NPL Report CISE 14/98

February 1998

Position Calibration Software

Fortran implementation of algorithms to determine frame of reference from measurements of registration points

B P Butler, A B Forbes and P D Kenward
Centre for Information Systems Engineering
National Physical Laboratory
Teddington
Middlesex
United Kingdom
TW11 0LW

ABSTRACT

This report documents a Fortran implementation of software to determine transformations linking different frames of reference from the measurement of registration points. This problem arises in determining the position and orientation of an artefact by tracking the position of three or more points (registration points) which move rigidly with the artefact. The software finds least squares estimates of the transformation parameters and locations of the registration points in a common frame of reference. These estimates are calculated efficiently by a nonlinear least squares solver which takes full account of the structure of the matrix equations defining the solution. An initial estimation module determines good starting values for the optimisation parameters. There are also modules for calculating standard uncertainties of the fitted parameters and related quantities. The software can be applied directly to the measurement of large or complex artefacts using staging or repositioning methods in coordinate metrology.

NPL Report CISE 14/98

ISSN 1361-407X

National Physical Laboratory
Teddington, Middlesex, United Kingdom, TW11 0LW

Approved on behalf of Managing Director, NPL
by Dr D Rayner, Head, Centre for Information Systems Engineering

CONTENTS

1 INTRODUCTION

This report describes software for determining accurately the location and orientation of an object from measurements of the locations of points (*registration points)* that move rigidly with the object. The basic fact that is required for position calibration (or *registration*) is that the locations of three (non-collinear) points are necessary and sufficient to determine a frame of reference in three dimensions. An important application of this software is in the use of repositioning methods in coordinate metrology to measure large or complex artefacts for which only part of the artefact's surface can be accessed for any given position of the artefact. In this application an artefact is moved from position to position within the working volume of a coordinate measuring machine while measurements of the coordinates of points on the artefact's surface are gathered. These measurements can be transformed into a common frame of reference if the position and orientation of the artefact are recorded through the measurements of the centres of spheres that move rigidly with the artefact. This approach allows for an assessment of the artefact as though the measurements were taken with the artefact in one position (Forbes, 1992, Cox *et al.*, 1997).

This report is structured as follows. In section 2 we present the mathematical model for the position calibration problem while section 3 describes algorithms to determine initial and solution estimates of the model parameters along with algorithms for checking the consistency of the data and for statistical calculations. Section 4 provides comprehensive documentation of the main Fortran 77 subroutines implementing the algorithms. A sample main program, data and results are discussed in section 5. Following the bibliography in section 6, four Annexes contain a listing of the sample programme, sample data and results output by the software along with a description of the modular structure of the software.

2 MATHEMATICAL MODEL

Let:

- $\{y_j\}_1^{n_Y}$ represent the coordinates of the registration points in a fixed (*0*th) frame of reference. We suppose that these points are fixed relative to each other.

  $\{y_{j,k}\}_1^{n_Y}$ denote the coordinates of the registration points in the *k*th position, $k = 0, 1, ..., n_T$.

  $\{x_i\}_1^{m_X}$ denote a set of measurements of (subsets of) the registration points in the various positions. We assume that there are two index functions associated with these measurements: $j(i)$ points to the corresponding registration point, $k(i)$ points to the corresponding position.

  $I_k = \{i: k(i) = k\}$, i.e., the index set of registration point measurements gathered in the *k*th position.

  $I^j = \{i: j(i) = j\}$, i.e., the index set of measurements of the *j*th registration point.

  $S_k = \{j(i): i \in I_k\}$, i.e., the index set of registration points that are measured at least once in the *k*th position.

$\{\epsilon_i\}_1^{m_x}$ denote the measurement errors associated with $\{x_i\}_1^{m_x}$

$\{D_i\}_i^{m_x}$ denote 3×3 diagonal weighting matrices chosen to reflect the relative measurement accuracies associated with $\{x_i\}_1^{m_x}$.

The basic model equations arise from the fact that the locations of the registration points in the kth frame of reference are given by

$$y_{j,k} = T_k(y_j), \tag{1}$$

where T_k is a transformation composed of a translation and a rotation:

$$T_k(x) = R_k(x - a_k).$$

We parametrize a rotation matrix R as a product

$$R = R_z R_y R_x R_0$$

of three plane rotations

$$R_z = \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 \\ -\sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_y \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix}, \quad R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix},$$

and a fixed rotation matrix R_0. Thus, a general transformation $T = T(x, a) = R(x - a)$ is specified by a fixed rotation R_0 and the parameters $t = (a^T, \theta^T)^T$.

Adopting the notation set out above the model equations are:

$$x_i = y_{j(i),k(i)} + \epsilon_i \tag{2}$$

If we assume that the components of the errors ϵ_i are uncorrelated and normally distributed then an efficient, unbiased estimate of the parameters can be determined by minimising the sum-of-squares objective function

$$\sum_{i=1}^{m_x} \epsilon_i^T D_i^2 \epsilon_i = \sum_{i=1}^{m_x} (x_i - y_{j(i),k(i)})^T D_i^2 (x_i - y_{j(i),k(i)}) \tag{3}$$

with respect to the parameters $\{y_j\}$ and $\{t_k\}$. In order to fix an overall frame of reference the initial (0th) transformation is set equal to the identity transformation. In this way, it is possible to formulate the problem of determining the relative frames of reference as an unconstrained nonlinear least squares problem which can be solved using the optimisation techniques described below.

3 ALGORITHM DESCRIPTION

3. NONLINEAR LEAST SQUARES ALGORITHM

The most common approach to solving nonlinear least squares problems is derived from the *Gauss-Newton algorithm*, which we now briefly describe; see, for example, Gill, Murray and Wright (1981). Suppose we wish to minimise

$$F(\boldsymbol{a}) = \sum_{i=1}^{m} f_i^2(\boldsymbol{a})$$

with respect to the parameters $\boldsymbol{a} = (a_1, ..., a_n)^T$. If J is the associated *Jacobian matrix* defined at an estimate $\boldsymbol{a}$ of the solution by

$$J_{ij} = \frac{\partial f_i}{\partial a_j},$$

then an updated estimate of the solution is given by $\boldsymbol{a} + \boldsymbol{p}$, where $\boldsymbol{p}$ (known as the *Gauss-Newton step*) solves the Jacobian system $J\boldsymbol{p} = -\boldsymbol{f}$ in the least squares sense. These steps are repeated until convergence criteria involving the norms of $\boldsymbol{p}$ and gradient $\boldsymbol{g} = J^T\boldsymbol{f} = \nabla_{\boldsymbol{a}} F/2$ are met.

A numerically stable method of solving the Jacobian system is to find a factorisation $J = QU$, where Q is an $m \times n$ orthogonal matrix and U an $n \times n$ upper triangular matrix. The solution $\boldsymbol{p}$ can be found efficiently by solving the upper-triangular system

$$U\boldsymbol{p} = Q^T\boldsymbol{f}.$$

The orthogonal factorisation stage requires $O(mn^2)$ operations and for large matrices it may be beneficial to exploit any structure in the Jacobian matrix to provide better computational performance, both in terms of speed and memory.

In practice, the update is usually of the form $\boldsymbol{a} = \boldsymbol{a} + \beta\boldsymbol{p}$ where β is chosen to ensure that sufficient progress towards the solution is made. For example, a simple parabolic search strategy may be employed. Initially, β is set equal to 1 (the full Gauss-Newton step). If $\boldsymbol{a} + \beta\boldsymbol{p}$ fails to provide a satisfactory decrease in F, a new β is determined by the minimiser of the quadratic determined by the information $F(\boldsymbol{a})$, $F(\boldsymbol{a}+\beta\boldsymbol{p})$ and $\partial F(\boldsymbol{a})/\partial\boldsymbol{p}$. The Gauss-Newton iterative scheme is judged to have converged if criteria involving the change in the objective function value, the norm of the Gauss-Newton step $\boldsymbol{p}$ and the norm of gradient $\boldsymbol{g}$ are met.

Estimates of the variances and covariances of linear combinations of the fitted parameters $\boldsymbol{a}$ can be calculated directly from the Jacobian system. Suppose J is the Jacobian at the solution $\boldsymbol{a}$ and that $h_1 = \boldsymbol{h}_1^T\boldsymbol{a}$ and $h_2 = \boldsymbol{h}_2^T\boldsymbol{a}$ are linear combinations of the fitted parameters. The *covariance* of h_1 with h_2, $\mathrm{cov}(h_1, h_2)$, is defined by

$$\mathrm{cov}(h_1, h_2) = \hat{\sigma}^2\boldsymbol{h}_1^T\left(J^TJ\right)^{-1}\boldsymbol{h}_2,$$

where $\hat{\sigma} = \|\boldsymbol{f}\|/\sqrt{m-n}$ is an estimate of the standard deviation σ of the residual errors. The *variance* of h is given by $\mathrm{var}(h) = \mathrm{cov}(h,h)$.

If the upper-triangular factor U of J is available then

$$\boldsymbol{h}_1^{\mathrm{T}}\left(J^{\mathrm{T}}J\right)^{-1}\boldsymbol{h}_2 = \boldsymbol{h}_1^{\mathrm{T}}U^{-1}U^{-\mathrm{T}}\boldsymbol{h}_2$$

Hence, $\mathrm{cov}(h_1, h_2) = \sigma^2 \tilde{\boldsymbol{h}}_1^{\mathrm{T}} \tilde{\boldsymbol{h}}_2$, where $\tilde{\boldsymbol{h}}_k$ solves $U^{\mathrm{T}}\tilde{\boldsymbol{h}}_k = \boldsymbol{h}_k$, $k = 1, 2$, respectively. These lower triangular systems can be solved efficiently using forward substitution.

The same approach can be used to determine estimates of the covariance of nonlinear functions $h_k(\boldsymbol{a})$, $k = 1, 2$, of the fitted parameters if we set $\boldsymbol{h}_k = \nabla_{\boldsymbol{a}} h_k$.

3.2 DERIVATIVE CALCULATIONS FOR POSITION CALIBRATION

The non-zero elements of the Jacobian matrix associated with (3) are easily calculated using the following formulae. Suppose $\hat{\boldsymbol{x}} = T(\boldsymbol{x}, \boldsymbol{t})$ where $\boldsymbol{t} = (\boldsymbol{a}^{\mathrm{T}}, \theta^{\mathrm{T}})^{\mathrm{T}}$ as before. Then

$$\frac{\partial \hat{\boldsymbol{x}}}{\partial \boldsymbol{x}^{\mathrm{T}}} = R, \qquad \frac{\partial \hat{\boldsymbol{x}}}{\partial \theta_x} = R_z R_y \dot{R}_x R_0 (\boldsymbol{x} - \boldsymbol{a}),$$

$$\frac{\partial \hat{\boldsymbol{x}}}{\partial \boldsymbol{a}^{\mathrm{T}}} = -R, \qquad \frac{\partial \hat{\boldsymbol{x}}}{\partial \theta_y} = R_z \dot{R}_y R_x R_0 (\boldsymbol{x} - \boldsymbol{a}),$$

$$\frac{\partial \hat{\boldsymbol{x}}}{\partial \theta_z} = \dot{R}_z R_y R_x R_0 (\boldsymbol{x} - \boldsymbol{a}).$$

Here,

$$\dot{R} = \frac{\partial R}{\partial \theta_z} = \begin{bmatrix} -\sin\theta_z & \cos\theta_z & 0 \\ -\cos\theta_z & -\sin\theta_z & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

etc.

3.3 EXPLOITING JACOBIAN STRUCTURE

Not all the observation equations associated with the position calibration problem involve all the parameters and this will give rise to structure in the Jacobian matrix, as the following example makes clear. Suppose there are four registration points, $\{y_j\}_1^4$, all of which are measured in the initial position, $\{y_j\}_1^3$ are measured in a first transformed position and $\{y_j\}_2^4$ are measured in a second transformed position. The rows and columns of the Jacobian can be ordered so as to look like:

$$\begin{bmatrix} -I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -R_1 & 0 & 0 & 0 & R_1 & B_{1,1} & 0 & 0 \\ 0 & -I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -R_1 & 0 & 0 & R_1 & B_{2,1} & 0 & 0 \\ 0 & -R_2 & 0 & 0 & 0 & 0 & R_2 & B_{2,2} \\ 0 & 0 & -I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -R_1 & 0 & R_1 & B_{3,1} & 0 & 0 \\ 0 & 0 & -R_2 & 0 & 0 & 0 & R_2 & B_{3,2} \\ 0 & 0 & 0 & -I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -R_2 & 0 & 0 & R_2 & B_{4,2} \end{bmatrix} \tag{4}$$

Here the columns are ordered with the y parameters appearing first and the transformation parameters second. The rows reflect a partitioning of the data $\{x_i\}$ according to the measurements of the registration points: the first block of rows arises from measurements of the registration point 1, and so on. R_k is the rotation matrix associated with the kth transformation (and the identity matrix I is associated with the initial (*zero*th) frame of reference). The $B_{j,k}$ represent derivatives with respect to the rotation angles.

In general, the Jacobian matrix can be regarded as block angular (Cox, 1990) and this structure is reflected in that of its upper-triangular factor U which will have the form:

$$U = \begin{bmatrix} U_1 & & & & & B_1 \\ & U_2 & & & & B_2 \\ & & \ddots & & & \vdots \\ & & & U_j & & B_j \\ & & & & \ddots & \vdots \\ & & & & U_{n_Y} & B_{n_Y} \\ & & & & & U_T \end{bmatrix}, \tag{5}$$

where the U_j are 3×3 upper triangular, the border blocks B_j are $3 \times 6n_T$, and U_T is $6n_T \times 6n_T$ upper triangular (n_T is the number of transformations).

The block angular structure can be exploited during the orthogonal factorisation process as described below. Each measurement x_i gives rise to three rows in the Jacobian matrix, the non-zero elements of which are limited to the three columns $J_{i,j}$ corresponding to the parameters $y_{j(i)}$ and the columns $J_{i,T}$ corresponding to the transformation parameters $\{t_k\}$. These rows can be incorporated by using Givens rotations (Golub and Van Loan, 1996) to determine an orthogonal matrix Q_i such that

$$\begin{bmatrix} U_j & B_j \\ 0 & U_T \end{bmatrix} := Q_i^{\mathrm{T}} \begin{bmatrix} U_j & B_j \\ 0 & U_T \\ J_{i,j} & J_{i,T} \end{bmatrix}$$

This approach can be further refined by processing the rows associated with the zeroth transformation first, and then those associated with transformations $k = n_T, \ldots, 1$ (that is, in reverse order). The advantage of this strategy is that processing the measurements corresponding to transformation $k > 0$ only involves the border columns associated with the transformation parameters t_r, $r \geq k$ (and the measurements in the zeroth position do not involve the border at all). On average this speeds up the computation by a factor of three.

The block angular structure can be exploited also in the calculation of the covariance matrices for the registration points and related calculations as is shown below.

3.4 TRANSFORMING DATA POINTS

Once the transformation parameters t_k have been determined, it is possible to use the inverse transformations to transform points $W_k = \{w_l,\ l \in L_k\}$ in the different frames of reference into the common frame of reference. The inverse to the transformation T defined in section 2 is

$$\check{w} = T^{-1}(w, t) = a + (R_z R_y R_x R_0)^{\mathrm{T}} w .$$

We note that

$$\frac{\partial \check{w}}{\partial w^{\mathrm{T}}} = R^{\mathrm{T}}, \qquad \frac{\partial \check{w}}{\partial \theta_x} = (R_z R_y \dot{R}_x R_0)^{\mathrm{T}} w,$$

$$\frac{\partial \check{w}}{\partial a^{\mathrm{T}}} = I, \qquad \frac{\partial \check{w}}{\partial \theta_y} = (R_z \dot{R}_y R_x R_0)^{\mathrm{T}} w,$$

$$\frac{\partial \check{w}}{\partial \theta_z} = (\dot{R}_z R_y R_x R_0)^{\mathrm{T}} w$$

These derivatives can be used to calculate the covariance matrices relating to transformed points, given the covariance matrix of the transformation parameters as described below.

3.5 COVARIANCE CALCULATIONS

In many applications it is important to derive estimates of the uncertainty of fitted parameters and related quantities. We assume that the upper-triangular factor of the Jacobian matrix at the solution is as in (5) and that $\hat{\sigma}$ is an estimate of the standard deviation of the weighted residual errors.

3.5.1 *Covariance matrix for the transformation parameters* t_k

If H_k is the $6n_T \times 6$ matrix with the 6×6 identity matrix stored in rows $6k-5$ to $6k$ and zeros everywhere else, and $\tilde{H}_k$ solves $U_T^{\mathrm{T}} \tilde{H}_k = H_k$, then the covariance matrix V_{t_k} of t_k is given by

$$V_{t_k} = \hat{\sigma}^2 \tilde{H}_k^{\mathrm{T}} \tilde{H}_k.$$

3.5.2 *Covariance matrix for the registration point parameters* y_j

Let I_3 be the 3×3 identity matrix and let $\tilde{H}_j$ solve

$$\begin{bmatrix} U_j & B_j \\ 0 & U_T \end{bmatrix}^{\mathrm{T}} \tilde{H}_j = \begin{bmatrix} I_3 \\ 0 \end{bmatrix}$$

Then the covariance matrix V_{y_j} of y_j is given by $V_{y_j} = \hat{\sigma}^2 \tilde{H}_j^{\mathrm{T}} \tilde{H}_j$.

3.5.3 *Standard uncertainty for the distance between two registration points*

If $d_{jl} = d(y_j, y_l) = \|y_j - y_l\|$, then

$$\frac{\partial d_{jl}}{\partial y_j} = n_{jl},$$
$$\frac{\partial d_{jl}}{\partial y_l} = -n_{jl},$$

where $n_{jl} = (y_j - y_l)/d_{jl}$. Let $\tilde{h}_{jl}$ solve

$$\begin{bmatrix} U_j & & B_j \\ & U_l & B_l \\ & & U_T \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} n_{jl} \\ -n_{jl} \\ 0 \end{bmatrix}.$$

Then the variance of d_{jl} is estimated by $\mathrm{var}(d_{jl}) = \hat{\sigma}^2 \tilde{h}_{jl}^{T} \tilde{h}_{jl}$ and the standard uncertainty is given by $u(d_{jl}) = \hat{\sigma}\|\tilde{h}_{jl}\|$.

3.5.4 *Standard uncertainty for the distance between a registration point and a transformed point*

Suppose y_j is a registration point, $\hat{w}_{p,k} = T_k(w_p, t_k)$ is a transformed point, and that the standard uncertainty of

$$d_{jp,k} = \|y_j - \hat{w}_{p,k}\|$$

is required. In this situation the covariance of $\boldsymbol{y}_j$ with $\boldsymbol{t}_k$ has to be taken into account. Let

- $\boldsymbol{n}_{jp,k} = (\boldsymbol{y}_j - \hat{\boldsymbol{w}}_{p,k})/d_{jp,k}$,

  $H_{p,k}$ be the $6n_T \times 3$ matrix with $-\partial\hat{\boldsymbol{w}}_{p,k}^{\mathrm{T}}/\partial\boldsymbol{t}_k$ in rows $6k-5$ to $6k$ and zeros everywhere else,

  $\boldsymbol{h}_{jp,k} = H_{p,k}\boldsymbol{n}_{jp,k}$, and

  $\tilde{\boldsymbol{h}}_{jp,k}$ the solution of

$$\begin{bmatrix} U_j & B_j \\ \mathbf{0} & U_T \end{bmatrix} \tilde{\boldsymbol{h}}_{jp,k} \quad \begin{bmatrix} \boldsymbol{n}_{jp,k} \\ \boldsymbol{h}_{jp,k} \end{bmatrix}$$

Then $var(d_{jp,k})$ is given by

$$\mathrm{var}(d_{jp,k}) = \hat{\sigma}^2 \tilde{\boldsymbol{h}}_{jp,k}^{\mathrm{T}} \tilde{\boldsymbol{h}}_{jp,k},$$

and its standard uncertainty is given by $u(d_{jp,k}) = \hat{\sigma}\|\tilde{\boldsymbol{h}}_{jp,k}\|$

3.5.5 Standard uncertainty for the distance between two points transformed by different transformations

Suppose $\hat{\boldsymbol{w}}_{p,k} = T_k(\boldsymbol{w}_p, \boldsymbol{t}_k)$, $\hat{\boldsymbol{w}}_{q,r} = T_r(\boldsymbol{w}_q, \boldsymbol{t}_r)$, and that the standard uncertainty of

$$\hat{d}_{pq,kr} = \|\hat{\boldsymbol{w}}_{p,k} - \hat{\boldsymbol{w}}_{q,r}\|$$

is required. In this situation the covariance of $\boldsymbol{t}_k$ with $\boldsymbol{t}_r$ has to be taken into account. Let

  $\hat{\boldsymbol{n}}_{pq,kr} = (\hat{\boldsymbol{w}}_{p,k} - \hat{\boldsymbol{w}}_{q,r})/\hat{d}_{pq,kr}$,

  $H_{pq,kr}$ be the $6n_T \times 3$ matrix with $\partial\hat{\boldsymbol{w}}_{p,k}^{\mathrm{T}}/\partial\boldsymbol{t}_k$ in rows $6k-5$ to $6k$, $-\partial\hat{\boldsymbol{w}}_{q,r}^{\mathrm{T}}/\partial\boldsymbol{t}_r$ in rows $6r-5$ to $6r$, and zeros everywhere else,

  $\boldsymbol{h}_{pq,kr} = H_{pq,kr}\hat{\boldsymbol{n}}_{pq,kr}$, and

- $\tilde{\boldsymbol{h}}_{pq,kr}$ the solution of $U_T^{\mathrm{T}} \tilde{\boldsymbol{h}}_{pq,kr} = \boldsymbol{h}_{pq,kr}$.

Then $var(\hat{d}_{pq,kr})$ is given by

$$\mathrm{var}(\hat{d}_{pq,kr}) = \hat{\sigma}^2 \tilde{\boldsymbol{h}}_{pq,kr}^{\mathrm{T}} \tilde{\boldsymbol{h}}_{pq,kr},$$

and its standard uncertainty is given by $u(d_{pq,kr}) = \hat{\sigma}\|\tilde{\boldsymbol{h}}_{pq,kr}\|$

3.6 INITIAL ESTIMATES OF THE TRANSFORMATION PARAMETERS

3.6.1 Matching two sets of points

Suppose $X = \{x_i\}_1^m$ and $W = \{w_i\}_1^m$ are two sets of points in $\mathbb{R}^n$, one a nominal transformation of the other. The problem of finding the best match of one set of points to another in the least squares sense can be solved in a two-stage process. Firstly, it can be shown that the centroid $\bar{x}$ of $\{x_i\}$ must be transformed onto the centroid $\bar{w}$ of $\{w_i\}$. Let $\tilde{X}$ be the matrix whose ith row is $(x_i - \bar{x})^{\mathrm{T}}$ and define $\tilde{W}$ similarly. The optimal R_0 must therefore minimise

$$\|\tilde{W} - \tilde{X}R_0^{\mathrm{T}}\|_F,$$

where $\|A\|_F$ is the Frobenius norm of A (the square root of the sum of the squares of the elements of A). If

$$USV^{\mathrm{T}} = \hat{X}^{\mathrm{T}}\hat{W}$$

is the singular value decomposition (SVD) of the $n \times n$ matrix $\tilde{X}^{\mathrm{T}}\tilde{W}$, then it can be shown that $R_0 = VU^{\mathrm{T}}$ is the optimal *orthogonal* matrix. This leads to the following scheme (see, for example, Golub and van Loan (1996), Chapter 12) for determining the transformation T.

Calculate the matrices $\tilde{X}$ and $\tilde{W}$ and form the $n \times n$ matrix $A = \tilde{X}^{\mathrm{T}}\tilde{W}$

II Calculate the SVD of A: $USV^{\mathrm{T}} = A$.

III Set $R_0 = VU^{\mathrm{T}}$ and $a = \bar{x} - R_0^{\mathrm{T}}\bar{w}$

One important modification is required if the data points lie close to a hyperplane (and in particular if there are only n points in each data set). In this situation it is possible that the R_0 calculated by this scheme will be the composition of a rotation and a reflection. This can be addressed by adding one further step:

IV If $\det R_0 < 0$, set $R_0 = V\check{U}^{\mathrm{T}}$ where

$$\check{U} = U\begin{bmatrix} I_{n-1} & 0 \\ 0 & -1 \end{bmatrix}.$$

This last step has the effect of reflecting $\tilde{X}$ through the best-fit hyperplane to $\tilde{X}$ and then solving the problem for this modified data set.

With a and R_0 so defined the transformation $T(x, a) = R_0(x - a)$ maps the points X as close as possible to the points W in the least squares sense.

3.6.2 Estimates for identifiable designs

The above method can be used to relate one frame of reference to another assuming that there are at least three (non-collinear) points common to both. To determine all the transformation parameters in this way it is sufficient to find an ordering i_k of the index sets I_k such that I_{i_k} has at least three (non-collinear) points in common with the union of I_{j_k}, $j_k < i_k$. We will call an experimental design for which this property holds *identifiable.*

Example. Suppose the index sets are: $I_0 = \{1,2,3,4\}$, $I_1 = \{2,3,4,5\}$ and $I_2 = \{1,4,5,6\}$. Since I_0 and I_1 have three points in common it should be possible to relate the two frames of reference and give an estimate of the location of y_5 in the first frame of reference. Since I_2 and $I_0 \cup I_1$ have three points in common it is possible to relate the third frame of reference to the first and determine an estimate of y_6.

On the other hand, if the index sets are: $I_0 = \{1,2,3,4\}$, $I_1 = \{3,4,5,6\}$ and $I_2 = \{1,2,5,6\}$, there is no pair of sets with three points in common and the method described above cannot be implemented. This does not mean that the transformation parameters are not well-determined by the data, only that an initial estimation procedure based on matching two sets of points cannot be used.

To check if the identifiability property given above holds is purely a combinatorial problem and can be solved in the following manner. The algorithm below also calculates the transformation parameters that can be identified from stage to stage. Let $\{x_i\}_1^m$ be the measurements of the registration points, with indexing functions $j(i)$ and $k(i)$, and $S_k = \{j(i)\colon k(i) = k\}$ specify the registration points measured in the kth position. For each k:

Set $P_0 = S_0$, $Q_0 = \{0\}$, and $Y_0 = \{y_j, j \in P_0\}$

II Suppose P_q, Q_q and Y_q have been assigned. If $P_q = S$ and $Q_q = \{k\}_1^{n_T}$ then the system is identifiable starting from *0*.

Otherwise, set $P_{q,0} = P_q$, $Q_{q,0} = Q_q$ and

for $k = 1,\ldots,n_T$,

Set $M_k = S_k \cap P_{q,k-1}$ which indexes the registration points measured in the kth position for which estimates of their location in the 0th position are already known.

if M_k has more than three elements:

Set:

$$P_{q,k} = P_{q,k-1} \cup S_k, \quad Q_{q,k} = \{k\} \cup Q_{q,k-1},$$

$$\bar{X}_k = \{\bar{x}_{j,k}\colon \bar{x}_{j,k} = \text{mean}\{x_i\colon j(i) = j,\ k(i) = k,\ j \in M_k\}\}$$

$$Y_k = \{y_k\colon j \in M_k\}.$$

Note that $\bar{X}_k$ is nominally a transformation of Y_k

Determine the kth transformation by matching $\bar{X}_k$ with Y_k.

end if

next k

Set $P_{q+1} = P_{q,n_T}$. If $P_{q+1} = P_q$ ($\neq S$), then the system is not identifiable starting from S_0, and at least one of the registration points and possibly some of the transformations have to be estimated by other means.

If $P_{q+1} \neq P_q$, then set $q = q + 1$ and repeat step II.

If a number of the transformations parameter estimates are known *a priori*, then the analysis can proceed as above, except with enlarged initial sets P_0 and Q_0.

3.7 CONSISTENCY CHECK ON THE DATA

The following scheme provides a simple check on the consistency of the data by comparing the distances between registration points in two frames of reference. If one of the registration points has moved relative to the others then it is likely (but not guaranteed) that this will result in a change in one or more of the distances between it and other measured registration points.

Suppose $\tau > 0$ is a preassigned tolerance (generally related to the measurement accuracy).

For $k = 0, 1, \quad, n_T -$

I *Determine the set of registration points measured both in the* k*th position and in the* (k+1)*th position.* Set $C_k = S_k \cap S_{k+1}$. (Recall that S_k is the set of registration points measured in the kth position.)

II *Calculate distances.* Set

$$\bar{x}_{j,k} = \text{mean}\{x_i, j(i) = j, k(i) = k\}, \quad j \in C_k,$$

and define $\bar{x}_{j,k+1}$ similarly. For each pair $j,l \in C_k$, $j < l$, calculate $d_{jl,k} = \|\bar{x}_{j,k} - \bar{x}_{l,k}\|$, $d_{jl,k+1} = \|\bar{x}_{j,k+1} - \bar{x}_{l,k+1}\|$ and $\Delta_{jl,k} = d_{jl,k} - d_{jl,k+1}$.

III *Record exceptional results.* If $|\Delta_{jl,k}| > \tau$, store k, j, l and $\Delta_{jl,kr}$

4 SOFTWARE IMPLEMENTATION

Software based on the algorithms described above have been implemented by NPL. The minimum residual sum of squares is determined by a Gauss-Newton algorithm incorporating a safeguarded parabolic line search. This optimisation software uses Givens rotations to update the upper-triangular factor of the Jacobian matrix a row at a time, exploiting the block angular structure of the matrix saving computational time and memory.

The main features of the software are:

Portability. The software is written in ANSI Standard Fortran 77 and has been tested for portability using the Toolpack portability verifier (NAG, 1992) on a DEC Alpha running VMS.
Scope. The subroutines cater for an arbitrary number of points measured in an arbitrary number of positions in any order. In practice, the maximum size of the problem will be determined by the user's driver program which will define the maximum array sizes.
Computational requirements. The main optimisation algorithm implemented has $O(m_X n_T^2)$ performance per iteration, where m_X is the total number of measurements $\{x_i\}$ and n_T is the number of transformations $\{t_k\}$.
Memory requirements. The software requires $O((n_Y+n_T)n_T)$ memory, including work space.
Software validation. The main optimisation modules have been tested on reference data and results generated using null space methods (Forbes, 1994, Butler *et al.*, 1997).The tests show that the software is able to determine estimates of the transformation parameters without significant loss of numerical precision.

The main subroutines, documented below, are:

ATPCS	applies transformations to a set of measured points.
CHKDIS	performs a consistency check on the data by examining the distances between registration points from one position to the next.
LSPCS	implementation of the Gauss-Newton scheme to determine best estimates of the registration points and transformation parameters.
LSPTM	determines estimates of the transformation parameters linking a set of points measured in two frames of reference.
PCSCMR	determines the covariance matrix of a registration point.
PCSCMT	determines the covariance matrix of a set of transformation parameters.
PCSI	determines initial estimates of the registration points and transformation parameters through repeated calls to LSPTM.
RPDST	calculates the standard uncertainty for the distance between two registration points.

RTPDST calculates the standard uncertainty for the distance between a registration point and a transformed point.

TPDST calculates the standard uncertainty for the distance between two transformed points.

4.1 ATPCS

See section 3.4. Given data points $\{w_i\}_1^{m_W}$, indices $k(i)$, transformations T_k specified by transformation parameters t_k and fixed rotation matrices $R_{0,k}$, this module calculates the transformed points $\hat{w}_i = T_{k(i)}(w_i)$ or $\check{w}_i = T_{k(i)}^{-1}(w_i)$.

4.1.1 Signature

```
SUBROUTINE ATPCS(MW,NT,W,LDW,IW,T,R0,LDR0,INV)
INTEGER LDR0,LDW,MW,NT
LOGICAL INV
DOUBLE PRECISION R0(LDR0,*),T(*),W(LDW,*)
INTEGER IW(*)
```

4.1.2 Input and input/output parameters

MW — INTEGER specifying the number m_W of measurements of points to be transformed.

NT — INTEGER specifying the number n_T of transformations.

W — DOUBLE PRECISION 2-dimensional array of declared row length LDW storing x-, y- and z-coordinates of the points $\{w_i\}_1^{m_W}$ to be transformed.

IW — INTEGER 1-dimensional array storing the indexes $k(i)$. If w_i is a measurement of a point in the kth position then IW(I) = k.
Constraint: $0 \le$ IW(I) $\le$ NT.

Note. No transformation is applied to the ith point if IW(I) is out of bounds,

T — DOUBLE PRECISION 1-dimensional array storing the transformation parameters $\{a_k\}_1^{n_T}$ and $\{\theta_k\}_1^{n_T}$. The translation parameters a_k are stored in rows (6*k-5:6*k-3) and the rotation parameters θ_k are stored in rows (6*k-2:6*k).

R0 — DOUBLE PRECISION 2-dimensional array of declared row length LDR0 storing the fixed rotation matrices $\{R_{0,k}\}_1^{n_T}$. The kth rotation matrix is stored in R0(3*k-2:3*k,1:3).

Note. The kth transformation is defined by $T_k(x) = R_{z,k} R_{y,k} R_{x,k} R_{0,k}(x - a_k)$

INV — LOGICAL. If INV .EQ. .FALSE. on entry, the transformations T_k are applied to the data points. If INV .EQ. .TRUE. on entry, the inverse transformations T_k^{-1} are applied.

4.1.3 Output parameters

On exit, W contains the transformed data points. (See also IW.)

4.1.4 Dimensioning parameters

LDW
INTEGER giving the declared row dimension of W.
Constraint: LDW $\geq$ MW.

LDR0
INTEGER giving the declared row dimension of R0.
Constraint: LDR0 $\geq$ 3*NT.

4.2 CHKDIS

See section 3.7. If y_j and y_l and both measured in positions k and $k+1$, $k = 0, \ldots, n_T - 1$, this module compares $|d_{k,jl} - d_{k+1,jl}|$ with a tolerance where $d_{k,jl}$ represents the distance between points y_j and y_l as measured in the kth position.

4.2.1 Signature

```
SUBROUTINE CHKDIS(MX,NY,NT,X,LDX,IX,LDIX,TAU,MDE,IE,LDIE,DE,LDDE,
*                 IWS, INFORM)
INTEGER IX(LDIX,*),IE(LDIE,*),IWS(NY)
INTEGER MX,NY,NT,LDX,LDIX,MDE,LDIE,LDDE,INFORM
DOUBLE PRECISION X(LDX,*),DE(LDDE,*)
DOUBLE PRECISION TAU
```

4.2.2 Input and input/output parameters

MX — INTEGER specifying the number m_X of measurements of the registration points.

NY — INTEGER specifying the number n_Y of registration points.

NT — INTEGER specifying the number n_T of transformations.

X — DOUBLE PRECISION 2-dimensional array of declared row length LDX storing x-, y- and z-coordinates of the registration points $\{x_i\}_1^{m_X}$ measured in the various positions.

IX — INTEGER 2-dimensional array of declared row length LDIX storing the indexes $(j(i), k(i))$. If x_i is a measurement of the jth registration point in the kth position then IX(i,1) = j and IX(i,2) = k.

TAU — DOUBLE PRECISION variable storing the tolerance used in checking the distances.

MDE — INTEGER. On entry, MDE stores an upper bound on the number of exceptional distances to be recorded. On exit, MDE stores the actual number of exceptional distances recorded.

Note: If MDE ≥ NT*NY*(NY-1)/2 on entry, then all exceptional distances will definitely be recorded. For most practical applications, the actual number of exceptions is likely to be a small fraction of this number, assuming the TAU is not unrealistically small.

4.2.3 Output parameters

MDE — See above.

IE INTEGER 2-dimensional array of declared row dimension LDIE storing indices relating to the exceptional distances found. If IE(I,:) = (k,j,l), then DE(I,:) stores the distances between the *j*th and *l*th registration points as measured in the *k*th and (k+1)th positions.

DE DOUBLE PRECISION 2-dimensional array of declared row length LDDE storing the exceptional distanced found. If IE(I,:) = (k,j,l), then DE(i,1) stores the distances between the *j*th and *l*th registration points as measured in the *k*th position, DE(i,2) stores the distances between the *j*th and *l*th registration points as measured in the (*k*+1)th position and DE(i,3) stores DE(i,1) - DE(i,2). This information is stored only if ABS(DE(i,3) ≥ TAU.

4.2.4 Work space parameters

IWS INTEGER 1-dimensional array on length NY.

4.2.5 Dimensioning parameters

LDX INTEGER giving the declared row dimension of X.
Constraint: LDX ≥ MX.

LDIX INTEGER giving the declared row dimension of IX.
Constraint: LDIX ≥ MX.

LDIE INTEGER giving the declared row dimension of IE.
Constraint: LDIE ≥ MDE.

LDDE INTEGER giving the declared row dimension of IDE.
Constraint: LDDE ≥ MDE.

4.2.6 Diagnostic parameter

INFORM INTEGER

INFORM
- =K: The maximum number MDE of exceptional distances have been assigned while processing the measurements taken in positions K and K+1.
- =NT: All measurements have been processed.
- =-1: Invalid input value of MDE.
- =-2: Invalid LDX or LDIX.
- =-3: Invalid LDIE or LDDE.

4.3 LSPCS

See sections 2 and 3. Given data points $\{x_i\}_1^{m_X}$ representing measurements of registration points $\{y_i\}_1^{n_Y}$ in positions $k = 0, \ldots, n_T$, this module determines least squares best estimates of the position of the registration points in the initial (0th) position along with estimates of the transformations linking the frames for reference. Initial estimates of the optimisation parameters are required (see PCSI).

4.3.1 *Signature*

```
      SUBROUTINE LSPCS(MX,NY,NT,X,LDX,IX,LDIX,D,LDD,DTYP,Y,LDY,T,R0,
     +                 LDR0,E,LDE,U,LU,ENRM,SIGMA,TOLR,WS,LWS,INFORM)
      DOUBLE PRECISION ENRM,SIGMA,TOLR
      INTEGER DTYP,INFORM,LDD,LDE,LDIX,LDR0,LDX,LDY,LU,LWS,MX,NT,NY
      DOUBLE PRECISION D(LDD,*),E(LDE,*),U(LU),R0(LDR0,*),T(*),WS(LWS),
     +                 X(LDX,*),Y(LDY,*)
      INTEGER IX(LDIX,*)
```

4.3.2 *Input and input/output parameters*

MX INTEGER specifying the number m_X of measurements of the registration points.

NY INTEGER specifying the number n_Y of registration points.

NT INTEGER specifying the number n_T of transformations.

X DOUBLE PRECISION 2-dimensional array of declared row length LDX storing x-, y- and z-coordinates of the measurements $\{x_i\}_1^{m_X}$ of the registration points in the various positions.

IX INTEGER 2-dimensional array of declared row length LDIX storing the indexes $(j(i), k(i))$. If x_i is a measurement of the jth registration point in the kth position then IX(i,1) = j and IX(i,2) = k.
Constraint: $1 \le$ IX(i,1) $\le$ NY, $0 \le$ IX(i,2) $\le$ NT.

Note: It is required that each registration point is measured in at least one position and that at least three registration points are measured in each position.

D DOUBLE PRECISION 2-dimensional array of declared row length LDD storing the diagonal elements of the diagonal weighting matrices $\{D_i\}$. See DTYP.

DTYP INTEGER control parameter.
DTYP=1: each component of each measured point is potentially subject to a different weighting factor. The diagonal elements of D_i are stored in D(i,1:3).
DTYP=2: each measured point is potentially subject to a different weighting factor. The diagonal elements D_i are stored in D(I).

DTYP=3: each component (over all measured points) is potentially subject to a different weighting factor. The elements of D are stored in D(1:3).
Constraint: $1 \leq$ DTYP≤ 3

Y DOUBLE PRECISION 2-dimensional array of declared row length LDY storing estimates of the registration points $\{y_i\}_1^{n_Y}$ in the 0th position.

T DOUBLE PRECISION 1-dimensional array storing estimates of the transformation parameters $\{\boldsymbol{a}_k\}_1^{n_T}$ and $\{\theta_k\}_1^{n_T}$. The translation parameters $\boldsymbol{a}_k$ are stored in rows (6*k-5:6*k-3) and the rotation parameters θ_k are stored in rows (6*k-2:6*k).

R0 DOUBLE PRECISION 2-dimensional array of declared row length LDR0 storing the fixed rotation matrices $\{R_{0,k}\}_1^{n_T}$. The kth rotation matrix is stored in R0(3*k-2:3*k,1:3).

Note: The kth transformation is defined by $T_k(\boldsymbol{x}) = R_{z,k} R_{y,k} R_{x,k} R_{0,k}(\boldsymbol{x} - \boldsymbol{a}_k)$

TOLR DOUBLE PRECISION relative tolerance used to determine convergence criteria. On a successful exit, the relative error in the residual errors is of the order of TOLR.

Note: Suggested value of TOLR is 1.0*e*-9.

4.3.3 *Output parameters*

On exit, Y and T contain updated estimates of the registration points and transformation parameters, respectively.

E DOUBLE PRECISION 2-dimension array of declared row length LDE storing the weighted residuals $D_i\epsilon_i$ in E(i,1:3).

U DOUBLE PRECISION 1-dimensional array storing the packed upper-triangular factor for the Jacobian matrix. The non-zero elements of the Jacobian are stored row-wise in the first NY*(6 + 3*6*NT) + 6*NT*(6*NT+1)/2 elements of U.

ENRM DOUBLE PRECISION variable storing the norm of the weighted residuals, i.e.,

$$\left[\sum_i (D_i\epsilon_i)^T D_i\epsilon_i\right]^{1/2}$$

SIGMA DOUBLE PRECISION variable storing an estimate of the standard deviation of the weighted residual errors.

Note: SIGMA is zero if there are no degrees of freedom, i.e., 3*MX = 3*NY+6*NT.

4.3.4 Work space parameters

WS DOUBLE PRECISION 1-dimensional array.

4.3.5 Dimensioning parameters

LDX INTEGER giving the declared row dimension of X.
Constraint: LDX ≥ MX.

LDIX INTEGER giving the declared row dimension of IX.
Constraint: LDIX ≥ MX.

LDD INTEGER giving the declared row dimension of D.
Constraint: If DTYP = 1, LDD ≥ MX.
If DTYP = 2, LDD ≥ MX.
If DTYP = 3, LDD ≥ 3.

LDY INTEGER giving the declared row dimension of Y.
Constraint: LDY ≥ NY.

LDR0 INTEGER giving the declared row dimension of R0.
Constraint: LDR0 ≥ 3*NT.

LDE INTEGER giving the declared row dimension of E.
Constraint: LDE ≥ MX.

LU INTEGER giving the declared length or U.
Constraint: LR ≥ NY*(6+3*6*NT) + 6*NT*(6*NT+1)/2.

LWS INTEGER giving the length of WS.
Constraint: LWS ≥ 6 + 6*NY + 12*NT.

4.3.6 Diagnostic parameter

INFORM INTEGER
INFORM
=0: Normal termination
=-1: Invalid MX, NY or NT.
=-2: Invalid LDX, LDIX, LDD, LDY, LDR0, or LDE.
=-3: Invalid LU or LWS.
=-4: Elements of IX are out of bounds.
=-5: Weights in D are either negative or all zero.
=1: Jacobian system could not be upper-triangularised.
=2: Jacobian system could not be solved.
=3: Maximum number of iterations reached.

Notes. An INFORM exit value of 1 is not anticipated.

An INFORM exit value of 2 is possible if, for example, the registration points in one position are (almost exactly) co-linear.

An INFORM exit value of 3 is unlikely but possible if the entry estimates are extremely poor and the measurement results are extremely inaccurate. For data with measurement noise less that 10%, and estimates provided by PCSI, the algorithm should converge easily.

4.4 LSPTM

See section 3.6. Given two sets of points $\{x_i\}_1^m$ and $\{w_i\}_1^m$ in $\mathbb{R}^n$, this module calculates the rotation matrix R_0 and translation vector x_0 such that the transformed points

$$\hat{x}_i = R_0(x_i - x_0)$$

are as close to w_i in the least squares sense, i.e., R_0 and x_0 are such that

$$\sum_i^m (\hat{x}_i - w_i)^T(\hat{x}_i - w_i)$$

is minimised.

4.4.1 Signature

```
SUBROUTINE LSPTM(M,N,X,LDX,W,LDW,X0,R0,LDR0,S,WS,LWS,INFORM)
INTEGER        M,N,LDX,LDW,LDR0,LWS,INFORM
DOUBLE PRECISION X(LDX,*),W(LDW,*),R0(LDR0,N),S(N),WS(LWS)
```

4.4.2 Input parameters

M — INTEGER specifying the number of points m in the data sets X and W.

N — INTEGER specifying the dimension n (number of components).
Constraint: M ≥ N.

X — DOUBLE PRECISION 2-dimensional array of declared row dimension LDX storing the data points $\{x_i\}_1^m$.

W — DOUBLE PRECISION 2-dimensional array of declared row dimension LDW storing the data points $\{w_i\}_1^m$.

4.4.3 Output parameters

X0 — DOUBLE PRECISION 1-dimensional array storing the computed translation vector x_0.

R0 — DOUBLE PRECISION 2-dimensional array of declared row dimension LDR0 storing the computed $n \times n$ rotation matrix R_0.

S — DOUBLE PRECISION 1-dimensional array storing the singular values of VU^T

4.4.4 *Work space parameters*

WS DOUBLE PRECISION 1-dimensional work space array.

4.4.5 *Dimensioning parameters*

LDX INTEGER specifying the declared row dimension of X.
Constraint: LDX $\geq$ M.

LDW INTEGER specifying the declared row dimension of W.
Constraint: LDW $\geq$ M.

LDR0 INTEGER specifying the declared row dimension of R0.
Constraint: LDR0 $\geq$ N.

LWS INTEGER specifying the length of WS.
Constraint: LWS $\geq$3*N*N + 3*N.

4.4.6 *Diagnostic parameter*

INFORM INTEGER.

INFORM	=0:	Normal termination.
	=1:	Invalid M or N (M < N).
	=2:	Invalid LDX or LDW.
	=3:	Invalid LDR0.
	=4:	Invalid LWS.
	=5:	Singular values could not be computed.

Note: The software calls EISPACK module DSVDC to calculate the singular values of an N x N matrix. This module implements an iterative algorithm which on extremely rare occasions may fail to converge, giving rise to an exit INFORM value of 5 in LSPTM. It is not anticipated that this will occur within LSPTM.

4.5 PCSCMR

See section 3.5. This module calculates an estimate of the covariance matrix V_{y_j} associated with solution estimates of a registration point y_j.

Signature

```
SUBROUTINE PCSCMR(NY,NT,U,SIGMA,J,V,LDV,WS,LWS,INFORM)
DOUBLE PRECISION SIGMA
INTEGER INFORM,J,LDV,LWS,NT,NY
DOUBLE PRECISION U(*),V(LDV,*),WS(LWS)
```

Input parameters

NY — INTEGER specifying the number n_Y of registration points.

NT — INTEGER specifying the number n_T of transformations.

U — DOUBLE PRECISION 1-dimensional array storing the packed upper-triangular factor for the Jacobian matrix (as output) from LSPCS.

SIGMA — DOUBLE PRECISION variable storing an estimate of the standard deviation of the weighted residual errors (as output from LSPCS)

INTEGER specifying the registration point for which the covariance matrix is required.
Constraint: $1 \le J \le NY$.

Output parameters

V — DOUBLE PRECISION 2-dimension array of declared row dimension LDV storing the covariance matrix V_{y_j} of the fitted registration point y_j. The 3×3 matrix V_v is stored in the first 3 rows and columns of V.

Note: The covariances are assigned to the upper-triangle of V. The strictly lower-triangular elements are set to zero.

4.5.4 Work space parameter

WS — DOUBLE PRECISION 1-dimensional array of declared length LWS.

4.5.5 Dimensioning parameters

LDV — INTEGER giving the declared row dimension of V.
Constraint: $LDV \ge 3$.

LWS	INTEGER giving the declared length of LW.		
	Constraint:	LWS ≥ 6 + 3*(3+6*NT).	

4.5.6 Diagnostic parameter

INFORM	INTEGER		
	INFORM	=0:	Normal termination.
		=1:	Invalid J, LDV or LWS.
		=2:	Triangular system judged to be singular.

Note: An exit INFORM value of 2 is not anticipated if U is determined by a successful call to LSPCS.

4.6 PCSCMT

See section 3.5. This module calculates an estimate of the covariance matrix V_{t_k} associated with solution estimates of a set of transformation parameters t_k.

Signature

```
SUBROUTINE PCSCMT(NY,NT,U,SIGMA,K,V,LDV,WS,LWS,INFORM)
DOUBLE PRECISION SIGMA
INTEGER INFORM,K,LDV,LWS,NT,NY
DOUBLE PRECISION U(*),V(LDV,*),WS(LWS)
```

Input parameters

NY INTEGER specifying the number n_Y of registration points.

NT INTEGER specifying the number n_T of transformations.

U DOUBLE PRECISION 1-dimensional array storing the packed upper-triangular factor for the Jacobian matrix (as output from LSPCS).

SIGMA DOUBLE PRECISION variable storing an estimate of the standard deviation of the weighted residual errors (as output from LSPCS)

K INTEGER specifying the transformation parameters for which the covariance matrix is required.
Constraint: $1 \le K \le NT$.

4.6.3 *Output parameters*

V DOUBLE PRECISION 2-dimension array of declared row dimension LDV storing the covariance matrix V_{t_k} of the transformation parameters t_k. The 6×6 matrix V_{y_j} is stored in the first 6 rows and columns of V.

Note: The covariances are assigned to the upper-triangle of V. The strictly lower-triangular elements are set to zero.

4.6.4 *Work space parameter*

WS DOUBLE PRECISION 1-dimensional array of declared length LWS.

4.6.5 *Dimensioning parameters*

LDV INTEGER giving the declared row dimension of V.
Constraint: $LDV \ge 6$.

LWS	INTEGER giving the declared length of LW.		
	Constraint:	LWS $\geq$ 6*6*NT.	

Diagnostic parameter

INFORM	INTEGER		
	INFORM	=0:	Normal termination.
		=1:	Invalid K, LDV or LWS.
		=2:	Triangular system judged to be singular.

Note: An exit INFORM value of 2 is not anticipated if U is determined by a successful call to LSPCS.

4.7

See section 3.6. Given data points $\{x_i\}_1^{m_X}$ representing measurements of registration points $\{y_i\}_1^{n_Y}$ in positions $k = 0, \ldots, n_T$, this module determines initial estimates of the location of the registration points in the initial (0th) position along with estimates of the transformations linking the frames for reference.

4.7.1 *Signature*

```
SUBROUTINE PCSI(MX,NY,NT,X,LDX,IX,LDIX,Y,LDY,A,R0,LDR0,IY,IT,
               LWS,IWS,LIWS,INFORM)

INTEGER MX,NY,NT,LDX,LDIX,LDY,LDR0,LWS,LIWS,INFORM
INTEGER IW(LIWS),IY(NY),IT(NT),IX(LDIX,*)
DOUBLE PRECISION X(LDX,*),Y(LDY,*),A(*),R0(LDR0,*),WS(LWS)
```

4.7.2 *Input parameters*

MX INTEGER specifying the number m_X of measurements of the registration points.

NY INTEGER specifying the number n_Y of registration points.

NT INTEGER specifying the number n_T of transformations.

X DOUBLE PRECISION 2-dimensional array of declared row length LDX storing x-, y- and z-coordinates of the registration points $\{x_i\}_1^{m_X}$ measured in the various positions.

IX INTEGER 2-dimensional array of declared row length LDIX storing the indexes $(j(i), k(i))$. If x_i is a measurement of the jth registration point in the kth position then IX(i,1) = j and IX(i,2) = k.

In order for all registration points and transformation parameters to be determined, it is required that each registration point is measured in at least one position and that at least three registration points are measured in each position.

4.7.3 *Output parameters*

Y DOUBLE PRECISION 2-dimensional array of declared row length LDY storing estimates of the registration points $\{y_i\}_1^{n_Y}$ in the 0th position. See also IY.

A DOUBLE PRECISION 1-dimensional array of estimates of the translation parameters $\{a_k\}_1^{n_T}$. A(3*k-2:3*k) stores a_k. See also IT.

R0 DOUBLE PRECISION 2-dimensional array of declared row length LDR0 storing estimates of the rotation matrices. The kth rotation $R_{0,k}$ matrix is stored in R0(3*k-3:3*k,1:3). See also IT.

Note: The kth transformation T_k is given by $T_k(\boldsymbol{x}) = R_{0,k}(\boldsymbol{x} - \boldsymbol{a}_k)$.

IY INTEGER 1-dimensional array indicating which registration points have been estimated. If the coordinates of $\boldsymbol{y}_j$ have been estimated, then IY(j) = 1; otherwise IY(j) = 0.

IT INTEGER 1-dimensional array indicating which transformations have been estimated. If $\boldsymbol{a}_k$ and $R_{0,k}$ have been estimated, then IT(k) = 1; otherwise IT(k) = 0.

4.7.4 Work space parameters

WS DOUBLE PRECISION 1-dimensional array.

IWS INTEGER 1-dimensional array.

4.7.5 Dimensioning parameters

LDX INTEGER giving the declared row dimension of X.
Constraint: LDX ≥ MX.

LDIX INTEGER giving the declared row dimension of IX.
Constraint: LDIX ≥ MX.

LDY INTEGER giving the declared row dimension of Y.
Constraint: LDY ≥ NY.

LDR0 INTEGER giving the declared row dimension of R0.
Constraint: LDR0 ≥ 3*NT.

LWS length of WS.
Constraint: LW ≥ 9*NY + 93

LIWS length of IWS.
Constraint: LIWS ≥ 4*NY.

4.7.6 Diagnostic parameter

INFORM INTEGER

INFORM
- =0: Normal termination.
- =1: Invalid LDX or LDIX.
- =2: Invalid LDY.
- =3: Invalid LDR0.
- =4: Invalid LWS.
- =5: Invalid LIWS.

=6: LSPTM returned with a non-zero INFORM value.

Note: An exit INFORM value of 6 is not anticipated.

4.8 RPDST

See section 3.5. This module calculates the distance between the two registration points along with its standard uncertainty.

4.8.1 Signature

```
SUBROUTINE RPDST(NY,NT,Y,LDY,U,SIGMA,J1,J2,D,UD,WS,LWS,INFORM)
DOUBLE PRECISION D,SIGMA,UD
INTEGER INFORM,J1,J2,LDY,LWS,NT,NY
DOUBLE PRECISION U(*),WS(LWS),Y(LDY,*)
```

4.8.2 Input parameters

NY — INTEGER specifying the number n_Y of registration points.

NT — INTEGER specifying the number n_T of transformations.

Y — DOUBLE PRECISION 2-dimensional array of declared row length LDY storing solution estimates of the registration points $\{y_i\}_1^{n_Y}$ in the 0th position.

U — DOUBLE PRECISION 1-dimensional array storing the packed upper-triangular factor for the Jacobian matrix (as output from LSPCS).

SIGMA — DOUBLE PRECISION variable storing an estimate of the standard deviation of the weighted residual errors (as output from LSPCS)

J1, J2 — INTEGERs storing the pair of indices j_1, j_2 of the registration points for which the distance and standard uncertainty calculations are required.
Constraint: $1 \le$ J1, J2 $\le$ NY, J1 $\ne$ J2.

4.8.3 Output parameters

D — DOUBLE PRECISION storing the distance $\|y_{j_1} - y_{j_2}\|$ between the pairs of registration points specified by J1 and J2.

UD — DOUBLE PRECISION storing the standard uncertainty of the distance D.

4.8.4 Work space parameter

WS — DOUBLE PRECISION 1-dimensional array of declared length LWS

4.8.5 Dimensioning parameters

LDY — INTEGER giving the declared row dimension of Y
Constraint: LDY $\ge$ NY.

INTEGER giving the declared length of WS.
Constraint: LWS ≥ 12 + 6*NT.

Diagnostic parameter

INFORM INTEGER.

INFORM	= 0	Normal exit; all requested calculations completed..
	= 1	Invalid LDY, LWS, J1 or J2.
	= 2	Triangular system could not be solved.

Note An exit INFORM value of 2 is not anticipated if U is determined by a successful call to LSPCS.

4.9 RTPDST

See section 3.5. This module calculates the distance between a registration point and a transformed point along with its standard uncertainty.

4.9.1 Signature

```
SUBROUTINE RTPDST(NY,NT,Y,LDY,T,R0,LDR0,U,SIGMA,J,W,INV,K,WT,D,UD,
+                    WS,LWS,INFORM)
DOUBLE PRECISION D,SIGMA,UD
INTEGER INFORM,J,K,LDR0,LDY,LWS,NT,NY
LOGICAL INV
DOUBLE PRECISION U(*),R0(LDR0,*),T(*),W(3),WS(LWS),WT(3),Y(LDY,*)
```

4.9.2 Input parameters

NT — INTEGER specifying the number n_T of transformations.

NY — INTEGER specifying the number n_Y of registration points.

Y — DOUBLE PRECISION 2-dimensional array of declared row length LDY storing solution estimates of the registration points $\{y_i\}_1^{n_Y}$ in the 0th position.

T — DOUBLE PRECISION 1-dimensional array storing the transformation parameters $t_k = (a_k^T, \theta_k^T)^T$.

R0 — DOUBLE PRECISION 2-dimensional array of declared row length LDR0 storing the fixed rotation matrices. The *k*th fixed rotation matrix $R_{0,k}$ is stored as R0(3*k-3:3*k,1:3).

U — DOUBLE PRECISION 1-dimensional array storing the packed upper-triangular factor for the Jacobian matrix (as output from LSPCS).

SIGMA — DOUBLE PRECISION variable storing an estimate of the standard deviation of the weighted residual errors (as output from LSPCS)

J — INTEGER variable storing the pair of index of the registration point
Constraint: $1 \le J \le NY$.

W — DOUBLE PRECISION 1-dimensional array storing the point w to be transformed.

INV — LOGICAL specifying whether the calculations relate to the transformation itself, or its inverse.

K — INTEGER variable storing the index k of the transformation T_k to be applied.
Constraint: $0 \le K \le NT$.

4.9.3 Output parameters

WT DOUBLE PRECISION 1-dimensional array storing the transformed point $\hat{w} = T_k(w)$ or $\hat{w} = T_k^{-1}(w)$.

D DOUBLE PRECISION storing the distance $\|y_j - \hat{w}\|$ between the registration point and the transformed point.

UD DOUBLE PRECISION storing the standard uncertainty of the distance D.

4.9.4 Work space parameter

WS DOUBLE PRECISION 1-dimensional array of declared length LWS.

4.9.5 Dimensioning parameters

LDY INTEGER giving the declared row dimension of Y.
Constraint: LDY ≥ NY.

LDR0 INTEGER giving the declared row dimension of LDR0.
Constraint: LDR0 ≥ 3*NT.

LWS INTEGER giving the declared length of WS.
Constraint: LWS ≥ 9 + 6*NT.

4.9.6 Diagnostic parameter

INFORM INTEGER.

INFORM	= 0	Normal exit; all requested calculations completed..
	= 1	Invalid LDY, LDR0, LWS, J or K.
	= 2	Triangular system could not be solved.

Note: An exit INFORM value of 2 is not anticipated if U is determined by a successful call to LSPCS.

4.10 TPDST

See section 3.5. This module calculates the distance between two transformed points transformed by different transformations along with its standard uncertainty. Note that the uncertainty in a set of transformation parameters will not contribute to the variance of the distance between two points transformed by the same transformation.

4.10.1 Signature

```
      SUBROUTINE TPDST(NT,T,R0,LDR0,UT,SIGMA,W1,W2,INV1,INV2,K1,K2,
     *                  WT1,WT2,D,UD,WS,LWS,INFORM)

      LOGICAL INV1,INV2
      INTEGER NT,LDR0,K1,K2,LWS,INFORM
      DOUBLE PRECISION T(*),R0(LDR0,*),UT(*),W1(3),W2(3),WT1(3),WT2(3)
      DOUBLE PRECISION SIGMA,D,UD
```

4.10.2 Input parameters

NT — INTEGER specifying the number n_T of transformations.

T — DOUBLE PRECISION 1-dimensional array storing the transformation parameters $t_k = (a_k^T, \theta_k^T)^T$.

R0 — DOUBLE PRECISION 2-dimensional array of declared row length LDR0 storing the fixed rotation matrices $R_{0,k}$.

W1,W2 — DOUBLE PRECISION 1-dimensional arrays of length =3 storing the coordinates of the points w_p, $p = 1, 2$.

INV1,INV2 — LOGICALs specifying whether the calculations relate to the transformation itself, or its inverse.

K1,K2 — INTEGERS specifying which transformations are to be applied.
Constraints: $0 \le$ K1,K2 $\le$ NT.

UT — DOUBLE PRECISION 1-dimensional array of storing the upper-triangular factor corresponding to the parameters T.

Note: If U is as output from LSPCS, then UT is the sub-array of U of length 6*NT*(6*NT+1)/2 starting at element NY*(6 + 3*6*NT)+1.

SIGMA — DOUBLE PRECISION variable storing an estimate of the standard deviation of the weighted residual errors.

4.10.3 Output parameters

WT1,WT2 — DOUBLE PRECISION 1-dimensional arrays of declared length 3 storing the computed transformed points $\hat{w}_p = T_p(w_p)$ or $\hat{w}_p = T_p^{-1}(w_p)$.

D — DOUBLE PRECISION scalar variable storing the distance $\|\hat{w}_1 - \hat{w}_2\|$ between the transformed points.

UD — DOUBLE PRECISION scalar variable storing the standard uncertainty of D.

4.10.4 Work space parameter

WS — DOUBLE PRECISION 1-dimensional array of declared length LWS.

4.10.5 Dimensioning parameters

LDR0 — INTEGER giving the declared row dimension of R0.
Constraint: LDR0 ≥ 3*NT.

LWS — INTEGER giving the declared row dimension of WS.
Constraint: LWS ≥ 6*NT.

4.10.6 Diagnostic parameter

INFORM — INTEGER

INFORM	= 0	Normal termination.
	= 1	Invalid LDR0, K1 or K2.

Note: No checks are performed to determine if UT is singular. If UT represents a matrix that is singular then a run-time error will occur. If UT is determined by a successful call to LSPCS then no problem is anticipated (as UT will have been checked within LSPCS).

4.11 SUPPORT MODULES

The modules documented above call a number of support modules which are listed here for completeness. The modular structure for the sample programme (Annex A) is listed in Annex D.

4.11.1 Modules related to position calibration

GNPCS	Implements the Gauss-Newton iterative scheme.
USPCS	Determines the upper-triangular system associated with solving for the Gauss-Newton step (sections 3.1, 3.3).
FGPCS	Function and gradient evaluation associated with one measured point to determine three rows of the Jacobian matrix (section 3.2).
FPCS	Function evaluation for all the data points used to evaluate the objective function along a step direction within the parabolic search (section 3.1).
ROTTR3	Calculates a transformed point along with derivative information.

4.11.2 Modules related to Gauss-Newton optimisation

GNCC	Check if convergence criteria have been met (section 3.1).
PARBLS	Parabolic line search (section 3.1).

4.11.3 Linear algebra modules

DRBCM	Determine the covariance matrix associated with a block of parameters in a triangularised regular block angular system (section 3.5.2). (Regular here signifies that each diagonal block is of the same dimension.)
DRBMV	$\boldsymbol{x} := R\boldsymbol{x}$ or $\boldsymbol{x} := R^{T}\boldsymbol{x}$ where R is a regular block-angular upper-triangular matrix.
DRBSL	$\boldsymbol{x} := R^{-1}\boldsymbol{x}$ or $\boldsymbol{x} := R^{-T}\boldsymbol{x}$ where R is a regular block-angular upper-triangular matrix.
DTPCD	Check for near-zero diagonal elements in a upper-triangular matrix.
DTPCM	Determine the covariance matrix for a subset of parameters from the upper-triangular factor (section 3.1).
DTPUR	Update an upper-triangular system by a row.

4.11.4 Basic support modules

DCONSV Assign a constant to each element of an vector.

DMAXV Determine the maximum element of a vector.

DMEANM Determine the means of the columns of a matrix.

DMEANV Determine the mean of the elements of a vector.

DMINV Determine the minimum element of a vector.

DNRM2V Normalise a vector.

XTWMC Given two $m \times n$ matrices X and W, form $\tilde{X}^T\tilde{W}$ where $\tilde{A}$ represents A mean-centred (section 3.6).

4.11.5 EISPACK modules

See Smith *et al.*, 1976. (See also Anderson, *et al.*, 1992.)

DSVDC.

4.11.6 LINPACK modules

See Bunch *et al.*, 1979. (See also Anderson, *et al.*, 1992.)

DGEDI, DGEFA.

4.11.7 BLAS modules

See Lawson *et al.*, 1979, Dongarra *et al.*, 1988 and Dongarra *et al.*, 1990. (See also Anderson, *et al.*, 1992.)

DAXPY, DCOPY, DDOT, DGEMM, DGEMV, DNRM2, DROT, DROTG, DSCAL, DSYRK, DTPMV, DTPSV.

5 SAMPLE PROGRAM, DATA AND RESULTS

5. DRPCS

The module DRPCS, listed in Annex A, is an example driver routine which calls CHKDIS, PCSI, LSPCS, RPDST, RTPDST, TPDST, PCSCMR, and PCSCMT, reading input data from a file and outputting results to a file.

The module performs the following tasks

Assign parameters defining the size of the fixed length arrays:

MAXNT	maximum number of transformations $\{t_k\}$
MAXNY	maximum number of registration points $\{y_j\}$.
MAXMX	maximum number of measurements $\{x_i\}$.
MAXDE	maximum number of exceptional distances to be recorded within CHKDIS.

Constraints:

$1 \leq$ NT $\leq$ MAXNT
$3 \leq$ NY $\leq$ MAXNY
$6 \leq$ MX $\leq$ MAXMX (measurements $\{x_i\}$ of registration points)
$0 \leq$ MW $\leq$ MAXMX (additional measurements $\{w_i\}$ to be transformed)

Assign all weights (elements of the diagonals of D_i) to

2 Read the data X and IX.

Read NT, NY, MX
Read X(I,:), I = 1,..,MX
Read IX(I,:), I = 1,..,MX

3 Check the distances: call CHKDIS.

4 Determine initial estimates: call PCSI.

5 Determine best estimates call LSPCS.

6 Read in data to be transformed.

Read MW
Read W(I,:), I = 1,..,MW
Read IW(I), I = 1,..,MW

7 Transform data points (inverse transformations): call ATPCS

8 Calculate distances between registration points and associated standard uncertainties:

Read NSY
Read J, L call RPDST, I = 1,...,NSY.

9 Calculate distances between registration points and transformed points and associated standard uncertainties:

Read NSYW
Read J, L call RTPDST, I = 1,...,NSYW.

10 Calculate distances between transformed points and associated standard uncertainties:

Read NSW
Read J, L call TPDST, I = 1,...,NSW.

1 Calculate covariance matrices associated with the registration points:

call PCSCMR, J = 1,...,NY.

12 Calculate covariance matrices associated with the transformation parameters:

call PCSCMT, K = 1, ..., NT.

Information is required of the user in the form of input from the terminal:

Name of the input data file (with no file extension; it is assumed that the data file has extension .DAT). e.g.

```
xtest1
```

Tolerance with which to compare differences in distances between registration points as measured in different positions.

Relative tolerance for convergence criteria.

Print residual errors. Enter "1" for yes, "0" for no.

Print covariance matrices. Enter "1" for yes, "0" for no.

The module requires an input date *filename*.DAT with a format following the scheme below. Two examples are listed in Annex B.

```
NT  NY MX
X1  Y1  Z1
X2  Y2  Z2
...
XMX YMX ZMX
J1  K1
```

```
J2   K2
...
JMX  KMX
MW
U1   V1   W1
U2   V2   W2
...
UMW  VMW  WMW
K1
K2
...
KMW
NSY
J1    L1
J2    L2
...
JNSY  LNSY
NSYW
J1     L1
J2     L2
...
JNSYW  LNSYW
NSW
J1    L1
J2    L2

JNSW  LNSW
```

Any of MW, NSY, NSYW and NSW can be zero (with the constraint that if MW is zero, then NSYW and NSW are also zero).

5.2 TWO SAMPLE DATA SETS

Two sample data sets are (partially) listed in Annex B.

5.2.1 XTEST1.DAT

The first data set, XTEST1.DAT involves 16 measurements (MX = 16) of 7 registration points Y1, ...,Y7, (NY = 7) in an initial position and three subsequent positions (NT=3) according to the following table.

Position	Points measured
0	1,2,3,4
1	2,3,4,5
2	3,4,5,6
3	4,5,6,7

Eight points W1, ...,W8, 2 points in each of the 4 positions are to be transformed into the initial frame of reference (MW=8).

Distances and associated statistics are requested for all relevant pairs of points: 21 distances between registration points corresponding to the pairs (Y1,Y2), (Y1,Y3), ..., (Y6,Y7), (NSY=21), 56 distances between registration points and transformed points corresponding to the pairs (Y1,W1), (Y1,W2), ..., (Y7,W8), (NSYW=56) and 28 distances between transformed points corresponding to the pairs (W1,W2), (W1,W3), ..., (W7,W8) (NSW=28).

XTEST2.DAT

The second data set, XTEST2.DAT involves 14 measurements (MX = 14) of 5 registration points Y1, ...,Y5, (NY = 5) in an initial position and three subsequent positions (NT=3) in a random way specified by the indexing array. Ten points W1, ...,W10, are to be transformed into the initial frame of reference (MW=10).

Distances and associated statistics are requested for all relevant pairs of points: 10 distances between registration points corresponding to the pairs (Y1,Y2), (Y1,Y3), ..., (Y4,Y5), (NSY=10), 50 distances between registration points and transformed points corresponding to the pairs (Y1,W1), (Y1,W2), ..., (Y5,W10), (NSYW=50) and 45 distances between transformed points corresponding to the pairs (W1,W2), (W1,W3), ..., (W9,W10), (NSW=45).

5.3 TEST RESULTS

The results output by DRPCS for input data XTEST1.DAT and XTEST2.DAT are listed in Annex C in XTEST1.RES and XTEST2.RES, respectively. The results were obtained by running DRPCS on a DEC Alpha under the VMS operating system.

XTEST1.RES

After some summary information, the first set of results under `Exceptional distances` record changes in measured distances which are judged to be out of tolerance. The first column gives the position K, the second and third column give the indices J and L of the registration points, the fourth column gives the distance D(K) between YJ and YL as measured in the Kth position, while the next column gives the distance D(K+1) as measured in the (K+1)th position. The last column gives the difference D(K) - D(K+1).

The initial estimates of the registration points and transformation parameters are listed next along with the fixed rotation matrices. Note that the estimates of the parameters associated with the rotation angles are identically zero.

The solution estimates of the parameters are then given and can be seen to be close to the initial estimates.

The residual errors are then printed. For measurements 2 and 13 the residual errors are zero. This is because registration points Y1 and Y7 are measured in only one position and are therefore determined by the solution of a square system of equations (rather than an overdetermined system).

The results file finishes with the requested distance and statistics calculations. Note that distances between points transformed by the same transformation (e.g. W1 and W2) are assigned a zero standard uncertainty. Note also that only the elements of the upper-triangles of the symmetrical covariance matrices are calculated.

5.3.2 XTEST2.RES

The results file XTEST2.RES has the same format as XTEST1.RES. Note that no residual errors are identically zero since all the registration points are measured at least twice.

ACKNOWLEDGEMENTS

The work reported on here was jointly supported by the European Commission Directorate-General XII, Measurement and Testing (contract MAT1-CT94-00012) and the Department of Trade and Industry's National Measurement System Policy Unit Programme for Length. The contribution of project team members M G Cox, N R Cross, J R Dotson, L Ferris, D R Flack, S A Hannaby, P M Harris and G N Peggs (NPL), E Prieto (CEM, Madrid), R Corta (Tekniker, Eibar, Spain) and J O'Donnell (NML, Dublin) is gratefully acknowledged. We thank M G Cox, L Ferris and B J Chorley (NPL) for careful reviews of drafts of this document.

6 REFERENCES

Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKinney, A., Ostrouchov, S. and Sorensen, D. *LAPACK user's guide.* Philadelephia, SIAM, 1992.

Bunch, J., Dongarra, J. J., Moler, C. and Stewart, G. W. *LINPACK user's guide.* Philadelphia, SIAM, 1979.

Butler, B. P., Cox, M. G. , Forbes, A. B., Hannaby, S. A., and Harris, P. M. A methodology for testing the numerical correctness of approximation and optimisation software. In *The Quality of Numerical Software: Assessment and Enhancement,* Boisvert, R., Ed., London, Chapman & Hall, 1997, 138-151.

Cox, M. G. Linear algebra support modules for approximation and other software. In *Scientific Software Systems*, Mason, J. C. and Cox, M. G., Eds., London, Chapman and Hall, 1990, 21-29.

Cox, M. G., Cross, N. R., Flack, D. R., Forbes, A. B. and Peggs, G. N. Measurement of artefacts using repositioning methods. *NPL Report, CLM 2,* November 1997.

Dongarra, J. J., DuCroz, J., Hammerling, S. and Hanson, R. An extended set of Fortran Basic

Linear Algebra Subprograms. *ACM Trans. on Math. Soft.*, 1988, **14**, 1-32.

Dongarra, J. J., Duff, I., Du Croz, J. and Hammarlng, S. (1990): A set of level 3 Basic Linear Algebra Subprograms. *ACM Trans. on Math. Soft.*, **14**, 1-32.

Forbes, A. B. Position calibration software for the form assessment of complex workpieces. *NPL Report DITC 203/92*, July 1992.

Forbes, A. B. Validation of assessment software in dimensional metrology. *NPL Report DITC 225/94*, February 1994.

Gill, P. E., Murray, W. and Wright, M. H. *Practical Optimization*. London, Academic Press, 1981.

Golub, G. H. and Van Loan, C. F. *Matrix Computations*. Baltimore, Johns Hopkins University Press, 1996, (3rd edition).

Lawson, C., Hanson, R., Kincaid, D. and Krogh, F (1976): Basic Linear Algebra Subprograms for Fortran Usage. *ACM Trans. on Math. Soft.* **5**, 308-325.

NAG. *NAGWare f77 Tools (VAX/VMS)*. Oxford, The Numerical Algorithms Group, Ltd, 1992

Smith, B. T., Boyle, J. M., Dongarra, J. J., Garbow, B. S., Ikebe, Y., Clema, V. C. and Moler, C. B. (1977): *Matrix Eigensystems Routines - EISPACK guide*. New York, Springer-Verlag, Lecture Notes in Computer Science, **51**.

A LISTING OF DRPCS.FOR

```
c ==================================================================
c     DRPCS.FOR
c
c     Example driver module (main program) for position calibration.
c
c     DRPCS executes the following main steps:
c
c     1) Reads in data X and IX.
c
c     2) Calls CHKDIS to check changes in distances and prints out
c        exceptions.
c
c     3) Calls PCSI to determine initial estimates, reporting if not
c        all the parameters have been estimated.
c
c     4) Calls LSPCS to determine solution estimates of the parameters.
c
c     5) Reads in data to be transformed, if any.
c
c     6) Calls ATPCS to transform data, if required.
c
c     7) Calculates required distances and statistics associated with
c
c        pairs of registration points, calling RPDST,
         registration and transformed points, calling RTPDST
         pairs of transformed points, calling TPDST.

      8) Calculates required covariance matrices associated with
c        registration points and transformation parameters, calling
c        PCSCMR and PCSCMT.
c     ----------------------------------------------------------------
c     Version 1.0                Released 01 February 1998
c     Specification document     NPL Report CISE 14/98
c     Authors                    B P Butler, A B Forbes, P D Kenward
c                                Centre for Information Systems
c                                Engineering
c                                National Physical Laboratory
c                                Queens Road
c                                Teddington
c                                Middlesex
c                                TW11 0LW
c                                Great Britain
c
c =================
      PROGRAM DRPCS
c ----------------------------------
c     Variable declarations
c ----------------------------------
c     Assign maximum size of problem
c
C     .. Parameters ..
      INTEGER MAXNY,MAXNT,MAXMX,MAXDE
      PARAMETER (MAXNY=200,MAXNT=100,MAXMX=2000,MAXDE=200)
c
c     The following parameters depend on the size of the problem.
c
      INTEGER LDE,LDX,LDR0,LDIX,LDED,LDW,LDY,LT,LDIED
      PARAMETER (LDE=MAXMX,LDX=MAXMX,LDR0=3*MAXNT,LDIX=MAXMX,LDED=MAXDE
```

```
     +           LDW=MAXMX,LDY=MAXNY,LT=6*MAXNT,LDIED=MAXDE)
c
c      Assign size of upper-triangle array.
c      Uj needs 6, Bj needs 3*6*MAXNT.
c
       INTEGER LR
       PARAMETER (LR= (6+3*6*MAXNT)*MAXNY+6*MAXNT* (6*MAXNT+1)/2)
c
c      Assign size of workspace
c
       INTEGER LWS,LIWS
       PARAMETER (LWS=93+9*MAXNY+6*6*MAXNT,LIWS=MAXNY+MAXNT+6+4*MAXNY)
c
c      .. Local Scalars ..
       DOUBLE PRECISION D12,ENRM,SIGMA,TAU,TOLR,UD12
       INTEGER DTYP,I,INAT,INAY,INFO,INFORM,IREF,IRT,J,J1,J2,K,K1,K2,KIT,
     +         KIWS,KIY,L,LDD,LKIWS,MDE,MW,MX,NNT,NSW,NSY,NSYW,NT,NY
       LOGICAL INV1,INV2
       CHARACTER*20 FNAM
       CHARACTER*24 INFIL,OUTFIL
C      ..
C      .. Local Arrays ..
       DOUBLE PRECISION D(1,3),E(LDE,3),ED(LDED,3),R(LR),R0(LDR0,3)
     +                  T(LT),VT(6,6),W(LDW,3),W1(3),W2(3),WS(LWS),
     +                  WT1(3),WT2(3),X(LDX,3),Y(LDY,3)
       INTEGER IED(LDIED,3),IWS(LIWS),IX(LDIX,2)
C      ..
C      ------------------------------------------------------------
C      DRPCS calls:
C
C      .. External Subroutines ..
       EXTERNAL ATPCS,CHKDIS,DCONSV,DCOPY,LSPCS,PCSCMR,PCSCMT,PCSI,RPDST,
     +          RTPDST,TPDST
C
C      ------------------------------------------------------------
C      .. Intrinsic Functions ..
       INTRINSIC INDEX
C      ..
       INTEGER INF,OUTF
       PARAMETER (INF=1,OUTF=2)
       DOUBLE PRECISION ONE,ZERO
       PARAMETER (ONE=1.0D0,ZERO=0.0D0)
       CHARACTER*60 AUTHRS,VERSN
       PARAMETER (AUTHRS='B P Butler, A B Forbes, P D Kenward, NPL'
     +            VERSN='Version 1.0/VMS Release date 1998-02-01')
C
C      .. Data statements
       DATA MDE/MAXDE/
C      ..
c ------------------------------------------------
c      Start of calculations
c ------------------------------------------------
c      Assign the initial values of weights

       CALL DCONSV(3,ONE,D,1)
       DTYP = 3
       LDD = 1
c
c      The program identifies itself..
c
       WRITE (6,FMT=*)
     + '  ** PROGRAM DRPCS: Position Calibration Software **'
```

```
      WRITE (6,FMT=*) VERSN
      WRITE (6,FMT=*) AUTHRS
c
c     Invite the user to identify an input file
c
      WRITE (6,FMT=*) 'Please enter name of data file (.dat assumed)'
      READ (6,FMT='(A20)') FNAM
c
c     Now assign infile and outfile...
c
      INFIL = FNAM(1:INDEX(FNAM,' ')-1)//'.dat'
      OUTFIL = FNAM(1:INDEX(FNAM,' ')-1)//'.res'
      OPEN (INF,FILE=INFIL)
      OPEN (OUTF,FILE=OUTFIL)
c
c     Tell the user what files are being used..
c
      WRITE (6,FMT=*)
      WRITE (6,FMT=*) 'Input  file is ',INFIL
      WRITE (6,FMT=*) 'Output file is ',OUTFIL
c
c     Begin logging decisions and results to the output file
c
      WRITE (OUTF,FMT=*) ' '
      WRITE (OUTF,FMT=*) 'DRPCS  position calibration software'
      WRITE (OUTF,FMT=*) ' '

      WRITE (OUTF,FMT=*) VERSN
      WRITE (OUTF,FMT=*) AUTHRS
      WRITE (OUTF,FMT=*)
      WRITE (OUTF,FMT=*) 'Input file is ',INFIL
c
c     Read data from input file.
c
c     Read actual dimensions
c
      WRITE (OUTF,FMT=*
      WRITE (OUTF,FMT=*  'Input data statistics:'
c
      READ (INF,FMT=*) NT,NY,MX
      WRITE (OUTF,FMT=9140) NT
      WRITE (OUTF,FMT=9130) NY
      WRITE (OUTF,FMT=9120) MX
c
c     STOP if NT, NY, MX are invalid.
c
      IF  (NT.GT.MAXNT)  OR  (NY.GT.MAXNY)  OR. (MX.GT.MAXMX)  THEN

         WRITE (OUTF,FMT=*)
         WRITE (OUTF,FMT=*) 'Maximum number of transformations:     '
     +     MAXNT
         WRITE (OUTF,FMT=*) 'Maximum number of registration points:'
     +     MAXNY
         WRITE (OUTF,FMT=*) 'Maximum number of measurements:        '
     +     MAXMX
         WRITE (OUTF,FMT=*) '*** Maximum exceeded ***'

         WRITE (OUTF,FMT=9090)
         WRITE (6,FMT=9090)
         STOP
      END IF
```

```
c
c     STOP if there is not enough data.
c
      IF  (NT.LT.1) .OR  (NY.LT.3)  OR  (MX.LT.6)  THEN
         WRITE (OUTF,FMT=*)
         WRITE (OUTF,FMT=*) 'Minimum number of transformations:     ',1
         WRITE (OUTF,FMT=*) 'Minimum number of registration points:',3
         WRITE (OUTF,FMT=*) 'Minimum number of measurements:        ',6
         WRITE (OUTF,FMT=*) '*** Minimum exceeded ***'
c
         WRITE (OUTF,FMT=9090)
         WRITE (6,FMT=9090)
         STOP
      END IF
c
c     Compute derived dimensioning parameters.
c
      NNT = 6*NT
c
c     Read measured points X and the index array IX which identify the
c     associated registration point and transformation indices j,k...
c
      DO 10 I = 1,MX
         READ (INF,FMT=*  (X(I,J),J=1,3)
   10 CONTINUE
c
      DO 20 I = 1,MX
         READ (INF,FMT=*  (IX(I,J),J=1,2)
   20 CONTINUE

      Call CHKDIS to check the distances

      WRITE (OUTF,FMT=*)
      WRITE (OUTF,FMT=*) 'Distance consistency check: CHKDIS'
      WRITE (6,FMT=*) 'Please enter tolerance on distances:'
      READ (6,FMT=*) TAU

      IF (TAU.LT.ZERO) TAU = ZERO

      WRITE (OUTF,FMT=9100) TAU

      CALL CHKDIS(MX,NY,NT,X,LDX,IX,LDIX,TAU,MDE,IED,LDIED,ED,LDED,IWS,
     +            INFO)
c
c     STOP if CHKDIS returns INFO not equal to zero
c
      IF (INFO.GE.0) THEN
         WRITE (OUTF,FMT=9110) INFO
      ELSE
         WRITE (6,FMT=*) 'CHKDIS exit value of INFO =   ',INFO
         WRITE (OUTF,FMT=9090)
         WRITE (6,FMT=9090)
         STOP
      END IF
c
      WRITE (OUTF,FMT=9300) MDE

      IF (MDE.GT.0) THEN

         WRITE (OUTF,FMT=9310) (IED(I,1),IED(I,2),IED(I,3),ED(I,1),
     +      ED(I,2),ED(I,3),I=1,MDE)
```

```
      END IF
c
c
c Call PCSI to determine initial estimates of the registration points
c and transformation parameters.

      KIY = 1
      KIT = NY + 1
      KIWS = KIT + NT
      LKIWS = 4*NY
c
      WRITE (OUTF,FMT=*)
      WRITE (OUTF,FMT=*) 'Initial estimation module: PCSI'

      CALL PCSI(MX,NY,NT,X,LDX,IX,LDIX,Y,LDY,T,R0,LDR0,IWS(KIY),
     .          IWS(KIT),WS,LWS,IWS(KIWS),LKIWS,INFORM)
c
c     Check INFORM
c
      IF (INFORM.GT.0) THEN
         WRITE (6,FMT=*) 'PCSI exit value of INFORM:  ',INFORM
         WRITE (OUTF,FMT=*) 'PCSI exit value of INFORM:  ',INFORM
         WRITE (OUTF,FMT=9090)
         WRITE (6,FMT=9090)
         STOP
      END IF
c
c     Check IY and IT
c
      INAY = 0

      DO 30 J = 1,NY

         IF (IWS(KIY+J-1).EQ.0) INAY = INAY + 1

   30 CONTINUE

      IF (INAY.GT.0) THEN

         WRITE (OUTF,FMT=9050)
         WRITE (6,FMT=9060)
c
         DO 40 J = 1,NY
c
            IF (IWS(KIY+J-1).EQ.0) THEN
               WRITE (OUTF,FMT=*) J
            END IF
c
   40    CONTINUE
c
      END IF

      INAT = 0
c
      DO 50 K = 1,NT
c
         IF (IWS(KIT+K-1).EQ.0) INAT = INAT + 1

   50 CONTINUE
```

```
      IF (INAT.GT.0) THEN
C
         WRITE (OUTF,FMT=9070)
         WRITE (6,FMT=9080)
C
         DO 60 K = 1,NT

            IF (IWS(KIT+K-1).EQ.0) THEN
               WRITE (OUTF,FMT=*) K
            END IF

   60    CONTINUE
C
      END IF
C
C     STOP if not all the parameters have been estimated.
C
      IF (INAY.GT.0 .OR. INAT.GT.0) THEN
         WRITE (OUTF,FMT=9090)
         WRITE (6,FMT=9090)
         STOP
      END IF
C
C     Copy translation parameters into T.
C
      CALL DCOPY(3*NT,T,1,WS,1)
      CALL DCONSV(6*NT,ZERO,T,1)
C
      DO 70 K = 1,NT
C
         CALL DCOPY(3,WS(3*K-2),1,T(6*K-5),1)
C
   70 CONTINUE
C
C     Print the initial estimates.
C
C
C     Print Y, the coordinates of the registration points
C
      WRITE (OUTF,FMT=9170) (J,Y(J,1),Y(J,2),Y(J,3),J=1,NY)
C
C     Print T, the transformation parameters
C
      WRITE (OUTF,FMT=9190) (J,T(J),J=1,NNT)
C
C     Print fixed rotation matrices
C
      WRITE (OUTF,FMT=9240) (K,R0(K,1),R0(K,2),R0(K,3),K=1,NT*3)
C
C     Call LSPCS to determine the nonlinear least-squares
C     solution, using the results from PCSI as initial estimates
C
      WRITE (6,FMT=*) 'Please enter convergence tolerance:'
      READ (6,FMT=*) TOLR

      WRITE (OUTF,FMT=*)
      WRITE (OUTF,FMT=*) 'Position calibration solver: LSPCS'
C
C     Set default value for TOLR if input TOLR is inappropriate
C
      IF (TOLR.LE.ZERO) TOLR = 1.0D-9

      WRITE (OUTF,FMT=9150) TOLR
```

```
C
C
      CALL LSPCS(MX,NY,NT,X,LDX,IX,LDIX,D,LDD,DTYP,Y,LDY,T,R0,LDR0,E,
     .           LDE,R,LR,ENRM,SIGMA,TOLR,WS,LWS,INFORM)
C
C     Check the value of INFORM.
C
      WRITE (OUTF,FMT=9160) INFORM
      WRITE (6,FMT=9160) INFORM
C
C     ... and STOP if INFORM is not zero
C
      IF (INFORM.LT.0) THEN
         WRITE (OUTF,FMT=9090)
         WRITE (6,FMT=9090)
         STOP

      END IF
C
C     Otherwise, print the results
C
C     Print Y, the coordinates of the registration points
C
      WRITE (OUTF,FMT=9180) (J,Y(J,1),Y(J,2),Y(J,3),J=1,NY)
C
C     Print T, the transformation parameters
C
      WRITE (OUTF,FMT=9200) (J,T(J),J=1,NNT)
C
C     Print E, the weighted residual values.
C
      WRITE (OUTF,FMT=9220) ENRM
      WRITE (OUTF,FMT=9230) SIGMA
C
C     Output residual errors if required (terminal input = 1)
C
      WRITE (6,FMT=*) 'Output residuals: 0/1'

      READ (6,FMT=*) IREF

      IF (IREF.EQ.1) THEN

         WRITE (OUTF,FMT=9210) (I,E(I,1),E(I,2),E(I,3),I=1,MX)
C
      END IF
C
      IF (INFORM.GT.0) THEN
         WRITE (OUTF,FMT=9090)
         WRITE (6,FMT=9090)
         STOP
      END IF
C
C     Read MW, the number of points to be transformed
C
      READ (INF,FMT=*) MW
C
C  Check whether MW is feasible..
C
      IF (MW.GT.MAXMX) THEN
         WRITE (OUTF,FMT=*)
         WRITE (OUTF,FMT=*) 'Number of transformed points:          ',MW
```

```
          WRITE (OUTF,FMT=*) 'Maximum number of transformed points
     +      MAXMX
          WRITE (OUTF,FMT=*) '*** Maximum exceeded ***'
          WRITE (OUTF,FMT=9090)
          WRITE (6,FMT=9090)
          STOP
       END IF
C
C      Now read the additional points W and the index array IW,
C      overwriting (part of) X and IX.
C
       IF (MW.GT.0) THEN
C
          DO 80 I = 1,MW
             READ (INF,FMT=*  (W(I,J),J=1,3)
   80     CONTINUE
C
          DO 90 I = 1,MW
             READ (INF,FMT=*) IX(I,1)
   90     CONTINUE
C
C      Copy W into X and transform the points. Note: a copy of
C      W is required for calculating statistics later
C
          DO 100 K = 1,3
             CALL DCOPY(MW,W(1,K),1,X(1,K),1
  100     CONTINUE

          INV1 = .TRUE

          CALL ATPCS(MW,NT,X,LDX,IX(1,1),T,R0,LDR0,INV1

       END IF
C
C      Write out transformed points
C
       WRITE (OUTF,FMT=9250) MW

       DO 110 I = 1,MW
          WRITE (OUTF,FMT=9040) I,X(I,1),X(I,2),X(I,3)
  110  CONTINUE
C
C      Now calculate distances and associated statistics
C
C      The part of the upper-triangle R relating to transformations
C      (RT, say) starts at IRT.
C
       IRT = NY* (6+3*6*NT) + 1

C      Read NSY, the number of registration point pairs for which
C      the distances and their uncertainties are required.

       READ (INF,FMT=*) NSY
C
C      Now read the indices specifying the pairs for which the
C      distances and their uncertainties are required.
C
       WRITE (OUTF,FMT=9270) NSY
C
       IF (NSY.GT.0) THEN
C
          WRITE (OUTF,FMT=9260)
```

```
         DO 120 I = 1,NSY

            READ (INF,FMT=*  J1,J2
c
c     Go to next point if indices out of range..
c
            IF (J1.LT.1 .OR. J2.LT.: .OR. J1.GT.NY .OR.
                J2.GT.NY) GO TO 120

      ...otherwise

            CALL RPDST(NY,NT,Y,LDY,R,SIGMA,J1,J2,D12,UD12,WS,LWS,INFO)
c
            WRITE (OUTF,FMT=9030) J1,J2,D12,UD12

  120    CONTINUE
c
      END IF
c
c     Read NSYW, the number of registration-transformed point pairs
c     for which the distances and their uncertainties are required.
c
      READ (INF,FMT=*) NSYW
c
c     Now read the indices specifying the pairs for which the
c     distances and their uncertainties are required.
c
      WRITE (OUTF,FMT=9290) NSYW

      IF (NSYW.GT.0) THEN

         WRITE (OUTF,FMT=9260)

         INV2 = .TRUE.

         DO 130 I = 1,NSYW

            READ (INF,FMT=*) J1,J2

      Go to next point if the indices are out of range.

            IF (J1.LT.1 .OR. J2.LT.1 .OR. J1.GT.NY .OR.
     +          J2.GT.MW) GO TO 130

         otherwise

            CALL DCOPY(3,W(J2,1),LDW,W2,1)
            K = IX(J2,1)

            CALL RTPDST(NY,NT,Y,LDY,T,R0,LDR0,R,SIGMA,J1,W2,INV2,K,WT2,
                        D12,UD12,WS,LWS,INFO)
c
            WRITE (OUTF,FMT=9030) J1,J2,D12,UD12

  130    CONTINUE

      END IF
c
c     Read NSW, the number of registration point pairs for which
c     the distances and their uncertainties are required.
```

```
      READ (INF,FMT=*  NSW

      Now read the indices specifying the pairs for which the
c     distances and their uncertainties are required.
c
      WRITE (OUTF,FMT=9280) NSW

      IF (NSW.GT.0) THEN

         WRITE (OUTF,FMT=9260)

         INV1 = .TRUE
         INV2 = .TRUE

         DO 140 I = 1,NSW
c
            READ (INF,FMT=*  J1,J2
c
c     Go to next point if the indices are out of range...
c
            IF (J1.LT.1 .OR. J2.LT.1  OR. J1.GT.MW .OR.
                J2.GT.MW) GO TO 140

       ..otherwise...

            CALL DCOPY(3,W(J1,1),LDW,W1,1)
            CALL DCOPY(3,W(J2,1),LDW,W2,1)

            K1 = IX(J1,1)
            K2 = IX(J2,1)
c
            CALL TPDST(NT,T,R0,LDR0,R(IRT),SIGMA,W1,W2,INV1,INV2,K1,K2,
                       WT1,WT2,D12,UD12,WS,LWS,INFO)
c
            WRITE (OUTF,FMT=9030) J1,J2,D12,UD12

  140    CONTINUE

      END IF
c
c     Output covariance matrices if required (terminal input = 1)
c
      WRITE (6,FMT=*) 'Output covariances  0/1'

      READ (6,FMT=*  IREF
c
c     Calculate covariance matrices
c
      IF (SIGMA.GT.ZERO  AND. IREF.EQ.1) THEN
c
         WRITE (OUTF,FMT=9320)

         DO 160 J = 1,NY
c
            CALL PCSCMR(NY,NT,R,SIGMA,J,VT,6,WS,LWS,INFO)
c
c     Print statistics.
c
            WRITE (OUTF,FMT=9330) J
c
            DO 150 L = 1,3
```

```
            WRITE (OUTF,FMT=9000) (VT(L,I),I=1,3)
  150     CONTINUE
  160   CONTINUE

        DO 180 K = 1,NT

           CALL PCSCMT(NY,NT,R,SIGMA,K,VT,6,WS,LWS,INFO)
c
c     Print statistics.
c
           WRITE (OUTF,FMT=9340) K

           DO 170 L = 1,6
              WRITE (OUTF,FMT=9010) (VT(L,I),I=1,6)
  170      CONTINUE
c
  180   CONTINUE

      END IF

c     End of calculations.
c     ------------------------------------------------------------
      STOP
:     ------------------------------------------------------------
:     End of PCSDR.FOR.
:     ------------------------------------------------------------
 9000 FORMAT (1P,3D12.4)
 9010 FORMAT (1P,6D12.4)
 9020 FORMAT (1I5,2X,1P,3D22.14)
 9030 FORMAT (2I5,2X,2F15.5)
 9040 FORMAT (1I5,2X,3F15.5)
 9050 FORMAT (/,
     +' The following registration points have not been assigned:')
 9060 FORMAT (/,
     +' Not all the registration points have been assigned.')
 9070 FORMAT (/,
     +' The following transformations have not been assigned:')
 9080 FORMAT (/,
     +' Not all the transformations have been assigned.')
 9090 FORMAT (/,' *** Calculations terminated ***')
 9100 FORMAT (/,' Tolerance on distances:             ', (1F15.5))
 9110 FORMAT (/,' Distances checked in positions 0 to:', (I5))
 9120 FORMAT (/,' Number of data points:              ',I10)
 9130 FORMAT (/,' Number of registration points:      ',I10)
 9140 FORMAT (/,' Number of transformations:          ',I10)
 9150 FORMAT (/,' Convergence tolerance:              ',1P,1D12.4)
 9160 FORMAT (/,' LSPCS exit value of INFORM:         ',I2)
 9170 FORMAT (/,' Initial  estimates of registration points:      ',/,
     +        (1I5,2X,3F15.5))
 9180 FORMAT (/,' Solution estimates of registration points:      ',/,
     +        (1I5,2X,3F15.5))
 9190 FORMAT (/,' Initial  estimates of transformation parameters:',/,
     +        (1I5,2X,1P,1D22.14))
 9200 FORMAT (/,' Solution estimates of transformation parameters:',/,
     +        (1I5,2X,1P,1D22.14))
 9210 FORMAT (/,' Residual values for each data point: ',/,
     +        (1I5,2X,1P,3D22.14))
 9220 FORMAT (/,' Norm of residuals:',1P,1D22.14)
 9230 FORMAT (/,' SIGMA value:      ',1P,1D22.14)
 9240 FORMAT (/,' Fixed rotation matrices:',/, (1I3,2X,1P,3D22.14))
 9250 FORMAT (/,' Number of transformed points: ', (I5))
```

```
 9260 FORMAT (/,' Indices, distances and standard uncertainties: ')
 9270 FORMAT (/,' Number of distances between registration points: ',
     +       (I5))
 9280 FORMAT (/,' Number of distances between transformed points:  ',
     +       (I5))
 9290 FORMAT (/,' Number of distances between registration and',/,
     +       ' transformed points:                              ', (I5)
 9300 FORMAT (/,' Number of distances out of tolerance: ', (I5))
 9310 FORMAT (/,' Exceptional distances:',/, (3I4,2X,3F15.5))
 9320 FORMAT (/,' Covariance matrices:')
 9330 FORMAT (/,' Covariance matrix for registration point:',I5)
 9340 FORMAT (/,' Covariance matrix for transformation:    ',I5)
      END
c     ------------------------------------------------------------
c     End of DRPCS.FOR.
c     ------------------------------------------------------------
```

B LISTING OF SAMPLE DATA SETS

B.1 XTEST1.DAT

```
 3 7 16
  62.6851   26.4060  -36.0043
 116.4950  179.7064   57.7345
   7.5087   87.1688  -13.5562
  35.1603 -144.6175 -134.9344
-110.6373  -59.1077 -173.5349
 -32.7928  -90.5484 -187.3606
-172.1555  128.3071 -143.7501
 -47.0137  117.1560 -172.2567
 166.3421   61.6911  -65.2964
  39.4671 -105.2504   93.6175
  -9.6320  -12.0309  264.2849
 151.2604  -65.6112  -78.1436
 -84.4143 -207.9049 -105.4604
 -48.8440 -177.7650 -254.5033
 -59.3731 -449.9700   11.6095
  12.4952 -115.9752 -159.5402
    2     0
    1     0
    3     0
    4     0
    2     1
    3     1
    4     1
    5     1
    4     2
    3     2
    6     2
    5     2
    7     3
    4     3
    6     3
    5     3
8
  69.1160  -16.5923  104.8423
 -75.8618   30.0907   42.2724
  -9.6972  -32.2467  -84.4414
-140.6949  -36.8411  -31.1630
 103.0812  114.7895   39.7810
 -75.9874    4.1430  104.9786
  87.4127 -109.8050  -34.0796
  76.1127  156.6724   33.6297
    0
    0
    1
    1
    2
    2
    3
    3
21
```

```
1 2
1 3
1 4
1 5
1 6
1 7
2 3

5 7
6 7
56
1 1
1 2
1 3
1 4
1 5
1 6
1 7
1 8
2 1

6 8
7 1
7 2
7 3
7 4
7 5
7 6
7 7
7 8
28
1 2
1 3
1 4
1 5
1 6
1 7
1 8
2 3

. .
6 8
7 8
```

B.2 XTEST2.DAT

```
  3 5 14
 -111.3573  184.2293   -3.2437
 -131.4903  191.5368  -54.0810
  -30.8841  105.5415  -80.4687
  -97.7818  -42.1923   54.5705
 -174.6920  343.6451   61.6834
  196.9432  221.2251   57.9947
 -142.6090  175.9157  -16.3107
```

```
      7.5067  179.7070  124.5986
     35.1600   26.4062  -63.8980
     40.1820   -5.3978   61.5502
    -29.6288 -246.0306  212.5797
    223.9851  -86.3299   50.1809
     68.6456  -13.5799   43.1268
    -69.6492   87.1681   57.7349
           4        2
           1        1
           1        3
           2        1
           1        2
           3        3
           2        2
           3        0
           4        0
           2        3
           3        1
           5        3
           4        3
           5        0
10
     35.1589  102.8193  -55.0021
    113.3000   39.4600    3.9885
     14.9994   63.9406 -248.2843
     70.3144   87.4213  115.8655
     -5.2412  175.2402 -102.6279
    201.8496  -32.0051  115.3487
     92.4159  -13.7414  -78.6457
   -181.4115   61.5770   63.4809
      3.4973   97.7894   82.0410
   -180.7862 -111.5348  -17.6027
           2
           0
           2
           1
           2
           3
           3
           1
           0
           2
10
1 2
1 3
1 4
1 5
2 3
2 4
2 5
3 4
3 5
4 5
50
1 1
```

```
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
1 10
2 1

. .
4 10
5 1
5 2
5 3
5 4
5 5
5 6
5 7
5 8
5 9
5 10
45
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
1 10
2 3


8 10
9 10
```

C LISTING OF SAMPLE RESULTS

C. XTEST1.RES

```
DRPCS: position calibration software

Version 1.0/VMS Release date 1998-02-01
B P Butler, A B Forbes, P D Kenward, NPL

Input file is xtest1.dat

Input data statistics:

Number of transformations:                  3

Number of registration points:

Number of data points:                     16

Distance consistency check: CHKDIS

Tolerance on distances:                    0.00100

Distances checked in positions 0 to:     3

Number of distances out of tolerance:     8

Exceptional distances:
  0   2   3         85.09095       85.08486          0.00609
  0   2   4        199.48388      199.48917         -0.00530
  0   3   4        263.10106      263.09992          0.00114
  1   3   4        263.09992      263.09762          0.00230
  1   3   5        208.73782      208.73640          0.00143
  1   4   5        128.83107      128.83472         -0.00365
  2   4   6        380.82233      380.81813          0.00421
  2   5   6        382.11842      382.11229          0.00614

Initial estimation module: PCSI

Initial  estimates of registration points:
   1        116.49500      179.70640       57.73450
   2         62.68510       26.40600      -36.00430
   3          7.50870       87.16880      -13.55620
   4         35.16030     -144.61750     -134.93440
   5        -69.65056      -70.12115     -127.03984
   6        169.62614      124.60427       98.44408
   7          5.91177      -63.89719       -4.48771

Initial  estimates of transformation parameters
   1   -7.98841768381990D+01
   2   -7.65198016931070D+01
   3    8.61795682858721D+01
   4    0.00000000000000D+00
```

```
   5    0.00000000000000D+00
   6    0.00000000000000D+00
   7    7.56166687642640D+01
   8    4.00501282043365D+01
   9   -1.34143204373916D+02
  10    0.00000000000000D+00
  11    0.00000000000000D+00
  12    0.00000000000000D+00
  13   -2.37740329549694D+02
  14   -2.73821277837914D+01
  15   -3.22802187545757D+01
  16    0.00000000000000D+00
  17    0.00000000000000D+00
  18    0.00000000000000D+00

Fixed rotation matrices:
 1  -8.99015594116237D-01  3.94937034452139D-01  1.89197516775401D-01
 2  -4.25522006215086D-01 -6.85771021291741D-01 -5.90465179822799D-01
 3  -1.03450392761538D-01 -6.11345111352526D-01  7.84573241363004D-01
 4  -4.03350978797937D-01 -8.14193876156825D-01  4.17607854250218D-01
 5   8.32899535425300D-01 -5.15667342875688D-01 -2.00911809956126D-01
 6   3.78927897879089D-01  2.66787412597095D-01  8.86136628680184D-01
 7  -1.63437373451424D-01  6.70139198312727D-01 -7.24017734481743D-01
 8  -8.94486045746520D-01 -4.10229058788895D-01 -1.77783107436934D-01
 9  -4.16152542854246D-01  6.18567356243403D-01  6.66476921480401D-01

Position calibration solver: LSPCS

Convergence tolerance:                    1.0000D-12

LSPCS exit value of INFORM          0

Solution estimates of registration points:
   1        116.49500       179.70640         57.73450
   2         62.68465        26.40785        -36.00338
   3          7.50944        87.16719        -13.55693
   4         35.16001      -144.61774       -134.93459
   5        -69.65269       -70.11895       -127.03937
   6        169.62654       124.60055         98.44210
   7          5.91099       -63.89757         -4.48822

Solution estimates of transformation parameters:
   1   -7.98845960311130D+01
   2   -7.65190492981710D+01
   3    8.61798120613477D+01
   4   -1.53271905937693D-06
   5    8.17835714076067D-07
   6   -5.74084102977151D-06
   7    7.56163813052544D+01
   8    4.00491675916800D+01
   9   -1.34143848709865D+02
  10   -8.42765747079475D-06
  11    2.31992449593462D-06
  12   -6.31379917128618D-06
  13   -2.37740881024714D+02
  14   -2.73800976861490D+01
```

```
  15    -3.22795225412436D+01
  16     9.29989631941325D-06
  17    -3.14301579054246D-06
  18    -5.01764907462005D-06

Norm of residuals:   5.91481937945718D-03

SIGMA value:         1.97160645981906D-03

Residual values for each data point:
   1     4.50281812575781D-04 -1.84508370772818D-03 -9.21200068020767D-04
   2     0.00000000000000D+00  0.00000000000000D+00  0.00000000000000D+00
   3    -7.43209709470172D-04  1.60827934418251D-03  7.34394552317497D-04
   4     2.92927896893502D-04  2.36804363538567D-04  1.86805515710375D-04
   5     1.30780060426616D-03 -1.61762862467185D-03 -3.58653560937228D-04
   6    -6.82790057098259D-04 -2.07262454466672D-04  7.89808417778204D-05
   7     1.00011291476676D-03  1.41854131121022D-03 -1.05552033545564D-04
   8    -1.62512346188493D-03  4.06349767970937D-04  3.85224752704971D-04
   9    -2.24376317675024D-04  3.42182906933886D-05  2.50647999806120D-04
  10     8.16672072026847D-04  9.38423334162053D-04 -1.06311135219528D-03
  11    -8.78907692689168D-04  2.25903487223178D-05  1.77884151844410D-03
  12     2.86611938321357D-04 -9.95231973618615D-04 -9.66378166083359D-04
  13     0.00000000000000D+00  0.00000000000000D+00  0.00000000000000D+00
  14    -4.37256901470562D-04 -9.74450123379711D-04 -1.00475887791163D-03
  15     2.53642506521601D-04  1.63450718514468D-03 -1.09604650669937D-03
  16     1.83614394938303D-04 -6.60057061651287D-04  1.19652239445145D-03

Number of transformed points:     8
   1          69.11600         -16.59230         -104.84230
   2         -75.86180          30.09070           42.27240
   3         -48.70923          -6.61223           37.13526
   4          65.50271         -87.76949           56.86406
   5         144.71968         -92.45986          -78.90816
   6         149.49708         127.78732          -73.68381
   7        -139.62554          55.16254          -98.75981
   8        -404.31710         -19.84154          -92.82599

Number of distances between registration points:    21

Indices, distances and standard uncertainties
   1    2        187.57076         0.00254
   1    3        159.76178         0.00260
   1    4        385.90530         0.00249
   1    5        362.22238         0.00281
   1    6         86.69904         0.00755
   1    7        274.66959         0.00468
   2    3         85.08728         0.00190
   2    4        199.48619         0.00185
   2    5        187.39836         0.00207
   2    6        197.87360         0.00253
   2    7        111.22734         0.00402
   3    4        263.09949         0.00155
   3    5        208.73713         0.00178
   3    6        200.56668         0.00241
   3    7        151.34517         0.00326
```

4	5	128.83364	0.00159
4	6	380.82071	0.00188
4	7	156.16500	0.00254
5	6	382.11533	0.00187
5	7	144.10885	0.00256
6	7	270.05364	0.00256

Number of distances between registration and transformed points: 56

Indices, distances and standard uncertainties:

1	1	259.24731	0.00197
1	2	244.18246	0.00197
1	3	249.86276	0.00400
1	4	272.29455	0.00453
1	5	305.84690	0.00314
1	6	145.10514	0.00725
1	7	324.96057	0.00396
1	8	577.69636	0.00449
2	1	81.41973	0.00171
2	2	159.17217	0.00175
2	3	137.28868	0.00181
2	4	147.20315	0.00192
2	5	150.66552	0.00231
2	6	138.68673	0.00303
2	7	213.76299	0.00268
2	8	472.71392	0.00271
3	1	151.30901	0.00154
3	2	115.43572	0.00190
3	3	120.51895	0.00184
3	4	197.29465	0.00175
3	5	235.29410	0.00166
3	6	159.45446	0.00192
3	7	173.01021	0.00254
3	8	432.82277	0.00282
4	1	135.82735	0.00156
4	2	272.49073	0.00160
4	3	235.98220	0.00150
4	4	202.33419	0.00171
4	5	133.65151	0.00161
4	6	301.71035	0.00152
4	7	267.90066	0.00156
4	8	458.78356	0.00164
5	1	150.38139	0.00307
5	2	196.84258	0.00469
5	3	177.27109	0.00170
5	4	228.90827	0.00165
5	5	220.84213	0.00149
5	6	300.06721	0.00152
5	7	146.25794	0.00164
5	8	340.14502	0.00160
6	1	267.13730	0.00361
6	2	268.98266	0.00686
6	3	262.00349	0.00376
6	4	240.14899	0.00420
6	5	281.40485	0.00167

6	6	173.32825	0.00182
6	7	373.29222	0.00170
6	8	621.97934	0.00173
7	1	127.68557	0.00354
7	2	133.06818	0.00472
7	3	89.42870	0.00336
7	4	88.79833	0.00314
7	5	160.06869	0.00297
7	6	249.29519	0.00257
7	7	210.34098	0.00197
7	8	421.93789	0.00197

```
Number of distances between transformed points:     28

Indices, distances and standard uncertainties
```

1	2	211.75599	0.00000
1	3	184.77016	0.00159
1	4	176.71501	0.00285
1	5	110.20156	0.00421
1	6	168.15898	0.00200
1	7	220.81391	0.00374
1	8	473.59671	0.00342
2	3	45.94298	0.00544
2	4	184.62899	0.00701
2	5	279.92777	0.00382
2	6	271.61940	0.00276
2	7	156.79444	0.00992
2	8	358.64702	0.00846
3	4	141.49238	0.00000
3	5	241.35168	0.00287
3	6	263.87468	0.00276
3	7	174.78377	0.00393
3	8	378.84284	0.00394
4	5	157.26228	0.00492
4	6	265.63610	0.00437
4	7	294.49271	0.00290
4	8	497.74689	0.00418
5	6	220.36093	0.00000
5	7	320.99637	0.00171
5	8	553.99324	0.00171
6	7	299.15723	0.00317
6	8	573.47262	0.00315
7	8	275.17711	0.00000

```
Covariance matrices

Covariance matrix for registration point:      1
 3.8872D-06  0.0000D+00  0.0000D+00
 0.0000D+00  3.8872D-06  0.0000D+00
 0.0000D+00  0.0000D+00  3.8872D-06

Covariance matrix for registration point       2
 2.6962D-06  2.7916D-08 -2.1760D-07
 0.0000D+00  2.9044D-06 -5.3213D-07
 0.0000D+00  0.0000D+00  3.5536D-06
```

```
Covariance matrix for registration point:     3
 3.1857D-06  2.2013D-07 -1.6763D-08
 0.0000D+00  2.6113D-06 -6.5490D-07
 0.0000D+00  0.0000D+00  3.5257D-06

Covariance matrix for registration point:     4
 3.7624D-06  3.8724D-08 -3.2254D-09
 0.0000D+00  2.5902D-06 -7.0191D-07
 0.0000D+00  0.0000D+00  3.5027D-06

Covariance matrix for registration point:     5
 9.1488D-06  7.8816D-06 -1.6230D-05
 0.0000D+00  2.0629D-05 -2.7571D-05
 0.0000D+00  0.0000D+00  5.4248D-05

Covariance matrix for registration point:     6
 6.1510D-05  4.0306D-05 -8.3110D-05
 0.0000D+00  6.1922D-05 -8.5376D-05
 0.0000D+00  0.0000D+00  1.5065D-04

Covariance matrix for registration point:     7
 4.9327D-05 -1.7489D-06  1.0430D-05
 0.0000D+00  1.1865D-05 -1.2502D-06
 0.0000D+00  0.0000D+00  1.1118D-05

Covariance matrix for transformation:        1
 1.3923D-04 -3.9811D-05  8.8663D-05  4.0707D-07 -7.3100D-07 -1.4153D-07
 0.0000D+00  2.0688D-05 -2.2954D-05 -9.0646D-08  2.3150D-07  2.5762D-08
 0.0000D+00  0.0000D+00  6.1364D-05  2.7661D-07 -4.6572D-07 -9.8121D-08
 0.0000D+00  0.0000D+00  0.0000D+00  1.3806D-09 -2.0970D-09 -4.0162D-10
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  3.9586D-09  7.1195D-10
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  3.1918D-10

Covariance matrix for transformation:        2
 5.4715D-05 -1.2196D-05  3.0005D-05  2.9175D-07  3.6806D-07 -3.7952D-07
 0.0000D+00  7.3143D-06 -6.2123D-06 -6.6352D-08 -7.3169D-08  9.8617D-08
 0.0000D+00  0.0000D+00  2.2620D-05  1.7350D-07  2.3051D-07 -2.1286D-07
 0.0000D+00  0.0000D+00  0.0000D+00  1.8604D-09  2.0703D-09 -2.1030D-09
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  2.7976D-09 -2.6119D-09
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  2.8623D-09

Covariance matrix for transformation:        3
 1.3640D-05 -2.5570D-05  5.7702D-05  7.9138D-08 -8.6290D-08  2.1532D-07
 0.0000D+00  1.2482D-04 -1.8894D-04 -1.4385D-07  3.6242D-07 -7.4375D-07
 0.0000D+00  0.0000D+00  3.7879D-04  4.3766D-07 -6.2425D-07  1.4048D-06
 0.0000D+00  0.0000D+00  0.0000D+00  7.4314D-10 -5.5759D-10  1.5600D-09
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  1.3619D-09 -2.2138D-09
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  5.4866D-09
```

C.2 XTEST2.RES

```
DRPCS: position calibration software

Version 1.0/VMS Release date 1998-02-01
B P Butler, A B Forbes, P D Kenward, NPL

Input file is xtest2.dat

Input data statistics:

Number of transformations                          3

Number of registration points:                     5

Number of data points:                            14

Distance consistency check: CHKDIS

Tolerance on distances:                           0.00100

Distances checked in positions 0 to:     3

Number of distances out of tolerance:      2

Exceptional distances:
  2   1   2          187.73798       187.74589        -0.00791
  2   1   4          183.41268       183.40923         0.00345

Initial estimation module: PCSI

Initial  estimates of registration points:
   1        116.49530      169.61070      -144.62203
   2         62.68128        5.89899       -70.11669
   3          7.50670      179.70700       124.59860
   4         35.16000       26.40620       -63.89800
   5        -69.64920       87.16810        57.73490

Initial  estimates of transformation parameters
   1    -3.60055493200481D+01
   2    -1.35619075998164D+01
   3    -1.34933293629729D+02
   4     0.00000000000000D+00
   5     0.00000000000000D+00
   6     0.00000000000000D+00
   7    -7.98899763798992D+01
   8    -7.65156543573978D+01
   9     8.61807331289695D+01
  10     0.00000000000000D+00
  11     0.00000000000000D+00
  12     0.00000000000000D+00
  13     7.56201293286512D+01
  14     4.00446229849847D+01
  15    -1.34138703175528D+02
```

```
   16     0.00000000000000D+00
   17     0.00000000000000D+00
   18     0.00000000000000D+00

 Fixed rotation matrices:
  1    -9.93755526531667D-01  1.08014149299534D-01 -2.79802973334413D-02
  2    -9.99846722638235D-02 -9.73347214504956D-01 -2.06393472105442D-01
  3    -4.95279597809719D-02 -2.02307052686114D-01  9.78068932966073D-01
  4    -9.77490565484745D-01  1.77346760686061D-01  1.14281760847802D-01
  5    -3.72962249351267D-02 -6.78394315316606D-01  7.33750737343209D-01
  6     2.07656413322347D-01  7.12972144911899D-01  6.69738407578529D-01
  7    -8.67128475044304D-01  7.50344633140655D-02  4.92400281359089D-01
  8    -3.79502722427240D-01 -7.39804520294022D-01 -5.55578037203456D-01
  9     3.22592454092860D-01 -6.68624783468054D-01  6.69981348616993D-01

 Position calibration solver: LSPCS

 Convergence tolerance                       1.0000D-12

 LSPCS exit value of INFORM:            0

 Solution estimates of registration points:
    1          116.49536        169.61417      -144.61716
    2           62.68394          5.90590       -70.11645
    3            7.50801        179.70720       124.59817
    4           35.16072         26.40679       -63.89768
    5          -69.65123         87.16731        57.73501

 Solution estimates of transformation parameters:
    1     -3.60028998011615D+01
    2     -1.35577575909651D+01
    3     -1.34933742345100D+02
    4     -1.32342491503504D-05
    5      4.50337393217774D-07
    6      7.16146055893489D-06
    7     -7.98944563883421D+01
    8     -7.65172718104328D+01
    9      8.61735486145438D+01
   10     -2.20064997232970D-06
   11      7.30454556872189D-05
   12     -1.94401085680462D-05
   13      7.56219334379847D+01
   14      4.00485456798413D+01
   15     -1.34138047947431D+02
   16     -7.28363925110969D-06
   17     -4.44244614573574D-06
   18      1.45038189188719D-05

 Norm of residuals:   1.00161763892279D-02

 SIGMA value:         3.33872546307598D-03

 Residual values for each data point:
    1     1.38596049178830D-03  1.95950823615476D-03 -6.27881153116849D-04
    2     9.33736970750942D-04  3.18658667310956D-04  2.54416117687839D-03
    3    -1.66656640928053D-03  3.10425053456242D-04 -1.93355029418285D-03
```

4	3.73687339873641D-04	-2.34052378503691D-03	7.17281970949557D-04
5	-4.23891977135327D-04	5.17762649110409D-05	-1.45610635819082D-04
6	1.62765752770611D-03	-2.98061045924669D-03	1.01955476286975D-03
7	-9.62068514610337D-04	-2.01128450095212D-03	7.73491788926606D-04
8	-1.30728590139029D-03	-1.96131055417936D-04	4.30123555418049D-04
9	-7.19297976147004D-04	-5.92850813291079D-04	-3.15871540855994D-04
10	5.23415603183253D-04	3.33897436475095D-03	2.22516705394327D-03
11	-1.30742431063169D-03	2.02186511762648D-03	-3.26144314789190D-03
12	1.75438395962146D-03	1.28928545805707D-03	-4.96795195132904D-05
13	-2.23889068119831D-03	-1.95807441710194D-03	-1.26149200309555D-03
14	2.02658387753729D-03	7.88981868694805D-04	-1.14252014562055D-04

Number of transformed points: 10

1	-129.51684	-179.25075	128.79560
2	113.30000	39.46000	3.98850
3	-148.48251	-294.25721	-31.48474
4	-120.35717	-114.49630	-41.62147
5	-102.61437	-269.50021	145.42279
6	-50.05198	1.74319	60.31420
7	-24.67074	109.73247	-133.68898
8	134.97544	-105.93065	-80.47880
9	3.49730	97.78940	82.04100
10	97.33290	-45.45816	-28.10380

Number of distances between registration points 10

Indices, distances and standard uncertainties:

1	2	187.74030	0.00273
1	3	290.61488	0.00321
1	4	183.41022	0.00316
1	5	287.04432	0.00375
2	3	266.76779	0.00311
2	4	34.87818	0.00317
2	5	201.15171	0.00364
3	4	244.53297	0.00313
3	5	137.79644	0.00328
4	5	171.67381	0.00324

Number of distances between registration and transformed points: 50

Indices, distances and standard uncertainties

1	1	506.93517	0.00259
1	2	197.57014	0.00757
1	3	546.06673	0.00260
1	4	383.96092	0.00276
1	5	570.04709	0.00256
1	6	312.91460	0.00248
1	7	153.73064	0.00296
1	8	283.51400	0.00282
1	9	263.25136	0.00357
1	10	245.35417	0.00261
2	1	332.85151	0.00266
2	2	95.80922	0.00512
2	3	369.02788	0.00269

2	4	220.93597	0.00298
2	5	386.81937	0.00260
2	6	172.44962	0.00268
2	7	149.84079	0.00282
2	8	133.56966	0.00287
2	9	187.34333	0.00360
2	10	74.85907	0.00282
3	1	384.24502	0.00939
3	2	213.09133	0.00284
3	3	522.81658	0.01100
3	4	361.29512	0.00302
3	5	462.97721	0.00730
3	6	197.77959	0.00270
3	7	269.52585	0.00255
3	8	374.02338	0.00279
3	9	92.39980	0.00274
3	10	286.50627	0.01208
4	1	326.41143	0.00272
4	2	104.32961	0.00318
4	3	370.94587	0.00280
4	4	211.03489	0.00370
4	5	387.76023	0.00273
4	6	152.63712	0.00260
4	7	124.07174	0.00296
4	8	166.58665	0.00405
4	9	165.51782	0.00269
4	10	101.54385	0.00291
5	1	282.15613	0.00691
5	2	196.55998	0.00277
5	3	399.57378	0.01248
5	4	230.45827	0.00465
5	5	368.76475	0.00743
5	6	87.68160	0.00315
5	7	197.92821	0.00299
5	8	313.46763	0.00466
5	9	77.80949	0.00284
5	10	229.87283	0.00993

Number of distances between transformed points 45

Indices, distances and standard uncertainties:

1	2	349.81598	0.00939
1	3	198.18165	0.00000
1	4	182.53497	0.01605
1	5	95.63039	0.00000
1	6	209.19649	0.00492
1	7	404.23033	0.00641
1	8	345.14891	0.00755
1	9	310.85366	0.00784
1	10	306.55943	0.00000
2	3	425.62379	0.00681
2	4	283.51099	0.00647
2	5	402.59047	0.00816
2	6	176.85874	0.00473
2	7	207.19371	0.00506
2	8	169.53758	0.00484

```
2    9      146.80308       0.00000
2   10       92.17352       0.01160
3    4      182.23001       0.01064
3    5      184.42633       0.00000
3    6      325.16435       0.01060
3    7      434.72144       0.00616
3    8      343.82513       0.01102
3    9      435.53014       0.00940
3   10      349.76795       0.00000
4    5      243.57048       0.01123
4    6      169.83910       0.00434
4    7      260.59712       0.00499
4    8      258.41440       0.00000
4    9      275.13188       0.00399
4   10      228.77491       0.00601
5    6      289.10078       0.00545
5    7      477.27981       0.00494
5    8      366.38156       0.00740
5    9      387.52884       0.00620
5   10      346.82161       0.00000
6    7      223.47958       0.00000
6    8      256.22543       0.00466
6    9      112.09127       0.00320
6   10      178.23584       0.00941
7    8      273.54854       0.00423
7    9      217.88873       0.00267
7   10      223.86882       0.00314
8    9      291.89218       0.00374
8   10       88.41393       0.01205
9   10      203.60958       0.01225

Covariance matrices

Covariance matrix for registration point
 1.3104D-04 -4.3132D-05  6.0906D-05
 0.0000D+00  5.1491D-05 -1.0877D-05
 0.0000D+00  0.0000D+00  4.6896D-05

Covariance matrix f      gistration point:
 2.3873D-05  3.6622      6.4140D-06
 0.0000D+00  2.2741     -8.5165D-06
 0.0000D+00  0.000C      2.4744D-05

Covariance matrix f      gistration point:     3
 9.7226D-06 -5.314C      1.0669D-07
 0.0000D+00  9.1955     -1.9579D-06
 0.0000D+00  0.000C      8.8629D-06

Covariance matrix for registration point:     4
 9.9035D-06  2.8902D-07  9.5105D-07
 0.0000D+00  9.7527D-06 -1.7333D-06
 0.0000D+00  0.0000D+00  8.6967D-06

Covariance matrix for registration point:     5
 7.7075D-06 -7.6119D-07  8.5235D-07
```

```
 0.0000D+00  9.5923D-06 -1.3910D-06
 0.0000D+00  0.0000D+00  9.1361D-06

Covariance matrix for transformation:      1
 5.1757D-05 -1.1767D-05 -4.9789D-07  7.5686D-09  1.2613D-07  1.3160D-07
 0.0000D+00  5.6288D-05 -3.0465D-05 -1.8622D-07  1.7773D-07 -2.4762D-07
 0.0000D+00  0.0000D+00  3.5946D-05  1.4123D-07 -1.5593D-07  1.4196D-07
 0.0000D+00  0.0000D+00  0.0000D+00  9.1334D-10 -5.4883D-10  6.3189D-10
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  3.0094D-09 -2.0169D-09
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  2.7037D-09

Covariance matrix for transformation:      2
 4.2431D-04 -6.3141D-05  2.7674D-04 -4.8525D-07 -2.5019D-06  2.0056D-06
 0.0000D+00  6.6349D-05  3.3660D-06  9.6514D-08 -7.0622D-08 -3.5956D-07
 0.0000D+00  0.0000D+00  2.4936D-04 -2.8257D-07 -2.1714D-06  1.2867D-06
 0.0000D+00  0.0000D+00  0.0000D+00  3.8549D-09  7.6671D-09 -2.2867D-09
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  2.7862D-08 -1.1562D-08
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  9.9355D-09

Covariance matrix for transformation:      3
 3.7982D-05 -5.8445D-06  1.7160D-05 -9.4789D-08  2.1902D-07  7.4994D-08
 0.0000D+00  3.1975D-05 -1.1297D-05 -5.5733D-09 -1.6032D-07  6.3584D-08
 0.0000D+00  0.0000D+00  1.9007D-05 -4.2033D-08  1.5898D-07  1.5328D-08
 0.0000D+00  0.0000D+00  0.0000D+00  1.7035D-09 -1.4572D-09  1.3259D-10
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  2.7297D-09 -1.4051D-10
 0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  0.0000D+00  5.7347D-10
```

D MODULAR STRUCTURE OF POSITION CALIBRATION SOFTWARE

The following diagram lists the call tree for the main modules as determined by the NAGWare f77 Tool NAG_FCALLS (NAG 1992). See also section 4.11.

```
 1  DRPCS
 2      ATPCS
 3          DCOPY [external]
 4          ROTTR3
 5              DGEMV [external
 6              ROT3
 7                  DCOPY [external
 8                  DROT [external]
 9      CHKDIS
10      DCONSV
11      DCOPY [external]
12      LSPCS
13          DNRM2 [external]
14          DCOPY [external]
15          DMINV
16          GNPCS
17              DDOT [external]
18              DNRM2 [external]
19              DAXPY [external]
20              DCONSV
21              DCOPY [external]
22              DMAXV
23              DMINV
24              DRBMV
25                  DDOT [external]
26                  DCOPY [external]
27                  DTPMV [external]
28              DRBSL
29                  DDOT [external]
30                  DCOPY [external]
31                  DTPCD
32                  DTPSV [external]
33              DSCAL [external]
34              FPCS
35                  DNRM2 [external]
36                  DCOPY [external]
37                  FGPCS
38                      DCONSV
39                      DCOPY [external]
40                      DSCAL [external]
41                      ROTTR3 [see line 4]
42              GNCC
43              PARALS
44              USPCS
45                  DCONSV
46                  DCOPY [external]
47                  DTPUR
48                      DROT [external]
49                      DROTG [external]
```

```
50                  FGPCS [see line 37]
51                  DNRM2 [external]
52      PCSCMR
53          DRBCM
54              DCONSV
55              DRBSL [see line 28]
56              DSYRK [external]
57      PCSCMT
58          DTPCM
59              DCONSV
60              DSYRK [external]
61              DTPSV [external]
62      PCSI
63          LSPTM
64              DMEANM
65                  DMEANV
66              DGEDI [external]
67              DGEFA [external]
68              DGEMM [external]
69              DGEMV [external]
70              DSVDC [external]
71              XTWMC
72          ROTTR3 [see line 4]
73      RPDST
74          DNRM2 [external]
75          DAXPY [external]
76          DCONSV
77          DCOPY [external]
78          DNRM2V
79              DNRM2 [external]
80          DRBSL [see line 28]
81          DSCAL [external]
82      RTPDST
83          DDOT [external]
84          DNRM2 [external]
85          DCONSV
86          DCOPY [external]
87          DNRM2V [see line 78]
88          DRBSL [see line 28]
89          ROTTR3 [see line 4]
90      TPDST
91          DDOT [external]
92          DNRM2 [external]
93          DCONSV
94          DCOPY [external]
95          DNRM2V [see line 78]
96          DTPSV [external]
97          ROTTR3 [see line 4]
```

Called-by Listing

```
DRPCS is called by: nothing
CHKDIS is called by: DRPCS
PCSI is called by: DRPCS
LSPTM is called by: PCSI
XTWMC is called by: LSPTM
```

```
LSPCS is called by: DRPCS
GNPCS is called by: LSPCS
USPCS is called by: GNPCS
FGPCS is called by: USPCS, FPCS
FPCS is called by: GNPCS
ATPCS is called by: DRPCS
ROT3 is called by: ROTTR3
ROTTR3 is called by: PCSI, FGPCS, ATPCS, RTPDST, TPDST
PCSCMR is called by: DRPCS
PCSCMT is called by: DRPCS
RPDST is called by: DRPCS
RTPDST is called by: DRPCS
TPDST is called by: DRPCS
DCONSV is called by: DRPCS, GNPCS, USPCS, FGPCS, RPDST
            RTPDST, TPDST, DTPCM, DRBCM
DMAXV is called by: GNPCS
DMINV is called by: LSPCS, GNPCS
DMEANV is called by: DMEANM
DMEANM is called by: LSPTM
DNRM2V is called by: RPDST, RTPDST, TPDST
DTPUR is called by: USPCS
DTPCM is called by: PCSCMT
DTPCD is called by: DRBSL
DRBMV is called by: GNPCS
DRBSL is called by: GNPCS, RPDST, RTPDST, DRBCM
DRBCM is called by: PCSCMR
GNCC is called by: GNPCS
PARALS is called by: GNPCS
```

Note that all the modules denoted as “external” are BLAS modules (see section 4.11).