Computational Study of 3D Affine Coordinate Transformation

Part I. 3-point Problem

Bela Palancz[1], Robert H. Lewis[2], Piroska Zaletnyik[3] and Joseph Awange[4]

[1] Department of Photogrammetry and Geoinformatics
Budapest University of Technology and Economy, H – 1521, Hungary
e – mail : palancz@epito.bme.hu

[2] Department of Mathematics
Fordham University, Bronx, NY 10458, USA
e – mail : rlewis@fordham.edu

[3] Department of Geodesy and Surveying and
Research Group of Physical Geodesy and Geodynamics of the Hungarian Academy of Sciences
Budapest University of Technology and Economy, H – 1521, Hungary
e – mail : zaletnyikp@hotmail.com

[4] Western Australian Centre for Geodesy
Department of Spatial Sciences,
Division of Resource and Environmental
Curtin University of Technology, Australia
e – mail : J.awange@curtin.edu.au

Abstract

In case of considerable nonlinearity e.g. in geodesy, photogrammetry, robotics, it is difficult to find proper initial values to solve the parameter estimation problem of 3D affine transformation with 9 parameters via linearization and/or iteration. In this paper we develop a symbolic - numeric method to achieve the solution without initial guess. Our method employs explicit analytical expressions developed by the computer algebra technique *Dixon resultant* as well as by *reduced Grobner* basis for solving 3 points problem. Criteria for the proper selection of the 3 points from the *N* ones, is also given. Numerical illustration is presented with real world geodetic coordinates representing Hungarian Datum.
For systems of algebraic equations, `NSolve` computes a numerical Gröbner basis using an efficient monomial ordering, then uses eigensystem methods to extract numerical roots.

***Keywords**:* 9-parameter 3D affine transformation, solution of polynomial system, symbolic solution, Dixon resultant, early discovery factors, reduced Groebner basis, homotopy, numerical Gröbner basis , global minimization, genetic algorithm.

1. Introduction

Three-dimensional coordinate transformations play a central role in contemporary Euclidean point positioning. In precise positioning with global positioning system (GPS), coordinates given in the World Geodetic System 1984 (WGS84) often have to be transformed into local geodetic coordinate system. The transformation between the traditional terrestrial coordinate system and the satellite observations derived network is a difficult task due to the heterogeneity of the data.

Due to the distorsions between the traditional terrestrial and the GPS derived networks, the 7-parameter similarity transformations in some cases may not offer satisfactory precision. For example transforming GPS local coordinates to the local Hungarian system with global similarity transformation gives 0.5 meter maximal residuals, see Papp and Szucs (2005). To reduce the remaining residuals after the transformation, other transformation models with more parameters can be used. The 9-parameter affine transformation is not only a logical extention but even a generalization of the 7-parameter similarity transformation model. This transformation is the modification of the Helmert C7(3) transformation, where 3 different scales are used in the corresponding coordinate axes instead of one scale factor, while in case of the 3 scale parameters are equal, we get back the similarity transformation model. The estimation of the model parameters was achieved by Spath (2004) using numerical minimization technique of the residium vector, by Papp and Szucs (2005) using linearized least squares method. Watson (2006) pointed out that the Gauss-Newton method or its variants can be easily implemented for the nine parameter problem using separation of variables and iteration with respect to the rotation parameters alone, while other parameters can be calculated via simple linear least square solution. The method he suggested is analogous to other methods for separated least squares problems, which goes back at least to Golub and Pereyra (1973). The 9-parameter affine transformation is also included in some coordinate-transformation software developed by the request of GPS users (see e.g. Mathes 2002 and Frohlich and Broker 2003). Here we should mention transformation models with more then 9 parameters. Wolfrum (1992) even added one more parameter to the previous nine, one (horizontal) direction of maximal scale distortion. Grafarend and Kampmann (1996) applied ten parameter conformal group for geodetic datum transformation employing maximum likelihood estimations for numerical estimation of the parameter values. There are other models with even more parameters, like polynomial transformations (Volgyesi et al 1996, Cai and Grafarend 2004) and models using artificial neural networks (Barsi 2001, Zaletnyik 2004).

In this paper we solve the 3D affine transformation problem in symbolic form via Dixon resultant employing enhanced Dixon KSY and the Dixon EDF algorithms. We also investigate global numerical techniques, like global minimization with genetic algorithm, global polynomial solver method employing numerical Gröbner basis using an efficient monomial ordering and eigensystem methods to extract numerical roots as well as linear homotopy continuation method. In case of 3 point problem, symbolic solution represented by explicit expressions for the parameters, proved to be very fast and robust, while in addition, it is infinite precision, comparing with the global numerical methods.

Our suggested method employs explicit, analytical expressions developed by computer algebra technique via Dixon resultant for solving 3 point problem. Then the result of the 3 point problem can be used as a good initial guess for a Newton -Raphson- Krylov numerical method to solve the N point problem in a form of determined system of 9 polynomials developed by symbolic computation.

Numerical illustration is presented with real world geodetic coordinates representing Hungarian Datum. The computations were carried out with hp workstation xw 4100 with XP operation system, 3 GHz P4 Intel processor and 1 GB RAM.

2. Definition of the 3 points problem

The $C_9(3,3)$ 3D affine transformation is one possible generalization of the $C_7(3,3)$ Helmert transformation, using three different scale $(s_1,\ s_2,\ s_3)$ parameters instead of a single one. In this case the scale factors can be modeled by a diagonal matrix (W).

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = W\,R \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix}$$

where $W=\begin{pmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{pmatrix}$ - scale matrix, X_0, Y_0, Z_0 - 3 translation parameters and R the rotation matrix

The rotation matrix in general is given by using 3 axial rotation (α,β,γ - Cardan angles).

$$R = R_1(\alpha)\, R_2(\beta)\, R_3(\gamma) \qquad \text{with} \qquad R_1(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{pmatrix},$$

$$R_2(\beta) = \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix}, \; R_3(\gamma) = \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

leading to

$$\text{R}\,(\alpha,\ \beta,\ \gamma) = \begin{pmatrix} \cos\beta\cos\gamma & \cos\beta\sin\gamma & -\sin\beta \\ \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\cos\beta \\ \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\cos\beta \end{pmatrix}$$

In the traditional 7 or 9 parameter transformation solution (Papp E., Szucs L. (2005)) the single three by - three rotation matrix is simplified from three separate rotation matrices by assuming that each axial rotation is differentially small (typically less than five arc seconds for most geodetic networks), thus permitting binomial series expansions of the sine and cosine terms for radian measure.

The rotation matrix can be expressed with the skew-symmetric matrix (S) also (see Awange and Grafarend (2003)), and this facilitates the symbolical-numerical solution of the problem without using simplifications.

$$\mathbf{S} = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix};$$

The rotation matrix is

$$\text{R}=(I_3 - S)^{-1}\,(I_3 + S),$$

where I_3 is a 3×3 identity matrix.

```
I3 = IdentityMatrix[3];
R = Inverse[(I3 - S)].(I3 + S) // Simplify; MatrixForm[R]
```

$$\begin{pmatrix} \frac{1+a^2-b^2-c^2}{1+a^2+b^2+c^2} & \frac{2\,a\,b-2\,c}{1+a^2+b^2+c^2} & \frac{2\,(b+a\,c)}{1+a^2+b^2+c^2} \\ \frac{2\,(a\,b+c)}{1+a^2+b^2+c^2} & \frac{1-a^2+b^2-c^2}{1+a^2+b^2+c^2} & -\frac{2\,(a-b\,c)}{1+a^2+b^2+c^2} \\ \frac{2\,(-b+a\,c)}{1+a^2+b^2+c^2} & \frac{2\,(a+b\,c)}{1+a^2+b^2+c^2} & \frac{1-a^2-b^2+c^2}{1+a^2+b^2+c^2} \end{pmatrix}$$

for which the next restriction is true:

```
R.Transpose[R] // Simplify // MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The axial rotation angles (Cardan angles) can be obtained from the rotation matrix (R) through:

$$\tan\alpha = \frac{r_{23}}{r_{33}} \Rightarrow \Lambda_\Gamma = \arctan\frac{r_{23}}{r_{33}}$$

$$\tan\beta = \frac{-r_{31}}{\sqrt{r_{11}^2 + r_{12}^2}} \Rightarrow \phi_\Gamma = \arctan\frac{-r_{31}}{\sqrt{r_{11}^2 + r_{12}^2}}, \text{ vagy } \beta = -\arcsin r_{13}$$

$$\tan\gamma = \frac{r_{12}}{r_{11}} \Rightarrow \Xi_\Gamma = \arctan\frac{r_{12}}{r_{11}}$$

Cardan angles in degree

```
αR[R_] := ArcTan[R[[3, 3]], R[[2, 3]]] * 180/π;
βR[R_] := -ArcSin[R[[1, 3]]] * 180/π; γR[R_] := ArcTan[R[[1, 1]], R[[1, 2]]] * 180/π;
Cardan[R_] := {αR[R], βR[R], γR[R]}
```

Cardan angles in seconds

```
αRs[R_] := αR[R] * 3600; βRs[R_] := βR[R] * 3600; γRs[R_] := γR[R] * 3600;
CardanS[R_] := {αRs[R], βRs[R], γRs[R]}
```

Instead of the scale matrix (W) we can use the inverse of the scales getting simplier equations in this way.

Let us call $\sigma_i = \frac{1}{s_i}$ and $\Omega = W^{-1}$

```
Ω = (σ1  0   0
     0   σ2  0
     0   0   σ3);
```

For the determination of the 9 parameters of the transformation (a, b, c, X_0, Y_0, Z_0, s_1, s_2, s_3) we need 3 non-collinear points with known coordinates in both coordinate systems. In further, instead of the scale parameters (s_1, s_2, s_3), we will use (σ_1, σ_2, σ_3) to get more simple equations.
Expressing the rotation matrix with the skew-symmetric matrix and using the inverse of the scale matrix (Ω) the nonlinear system to be solved for ($f_i = 0$) determining the 9 parameters leads to:

For $i = 1$ point

```
(f1; f2; f3) = ((I3 - S).Ω.(xi; yi; zi) - (I3 + S).(Xi; Yi; Zi) - (I3 - S).Ω.(X0; Y0; Z0)) /. i → 1 // Expand;
MatrixForm[(f1; f2; f3)];
```

For $i = 2$ point

```
(f4; f5; f6) = ((I3 - S).Ω.(xi; yi; zi) - (I3 + S).(Xi; Yi; Zi) - (I3 - S).Ω.(X0; Y0; Z0)) /. i → 2 // Expand;
MatrixForm[(f4; f5; f6)];
```

For $i = 3$ point

```
(f7; f8; f9) = ((I3 - S).Ω.(xi; yi; zi) - (I3 + S).(Xi; Yi; Zi) - (I3 - S).Ω.(X0; Y0; Z0)) /. i → 3 // Expand;
MatrixForm[(f7; f8; f9)];
```

or

```
sys = Table[fi, {i, 1, 9}];
eq = Table[{"f"i, sys[[i]]}, {i, 1, 9}]; TableForm[eq]
```

$$
\begin{array}{ll}
f_1 & -X_1 + c\,Y_1 - b\,Z_1 + x_1\,\sigma_1 - X_0\,\sigma_1 + c\,y_1\,\sigma_2 - c\,Y_0\,\sigma_2 - b\,z_1\,\sigma_3 + b\,Z_0\,\sigma_3 \\
f_2 & -c\,X_1 - Y_1 + a\,Z_1 - c\,x_1\,\sigma_1 + c\,X_0\,\sigma_1 + y_1\,\sigma_2 - Y_0\,\sigma_2 + a\,z_1\,\sigma_3 - a\,Z_0\,\sigma_3 \\
f_3 & b\,X_1 - a\,Y_1 - Z_1 + b\,x_1\,\sigma_1 - b\,X_0\,\sigma_1 - a\,y_1\,\sigma_2 + a\,Y_0\,\sigma_2 + z_1\,\sigma_3 - Z_0\,\sigma_3 \\
f_4 & -X_2 + c\,Y_2 - b\,Z_2 + x_2\,\sigma_1 - X_0\,\sigma_1 + c\,y_2\,\sigma_2 - c\,Y_0\,\sigma_2 - b\,z_2\,\sigma_3 + b\,Z_0\,\sigma_3 \\
f_5 & -c\,X_2 - Y_2 + a\,Z_2 - c\,x_2\,\sigma_1 + c\,X_0\,\sigma_1 + y_2\,\sigma_2 - Y_0\,\sigma_2 + a\,z_2\,\sigma_3 - a\,Z_0\,\sigma_3 \\
f_6 & b\,X_2 - a\,Y_2 - Z_2 + b\,x_2\,\sigma_1 - b\,X_0\,\sigma_1 - a\,y_2\,\sigma_2 + a\,Y_0\,\sigma_2 + z_2\,\sigma_3 - Z_0\,\sigma_3 \\
f_7 & -X_3 + c\,Y_3 - b\,Z_3 + x_3\,\sigma_1 - X_0\,\sigma_1 + c\,y_3\,\sigma_2 - c\,Y_0\,\sigma_2 - b\,z_3\,\sigma_3 + b\,Z_0\,\sigma_3 \\
f_8 & -c\,X_3 - Y_3 + a\,Z_3 - c\,x_3\,\sigma_1 + c\,X_0\,\sigma_1 + y_3\,\sigma_2 - Y_0\,\sigma_2 + a\,z_3\,\sigma_3 - a\,Z_0\,\sigma_3 \\
f_9 & b\,X_3 - a\,Y_3 - Z_3 + b\,x_3\,\sigma_1 - b\,X_0\,\sigma_1 - a\,y_3\,\sigma_2 + a\,Y_0\,\sigma_2 + z_3\,\sigma_3 - Z_0\,\sigma_3
\end{array}
$$

Similarly to C_7 (3, 3) problem (see Awange and Grafarend (2005)) we can reduce the equation system by subtracting the equations. In this way the translation parameters can be eliminated. The elimination of the translation parameters can be done according to the next substractions

```
r1 = f1 - f7;

r2 = f4 - f7;

r3 = f2 - f8;

r4 = f5 - f8;

r5 = f3 - f9;

r6 = f6 - f9;
```

In the remaining 6 equations there is only 6 unknown parameters (a, b, c, σ_1, σ_2, σ_3).

```
sysR = {r1, r2, r3, r4, r5, r6}
```

$$
\begin{aligned}
\{&-X_1 + X_3 + c\,Y_1 - c\,Y_3 - b\,Z_1 + b\,Z_3 + x_1\,\sigma_1 - x_3\,\sigma_1 + c\,y_1\,\sigma_2 - c\,y_3\,\sigma_2 - b\,z_1\,\sigma_3 + b\,z_3\,\sigma_3, \\
&-X_2 + X_3 + c\,Y_2 - c\,Y_3 - b\,Z_2 + b\,Z_3 + x_2\,\sigma_1 - x_3\,\sigma_1 + c\,y_2\,\sigma_2 - c\,y_3\,\sigma_2 - b\,z_2\,\sigma_3 + b\,z_3\,\sigma_3, \\
&-c\,X_1 + c\,X_3 - Y_1 + Y_3 + a\,Z_1 - a\,Z_3 - c\,x_1\,\sigma_1 + c\,x_3\,\sigma_1 + y_1\,\sigma_2 - y_3\,\sigma_2 + a\,z_1\,\sigma_3 - a\,z_3\,\sigma_3, \\
&-c\,X_2 + c\,X_3 - Y_2 + Y_3 + a\,Z_2 - a\,Z_3 - c\,x_2\,\sigma_1 + c\,x_3\,\sigma_1 + y_2\,\sigma_2 - y_3\,\sigma_2 + a\,z_2\,\sigma_3 - a\,z_3\,\sigma_3, \\
&b\,X_1 - b\,X_3 - a\,Y_1 + a\,Y_3 - Z_1 + Z_3 + b\,x_1\,\sigma_1 - b\,x_3\,\sigma_1 - a\,y_1\,\sigma_2 + a\,y_3\,\sigma_2 + z_1\,\sigma_3 - z_3\,\sigma_3, \\
&b\,X_2 - b\,X_3 - a\,Y_2 + a\,Y_3 - Z_2 + Z_3 + b\,x_2\,\sigma_1 - b\,x_3\,\sigma_1 - a\,y_2\,\sigma_2 + a\,y_3\,\sigma_2 + z_2\,\sigma_3 - z_3\,\sigma_3\}
\end{aligned}
$$

Employing Groebner basis and Dixon resultant, the above described reduced equation system can be easily reduced further , if we introduce some new variables, let us call them relative coordinates instead of original ones, see Awange-Grafarend (2003). Let

```
SYSR = sysR // FullSimplify; SYSR // TableForm
```

$-X_1 + X_3 + (x_1 - x_3)\sigma_1 + c(Y_1 - Y_3 + (y_1 - y_3)\sigma_2) + b(-Z_1 + Z_3 + (-z_1 + z_3)\sigma_3)$
$-X_2 + X_3 + (x_2 - x_3)\sigma_1 + c(Y_2 - Y_3 + (y_2 - y_3)\sigma_2) + b(-Z_2 + Z_3 + (-z_2 + z_3)\sigma_3)$
$-Y_1 + Y_3 + c(-X_1 + X_3 + (-x_1 + x_3)\sigma_1) + (y_1 - y_3)\sigma_2 + a(Z_1 - Z_3 + (z_1 - z_3)\sigma_3)$
$-Y_2 + Y_3 + c(-X_2 + X_3 + (-x_2 + x_3)\sigma_1) + (y_2 - y_3)\sigma_2 + a(Z_2 - Z_3 + (z_2 - z_3)\sigma_3)$
$-Z_1 + Z_3 + b(X_1 - X_3 + (x_1 - x_3)\sigma_1) + a(-Y_1 + Y_3 + (-y_1 + y_3)\sigma_2) + (z_1 - z_3)\sigma_3$
$-Z_2 + Z_3 + b(X_2 - X_3 + (x_2 - x_3)\sigma_1) + a(-Y_2 + Y_3 + (-y_2 + y_3)\sigma_2) + (z_2 - z_3)\sigma_3$

It is clear that the introduction of following variables can simplify the description of the system

```
newVarsA = {x12 → x1 - x2, x13 → x1 - x3, x23 → x2 - x3, y12 → y1 - y2, y13 → y1 - y3, y23 → y2 - y3,
   z12 → z1 - z2, z13 → z1 - z3, z23 → z2 - z3,
   X12 → X1 - X2, X13 → X1 - X3, X23 → X2 - X3,
   Y12 → Y1 - Y2, Y13 → Y1 - Y3, Y23 → Y2 - Y3,
   Z12 → Z1 - Z2, Z13 → Z1 - Z3, Z23 → Z2 - Z3};
```

then our equation system is

```
g1 = -X13 + c Y13 - b Z13 + x13 σ1 + c y13 σ2 - b z13 σ3;

g2 = -X23 + c Y23 - b Z23 + x23 σ1 + c y23 σ2 - b z23 σ3;

g3 = -c X13 - Y13 + a Z13 - c x13 σ1 + y13 σ2 + a z13 σ3;

g4 = -c X23 - Y23 + a Z23 - c x23 σ1 + y23 σ2 + a z23 σ3;

g5 = b X13 - a Y13 - Z13 + b x13 σ1 - a y13 σ2 + z13 σ3;

g6 = b X23 - a Y23 - Z23 + b x23 σ1 - a y23 σ2 + z23 σ3;
```

Let us check these equations

```
({g1, g2, g3, g4, g5, g6} /. newVarsA) - SYSR // FullSimplify
```

```
{0, 0, 0, 0, 0, 0}
```

3. Numerical Solutions

Here we shall consider three global methods working without guess of initial values:

a) a general solver for polynomial systems based on numerical Groebner basis and eigensystem methods, built in the *Mathematica* system as a function **`NSolve(#)`**
b) Global minimization based on genetic algorithm, built -in in *Mathematica* as **`NMinimize(#)`**
c) A linear homotopy methods written in *Mathematica* as a function **`LinearHomotopyFR(#),`** see Palancz (2008) and using the built in function **`FindRoot(#)`** employing Newton-Raphson method.

Let us consider the numerical values of 3 Hungarian points in the system of ETRS89 (x_1, y_1, z_1, ..., z_3) and in the local Hungarian system HD72 (Hungarian Datum 1972) ($X_!$, Y_1, Z_1, ..., Z_3).

```
numericalValues = {x1 → 4171409.677, x2 → 4146957.889,
   x3 → 3955632.880, y1 → 1470823.777, y2 → 1277033.850, y3 → 1611863.197,
   z1 → 4580140.907, z2 → 4659439.264, z3 → 4720991.316,
   X1 → 4171352.311, X2 → 4146901.301, X3 → 3955575.649,
   Y1 → 1470893.887, Y2 → 1277104.509, Y3 → 1611933.124,
   Z1 → 4580150.178, Z2 → 4659448.287, Z3 → 4721000.952};
```

The system in numerical form

```
Neqs = {g1, g2, g3, g4, g5, g6} /. newVarsA /. numericalValues
```

$\{-215\,777. + 140\,851.\,b - 141\,039.\,c + 215\,777.\,\sigma_1 - 141\,039.\,c\,\sigma_2 + 140\,850.\,b\,\sigma_3,$
$-191\,326. + 61\,552.7\,b - 334\,829.\,c + 191\,325.\,\sigma_1 - 334\,829.\,c\,\sigma_2 + 61\,552.1\,b\,\sigma_3,$
$141\,039. - 140\,851.\,a - 215\,777.\,c - 215\,777.\,c\,\sigma_1 - 141\,039.\,\sigma_2 - 140\,850.\,a\,\sigma_3,$
$334\,829. - 61\,552.7\,a - 191\,326.\,c - 191\,325.\,c\,\sigma_1 - 334\,829.\,\sigma_2 - 61\,552.1\,a\,\sigma_3,$
$140\,851. + 141\,039.\,a + 215\,777.\,b + 215\,777.\,b\,\sigma_1 + 141\,039.\,a\,\sigma_2 - 140\,850.\,\sigma_3,$
$61\,552.7 + 334\,829.\,a + 191\,326.\,b + 191\,325.\,b\,\sigma_1 + 334\,829.\,a\,\sigma_2 - 61\,552.1\,\sigma_3\}$

3.1 Global polynomial solver based on numerical Groebner basis and eigensystem methods

The solution providing all roots,

```
sol = NSolve[Neqs, {a, b, c, σ1, σ2, σ3}]
```

$\{\{a \to 15.2489,\ b \to 2.85737,\ c \to 3.08269 \times 10^{6},\ \sigma_1 \to -1.00001,\ \sigma_2 \to -0.999998,\ \sigma_3 \to 0.999989\},$
$\{a \to -1.07886 \times 10^{6},\ b \to -0.349972,\ c \to 5.33669,\ \sigma_1 \to 1.00001,\ \sigma_2 \to -0.999998,\ \sigma_3 \to -0.999989\},$
$\{a \to -0.0655786,\ b \to 202\,158.,\ c \to -0.187382,\ \sigma_1 \to -1.00001,\ \sigma_2 \to 0.999998,\ \sigma_3 \to -0.999989\},$
$\{a \to 825\,376.,\ b \to 293\,148.,\ c \to 970\,906.,\ \sigma_1 \to -1.00001,\ \sigma_2 \to -0.999998,\ \sigma_3 \to -0.999989\},$
$\{a \to -3.31199,\ b \to -3.41124 \times 10^{-6},\ c \to 2.81556,\ \sigma_1 \to 1.00001,\ \sigma_2 \to -0.999998,\ \sigma_3 \to 0.999989\},$
$\{a \to -1.21157 \times 10^{-6},\ b \to 1.17632,\ c \to -0.35517,\ \sigma_1 \to -1.00001,\ \sigma_2 \to 0.999998,\ \sigma_3 \to 0.999989\},$
$\{a \to 9.26908 \times 10^{-7},\ b \to -4.94662 \times 10^{-6},$
$c \to -3.24392 \times 10^{-7},\ \sigma_1 \to 1.00001,\ \sigma_2 \to 0.999998,\ \sigma_3 \to 0.999989\},$
$\{a \to 0.301933,\ b \to -0.850109,\ c \to -1.02997 \times 10^{-6},\ \sigma_1 \to 1.00001,\ \sigma_2 \to 0.999998,\ \sigma_3 \to -0.999989\}\}$

From the eight solutions we need the only one, where the scale variables are positive, $\sigma_i > 0$, therefore

```
sol3NSolve = Select[sol, (#[[4, 2]] > 0) ∧ (#[[5, 2]] > 0) ∧ (#[[6, 2]] > 0) &]
```

$\{\{a \to 9.26908 \times 10^{-7},\ b \to -4.94662 \times 10^{-6},$
$c \to -3.24392 \times 10^{-7},\ \sigma_1 \to 1.00001,\ \sigma_2 \to 0.999998,\ \sigma_3 \to 0.999989\}\}$

where

$\sigma_1 = 1.0000054081636032$

The computation time

```
sol = Timing[NSolve[Neqs, {a, b, c, σ1, σ2, σ3}];]
```

{0.125, Null}

Rotation angles in seconds :

```
CardanS[R /. sol3NSolve[[1]]]
```

{-0.382376, 2.04063, 0.13382}

Scale parameters :

```
SetPrecision[{1 / σ1, 1 / σ2, 1 / σ3} /. sol3NSolve[[1]], 10]
```

{0.9999945919, 1.000002156, 1.000010708}

3.2 Global minimization

Using directly the least squares method, we can define the an objective function, the sum of the squares of the errors, which should be minimized. the objective function is

```
nobj = Apply[Plus, Map[#^2 &, Neqs]]; Short[nobj, 10]
```

$(140851. + 141039.\ a + 215777.\ b + 215777.\ b\ \sigma_1 + 141039.\ a\ \sigma_2 - 140850.\ \sigma_3)^2 +$
$(61552.7 + 334829.\ a + 191326.\ b + 191325.\ b\ \sigma_1 + 334829.\ a\ \sigma_2 - 61552.1\ \sigma_3)^2 +$
$(141039. - 140851.\ a - 215777.\ c - 215777.\ c\ \sigma_1 - 141039.\ \sigma_2 - 140850.\ a\ \sigma_3)^2 +$
$(334829. - 61552.7\ a - 191326.\ c - 191325.\ c\ \sigma_1 - 334829.\ \sigma_2 - 61552.1\ a\ \sigma_3)^2 +$
$(-191326. + 61552.7\ b - 334829.\ c + 191325.\ \sigma_1 - 334829.\ c\ \sigma_2 + 61552.1\ b\ \sigma_3)^2 +$
$(-215777. + 140851.\ b - 141039.\ c + 215777.\ \sigma_1 - 141039.\ c\ \sigma_2 + 140850.\ b\ \sigma_3)^2$

```
NMin = NMinimize[nobj, {a, b, c, σ1, σ2, σ3}, Method → "DifferentialEvolution"] // Timing
```

$\{1.11, \{1.14842\times10^{-21}, \{a\to 9.26908\times10^{-7}, b\to -4.94662\times10^{-6},$
$c\to -3.24392\times10^{-7}, \sigma_1\to 1.00001, \sigma_2\to 0.999998, \sigma_3\to 0.999989\}\}\}$

where for example

$\sigma_1 = 1.0000054081636032$

Rotation angles in seconds :

```
CardanS[R /. NMin[[2, 2]]]
```

{-0.382376, 2.04063, 0.13382}

Scale parameters :

```
SetPrecision[{1 / σ1, 1 / σ2, 1 / σ3} /. NMin[[2, 2]], 10]
```

{0.9999945919, 1.000002156, 1.000010708}

3.3 Homotopy Solution

We can employ the convex linear or *BØzout* homotopy, see Drexler(1977), Garcia and Zangwill (1979). The homotopy function is

$$H(x, \beta) = \gamma (1 - \beta) G(x) + \beta F(x)$$

and while β is changing from 0 to 1, the starting system $G(x)$ will be transformed into the original system $F(x)$ to be solved. The method has been implemented into *Mathematica* as a function **LinearHomotopyFR**, see PalÆncz(2008) and using the built in function **FindRoot** employing Newton-Raphson method.

```
LinearHomotopyFR[F_, G_, X_, X0_, γ_, n_] := Module[{H, X0L, λ0, i, β, m, R, RR, j, k, sol},
  Off[FindRoot::"lstol"];
  λ0 = Table[i 1/n, {i, 0, n}];
  H = Flatten[(1 - β) Thread[G γ] + β F];
  m = Length[X];
  k = Length[X0];
  RR = {};
  Do[
   X0L = {X0[[j]]};
   Do[AppendTo[X0L, Map[#[[2]] &,
      FindRoot[H /. β → λ0[[i + 1]], MapThread[{#1, #2} &, {X, X0L[[i]]}]]]], {i, 1, n}];
   R = {};
   Do[AppendTo[R, Interpolation[MapThread[{#1, #2[[i]]} &, {λ0, X0L}]]], {i, 1, m}];
   AppendTo[RR, {Map[Chop[N[#[1]]] &, R], R}],
   {j, 1, k}];
  sol = Transpose[RR];
  {sol[[1]], Table[MapThread[#1[λ] → #2[λ] &, {X, sol[[2, i]]}], {i, 1, Length[X0]}]}]
```

This function will compute the homotopy paths using successive Newton-Raphson method,

Input variables

F - list of functions of the polynomial system to be solved, $F = \{ f_1(x), f_2(x),..., f_n(x)\}$

G - list of the starting system, $G = \{g_1(x), g_2(x),..., g_n(x)\}$,

X - list of the independent variables $X = \{x_1, x_2,..., x_n\}$

$X0$ - list of initial values, $X0 = \{\{x0_1, x0_2,..., x0_n\}_1, \{x0_1, x0_2,..., x0_n\}_2,..., \{x0_1, x0_2,..., x0_n\}_m\}$
where m is the number of the roots of F

γ - list of complex numbers, $\{\gamma_1, \gamma_2,..., \gamma_n\}$, it means γ_i can be different for every $g_i(x)$,
If the starting system is generated for polynomials all $\gamma_i = 1$.

n - number of the subintervalls in [0, 1].

Output variables

sol[1] - list of the *i*-th solutions correspondig to the *i*-th initial values, $\{x0_1, x0_2,..., x0_n\}_i$, $i = 1...m$

sol[2] - list of the path of *i*-th solutions in form of interpolating functions of the variables
correspondig to the *i*-th initial values, $\{x0_1, x0_2,..., x0_n\}_i$, $i = 1...m$,
$\{\{\varphi_1, \varphi_2, ..., \varphi_n\}_1, \{\varphi_1, \varphi_2, ..., \varphi_n\}_2,..., \{\varphi_1, \varphi_2, ..., \varphi_n\}_m\}$, $\varphi_i = \varphi_i(\lambda)$

Instead of computing all of the roots bounded by the Bezout bound, which is 64 in this case (six quadratics), let us consider our starting sytem as the linear part of the nonlinear system,

```
G1 = -X13 + c Y13 - b Z13 + x13 σ1;

G2 = -X23 + c Y23 - b Z23 + x23 σ1;

G3 = -c X13 - Y13 + a Z13 + y13 σ2;

G4 = -c X23 - Y23 + a Z23 + y23 σ2;

G5 = b X13 - a Y13 - Z13 + z13 σ3;

G6 = b X23 - a Y23 - Z23 + z23 σ3;
```

In numerical form

```
Geqs = {G1, G2, G3, G4, G5, G6} /. newVarsA /. numericalValues
```

```
{-215777. + 140851. b - 141039. c + 215777. σ1, -191326. + 61552.7 b - 334829. c + 191325. σ1,
 141039. - 140851. a - 215777. c - 141039. σ2, 334829. - 61552.7 a - 191326. c - 334829. σ2,
 140851. + 141039. a + 215777. b - 140850. σ3, 61552.7 + 334829. a + 191326. b - 61552.1 σ3}
```

The starting values can be easily computed by solving this linear system

```
solG = NSolve[Geqs, {a, b, c, σ1, σ2, σ3}] // Flatten
```

$\{a \to 1.85381 \times 10^{-6},\ b \to -9.89319 \times 10^{-6},\ c \to -6.48787 \times 10^{-7},\ \sigma_1 \to 1.00001,\ \sigma_2 \to 0.999998,\ \sigma_3 \to 0.999989\}$

The list of the variables

```
XX = {a, b, c, σ1, σ2, σ3};
```

The initial values

```
X0 = {Map[#[[2]] &, solG]}
```

$\{\{1.85381 \times 10^{-6}, -9.89319 \times 10^{-6}, -6.48787 \times 10^{-7}, 1.00001, 0.999998, 0.999989\}\}$

Now, the starting system is **Geqs** and the target system is **Neqs**. In this case , we can work with real values, there is no need to use complex number to avoid singularity of the homotopy function

```
γ = {1, 1, 1, 1, 1, 1};

sol = LinearHomotopyFR[Neqs, Geqs, XX, X0, γ, 10];

sol[[1]]
```

$\{\{9.26908 \times 10^{-7}, -4.94662 \times 10^{-6}, -3.24392 \times 10^{-7}, 1.00001, 0.999998, 0.999989\}\}$

where

$\sigma_1 = 1.000005408163603$

Rotation angles in seconds :

```
CardanS[R /. {a → sol[[1, 1, 1]], b → sol[[1, 1, 2]], c → sol[[1, 1, 3]]}]
```

```
{-0.382376, 2.04063, 0.13382}
```

Scale parameters :

```
SetPrecision[
 {1 / σ1, 1 / σ2, 1 / σ3} /. {σ1 → sol[[1, 1, 4]], σ2 → sol[[1, 1, 5]], σ3 → sol[[1, 1, 6]]}, 10]
```

```
{0.9999945919, 1.000002156, 1.000010708}
```

The computation time

```
sol = Timing[LinearHomotopyFR[Neqs, Geqs, XX, X0, γ, 10];]
```

```
{0.031, Null}
```

Remarks : According to our numerical experiences, the generation of the starting system for the homotopy function from the linear part of the original nonlinear one is proved to be reliable in case of many different systems. Homotopy solution is roughly two-three times faster than the global polynomial solver (**NSolve(#)**) and 20- 30 times faster than global minimization (**NMinimize(#)**). In addition there is no need to select the proper (positive) solution like in case of using

NSolve(#).

Table 1.
Comparing Global Numerical Methods for solving 3 Point Problem

Method	Time of computation (sec)	Solution
Numerical Groebner basis & eigensystem method	0.125	many
Global minimization	1.110	one
Homotopy method	0.031	one

4. Symbolic solutions

4.1 Improved Dixon resultant - Kapur, Saxena and Yang method

Our intention is to reduce the computation time of the 3 point solution, because, for example, in case of Gauss-Jacobi method for N points, it is very important to use an effective method for solving one of the many combinations. This reduction can be done by Dixon resultant - see Nakos and Williams (2002) and originally suggested by Kapur, Saxena and Yang (1994) - or by reduced Groebner basis as well as by using other multipolynomial resultant. Here we employ the Dixon resultant.

```
<< Resultant`Dixon`
```

Unfortunately, even the reduced equation system with 6 nonlinear equations and with 6 unknown parameters had no direct reduction to a univariate polynomial using Dixon resultant via *Mathematica.* But we can use either method to reduce the equation system to 3 equations with 3 parameters. For the sake of illustration and completion we will carry out this computation. However , a far superior method will be presented in the next section. To begin, one has to three times select 4 - 4 independent equations (fourplexes) from the 6 equations, arbitrarily. From a fourplex, we can eliminate *a*, *b*, *c* parameters using Dixon or Grobner method (both give the same result, here we employed the Dixon resultant). Choosing three independent combinations of the fourplexes, we get three nonlinear equations for the scale parameters (respectively for the inverse of the 3 scale parameters, σ_i).

Elimination of *a*, *b* and *c* from the fourplexes:

```
dr1σ123 = DixonResultant[{g1, g2, g3, g4}, {a, b, c}, {s1, s2, s3}]
```

$$X23^2\,Z13^2 + Y23^2\,Z13^2 - 2\,X13\,X23\,Z13\,Z23 - 2\,Y13\,Y23\,Z13\,Z23 + X13^2\,Z23^2 + Y13^2\,Z23^2 - x23^2\,Z13^2\,\sigma_1^2 + 2\,x13\,x23\,Z13\,Z23\,\sigma_1^2 - x13^2\,Z23^2\,\sigma_1^2 - y23^2\,Z13^2\,\sigma_2^2 + 2\,y13\,y23\,Z13\,Z23\,\sigma_2^2 - y13^2\,Z23^2\,\sigma_2^2 + 2\,X23^2\,z13\,Z13\,\sigma_3 + 2\,Y23^2\,z13\,Z13\,\sigma_3 - 2\,X13\,X23\,Z13\,z23\,\sigma_3 - 2\,Y13\,Y23\,Z13\,z23\,\sigma_3 - 2\,X13\,X23\,z13\,Z23\,\sigma_3 - 2\,Y13\,Y23\,z13\,Z23\,\sigma_3 + 2\,X13^2\,z23\,Z23\,\sigma_3 + 2\,Y13^2\,z23\,Z23\,\sigma_3 - 2\,x23^2\,z13\,Z13\,\sigma_1^2\,\sigma_3 + 2\,x13\,x23\,Z13\,z23\,\sigma_1^2\,\sigma_3 + 2\,x13\,x23\,z13\,Z23\,\sigma_1^2\,\sigma_3 - 2\,x13^2\,z23\,Z23\,\sigma_1^2\,\sigma_3 - 2\,y23^2\,z13\,Z13\,\sigma_2^2\,\sigma_3 + 2\,y13\,y23\,Z13\,z23\,\sigma_2^2\,\sigma_3 + 2\,y13\,y23\,z13\,Z23\,\sigma_2^2\,\sigma_3 - 2\,y13^2\,z23\,Z23\,\sigma_2^2\,\sigma_3 + X23^2\,z13^2\,\sigma_3^2 + Y23^2\,z13^2\,\sigma_3^2 - 2\,X13\,X23\,z13\,z23\,\sigma_3^2 - 2\,Y13\,Y23\,z13\,z23\,\sigma_3^2 + X13^2\,z23^2\,\sigma_3^2 + Y13^2\,z23^2\,\sigma_3^2 - x23^2\,z13^2\,\sigma_1^2\,\sigma_3^2 + 2\,x13\,x23\,z13\,z23\,\sigma_1^2\,\sigma_3^2 - x13^2\,z23^2\,\sigma_1^2\,\sigma_3^2 - y23^2\,z13^2\,\sigma_2^2\,\sigma_3^2 + 2\,y13\,y23\,z13\,z23\,\sigma_2^2\,\sigma_3^2 - y13^2\,z23^2\,\sigma_2^2\,\sigma_3^2$$

and similarly

```
dr2σ123 = DixonResultant[{g2, g3, g4, g5}, {a, b, c}, {s1, s2, s3}];
```

```
dr3σ123 = DixonResultant[{g3, g4, g5, g6}, {a, b, c}, {s1, s2, s3}];
```

The succesfull application of Dixon - KSY, in order to carry out further reduction, needs to compute compact coefficients of the remained three polynomials.

In order to determine the compact form of the polynomial equations, the following functions, providing to find coefficients in a multivariate polynomial, are developed,

```
Expon[equ_] := Union[Table[Exponent[equ[[i]], Reverse[vari]], {i, Length[equ]}]]

Koeff[poly_, {i_, j_, k_}] := CoefficientList[poly, Reverse[vari]][[i + 1, j + 1, k + 1]]

KoeffList[poly_] := Table[Koeff[poly, Expon[poly][[i]]], {i, Length[Expon[poly]]}]

ExpVari[{x_, y_, z_}, {i_, j_, k_}] := x^i y^j z^k

ExpVariList[poly_] :=
 Table[ExpVari[Reverse[vari], Expon[poly][[i]]], {i, Length[Expon[poly]]}]

vari = {x, y, z};

pl = 15 + x^2 + 2 y + 3 x y + 6 x^3 + 8 x  +  9 x + 20 z
```

$$15 + 17\,x + x^2 + 6\,x^3 + 2\,y + 3\,x\,y + 20\,z$$

```
ExpVariList[pl]
```

$$\{1,\ x,\ x^2,\ x^3,\ y,\ x\,y,\ z\}$$

```
KoeffList[pl]
```

$$\{15,\ 17,\ 1,\ 6,\ 2,\ 3,\ 20\}$$

Let us apply this method for our system. The variables are

```
vari = {σ1, σ2, σ3};
```

The first equation is

```
dr1P = dr1σ123 // Expand
```

$$\begin{aligned}
&X23^2\,Z13^2 + Y23^2\,Z13^2 - 2\,X13\,X23\,Z13\,Z23 - 2\,Y13\,Y23\,Z13\,Z23 + X13^2\,Z23^2 + Y13^2\,Z23^2 - \\
&x23^2\,Z13^2\,\sigma_1^2 + 2\,x13\,x23\,Z13\,Z23\,\sigma_1^2 - x13^2\,Z23^2\,\sigma_1^2 - y23^2\,Z13^2\,\sigma_2^2 + 2\,y13\,y23\,Z13\,Z23\,\sigma_2^2 - \\
&y13^2\,Z23^2\,\sigma_2^2 + 2\,X23^2\,z13\,Z13\,\sigma_3 + 2\,Y23^2\,z13\,Z13\,\sigma_3 - 2\,X13\,X23\,Z13\,z23\,\sigma_3 - \\
&2\,Y13\,Y23\,Z13\,z23\,\sigma_3 - 2\,X13\,X23\,z13\,Z23\,\sigma_3 - 2\,Y13\,Y23\,z13\,Z23\,\sigma_3 + 2\,X13^2\,z23\,Z23\,\sigma_3 + \\
&2\,Y13^2\,z23\,Z23\,\sigma_3 - 2\,x23^2\,z13\,Z13\,\sigma_1^2\,\sigma_3 + 2\,x13\,x23\,Z13\,z23\,\sigma_1^2\,\sigma_3 + 2\,x13\,x23\,z13\,Z23\,\sigma_1^2\,\sigma_3 - \\
&2\,x13^2\,z23\,Z23\,\sigma_1^2\,\sigma_3 - 2\,y23^2\,z13\,Z13\,\sigma_2^2\,\sigma_3 + 2\,y13\,y23\,Z13\,z23\,\sigma_2^2\,\sigma_3 + 2\,y13\,y23\,z13\,Z23\,\sigma_2^2\,\sigma_3 - \\
&2\,y13^2\,z23\,Z23\,\sigma_2^2\,\sigma_3 + X23^2\,z13^2\,\sigma_3^2 + Y23^2\,z13^2\,\sigma_3^2 - 2\,X13\,X23\,z13\,z23\,\sigma_3^2 - \\
&2\,Y13\,Y23\,z13\,z23\,\sigma_3^2 + X13^2\,z23^2\,\sigma_3^2 + Y13^2\,z23^2\,\sigma_3^2 - x23^2\,z13^2\,\sigma_1^2\,\sigma_3^2 + 2\,x13\,x23\,z13\,z23\,\sigma_1^2\,\sigma_3^2 - \\
&x13^2\,z23^2\,\sigma_1^2\,\sigma_3^2 - y23^2\,z13^2\,\sigma_2^2\,\sigma_3^2 + 2\,y13\,y23\,z13\,z23\,\sigma_2^2\,\sigma_3^2 - y13^2\,z23^2\,\sigma_2^2\,\sigma_3^2
\end{aligned}$$

The corresponding terms are

```
d1 = ExpVariList[dr1P]
```

$$\{1,\ \sigma_1^2,\ \sigma_2^2,\ \sigma_3,\ \sigma_1^2\,\sigma_3,\ \sigma_2^2\,\sigma_3,\ \sigma_3^2,\ \sigma_1^2\,\sigma_3^2,\ \sigma_2^2\,\sigma_3^2\}$$

and their coefficients

```
C1 = KoeffList[dr1P]
```

$$\begin{aligned}
&\{X23^2\,Z13^2 + Y23^2\,Z13^2 - 2\,X13\,X23\,Z13\,Z23 - 2\,Y13\,Y23\,Z13\,Z23 + X13^2\,Z23^2 + Y13^2\,Z23^2, \\
&-x23^2\,Z13^2 + 2\,x13\,x23\,Z13\,Z23 - x13^2\,Z23^2,\ -y23^2\,Z13^2 + 2\,y13\,y23\,Z13\,Z23 - y13^2\,Z23^2, \\
&2\,X23^2\,z13\,Z13 + 2\,Y23^2\,z13\,Z13 - 2\,X13\,X23\,Z13\,z23 - 2\,Y13\,Y23\,Z13\,z23 - \\
&\quad 2\,X13\,X23\,z13\,Z23 - 2\,Y13\,Y23\,z13\,Z23 + 2\,X13^2\,z23\,Z23 + 2\,Y13^2\,z23\,Z23, \\
&-2\,x23^2\,z13\,Z13 + 2\,x13\,x23\,Z13\,z23 + 2\,x13\,x23\,z13\,Z23 - 2\,x13^2\,z23\,Z23, \\
&-2\,y23^2\,z13\,Z13 + 2\,y13\,y23\,Z13\,z23 + 2\,y13\,y23\,z13\,Z23 - 2\,y13^2\,z23\,Z23, \\
&X23^2\,z13^2 + Y23^2\,z13^2 - 2\,X13\,X23\,z13\,z23 - 2\,Y13\,Y23\,z13\,z23 + X13^2\,z23^2 + Y13^2\,z23^2, \\
&-x23^2\,z13^2 + 2\,x13\,x23\,z13\,z23 - x13^2\,z23^2,\ -y23^2\,z13^2 + 2\,y13\,y23\,z13\,z23 - y13^2\,z23^2\}
\end{aligned}$$

It is easy to check our result

```
dr1P - (C1.d1) // Simplify
```

```
0
```

Now we shall introduce the following new coefficients representing the original ones

```
HH = Table[H_i, {i, 0, Length[d1] - 1}]
```

$\{H_0, H_1, H_2, H_3, H_4, H_5, H_6, H_7, H_8\}$

Then the proper assignments are

```
C1s = MapThread[#1 → #2 &, {HH, C1}]
```

$\{H_0 \to X23^2\ Z13^2 + Y23^2\ Z13^2 - 2\ X13\ X23\ Z13\ Z23 - 2\ Y13\ Y23\ Z13\ Z23 + X13^2\ Z23^2 + Y13^2\ Z23^2,$
$H_1 \to -x23^2\ Z13^2 + 2\ x13\ x23\ Z13\ Z23 - x13^2\ Z23^2,\ H_2 \to -y23^2\ Z13^2 + 2\ y13\ y23\ Z13\ Z23 - y13^2\ Z23^2,$
$H_3 \to 2\ X23^2\ z13\ Z13 + 2\ Y23^2\ z13\ Z13 - 2\ X13\ X23\ Z13\ z23 - 2\ Y13\ Y23\ Z13\ z23 -$
$2\ X13\ X23\ z13\ Z23 - 2\ Y13\ Y23\ z13\ Z23 + 2\ X13^2\ z23\ Z23 + 2\ Y13^2\ z23\ Z23,$
$H_4 \to -2\ x23^2\ z13\ Z13 + 2\ x13\ x23\ Z13\ z23 + 2\ x13\ x23\ z13\ Z23 - 2\ x13^2\ z23\ Z23,$
$H_5 \to -2\ y23^2\ z13\ Z13 + 2\ y13\ y23\ Z13\ z23 + 2\ y13\ y23\ z13\ Z23 - 2\ y13^2\ z23\ Z23,$
$H_6 \to X23^2\ z13^2 + Y23^2\ z13^2 - 2\ X13\ X23\ z13\ z23 - 2\ Y13\ Y23\ z13\ z23 + X13^2\ z23^2 + Y13^2\ z23^2,$
$H_7 \to -x23^2\ z13^2 + 2\ x13\ x23\ z13\ z23 - x13^2\ z23^2,\ H_8 \to -y23^2\ z13^2 + 2\ y13\ y23\ z13\ z23 - y13^2\ z23^2\}$

Consequently, our first equation can be written in a compact form, namely

```
e1 = HH.d1
```

$H_0 + H_1\ \sigma_1^2 + H_2\ \sigma_2^2 + H_3\ \sigma_3 + H_4\ \sigma_1^2\ \sigma_3 + H_5\ \sigma_2^2\ \sigma_3 + H_6\ \sigma_3^2 + H_7\ \sigma_1^2\ \sigma_3^2 + H_8\ \sigma_2^2\ \sigma_3^2$

Similarly, one can carry out these computations for the two other polynomials.

The compact form of the second and third equations,

$f1 = J_0 + J_1\ \sigma_1 + J_2\ \sigma_1^2 + J_3\ \sigma_1^3 + J_4\ \sigma_2^2 + J_5\ \sigma_1\ \sigma_2^2 + J_6\ \sigma_3 + J_7\ \sigma_1\ \sigma_3 +$
$J_8\ \sigma_1^2\ \sigma_3 + J_9\ \sigma_1^3\ \sigma_3 + J_{10}\ \sigma_2^2\ \sigma_3 + J_{11}\ \sigma_1\ \sigma_2^2\ \sigma_3 + J_{12}\ \sigma_3^2 + J_{13}\ \sigma_1\ \sigma_3^2 + J_{14}\ \sigma_3^3 + J_{15}\ \sigma_1\ \sigma_3^3;$

$g1 = K_0 + K_1\ \sigma_1 + K_2\ \sigma_1^2 + K_3\ \sigma_2^2 + K_4\ \sigma_1\ \sigma_2^2 + K_5\ \sigma_1^2\ \sigma_2^2 + K_6\ \sigma_3^2 + K_7\ \sigma_1\ \sigma_3^2 + K_8\ \sigma_1^2\ \sigma_3^2;$

and their coefficients

$C2s = \{J_0 \to X13\ X23^2\ Z13 + X13\ Y23^2\ Z13 - X13^2\ X23\ Z23 + X23\ Y13^2\ Z23 - 2\ X13\ Y13\ Y23\ Z23 +$
$X23\ Z13^2\ Z23 - X13\ Z13\ Z23^2,\ J_1 \to x13\ X23^2\ Z13 + x13\ Y23^2\ Z13 + X13^2\ x23\ Z23 -$
$2\ x13\ X13\ X23\ Z23 + x23\ Y13^2\ Z23 - 2\ x13\ Y13\ Y23\ Z23 + x23\ Z13^2\ Z23 - x13\ Z13\ Z23^2,$
$J_2 \to -X13\ x23^2\ Z13 + 2\ x13\ X13\ x23\ Z23 - x13^2\ X23\ Z23,\ J_3 \to -x13\ x23^2\ Z13 + x13^2\ x23\ Z23,$
$J_4 \to -X13\ y23^2\ Z13 - X23\ y13^2\ Z23 + 2\ X13\ y13\ y23\ Z23,$
$J_5 \to -x13\ y23^2\ Z13 - x23\ y13^2\ Z23 + 2\ x13\ y13\ y23\ Z23,$
$J_6 \to X13\ X23^2\ z13 + X13\ Y23^2\ z13 - X13^2\ X23\ z23 + X23\ Y13^2\ z23 -$
$2\ X13\ Y13\ Y23\ z23 + X23\ Z13^2\ z23 - 2\ X13\ Z13\ z23\ Z23 + X13\ z13\ Z23^2,$
$J_7 \to x13\ X23^2\ z13 + x13\ Y23^2\ z13 + X13^2\ x23\ z23 - 2\ x13\ X13\ X23\ z23 + x23\ Y13^2\ z23 -$
$2\ x13\ Y13\ Y23\ z23 + x23\ Z13^2\ z23 - 2\ x13\ Z13\ z23\ Z23 + x13\ z13\ Z23^2,$
$J_8 \to -X13\ x23^2\ z13 + 2\ x13\ X13\ x23\ z23 - x13^2\ X23\ z23,\ J_9 \to -x13\ x23^2\ z13 + x13^2\ x23\ z23,$
$J_{10} \to -X13\ y23^2\ z13 - X23\ y13^2\ z23 + 2\ X13\ y13\ y23\ z23,$
$J_{11} \to -x13\ y23^2\ z13 - x23\ y13^2\ z23 + 2\ x13\ y13\ y23\ z23,$
$J_{12} \to -X13\ Z13\ z23^2 - X23\ z13^2\ Z23 + 2\ X13\ z13\ z23\ Z23,$
$J_{13} \to -x13\ Z13\ z23^2 - x23\ z13^2\ Z23 + 2\ x13\ z13\ z23\ Z23,$
$J_{14} \to -X23\ z13^2\ z23 + X13\ z13\ z23^2,\ J_{15} \to -x23\ z13^2\ z23 + x13\ z13\ z23^2\};$

$$C3s = \{K_0 \to -X23^2\, Y13^2 + 2\, X13\, X23\, Y13\, Y23 - X13^2\, Y23^2 - X23^2\, Z13^2 + 2\, X13\, X23\, Z13\, Z23 - X13^2\, Z23^2,$$
$$K_1 \to -2\, x23\, X23\, Y13^2 + 2\, X13\, x23\, Y13\, Y23 + 2\, x13\, X23\, Y13\, Y23 - 2\, x13\, X13\, Y23^2 - 2\, x23\, X23\, Z13^2 + 2\, X13\, x23\, Z13\, Z23 + 2\, x13\, X23\, Z13\, Z23 - 2\, x13\, X13\, Z23^2,$$
$$K_2 \to -x23^2\, Y13^2 + 2\, x13\, x23\, Y13\, Y23 - x13^2\, Y23^2 - x23^2\, Z13^2 + 2\, x13\, x23\, Z13\, Z23 - x13^2\, Z23^2,$$
$$K_3 \to X23^2\, y13^2 - 2\, X13\, X23\, y13\, y23 + X13^2\, y23^2,$$
$$K_4 \to 2\, x23\, X23\, y13^2 - 2\, X13\, x23\, y13\, y23 - 2\, x13\, X23\, y13\, y23 + 2\, x13\, X13\, y23^2,$$
$$K_5 \to x23^2\, y13^2 - 2\, x13\, x23\, y13\, y23 + x13^2\, y23^2,\ K_6 \to X23^2\, z13^2 - 2\, X13\, X23\, z13\, z23 + X13^2\, z23^2,$$
$$K_7 \to 2\, x23\, X23\, z13^2 - 2\, X13\, x23\, z13\, z23 - 2\, x13\, X23\, z13\, z23 + 2\, x13\, X13\, z23^2,$$
$$K_8 \to x23^2\, z13^2 - 2\, x13\, x23\, z13\, z23 + x13^2\, z23^2\};$$

Now, we can reduce our three polynomial system to two polynomials with two variables via pairwise elimination technique. First, let us consider the first and third polynomials and eliminate the variable σ_2 from both, employing reduced Gröbner basis computed via Groebner - walk and then do the same with the first and second polynomails,

eg13 = GroebnerBasis[{e1, g1}, {σ_1, σ_2, σ_3}, {σ_2}]

$$\{-H_2 K_0 + H_0 K_3 - H_2 K_1 \sigma_1 + H_0 K_4 \sigma_1 - H_2 K_2 \sigma_1^2 + H_1 K_3 \sigma_1^2 + H_0 K_5 \sigma_1^2 + H_1 K_4 \sigma_1^3 + H_1 K_5 \sigma_1^4 - H_5 K_0 \sigma_3 + H_3 K_3 \sigma_3 - H_5 K_1 \sigma_1 \sigma_3 + H_3 K_4 \sigma_1 \sigma_3 - H_5 K_2 \sigma_1^2 \sigma_3 + H_4 K_3 \sigma_1^2 \sigma_3 + H_3 K_5 \sigma_1^2 \sigma_3 + H_4 K_4 \sigma_1^3 \sigma_3 + H_4 K_5 \sigma_1^4 \sigma_3 - H_8 K_0 \sigma_3^2 + H_6 K_3 \sigma_3^2 - H_2 K_6 \sigma_3^2 - H_8 K_1 \sigma_1 \sigma_3^2 + H_6 K_4 \sigma_1 \sigma_3^2 - H_2 K_7 \sigma_1 \sigma_3^2 - H_8 K_2 \sigma_1^2 \sigma_3^2 + H_7 K_3 \sigma_1^2 \sigma_3^2 + H_6 K_5 \sigma_1^2 \sigma_3^2 - H_2 K_8 \sigma_1^2 \sigma_3^2 + H_7 K_4 \sigma_1^3 \sigma_3^2 + H_7 K_5 \sigma_1^4 \sigma_3^2 - H_5 K_6 \sigma_3^3 - H_5 K_7 \sigma_1 \sigma_3^3 - H_5 K_8 \sigma_1^2 \sigma_3^3 - H_8 K_6 \sigma_3^4 - H_8 K_7 \sigma_1 \sigma_3^4 - H_8 K_8 \sigma_1^2 \sigma_3^4\}$$

ef13 = GroebnerBasis[{e1, f1}, {σ_1, σ_2, σ_3}, {σ_2}]

$$\{-H_2 J_0 + H_0 J_4 - H_2 J_1 \sigma_1 + H_0 J_5 \sigma_1 - H_2 J_2 \sigma_1^2 + H_1 J_4 \sigma_1^2 - H_2 J_3 \sigma_1^3 + H_1 J_5 \sigma_1^3 - H_5 J_0 \sigma_3 + H_3 J_4 \sigma_3 - H_2 J_6 \sigma_3 + H_0 J_{10} \sigma_3 - H_5 J_1 \sigma_1 \sigma_3 + H_3 J_5 \sigma_1 \sigma_3 - H_2 J_7 \sigma_1 \sigma_3 + H_0 J_{11} \sigma_1 \sigma_3 - H_5 J_2 \sigma_1^2 \sigma_3 + H_4 J_4 \sigma_1^2 \sigma_3 - H_2 J_8 \sigma_1^2 \sigma_3 + H_1 J_{10} \sigma_1^2 \sigma_3 - H_5 J_3 \sigma_1^3 \sigma_3 + H_4 J_5 \sigma_1^3 \sigma_3 - H_2 J_9 \sigma_1^3 \sigma_3 + H_1 J_{11} \sigma_1^3 \sigma_3 - H_8 J_0 \sigma_3^2 + H_6 J_4 \sigma_3^2 - H_5 J_6 \sigma_3^2 + H_3 J_{10} \sigma_3^2 - H_2 J_{12} \sigma_3^2 - H_8 J_1 \sigma_1 \sigma_3^2 + H_6 J_5 \sigma_1 \sigma_3^2 - H_5 J_7 \sigma_1 \sigma_3^2 + H_3 J_{11} \sigma_1 \sigma_3^2 - H_2 J_{13} \sigma_1 \sigma_3^2 - H_8 J_2 \sigma_1^2 \sigma_3^2 + H_7 J_4 \sigma_1^2 \sigma_3^2 - H_5 J_8 \sigma_1^2 \sigma_3^2 + H_4 J_{10} \sigma_1^2 \sigma_3^2 - H_8 J_3 \sigma_1^3 \sigma_3^2 + H_7 J_5 \sigma_1^3 \sigma_3^2 - H_5 J_9 \sigma_1^3 \sigma_3^2 + H_4 J_{11} \sigma_1^3 \sigma_3^2 - H_8 J_6 \sigma_3^3 + H_6 J_{10} \sigma_3^3 - H_5 J_{12} \sigma_3^3 - H_2 J_{14} \sigma_3^3 - H_8 J_7 \sigma_1 \sigma_3^3 + H_6 J_{11} \sigma_1 \sigma_3^3 - H_5 J_{13} \sigma_1 \sigma_3^3 - H_2 J_{15} \sigma_1 \sigma_3^3 - H_8 J_8 \sigma_1^2 \sigma_3^3 + H_7 J_{10} \sigma_1^2 \sigma_3^3 - H_8 J_9 \sigma_1^3 \sigma_3^3 + H_7 J_{11} \sigma_1^3 \sigma_3^3 - H_8 J_{12} \sigma_3^4 - H_5 J_{14} \sigma_3^4 - H_8 J_{13} \sigma_1 \sigma_3^4 - H_5 J_{15} \sigma_1 \sigma_3^4 - H_8 J_{14} \sigma_3^5 - H_8 J_{15} \sigma_1 \sigma_3^5\}$$

The succesfull application of Dixon - KSY, in order to carry out further reduction, needs to compute compact coefficients of the remaining three polynomials, similarly as before.

The compact form of the two equations are,

$$u1 = U_0 + U_1 \sigma_1 + U_2 \sigma_1^2 + U_3 \sigma_1^3 + U_4 \sigma_1^4 + U_5 \sigma_3 + U_6 \sigma_1 \sigma_3 + U_7 \sigma_1^2 \sigma_3 + U_8 \sigma_1^3 \sigma_3 + U_9 \sigma_1^4 \sigma_3 + U_{10} \sigma_3^2 + U_{11} \sigma_1 \sigma_3^2 + U_{12} \sigma_1^2 \sigma_3^2 + U_{13} \sigma_1^3 \sigma_3^2 + U_{14} \sigma_1^4 \sigma_3^2 + U_{15} \sigma_3^3 + U_{16} \sigma_1 \sigma_3^3 + U_{17} \sigma_1^2 \sigma_3^3 + U_{18} \sigma_3^4 + U_{19} \sigma_1 \sigma_3^4 + U_{20} \sigma_1^2 \sigma_3^4;$$

$$v1 = V_0 + V_1 \sigma_1 + V_2 \sigma_1^2 + V_3 \sigma_1^3 + V_4 \sigma_3 + V_5 \sigma_1 \sigma_3 + V_6 \sigma_1^2 \sigma_3 + V_7 \sigma_1^3 \sigma_3 + V_8 \sigma_3^2 + V_9 \sigma_1 \sigma_3^2 + V_{10} \sigma_1^2 \sigma_3^2 + V_{11} \sigma_1^3 \sigma_3^2 + V_{12} \sigma_3^3 + V_{13} \sigma_1 \sigma_3^3 + V_{14} \sigma_1^2 \sigma_3^3 + V_{15} \sigma_1^3 \sigma_3^3 + V_{16} \sigma_3^4 + V_{17} \sigma_1 \sigma_3^4 + V_{18} \sigma_3^5 + V_{19} \sigma_1 \sigma_3^5;$$

and their coefficients

$$C13s = \{U_0 \to -H_2 K_0 + H_0 K_3,\ U_1 \to -H_2 K_1 + H_0 K_4,\ U_2 \to -H_2 K_2 + H_1 K_3 + H_0 K_5,\ U_3 \to H_1 K_4,\ U_4 \to H_1 K_5,$$
$$U_5 \to -H_5 K_0 + H_3 K_3,\ U_6 \to -H_5 K_1 + H_3 K_4,\ U_7 \to -H_5 K_2 + H_4 K_3 + H_3 K_5,\ U_8 \to H_4 K_4,\ U_9 \to H_4 K_5,$$
$$U_{10} \to -H_8 K_0 + H_6 K_3 - H_2 K_6,\ U_{11} \to -H_8 K_1 + H_6 K_4 - H_2 K_7,\ U_{12} \to -H_8 K_2 + H_7 K_3 + H_6 K_5 - H_2 K_8,\ U_{13} \to H_7 K_4,$$
$$U_{14} \to H_7 K_5,\ U_{15} \to -H_5 K_6,\ U_{16} \to -H_5 K_7,\ U_{17} \to -H_5 K_8,\ U_{18} \to -H_8 K_6,\ U_{19} \to -H_8 K_7,\ U_{20} \to -H_8 K_8\};$$

$$F13s = \{V_0 \to -H_2 J_0 + H_0 J_4,\ V_1 \to -H_2 J_1 + H_0 J_5,\ V_2 \to -H_2 J_2 + H_1 J_4,$$
$$V_3 \to -H_2 J_3 + H_1 J_5,\ V_4 \to -H_5 J_0 + H_3 J_4 - H_2 J_6 + H_0 J_{10},\ V_5 \to -H_5 J_1 + H_3 J_5 - H_2 J_7 + H_0 J_{11},$$
$$V_6 \to -H_5 J_2 + H_4 J_4 - H_2 J_8 + H_1 J_{10},\ V_7 \to -H_5 J_3 + H_4 J_5 - H_2 J_9 + H_1 J_{11},$$
$$V_8 \to -H_8 J_0 + H_6 J_4 - H_5 J_6 + H_3 J_{10} - H_2 J_{12},\ V_9 \to -H_8 J_1 + H_6 J_5 - H_5 J_7 + H_3 J_{11} - H_2 J_{13},$$
$$V_{10} \to -H_8 J_2 + H_7 J_4 - H_5 J_8 + H_4 J_{10},\ V_{11} \to -H_8 J_3 + H_7 J_5 - H_5 J_9 + H_4 J_{11},$$
$$V_{12} \to -H_8 J_6 + H_6 J_{10} - H_5 J_{12} - H_2 J_{14},\ V_{13} \to -H_8 J_7 + H_6 J_{11} - H_5 J_{13} - H_2 J_{15},\ V_{14} \to -H_8 J_8 + H_7 J_{10},$$
$$V_{15} \to -H_8 J_9 + H_7 J_{11},\ V_{16} \to -H_8 J_{12} - H_5 J_{14},\ V_{17} \to -H_8 J_{13} - H_5 J_{15},\ V_{18} \to -H_8 J_{14},\ V_{19} \to -H_8 J_{15}\};$$

The last step is to eliminate σ_3 from both polynomials in order to get a monomial containing only one variable, namely σ_1. This is a very difficult problem because of the big size of the result. The enhanced Dixon resultant as well as the Buchberger and Groebner - walk algorithms all failed because they employ expanded form of the polynomials during the computation, however classical Dixon method working with factorized form was successful.

```
uv1 = ClassicalDixonResultant[{u1, v1}, {σ3}, {s1}]
```

A very large output was generated. Here is a sample of it:

$$\ll 6 \gg + \left(-U_{18} V_{18} - U_{19} V_{18} \sigma_1 - U_{18} V_{19} \sigma_1 - U_{20} V_{18} \sigma_1^2 - U_{19} V_{19} \sigma_1^2 - U_{20} V_{19} \sigma_1^3\right) \left(-\left(U_{18} V_8 - U_{10} V_{16} - U_5 V_{18} + U_{19} V_8 \sigma_1 + U_{18} V_9 \sigma_1 - U_{11} V_{16} \sigma_1 - U_{10} V_{17} \sigma_1 - U_6 V_{18} \sigma_1 - U_5 V_{19} \sigma_1 + U_{20} V_8 \sigma_1^2 + U_{19} V_9 \sigma_1^2 + U_{18} V_{10} \sigma_1^2 - U_{12} V_{16} \sigma_1^2 - U_{11} V_{17} \sigma_1^2 - U_7 V_{18} \sigma_1^2 - U_6 V_{19} \sigma_1^2 + U_{20} V_9 \sigma_1^3 + U_{19} V_{10} \sigma_1^3 + U_{18} V_{11} \sigma_1^3 - U_{13} V_{16} \sigma_1^3 - U_{12} V_{17} \sigma_1^3 - U_8 V_{18} \sigma_1^3 - U_7 V_{19} \sigma_1^3 + U_{20} V_{10} \sigma_1^4 + U_{19} V_{11} \sigma_1^4 - U_{14} V_{16} \sigma_1^4 - U_{13} V_{17} \sigma_1^4 - U_9 V_{18} \sigma_1^4 - U_8 V_{19} \sigma_1^4 + U_{20} V_{11} \sigma_1^5 - U_{14} V_{17} \sigma_1^5 - U_9 V_{19} \sigma_1^5\right) (\ll 1 \gg) + (\ll 1 \gg)(\ll 1 \gg) - \ll 1 \gg + \left(\ll 41 \gg + U_{20} V_{15} \sigma_1^5 - U_{14} V_{19} \sigma_1^5\right)(\ll 1 \gg)\right)$$

Show Less | Show More | Show Full Output | Set Size Limit...

The result is very big in size, in printed form with normal style is about 30 pages! Expanding and count of the number of terms was impossible with our machine!

```
uv1 = ClassicalDixonResultant[{u1, v1}, {σ3}, {s1}]; // Timing
```

{0.187, Null}

Substituting the numerical values into the monomial containing variable σ_1

```
uv1N = uv1 //. F13s /. C13s /. C1s /. C2s /. C3s /. newVarsA /. numericalValues // Expand
```

$$-1.72769819627 \times 10^{401} - 9.7449866389 \times 10^{401} \sigma_1 - 3.8660753633 \times 10^{402} \sigma_1^2 - 9.207256511 \times 10^{402} \sigma_1^3 - 2.506853275 \times 10^{402} \sigma_1^4 + 9.8332329833 \times 10^{403} \sigma_1^5 + 2.6870982810 \times 10^{404} \sigma_1^6 - 1.6775706965 \times 10^{404} \sigma_1^7 - 1.85277067949 \times 10^{405} \sigma_1^8 - 3.40560544729 \times 10^{405} \sigma_1^9 - 2.43384860299 \times 10^{405} \sigma_1^{10} + 9.3794929174 \times 10^{404} \sigma_1^{11} + 4.05119205901 \times 10^{405} \sigma_1^{12} + 4.4796180942 \times 10^{405} \sigma_1^{13} + 1.8599163055 \times 10^{405} \sigma_1^{14} - 1.44260107803 \times 10^{405} \sigma_1^{15} - 2.4810011704 \times 10^{405} \sigma_1^{16} - 1.19345122115 \times 10^{405} \sigma_1^{17} + 2.7798768034 \times 10^{404} \sigma_1^{18} + 7.5039835936 \times 10^{404} \sigma_1^{19} + 5.1504776305 \times 10^{404} \sigma_1^{20} + 1.2716799200 \times 10^{404} \sigma_1^{21} - 1.04215606242 \times 10^{404} \sigma_1^{22} - 1.36695003155 \times 10^{404} \sigma_1^{23} - 8.3629910637 \times 10^{403} \sigma_1^{24} - 3.50247307754 \times 10^{403} \sigma_1^{25} - 1.067078886591 \times 10^{403} \sigma_1^{26} - 2.193636502003 \times 10^{402} \sigma_1^{27} - 2.150540533771065 \times 10^{401} \sigma_1^{28} - 5.5231430672 \times 10^{395} \sigma_1^{29}$$

which can be simplified as,

```
uv1Ns = uv1N 10^-395 // Simplify
```

$$-1.727698196 \times 10^{6} - 9.74498664 \times 10^{6} \sigma_1 - 3.866075363 \times 10^{7} \sigma_1^2 - 9.20725651 \times 10^{7} \sigma_1^3 - 2.506853275 \times 10^{7} \sigma_1^4 + 9.83323298 \times 10^{8} \sigma_1^5 + 2.687098281 \times 10^{9} \sigma_1^6 - 1.677570696 \times 10^{9} \sigma_1^7 - 1.852770679 \times 10^{10} \sigma_1^8 - 3.405605447 \times 10^{10} \sigma_1^9 - 2.433848603 \times 10^{10} \sigma_1^{10} + 9.37949292 \times 10^{9} \sigma_1^{11} + 4.051192059 \times 10^{10} \sigma_1^{12} + 4.479618094 \times 10^{10} \sigma_1^{13} + 1.859916305 \times 10^{10} \sigma_1^{14} - 1.442601078 \times 10^{10} \sigma_1^{15} - 2.481001170 \times 10^{10} \sigma_1^{16} - 1.193451221 \times 10^{10} \sigma_1^{17} + 2.779876803 \times 10^{9} \sigma_1^{18} + 7.50398359 \times 10^{9} \sigma_1^{19} + 5.15047763 \times 10^{9} \sigma_1^{20} + 1.271679920 \times 10^{9} \sigma_1^{21} - 1.042156062 \times 10^{9} \sigma_1^{22} - 1.366950032 \times 10^{9} \sigma_1^{23} - 8.36299106 \times 10^{8} \sigma_1^{24} - 3.502473078 \times 10^{8} \sigma_1^{25} - 1.067078887 \times 10^{8} \sigma_1^{26} - 2.193636502 \times 10^{7} \sigma_1^{27} - 2.150540534 \times 10^{6} \sigma_1^{28} - 5.52314307 \sigma_1^{29}$$

Fortunately we do not need to find all of the roots of this polynomial, because for the value of σ_1 a very good estimation can be given. In case of the 7 parameter similarity transformation the scale value (s) can be estimated by dividing the sum of length in both systems from the centre of gravity, see Albertz and Kreiling (1975). In case of the 9 parameter affine transformation where 3 different scale values (s_1, s_2, s_3 are applied accordingt to the 3 coordinate axis, a good approach for the scale parameters can be given modifying the Albertz-Kreiling expression. Instead of the quotient of the two lenghtes in the centre of gravity system we can use the quotients of the sum of the lengthes in the corresponding coordinate axes directions.

The center of gravity in the two systems (x_s,y_s,z_s) and (X_s,Y_s,Z_s).

$$x_s = \frac{\sum_{i=1}^{3} x_i}{3} \text{ /. numericalValues; } y_s = \frac{\sum_{i=1}^{3} y_i}{3} \text{ /. numericalValues;}$$
$$z_s = \frac{\sum_{i=1}^{3} z_i}{3} \text{ /. numericalValues; } X_s = \frac{\sum_{i=1}^{3} X_i}{3} \text{ /. numericalValues;}$$
$$Y_s = \frac{\sum_{i=1}^{3} Y_i}{3} \text{ /. numericalValues; } Z_s = \frac{\sum_{i=1}^{3} Z_i}{3} \text{ /. numericalValues;}$$

The estimated scale parameter according to Albertz and Kreimlig in the Helmert similarity transformation

$$\text{spriori} = \frac{\sum_{i=1}^{3} \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2}}{\sum_{i=1}^{3} \sqrt{(X_i - X_s)^2 + (Y_i - Y_s)^2 + (Z_i - Z_s)^2}}$$

The estimated scale parameters according to the modified Albertz-Kreimlig expression for the 9 parameter transformation

$$s_{1,\text{priori}} = \frac{\sum_{i=1}^{3} \sqrt{(x_i - x_s)^2}}{\sum_{i=1}^{3} \sqrt{(X_i - X_s)^2}},\ s_{2,\text{priori}} = \frac{\sum_{i=1}^{3} \sqrt{(y_i - y_s)^2}}{\sum_{i=1}^{3} \sqrt{(Y_i - Y_s)^2}},\ s_{3,\text{priori}} = \frac{\sum_{i=1}^{3} \sqrt{(z_i - z_s)^2}}{\sum_{i=1}^{3} \sqrt{(Z_i - Z_s)^2}},$$

and $\sigma_i = 1/s_i$.

$$\text{mxpriori} = \frac{\sum_{i=1}^{3} \text{Abs}[x_i - x_s]}{\sum_{i=1}^{3} \text{Abs}[X_i - X_s]} \text{ /. numericalValues;}$$
$$\text{mypriori} = \frac{\sum_{i=1}^{3} \text{Abs}[y_i - y_s]}{\sum_{i=1}^{3} \text{Abs}[Y_i - Y_s]} \text{ /. numericalValues; mzpriori} = \frac{\sum_{i=1}^{3} \text{Abs}[z_i - z_s]}{\sum_{i=1}^{3} \text{Abs}[Z_i - Z_s]} \text{ /. numericalValues;}$$

```
σ1priori = 1 / mxpriori; SetPrecision[σ1priori, 10]
```

```
1.000001248
```

Therefore Newton - Raphson method can be easily employed

```
solσ1 = FindRoot[SetPrecision[uv1Ns == 0, 16], {σ1, σ1priori}, WorkingPrecision → 16]
```

```
{σ1 → 1.000005408167503}
```

4.2 Accelerated Dixon resultant - Lewis EDF method

The univariate polynomial for σ_1 also can be computed employing the accelerated Dixon resultant by the Early Discovery Factors (EDF) algorithm, which was suggested and implemented in the computer algebra system *Fermat* by Lewis (2007 a, b). Using this method one can get the result via direct elimination in one step in the following factored form in less than two seconds of CPU time

$$\prod_{i=1}^{5} \varphi_i(\sigma_1)^{K_i}$$

where $\varphi_i(\sigma_1)$ are irreducible polynomials with low degree, but their powers,
K_i are big positive integer numbers, so expanding this expression would result in millions of terms!

Consequently, it suffices to use K_i= 1, for i = 1,...,5, namely

$$\prod_{i=1}^{5} \varphi_i(\sigma_1)$$

as the determinant of the Dixon matrix for the resultant. These polynomials are the following

```
φ11 = y13 * z23 - y23 * z13;

φ21 = x13^2 * y23 * z23 * σ1^2 - x13 * x23 * y13 * z23 * σ1^2 -
   x13 * x23 * y23 * z13 * σ1^2 + x23^2 * y13 * z13 * σ1^2 - Z13^2 * y23 * z23 -
   Y13^2 * y23 * z23 - X13^2 * y23 * z23 + Z13 * Z23 * y13 * z23 + Y13 * Y23 * y13 * z23 +
   X13 * X23 * y13 * z23 + Z13 * Z23 * y23 * z13 + Y13 * Y23 * y23 * z13 +
   X13 * X23 * y23 * z13 - Z23^2 * y13 * z13 - Y23^2 * y13 * z13 - X23^2 * y13 * z13;

φ31 = x13 * y23 * σ1 - x23 * y13 * σ1 + X13 * y23 - X23 * y13;

φ41 = Z13 * x13 * x23 * z23 * 1^σ1^2 - Z23 * x13^2 * z23 * σ1^2 -
   Z13 * x23^2 * z13 * σ1^2 + Z23 * x13 * x23 * z13 * σ1^2 + X13 * Z13 * x23 * z23 * σ1 -
   2 * X13 * Z23 * x13 * z23 * σ1 + X23 * Z13 * x13 * z23 * σ1 + X13 * Z23 * x23 * z13 * σ1 -
   2 * X23 * Z13 * x23 * z13 * σ1 + X23 * Z23 * x13 * z13 * σ1 - X13^2 * Z23 * z23 +
   X13 * X23 * Z13 * z23 + X13 * X23 * Z23 * z13 - X23^2 * Z13 * z13;

φ51 = Z13 * z23 - Z23 * z13;
```

Let us compute the roots of these factors

```
Map[NSolve[SetPrecision[# /. newVarsA /. numericalValues, 16], σ1 , WorkingPrecision → 16] &,
 {φ11, φ21, φ31, φ41, φ51}]
```

```
{{}, {{σ1 → -1.000005408163603}, {σ1 → 1.000005408163603}},
 {{σ1 → -0.999996997830469}}, {{σ1 → -0.999999999986342}, {σ1 → 0.3141834844679981}}, {}}
```

We have here two positive solutions, but the only one is realistic, see modified Albertz and Kreiling estimation,

$\sigma_1 = 1.000005408163603$

The numerical solution with global minimization,

$\sigma_1 = 1.0000054081636032$

The result computed using Dixon-KSY method is,

$\sigma_1 \to 1.000005408167503$

It means that Dixon - EDF method provides a bit more precise solution.

The result for σ_1 in symbolic form, considering the positive root of $\varphi_2 = 0$ is

```
solσ1 = (Solve[φ21 == 0, σ1][[2]] // Simplify)
```

$$\Big\{\sigma_1 \to \Big(\sqrt{}\,\big(X23^2\, y13\, z13 + y13\, Y23^2\, z13 + X13^2\, y23\, z23 + Y13^2\, y23\, z23 + y23\, Z13^2\, z23 - X13\, X23\, (y23\, z13 + y13\, z23) - Y13\, Y23\, (y23\, z13 + y13\, z23) - y23\, z13\, Z13\, Z23 - y13\, Z13\, z23\, Z23 + y13\, z13\, Z23^2\big)\Big) \Big/ \Big(\sqrt{(x23\, y13 - x13\, y23)\, (x23\, z13 - x13\, z23)}\Big)\Big\}$$

Similarly we can get simple explicite form for σ_2 and σ_3, too.

For σ_2 we get

```
φ12 = x13 * z23 - x23 * z13;
```

```
φ22 = x13 * y13 * y23 * z23 * σ2^2 - x23 * y13^2 * z23 * σ2^2 -
   x13 * y23^2 * z13 * σ2^2 + x23 * y13 * y23 * z13 * σ2^2 + Z13^2 * x23 * z23 +
   Y13^2 * x23 * z23 + X13^2 * x23 * z23 - Z13 * Z23 * x13 * z23 - Y13 * Y23 * x13 * z23 -
   X13 * X23 * x13 * z23 - Z13 * Z23 * x23 * z13 - Y13 * Y23 * x23 * z13 -
   X13 * X23 * x23 * z13 + Z23^2 * x13 * z13 + Y23^2 * x13 * z13 + X23^2 * x13 * z13;

φ32 = x13 * y23 * σ2 - x23 * y13 * σ2 - Y13 * x23 + Y23 * x13;

φ42 = Z13 * y13 * y23 * z23 * σ2^2 - Z23 * y13^2 * z23 * σ2^2 -
   Z13 * y23^2 * z13 * σ2^2 + Z23 * y13 * y23 * z13 * σ2^2 + Y13 * Z13 * y23 * z23 * σ2 -
   2 * Y13 * Z23 * y13 * z23 * σ2 + Y23 * Z13 * y13 * z23 * σ2 + Y13 * Z23 * y23 * z13 * σ2 -
   2 * Y23 * Z13 * y23 * z13 * σ2 + Y23 * Z23 * y13 * z13 * σ2 - Y13^2 * Z23 * z23 +
   Y13 * Y23 * Z13 * z23 + Y13 * Y23 * Z23 * z13 - Y23^2 * Z13 * z13;

φ52 = Z13 * z23 - Z23 * z13;
```

Let us compute the roots of these factors

```
Map[NSolve[SetPrecision[# /. newVarsA /. numericalValues, 16], σ2 , WorkingPrecision → 16] &,
 {φ12, φ22, φ32, φ42, φ52}]
```

$\{\{\}, \{\{\sigma_2 \to -0.9999978437466494\}, \{\sigma_2 \to 0.9999978437466494\}\}, \{\{\sigma_2 \to -0.999997284021702\}\},$
$\{\{\sigma_2 \to -0.9999976 - 0. \times 10^{-8}\, i\}, \{\sigma_2 \to -0.9999976 + 0. \times 10^{-8}\, i\}\}, \{\}\}$

The only positive real solution is

$\sigma_2 \to 0.9999978437466494$

The numerical solution with global minimization

$\sigma_2 \to 0.9999978437466492$

The result computed using Dixon-KSY method is

$\sigma_2 \to 0.9999978437461572$

It means that Dixon - EDF method provides a bit more precise solution again

The result for σ_2 in symbolic form, considering the positive root of $\varphi_2 = 0$ is

```
solσ2 = (Solve[φ2 == 0, σ2][[2]] // Simplify)
```

$$\left\{\sigma_2 \to \left(\sqrt{\left(-X13^2\, x23\, z23 + X13\, X23\, (x23\, z13 + x13\, z23) + x23\, \left(Y13\, Y23\, z13 - Y13^2\, z23 - Z13^2\, z23 + z13\, Z13\, Z23\right) - x13\, \left(X23^2\, z13 + Y23^2\, z13 - Y13\, Y23\, z23 - Z13\, z23\, Z23 + z13\, Z23^2\right)\right)}\right) \Big/ \left(\sqrt{-(x23\, y13 - x13\, y23)\, (-y23\, z13 + y13\, z23)}\right)\right\}$$

For σ_3 we get only four factors

```
φ13 = y13 * z23 * σ3 - y23 * z13 * σ3 - Z13 * y23 + Z23 * y13;

φ23 = x13 * y23 - x23 * y13;

φ33 = x13 * z23 * σ3 - x23 * z13 * σ3 - Z13 * x23 + Z23 * x13;

φ43 = x13 * y13 * z23^2 * σ3^2 - x13 * y23 * z13 * z23 * σ3^2 - x23 * y13 * z13 * z23 * σ3^2 +
x23 * y23 * z13^2 * σ3^2 - Z13^2 * x23 * y23 - Y13^2 * x23 * y23 -
   X13^2 * x23 * y23 + Z13 * Z23 * x13 * y23 + Y13 * Y23 * x13 * y23 +
   X13 * X23 * x13 * y23 +  Z13 * Z23 * x23 * y13 + Y13 * Y23 * x23 * y13 +
   X13 * X23 * x23 * y13 - Z23^2 * x13 * y13 - Y23^2 * x13 * y13 - X23^2 * x13 * y13;
```

Let us compute the roots of these factors

```
Map[NSolve[SetPrecision[# /. newVarsA /. numericalValues, 16], σ3 , WorkingPrecision → 16] &,
 {φ13, φ23, φ33, φ43}]
```

$\{\{\{\sigma_3 \to -1.000000929208230\}\}, \{\}, \{\{\sigma_3 \to -0.999995431410408\}\}, \{\{\sigma_3 \to -0.9999892916567598\}, \{\sigma_3 \to 0.9999892916567598\}\}\}$

The only positive solution is

$\sigma_3 \to 0.9999892916567598$

The numerical solution with global minimization

$\sigma_3 \to 0.9999892916567604$

The result computed using Dixon-KSY method is

$\sigma_3 \to 0.9999892916621562$

It means that Dixon - EDF method provides a bit more precise solution again.

The result for σ_3 in symbolic form, considering the positive root of $\varphi_4 = 0$ is

```
solσ3 = (Solve[φ43 == 0, σ3][[2]] // Simplify)
```

$$\left\{\sigma_3 \to \left(\sqrt{\left(X13^2\, x23\, y23 - X13\, X23\, (x23\, y13 + x13\, y23) + x23\left(Y13^2\, y23 - y13\, Y13\, Y23 + y23\, Z13^2 - y13\, Z13\, Z23\right) + x13\left(X23^2\, y13 - Y13\, y23\, Y23 + y13\, Y23^2 - y23\, Z13\, Z23 + y13\, Z23^2\right)\right)}\right) \Big/ \left(\sqrt{(x23\, z13 - x13\, z23)\,(y23\, z13 - y13\, z23)}\right)\right\}$$

Remark : The result of Dixon-EDF method is not only faster and more elegant but also a bit more precise than that of the iterative Dixon- KSY method. Although, one should check the solutions of all polynomials with degree 1 and 2 in order to choose the proper result!

4.3 Reduced Groebner basis

The same result can be computed wia reduced Grobner basis, namely

```
gbσ1 = GroebnerBasis[{g1, g2, g3, g4, g5, g6},
   σ1, {a, b, c, σ2, σ3}, MonomialOrder → EliminationOrder]
```

$$\{-X23^2\, y13\, z13 + X13\, X23\, y23\, z13 + Y13\, y23\, Y23\, z13 - y13\, Y23^2\, z13 + X13\, X23\, y13\, z23 - X13^2\, y23\, z23 - Y13^2\, y23\, z23 + y13\, Y13\, Y23\, z23 - y23\, Z13^2\, z23 + y23\, z13\, Z13\, Z23 + y13\, Z13\, z23\, Z23 - y13\, z13\, Z23^2 + x23^2\, y13\, z13\, \sigma_1^2 - x13\, x23\, y23\, z13\, \sigma_1^2 - x13\, x23\, y13\, z23\, \sigma_1^2 + x13^2\, y23\, z23\, \sigma_1^2\}$$

```
gbσ2 = GroebnerBasis[{g1, g2, g3, g4, g5, g6},
   σ2, {a, b, c, σ1, σ3}, MonomialOrder → EliminationOrder]
```

$$\{-X13\, x23\, X23\, z13 + x13\, X23^2\, z13 - x23\, Y13\, Y23\, z13 + x13\, Y23^2\, z13 + X13^2\, x23\, z23 - x13\, X13\, X23\, z23 + x23\, Y13^2\, z23 - x13\, Y13\, Y23\, z23 + x23\, Z13^2\, z23 - x23\, z13\, Z13\, Z23 - x13\, Z13\, z23\, Z23 + x13\, z13\, Z23^2 + x23\, y13\, y23\, z13\, \sigma_2^2 - x13\, y23^2\, z13\, \sigma_2^2 - x23\, y13^2\, z23\, \sigma_2^2 + x13\, y13\, y23\, z23\, \sigma_2^2\}$$

```
gbσ3 = GroebnerBasis[{g1, g2, g3, g4, g5, g6},
  σ3, {a, b, c, σ1, σ2}, MonomialOrder → EliminationOrder]
```

$\{X13\ x23\ X23\ y13 - x13\ X23^2\ y13 - X13^2\ x23\ y23 + x13\ X13\ X23\ y23 - x23\ Y13^2\ y23 + x23\ y13\ Y13\ Y23 + x13\ Y13\ y23\ Y23 - x13\ y13\ Y23^2 - x23\ y23\ Z13^2 + x23\ y13\ Z13\ Z23 + x13\ y23\ Z13\ Z23 - x13\ y13\ Z23^2 + x23\ y23\ z13^2\ \sigma_3^2 - x23\ y13\ z13\ z23\ \sigma_3^2 - x13\ y23\ z13\ z23\ \sigma_3^2 + x13\ y13\ z23^2\ \sigma_3^2\}$

These polynomials are the same as the polynomials selected from the result of the Dixon - EDF algorithm

```
gbσ1 - φ21 // Simplify
```

{0}

```
gbσ2 - φ22 // Simplify
```

{0}

```
gbσ3 - φ43 // Simplify
```

{0}

4.4 Symbolic expressions for *a, b* and *c*

Further unknown parameters can be easily computed from the system of the reduced equations, g_1,...,g_6.

Let us express *a* from the equations g_5 and g_6,

```
drba = DixonResultant[{g5, g6}, {b}, {s1}]
```

$a\ X23\ Y13 - a\ X13\ Y23 + X23\ Z13 - X13\ Z23 + a\ x23\ Y13\ \sigma_1 - a\ x13\ Y23\ \sigma_1 + x23\ Z13\ \sigma_1 - x13\ Z23\ \sigma_1 + a\ X23\ y13\ \sigma_2 - a\ X13\ y23\ \sigma_2 + a\ x23\ y13\ \sigma_1\ \sigma_2 - a\ x13\ y23\ \sigma_1\ \sigma_2 - X23\ z13\ \sigma_3 + X13\ z23\ \sigma_3 - x23\ z13\ \sigma_1\ \sigma_3 + x13\ z23\ \sigma_1\ \sigma_3$

```
{grba} = GroebnerBasis[{g5, g6}, {a, b}, {b}]
```

$\{-a\ X23\ Y13 + a\ X13\ Y23 - X23\ Z13 + X13\ Z23 - a\ x23\ Y13\ \sigma_1 + a\ x13\ Y23\ \sigma_1 - x23\ Z13\ \sigma_1 + x13\ Z23\ \sigma_1 - a\ X23\ y13\ \sigma_2 + a\ X13\ y23\ \sigma_2 - a\ x23\ y13\ \sigma_1\ \sigma_2 + a\ x13\ y23\ \sigma_1\ \sigma_2 + X23\ z13\ \sigma_3 - X13\ z23\ \sigma_3 + x23\ z13\ \sigma_1\ \sigma_3 - x13\ z23\ \sigma_1\ \sigma_3\}$

Then a can be computed form

```
sola = Solve[grba == 0, a] // Simplify // Flatten
```

$$\left\{a \to \frac{X23\ Z13 - X13\ Z23 + (-X23\ z13 + X13\ z23)\ \sigma_3 + \sigma_1\ (x23\ Z13 - x13\ Z23 + (-x23\ z13 + x13\ z23)\ \sigma_3)}{-X23\ Y13 + X13\ Y23 + (-X23\ y13 + X13\ y23)\ \sigma_2 + \sigma_1\ (-x23\ Y13 + x13\ Y23 + (-x23\ y13 + x13\ y23)\ \sigma_2)}\right\}$$

The parameter *b* is given by one of the two equations, let say from g_5,

```
solb = Solve[g5 == 0, b] // Simplify // Flatten
```

$$\left\{b \to \frac{a\ Y13 + Z13 + a\ y13\ \sigma_2 - z13\ \sigma_3}{X13 + x13\ \sigma_1}\right\}$$

and parameter *c*

```
solc = Solve[g1 == 0, c] // Simplify // Flatten
```

$$\left\{c \to \frac{X13 + b\ Z13 - x13\ \sigma_1 + b\ z13\ \sigma_3}{Y13 + y13\ \sigma_2}\right\}$$

4.5 Symbolic expressions of X_0, Y_0 and Z_0.

The translation parameters can be similarly computed, but now from the original system of equations, f_i.

```
{grbX0} = GroebnerBasis[{f1, f2, f3}, {X0, Y0, Z0}, {Y0, Z0}]
```

$$\{X_1 + a^2 X_1 - b^2 X_1 - c^2 X_1 + 2\,a\,b\,Y_1 - 2\,c\,Y_1 + 2\,b\,Z_1 + 2\,a\,c\,Z_1 - x_1\,\sigma_1 - a^2\,x_1\,\sigma_1 - b^2\,x_1\,\sigma_1 - c^2\,x_1\,\sigma_1 + X_0\,\sigma_1 + a^2\,X_0\,\sigma_1 + b^2\,X_0\,\sigma_1 + c^2\,X_0\,\sigma_1\}$$

```
solX0 = Solve[grbX0 == 0, X0] // Simplify // Flatten
```

$$\left\{X_0 \to \frac{(-1 - a^2 + b^2 + c^2)\,X_1 + (-2\,a\,b + 2\,c)\,Y_1 - 2\,b\,Z_1 - 2\,a\,c\,Z_1 + x_1\,\sigma_1 + a^2\,x_1\,\sigma_1 + b^2\,x_1\,\sigma_1 + c^2\,x_1\,\sigma_1}{(1 + a^2 + b^2 + c^2)\,\sigma_1}\right\}$$

similarly we get

```
{grbY0} = GroebnerBasis[{f1, f2, f3}, {X0, Y0, Z0}, {X0, Z0}]
```

$$\{2\,a\,b\,X_1 + 2\,c\,X_1 + Y_1 - a^2\,Y_1 + b^2\,Y_1 - c^2\,Y_1 - 2\,a\,Z_1 + 2\,b\,c\,Z_1 - y_1\,\sigma_2 - a^2\,y_1\,\sigma_2 - b^2\,y_1\,\sigma_2 - c^2\,y_1\,\sigma_2 + Y_0\,\sigma_2 + a^2\,Y_0\,\sigma_2 + b^2\,Y_0\,\sigma_2 + c^2\,Y_0\,\sigma_2\}$$

```
solY0 = Solve[grbY0 == 0, Y0] // Simplify // Flatten
```

$$\left\{Y_0 \to \frac{-2\,(a\,b + c)\,X_1 + (-1 + a^2 - b^2 + c^2)\,Y_1 + 2\,a\,Z_1 - 2\,b\,c\,Z_1 + y_1\,\sigma_2 + a^2\,y_1\,\sigma_2 + b^2\,y_1\,\sigma_2 + c^2\,y_1\,\sigma_2}{(1 + a^2 + b^2 + c^2)\,\sigma_2}\right\}$$

and

```
solZ0 = Solve[f1 == 0, Z0] // Simplify // Flatten
```

$$\left\{Z_0 \to \frac{X_1 - c\,Y_1 + b\,Z_1 - x_1\,\sigma_1 + X_0\,\sigma_1 - c\,y_1\,\sigma_2 + c\,Y_0\,\sigma_2 + b\,z_1\,\sigma_3}{b\,\sigma_3}\right\}$$

The numerical result using these symbolic results is the following

```
{tσ1, σ1} = SetPrecision[(σ1 /. solσ1 /. newVarsA /. numericalValues), 16] // Chop // Timing
```

$\{2.84495 \times 10^{-15}, 1.0000054081636034\}$

```
{tσ2, σ2} = SetPrecision[(σ2 /. solσ2 /. newVarsA /. numericalValues), 16] // Chop // Timing
```

{0., 0.9999978437466494}

```
{tσ3, σ3} = SetPrecision[(σ3 /. solσ3 /. newVarsA /. numericalValues), 16] // Chop // Timing
```

{0., 0.9999892916567600}

```
sσ = {σ1 -> σ1, σ2 -> σ2, σ3 -> σ3}
```

$\{\sigma_1 \to 1.0000054081636034, \sigma_2 \to 0.9999978437466494, \sigma_3 \to 0.9999892916567600\}$

```
{ta, aa} = SetPrecision[(a /. sola /. newVarsA /. sσ /. numericalValues), 16] // Chop // Timing
```

$\{3.38618 \times 10^{-15}, 9.269080332103910 \times 10^{-7}\}$

```
{tb, bb} =
 SetPrecision[(b /. solb /. newVarsA /. sσ /. {a -> aa} /. numericalValues), 16] // Chop //
  Timing
```

$\{0., -4.946616458808342 \times 10^{-6}\}$

```
{tc, cc} =
 SetPrecision[(c /. solc /. newVarsA /. sσ /. {b -> bb} /. numericalValues), 16] // Chop //
  Timing
```

$\{1.69309\times10^{-15}, -3.243922351302802\times10^{-7}\}$

```
sabc = {a -> aa, b -> bb, c -> cc}
```

$\{a\to 9.269080332103910\times10^{-7}, b\to -4.946616458808342\times10^{-6}, c\to -3.243922351302802\times10^{-7}\}$

```
{tX0, X0} =
 SetPrecision[(X0 /. solX0 /. newVarsA /. sσ /. sabc /. numericalValues), 16] // Chop // Timing
```

{0., 124.2834144988798}

```
{tY0, Y0} =
 SetPrecision[(Y0 /. solY0 /. newVarsA /. sσ /. sabc /. numericalValues), 16] // Chop // Timing
```

{0., -62.08451159187030}

```
{tZ0, Z0} =
 SetPrecision[(Z0 /. solZ0 /. newVarsA /. sσ /. sabc /. {X0 -> X0, Y0 -> Y0} /. numericalValues),
    16] // Chop // Timing
```

{0., -102.3123880824574}

Rotation angles in seconds :

```
CardanS[R /. sabc]
```

{-0.3823763498040887, 2.040625894931123, 0.1338195115851545}

Scale parameters :

```
SetPrecision[{1 / σ1, 1 / σ2, 1 / σ3} /. sσ, 10]
```

{0.9999945919, 1.000002156, 1.000010708}

The total running time of the evaluation of the analytic expressions developed by computer algebra

```
Apply[Plus, {tσ1, tσ2, tσ3, ta, tb, tc, tX0, tY0, tZ0}]
```

7.92422×10^{-15}

Let us summarize our results in case of the 3 points problem:

Table 2.
Results in case of 3 points Problem

Method	*Running Time* (sec)
Numerical Groebner basis & eigensystem method	0.11
Global minimization with genetic algorithm	1.17
Linear Homotopy	0.03
Analytic solution computed via computer algebra	~ 0.00

Remarks : Keep in mind, that built − in functions are quicker than functions written in *Mathematica* own language,
which is an interpreter, but built − in functions have a big overhead,
sometimes their code can be many hundred pages, therefore they are not always the faster!

Considering *Table* 2. it is clear that for 3 points problem the best choice is the application of the analytic form. The advantage of the analytic expression is not only the short computation time. The running time of the homotopy and any iterative algorithm may depend the actual values of the coefficients, while in case of the evaluation of analytic expressions the running time practically is constant. In addition these analytic expressions can be directly transformed into C which is faster with roughly two magnitudes than any interpreter languages.
For example, in case σ_1

$$\text{CForm}\left[\frac{1}{\sqrt{(x23\,y13 - x13\,y23)\,(x23\,z13 - x13\,z23)}}\left(\sqrt{\left(X23^2\,y13\,z13 + y13\,Y23^2\,z13 + X13^2\,y23\,z23 + Y13^2\,y23\,z23 + y23\,Z13^2\,z23 - X13\,X23\,(y23\,z13 + y13\,z23) - Y13\,Y23\,(y23\,z13 + y13\,z23) - y23\,z13\,Z13\,Z23 - y13\,Z13\,z23\,Z23 + y13\,z13\,Z23^2\right)}\right)\right]$$

```
Sqrt(Power(X23,2)*y13*z13 + y13*Power(Y23,2)*z13 + Power(X13,2)*y23*z23 + Power(Y13,2)*y2
     Y13*Y23*(y23*z13 + y13*z23) - y23*z13*Z13*Z23 - y13*Z13*z23*Z23 + y13*z13*Power(Z23,
```

We should emphasize again, there is no guarantee, which polynomials give the proper result for a σ_i. Therefore in case of any index *i*, all corresponding terms should be investigated in order to choose the proper polynomial!

Conclusions

For the 3 point problem the computer algebra method, namely the accelerated Dixon resultant with the technique of Early Discovery Factors as well as reduced Groebner basis provide a very simple, elegant symbolic solution. It is enormously better than using the Dixon Resultant Application developed for *Mathematica*, which could only succeed by a long and tedious step-by step reduction and even then, the answer given is difficult to work with it.
Although, numerical methods without initial guess values, like linear homotopy and numerical Groebner basis with eigensystem method are also very efficient. The main advantages of the symbolic solution originated from its iteration-free feature, are the very short computation time and the independence on value of the actual numerical data.
The solution of the 9 parameter affine coordinate transformation problem for 3 points can provide a good inital value for the *N* point coordinate transformation. Solving this problem, it is not only the choice of the employed method, but the computation of a good initial guess value on the bases of the geometry of the properly selected three points are also important (collinear triplets should be avoided).

Acknowledgement

The author are indebted to Daniel Lichtblau (Wolfram Research) for his valueable remarks and suggestions, especially calling up their attention to the proper implementation of **`NSolve`** as well as to the application of the reduced Groebner basis.

References

Albertz J., Kreiling W. (1975): Photogrammetric Guide, Herbert Wichmann Verlag, Karlsruhe, pp. 58-60.

Awange J.L. (2002).: Gröbner bases, multipolynomial resultants and Gauss-Jacobi combinatorial algorithms - adjustment of nonlinear GPS/LPS observations, Dissertation (D93), Geodätisches Institute der Universität Stuttgart

Awange J.L, Grafarend E.W. (2003).: Closed form solution of the overdetermined nonlinear 7 parameter datum transformation. Allgemeine VermessungsNachrichten (AVN) 110, pp.130-148.

Awange J.L. and Grafarend E.W. (2005): Solving Algebraic Computational Problems in Geodesy and Geoinformatics, Springer, Berlin

Bancroft, S. (1985).: An algebraic solution of the GPS equations, IEEE Transaction on Aerospace and Electronic Systems AES-21, pp. 56-59.

Barsi A. (2001): Performing coordinate transfornations by artificial neural network, Allgemeine VermessungsNachrichten 108, pp. 134-137.

Cai, J., Grafarend, E. W. (2004): Systematische Analyse der Transformation zwischen Gauß-Krüger-Koordinaten/DHDN und UTM-Koordinaten/ETRS89 angewandt auf Baden-Wurttemberg, Geodatische Woche 2004 - Stuttgart 12.10.-15.10. poster presentation

Drexler, F.J. (1977): Eine Methode zur Berechnung sämtlicher Lösungen von Ploynomgleichungssystemen, Numer. Math. 29. pp.45-58.

Fröhlich H, Bröker G. (2003): Trafox version 2.1. - 3d-kartesicsche Helmert - Transformation, http://www.koordinatentransformation.de/data/trafox.pdf

Garcia C.B. and Zangwill, W.I. (1979): Determinung all solutions to certain systems of nonlinear equations, Math. Operations Res. 4. pp. 1-14

Golub, G. H., Pereyra, V. (1973): The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate, SIAM J. Num. Anal. 10, pp. 413-432.

Grafarend E. and Schaffrin, B. (1993): Ausglechungsrechnung in Linearen Modellen, B.I. Wissenschaftsverlag, Mannheim

Grafarend E. and Kampmann G. (1996): $C_{10}(3)$: The tenparameter conformal group as a datum transformation in threedimensional Euclidean space, eitschrift für Vermessungswesen 121,pp. 68-77.

Kapur D, Saxena T. and Yang L. (1994) : Algebraic and geometric reasoning using Dixon resultants. In : ACM ISSAC 94. Oxford, England, July 1994, pp. 99 - 107.

Kleusberg A. (1994): Die direkte Lösung des räumlichen Hyperbelschnitts. Zeitschrift für Vermessungswesen, 119, pp. 188 - 192.

Kleusberg A. (2003): Analytical GPS navigation solution. In Grafarend EW., Krumm FW. and Schwarze VS. (eds) Geodesy -the Challenge of the 3rd Millennium, Springer, Heidelberg, pp. 93 - 96.

Lewis, R.H. and Bridgett, S.(2003): Conic tangency equations and Apollonius problems in biochemistry and pharmacology, Mathematics and Computers in Simulation 61. pp. 101-114

Lewis, R. H. (2007 a): Heuristics to accelerate the Dixon resultant, Mathematics and Computers in Simulation /In Press/, Available online 11 April 2007

Lewis, R. H. (2007 b): Computer algebra system *Fermat*, http://home.bway.net/lewis/

Lichtenegger H. (1995): Eine direkte Lösung des räumlichen Bogenschnitts., Osterreichische Zeitschrift fur Vermessung und Geoinformation 83, pp. 224 - 226.

Mathes A. (2002): EasyTrans Pro-Edition, Professionelle Koordinatentransformation für Navigation, Vermessung und GIS, ISBN 978-3-87907-367-2, CD-ROM mit Benutzerhandbuch

Nakos G. and Williams R. (2002): A fast algorithm implemented in Mathematica provides one-step elimination of a block of unknowns from a system of polynomial equations, http://wolfram.com/infocenter/Articles/2597

Palancz, B. (2008) Introduction to Linear Homotopy, *MathSource* (submitted)

Papp E., Szucs L. (2005): Transformation Methodes of the Traditional and Satellite Based Networks (in Hungarian with English abstract), Geomatikai Kozlemenyek VIII. 85-92.

Singer P., Strobel D., Hordt R., Bahndorf J., Linkwitz K. (1993): Direkte Losung des raumlichen Bogenschnitts. Zeitschrift für Vermessungswesen, 124, S. 295 - 297.

Späth,H. (2004): A numerical method for determining the spatial Helmert transformation in case of different scale factors, Zeitschrift für Geodäsie, Geoinformation und Landmanagement 129, pp. 255-257.

Volgyesi L, Toth Gy, Varga J. (1996): Conversion between Hungarian Map Projection Systems. Periodica Polytechnica Civ.Eng., Vo1.40, Nr.1, pp. 73-83.

Watson , G.A. (2006): Computing Helmert transformations, Journal of Computational and Applied Mathematics 197, pp. 387-395.

Wolfrum, O. (1992): Merging terrestrial and satellite networks by a ten-parameter transformation model, Manuscripta Geodetica 17, pp. 210-214.

Zaletnyik P. (2004): Coordinate transformation with neural networks and with polynomials in Hungary , International Symposium on Modern technologies, education, and professional practice in geodesy and related fields, 4-5 November 2004, Sofia, Bulgaria, pp. 471-479.