THE UNIVERSITY OF CALGARY

COMPUTER AIDED SURVEY NETWORK DESIGN

by

MICHAEL P.A. MEPHAM

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ENGINEERING

DEPARTMENT OF CIVIL ENGINEERING

DIVISION OF SURVEYING ENGINEERING

CALGARY, ALBERTA

APRIL, 1983

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read and recommend to the faculty of Graduate Studies for acceptance, a thesis entitled COMPUTER AIDED SURVEY NETWORK DESIGN by Michael P. Mepham, in partial fulfillment of the requirements for the degree of Master of Science in Engineering.

E.J. Krakiwsky

Dr. E.J. Krakiwsky (Supervisor)
Department of Civil Engineering
Division of Surveying Engineering

Fraser

Dr. C.S. Fraser
Department of Civil Engineering
Division of Surveying Engineering

W.F. Teskey

Mr. W.F. Teskey
Department of Civil Engineering
Division of Surveying Engineering

Brian Wyvill.

Dr. B. Wyvill
Department of Computer Science

29 April, 1983

ABSTRACT

The development of the program CANDSN for designing and adjusting horizontal survey networks using interactive computer graphics is presented in this thesis. The design mode uses an iterative approach for the solution of the first, second, and third order problems. The adjustment mode includes a complete analysis of the observations and the results.

The method of least squares is used for the computations. Sequential adjustment expressions are presented that enable the rapid updating of a previous solution when adding or deleting observations or paramaters. Special techniques for saving computation time and computer memory are employed. These include sequential formation of the normal equations, elimination of the orientation unknowns, efficient matrix inversion and storage algorithms, and reuse of the inverse of the normal equations during iteration.

The program employs graphical input and output techniques that make it easy to use. This is achieved through the use of a graphics terminal, a digitising tablet, and a flatbed plotter. The program has been implemented on a PDP-11/23 mini-computer and is written as a collection of FORTRAN subroutines.

Experience with the program has shown it to be a powerful tool with applications in the fields of geodetic, engineering, and cadastral surveys; as well, it is useful in the teaching of network design.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the help and encouragement I received from many people. Dr. E.J. Krakiwsky was my supervisor. His enthusiasm for the topic and generosity with his time and ideas made the successful completion of the project possible.

I would also like to express my appreciation to Dr. A.G. Herron who wrote most of the menu processing software and Dr. B. Wyvill who wrote the ellipse plotting subroutine. Ms. Shelley Webber did a superb job of typing a difficult manuscript. Finally, I would like to extend my thanks to the many faculty members and fellow graduate students who helped clarify issues by discussing and debating ideas and concepts with me.

Financial support for this research came from the Division of Surveying Engineering of the University of Calgary; the Natural Sciences and Engineering Research Council in the form of a graduate student assistantship; and the 1981 JMR Canada, Ltd. Research Award.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

In any engineering project, the design stage is critical to the successful completion of the project. Surveying Engineering is no exception. The design of a survey network, the subject of this research, involves deciding where the stations must be placed and what observations (including their accuracies) must be made in order to satisfy the accuracy requirements for the survey. Anderson [1982] introduces the concept of total optimisation of the design of a survey project. Included in this concept is the simultaneous optimisation of accuracy, sensitivity, reliability, and logistics including manpower, equipment availability, the time frame of the project, and costs. In his conclusions, he points out that the model for such a simultaneous optimisation is yet to be formulated. He does, however, outline a sequential model in which each component is optimised separately, but not independently of the other components.

The research reported in this thesis is aimed at implementing an efficient program for designing survey networks for optimum accuracy that could be easily extended to include other design criteria. This is the accuracy component of Anderson's total optimisation model. The technique of least-squares preanalysis was chosen at the start of the project as the method to use for the accuracy design as it is widely accepted by the surveying community. Preanalysis (called covariance analysis in mathematics) is the simulation of the propagation of uncertainties in the observations to uncertainties in the results. The mathematical basis and equations for this are presented in Chapter 2. Alternative approaches to the preanalysis of a network and the reasons for the approach chosen are presented in the remaining sections of this chapter.

1.1 Design Types

When designing a survey network to meet a given set of specifications, three different types of design problems may occur [Grafarend, 1974; Vanicek and Krakiwsky, 1982]. In all the problems, the accuracy specification is given in terms of the desired covariance matrix of the coordinates (C_x) or a subset of it. In the first-order design problem, the covariance matrix of the observables (C_ℓ) is assumed known. The problem is to find what geometric combination of observables (design matrix A) will yield the desired results. In the second-order design problem, the design matrix A is known and the required accuracy of the observables (C_ℓ) is determined. If both A and C_ℓ are unknown the problem is referred to as being a third-order design problem.

The first-order design problem corresponds to a situation where there is no choice of instrumentation for the survey. If the only equipment available is, say, a T2 theodolite and an AGA112 EDM, the only variable in the design is the configuration of the network. The second-order design problem is the reverse situation. The configuration is fixed but there is a choice of instrumentation. An example of this situation would be a monitoring survey for a dam. There are typically only a few stable points available that can be occupied, and the locations of the target points on the dam will be specified by the engineer so that the best results may be obtained from the deformation model.

The first and second-order design problems are subsets of the third-order design problem. For this reason, the program developed for this research was configured to solve the third-order problem. The first and second-order problems can be solved by simply not allowing the C_ℓ or A matrices to change.

1.2 Design Methods

There are several approaches to the calculation of the optimal design. The approaches and their advantages and disadvantages are presented here. The approach chosen for this research and the reasons for its choice are given in this section.

The first possibility for the solution is the direct mathematical approach. With this approach the unknowns, C_ℓ and or A, are solved for directly as a function of C_x using constraint and condition equations [Schmitt, 1981]. This technique has been applied to the solution of the second-order design problem [Grafarend, 1981; Koch, 1981] but is not without its problems. The C_ℓ matrix that results from the computations is generally singular and occasionally has negative variances in it. A singular C_ℓ matrix is in direct conflict with reality. Survey observations are from the real world and as such have finite positive variances. In addition, they are generally uncorrelated and where correlations do exist, they are small. If the negative variances in the computed C_ℓ matrix are interpreted as indicating that the observation does not add to the accuracy of the network [Schaffrin, 1981] and these observations are deleted from the design, the network may break into several individual networks with single ties between them. This conflict with good surveying practice casts considerable doubt on the pure mathematical approach.

The alternative to the direct solution is the iterative (trial and error) solution shown in Figure 1.1. Using this method, the user enters an initial design for the network based on a combination of practical experience, intuition, and common sense. The C_x matrix for this design is then computed and the differences between it and the desired C_x are analyzed. On the basis of the analysis and the user's knowledge of error propagation in surveying networks, the design is updated and a new C_x computed and analyzed. This process continues until the design is acceptable. This process will not generally yield an exact optimum design (in the mathematical sense) but the design will be a reasonable compromise. This is particularly true when we consider that the range of variances on surveying observables is not continuous but consists of only several discrete possibilities. As an example, the electronic distance measuring instruments in use today have accuracies of 1mm + 1ppm, 5mm + 2ppm, 5mm + 5ppm,

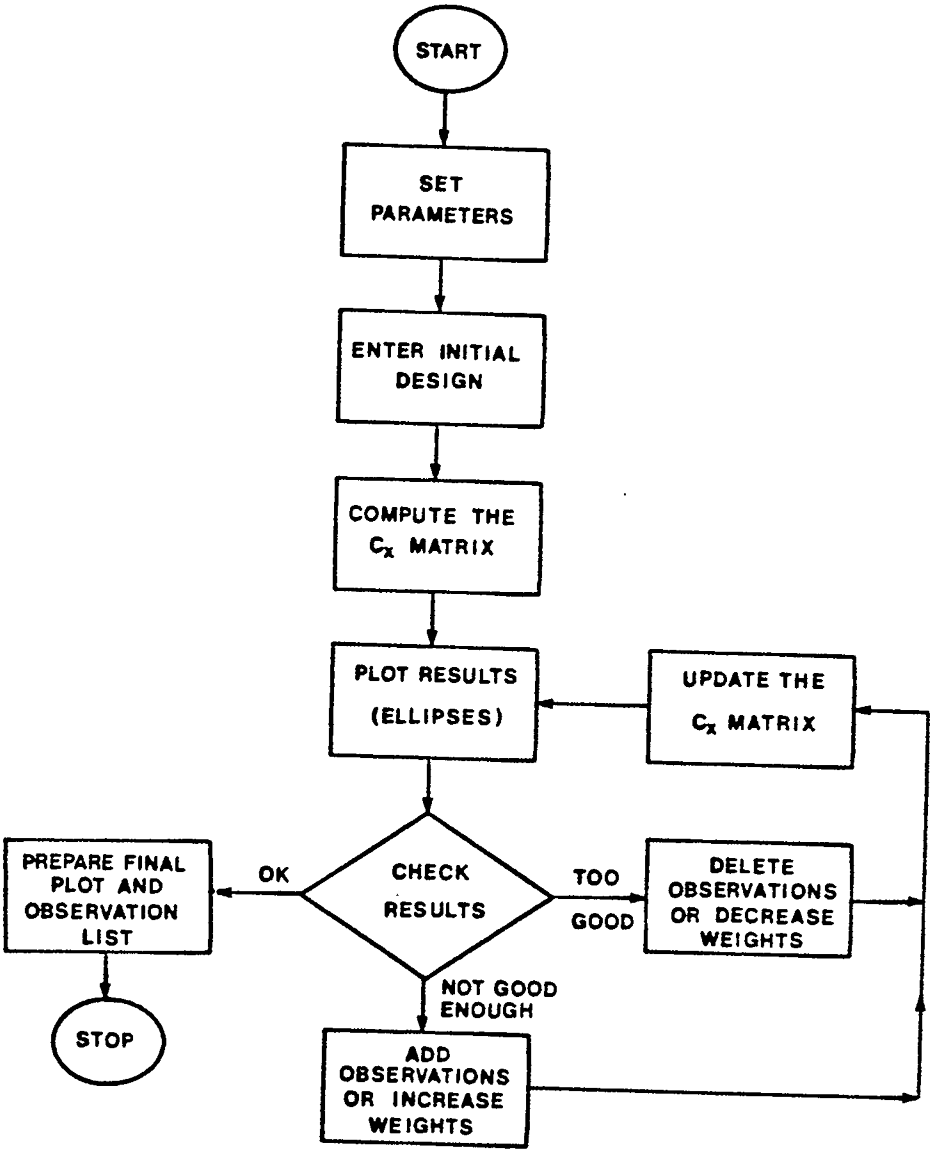


Figure 1.1

Iterative Preanalysis

or 15mm + 3ppm. If the exact optimum solution specifies the use of a 10mm + 6ppm instrument, a 5mm + 5ppm instrument would be used anyway as the specified instrument does not exist.

There are two approaches to the iterative solution. The first of these is the batch approach. With this method, the computation and analysis of the trial C_x matrix is submitted as a batch job on the computer and the results presented in the form of a printout. The results may also be plotted on an offline plotter. The disadvantage of this approach is the time delay between the submission of the batch job and the receipt of the results. This delay can be as long as a day. This interruption of the design process is a significant hindrance to an efficient design as the designer loses track of what has been done.

The alternative to the batch approach is to use an interactive approach. Using this method, the computations are done in real time and the results presented pictorially using a graphics terminal. This fast presentation of the results allows the user to maintain his concentration on the problem. The user does not have to wait until tomorrow to see the results of a change he made today. In addition, the pictorial representation of the results is an invaluable aid in their interpretation (see Section 6.2 for examples).

It should also be noted that a graphics terminal is not only an output device, but also a very powerful and useful input device. With a graphics terminal, commands may be entered using the cursor and a menu and stations can be identified quickly and naturally by pointing to them with a light-pen or the cursor. Not only do features like these make a program easier to use, but the number of user errors tends to decrease markedly.

1.3 Program CANDSN

Program CANDSN (Computer Aided Network DeSigN) was written as a result of the research reported in this thesis. It uses the iterative approach to solve the third-order design problem using interactive computer graphics. The idea originated with E.J. Krakiwsky at the University of New Brunswick, and Nickerson [1977] did a feasibility study on this technique concluding that it was a viable, simple, and useful tool.

Nickerson developed specialised equations for the various cases of addition or deletion of observables and stations. A unified approach has been developed in this thesis that uses only one sequential update equation for all network modifications. This approach is developed in Sections 2.2 and 2.3.

One of the objectives of the research was to implement the design on a small computer. It was felt that by making use of a smaller, and therefore less expensive, computer the use of the software package by smaller survey companies would be more likely. Special techniques had to be employed that would permit the rapid solution of large problems with such a small machine. These techniques are presented in Chapter 3. The result of the use of these methods is that the program is theoretically capable of solving any size problem. The limits to the size of problem that can be solved with CANDSN and the reasons for the limits are presented in Section 6.1.4. A discussion of the hardware configuration assembled is presented in Chapter 5 along with a description of the programming techniques.

Two of Nickerson's key recommendations were that the inclusion of weighted stations (stations with apriori covariance information on their coordinates) in the network being designed be possible, and that the program also have the ability to perform the final adjustment of the network once the observations have been made. Both of these recommendations have been implemented. The first of these is accomplished by the unified approach developed in Sections 2.2 and 2.3. The second recommendation was implemented by simply using the complete set of least squares adjustment equations presented in Section 2.1. In addition, most of the information entered at the design stage is also needed for the adjustment. Accordingly, program CANDSN has been written with network adjustment as one of its options. Once a network has been adjusted, the results of the adjustment must be analyzed. The tools used in this analysis are presented in Chapter 4.

No program will receive extensive use unless it is "user friendly". This implies that it must be easy to use, easy to learn, and handle bad data gracefully. CANDSN has been carefully designed and written with these considerations in mind. Section 6.1 discusses these concepts further and explains how they are implemented in CANDSN. This is followed in Section 6.2 by some examples of network designs done with the CANDSN software package.

Chapter 2

MATHEMATICAL BASIS

A well designed survey network is normally overdetermined, that is, there are more observations made than are necessary to uniquely determine the coordinates of the unknown points. The advantages of having an overdetermined network include; (a) an increase in the accuracy of the results; (b) the ability to statistically analyse the observations for blunders; and (c) the possibility of deleting bad observations without having to return to the field to collect more data.

An overdetermined network leads to an inconsistent set of equations. There are an infinite number of possible solutions to such a set of equations. The method of least squares is presented in the next section as the solution to an overdetermined set of linear equations. This is followed by a presentation of the mathematical models used for the survey observation types that are included in CANDSN and the application of the method of least squares to the solution of an overdetermined set of non-linear equations.

The last section of this chapter shows a reformulation of the least squares equations for sequential updating of the solution. This is essential for CANDSN as it allows the efficient updating of a trial solution during the design stage. It is in this section that the unified update approach used in CANDSN is presented.

2.1 The Method of Least Squares

The least squares method is a standard mathematical tool used for the solution of a set of overdetermined linear equations. It gives a solution that is an unbiased minimum variance estimate of the unknown parameters [Hamilton, 1964]. The solution is also a maximum likelihood estimate if the observations have a symmetric unimodal distribution [Aoki, 1967]. The least squares adjustment equations will be presented here without proof or derivation. Readers interested in the derivation are referred to any of the many books and papers on the subject, eg. [Mikhail, 1976; and Vanicek and Krakiwsky, 1982].

The least squares equations used in CANDSN are given below. The linear mathematical model is:

$$Ax + B\ell + c = 0, \tag{2.1}$$

where x is the vector of unknown paramaters,

ℓ is the vector of observations,

c is a vector of constants,

A is the first design matrix, and

B is the second design matrix.

The solution is

$$\hat{x} = -[A^T(BC_\ell B^T)^{-1}A + C_x^{-1}]^{-1}[A^T(BC_\ell B^T)^{-1}w] \tag{2.2}$$

where C_ℓ is the covariance matrix of the observations,

C_x is the apriori weight matrix of the unknowns, and

$\hat{}$ indicates a least squares estimate.

The covariance matrix of the estimated paramaters is

$$C_{\hat{x}} = N^{-1} = [A^T(BC_\ell B^T)^{-1} A + C_x^{-1}]^{-1} . \tag{2.3}$$

If the scale of $C_\ell(\sigma_0^2)$ is not known then the covariance matrix is given by

$$\hat{C}_{\hat{x}} = \hat{\sigma}_0^2 N^{-1} , \tag{2.4}$$

where $\hat{\sigma}_0^2$ is the estimated variance factor and is given by Equation 2.9. The adjusted observations $(\hat{\ell})$ are given by

$$\hat{\ell} = \ell + \hat{r} , \tag{2.5}$$

where $$\hat{r} = -C_\ell B^T(BC_\ell B^T)^{-1} (A\hat{x} + w) \tag{2.6}$$

and is referred to as the vector of residuals. The covariance matrix of $\hat{\ell}$ is given by the difference of two covariance matrices

$$C_{\hat{\ell}} = C_\ell - C_{\hat{r}} , \tag{2.7}$$

where the covariance matrix of the estimated residuals $(C_{\hat{r}})$ is given by the expression

$$C_{\hat{r}} = C_\ell B^T (BC_\ell B^T)^{-1} [I - AC_{\hat{x}}A^T(BC_\ell B^T)^{-1}] BC_\ell . \tag{2.8}$$

The estimated variance factor is given by

$$\hat{\sigma}_0^2 = \frac{\hat{r}^T C^{-1} \hat{r} + \hat{x}^T C_x^{-1} \hat{x}}{\nu + n_x} , \tag{2.9}$$

where ν is the number of degrees of freedom (= n - u),

m is the number of equations,

u is the number of unknowns, and

n_x is the number of paramaters with apriori variances and covariances.

At this point, a brief discussion of the computed values is appropriate. The vector $\hat{x}$ is the estimate of the unknown paramaters and needs no further elaboration. The matrix $C_{\hat{x}}$ (or $\hat{C}_{\hat{x}}$) is the covariance matrix of $\hat{x}$ and fully describes the accuracy and interdependence of the paramaters. It reflects both the accuracy of the observations and the effect of the geometry of the network on the accuracy of $\hat{x}$. This matrix is essential to both the design of a network and the analysis of the results of an adjustment. Interpretion of the contents of $C_{\hat{x}}$ will be discussed in Section 4.3. An examination of Equation 2.3 shows that no observed values are required for the computation of $C_{\hat{x}}$. It is strictly a function of the design matrices, the observational accuracies, and the apriori accuracies of the unknown paramaters. The first two are the variable elements in the third-order design problem, while one of them is known in the first and second-order design problems. Thus it is possible to compute the $C_{\hat{x}}$ matrix <u>before</u> conducting a survey. This is the basis of preanalysis.

The vector $\hat{\ell}$ is the estimate of the true value of the observation. The observations, $\hat{\ell}$, have random errors associated with them. The residual vector, $\hat{r}$, is the estimate of the random errors. The analysis of the observations (Section 4.2) uses this vector to test the correctness of the vector of observations $(\hat{\ell})$. The associated covariance matrix $C_{\hat{r}}$ is also required for the analysis.

2.2 Mathematical Models

A math model is an equation defining the functional relationship between the observables (ℓ) and the unknowns (x). The general form of a math model is $f(x,\ell) = 0$. The model is formulated on the basis of physical and geometric laws and must have the following properties:

a) It must fully describe the geometry of the network;

b) All observations must be used;

c) The equations must not be linearly dependent; and

d) The unknown paramaters must not be linearly dependent.

There are eight math models used in CANDSN. They are all formulated on the conformal mapping plane and have the special form $f(x) - \ell = 0$. The first four models deal with the standard surveying observations. The distance model is

$$[(N_j - N_i)^2 + (E_j - E_i)^2]^{\frac{1}{2}} - S_{ij} = 0 \ , \tag{2.10}$$

where N is the unknown northing,

E is the unknown easting, and

S_{ij} is the distance between points i and j.

The azimuth model is

$$tan^{-1}\left(\frac{N_j - N_i}{E_j - E_i}\right) - \alpha_{ij} = 0, \tag{2.11}$$

where α_{ij} is the azimuth from i to j.

The angle model is

$$\tan^{-1}\left(\frac{N_k - N_i}{E_k - E_i}\right) - \tan^{-1}\left(\frac{N_j - N_i}{E_j - E_i}\right) - \theta_{ijk} = 0, \tag{2.12}$$

where θ_{ijk} is the angle from j to k at i.

Finally, the direction model is

$$\tan^{-1}\left(\frac{N_j - N_i}{E_j - E_i}\right) - \Omega - d_{ij} = 0, \tag{2.13}$$

where d_{ij} is the direction from i to j, and

Ω is the orientation unknown.

The newly introduced paramater Ω simply transforms a direction to an azimuth. One orientation unknown is introduced for each round of directions or one for each round of adjusted directions resulting from the station adjustment of a set of direction observations. The elimination of this nuisance paramater is presented in Section 3.2.

The fifth and sixth math models used in CANDSN are the coordinate difference models. They are used to allow the inclusion of , e.g., satellite translocation or intertial survey results in the network. The models are

$$(N_j - N_i) - \Delta N_{ij} = 0, \tag{2.14}$$

and

$$(E_j - E_i) - \Delta E_{ij} = 0 , \qquad (2.15)$$

where ΔN_{ij} is the difference in the northings of points i and j

and ΔE_{ij} is the difference in the eastings of points i and j.

The last math models used in CANDSN are the coordinate observation models. They have two different applications. They may be used to include stations established by, e.g., satellite point positioning methods or previously determined in a separate network adjustment. They are also used as part of the unified approach for the addition and deletion of weighted stations as shown in Section 2.3. The models are simply

$$N_i - N_i = 0. \qquad (2.16)$$

and

$$E_i - E_i = 0. \qquad (2.17)$$

The C_ℓ matrix for the first four observation types listed is normally a diagonal matrix. If angles are computed from observed directions, there will be strong correlations between calculated angles at a common station. For this reason, the observed directions should be used as the observables and the elimination procedure presented in Section 3.2 used to eliminate the orientation unknowns. CANDSN does not allow the use of correlated angles. The last two observation types have a fully populated C_ℓ matrix associated with them. The correlations are significant and cannot be neglected. The full C_ℓ matrix is used in CANDSN for these observations.

2.2.1 Linearisation and Iteration

The first four models presented (Eqns. 2.10 to 2.13) are non-linear while the least squares method shown earlier is for solving sets of linear equations. A Taylor's series expansion is used to linearise the math models in order to overcome this problem. Applying Taylor's series expansion to the problem at hand and neglecting the second-order terms, we get [Pope, 1974].

$$A = \left.\frac{\partial f(x, \ell)}{\partial x}\right|_{x^o,\ \ell^o} \tag{2.18}$$

$$B = \left.\frac{\partial f(x, \ell)}{\partial \ell}\right|_{x^o,\ \ell^o} \tag{2.19}$$

and

$$w = f(x, \ell)\ \Big|_{x^o,\ \ell^o} \tag{2.20}$$

where x^o is the initial approximation to x, and

ℓ^o is the observed value of ℓ.

Noting that for all the math models, B = -I, the least squares equations become the following:

$$\hat{\delta} = -[A^T C_\ell^{-1} A + C_x^{-1}]^{-1} [A^T C_\ell^{-1} w] \quad , \qquad (2.21)$$

$$\hat{x} = x^o + \hat{\delta} \qquad (2.22)$$

$$C_{\hat{x}} = N^{-1} = [A^T C_\ell^{-1} A + C_x^{-1}]^{-1} \quad , \qquad (2.23)$$

$$\hat{r} = A\hat{\delta} + w \ , \qquad (2.24)$$

$$C_{\hat{r}} = C_\ell - AC_{\hat{x}}A^T \ , \qquad (2.25)$$

and

$$\hat{\sigma}_0^2 = \frac{\hat{r}^T C_\ell^{-1} \hat{r} + \hat{\delta}^T C_x^{-1} \hat{\delta}}{\nu + n_x} \quad , \qquad (2.26)$$

where $\hat{\delta}$ is the vector of corrections to x^o, the initial approximations to x.

In this linearisation, the second and higher-order terms of the Taylor's series expansion have been neglected. The effect of this is that the solution must be iterated. This is accomplished by repeating the solution replacing x^o with the last $\hat{x}$ until the solution has converged. The test for convergence is

$$\left| \hat{\delta}_i \right| < \varepsilon_i \text{ for all } i \ , \qquad (2.27)$$

where ε_i is some small quantity chosen by the user. The convergence criteria ε_i should be chosen carefully. If it is too large, the results will be inaccurate, while if it is too small, an excessive number of iterations will be performed. This value typically ranges from 1 cm for large geodetic networks to 0.1 mm for precise engineering surveys.

In addition to the convergence criteria, a maximum number of iterations should also be specified. This protects against infinite loops in the event of too small an ε or an unstable network. [Mikhail 1976] gives several alternate forms of convergence criteria.

2.3 Sequential Techniques

Using the iterative design method, the changes to be made between trial solutions are usually small in number. The least squares equations given earlier are transformed in this section into a set of sequential adjustment equations that allow the efficient updating of a design or adjustment, given a small number of changes.

The development starts with the partitioning of A, C_ℓ, and w into old and new information. This results in

$$A^* = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} , \tag{2.28}$$

$$C_\ell^* = \begin{bmatrix} C_{\ell_1} & 0 \\ 0 & C_{\ell_2} \end{bmatrix} \tag{2.29}$$

and

$$w^* = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{2.30}$$

The subscript 1 in Equations 2.28 to 2.30 denotes original information while 2 denotes the new information. No subscript indicates the total solution.

Placing Equations 2.28 to 2.30 into Equation 2.21 and expanding gives

$$\hat{\delta} = -[N_1 + A_2^T C_{\ell_2}^{-1} A_2]^{-1} \; [u_1 + A_2^T C_{\ell_2}^{-1} w_2]$$

$$= -[N_1 + A_2^T C_{\ell_2}^{-1} A_2]^{-1} \; u_1 - [N_1 + A_2^T C_{\ell_2}^{-1} A_2]^{-1} \; [A_2^T C_{\ell_2}^{-1} w_2] \; , \qquad (2.31)$$

where

$$N = A^T C_\ell^{-1} A \; , \qquad (2.32)$$

and

$$u = A^T C_\ell^{-1} w \; . \qquad (2.33)$$

Applying the matrix identities [Krakiwsky, 1981; Mikhail, 1976]

$$[S^{-1} + T^T R^{-1} T]^{-1} = S - ST^T [R + TST^T]^{-1} TS \qquad (2.34)$$

and

$$[S^{-1} + T^T R^{-1} T]^{-1} T^T R^{-1} = ST^T [R + TST^T]^{-1} \qquad (2.35)$$

to Equation 2.31 gives the following sequential expressions

$$\hat{\delta} = \hat{\delta}_1 - N_1^{-1} A^T [\pm \; C_{\ell_2} + A_2 N_1^{-1} A_2^T]^{-1} \; [A_2 \hat{\delta}_1 + w_2] \qquad (2.36)$$

and

$$C_{\hat{x}} = N_1^{-1} - N_1^{-1} A_2^T [\pm \; C_{\ell_2} + A_2 N_1^{-1} A_2^T]^{-1} A_2 N_1^{-1} \qquad (2.37)$$

The "+" sign in Equations 2.36 and 2.37 corresponds to the addition of data while the "-" sign corresponds to the deletion of data.

The efficiency of using the sequential expressions is shown by examining the sizes of the inversions [Krakiwsky, 1981] . If the problem was solved by starting from scratch using Equation 2.21 a u by u matrix would have to be inverted where u is the number of unknown paramaters. This is independent of the number of observations added (or deleted). Using the sequential approach (Eqn. 2.36), the only new inverse is of an n_2 by n_2 matrix where n_2 is the number of observations added (or deleted) at a time.

The point of expansion (x^o) for Equation 2.36 as written is the same point of expansion that was used in the original solution for $\hat{\delta}_1$. The term $A_2\hat{\delta}_1 + w_2$ is a linear approximation to $f_2(x^o + \hat{\delta}_1, \ell)$ given w_2 evaluated as $f_2(x^o, \ell)$. The alternative to this approach is to use the current adjusted value of x as the point of expansion. This implies that $\hat{\delta}_1$ is zero and w_2 is calculated using the non-linear equation

$$w_2 = f_2(\hat{x}_1, \ell) \ . \tag{2.38}$$

Using this alternative approach, the update does not have to be iterated. The correction to the paramaters due to the addition of one observation is small enough that the non-linear component can be neglected. This would not be true if the w_2 term was large as may be the case when using the original point of expansion. The use of the current best estimate ($\hat{x}_1$) as the point of expansion has the added advantage of simplifying the programming. A record of the original point of expansion does not have to be kept.

The equations just developed are used in CANDSN for all updates to the network. This is the heart of the unified approach used. In his development, Nickerson [1977] set up different update equations for adding and deleting observations, nuisance parameters, and weighted stations. These are all accomplished by the use of the appropriate math model and Equations 2.36 to 2.38 in CANDSN. The steps involved in deleting a weighted station from a network using both methods are shown next as a comparison of the two approaches. Using the unified approach, the steps are

1. Add false C_X^{-1} to station being deleted. The C_X^{-1} used is the identity matrix. This step is necessary to keep the $C_{\hat{X}}$ matrix non-singular during the next stages.
2. Using the coordinate observation math models (Eqns. 2.16 and 2.17) and Equation 2.36 remove the original apriori covariance matrix.
3. Using the observation math models (Eqns. 2.10 to 2.15) and Equation 2.36 remove all the observations connecting the station being deleted to the network.
4. Now that all connections between the network and the station being deleted have been removed, only the false C_X^{-1} matrix added in Step 1 remains. The final step is to simply eliminate the rows and columns of $C_{\hat{X}}$ that corresponded to the deleted station.

The equation given by Nickerson for this is

$$C_{\hat{x}} = D + DA_{22}^T[C_{\ell_2}^{-1} - A_2DA_2^T]^{-1} A_{22}D \quad , \tag{2.39}$$

where

$$D = Q - QN_{12}[N_2 + N_{12}^TQN_{12}]^{-1} N_{12}Q \quad . \tag{2.40}$$

The matrix Q in Equation 2.40 is the submatrix of the existing $C_{\hat{x}}$ corresponding to the stations being kept. The A_{12} and A_{22} matrices are the portions of A_2 corresponding to the stations being kept and deleted, respectively. The N matrices are given by

$$N_{12} = A_{12}^TC_{\ell_2}^{-1}A_{12} \quad , \tag{2.41}$$

and

$$N_2 = C_{x_2}^{-1} + A_2^TC_{\ell_2}^{-1}A_2 \quad . \tag{2.42}$$

Examination of these two alternatives shows them to be of equal efficiency. The inverses that need to be computed are the same size in each method. The advantages of the method used in CANDSN are that there are no large intermediate products that need to be computed and no specialised computation routines are required. The same update routine is used for all types of updates.

The unified approach also makes it possible to change the point of reference to the network. This has particular application to engineering networks where it is common practice to choose one point and arbitrarily consider its assigned coordinates to be known and errorless. The choice of the fixed point has an effect on the point accuracy of the other stations. A discussion of point accuracy is given in Section 4.3.1. Using the update equations, it is possible to change the fixed station and investigate the effects of the change. This is done by using the coordinate observation math model to change an unknown station to a weighted station using a very small variance on the "observed" coordinates and then releasing the previously fixed station. The effect of this approach is the same as using an S-transform [Baarda, 1973] although the mathematics are very different.

Chapter 3

SPACE AND TIME SAVING TECHNIQUES

The techniques used in CANDSN to fit a large least squares problem onto a small computer are described in this chapter. In the selection of techniques, only those that would also have a high enough execution speed that the user would not have to wait excessively long for results were considered. The techniques described illustrate methods of efficient storage and inversion of matrices. The first section deals with the elimination of the largest matrix used in the least square method, A, and the remaining sections deal with the efficient solution of linear systems of equations. This includes a description of the storage technique used in CANDSN for the normal equation matrix.

3.1 Sequential Formation of the Normal Equations

The design matrix A is the largest matrix in the set of least squares equations. It has one row per observation and one column per unknown. The matrix is also very sparse. There are, at most, six of the u elements per row that are not zero. In this section, a simple technique for the total elimination of this matrix is presented.

If we partition the A matrix into rows ($[a_i]$ denotes the i'th row of A),

$$A = \begin{bmatrix} [a_1] \\ [a_2] \\ \vdots \\ [a_n] \end{bmatrix} \tag{3.1}$$

the diagonal C_ℓ matrix into its individual elements

$$C = \begin{bmatrix} C_{\ell_1} & 0 & 0 \\ 0 & C_{\ell_2} \ddots & 0 \\ 0 & 0 & C_{\ell_n} \end{bmatrix} \tag{3.2}$$

and the w vector into its individual elements

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \tag{3.3}$$

and substitute these into the equations for N and u we get

$$N = [a_1]^T C_{\ell_1}^{-1} [a_1] + [a_2]^T C_{\ell_2}^{-1} [a_2] + \ldots + [a_n]^T C_{\ell_n}^{-1} [a_n] \ , \tag{3.4}$$

and

$$u = [a_1]^T C_{\ell_1}^{-1} w_1 + [a_2]^T C_{\ell_2}^{-1} w_2 + \ldots + [a_n]^T C_{\ell_n}^{-1} w_n \tag{3.5}$$

From these equations it is apparent that N and u can be formed sequentially (by summation) one observation at a time and that no matrix A or vector w are needed. A further saving in space and execution time can be realised if only the non-zero elements of each row of the matrix A are used in the computations.

This technique is valid only if the C_ℓ matrix is a diagonal matrix. Frequently this is not the case. The elimination of the orientation unknowns (Section 3.2) creates a block diagonal C_ℓ matrix, and coordinate-difference observations and coordinate observations do not generally have diagonal C_ℓ matrices. In this case, the A matrix is partitioned into blocks of rows corresponding to the block diagonal structure of the C_ℓ matrix.

The equation for $\hat{r}$ given in Chapter 2 is

$$\hat{r} = A\hat{\delta} + w \ ; \tag{3.7}$$

but using the technique just outlined, there is no A matrix or w vector available for the computation of $\hat{r}$. This difficulty may be overcome by noting that

$$f\ (\hat{x}) - \hat{\ell} = 0 \tag{3.8}$$

and

$$\hat{\ell} = \ell + \hat{r}. \tag{3.8}$$

Substituting Equation 3.7 into Equation 3.6 we get

$$f\ (\hat{x}) - \ell - \hat{r} = 0 \tag{3.9}$$

or

$$\hat{r} = \ell - f(\hat{x}). \tag{3.10}$$

Note that Equation 3.10 is exact while Equation 3.6 is actually a linear approximation.

3.2 Elimination of Orientation Unknowns

The mathematical model for direction observations (Eqn. 2.13) introduced a new unknown, Ω, for each round of directions. There are two coordinate unknowns and, typically, one orientation unknown per station in a survey network. The time required to compute the inverse, or Cholesky root, of a matrix is proportional to u^3 so the elimination of the orientation unknowns will reduce the execution time by a factor of $2^3 / 3^3$ or almost 30%.

The orientation unknowns can be eliminated as follows [Steeves, 1974]. Partition the δ vector into δ_1 corresponding to the coordinate unknowns and δ_2 corresponding to the orientation unknowns. The A matrix is similarly partitioned. This leads to the equation

$$[A_1 \quad A_2]^T \, C_\ell^{-1} \, [A_1 \quad A_2] \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} + [A_1 \quad A_2]^T \, C_\ell^{-1} w = 0 \tag{3.11}$$

which, when expanded, becomes

$$\begin{bmatrix} A_1^T C_\ell^{-1} A_1 & A_1^T C_\ell^{-1} A_2 \\ A_2^T C_\ell^{-1} A_1 & A_2^T C_\ell^{-1} A_2 \end{bmatrix} \cdot \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} + \begin{bmatrix} A_1^T C_\ell^{-1} w \\ A_2^T C_\ell^{-1} w \end{bmatrix} = 0. \tag{3.12}$$

Applying the identify [Fadeev and Fadeeva, 1963] that if

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix} = 0, \tag{3.13}$$

then

$$[D - CA^{-1}B]y + v - CA^{-1}u = 0, \tag{3.14}$$

to Equation 3.12 we get

$$\delta_1 = -[A_1^T Q A_1]^{-1} \, [A_1^T Q w] \quad , \tag{3.15}$$

where

where

$$Q = [C_\ell^{-1} - C_\ell^{-1} A_2 \; [A_2^T C_\ell^{-1} A_2]^{-1} A_2^T C_\ell^{-1}]. \qquad (3.16)$$

Equation 3.15 is identical in form to the least squares equations given previously with the new matrix Q replacing C_ℓ^{-1}. Noting that A_2 consists entirely of partial columns of negative ones and C_ℓ is a diagonal matrix, the computation of Q is simply

$$q_{i,j} = p_{i,j} - p_{i,i}*p_{j,j}*Z \qquad (3.17)$$

where $q_{i,j}$ is the (i, j) element of Q,

$$Z = -\left(\sum_{k=1}^{n} p_{k,k}\right)^{-1} , \qquad (3.18)$$

and $p_{i,j}$ is the (i, j) element of C_ℓ^{-1}.

The computation of $\hat{r}$ using Equation 3.10 requires the adjusted value of the orientation unknown. Steeves [1974] shows that it may be computed after the adjustment and the residuals computed as follows

1. Compute an initial approximation $\tilde{r}$ from Equation 3.10 letting $\Omega = 0$;
2. Multiply $\tilde{r}$ by $-[p_{11}*Z, p_{22}*Z, \ldots , p_{nn}*Z]$ to get Ω;
3. Subtract Ω from each element of $\tilde{r}$ to get $\hat{r}$.

3.3 Matrix Inversion

All the matrices requiring inversion when using the method of least squares as described in Chapter 2 are symmetric. With one exception they are also positive definite. The exception is the term $[\pm C_{\ell_2}^{-1} + A_2 N_1^{-1} A_2^T]^{-1}$ in Equations 2.36 and 2.37. This matrix is positive definite when adding data and is negative definite when deleting data. Special algorithms have been developed that take advantage of the symmetry of matrices to economize on both the storage space required and the computational effort required for their inversion. The remainder of this section discusses the Cholesky, block Cholesky, and U-D algorithms used in CANDSN.

3.3.1 Cholesky Factorisation

A positive definite matrix, A, can be factored into the product of the transpose of an upper triangular matrix and itself [Burden et al, 1978], namely

$$R^T R = Q. \tag{3.19}$$

The algorithm for the computation of R is the following

```
For j = 1 to n
    For k = 1 to j - 1
        $r_{k,j} = \left[ q_{k,j} - \sum_{i=1}^{k-1} r_{i,k} * r_{i,j} \right] / r_{k,k}$
    End k
```

```
        $r_{j,j} = \left[ q_{j,j} - \sum_{i=1}^{j-1} q_{i,j}^2 \right]^{1/2}$

    End j.
```

Substituting Equation 3.19 into the least squares solution gives

$$\hat{\delta} = -N^{-1}u$$

$$= -(R^TR)^{-1}u = -R^{-1}R^{-T}u \qquad (3.20)$$

Since the matrix R is triangular, the equations

$$d = R^{-T}u \qquad (3.21)$$

and

$$\hat{\delta} = -R^{-1}d \qquad (3.22)$$

can be solved directly with no explicit inverse computations. The algorithms are the following

```
    For k = 1 to n

        $d_k = \left[ u_k - \sum_{i=1}^{k-1} r_{i,k} * u_i \right] / r_{k,k}$

    End k
```

and

```
    For k = n to 1 by -1
```

```
        $\hat{\delta}_k = -d_k / r_{k,k}$
         For i = 1 to k - 1
            $\hat{\delta}_i = \hat{\delta}_i - \hat{\delta}_k * r_{i,k}$
         End i
    End k.
```

These are referred to, respectively, as the forward and backward solutions. The inverse of the normal equations, when required, is computed from

$$N^{-1} = (R^T R)^{-1} = R^{-1} R^{-T}. \tag{3.23}$$

The algorithm for this is the following:

```
    For k = 1 to n
        $r_{k,k} = 1 / r_{k,k}$
        For i = 1 to k - 1
            $r_{i,k} = -r_{k,k} * r_{i,k}$
        End i

        For j = k + 1 to N
            $r_{k,j} = 0$
            For i = 1 to k
                $r_{i,j} = r_{i,j} + r_{k,j} * r_{i,k}$
            End i
        End j.
```

Examination of these algorithms shows that it is possible to overwrite N with R, R with N^{-1}, u with d, and d with $\hat{\delta}$. The result is that no matrices other than N and u are required for the solution. It is also possible to store only the upper triangular elements of N in a vector [Dongarra et al 1979]. This realises a space saving of just under 50% of that required to store the full N matrix.

It is also apparent that it is not necessary that N^{-1} be computed on every iteration. Only the Cholesky root, R, and the forward and backward solutions are needed. This results in a saving in computation time of approximately 50%.

3.3.2 Block Cholesky

The block Cholesky method is simply the application of the Cholesky algorithm to a matrix that has been partitioned such that all the on-diagonal blocks are square. The individual elements in the Cholesky expressions are now submatrices and the calculations involve matrix rather than scalar algebra. In the evaluation, the scalar division operation is replaced by the matrix inversion and multiplication operations. Since the only inverses are of blocks that are on the diagonal, the Cholesky algorithm can be used to compute the inverses. This is possible, because, by definition, all on-diagonal submatrices of a positive definite matrix are themselves positive definite. The square root operation required in the computation of R is replaced by the Cholesky decomposition. Examination of Equation 3.19 will reveal that R is in fact the square root of Q.

The block Cholesky method is used in CANDSN for the initial inversion of N when operating in the design mode and for the computation of $\hat{\delta}$ as well when operating in the adjustment mode. The normal equation matrix is partitioned into 10 by 10 blocks (referred to as "panels") and stored in a FORTRAN direct access file. During computations no more than three of these panels are held in core at any one time. The use of a direct access file allows the efficient reading and updating of any panel.

The major advantage of this technique is that the size of the problem to be solved is not dependent on the memory available on the computer. The only limit is that imposed by the space available on disk for file storage. This technique has been used for the iterative solution of a set of 82 000 normal equations for the readjustment of the Maritime Geodetic Network [Knight and Mepham, 1978; Nickerson 1981].

3.3.3 U-D Factorisation

The Cholesky method is only valid for positive definite matrices. This is a result of the square root operation required in the computation of R. If the matrix is negative definite, the square root is of a negative number. As mentioned at the beginning of this section, the deletion of observations requires the inverse of a matrix that is negative definite. An algorithm for the inversion of a symmetric definite matrix is therefore needed. The U-D factorisation algorithm was chosen for this as it only requires that the matrix be symmetric definite. It is also referred to as "Cholesky without square roots" [Lawson and Hanson, 1974].

A symmetric non-singular matrix Q may be factored into an upper triangular matrix U with all diagonal elements equal to one and a diagonal matrix D such that

$$Q = U^{T}DU \tag{3.24}$$

or

$$Q^{-1} = (U^{T}DU)^{-1} = U^{-1}D^{-1}U^{-T} \tag{3.25}$$

The algorithm for the factorisation, in place, is the following:

```
For i = 1 to n - 1
    $q_{i,n} = q_{i,n} / q_{n,n}$
End i
For j = n - 1 to 1
    $q_{j,j} = q_{j,j} - \sum_{k = j + 1}^{n} q_{j,k}^{2} * q_{k,k}$
    For k = j + 1 to n
        $t_k = q_{j,k} * q_{k,k}$
    End k
    For i = 1 to u - 1
        For k = j + 1 to n
            $q_{i,j} = q_{i,j} - q_{i,k} * t_k$
        End k
        $q_{i,j} = q_{i,j} / q_{j,j}$
        End i
    End j,
```

where t is a work vector of at least n - 1 elements. This factorisation puts the off diagonal elements of U over the original elements of Q and the diagonal elements of D over the original diagonal elements of Q. There is no need to save the diagonal elements of U as they are always one. The algorithm for computing, in place, Q^{-1} from the factorisation just given is the following:

```
For i = n - 1 to 1
    For j = n to i + 1
        s = 0
        For k = i + 1 to j - 1
            $s = s - q_{i,k}*q_{k,j}$
        End k
        $q_{i,j} = s - q_{i,j}$
    End j
End i
For i = 1 to n
    $q_{i,i} = 1 / q_{i,i}$
End i
For i = n to 2
    For j = n to i + 1
        s = 0
        For k = 1 to i - 1
            $s = s + q_{k,k}*q_{k,j}*q_{k,i}$
            End k
            $q_{i,j} = s + q_{i,j}*q_{i,i}$
        End j
        For j = 1 to i - 1
            $q_{i,i} = q_{i,i} + q^2_{j,i}*q_{j,j}$
        End j
    End i
    For j = 2 to n
        $q_{1,j} = q_{1,j}*q$
    End j.
```

The U-D technique can also be used to solve a set of linear equations using backward and forward solutions similar to those given for the Cholesky method. It can also be adapted to operate on panel matricies as with Block Cholesky. It was not chosen for the inversion of the full normals nor for the solution of the adjustment equation because of the necessity of the work vector t. It is used throughout CANDSN for all other matrix inversions.

3.4 Reuse of N^{-1} During Iteration

Computationally, the most time consuming aspect of a least squares adjustment is the calculation of the inverse, or Cholesky root, of the normal equations. The proposal made by Knight [1978] was to reuse the original Cholesky root for all iterations. The rationale is that N is a function of only A and C_ℓ. A is a function of the updated coordinates but the effect of the changes in coordinates, $\hat{\delta}$, is a second order one. C_ℓ does not change between iterations.

The change in A due to $\hat{\delta}$ is

$$\Delta A = A_2 - A_1 , \tag{3.24}$$

where

$$A_1 = \left.\frac{\partial f}{\partial x}\right|_{x^0, \ell} , \tag{3.25}$$

and

$$A_2 = \left.\frac{\partial f}{\partial x}\right|_{x^0 + \hat{\delta}, \ell} . \tag{3.26}$$

Since $\hat{\delta}$ is generally small, ΔA can be expressed as a differential change of the form

$$\Delta A = \left.\left[\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right)\right]\right|_{x^0, \ell} \cdot \hat{\delta} \tag{3.27}$$

$$= A' \cdot \hat{\delta} \tag{3.28}$$

A' is a three dimensional matrix of size n by u by u. The terms in it are second derivatives of the math models given in Section 2.2. The second derivatives of the coordinate difference and coordinate observation models (Eqns. 2.14 to 2.17) are all zero. There is no effect on A due to $\hat{\delta}$ for these models.

The second derivatives of the distance model (Eqn. 2.10) with respect to the occupied station are

$$\frac{\partial}{\partial N_i}\left(\frac{\partial f}{\partial N_i}\right) = \frac{(N_j - N_i)^2}{S^3} - \frac{1}{S} \quad , \tag{3.29}$$

and

$$\frac{\partial}{\partial E_i}\left(\frac{\partial f}{\partial N_i}\right) = \frac{(N_j - N_i) \cdot (E_j - E_i)}{S^3} \quad , \tag{3.30}$$

where

$$S = [(N_j - N_i)^2 + (E_j - E_i)^2]^{\frac{1}{2}} \quad . \tag{3.31}$$

The derivatives with respect to the sighted station are identical in form. The derivatives of the azimuth model (Eqn. 2.11) with respect to the occupied station are

$$\frac{\partial}{\partial N_i}\left(\frac{\partial f}{\partial N_i}\right) = \frac{2 \cdot (E_j - E_i) \cdot (N_j - N_i)}{S^4} \tag{3.32}$$

and

$$\frac{\partial}{\partial E_i}\left(\frac{\partial f}{\partial N_i}\right) = \frac{2 \cdot (E_j - E_i)}{S^4} - \frac{1}{S^2} \quad . \tag{3.33}$$

The derivatives with respect to the sighted station and the derivatives of the direction and angle models have the same form.

Examination of these equations show that all the terms in A' will vary inversely as either the distance or the square of the distance between the stations. The result of this is that the ΔA matrix will generally have all negligably small numbers and therefore the N matrix will not change appreciably during iteration. This procedure will fail if there are very short lines (as in the case of eccentric observations) included in the adjustment. The iteration scheme described next shows how this is taken into consideration.

It should be noted that while the N^{-1} can be reused, the same does not hold true for the u vector. It must be recomputed for every iteration.

The iterative procedure used in CANDSN is shown in Figure 3.1. The outer loop shown is there to guard against ΔA having a significant effect. Using this procedure, the best estimates of x will always have been used to form the N matrix for the last iteration. With this method, a network which requires three iterations to converge will require only two inverses resulting in a saving of approximately 25 to 30% of the execution time.

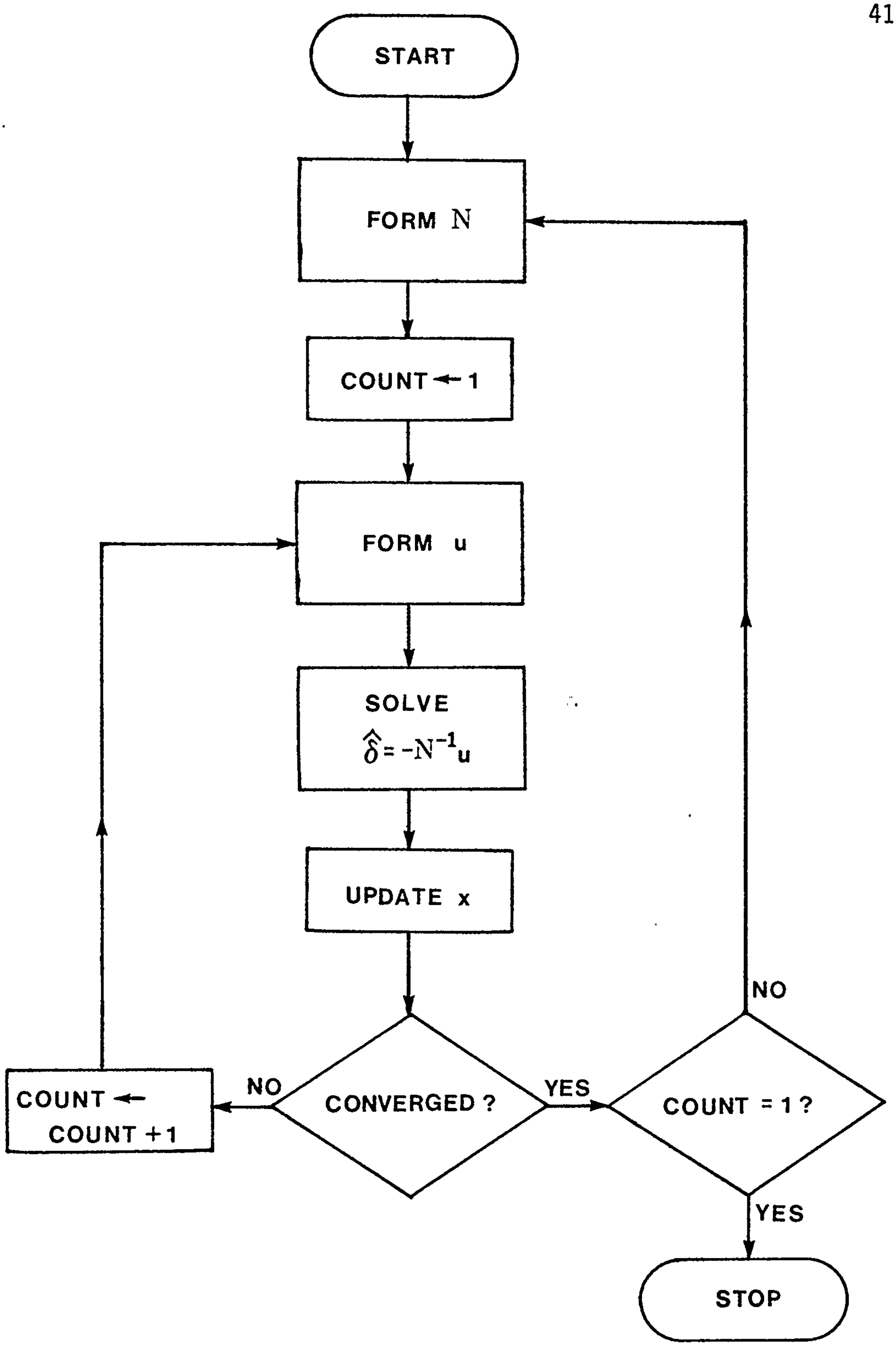


Figure 3.1

Iteration Reusing N^{-1}

Chapter 4

ANALYSIS OF RESULTS

The results of an adjustment or design must be analysed to ensure that they are correct and that the accuracy specifications have been met. The analysis of the results is at least as important as the observing and adjusting of the data and the same care and attention should be paid to it. It is only after a thorough analysis that they can be accepted as correct. Before applying the tests some general concepts are needed.

4.1 Concepts

The choice of the least squares method for the computation of the results is based on three assumptions. These are; (a) the observations are from a population with a normal distribution, (b) the accuracies (variances) of the observations are known or the relative accuracies of the observations are known, and (c) the mathematical model is correct, i.e. there are no unmodelled systematic or gross errors in the data. These three assumptions must be verified before the results can be accepted as correct.

The suitability of the network for the purpose it was established must also be examined. This requires an analysis of the accuracy of the results. The accuracy of the results can be expressed in several ways. The most useful is the relative accuracy between pairs of observation points. This may be given in terms of the relative confidence region or in terms of the accuracies of the computed distance and azimuth between pairs of stations (Section 4.3.2). The accuracy can also be expressed in terms of the point confidence regions.

The following section presents statistical tests and their application to the analysis of the observations. It should be fully understood that statistics is a blunt tool. The tests presented here will not give a clear yes-no answer to the hypothesis that the assumption being tested is valid. The testing is all done at some confidence level 1 - α. If α equals 0.05, the confidence level is 95%. This allows us to say, for example, that we are 95% confident the observation is a blunder if it fails the test given in Section 4.2.2. Note that in this case, about 5% of the observations rejected are not blunders. The chance of rejecting good data is decreased by decreasing α but this also increases the chances of accepting bad data. The choice of the confidence level to use in statistical testing must be made after careful consideration.

The results of the various statistical tests are interdependent. The first test presented is the chi-square goodness of fit test to ensure that the observations are each from a normal distribution. If this test fails then, theoretically, the second test for the detection of outliers can not be done as it requires that the distribution is known (i.e. normal). The problem is that the presence of outliers is one possible cause for the failure of the first test. All the tests should always be carried out and the results analysed together using not only the test results, but also a full understanding of the data being tested.

4.2 Assessment of the Observations

The tests presented in this section are used to verify that the three assumptions mentioned in the previous section have been met. The tests are presented without proof or derivation. Readers interested in probing deeper into these tests are referred to [Vanicek and Krakiwsky, 1982]. The purpose of the test, the computation of the test, and the interpretation of the test are presented for each test described.

4.2.1 X^2 Goodness of Fit Test

This test is used to verify that the observations are from a normal distribution. The standardised and pooled residuals (see Section 4.2.2) are grouped into histogram cells and the statistic y computed from

$$y = \sum_{i=1}^{c} \frac{(E_i - O_i)^2}{E_i} , \tag{4.1}$$

where c is the number of cells,

O_i is the observed frequency in cell i, and

E_i is the expected frequency for cell i.

The expected frequency is the area under the normal curve within the boundaries of the histogram cell. There must be at least five cells each with a minimum expected frequency of five for this test to be valid [Blank, 1980]. The expected frequencies of the first and last cells should include the left and right tails under the normal curve respectively.

The test is

$$y < \chi^2_{c-1,1-\alpha} \tag{4.2}$$

if the variance factor $\hat{\sigma}_1^2$ is known, or

$$y < \chi^2_{c-2,1-\alpha} \tag{4.3}$$

if σ_0^2 is unknown and the estimated variance factor $\hat{\sigma}_0^2$ used. $\chi^2_{\nu,1-\alpha}$ is the abscissa value from the chi-square distribution with ν degrees of freedom at the 1-α confidence level.

This test can fail for any of the following reasons:

1. The observations are not normally distributed;
2. There may be systematic or gross errors remaining in the data; or
3. The wrong variances (and covariances) were assigned to some or all of the observations.

If the test passes, we can conclude, at the 1-α confidence level, that the observations are normally distributed.

4.2.2 Detection of Outliers

The purpose of this test is to detect any observations that are blunders. This test requires that the distribution of the observations is known. The paramaters of the distribution may have been estimated.

The statistic y is

$$y = \frac{\hat{r}_i}{\sigma_{\hat{r}_i}} \quad , \tag{4.4}$$

which is simply the standardised residual. The standard deviations of the residuals, $\sigma_{\hat{r}_i}$, are the square roots of the diagonal elements of $C_{\hat{r}}$. This matrix is not generally computed, as it is much too large (n by n) to compute. The usual (incorrect) practice is to replace $\sigma_{\hat{r}_i}$ with the standard deviation of the observation, σ_{ℓ_i}, but a better approximation

$$\sigma_{\hat{r}_i} \simeq \left[\left(\frac{n - u}{n} \right) \frac{\hat{\sigma}_0^2}{\sigma_0^2} \; \sigma_{\ell_i}^2 \right]^{\frac{1}{2}} \tag{4.5}$$

is given by Pope [1976] for adjustments of models of the form $f(x) - \mathcal{L} = 0$. This approximation can only be used if the apriori variance factor, σ_0^2, is known. If σ_0^2 is not known the σ_{ℓ_i} must be used.

An observation is rejected if

$$-n_{(1-\frac{\alpha}{2})} < \frac{r_i}{\sigma_{\hat{r}_i}} < n_{(1-\frac{\alpha}{2})} \tag{4.6}$$

for the case of variance factor known, or

$$-\tau_{\nu,(1-\frac{\alpha}{2})} < \frac{\hat{r}_i}{\hat{\sigma}_{\hat{r}_i}} < \tau_{\nu,(1-\frac{\alpha}{2})} \tag{4.7}$$

if the variance factor was estimated. In these tests $n_{(1-\frac{\alpha}{2})}$ is the abscissa value of the standard normal distribution and $\tau_{\nu,(1-\frac{\alpha}{2})}$ is the abscissa value of the tau distribution [Pope, 1976] with ν (=n - u) degrees of freedom.

An observation may be rejected for any of the following reasons:

1. The observation is a blunder;
2. The observation is not from a normally distributed population; or
3. The wrong (too small) standard deviation was assigned to the observation.

The test can be done either "in context" or "out of context". [Vanicek and Krakiwsky, 1982]. In context testing has a wider acceptance range than out of context; that is, for a given set of data, fewer observations would be rejected using in context testing than when using out of context testing. In context testing implies that the observation being examined is a part of a sample taken from the population while with out of context testing, the observation is treated as being a sample of one. The reason for the wider acceptance range when using in context testing is that the probability of one of the sample values being far from the mean increases with the sample size. The test just presented is for out of context testing. To convert is to an in context test, simply replace α with α/n where n is the number of observations being tested in the context of each other.

4.2.3 Test of the Variance Factor

This test is used to verify that the scale of the C_ℓ matrix was in fact σ_0^2. The test is

$$\frac{\sigma_0^2}{\nu}\,\chi^2_{\nu,\frac{\alpha}{2}} < \hat{\sigma}_0^2 < \frac{\sigma_0^2}{\nu}\,\chi^2_{\nu,(1-\frac{\alpha}{2})} \quad , \qquad (4.8)$$

where σ_0^2 is the apriori variance factor,

$\hat{\sigma}_0^2$ is the estimated variance factor,

ν is the degrees of freedom of the adjustment, and

χ^2 is the abscissa value of the χ^2 distribution.

This test may fail for any of the following reasons:

1. The apriori variance factor was incorrect;
2. The observations were not normally distributed;
3. There may be systematic or gross errors in the observations;
4. An incorrect math model was used; or
5. The observations were correlated but the correlations were neglected.

The last two causes of failure listed are not normally encountered in least squares adjustment of survey networks. The math models used are well defined and the observations are uncorrelated or very weakly correlated, with the exception of coordinate-difference and coordinate observations. The correlations of the coordinate-difference and coordinate observations are not normally neglected.

4.3 Assessment of the Paramaters

The covariance matrix of the results, $C_{\hat{x}}$, fully describes the accuracy of the coordinates determined from a least square adjustment. It is difficult, if not impossible, to interpret the contents of the

C_x matrix directly in any meaningful way. For this reason, the accuracy of an adjustment or design is analysed using quantities computed from the C_x matrix. The confidence ellipse is the most important of these derived quantities.

The confidence region is an elliptically shaped area centered on the estimated paramaters. The confidence ellipse is the outer boundary of the confidence region. The paramaters of the confidence ellipse (Figure 4.1) are the semi-major axis (a), the semi-minor axis (b), and the azimuth of the semi-major axis (θ).

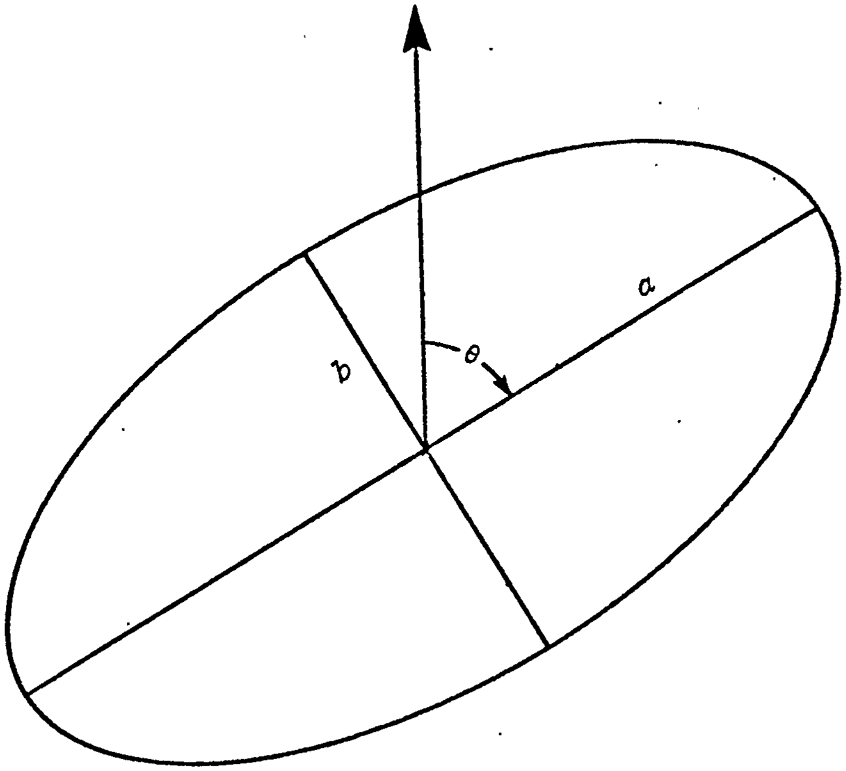


Figure 4.1

Confidence Ellipse

For horizontal networks, there is a 39% probability that the true coordinates of the station lie within the boundaries of the standard confidence ellipse. This is not a suitable confidence level for most work. The confidence level can be increase to 1 -α by multiplying the axes a and b by an expansion factor f where

$$f = (\chi^2_{2,1-\alpha})^{\frac{1}{2}} \tag{4.9}$$

if the apriori variance factor (σ_0^2) is known, and

$$f = (2F_{2,\nu,1-\alpha})^{\frac{1}{2}} \tag{4.10}$$

if the variance factor was estimated. The "2" in Equations 4.9 and 4.10 is the dimensionality of the problem and would be equal to one and three for vertical and three dimensional networks, respectively. The distribution used in Equation 4.10 is the Fisher distribution with two and ν degrees of freedom. It degenerates to the χ^2 distribution as the degrees of freedom of the adjustment (ν) approaches infinity.

4.3.1 Point Confidence Ellipses

Point ellipses are used to indicate the accuracy with which the coordinates of the stations have been determined. Sometimes the term absolute is used but it is a misnomer. The point ellipses show the accuracy with which the position has been determined relative to the coordinate system implied by the network [Vanicek and Krakiwsky, 1982]. If there are fixed (known and errorless) stations in the network, they define the coordinate system of the computed positions and the point ellipses simply reflect the accuracy of the computed positions relative to the fixed stations.

The paramaters of the point ellipse are [Richardus, 1974].

$$a = \tfrac{1}{2}\left\{(\sigma_N^2 + \sigma_E^2) + [(\sigma_N^2 - \sigma_E^2)^2 + 4\sigma_{NE}^2]^{\frac{1}{2}}\right\}^{\frac{1}{2}} . \qquad (4.11)$$

$$b = \tfrac{1}{2}\left\{(\sigma_N^2 + \sigma_E^2) - [(\sigma_N^2 - \sigma_E^2)^2 + \sigma_{NE}^2]^{\frac{1}{2}}\right\}^{\frac{1}{2}} , \qquad (4.12)$$

$$\theta = \tfrac{1}{2}\, tan^{-1}\left(\frac{-2\sigma_{NE}}{\sigma_E^2 - \sigma_N^2}\right) , \qquad (4.13)$$

where σ_N^2 is the variance of the northing,

σ_E^2 is the variance of the easting, and

σ_{NE} is the covariance between the northing and easting,

all taken from the $C_{\hat{x}}$ matrix. The calculation of θ frequently causes problems. θ is a clockwise angle from north while the arctan function on most computers returns the counter-clockwise angle from east. To obtain the correct θ subtract the value calculated using Equation 4.13 from 90° and use the FORTRAN ATAN2 function (or its equivalent) when evaluating equation 4.13.

4.3.2 Relative Confidence Ellipses

Relative confidence ellipses are used to indicate the accuracy that one station has been determined to, relative to another station. This is a generally more useful measure of the accuracy of a network. The Canadian government uses the ratio of the semi-major axis and the distance between the stations to classify a network as first, second or third order [Energy, Mines and Resources, 1978].

The relative ellipse is calculated using the covariance matrix of the coordinate differences computed from the adjusted coordinates. The covariance law is applied to the $C_{\hat{x}}$ matrix to get $C_{\Delta\hat{x}}$ as

$$C_{\Delta\hat{x}} = G\, C^{*}_{\hat{x}} G^{T} \quad , \tag{4.14}$$

where

$$G = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad , \tag{4.15}$$

and $C^{*}_{\hat{x}}$ is the 4 by 4 submatrix of $C_{\hat{x}}$ corresponding to the two stations of interest.

The computed $C_{\Delta\hat{x}}$ matrix that results is

$$C_{\Delta\hat{x}} = \begin{bmatrix} \sigma^2_{\Delta N} & \sigma_{\Delta N \Delta E} \\ \sigma_{\Delta N \Delta E} & \sigma^2_{\Delta E} \end{bmatrix} \tag{4.16}$$

$$= \begin{bmatrix} \sigma^2_{N_1} - 2\sigma_{N_1N_2} + \sigma^2_{N_2} & \sigma_{N_1E_1} - \sigma_{N_1E_2} - \sigma_{E_1N_2} + \sigma_{N_2E_2} \\ \sigma_{N_1E_1} - \sigma_{N_1E_2} - \sigma_{E_1N_2} + \sigma_{N_2E_2} & \sigma^2_{E_1} - 2\sigma_{E_1E_2} + \sigma^2_{E_2} \end{bmatrix} .$$

The paramaters of the relative ellipse are computed using Equations 4.11 to 4.13 replacing σ^2_N with $\sigma^2_{\Delta N}$, σ^2_E with $\sigma^2_{\Delta E}$ and σ_{NE} with $\sigma_{\Delta N\Delta E}$.

Two other values that are often useful are the accuracy of a distance or azimuth computed from the adjusted coordinates. The variance of a computed distance is [Richardus, 1974]

$$\sigma_S^2 = \sigma_{\Delta N}^2 \cos^2 \alpha + 2\sigma_{\Delta N \Delta E} \sin \alpha \cos \alpha + \sigma_{\Delta E}^2 \sin^2 \alpha , \tag{4.17}$$

and the variance of the computed azimuth is

$$\sigma_\alpha^2 = \frac{1}{S^2} (\sigma_{\Delta N}^2 \sin^2 \alpha + 2\sigma_{\Delta N \Delta E} \sin \alpha \cos \alpha + \sigma_{\Delta E}^2 \cos^2 \alpha) , \tag{4.18}$$

where S is the computed distance, and

α is the computed azimuth between the stations.

The analysis equations presented in this chapter have all been included in CANDSN, making it a complete package for the design, adjustment, and analysis of a network. The tests presented in Section 4.2 are only applicable to the results of an adjustment while the accuracy representation presented in Section 4.3 is applicable to the analysis of both the design and adjustment of a network.

Chapter 5

HARDWARE AND SOFTWARE

The hardware components and program libraries used for CANDSN are presented in this chapter. The first section is a discussion of the hardware, its features, and its limitations. The hardware is split into two principal groupings (Figure 5.1). The first group is the computer and its peripherals, while the graphics devices are in the second group. Section 5.2 consists of a discussion of the program libraries used. All extensions to the ANSI FORTRAN standard used are described with an explanation of how to revise the programs for use on computers without these extensions.

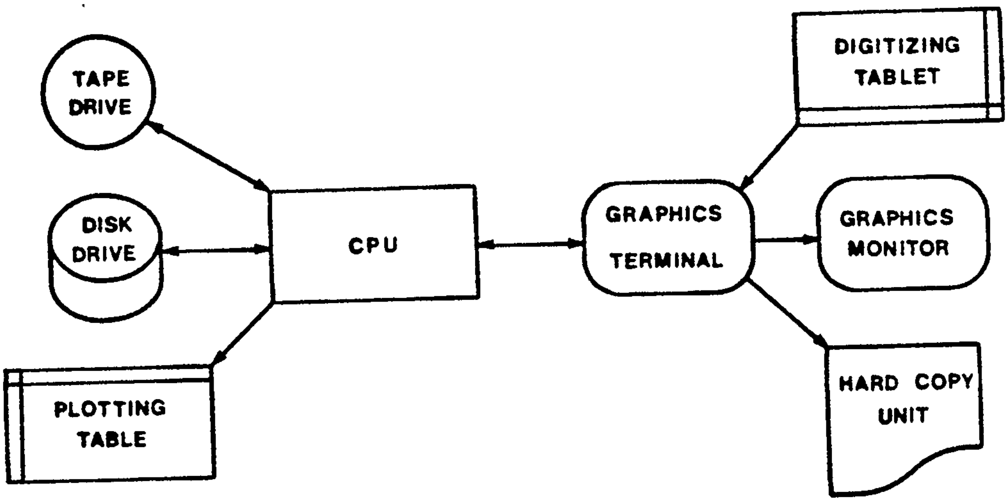


Figure 5.1

Hardware Components for CANDSN

5.1 Hardware

The computer used for the implementation of CANDSN is a Digital Equipment Corporation (DEC) PDP 11/23. The 11/23 is a 16 bit minicomputer with a direct addressing capability of 64K bytes. The computer used for CANDSN uses the DEC MMU option to expand the memory to 192K bytes. Only 54K bytes are available for program code and 128K bytes for data storage. The remaining portion of memory and address space is used by the operating system and peripherals. The KEF 11 floating point option provides 32 and 64 bit floating point arithmetic. Without this option, floating point arithmetic is still possible but must be done with software resulting in a much slower execution time and a slightly larger program. The operating system is RT-11. This is the smallest operating system offered by DEC that will allow both program development and execution.

This combination of CPU and processing options is the minimum that can be used for CANDSN. Installation of the program on a smaller computer could only be accomplished if the limits described in Section 6.1.4 were to be drastically reduced and the program totally restructured. If a computer with a slower execution speed was used, the time delay between the issuance of a command and the presentation of the results would become annoying. Installation of CANDSN on a faster or bigger computer (e.g. a VAX 750) would enhance the utility of the program considerably and allow further extensions to its capabilities and the size of problems that could be solved.

There are four peripheral devices of the 11/23 that are used by CANDSN. The first of these is a VT100 video terminal configured as the operator's console for the 11/23. It is used to issue the commands to the operating system to start the program (see Appendix I) and as an output device for large amounts of text. The graphics device described later in this section does not have a good text presentation ability. For this reason, long listings such as lists of all stations and their coordinates are sent to the VT100.

The second peripheral is the RL02 disk system. This is a mountable single platter hard disk with a 10 Megabyte capacity. All the system programs and the CANDSN program are kept on this disk. It is also used for all permanent and temporary files created by and for CANDSN. The third peripheral used is a standard nine-track tape drive. It is only used for the checkpoint files described in Section 6.1 This is not an essential item of equipment as the files can also be placed on the hard disk. On single disk system, the tape drive is essential as a backup device.

The last peripheral is a printer. The printer used in the current configuration is a DEC LA100. This is a dot matrix character printer with a speed of 240 cps (characters per second). Experience with a 30 cps printer has shown that it is not suitable. All printing in CANDSN is done online and the use of a slow printer introduces delays of up to 30 minutes.

The Princeton Electronic Products (PEP) 8500M intelligent graphics terminal is the heart of the CANDSN hardware. The 8500M has four components (Figure 5.2). The display processor uses a local microprocessor and special purpose hardware to generate the graphics image. It also controls all input to and output from the terminal. The image memory stores the graphics image and generates the video signal for the monitor. The memory is an analogue memory and cannot be written to, except via the display processor. It is not possible to read the contents of the image memory. The keyboard is a standard typewriter style keyboard with the addition of several special function keys and a joystick for graphics use. The monitor is a high resolution monochrome video display tube. It provides for 32 grey levels in the display and has a resolution of approximately 1024 by 1024 pixels.

One feature of the 8500M essential to CANDSN is the ability to erase selected lines on the screen. This allows the pages of the menu (Section 6.1.1) to change without disturbing the rest of the image. Lines connecting stations can also be immediately deleted when observations are removed, thus maintaining a current picture of the network at all times. This process is too slow to use if large areas of the screen need to be cleared. In these cases, it is faster to clear and redraw the entire screen. This disadvantage could be overcome if a terminal using a frame buffer system was used [Newman and Sproull, 1979].

There are three peripherals connected to the 8500M. The most important of these is the Summagraphics digitising tablet. The tablet

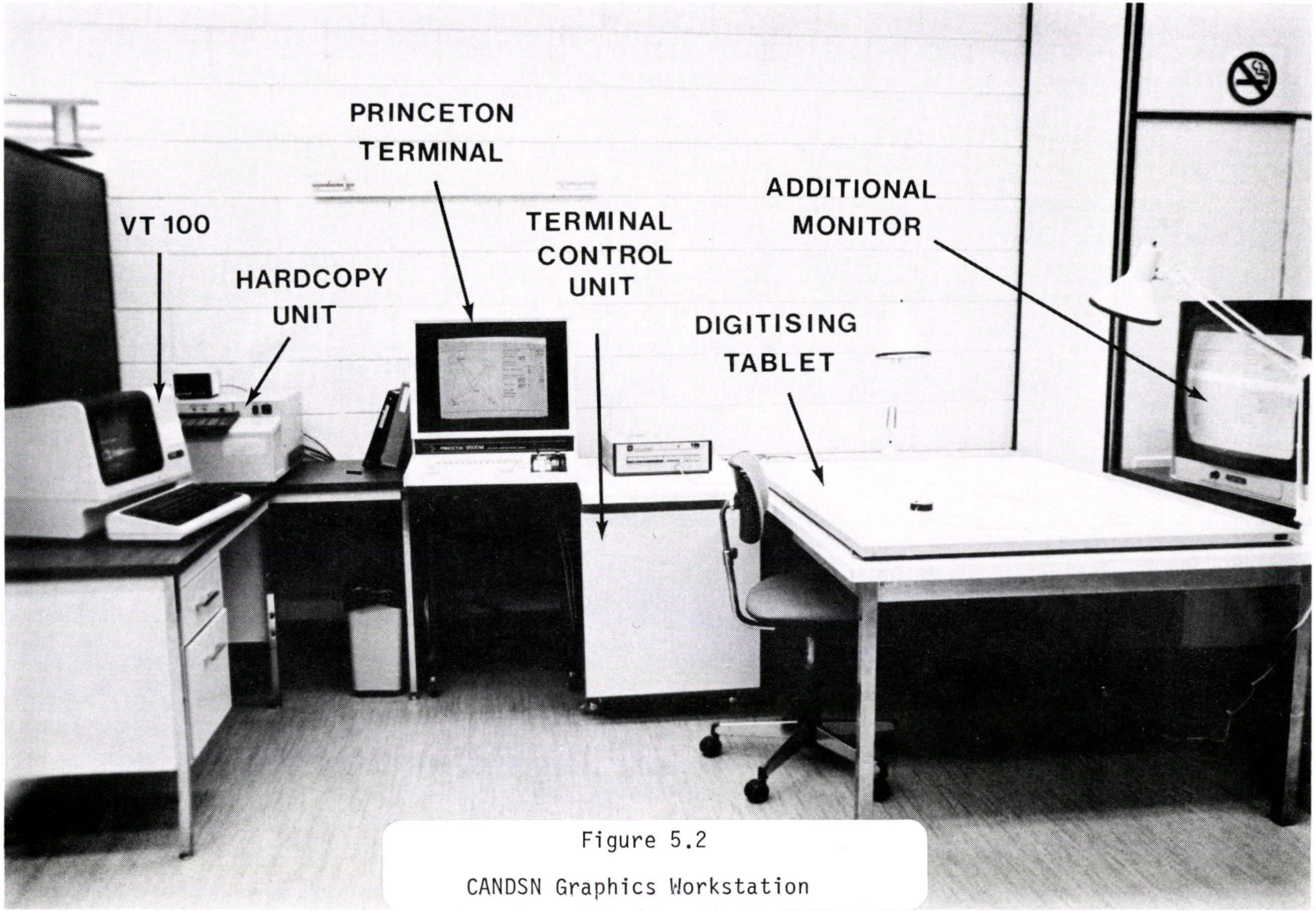


Figure 5.2

CANDSN Graphics Workstation

is 0.91 by 1.22 m (36" by 48") with a resolution of 0.15 mm. It is used to allow coordinate input directly from a map or plan and can also be used to enter menu commands. The tablet, while not essential to CANDSN, greatly enhances and simplifies the design of a network using CANDSN.

A second graphics monitor is also connected to the 8500M. It is useful when demonstrating the design process to large groups such as classes. It also is convenient when using the digitising tablet as it can be placed directly behind the tablet so that the design can be viewed without turning around. Up to four of these monitors can be connected to the graphics terminal.

The final peripheral to the graphics terminal is a Tektronix 4634 hard copy unit. This unit produces a paper copy of whatever is showing on the graphics monitor. It is convenient for keeping a record of intermediate design stages but the image quality is not sufficient to use for final results. The image is reduced to 15 cm by 20 cm (6" by 8") and most fine detail is lost. It is also expensive to use, costing approximately $0.50 for each copy.

The final hardware item in the configuration is the DataTech DT3454 flatbed plotter. This is a four pen plotter with a drawing area of 0.86 m by 1.37 m (34" by 54"). The plotter has an accuracy of 0.1 mm over the entire plotting range. The plotter is used to plot the final results of a design or adjustment at any scale with confidence regions and design accuracies shown. The plotter is connected directly to the PDP 11/23.

These hardware components were assembled by the Division of Surveying Engineering over a period of three years for this and other projects. They have proven satisfactory for this project, but are certainly not the only devices that could be used for CANDSN. The range of computer and graphics hardware available is growing rapidly. Anybody interested in installing CANDSN would be well advised to shop around to see what is currently available and assemble their own hardware configuration.

5.2 Software

All the routines written for CANDSN are written in FORTRAN with the exception of the 8500M - 11/23 interface which was written in MACRO-11. Three extensions to the ANSI FORTRAN standards were used in the programming. The first of these is the use of the VIRTUAL data type. This is a DEC extension which enables the use of memory beyond the 16 bit addressing range of the PDP-11. To install CANDSN on a computer without the VIRTUAL data type, simply replace the word VIRTUAL in the data declaration statements with the word DIMENSION.

The second extension is the use of the BYTE data type. This is used in two ways. The first is for the storage of character data. The replacement for byte data on another computer would probably be to declare it as CHARACTER*1 but this is highly dependent upon the manufacturer's implementation of FORTRAN. The second use of single byte data in CANDSN is for the storage of large tables of small integers.

The easiest replacement for this on another computer is to simply declare the tables as INTEGER if there is sufficient memory.

The third extension is the use of ASCIZ character strings. These are character strings that are terminated with a null byte. All the character string manipulation routines use this data type. Note that the strings are not declared as ASCIZ but as arrays of BYTE data with one extra element for the null byte.

In addition to these extensions, several of the DEC library routines are used. They are all character string manipulation routines with the exception of subroutines DATE and TIME, which are used to get the current date and time from the operating system. These routines are listed in Appendix II and will have to be replaced by the appropriate routines for the computer system being used.

The CANDSN software package consists of 143 subroutines grouped into eight libraries. A list of the subroutines and their purpose is given in Appendix II. The listings of the individual routines are not included in this thesis but are in External Appendix I. This large number of routines is a result of the initial decision to write CANDSN with a modular structure. This was essential as segments of the program must overlay each other in memory in order to fit as large a program as CANDSN into the approximately 54K bytes available for program storage on the PDP 11/23.

Of the eight libraries, six were written as a part of CANDSN. The graphics utility libraries for the 8500M and the DT3454 were provided by the manufacturers, Princeton Electronic Products and

DataTechnology Ltd. respectively, and are not included in the external appendix as they are copyrighted. Modifications to these libraries have been minor and consisted mostly of changes to take advantage of special features of the PDP 11/23. The six libraries written for CANDSN are presented in the rest of this chapter.

5.2.1 Application Library

The application routines are those routines including the main programs that were written to perform functions specific to CANDSN. They are all special purpose routines that use the routines in the other seven libraries to perform their function. Each menu page has an application routine associated with it that may call other application routines.

5.2.2 Panel Matrix Library

The collection of subroutines that are used to manipulate the panel matrices described in Section 3.3.2 are in this library. They include routines to compute the Cholesky root and inverse of a matrix, to compute the forward and backward solutions, and to multiply two panel matrices. Also included are utilities to print the contents of a panel file, read and write records, and initialise a new panel file. The library is intended to be a complete package that could easily be used in any application requiring the solution of large sets of linear equations. All routines in the library have names that end with PNL.

5.2.3 Station File Library

All the information pertaining to a single station with the exception of its covariance information is stored in a direct access file referred to as the station file. This includes its name, coordinates, elevation, gravity information, and status. A set of linked list pointers is also kept to allow the sequential processing of the data by both name and sequence number. All the routines used to store, retrieve, and maintain the data in this file are in the station file library. All routines in the library have names that end with STF.

5.2.4 Observation File Library

All observation data with the exception of coordinate and coordinate-difference observations are kept in a direct access observation file. Coordinate and coordinate-difference observations are maintained in special files because their covariance matrices are fully populated, and are processed with special application routines. The observation file is currently set up for distances, azimuths, directions, and angles, but can be easily extended to include any other uncorrelated observation types. The observation file library contains all the routines used to store, retrieve, and maintain the records contained in the observation file. All routines in the library have names that end with OBF.

5.2.5 Menu Library

The routines in the menu library are used to control all aspects of menu processing. This includes menu display, decoding meno commands, storing and retrieving response values and printing the menu pages for use on the digitiser. Included in this library is subroutine GET.RSP which is the sole routine through which input is received from the 8500M whether it be a menu command, a digitised point, or a text string. The routines in this library are heavily dependent on the routines in the 8500M library.

5.2.6 Utilities Library

The utilities library contains all the general purpose utility routines. These include routines for matrix manipulation, string manipulation, sorting, statistical analysis, and the myriad other things that need doing in a program the size of CANDSN.

All the libraries, with the exception of the application library, have been written as general purpose libraries. The intention is that they can be used in any other application program that needs them.

Chapter 6

CANDSN

Program CANDSN (Computer Aided Network DeSigN) was written to implement the components of survey network design and adjustment discussed in the previous chapters. The features of the program are presented in the first section of this chapter. This is followed by a presentation of the results of some sample networks designed and adjusted with CANDSN.

6.1 Features of CANDSN

The primary consideration in the design of the program was that it be "user friendly" (Section 5.2). One of the implications of this is that erroneous input must be processed in a reasonable manner. In the case of CANDSN, the program tries to correct the error itself and informs the user of what it has done. If this is not possible, it will give an error message and ignore the bad input. All the commands have an escape option that will allow the user to cancel what has been done. This is particularly useful if the user has inadvertently chosen the wrong menu page. In addition to the escape clause, the user initiates any action only after all options and input data are displayed on the screen.

Typing has been kept to a minimum. All commands are entered using graphical digitising from the menu. For those commands that have only a few possible responses, the program cycles through the possible responses, displaying them on the screen one at a time.

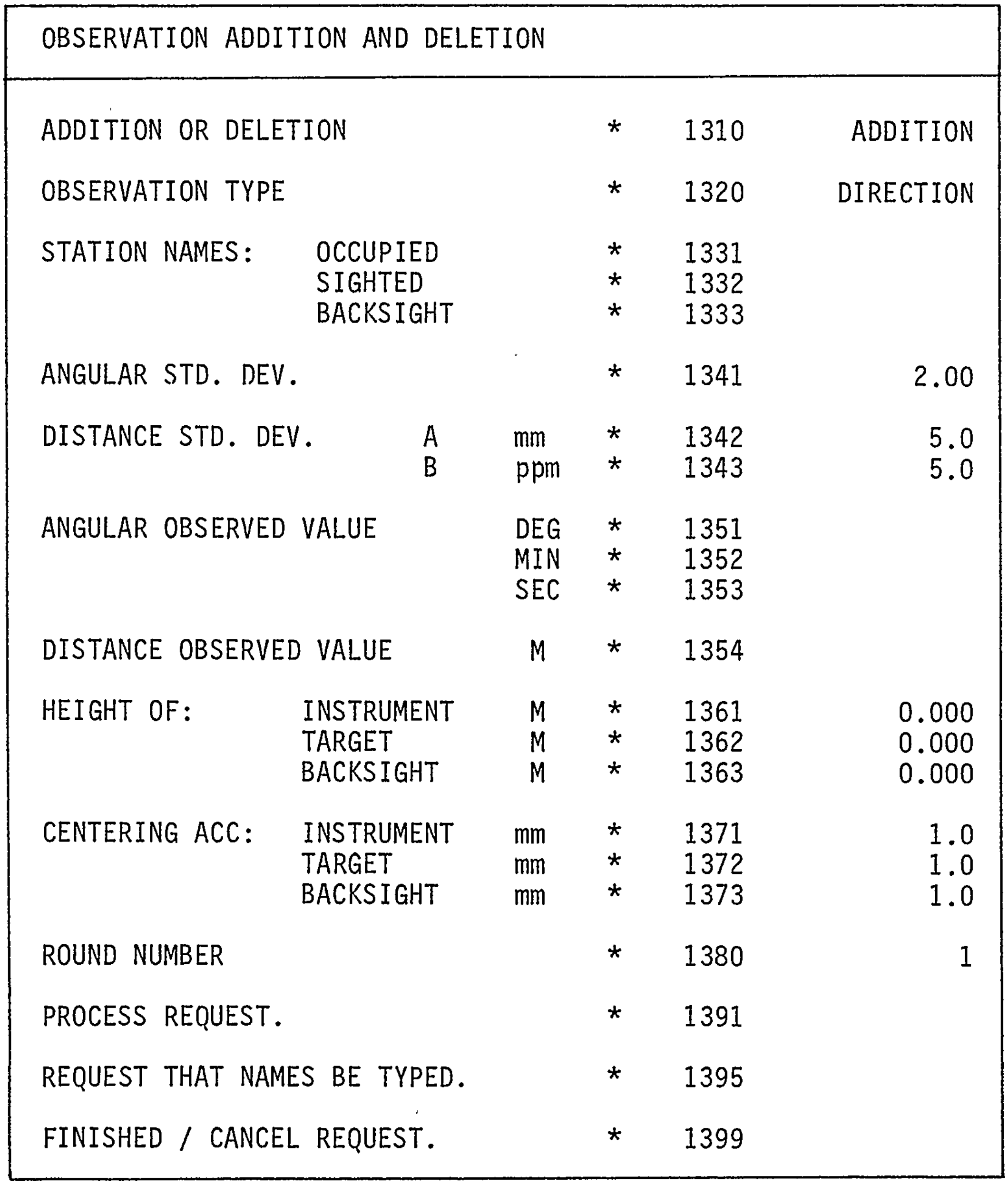
The user keeps selecting the menu command until the correct response is displayed. Wherever possible, default values are provided for the responses. Once the default has been changed by the user, that value becomes the default until changed again by the user. All text input is checked for validity. If a numeric value must be entered, but character data is received, the program erases the input from the screen, prints a message explaining that the input must be numeric and re-enters the input mode.

The user is prompted for all input. The two input prompts are the graphics cursor and the blinking text cursor. If the program is not waiting for input, no cursor is displayed on the screen and input is disabled. The input modes (graphics and text) can be switched at any time by the user by simply hitting the return key. Note that due to a restriction of the 8500M terminal, digitiser tablet input is only possible when in the text mode.

The program is designed to be self-teaching. An experienced user should be able to quickly use the full power of the program, by simply using it for a short period of time. All the features are displayed on the screen and the user can choose any of them. The command for each feature is self-explanatory.

6.1.1 Menu Driven

The program is completely menu driven. The only commands that need to be entered at the computer console are those used to start the program (see Appendix I). The menu is divided into pages (Figure 6.1) with a



```
OBSERVATION ADDITION AND DELETION

ADDITION OR DELETION                          *   1310      ADDITION

OBSERVATION TYPE                              *   1320     DIRECTION

STATION NAMES:    OCCUPIED                    *   1331
                  SIGHTED                     *   1332
                  BACKSIGHT                   *   1333

ANGULAR STD. DEV.                             *   1341          2.00

DISTANCE STD. DEV.       A      mm            *   1342           5.0
                         B      ppm           *   1343           5.0

ANGULAR OBSERVED VALUE          DEG           *   1351
                                MIN           *   1352
                                SEC           *   1353

DISTANCE OBSERVED VALUE          M            *   1354

HEIGHT OF:       INSTRUMENT      M            *   1361         0.000
                 TARGET          M            *   1362         0.000
                 BACKSIGHT       M            *   1363         0.000

CENTERING ACC:   INSTRUMENT     mm            *   1371           1.0
                 TARGET         mm            *   1372           1.0
                 BACKSIGHT      mm            *   1373           1.0

ROUND NUMBER                                  *   1380             1

PROCESS REQUEST.                              *   1391

REQUEST THAT NAMES BE TYPED.                  *   1395

FINISHED / CANCEL REQUEST.                    *   1399
```

Figure 6.1

Typical Menu Page

specific function (i.e. add/delete observations) assigned to each page. The first page is a master page used to control the selection of function pages. Each function page shows all the data required to execute the function. The data is grouped according to its natural relationships.

The data may be changed by simply moving the graphics cursor to the item to be changed and pressing the enter key. The user is then prompted for the new data value or the next value in the sequence is displayed. Grouped data may be changed as a group or individually. If the group is to be changed, the cursor is positioned over the menu text line and the user is prompted for the data values, one at a time, for the entire group. If only one value is to be changed, the cursor is placed on the value to be changed, and the user is prompted for only the single value.

The currently active menu page is always displayed on the right third of the screen. This is changed by the program whenever appropriate. At the user's option, all, some, or none of the menu pages may be used from the digitiser tablet. These pages are usually mounted during the initial setup stage of the program. They can, however, be mounted at any time the user wants. The only restriction on the use of the digitiser tablet is that only selections on the currently active page (as displayed on the screen) may be made. Any other menu selection will be considered as an invalid response.

6.1.2 Graphical Input

Wherever possible, CANDSN allows the use of graphical input. This input may be from either the graphics terminal or from the digitising tablet. Note that to use graphical input from the terminal, it must be in graphics mode, while it must be in text mode to use the digitising tablet. Positive feedback is used to indicate the successful digitising of a point. On the terminal, the cursor is erased as soon as a point is digitised while the digitising tablet beeps to indicate a successful point selection.

The use of graphical input for menu command selection has already been discussed. It is also used for station identification and coordinate input. As an example, when station identification is required, it is not necessary that the name of the station be typed. All that is necessary is that the cursor be moved close to the station required, on the screen, and the "enter" key pressed. The program will then find the station closest to the graphics cursor and use it.

During station addition, the coordinates of the digitised point are used as the approximate coordinates of the station. This method should not be used for entering the coordinates of known stations. The resolution of the digitser tablet is only 0.12 mm which corresponds to an inaccuracy of 6.35 metres, if digitising from a 1:50,000 map. Add to this the pointing error of the user and it is possible to be out by up to 50 metres causing unacceptable errors and distortions in the results.

6.1.3 The Checkpoint Function

Frequently, the design of a network takes place over a long period of time. Changes to an already designed network may become necessary, or it may not be possible to complete the design in one run of the program. The use of CANDSN to adjust a network that was originally designed on it will be a common occurrence.

To facilitate these operations, CANDSN has a checkpoint feature. The save command dumps all the current information to a file. The device and file name are specified by the user. The possible devices are magnetic tape, hard disk (RL02), or floppy disk (RX02). Of these, the magnetic tape is the preferred device. The RL02 disk is not a secure device for long term storage as it has no restrictions on its access. The RX02 floppy disk is a small device and would not have enough room for a large network.

The restore command is used to read the information from the checkpoint file. This places CANDSN into exactly the same condition it was in when the save command was executed, with the sole exception of items mounted on the digitiser tablet. Since there is no guarantee that the map and menu will be mounted in exactly the same place as they were when the save command was used, the current digitiser position information is maintained.

The checkpoint operations can also be used during a CANDSN run. If during a design, one possible solution is found, it can be saved before further changes are made in an effort to find a better solution. This way, if no better solution can be found, the original solution can be easily restored without having to undo the new modifications.

6.1.4 Program Limits

There were two factors which made it necessary to limit the size of the networks that could be designed and adjusted with CANDSN. The most critical of these was the memory available on the PDP 11/23. In the interest of speed of execution, some information contained in the observation and station files is also maintained in core in the form of tables. The size of these tables is limited by the available core memory. The same is true of some work matrices and vectors used in the evaluation of the sequential update expressions. The second factor is the FORTRAN requirement that a maximum size for a direct access file be declared when the file is created.

The limits established in CANDSN are:

a. A maximum of 100 stations may be in the network at any one time. There is no limit on the numbers of unknown, weighted, or fixed stations so long as the total does not exceed 100.

b. A maximum of 1000 observations may be in the network at any one time.

c. A maximum of 20 directions may be in a single round.

These limits are all established in the main program. If the limits are changed, no subroutines will require changes. The limit on the number of observations can be easily increased. It was established only because of the requirement that the maximum size of a direct access file be declared when the file is created. To increase the maximum, simply change the value of MXNOBS in the main program. This will have no effect on the memory requirements for CANDSN.

The other two limits can only be increased if there is more core memory available for the program. The matrices and tables whose dimensions are functions of these limits are clearly identified in the main program.

6.2 Examples of Results

Three examples of networks designed using CANDSN are presented in this section. They illustrate the use of CANDSN in three different fields of surveying. These examples were set up to illustrate the propagation of error in survey networks and to demonstrate some of the capabilities of CANDSN.

6.2.1 Lot Survey

The first example is a lot survey (Figure 6.2) in which two existing points 1500 metres apart are used to control the survey. Station 3 is positioned by interlining between stations 1 and 2 and measuring a distance to station 1. From here a traverse is run to position the corners of the lot. Stations 1 and 2 are not shown in Figures 6.2 and 6.3. The plotting area on the graphics terminal was purposely set so that only the region including stations 3 to 7 were

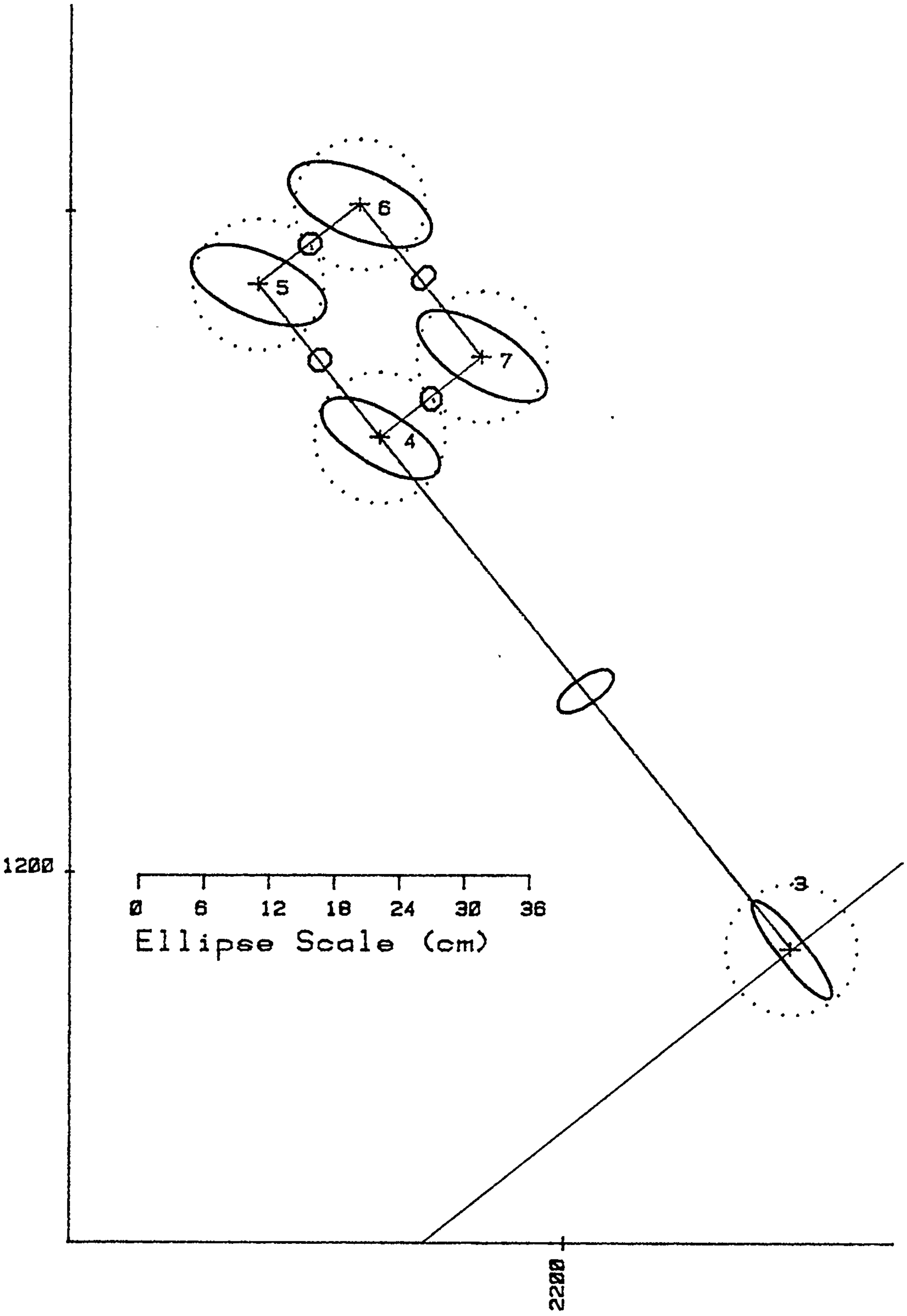


Figure 6.2

Initial Lot Survey

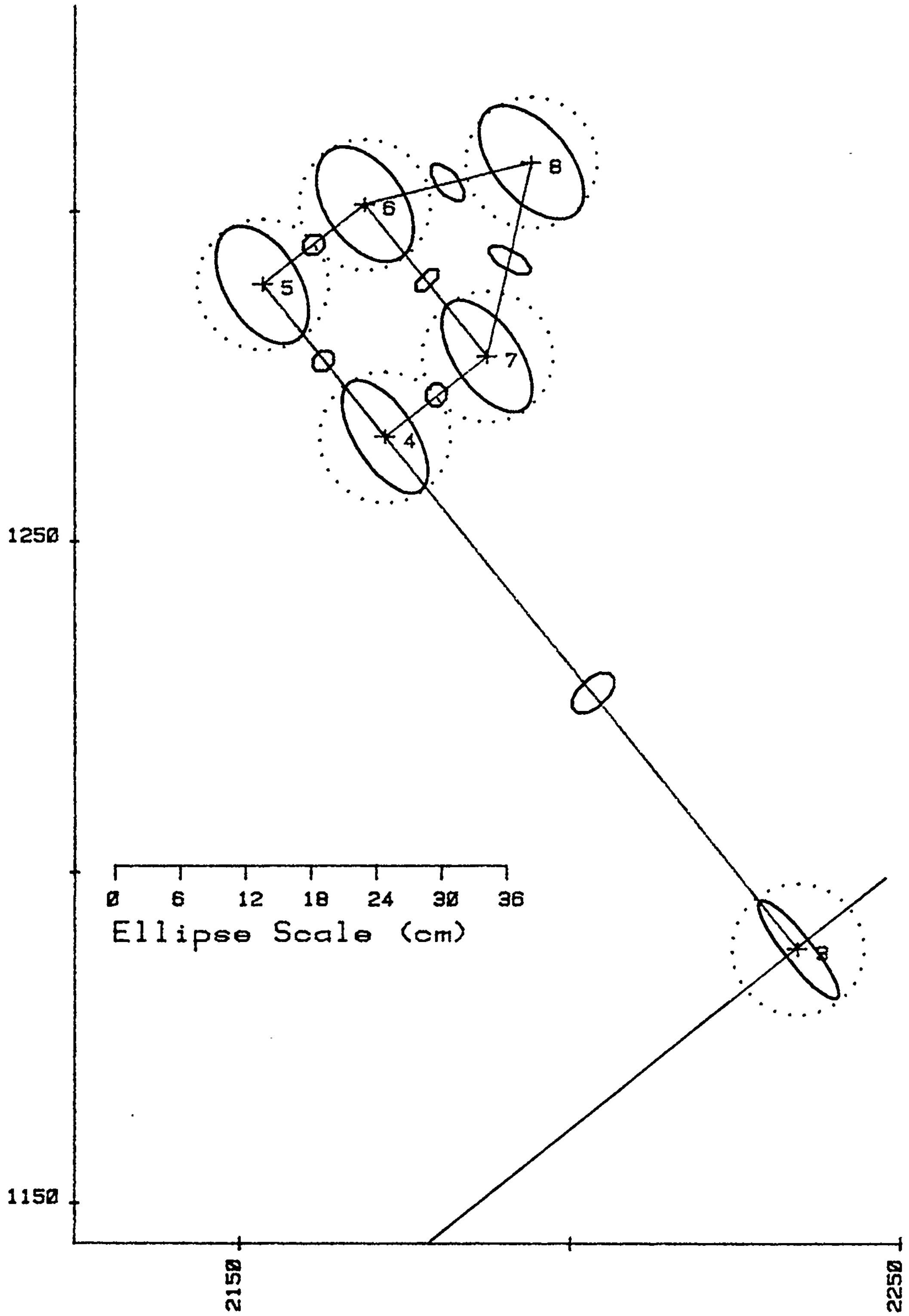


Figure 6.3

Updated Lot Survey

visible. If it had been set to include stations 1 and 2 the other stations would have been much too crowded. The last example illustrates this "windowing" process further.

Examining Figure 6.2 we notice that the station ellipses are all oriented at a slant to line 3 - 4. This is unexpected in a survey with all right angles. Further examination shows that the major axis of all the station ellipses is roughly perpendicular to the line drawn from station 1 to the unknown station. This indicates an uncertainty in the azimuth control. In an attempt to correct this, the angle 1 - 3 - 4 was removed and angle 2 - 3 - 4 substituted. The results of using the longer backsight for the initial angle are shown in Figure 6.3. This simple change in procedure resulted in a decrease of the semi-major axis of the station ellipse of point 6 from 7 cm to 5.9 cm and a greater uniformity in the size of the station ellipses. Station 8 was also added to the network when the angle at 3 was changed. The ellipse at 8 is too large indicating that further work is required to get the revised network to meet the specifications.

6.2.2 Geodetic Network

An example of a small scale geodetic network is shown in Figure 6.4. The initial solution was done considering only station 1 as known. Inspection of the resulting confidence regions shows a definite lack of azimuth control. This is typified by the major axes of the station ellipse being perpendicular to the line from the unknown station to the fixed station and the major axes of the relative ellipses being perpendicular to the line between the two unknown stations. This

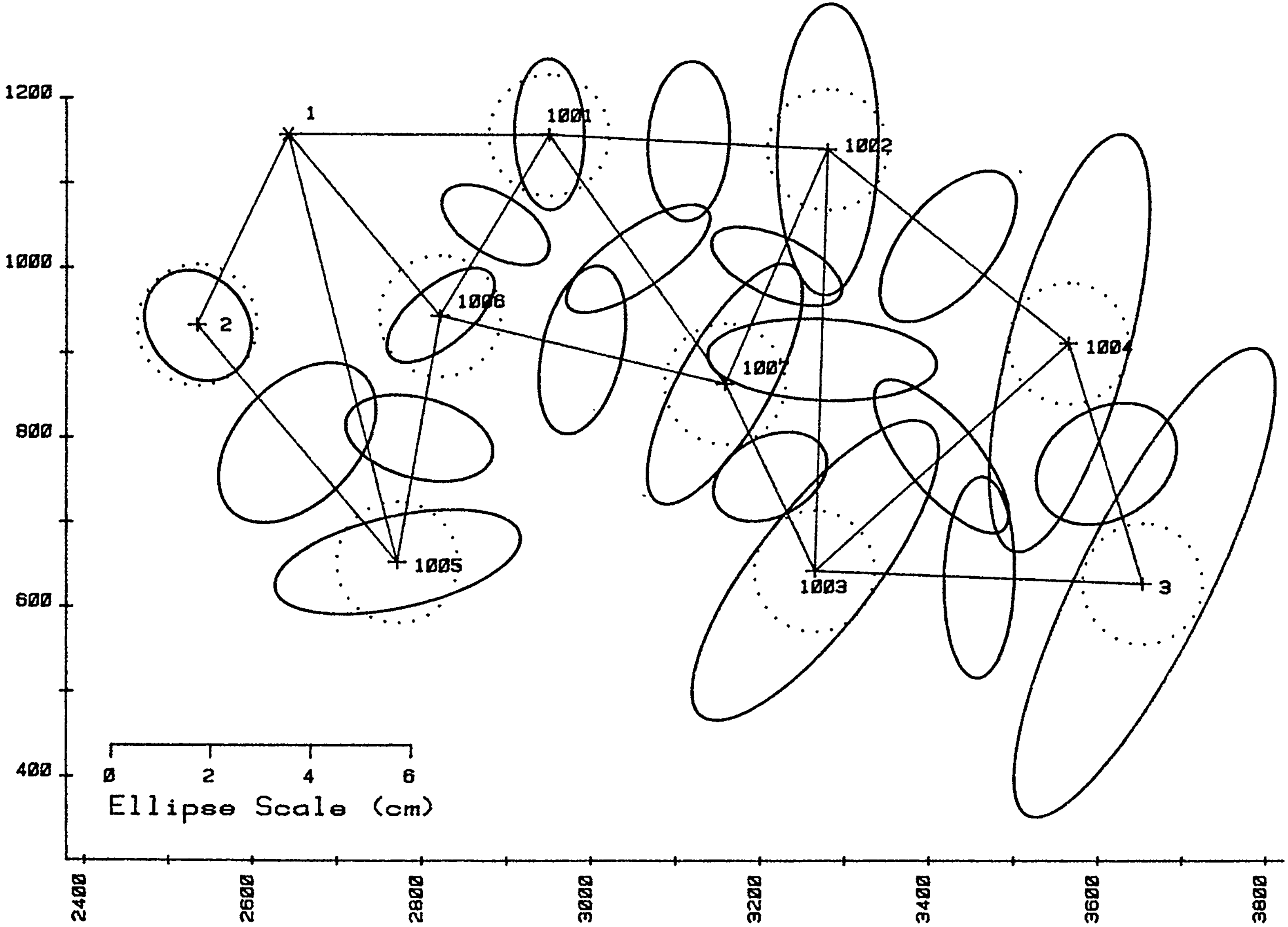


Figure 6.4 - Initial Network Survey

is the typical pattern of error propagation these days with current EDM equipment having the accuracies it does.

To illustrate the improvement that can result by "closing" on a known station, the network was updated by adding station 4 as a fixed station. The observations and accuracies used in the original solution were not changed. The new observations connecting station 4 to the network have the same accuracies as those in the rest of the network. The results are shown in Figure 6.5. The improvement is dramatic. The largest semi-major axis has been reduced to 1.1 cm from 5.8 cm and the ellipses are approaching a circular shape.

6.2.3 Deformation Survey

The last example is of a typical deformation, or monitoring, survey. The station to be monitored is station X in the center of the network. Stations A, B, C, and D form a local network used to monitor the station of interest and stations 1, 2, 3, and 4 form a regional network used to monitor the local network. The regional network is 3 kilometres across while the local network is only 200 metres across. Figure 6.6 illustrates the problem of overcrowding encountered when trying to display the entire network. The station of interest, X, is totally obscured by the ellipses plotted at the center of the network. A much clearer plot of the local network and station X is shown in Figure 6.7. This was achieved by windowing the plot so that only a portion of it was displayed on the screen. The figures for this example were produced on the Tektronix 4634 hard copy unit to illustrate the

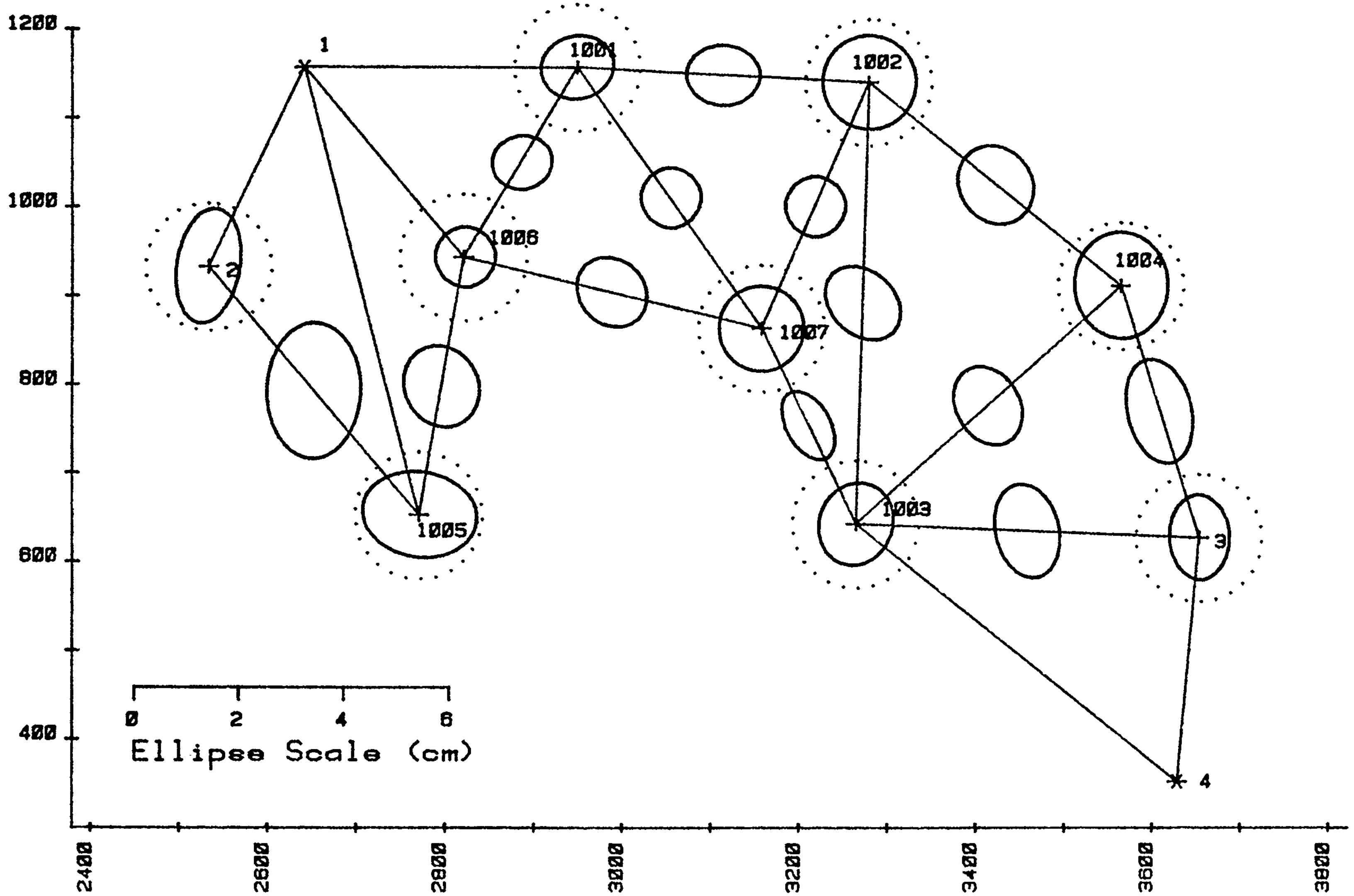
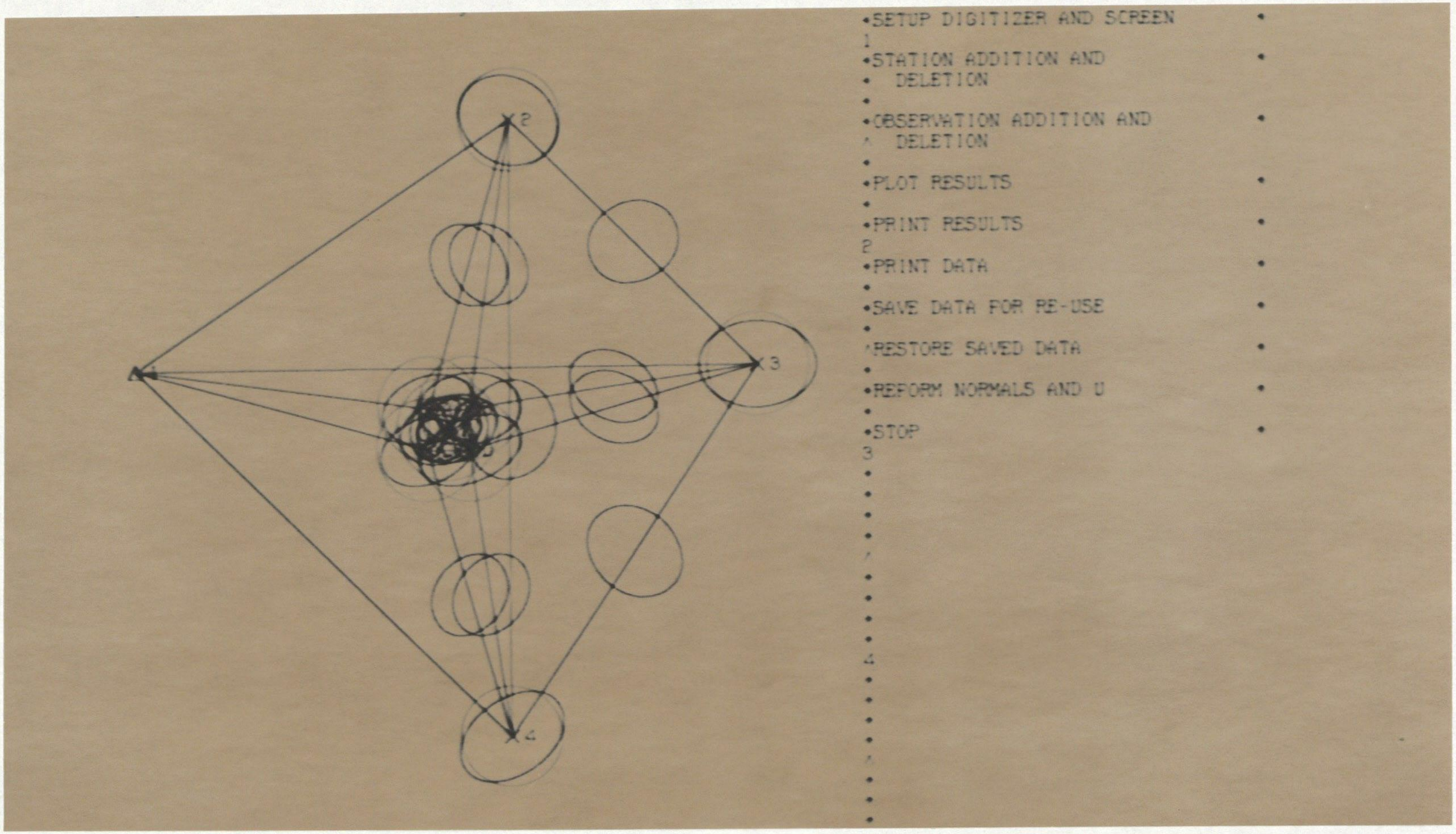


Figure 6.5 - Network Survey With Additional Fixed Station



```
•SETUP DIGITIZER AND SCREEN          •
1
•STATION ADDITION AND                •
•  DELETION
•
•OBSERVATION ADDITION AND            •
^  DELETION
•
•PLOT RESULTS                        •
•
•PRINT RESULTS                       •
2
•PRINT DATA                          •
•
•SAVE DATA FOR RE-USE                •
•
^RESTORE SAVED DATA                  •
•
•REFORM NORMALS AND U                •
•
•STOP                                •
3
•
•
•
•
^
•
•
•
•
4
•
•
•
•
^
•
•
•
```

Figure 6.6

Complete Deformation Network

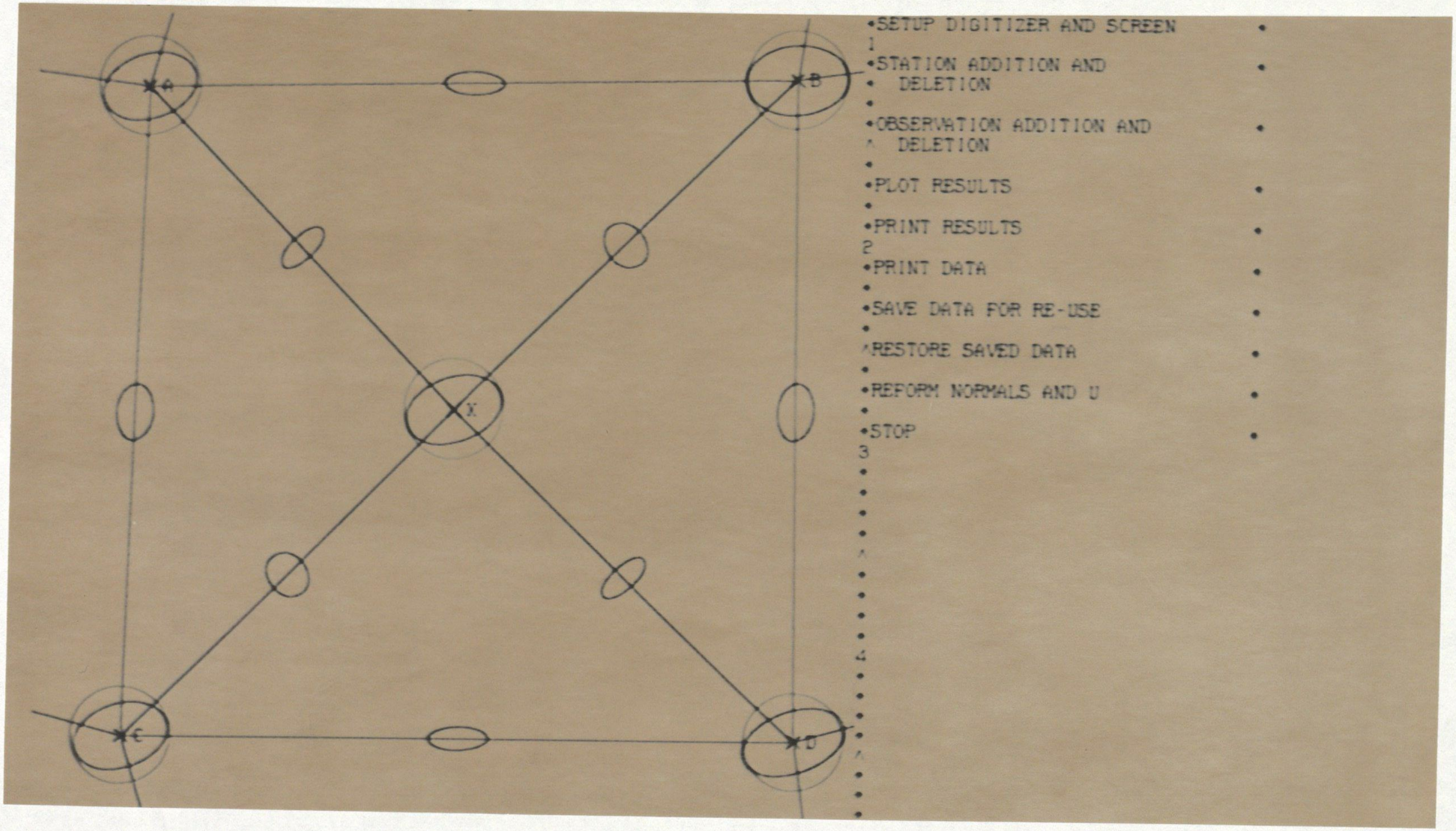


Figure 6.7

Window On Local Portion Of Deformation Network quality of the plots produced on this device compared to those produced on a flatbed plotter (Figures 6.2 to 6.5).

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

The aims for this research, as set out in the introduction to this thesis, have been met. CANDSN is a powerful and flexible tool that can be used to design and adjust horizontal networks quickly and easily. It has been installed on a small computer thus making possible its use by small survey firms without access to main-frame computer systems.

The use of computer graphics as an input and output technique is both quick and simple to learn and provides a wealth of information about the results that would be impossible to extract from a printed output. The example of the large network in Section 6.2.3 illustrates this. From a printed listing it would be impossible to see that the primary weakness in the net is a rotation about the single fixed point while from the plot of the networks and ellipses it is obvious.

The sequential expressions developed in Section 2.3 allow the rapid updating of the solution. Without these, the presentation of results would be much too slow and the delays would be a source of irritation for the user. With the space and time saving techniques presented in Chapter 3 the installation of CANDSN on a small computer would not have been possible.

The hardware configuration described in Chapter 5 has proven to be satisfactory. The graphics terminal is slow in its presentation and erasure of text information. This is particularly evident during changes of the menu page. This could be overcome by replacing the terminal with one employing a frame buffer system. The use of a colour graphics system would be a nice enhancement in that different colours could be used for the design circles and the confidence ellipses and the various observation types, making the interpretation of the results much easier.

The computer used has been stretched to its outer limits. The size of the core memory was the principal limiting factor on CANDSN. The installation of CANDSN on a larger, faster computer such as a VAX 750 would greatly enhance its capabilities. The size of network that could be processed and the speed of processing would be greatly increased. It would be possible, for example, to design and adjust networks of several hundred stations.

If CANDSN were installed on a larger computer, there are several extensions that should be incorporated. Among them are the following:

1. Extend the program to include one-dimensional and three-dimensional networks. This would make it a general purpose survey adjustment package.
2. Incorporate sensitivity and reliability testing into the program. This will make it a much more powerful tool for use in the design and analysis of engineering and deformation surveys.
3. Add terrain to mapping plane reduction of observations to the program. This will make it much easier to use when adjusting networks by further automating the processing of the data.

4. The statistical testing should be extended to include comparison of independent solutions and a more detailed analysis of the results. The testing procedures now used are those generally in use by the surveying community. The additional testing would be primarily aimed at research applications of CANDSN.
5. The use of criterion matrices to represent the required positional and relative accuracies should be investigated and included in CANDSN. This would be useful when designing to relative accuracy specifications as compared to point accuracy specifications.
6. The application of the sequential equations to free network adjustments should be investigated. This will be very useful in the analysis of deformation networks.
7. A coordinate geometry (COGO) package should be linked to CANDSN to facilitate further computations using the results of a network adjustment computed with CANDSN.
8. Further alternatives to typing of information when using CANDSN should be investigated and implemented. An example of this would be the use of a valuator algorithm for the input of numeric data using graphical methods. This would make CANDSN easier to use.

REFERENCES

Anderson, E.G., 1982; Towards Total Optimization of Surveying and Mapping Systems; Proceedings of Meeting of F.I.G. Study Group 5B (Survey Control Networks); Aalborg, Denmark; October, 1982.

Aoki, M., 1967; Optimization of Stochastic Systems; Academic Press; New York.

Baarda, W., 1973; S-Transformations and Criterion Matrices; Netherlands Geodetic Commission, Publications on Geodesy, New Series 5 (1); Delft, Netherlands.

Blank, L.T., 1980; Statistical Procedures for Engineering, Management, and Science; McGraw Hill, New York.

Burden, R.L., Faires, J.D., Reynolds, A.C., 1978; Numerical Analysis; Prindle, Weber and Schmidt, Boston.

Dongarra, J.J., Moler, C.B., Bunch, J.R., Stewart, G.W., 1979; LINPACK Users Guide; Society for Industrial and Applied Mathematics; Philadelphia.

Energy, Mines and Resources, 1978; Specifications and Recommendations for Control Surveys and Survey Markers; Surveys and Mapping Branch; Ottawa, Ontario.

Fadeev, D.K., and Fadeeva, V.N., 1963; Computational Methods of Linear Algebra; W.H. Freeman; San Francisco.

Grafarend, E.W., 1974; Optimization of Geodetic Networks; Proceedings of the International Symposium on Problems Related to the Redefinition of North American Geodetic Networks; The Canadian Surveyor, Vol. 28, No. 5, December 1974.

Grafarend, E.W., 1981; Optimization of Geodetic Networks; Proceedings of International Symposium on Geodetic Networks and Computations; Munich, West Germany; August 1981.

Hamilton, W.C., 1964; Statistics in Physical Science; The Ronald Press Company, New York.

Knight, W., 1978; A Program Package for Large Systems of Normal Equations of Least Squares Adjustment; Report to Surveys and Mapping Division, Land Registration and Information Service; Summerside, P.E.I.

Knight, W., Mepham, M.P., 1978; Report on Computer Programs for Solving Large Systems of Normal Equations; Proceedings of Second International Symposium on Problems Related to the Redefinition of North American Geodetic Networks; Arlington, Virginia; April 1978.

Koch, K.R., 1981; Otpimisation of the Configuration of Geodetic Networks; Proceedings of International Symposium on Geodetic Networks and Computations, Munich, West Germany; August, 1981.

Krakiwsky, E.J., 1981; A synthesis of Recent Advances in the Method of Least Squares; Publication 10003; Division of Surveying Engineering, University of Calgary, Calgary, Alberta.

Lawson, C.L., and Hanson, R.J., 1974; Solving Least Squares Problems; Prentice-Hall Series in Automatic Computations; Prentice-Hall, New Jersey.

Mikhail, 1976; Observations and Least Squares; IEP, New York.

Newman, W.M., and Sproull, R.W., 1979; Principles of Interactive Computer Graphics; McGraw-Hill, Toronto.

Nickerson, B.G., 1977; Horizontal Network Design Using Interactive Computer Graphics; M.Sc.E. Thesis; Department of Surveying Engineering; University of New Brunswick; Fredericton, New Brunswick.

Nickerson, B.G., 1981; LEAP-LRIS Ellipsoidal Adjustment Package System Overview; Surveys and Mapping Division, Land Registration and Information Service; Summerside, P.E.I.

Pope, A.J., 1974; Two Approaches to Non-Linear Least Squares Adjustments; Canadian Surveyor; Vol. 28, No. 5.

Pope, A.J., 1976; The Statistics of Residuals and the Detection of Outliers; NOAA Technical Report NOS 65 NGS1; U.S. Department of Commerce; Rockville, U.S.A.

Schaffrin, B., 1981; Some Considerations on the Optimal Design of Geodetic Networks; Proceedings of International Symposium on Geodetic Networks and Computations; Munich, West Germany; August, 1981.

Schmitt, G., 1981; Optimal Design of Geodetic Networks; Invited Review Paper; Proceedings of International Symposium on Geodetic Networks and Computations; Munich, West Germany; August 1981.

Steeves, P.A., 1974; Least Squares Adjustment of Horizontal Control on a Mapping Plane; M.Sc.E. Thesis; Department of Surveying Engineering, University of New Brunswick; Fredericton, New Brunswick.

Vanicek, P., and Krakiwsky, E.J., 1982; Geodesy: The Concepts; North-Holland Publishing, Amsterdam.

APPENDIX I

STARTING CANDSN

This appendix explains how to start CANDSN on the Division of Surveying Engineering PDP 11/23 computer system. It is not intended to be an instruction manual on the use of the PDP 11/23.

Before starting CANDSN, the Princeton terminal must be connected to the PDP. The cables are located behind the hard-copy unit. Plug the Princeton cable into one end of the short adaptor cable and the PDP cable into the other end. The GANDALF modem (blue box) is not used. The communication speed switches located behind the door of the Princeton controller should be set to 4800. The Princeton terminal should now be turned on using the key switch located in the top right of the Princeton keyboard.

Once the terminal has warmed up (30 to 60 seconds), the digitising tablet should be set up. The four button cursor should be plugged into the front of the controller. Set the controller to point digitising by pressing the "point" button in. It should lock in. If any other buttons arelocked in, release them. Now press the "clear" button momentarily.

You are now ready to start CANDSN. If the logout program is running, stop it by pressing and holding the "CTRL" key while typing "c" twice. (DO NOT PRESS "BREAK". If you do press the "BREAK" key, find someone who knows how to cold start the computer. The "BREAK key stops the operating system.) To start the program, enter the command

```
R CANDSN.
```

There are two possible results of this. One is that the program starts running and the second is that one of the following error messages will appear:

a) Not enough memory, or

b) CANDSN.SAV not found.

If the first message appears, it means that the wrong monitor is being used. To correct this, enter the command

```
BOOT RT11SJ
```

and answer yes to the question "Are you sure?". Once the monitor prompt (a period) appears, try the R CANDSN command again. The second error message indicates that the wrong disk is loaded. If this happens get someone to show you how to change disks.

---- GOOD LUCK ----

APPENDIX II

ROUTINES USED

IN CANDSN

This appendix is a list of the name of the routines used in CANDSN with a short description of what each routine does. The listing is broken down into 9 groups according to their use and source.

II.1 Application Routines

CPTVAR - Compute the variance of an observation.

FORMA - Form the A design matrix and w vector in compact form for a single observation.

OBS - Control the addition and deletion of observations.

OBSAD - Add an observation.

OBSDL - Delete an observation.

OBSDRW - Draw or erase an observation.

OBSDS - Display an observation record on the menu page.

OBSRC - Get the values for an observation record from the menu page.

PLTNET - Plot the network.

PRTDAT - Print the network data.

PRTRES - Print the results.

RSTDAT - Restore the network from a checkpoint file.

SAVDAT - Save the network data on a checkpoint file.

SETUP - Setup the terminal and digitising tablet.

SOLVE - Adjust the network.

STAT - Control the addition and deletion of stations.

STATAD - Add a station.

STATDL - Delete a station.

STATDS - Display a station record on the menu page.

STATRC - Get the values for a station record from the menu page.

STNPLT - Plot a station on the terminal.

II.2 Panel Matrix Routines

CHRPNL - Compute the Cholesky root of a matrix stored in a panel matrix file.

CLSPNL - Close a panel matrix file.

CPSPNL - Delete a set of rows and columns of a matrix stored in a panel matrix file.

CPYPNL - Copy the contents of one panel matrix file to another. Also used to add and subtract two panel matrices.

DLCPNL - Delete a set of columns of a matrix stored in a panel matrix file.

DLRPNL - Delete a set of rows of a matrix stored in a panel matrix file.

DMPPNL - Dump a panel matrix file to a sequential file for storage.

ERRPNL - Print error messages for the panel matrix routines.

EXTPNL - Extract a submatrix from a panel matrix.

GETPNL - Read one panel from the panel matrix file.

INTPNL - Initialise a panel matrix file to zero.

INVPNL - Compute the inverse of a panel matrix from the Cholesky root.

MPYPNL - Multiply two panel matrices.

OPNPNL - Open an existing panel matrix file.

PRTPNL - Print the contents of a panel matrix file.

RSTPNL - Restore a panel matrix from a sequential file created by DMPPNL.

STMPNL - Add a sparse matrix to the panel matrix.

WRTPNL - Write a panel to the panel matrix file.

ZROPNL - Set a panel matrix to zero.

II.3 Station File Routines

DLTSTF - Delete a record from the station file.

DMPSTF - Dump the station file to a sequential file for saving.

GETSTF - Read a record from the station file.

INTSTF - Initialise the station file.

MTRSTF - Return the record number of the next free record in the station file.

RSTSTF - Restore the station file from the save file created by DMPSTF.

UPDSTF - Update the coordinates of the stations in the station file.

WRTSTF - Write a record to the station file.

II.4 Observation File Routines

BLKOBF - Return the sequence numbers of all observations of a specified type involving a specified station.

DLTOBF - Delete a record from the observation file.

DMPOBF - Dump the observation file to a sequential file for saving.

GETOBF - Get a record from the observation file.

INTOBF - Initialise the observation file.

MTROBF - Return the record number of the next available observation file record.

PRTOBF - Print the contents of the observation file.

RSTOBF - Restore the observation file from the save file created by DMPOBF.

WRTOBF - Write a record to the observation file.

II.5 Menu Processing Routines

DGIDZN - Transform digitiser coordinates to screen coordinates.

DGMNLD - Load menu on digitiser tablet.

FRCHNG - Control menu page changes on the graphics screen.

FRCNG - Change menu pages on the screen.

GETRSP - Control all input from the graphics terminal.

GTRSVL - Return the current value of a menu option.

IDZNSL - Identify what screen zone the cursor was in and what menu item was selected.

MNULD - Load the menu file from disk.

MSGDSP - Display a message on the screen.

SETRSP - Assign a value to one of the menu options.

TXTERS - Erase selected portions of the screen.

II.6 General Purpose Utility Routines

CLRSTR - Set a character string to all blanks.

DEGRAD - Convert degrees, minutes and seconds to radians.

IBLOC - Find the position of an integer in a sorted list of virtual integers.

INSIDE - Logical function to determine whether or not a point is located within the drawing area.

LJSTFY - Left justify a character string.

LVECPY - Copy a vector of any type except virtual.

QUERY - Ask a yes/no question on the VT100.

RADEG - Convert radians to degrees, minutes and seconds.

RJSTFY - Right justify a character string.

UPPER - Convert a character string to all upper case.

II.7 Matrix Manipulation Routines

DDOT - Dot product of two double precision vectors.

DLTFIL - Copy upper triangle of a double precision matrix to the lower triangle.

DMXMUL - Double precision matrix multiplication.

DPODI - Determinant and/or inverse of a double precision matrix from the Cholesky root.

DPOFA - Cholesky root of a double precision matrix.

DUDDI - Determinant and/or inverse of a double precision matrix from the U-D factor.

DUDFA - U-D factorisation of a double precision matrix.

DUTIN - Inverse of an upper triangular double precision matrix.

VDDOT - Dot product of two virtual double precision vectors.

VDLTFL - Copy upper triangle of a virtual double precision matrix to the lower triangle.

VDMXML - Virtual double precision matrix multiplication.

VDPODI - Determinant and/or inverse of a virtual double precision matrix from the Cholesky root.

VDPOFA - Cholesky root of a virtual double precision matrix.

VDUDDI - Determinant and/or inverse of a virtual double precision matrix from the U-D factor.

VDUDFA - U-D factorisation of a virtual double precision matrix.

VDUTIN - Inverse of an upper triangular virtual double precision matrix.

II.8 Digital Library Routines

CONCAT - Concatenate two character strings.

DATE - Return current date as a character string.

GETSTR - Read a character string from a file.

GTLIN - Print a question and return the answer as a character string.

GTIM - Return current time as an integer value.

INDEX - Return the location of a substring in a character string.

ISCOMP - Compare two character strings.

LEN - Return the length of a character string.

PUTSTR - Write a character string to a file.

SCOPY - Copy one character string to another.

STRPAD - Pad a character string with blanks.

TIME - Return the current time as a character string.

TRIM - Remove trailing blanks from a character string.

II.9 Princeton Library Routines

PABCRD - Return the current cursor position.

PABSUC - Draw a line or move to a position specified in screen coordinates.

PCNTL - Issue control commands to the terminal.

PGRDAT - Convert response string to screen coordinates.

PGRREQ - Put the terminal into graphics mode.

PGS - Set grey scale (line intensity).

PREAD - Read a response string from the terminal.

PRELVC - Relative draw or move to a position specified in screen coordinates.

PTABLT - Enable or disable the digitising tablet.

PTBDAT - Decode the digitising tablet response string.

PTEXT - Print a text string on the graphics terminal.

PTXMVA - Move to an absolute position specified in terms of the text row and column numbers.

PXMITB - Buffer transmissions to the terminal.

PXMITC - Do not buffer transmissions to the terminal.

VABSVC - Draw a line or move to a position specified in user coordinates.

VCRCL - Draw a circle.

VDATCV - Convert screen coordinates to user coordinates.

VELLPS - Draw an ellipse.

VSWIND - Define a window on the screen.

VVWIND - Define a viewport.

In addition there are 56 routines used by these routines. They are not called directly from CANDSN and so are not listed here.