GO TO 905
IMIND=IMIND+60
IHRD=IHRD-1
905 WRITE (6,902) IHRD, IMIND, ISECD, IHUND
902 FORMAT (1X,'TIME USED :: ',I2,' : 'I2,' : ' ,I2,' .'I2)
C
CLOSE (10)
CLOSE (11)
CLOSE (12)
C
STOP
END
```