Appendix A

Tracking Magick, Resolution of Beam Telescope and More

This appendix explains track finding and track fitting methods used in analysis of beam test data. Beam telescope properties are studied in greater detail and methods used to extract them are discussed. Alignment procedure for inclined detector is described.

Please note that the term *detector* is used as an abbreviation for silicon strip detector.

1.1 Tracking

Method used for fitting the tracks is a bit of an overkill as detector orientation is allowed to be any member of vector space generated by O(3) operators. But since such representation was needed to fully explore the inclined detector the choice seemed quite natural. What follows is a description of path leading from clusters to space points and then on to tracks.

1.1.1 Representation of marks and points

After doing archaeological excavations on raw data one usually ends up with a set of clusters. For silicon–strip detectors these are just intervals of strips with signal above certain cuts (clustering procedure is discussed in Sec:2.3). Given this information one proceeds using one or the other algorithm for position prediction and comes up with a single number. Making an agreement with oneself as to where the origin of the *local coordinate system* (LCS) is (say, on the middle strip) and knowing the pitch of the detector one gets a measurement of track crossing point u in LCS. For convenience let us define perpendicular coordinate in detector plane as v and coordinate in direction normal to detector plane as w. With a strip detector we can't really say much about v coordinate (but that it is somewhere within the size of the detector) and the best one can do is to set it to value corresponding to the middle of the detector. w coordinate of the measurement is undoubtedly zero.

Trying to get this measurement useful for track fitting, measurements from all detectors must be transformed to *tracker's coordinate system* (TCS). This means that we must know positions and rotation angles of each detector. Which, with certain uncertainty, we do. Let us return to LCS and discuss errors. Error on u can be measured for given position prediction algorithm and is typically some fraction of detector's pitch. Error on v can not be defined reasonably in terms of a Gaussian error as distribution of residuals in this direction is basically flat (following track position distribution). It is best to leave it infinite and let detectors with different orientation of strips pinpoint track position in this direction. Error on w is the trickiest one as one is tempted to set it to zero. But since measurement vector will be transformed to TCS the wisest choice is to set it to precision with which detector position in w direction is determined. Here is formalistic compression of all the talking, giving definition of measured quantities in LCS:

$$\vec{R} = (u, 0, 0) \tag{A.1}$$

$$\vec{S} = (\sigma_u, \infty, \sigma_w). \tag{A.2}$$

These will be refered to as *marks* as opposed to their representation in TCS where they will be called *points*.

Transforming a mark into a point one first notices that vector representation for errors is inappropriate. It has to be a matrix as errors get correlated when subjected to rotations. For later convenience we define it as:

$$W_{\mathrm{LCS}} = \begin{pmatrix} \frac{1}{\sigma_u^2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_w^2} \end{pmatrix}. \tag{A.3}$$

I won't bother myself by writing down transformation rules ... perhaps it is only worth mentioning that I used `homogeneous coordinates` where position vector is extended by one dimension (called scale) and it enables one to write translation and rotation in LCS using a 4×4 matrix. This 'trick' is usually used in computer graphics and interested parties are kindly invited to consult the literature ([38] or just search the www).

1.1.2 Track fitting formulas

Fitting a straight line is to fitting as `hello world` example is to programming, so this little expose serves more for documentation purposes. Besides ... I want to argue how three degrees of freedom from 3D representation of a point are squished into a single one.

I will present formulas defining least squares fitting procedure with Gaussian errors. See one of the classics (e.g., [39]) or excellent series on applied fitting theory by Paul Avery ([40]) for details. This just wants to put some context to further manipulations.

Problem definition can be split into few points:

1. For start ... we have n measured values denoted by y_l.

2. We would like to 'summarize' this data in terms of a mathematical model with m parameters α_i, $m < n$. At each of the data points l our model makes a prediction $f_l(\alpha)$ (dropping an index implies a vector).

3. With this, we can define difference between measured data and model prediction at given values of parameters α:

$$\chi^2 = \sum_l \frac{\left(y_l - f_l(\alpha)\right)^2}{\sigma_l^2}, \tag{A.4}$$

where σ_l is the error (expected to be Gaussian and uncorrelated) of measurement l. Of course ... we want the best set of parameters, that is parameters that minimize χ^2:

$$\frac{\partial \chi^2}{\partial \alpha} = 0. \tag{A.5}$$

4. One expects to have certain clue about initial values of parameters $\widetilde{\alpha}$ ('seed' parameters) and that expansion to first order in $\alpha - \widetilde{\alpha}$ will take us on a way to the minimum. That is ... we approximate $f_l(\alpha)$ by:

$$f_l(\alpha) = f_l(\widetilde{\alpha}) + \sum_i (\alpha_i - \widetilde{\alpha}_i) \frac{\partial f_l(\alpha)}{\partial \alpha_i}\Bigg|_{\widetilde{\alpha}} \equiv f_l(\widetilde{\alpha}) + \sum_i A_{li}\eta_i \tag{A.6}$$

$$f(\alpha) = f(\widetilde{\alpha}) + \mathbf{A}\eta. \tag{A.7}$$

First we defined derivatives of model function with respect to parameters evaluated at their initial values ($\mathbf{A}_{n\times m}$) and distance in parameter space (η_n) from seed to improved values. Then we introduced matrix notation.

5. Rewriting χ^2 with this expansion yields:

$$\chi^2 = \sum_l \frac{\left(y_l - f_l(\widetilde{\alpha}) - \sum_i A_{li}\eta_i\right)^2}{\sigma_l^2}. \tag{A.8}$$

Let me define $\Delta \mathbf{y} = y_l - f_l(\widetilde{\alpha})$ which is difference between measurements and values predicted by seed parameters. More interesting quantity is a residual vector $\varepsilon = \Delta\mathbf{y} - \mathbf{A}\eta$ which is difference of measurements and final predicted values. To put χ^2 expression to matrix form, we need to define the weight matrix (this is just fitological term for inverse of correlation matrix) ... remember, we are still talking uncorrelated errors ... so this is really simple: $\mathbf{W}_{n\times n} = \mathrm{diag}(1/\sigma_1^2 \ldots\ 1/\sigma_\mathrm{i}^2 \ldots\ 1/\sigma_\mathrm{n}^2)$. This brings us to:

$$\chi^2 = (\Delta\mathbf{y} - \mathbf{A}\eta)^T \mathbf{W}\,(\Delta\mathbf{y} - \mathbf{A}\eta) = \varepsilon^T \mathbf{W}\,\varepsilon. \tag{A.9}$$

6. Doing the partial derivations with respect to α_m, we get m equations:

$$\mathbf{A}^T\mathbf{W}\,(\Delta\mathbf{y} - \mathbf{A}\eta) = 0. \tag{A.10}$$

Defining $\mathbf{V_A} = \left(\mathbf{A}^T\mathbf{W}\,\mathbf{A}\right)^{-1}$, we get solution for η:

$$\eta = \mathbf{V_A}\mathbf{A}^T\mathbf{W}\,\Delta\mathbf{y} \tag{A.11}$$

By using $\alpha = \widetilde{\alpha} + \eta$, we get the improved set of parameters. As we only made the linear approximation, several iterations of the above procedure are needed when model function is non-linear with respect to model parameters..

Not to be too verbose ... we're still interested in errors on fitted parameters and their correlation. One of the surprises of the least square method is that it always yields correlation matrix of the form : $\mathbf{V}_\alpha = \mathbf{V}_\eta = \mathbf{V}_A$.

7. Fitting with correlated errors basically means inverting the covariance matrix $\mathbf{V_y} = \langle \delta y^T \delta y \rangle$. This can get messy, especially with near–singular correlation matrices. There are ways around it, though not a general one.

1.1.3 Application to beam telescope track reconstruction

Since the beam energy is 300 GeV (90% pions, 10% muons) and longitudinal dimension of the telescope is 20 cm, track is modeled as a straight line without accounting for multiple scattering. Four parameters were used:

- x and y coordinate of track intersection with $z = 0$ plane, packed in a vector $\vec{p}$:

$$\vec{p} = \begin{pmatrix} x & y & 0 \end{pmatrix}$$

- directions q_x and q_y of track with q_z fixed to 1:

$$\vec{q} = \begin{pmatrix} q_x & q_y & 1 \end{pmatrix}.$$

This yields parametrization of a straight line $\vec{p} + t\vec{q} = \vec{x}(t)$ and n measurements $\vec{r_i}$ on i-th detector plane defined by a normal $\vec{z_i}$ (i going from 0 to $N_{\text{det}} - 1$). In order to calculate the distance from measured intersection to predicted one ($\vec{\varepsilon_i}$), we require for it to lie in the detector plane: $\vec{z_i} \cdot (\vec{r_i} - \vec{x}(t)) = 0$, giving:

$$t_i = \frac{\vec{z_i} \cdot (\vec{r_i} - \vec{p})}{\vec{z_i} \cdot \vec{q}} \tag{A.12}$$

$$\vec{\varepsilon_i} = \vec{r_i} - \vec{p} - t_i\vec{q}. \tag{A.13}$$

Or in language of model function $\vec{f_i}(\alpha)$:

$$\vec{f_i}(\vec{p}, \vec{q}) = \vec{p} + t_i\vec{q}. \tag{A.14}$$

Index i is counting the detector planes. So far nothing was said about direction in which actual measurement was done. Let's keep it this way a bit longer. Introducing a new index $j \in \{1, 2, 3\}$ to subscript x, y and z components of vectors and defining the overall index as $k = 3i + j$, we get close to 'linear' structure required by fitting procedure. For example $z_{1,2}$ stands for y component of normal of second detector. With this, we can calculate partial derivatives needed to form matrix $\mathbf{A}$:

$$\frac{\partial}{\partial p_m}\varepsilon_{i,j} = -\delta_{jm} + \frac{z_{i,m}}{\vec{z_i} \cdot \vec{q}}\, q_j, \tag{A.15}$$

$$\frac{\partial}{\partial q_m}\varepsilon_{i,j} = -\vec{z_i} \cdot (\vec{r_i} - \vec{p})\, \frac{\delta_{jm}(\vec{z_i} \cdot \vec{q}) - q_j z_{i,m}}{(\vec{z_i} \cdot \vec{q})^2}. \tag{A.16}$$

Index m is taking values 1 and 2 (as third components of $\vec{p}$ and $\vec{q}$ are held fixed) and δ_{jm} is the Kronecker delta symbol.

Now we have to build the weight matrix. This is done by simply transforming $\mathbf{W}_{\text{LCS}}$ with the transformation matrix $\mathbf{T}$ from LCS to TCS: $\mathbf{W}_{\text{TCS}} = \mathbf{T}\,\mathbf{W}_{\text{LCS}}\,\mathbf{T}^{-1}$ and building a block–diagonal matrix out of these transforms. Two points are crucial. First, the v direction 'measurement', which is in fact just detector position, does not contribute anything to χ^2 because its error is set to be infinite. Second, error on w is projected out by requiring the track intersection to lie in plane of the detector. General treatment was necessary to be able to cope with detectors under arbitrary orientations.

1.2 Resolution of beam telescope

Basic description of beam telescope is given in Sec:2.2.1. Material shown there is just about enough to convince a careless reader that error on predicted track position at $z = 0\,\mu$m is $\delta u_t = 2.5\,\mu$m. Here you can find the full story.

1.2.1 Alignment

Alignment of telescope detectors was carried out in $z =$ const plane. This amounts to moving the detector in direction perpendicular to strips and rotating it around the z axis. See [41]:Ch:4 for more details. In our case *eta* method was used all along for determination of hit positions.

Additionally, whole tracker was moved and rotated so that its z axis coincided with average track direction and that average x and y of reconstructed tracks were zero.

Results of alignment are implicitly visible on plots showing residual distributions of telescope planes.

1.2.2 Estimation of resolution of telescope detectors

It is a bit of a wrong idea to use residual distributions of detectors used in track fitting as a measure of their resolution. The proper thing to do is to exclude them, one by one, from the track fit. Fig:A.1 compares both cases. So we take resolution from non–fitted planes as basic input. One is tempted to forget that this is in fact a convolution of errors given by the detector and intrinsic error of the tracker:

$$\sigma_{i\,m}^2 = \sigma_i^2 + \sigma_t^2(z_i). \tag{A.17}$$

i is index of the plane, m stands for 'measured' and t for 'tracker'. The dependence of tracker's resolution on z coordinate is implied by functional notation. Resolution is better for planes 1-4 than for planes 5-8. This is explained by different number of interpolation strips (see Tab:2.1).

Assuming that planes with the same number of interpolation strips have the same resolution, one can argue that when one of the planes is not used in the fit, resolution of the other one (on its 'side' of the tracker) dominates the tracker's resolution. That is, e.g.,

$$\sigma_{5\,m}^2 = \sigma_5^2 + \sigma_8^2 = 2\sigma_5^2 \tag{A.18}$$

and all other combinations. From this follows

$$\sigma_i = \frac{\sigma_{i\,m}}{\sqrt{2}}. \tag{A.19}$$

Using this prescription we come up with values $\sigma_{1-4} = 2.28\,\mu$m and $\sigma_{5-8} = 5.48\,\mu$m.

These were used as first guesses for monte carlo simulation in which marks in detectors were smeared by this amount from values obtained by calculating the track intersection. The same resolution was used to calculate χ^2 contribution of point in question. Residuals obtained from MC runs were in agreement with the true ones but distribution of χ^2 per degree of freedom for real data had mean at 0.72. Honestly ... and quite obviously ... resolutions obtained in this way are worse than actual ones.

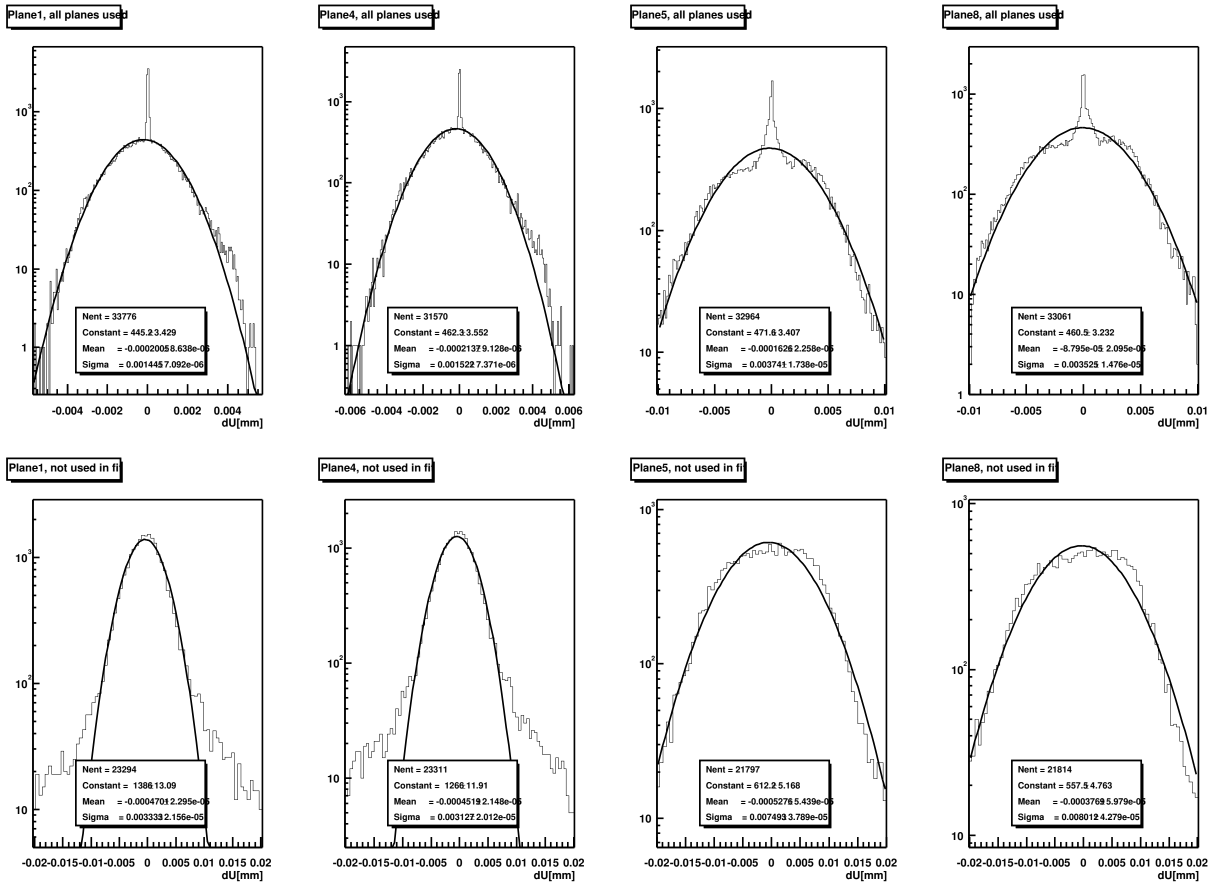

Figure A.1: Residuals for Planes 1, 4, 5 and 8 when they have been all used in the fit (top line) and when they were omitted (one at the time) from the fit (botom line). Note that scales of x-axes are different.

This leads us to the conclusion that we swallowed the tracker's resolution into the resolution of the detector plane. This excess can be attributed to telescope misalignment in directions where explicit alignment hasn't been performed. But we have another cross-check. When both planes are used in the fit, we are in fact looking at the distribution of $u = x_5 - (x_5 + x_8)/2 = (x_5 - x_8)/2$, where x_i is the position measurement of i-th plane. Resolution of this quantity, let's name it $\sigma_{i\,u}$, can be written:

$$\sigma_{i\,u}^2 = \frac{\sigma_i^2}{2}. \tag{A.20}$$

This way we obtain resolutions $\sigma_{1-4} = 2.1\,\mu$m and $\sigma_{5-8} = 5.15\,\mu$m. These were then used for calculation of the χ^2 of the track fit. Residuals obtained by MC simulation are shown in Fig:A.2 and basic parameters of fit quality on real data in Fig:A.3. Additionally, distributions of fitted track parameters (position and direction of track) is given in Fig:A.4.

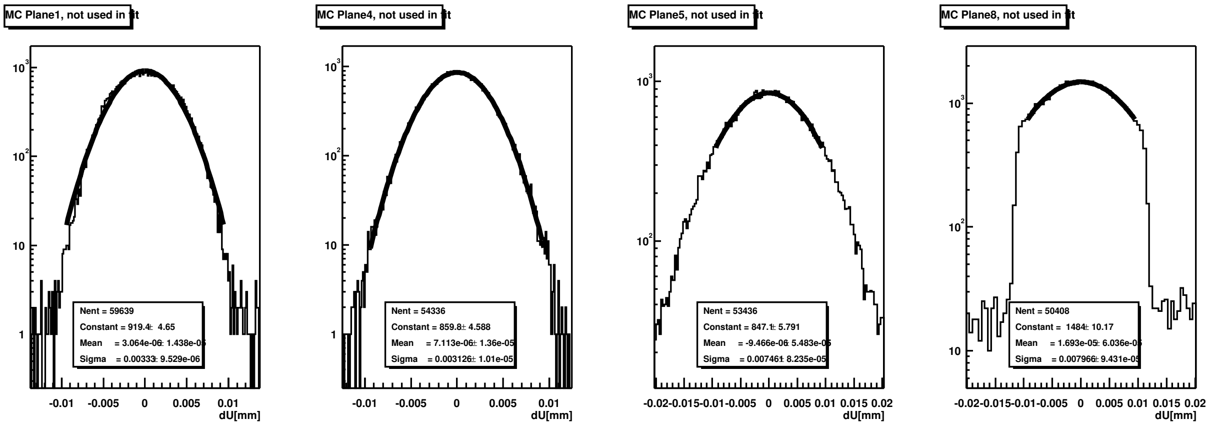

Figure A.2: Residuals for Planes 1, 4, 5 and 8 with monte-carlo generated tracks, where plane in question was not used in track fit.

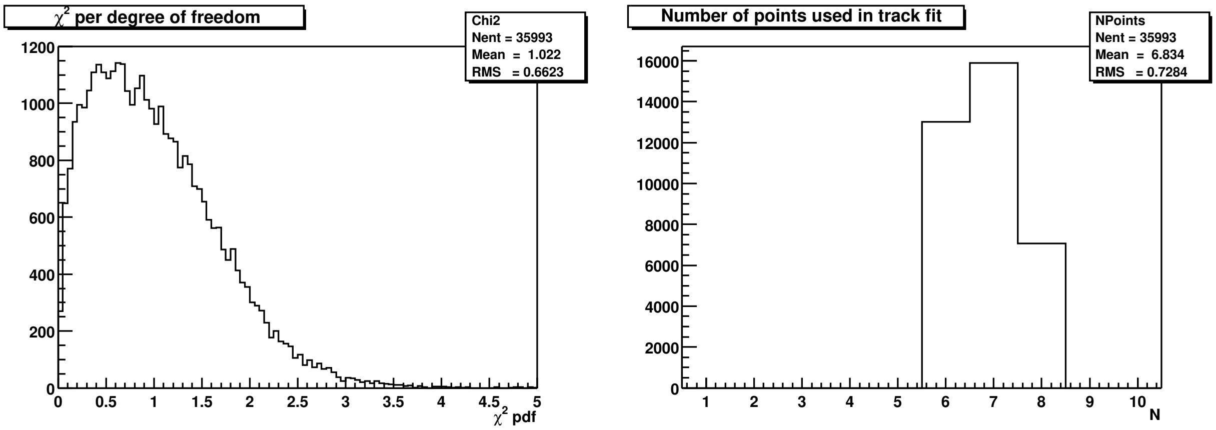

Figure A.3: Track fit quality for real data. Note that fitologically we expect $<\chi^2/\text{pdf}>= 1$ and $\mathsf{rms}(\chi^2/\text{pdf}) = 1/\sqrt{2} \approx 0.71$.

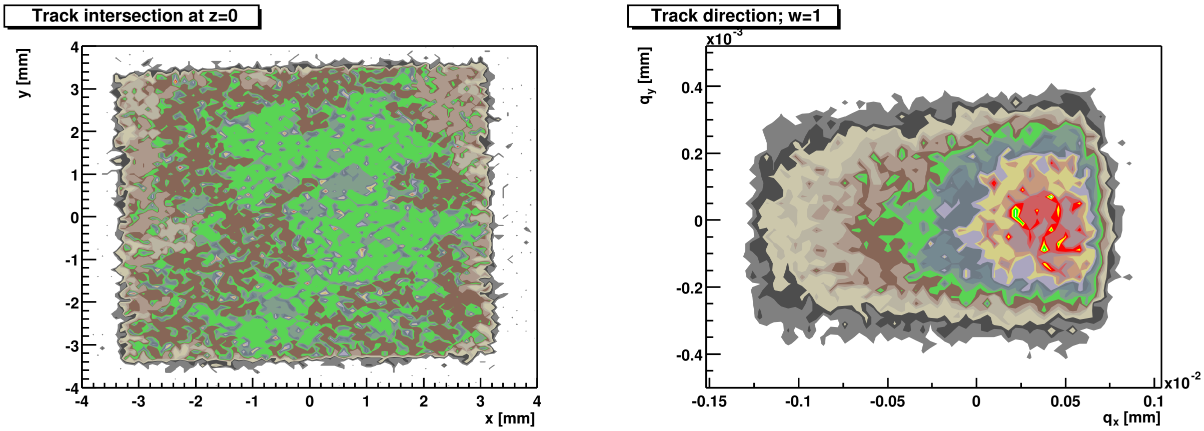

Figure A.4: Distributions of fitted parameters.

1.3 Alignment of the inclined detector

For detector under test a special alignment procedure was adapted as it had to be aligned with respect to six degrees of freedom. Positioning of the detector was described by position vector of center of the detector and three cardanian angles giving its orientation in local reference frame (beware ... different set of cardanian angles can represent the same physical orientation). Problem of alignment is basically equivalent to minimization of chi–square function for the detector for a set of events. Due to finite sample of events and its statistical fluctuations the problem is somewhat ill–defined and using traditional gradient methods tends not to converge but on a narrow area around the true minimum.

To find the way through the scattered landscape of the chi–square function a cross between simplex minimization and Metropolis algorithm was used (see [39], chapter 10) leading to the following algorithm:

1. Position detector 'by hand' to provide solid starting conditions. Especially bad choice of angular variables can lead to unnecessary long minimization.

2. Iterate through the available events. Choose events where there is no ambiguity (single track, single cluster in the detector) and cluster position reconstruction is good (i.e., for low inclinations use only two–strip clusters). Number of good events will be refered to as N_e.

3. For sizes of inital simplex use mean values of residual distributions for u, v coordinates, width of the detector for w coordinate and hard–coded value of $1°$ for angular variables.

4. Enter the minimization loop with inital value of temperature set to one tenth of inital χ^2. Set fractional convergence to $1/\sqrt{N_e}$. It is enough to set maximum number of iterations per given temperature to $\sqrt{N_e}$. Decrease the temperature in N_s steps (50 to 100 steps showed adequate for $N_e \approx 3000$) with annealing scheme $T_i = T_0(1 - i/N_s)^{1.2}$.

One has to admit that the method used was a bit of an overkill. But it is always like this with numerics: one either uses the most optimal algorithm and spends an eternity tuning it to work well in given case or chooses to use a more general one that safely bypasses the obstacles. Size of the problem and amount of data that has to be processed are the guidelines for deciding which path to take in a particular case. We only had to perform alignment six times (once for each run) and getting to the bottom of all pitfalls hidden in alignment was not our intention in presented work.