TESLA Report 2004-08

Tomographic Reconstruction of Transverse and Longitudinal Phase Space Distributions using the Maximum Entropy Algorithm

J. J. Scheins

Deutsches Elektronen-Synchrotron DESY
10. Mai 2004

Abstract

The Maximum Entropy criterion can be utilised for tomographic reconstruction of two-dimensional distributions from a set of one-dimensional projection profiles. In terms of entropy the reconstructed distributions represent the most probable solution which reproduces the experimental input data. Therefore the Maximum Entropy (MENT) algorithm is especially suited in case of incompleteness of information. Reconstruction artefacts due to the lack of information are minimised.

The MENT algorithm has been adapted for the reconstruction of the transverse and longitudinal phase space distributions at particle accelerators allowing a determination of crucial beam parameters. Only one-dimensinonal intensity profiles for different beam transfer matrices have to be measured. These profiles can be obtained from diverse standard monitoring devices, e.g. wirescanners or optical transition radiation stations. Besides the application for transverse phase space tomography only dealing with linear transformations, the algorithm has also been adopted for tomographic reconstruction of the longitudinal phase space where non-linear transformations have to be taken into account.

Technically, the MENT algorithm has been implemented and verified in the physics analysis framework of ROOT [1]. The applied user interface provides a generalised, very simple access to the full functionality of the algorithm and its results.

scheins@mail.desy.de *j.scheins@fz-juelich.de*

1 Introduction

In general computer tomography comprises algorithms to solve special kinds of inversion problems where measured distributions contain pieces of information of an underlying higher-dimensional initial distribution. Typically each measured distribution evolves from a projection of the initial distribution to a lower number of dimensions after applying a certain transformation. Here tomography algorithms combine all available pieces of information, i.e. measured projection profi les, to reconstruct the underlying, initial distribution.

Figure 1 illustrates a simple situation where a two-dimensional initial distribution (top) is rotated by the angle θ in the frame of reference (middle) and then projected onto the x_θ axis of the new coordinate system (bottom) by calculating the line integral over the orthogonal y_θ coordinate. The mathematical relation between coordinates in the initial and the rotated coordinate

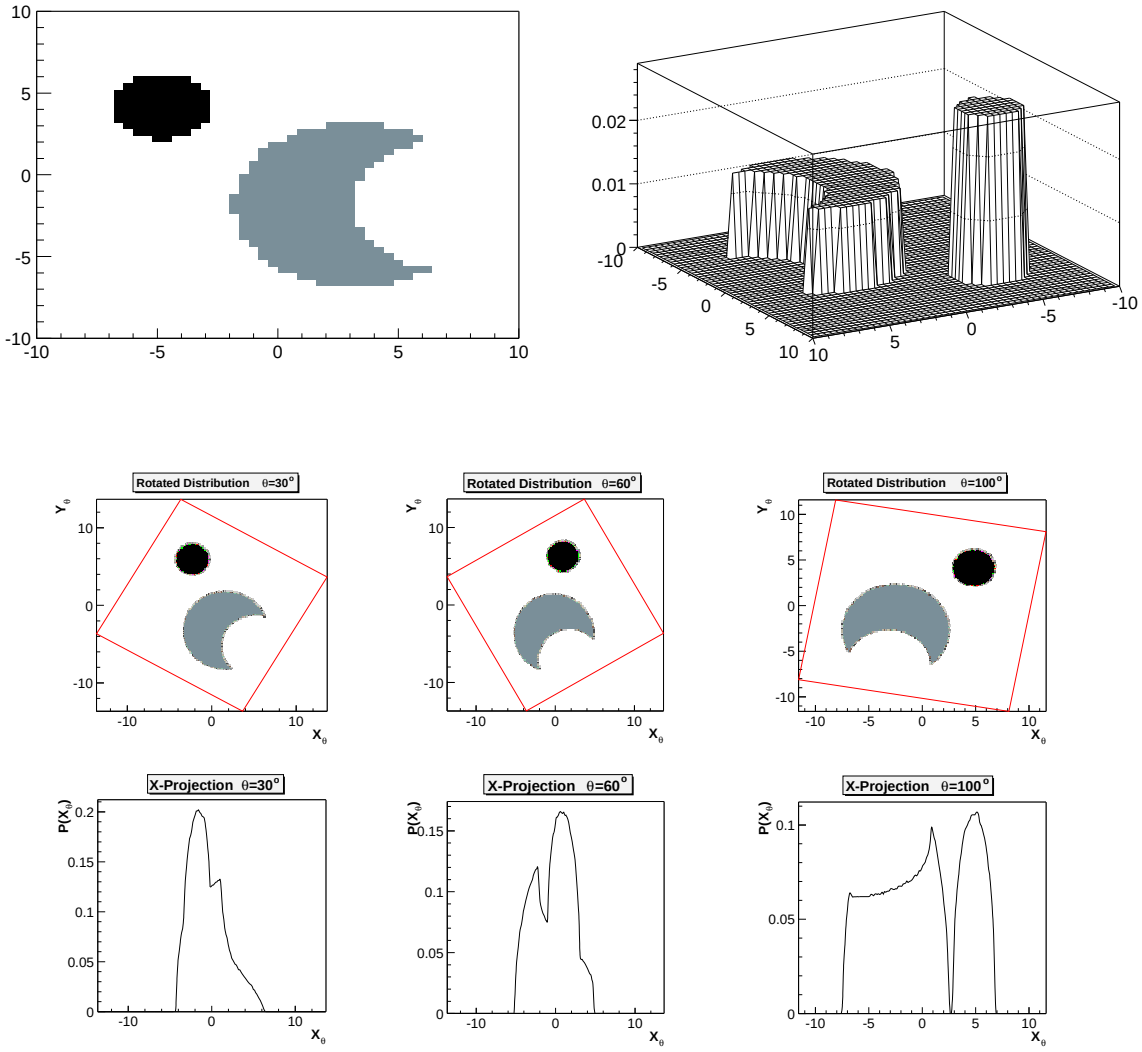

Figure 1: Example of a two-dimensional initial distribution (top), resulting rotated distributions (middle) for rotation angles $\theta = 30^o, 60^o, 100^o$ and applied parallel projection of the rotated distributions onto the x_θ axis (bottom).

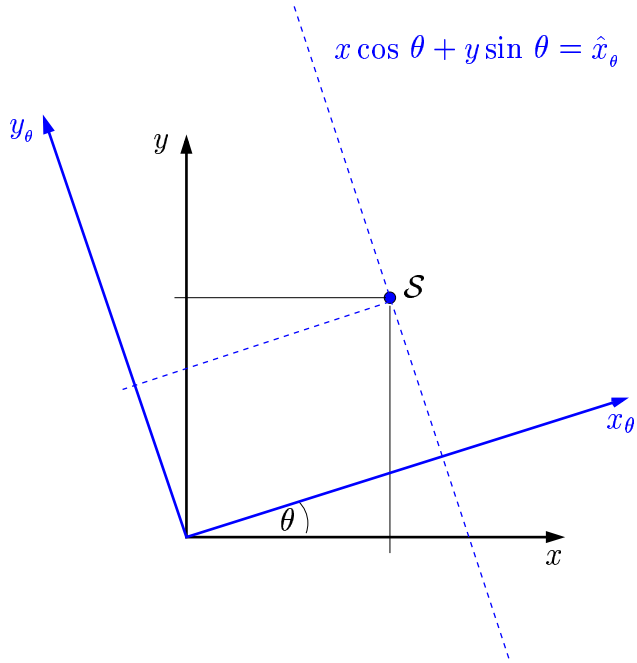

Figure 2: Transformation of coordinates due to rotation of the coordinate system around the origin with rotation angle θ; lines parallel to the y_θ axis are defined by eq. (3).

system can be seen in fi gure 2. For rotations around the origin each point $\mathcal{S} = (x, y)$ transforms linearly as

$$\begin{pmatrix} x_\theta \\ y_\theta \end{pmatrix} = M_\theta \begin{pmatrix} x \\ y \end{pmatrix} \tag{1}$$

with the rotation matrix M_θ

$$M_\theta = M(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \quad \text{fullfi lling} \quad M^{\mathrm{T}} = M^{-1} \ . \tag{2}$$

The rotation M_θ conserves the area of the initial distribution due to $\det(M_\theta) = 1$. Lines parallel to the y_θ axis of the rotated coordinate system are defi ned by

$$x\cos\theta + y\sin\theta = \hat{x}_\theta \tag{3}$$

for any value of $\hat{x}_\theta$. The continuous function $P(\theta, x_\theta)$

$$\begin{aligned} P(\theta, x_\theta) &= \int_{x\cos\theta + y\sin\theta = x_\theta} \rho_{\mathrm{ini}}(x, y)\ dy_\theta \\ &= \int_{-\infty}^{\infty} \rho_{\mathrm{ini}}(x_\theta\cos\theta - y_\theta\sin\theta\,,\ x_\theta\sin\theta + y_\theta\cos\theta)\ dy_\theta \end{aligned} \tag{4}$$

depending on the x_θ coordinate and the projection angle θ with $\theta = [-\pi, \pi]$ is known as Radon transform [2] of the initial function $\rho_{\mathrm{ini}}(x, y)$. The Radon transform represents a parallel projection which is formulated as line integral. Using the Dirac δ-function the integral can be extented over the full x-y plane

$$P(\theta, x_\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \rho_{\mathrm{ini}}(x, y)\,\delta(x\cos\theta + y\sin\theta - x_\theta)\,dx\,dy \ . \tag{5}$$

Now, the following theorem by Radon guarantees the existence of a solution for the inversion of the Radon transform:

The value of a two-dimensional function at an arbitrary point is uniquely determined by the integrals along the lines of all directions passing the point.

Obviously the initial function $\rho_{\text{ini}}(x, y)$ and the the Radon transform $P(\theta, x_\theta)$ contain the same degree of information, i.e. mathematically a 'complete' set of one-dimensional projections is suffi cient to reconstruct the two-dimensional underlying distribution.

To derive $\rho(x, y)$ from the Radon transform $P(\theta, x_\theta)$ fi rst the Fourier transform has to be calculated using eq. (4)

$$\begin{aligned}\widehat{P}(\theta,\omega) &= \int_{-\infty}^{\infty} P(\theta, x_\theta)\; e^{(-2\pi i\,\omega\, x_\theta)} dx_\theta \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \rho(x_\theta\cos\theta - y_\theta\sin\theta\,,\; x_\theta\sin\theta + y_\theta\cos\theta)\; e^{(-2\pi i\,\omega\, x_\theta)}\; dx_\theta\, dy_\theta \quad .\end{aligned} \tag{6}$$

With eq. (1) the Fourier transform can be rewritten as

$$\begin{aligned}\widehat{P}(\theta,\omega) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \rho(x,\; y)\; e^{(-2\pi i\,\omega\,(x\cos\theta + y\sin\theta))}\; dx\, dy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \rho(x,\; y)\; e^{(-2\pi i\,(x\cdot u + y\cdot v))}\; dx\, dy \equiv \mathcal{F}(u, v)\end{aligned} \tag{7}$$

with the substitutions $u = \omega\cos\theta$ and $v = \omega\sin\theta$. The obtained relation

$$\widehat{P}(\theta,\omega) = \mathcal{F}(\omega\cos\theta,\; \omega\sin\theta) \tag{8}$$

is the well known Fourier Slice theorem [3]. The one-dimensional Fourier transform of the Radon transform $P(\theta, x_\theta)$ at the angle θ is equal to the slice of the two-dimensional Fourier transform along the line through the origin in the u-v system at the same angle. The two-dimensional inverse Fourier transform yields the full reconstruction of $\rho(x, y)$.

$$\begin{aligned}\rho(x,\; y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathcal{F}(u, v)\; e^{(+2\pi i\,(x\cdot u + y\cdot v))}\; du\, dv \\ &= \int_{0}^{\pi}\int_{-\infty}^{\infty} \mathcal{F}(\omega\cos\theta, \omega\sin\theta)\; e^{(+2\pi i\,(x\omega\cos\theta + y\omega\sin\theta))} \begin{vmatrix} \frac{\partial u}{\partial\theta} & \frac{\partial v}{\partial\theta} \\ \frac{\partial u}{\partial\omega} & \frac{\partial v}{\partial\omega} \end{vmatrix} d\omega\, d\theta \\ &= \int_{0}^{\pi}\int_{-\infty}^{\infty} \widehat{P}(\theta,\omega)\; e^{(+2\pi i\,(x\omega\cos\theta + y\omega\sin\theta))}\; \omega\; d\omega\, d\theta\end{aligned} \tag{9}$$

However, in practice, we have to cope with incomplete sets of projections. Usually only a restricted number of projections is available and these projections are mostly measured as

non-continuous, i.e. binned, distributions. Then the above inversion of the Fourier transform to reconstruct the two-dimensional distributions may produce artefacts and falsify the results. In addition imperfect measuring devices produce shifts, tilts, smearing and noise in the measured projections and degrade the accuracy of reconstruction even more.

Therefore a variety of tomographic reconstruction algorithms like Filtered Backprojection [4] or Algebraic Reconstruction Technique [5] have been invented. Each algorithm has specific advantages, but also produces specific artefacts due to the lack of information. The Maximum Entropy Technique represents an interesting alternative. It provides reconstructed distributions with a minimised amount of artefacts. In the following chapters we concentrate on the Maximum Entropy algorithm and describe its functioning, implementation and implications.

2 Maximum Entropy Algorithm

The Maximum Entropy algorithm [6] mathematically bases on the definition of entropy in statistical thermodynamics [7]. It provides a solution of the reconstruction space which becomes most likely in terms of entropy[1] while simultaneously reproducing all entering projection data exactly. The generated solution has to be interpreted as the 'simplest' i.e. most probable possibility to describe the observed data. Consequently the MENT algorithm is well suited in case of incomplete information. In practice the immanent lack of information leads to artefacts in the reconstruction as for example observed for the filtered backprojection [8]. The ansatz of the MENT algorithm avoids artificial structures due to the correlated looser probabilities. It proves to be superior especially in situations where the projection space is not completely covered, i.e. only a few different projections are available. Under these circumstances other tomography algorithms often appear to be insufficient for a reasonable reconstruction.

2.1 Foundation

The following considerations refer to linear tranformations (simple rotations) which have already been introduced in section 1. But in principle the presented algorithm is not limited to rotations. Generalised linear transformations including also shearing of the reconstruction space are feasible. The only requirement is that the area has to be conserved under the applied transformations[2].

2.1.1 Representation of Experimental Data

In contrast to the Radon transform $P(\theta, x_\theta)$ we usually have a relatively small number of acessible projection profiles. Suppose we have measured J profiles characterised by the index

[1] Maximising the entropy directly leads to the most likely solution because the entropy $\mathcal{E}$ is proportional to the logarithm of the probability p, i.e. $\mathcal{E} \sim \log(p)$ [10].

[2] For the application to longitudinal phase space tomography (sec. 5) this requirement is not appropriate. Necessary modifications of the usage of the MENT algorithm are explained in the related section.

$j = 1, 2, ..., J$. Then each profile j results from a certain projection angle θ_j with the associated transformation matrix from eq. (2). In fact, this means a transition $\theta \rightarrow \theta_j$ from the continuous variable θ to a limited number of discrete, specific values θ_j. Furthermore, the measured projection profiles usually exists only as non-continuous, binned distributions where each profile j can have a different grid with different numbers of bins N_j and different, possibly non-equidistant, internal binning. Ignoring all kinds of measurement errors the binned projection data G_{jm} can be formulated due to eq. (4) as[3]

$$\begin{aligned} G_{jm} &= \int_{\hat{x}_{jm}}^{\hat{x}_{j(m+1)}} dx_j \, P_j(x_j) \\ &= \int_{\hat{x}_{jm}}^{\hat{x}_{j(m+1)}} dx_j \int_{-\infty}^{\infty} dy_j \; \rho_{\text{ini}}(x_j \cos\theta_j - y_j \sin\theta_j, \;\; x_j \sin\theta_j + y_j \cos\theta_j) \end{aligned} \tag{10}$$

with $\rho_{\text{ini}}(x, y)$ as the underlying initial distribution function. The additional index m refers to the m^{th} bin of the j^{th} projection profile with $m = 1, 2, ..., N_j$ and the values

$$\hat{x}_{j1} < \hat{x}_{j2} < ... < \hat{x}_{jN_j} < \hat{x}_{j(N_j+1)} \tag{11}$$

represent the bin boundaries of the projection profile j. The arguments in the function $\rho_{\text{ini}}(x, y)$ of eq. (10) simply follow from the rotational matrix $M_j \equiv M(\theta_j)$:

$$\begin{pmatrix} x \\ y \end{pmatrix} = M_j^{-1} \begin{pmatrix} x_j \\ y_j \end{pmatrix} \quad . \tag{12}$$

By introducing the characteristic functions[4]

$$\Gamma_{jm}(x_j) \equiv \begin{cases} 1 & \hat{x}_{jm} \leq x_j < \hat{x}_{j(m+1)} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

for each interval $[\hat{x}_{jm}, \hat{x}_{j(m+1)}[$ eq. (10) can be rewritten as

$$G_{jm} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dx \, dy \;\; \rho_{\text{ini}}(x, y) \; \Gamma_{jm}(x \cos\theta_j + y \sin\theta_j) \tag{14}$$

with an integration over the full parameter space of the initial coordinates. The charateristic function extracts a parallel strip from the full parameter space as illustrated in figure 3.

For any given point $\mathcal{S}$=(x, y) in the initial parameter space with $x_j \equiv x \cos\theta_j + y \sin\theta_j$ satisfying $\hat{x}_{j1} \leq x_j \leq \hat{x}_{j(N_j+1)}$ exactly one bin from projection j can be assigned unambigiuously,

[3]The discretised form of function $P(\theta, x_\theta)$ is inidicated by modified subscripts as follows: $P_j(x_j)$

[4]The intended simple definition does only hold in the corresponding rotated frame of reference, which means the argument x_j of the function has to be understood as $x_j \equiv x \cos\theta_j + y \sin\theta_j$ (see figure 3).

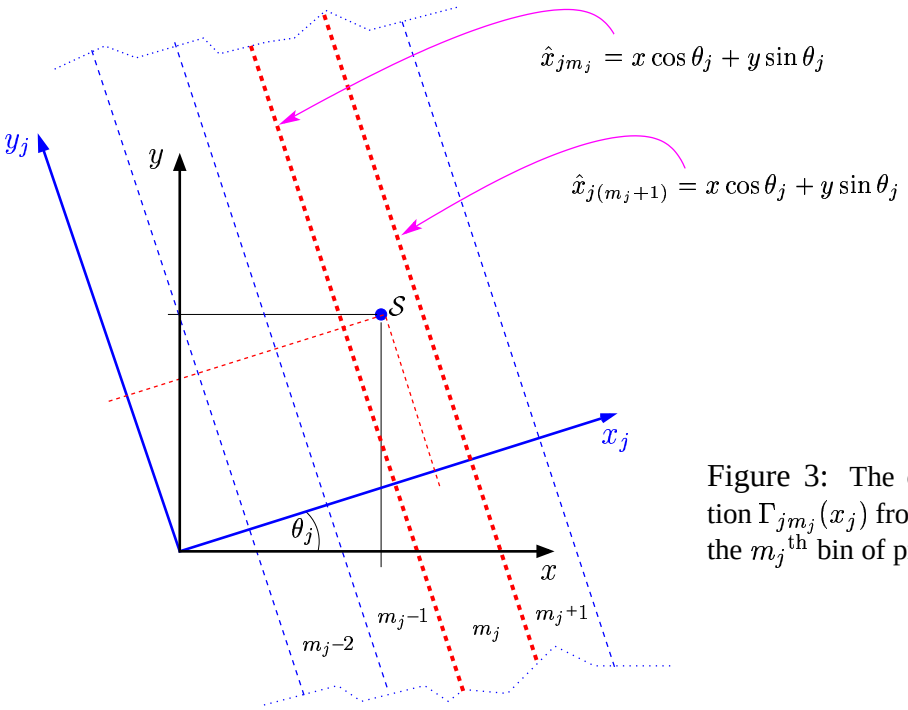

Figure 3: The charateristic function $\Gamma_{jm_j}(x_j)$ from eq. (13) extracts the $m_j{}^{\text{th}}$ bin of projection j.

because only one characteristic function Γ_{jm} appears non-zero (see fig. 3). The assigned bin index m_j is defined by the unique condition

$$\Gamma_{jm_j}(x_j) = 1 \quad \text{and} \quad \Gamma_{jm}(x_j) = 0 \quad \text{with} \quad m \neq m_j \tag{15}$$

and depends only on $\mathcal{S}$. From the properties of the characteristic functions another useful relation can be formulated

$$\sum_{m=1}^{N_j} z_{jm} \cdot \Gamma_{jm}(x_j) = \begin{cases} z_{jm} & \hat{x}_{jm_j} \leq x_j < \hat{x}_{j(m_j+1)} \\ 0 & x_j \notin [\hat{x}_{j1},\ \hat{x}_{j(N_j+1)}] \end{cases} \tag{16}$$

for any set of values z_{jm}. The sum reduces to one specific single value z_{jm} for any chosen point $\mathcal{S}$. Here, the defining condition $\hat{x}_{jm_j} \leq x_j < \hat{x}_{j(m_j+1)}$ provides the index m_j of the bin containing the point $\mathcal{S}$.

Supplementary to eq. (13) it will be convenient to define a global characteristic function to select the full valid range of projection j

$$\Omega_j(x_j) \equiv \sum_{m=1}^{N_j} \Gamma_{jm}(x_j) = \begin{cases} 1 & \hat{x}_{j1} \leq x_j \leq \hat{x}_{j(N_j+1)} \\ 0 & \text{otherwise} \end{cases}. \tag{17}$$

For each projection j it selects the specific area, i.e. parallel strip, in the full two-dimensional parameter space which can possibly contribute to the projection.

2.1.2 Entropy and Probability

Suppose we have a large number N_0 of distinguishable objects distributed over a limited area $\mathcal{A}$ with same probability at each position within $\mathcal{A}$. Intuitively a constant object density distribution should most likely be realised. However, also fluctuations can be observed with certain

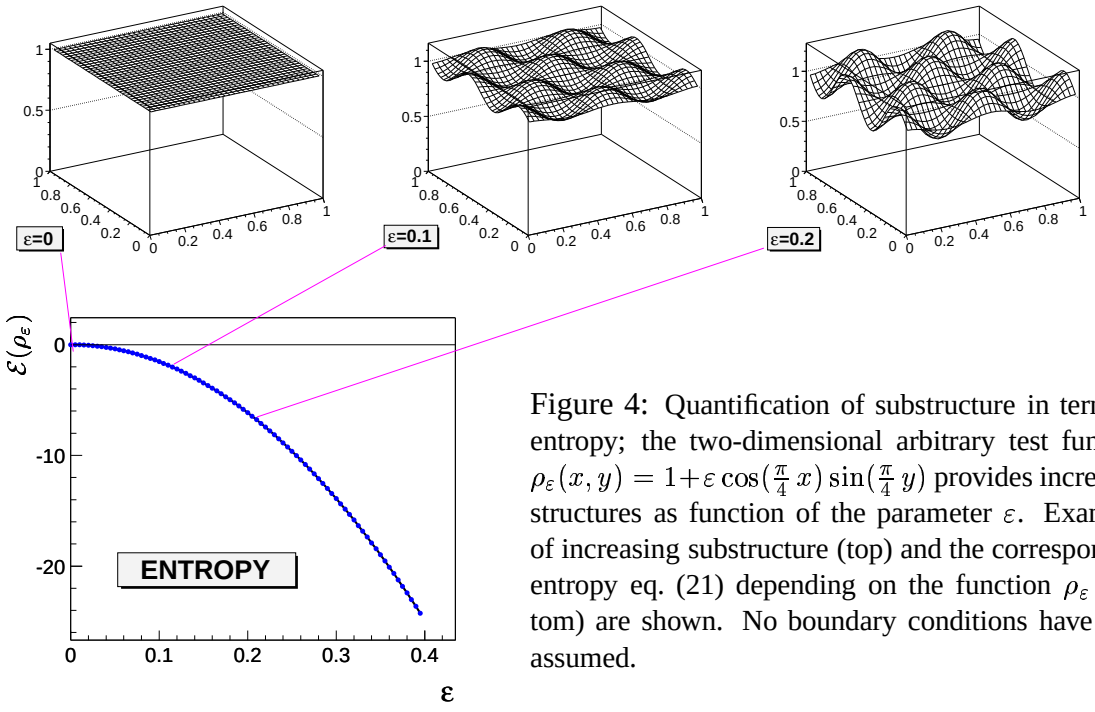

Figure 4: Quantification of substructure in terms of entropy; the two-dimensional arbitrary test function $\rho_\varepsilon(x,y) = 1+\varepsilon\cos(\frac{\pi}{4}x)\sin(\frac{\pi}{4}y)$ provides increasing structures as function of the parameter ε. Examples of increasing substructure (top) and the corresponding entropy eq. (21) depending on the function ρ_ε (bottom) are shown. No boundary conditions have been assumed.

probabilities. But larger fluctuations are expected to be less frequent compared to smaller fluctuations. Let $\rho(x,y)$ be a two-dimensional density function defined on $\mathcal{A}$ which represents a specific pattern[5]. Now we can ask for the probability to observe $\rho(x,y)$ in a sample [10].

Basically the density function is a global, macroscopic description of the underlying object distribution and does not distinguish microscopic details of the arrangement of objects. Due to the assumption of equal probability for each microscopic arrangement the probability $\mathcal{P}_\rho$ to observe $\rho(x,y)$ has to be proportional to the number of possible arrangements $\mathcal{W}_\rho$ which lead to the same pattern. In general $\mathcal{W}_\rho$ can be quantified from combinatorics.

First discretisation simplifies the problem. The area $\mathcal{A}$ is subdivided into K tiny pixels of equal size Δq accompanied by the transition $\rho(x,y) \to \rho_k$. Then the number of objects in each pixel is given by

$$N_k = \rho_k \Delta q \qquad \text{with} \qquad \sum_{k=1}^{K} N_k = N_0 \ . \tag{18}$$

According to the combinatorical theorem, the number of microscopic arrangements $\mathcal{W}_\rho$ to observe a certain pattern ρ is now

$$\mathcal{W}_\rho = \frac{N_0!}{\prod_{k=1}^{K} N_k!} \ . \tag{19}$$

[5]To simplify all considerations we assume $\mathcal{A} \equiv 1$ in the following; therefore relevant area sizes can be ignored in the derived formulae.

Using Stirling's approximation[6] leads to

$$\ln \mathcal{W}_\rho \sim -\sum_k N_k \ln N_k \ . \tag{20}$$

Then going back to the limit of a continuous distribution ($\Delta q \to 0$, $N_0 \to \infty$) provides the entropy equation [10] which quantifies the entropy $\mathcal{E}$ of each probability distribution[7] ρ:

$$\mathcal{E}(\rho) = -\int\int_{\mathcal{A}} dx\, dy \ \ \rho(x,y) \ln[\rho(x,y)] \ . \tag{21}$$

The entropy $\mathcal{E}(\rho)$ is a convenient measure [9] of the value of $\mathcal{W}_\rho$ via

$$\mathcal{E}(\rho) \sim \ln(\mathcal{W}_\rho) \sim \ln(\mathcal{P}_\rho) \ . \tag{22}$$

The function ρ_0 which maximises the entropy has the largest probability to be observed.

In addition entropy can be interpreted in terms of substructure as illustrated in figure 4. The arbitrary test function $\rho_\varepsilon(x,y) = 1 + \varepsilon \cos(\frac{\pi}{4}x) \sin(\frac{\pi}{4}y)$ has been considered where the parameter ε quantifies deviations from a structureless, flat distribution. The figure shows the test function for $\varepsilon = 0.0,\ 0.1,\ 0.2$ which generates increasing structures. The entropy calculated from eq. (21) decreases towards larger values of ε. Thus increasing structure is correlated to decreasing entropy. The distribution with vanishing structures has the largest entropy, i.e. largest probability to be observed. This is realised for the above test function by ε=0.

2.1.3 Maximised Entropy with Boundary Conditions

In order to apply the entropy criterion to tomography also boundary conditions have to be taken into account. We now have to find a function ρ which maximises the entropy defined in eq. (21) while simultaneously reproducing the experimental data G_{jm} from eq. (14). The constrained optimisation problem can be solved by the Lagrange multiplier technique. For each constraint eq. (14) one Lagrange multiplier Λ_{jm} is introduced. The correspondence between the number of constraints and the degrees of freedom also guarantees the uniqueness of the obtained solution.

The extented entropy functional can be written as follows [16]

$$\begin{aligned} \widehat{\mathcal{E}}(\rho, \underline{\Lambda}) = &-\int\int_{\mathcal{A}} dx\, dy \ \ \rho(x,y) \ \ln[\rho(x,y)] \\ &-\sum_{j=1}^{J} \sum_{m=1}^{N_j} \Lambda_{jm} \left[G_{jm} - \int\int_{\mathcal{A}} dx\, dy \ \ \rho(x,y) \ \Gamma_{jm}(x\cos\theta_j + y\sin\theta_j) \right] \end{aligned} \tag{23}$$

The functional derivative of $\widehat{\mathcal{E}}(\rho)$ with respect to $\rho(x,y)$ gives the equation

$$\frac{\partial \widehat{\mathcal{E}}(\rho, \underline{\Lambda})}{\partial \rho} = -\ln[\rho(x,y)] - 1 + \sum_{j=1}^{J} \sum_{m=1}^{N_j} \Lambda_{jm}\, \Gamma_{jm}(x\cos\theta_j + y\sin\theta_j) \equiv 0 \tag{24}$$

[6] $\ln(n!) \approx (n+\frac{1}{2})\ln n - n + \frac{1}{2}\ln(2\pi) \approx n \ln n$

[7] Here, the entropy formalism refers to a naturally continuous function ρ; however its reconstruction results in a discretised formulation as consequence of the binned representation of the measured data.

as necessary condition for an extremum of the entropy functional with respect to the function $\rho(x,y)$. The solution follows directly as

$$\begin{aligned}\rho(x,y) &= \exp\left(-1+\sum_{j=1}^{J}\sum_{m=1}^{N_j}\Lambda_{jm}\,\Gamma_{jm}(x\cos\theta_j+y\sin\theta_j)\right)\\ &= \frac{1}{e}\prod_{j=1}^{J}\prod_{m=1}^{N_j} e^{\Lambda_{jm}\;\Gamma_{jm}(x\cos\theta_j+y\sin\theta_j)}\;.\end{aligned} \tag{25}$$

The products over m can be simplified according to eq. (15)

$$\Pi_j = \prod_{m=1}^{N_j} e^{\Lambda_{jm}\;\Gamma_{jm}(x\cos\theta_j+y\sin\theta_j)} = e^{\Lambda_{jm_j}} \quad \text{for} \quad \mathcal{S}=(x,y)\in\mathcal{A}\;. \tag{26}$$

Due to eq. (16) this expression can be rearranged as a sum

$$\Pi_j = \sum_{m=1}^{N_j} e^{\Lambda_{jm}}\;\Gamma_{jm}(x\cos\theta_j+y\sin\theta_j) \tag{27}$$

and with the definition

$$H_{jm} = e^{-\frac{1}{J}}\,e^{\Lambda_{jm}} \tag{28}$$

we finally get

$$\Pi_j = e^{\frac{1}{J}}\sum_{m=1}^{N_j} H_{jm}\;\Gamma_{jm}(x\cos\theta_j+y\sin\theta_j)\;. \tag{29}$$

Therefore eq. (25) can be rewritten

$$\begin{aligned}\rho(x,y) &= e^{-1}\prod_{j=1}^{J}\Pi_j\\ &= e^{-1}\prod_{j=1}^{J}\left[e^{\frac{1}{J}}\sum_{m=1}^{N_j} H_{jm}\;\Gamma_{jm}(x\cos\theta_j+y\sin\theta_j)\right]\\ &= \prod_{j=1}^{J}\sum_{m=1}^{N_j} H_{jm}\;\Gamma_{jm}(x\cos\theta_j+y\sin\theta_j)\;.\end{aligned} \tag{30}$$

The original constrained variation problem eq. (23) is now converted into the task of finding the values of H_{jm} which reproduce all measured data elements G_{jm} eq. (14). Substitution of eq. (30) in eq. (14) yields an iteration relation for the determination of the coefficients H_{jm}.

$$\begin{aligned}G_{jm} &= \int_{-\infty}^{\infty}dx\int_{-\infty}^{\infty}dy\;\Gamma_{jm}(x\cos\theta_j+y\sin\theta_j)\cdot\left\{\prod_{k=1}^{J}\sum_{l=1}^{N_k}H_{kl}\;\Gamma_{kl}(x\cos\theta_k+y\sin\theta_k)\right\}\\ &= H_{jm}\cdot\\ &\quad\int_{-\infty}^{\infty}dx\int_{-\infty}^{\infty}dy\;\;\Gamma_{jm}(x\cos\theta_j+y\sin\theta_j)\cdot\left\{\prod_{k\neq j}^{J}\sum_{l=1}^{N_k}H_{kl}\;\Gamma_{kl}(x\cos\theta_k+y\sin\theta_k)\right\}\end{aligned} \tag{31}$$

We now have found constraints on the H_{jm}-multipliers:

$$H_{jm} = \frac{G_{jm}}{\iint dx\, dy\ \Gamma_{jm}(x\cos\theta_j + y\sin\theta_j)\left\{\prod_{k\neq j}\ \sum_l H_{kl}\ \Gamma_{kl}(x\cos\theta_k + y\sin\theta_k)\right\}} \quad . \tag{32}$$

Each multiplier H_{jm} has an interdependence to various multipliers H_{kl} which are explicitly not related to the same projection j, i.e. $k \neq j$ in the above expression. Furthermore, H_{jm} is proportional to G_{jm}. Consequently, the H_{jm}-multiplier dominantly determines the corresponding value G_{jm} whereas the impact on the other elements G_{kl} with $k \neq j$ is suppressed due to the small contribution within the sum over many products. So adjusting H_{jm} to G_{jm} leads only to minor numerical modifi cations within the other projections!

The non-linear Gauss-Seidel method is used to fi nd the solution of all elements H_{jm} iteratively. For the fi rst iteration all contributing elements $H_{jm}^{(1)}$ of the solution space are initialised with $H_{jm}^{(1)} \equiv 1$. In case of vanishing input elements $G_{jm} = 0$ the corresponding solution element is set equal to zero $H_{jm}^{(1)} \equiv 0$ according to eq. (32). The elements $H_{jm}^{(i+1)}$ of the next iteration $(i+1)$ results from the recent iteration (i) by applying the following iteration rule

$$H_{jm}^{(i+1)} = \frac{G_{jm}}{G_{jm}^{(i)}}\ H_{jm}^{(i)} \tag{33}$$

with

$$\begin{aligned} G_{jm}^{(i)} = \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy\ \Gamma_{jm}(x\cos\theta_j + y\sin\theta_j)\cdot \\ \left[\prod_{k=1}^{J}\ \sum_{l=1}^{N_k} H_{kl}^{(i+\boldsymbol{\delta(k)})}\ \Gamma_{kl}(x\cos\theta_k + y\sin\theta_k)\right] \quad . \end{aligned} \tag{34}$$

However, the order of update of the coeffi cients $H_{jm}^{(i)}$ within one complete iteration cycle is of importance and influences the result, because $G_{jm}^{(i)}$ depends on the recent values of H_{kl} with $k \neq j$. One complete iteration cycle has to span over all projections. Furthermore each iteration has strictly to be executed in closed sub-iterations on each projection j which means calculating all $G_{jm}^{(i+1)}$ for all values of $m = 1, ..., N_j$ with value j fi xed. The function $\delta(j)$ in eq. (34) defi nes the correct order of coeffi cient update as follows:

$$\delta(j) = \begin{cases} 1 & j \text{ already updated} \\ 0 & \text{otherwise} \end{cases} \quad . \tag{34a}$$

Note, the iteration cycles are indicated by the superscript (i) and in eq. (34) the expression $(i+\delta(k))$ refers to the recent iteration cycle (i) for $\delta = 0$ resp. already to the values of the next iteration cycle $(i+1)$ for $\delta = 1$. Mixing values $H_{kl}^{(i+1)}$ of the next iteration level already in the recent iteration cycle represented by values $H_{kl}^{(i)}$ seems to be an atypical procedure, however this is the only working solution to get consistent results and guarantees the conservation of the volume integral of the reconstructed function.

2.2 Implementation

The abstract mathematical formulation of the Maximum Entropy algorithm, which has been derived in the previous section, is not very descriptive. However, the expressions eq. (34) as bases of the iterative calculations can be interpreted geometrically. In the following section the geometrical relations are explained. These have been exploited for the technical design which is described in the subsequent section.

2.2.1 Geometrical Description

In section 2.1.1 we have introduced characteristic functions. Each of these functions explicitly extracts a unique parallel strip from the full x-y-parameter space (fi g. 3). Then the product of characteristic functions from different projections represents an overlay of strips and selects the common area of all affected strips. In fi gure 5 this is demonstrated for the product of the global characteristic functions eq. (17)

$$\prod_{j=1}^{N_j} \Omega_j(x\cos\theta_j + y\sin\theta_j) = \begin{cases} 1 & \mathcal{S} = (x,y) \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases} \quad . \tag{35}$$

The selected area $\mathcal{A}$ defi nes the allowed x-y-parameter space of the reconstruction function $\rho(x,y)$; for any point $\mathcal{S} \notin \mathcal{A}$ it vanishes. The picked-out area corresponds to a polygon whose edges are defi ned by the intersection points of the outer most boundaries of the different projections.

Now, supposed we have already found a solution to our problem, then all multipliers H_{jm} are known and the reconstructed function is unambigiuously given according to eq. (30). For any point $\mathcal{S} \in \mathcal{A}$ exactly one characteristic function Γ_{jm_j} for each projection j is non-zero due to eq. (15) . Thus the sum in eq. (30) only contains one non-vanishing element for each projection and gives

$$\rho(x,y) = \prod_{j=1}^{J} H_{jm_j} \tag{36}$$

where the indices m_j depend only on the chosen point $\mathcal{S}$. Analoguously to eq. (35), overlaying all non-vanishing characteristic functions $\Gamma_{jm_j}(x\,\cos\theta_j + y\,\sin\theta_j)$ geometrically defi nes one specifi c polygon $\Delta_{\{m_1,m_2,\ldots,m_J\}}$. The edges of each polygon are given by the related bin intersections. This is demonstrated in fi gure 6. All points $\mathcal{S} \in \Delta_{\{m_1,m_2,\ldots,m_J\}}$ refer to the same set of non-zero characteristic functions Γ_{jm_j}. Consequently they are associated to the identical function value of eq. (36). This means, the reconstructed function ρ is piecewise constant over unique polygons, i.e. all points belonging to one specifi c polygon are joined to the same function value. Integrals over the x-y-parameter space resp. $\mathcal{A}$ can be substituted by simple sums over all contributing polygons.

Each possible intersection combination $\mathsf{M}^{(\nu)} = \{m_1^{(\nu)}, \ldots, m_J^{(\nu)}\}$ is assigned exactly to one polygon $\Delta_{\mathsf{M}^{(\nu)}}$. The associated function value reads now as

$$\rho_\nu \equiv \prod_{\mathsf{M}^{(\nu)}}^{J} H_{jm_j^{(\nu)}} = H_{1m_1^{(\nu)}} \cdot H_{2m_2^{(\nu)}} \cdots H_{Jm_J^{(\nu)}} \quad . \tag{37}$$

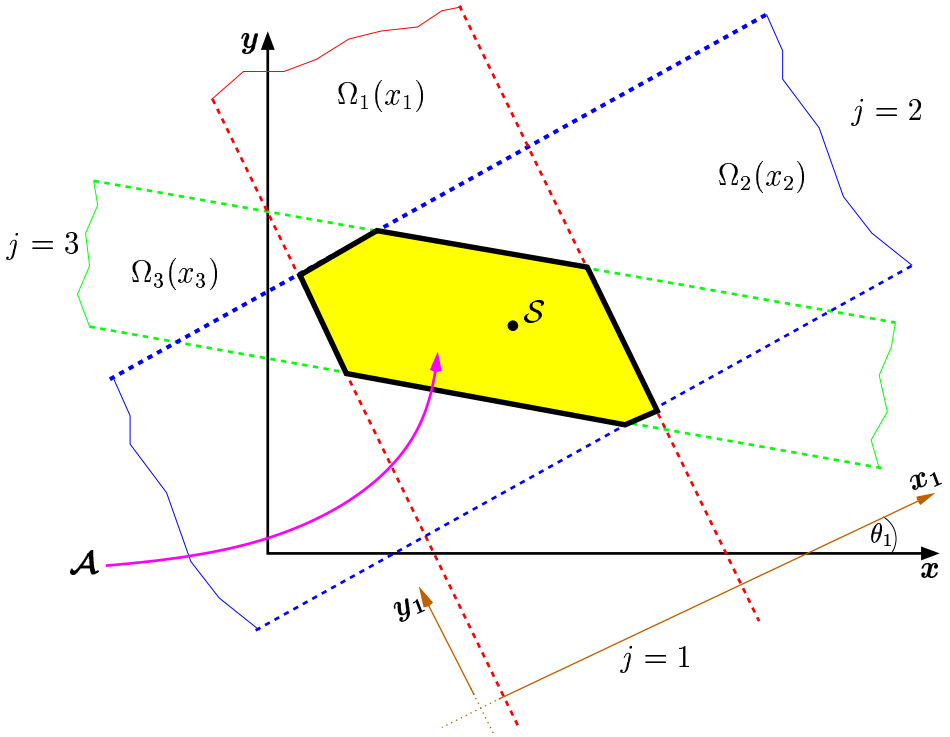

Figure 5: The overlay of the global characteristic functions $\Omega_j(x_j)$ of different projections j selects a unique polygon in the x-y space which defines the valid area of the reconstruction function ρ; the edges of the polygon are defined by the intersections of the corresponding bin boundaries.

The joined set of all intersection confi gurations resp. polygons is given by

$$\mathcal{M} = \left\{ \mathsf{M}^{(1)} \cup \mathsf{M}^{(2)} \cup \mathsf{M}^{(3)} \ldots \right\} \quad . \tag{38}$$

For convenience we also defi ne sub-sets of intersection combinations

$$\begin{aligned} \mathsf{M}_{jm}^{(\nu)} &= \left\{ m_1^{(\nu)}, \ \ldots, m_{j-1}^{(\nu)}, \ m, m_{j+1}^{(\nu)}, \ \ldots, m_J^{(\nu)} \right\} \\ \mathcal{M}_{jm} &= \left\{ \mathsf{M}_{jm}^{(1)} \cup \mathsf{M}_{jm}^{(2)} \cup \mathsf{M}_{jm}^{(3)} \ldots \right\} \end{aligned} \tag{39}$$

which reference all polygons of the m^{th} bin of the j^{th} projection. Finally, let F_ν be the size of the polygon $\Delta_{\mathsf{M}^{(\nu)}}$ then all integrals can be substituted by sums over polygons as follows:

$$\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \implies \sum_{\mathcal{M}} F_\nu \tag{40}$$

$$\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \ \Gamma_{jm}(x\cos\theta_j + y\sin\theta_j) \implies \sum_{\mathcal{M}_{jm}} F_\nu \tag{40a}$$

For the present we ignore the iteration cycle index in eq. (34) which allows us to rewrite it simply as

$$G_{jm} = \sum_{\mathcal{M}_{jm}} F_\nu \, \rho_\nu \tag{41}$$

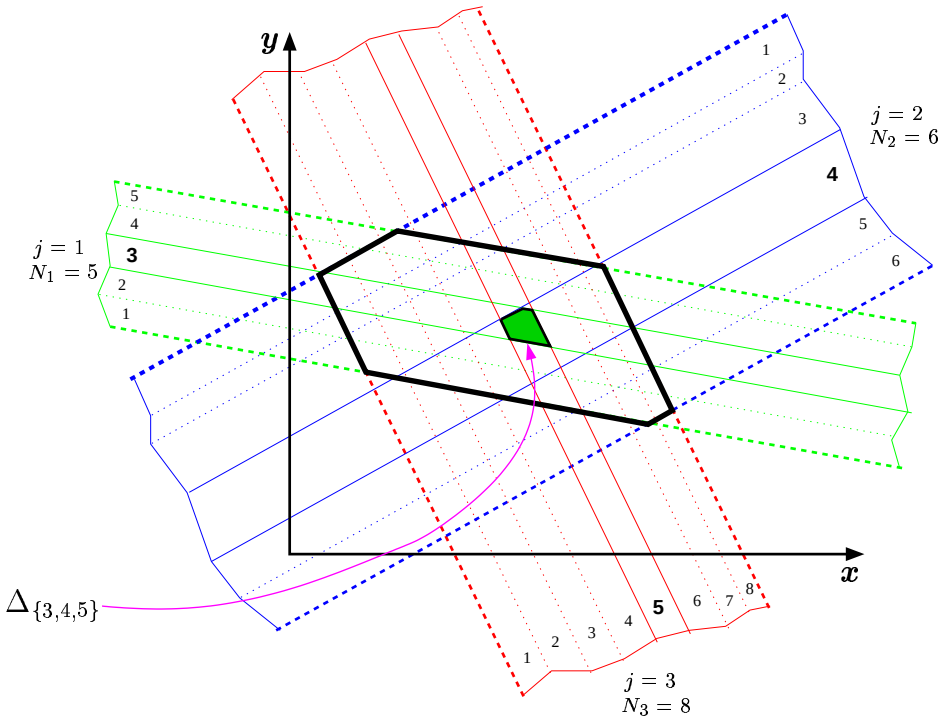

Figure 6: Overlay of characteristic functions $\Gamma_{jm}(x\,\cos\theta_j + y\,\sin\theta_j)$; as example characteristic functions of three different projections ($J = 3$) subdivide the area $\mathcal{A}$ into unique polygons as smallest entities. The edges of each polygon are given by the corresponding bin intersections. Each possible intersection combination $\mathsf{M}^{(\nu)} = \{m_1^{(\nu)}, m_2^{(\nu)}, m_3^{(\nu)}\}$ is assigned exactly to one polygon $\Delta_{\mathsf{M}^{(\nu)}}$.

using eq. (30) for the term in brackets. The geometrical interpretation becomes obvious now. Each element G_{jm} refers to a unique set of polygons $\Delta_{\mathsf{M}_{jm}^{(\nu)}}$ and all products of polygon size F_ν times the associated function value ρ_ν have to be summed up. Adapted polygons represent the smallest reasonable information units for the Maximum Entropy algorithm. The values ρ_ν are given as products of corresponding H-multipliers. Note, each polygon resp. H-multiplier provides multiple contributions, one to a specifi c bin of each projection. Finally, eq. (41) together with the iteration rule eq. (34) forms the base of the computational implementation of the Maximum Entropy algorithm.

2.2.2 Computational Design

Over the last years the focus of programming techniques has changed more and more to Object-Oriented concepts which incorporate essential aspects of modern software development like inheritance and data encapsulation. This becomes important even for complex software projects where portability appears as an important issue. In this context programming with C++ has developed to a powerful tool with growing acceptance and attraction, especially in the physics community. Furthermore, often the effort of programming can be reduced signifi cantly by conceptual re-usage of existing C++ tool boxes which provide adequate support to special kinds of problems. Therefore, it was decided to implement the Maximum Entropy algorithm in the context of C++. The implementation has been embbeded in the developing "Object-Oriented Data

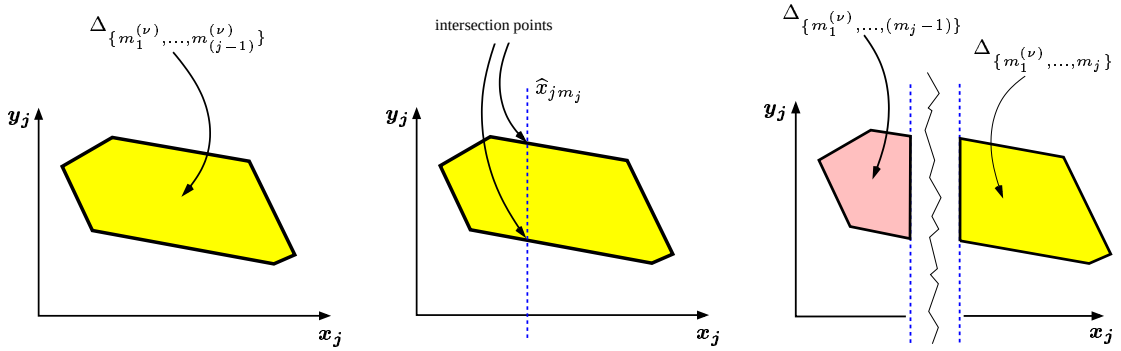

Figure 7: Working principle of method `Polygon::split(`x`)`: the initial polygon in the represention of the x_j-y_j coordinate system (left) will be splitted at the bin boundary $\widehat{x}_{jm_j}$ (middle) along the y_j axis; therefore the intersection points have to be calculated first. Finally the polygon will be splitted at the intersection points into two independent polygons (right). For each new polygon an additional subscript m_j resp. $(m_j - 1)$ from the recent projection j has to be assigned to uniquely identify its originating bin combination at a later stage.

Analysis Framework" of ROOT [1] which provides high level access to objects like histograms, graphs and 3D graphics. Especially the possibilities of data storage in histograms and graphical visualisation have been intensively used for programming and debugging the MENT algorithm. In addtion to running stand-alone programs, also interactive sessions using the command line interpreter allow for comfortable operation.

As pointed out in the previous section the main task refers to the calculation of the expressions eq. (41) which are fed in the iteration procedure eq. (34). One essential step now lies in the determination of all polygons $\Delta_{\mathsf{M}_{jm}^{(\nu)}}$ which are associated to G_{jm}. The polygons have to be calculated by evaluating and mapping the intersection points of the given bin boundaries (fi g. 6). To effi ciently handle these geometrical objects the class `Polygon` has been developed which provides related methods for all relevant manipulations of polygons, e.g. transformations to different coordinate systems, splitting along a certain line, calculation of areas, etc. (see also section A.5).

The essential calculations are coordinated within the class `MaxEntropy` (see also section A.1) which exhaustively uses the applied geometrical operations on polygons. The computational tasks are ordered according to the following scheme:

- **Initialisation**
  First of all, the input data G_{jm} with the appropriate bin boundaries have to be given as ROOT objects of one-dimensional histograms; also the related 2×2 transformation matrices (eq. (1)) as instances of class `Matrix` have to be given. Both information fragments are fed in `MaxEntropy` via method `MaxEntropy::set_projection(`$\cdots$`)` for each requested projection. The data will automatically be stored for the iteration procedure. To delete all input data `MaxEntropy::delete_all_projections()` has to be used.

- **Geometrical Calculations**
  The time-consuming geometrical calculations of polygons from bin intersections will be

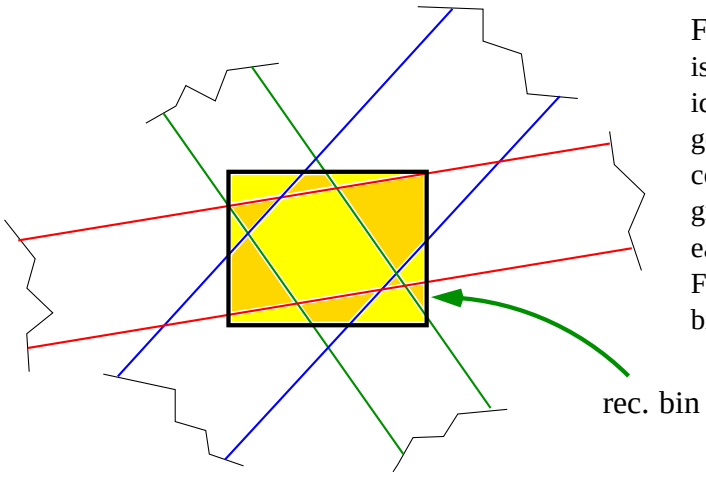

Figure 8: The reconstructed distribution is given in rectangular bins which typically do not match the underling polygon pattern; therefore $\rho_{\rm rec}$ which has a constant value over each polygon is integrated over all polygon fragments within each reconstruction bin (shaded areas). Finally the integral is normalised to the bin size of the rectangular bin.

initiated by calling the method `MaxEntropy::Iterate_from_storage(0)`. Then all contributing polygons are calculated recursively. Starting from a polygon simply defined as the largest rectangle which includes $\mathcal{A}$ (see fig. 5), this polygon is transformed into the coordinate system x_1-y_1 and subsequently cut into strips along the bin boundaries $\widehat{x}_{1m_1}$ for $m_1 = 1, ..., N_1$. The splitting procedure is illustrated in figure 7. The method `Polygon::split(`$\widehat{x}_{1m_1}$`)` calculates intersection points and provides two separated, independent polygons which represent the left and right side of the original polygon[8] with respect to the splitting line $\widehat{x}_{1m_1} = x\cos\theta_j + y\sin\theta_j$. Finally, we obtain several polygons $\Delta_{\{m_1^{(\nu)}\}}$ which are characterised by the evoking strip with index $m_1^{(\nu)}$. Each polygon is forwarded to the subsequent recursion level where the identical procedure is executed, however with coordinates in the representation of the subsequent coordinate system x_j-y_j and with corresponding bin boundaries $\widehat{x}_{jm_j}$. This recursive procedure has to be repeated for the subsequent projections until the terminating condition $j = J$ is fullfilled. We end up with <u>all</u> polygons $\Delta_{\{m_1^{(\nu)},...,m_J^{(\nu)}\}}$ covering the complete area $\mathcal{A}$. For the iterations only the area sizes and the assigned indices are needed. The latter is used to be able to map values ρ_ν to the related polygon $\Delta_{\mathsf{M}^{(\nu)}}$ according to eq. (37). In order to avoid the repeated execution of identical geometrical calculations for each iteration, thus saving much CPU time, the informations are stored in memory. Evaluated polygons are skipped after saving their size and intersection configuration $\mathsf{M}^{(\nu)}$.

➪ **Iterations**
Having all geometrical calculations finished single iterations can be initiated by calling the method `MaxEntropy::Iterate_from_storage(`i`)` with ($i > 0$). The relevant intersection combinations $\mathcal{M}_{jm}$ needed to calculate the elements $G_{jm}^{(i)}$ (eq. 34) will be recursively restored from memory. Then the $H_{jm}^{(i+1)}$ elements will be updated according to the iteration rules.

➪ **Reconstruction**
Whenever an iteration has been finished the recent values of the H_{jm}-multipliers can be

[8] In case of no valid intersection the complete initial polygon is attached either to the left or right side.

used to determine the actual solution of the reconstruction function according to eq. (36). Therefore the method `MaxEntropy::reconstruct_histo2D_from_Hspace(...)` has to be executed. Figure 8 illustrates the arising problem of mismatch between the intersection polygons and the rectangular bins of the two-dimensional histogram which should be filled. To avoid this intrinsic mismatch the reconstructed function is exactly integrated over the pieces of polygons within each rectangular bin. The integral value is normalised to the bin size and filled into the histogram. Thus artefacts due to binning effects do not appear. For large reconstruction bins with respect to typical polygon sizes mean values are obtained giving smoothed distributions whereas for smaller bins the polygon structure becomes more and more visible.

2.3 Numerical Examples and Performance

In the following sections two examples are investigated to document the performance of the Maximum Entropy algorithm. The first example gives an intuitive impression of the capability of reconstruction with the MENT algorithm. However, typical physical problems are often related to gaussian distributions. Therefore the second example deals with an arbitrary linear combination of gaussian distributions. Finally the topic of calculation time will be discussed briefly.

Before testing the algorithm with simulated projections we need to have a measure of the deviations between initial and reconstructed distributions. In the following sections the reconstruction error will simply be quantified as the mean-square norm of the deviation of the reconstruction solution $\rho_{\mathrm{rec}}(x,y)$ from the expected value $\rho_{\mathrm{ini}}(x,y)$. For the binned distributions the expression

$$\sigma = \sqrt{\frac{\sum_\alpha \sum_\beta \left[(\rho_{\mathrm{rec}}(x_\alpha, y_\beta) - \rho_{\mathrm{ini}}(x_\alpha, y_\beta)\right]^2}{\sum_\beta \sum_\alpha \left[\rho_{\mathrm{ini}}(x_\alpha, y_\beta)\right]^2}} \tag{42}$$

will be evaluated. The sums go over all two-dimensional rectangular bins which are referenced by the variables x_α and y_β. By construction all bins of the initial and reconstructed distributions are always of equal size. Therefore the bin sizes have been left out from expression eq. (42).

2.3.1 Example I: Flat Top Distributions

The capability of the implemented Maximum Entropy algorithm is impressively visualised in figure 9. The chosen initial test distribution has been composed of two separated flat top regions, one cylindrical and one sickle-shaped, which additionally have different levels. Then from the composed distribution the projection profiles have been calculated and fed in the reconstruction. All simulated projections are subdivided into 50 equidistant bins. The figure shows reconstructed distributions in contour view (left) and also in isometric view (right) for $J = 3, 4, 30$ applied projections.

As expected the quality of reconstruction increases with the number of available projections. For $J = 30$ projections the shape and amplitude of the reconstructed distribution are described with high accuracy. Using only $J = 4$ projections already allow an estimation of the underlying

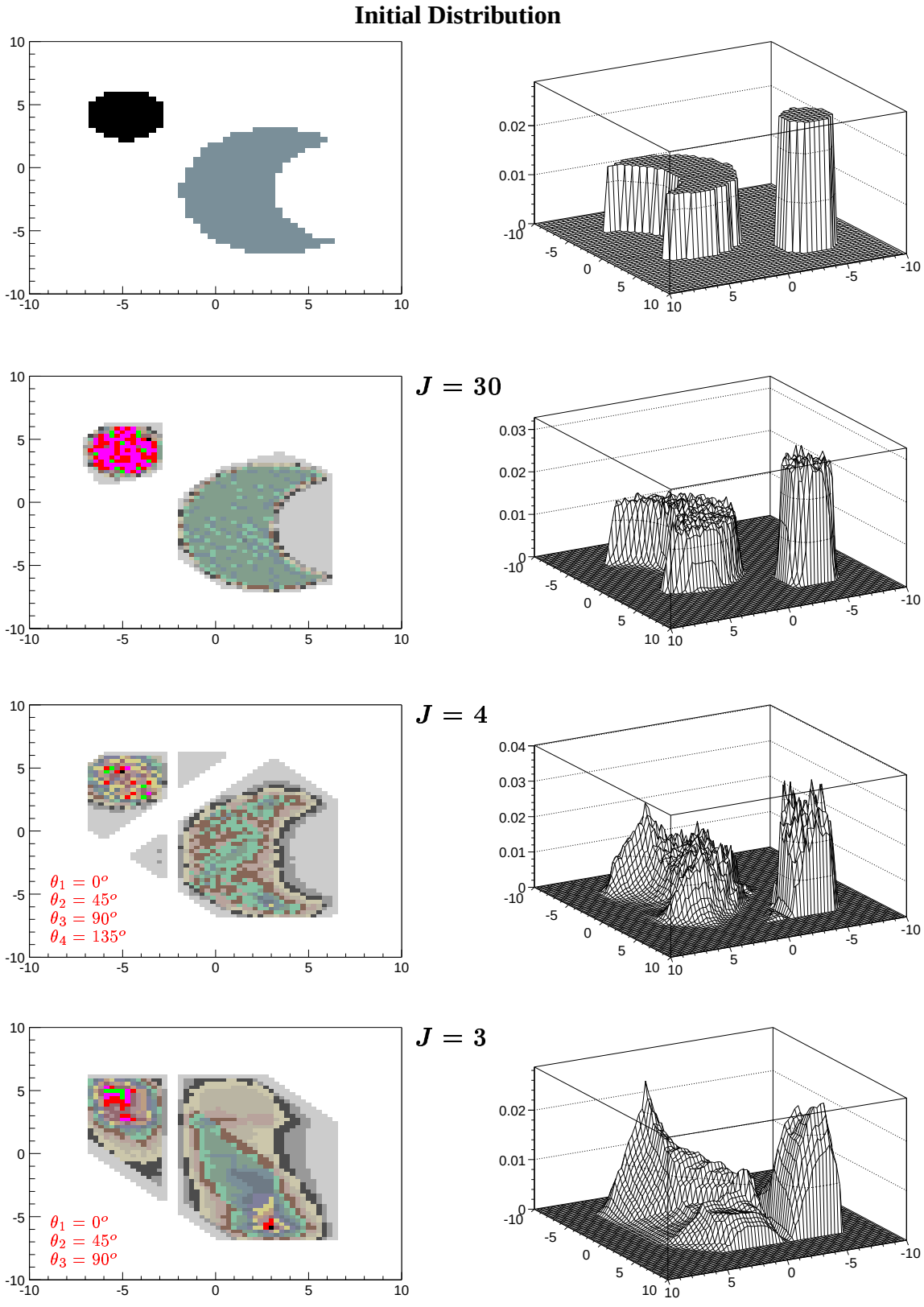

Figure 9: Illustration of the capability of the Maximum Entropy algorithm; the initial test distribution consisting of two separated, flat areas (top) has been used to calculate projections which are used in the reconstruction. The obtained reconstructed distributions are shown as contour plots (left) and isometric plots (right) for three different numbers of projections, i.e. $J = 3, 4, 30$.

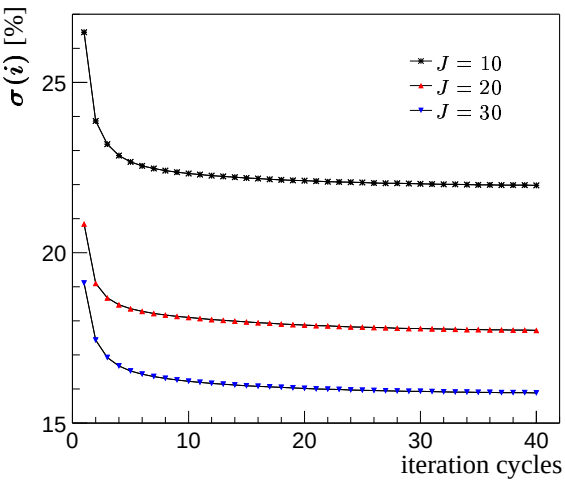

Figure 10: Deviation of the reconstructed distribution with respect to to the initial distribution according to eq. (42) as function of the number of iteration cycles; three curves for different numbers of applied projections $J = 10, 20, 30$ are shown. The observed deviations decrease with increasing amount of input information, i.e. increasing number of projections.

distribution in case of an optimised choice of rotation angles[9]. The main characteristics, i.e. position and shape, can clearly be identified. Some small artefacts arise in the "*projection shadow*" of the distribution. These artefacts further increase for $J = 3$ and the reconstruction is degraded to a rough estimate of the true distribution.

The convergence behaviour of the Maximum Entropy algorithm can be seen in figure 10. The figure shows the deviation of the reconstructed distribution from the initial distribution according to eq. (42) as function of the number of iteration cycles. The three curves for $J = 10, 20, 30$ projections decrease rapidly within the first $\mathcal{O}(10)$ iteration cycles. Afterwards the numerical changes between subsequent iteration cycles become small. In practice $\mathcal{O}(50)$ iterations seem to be sufficient to get numerically stable results. The observed deviations reduce with increasing number of projections. However, the improvement on precision is more emphasised for the transition $J = 10 \rightarrow 20$ than for $J = 20 \rightarrow 30$. Generally, inserting an additional projection evokes a higher gain on the accuracy of the reconstruction for smaller total numbers of projections[10].

2.3.2 Example II: Gaussian Distributions

The example in the previous section intuitively documents the capability of the Maximum Entropy technique. However, gaussian distributions have much more importance as realistic physics scenarios and should be used for meaningful benchmarks of the reconstruction accuracy. In this section the overlay of three gaussian distributions with different center of mass, width and amplitude pose as initial test distrubition. Figure 11 describes the applied initial distribution (top) as well as four examples of the resulting reconstruction by the Maximum Entropy algorithm. The four reconstructed distributions refer to different numbers of applied projections

[9] To maximise the amount of input information all angles should be chosen as different as possible with respect to each other.

[10] In principle we have two crucial degrees of freedom to increase the level of precision, numbers of projections and numbers of bins for each projection. Optimising the level of reconstruction precision depends on both parameters.

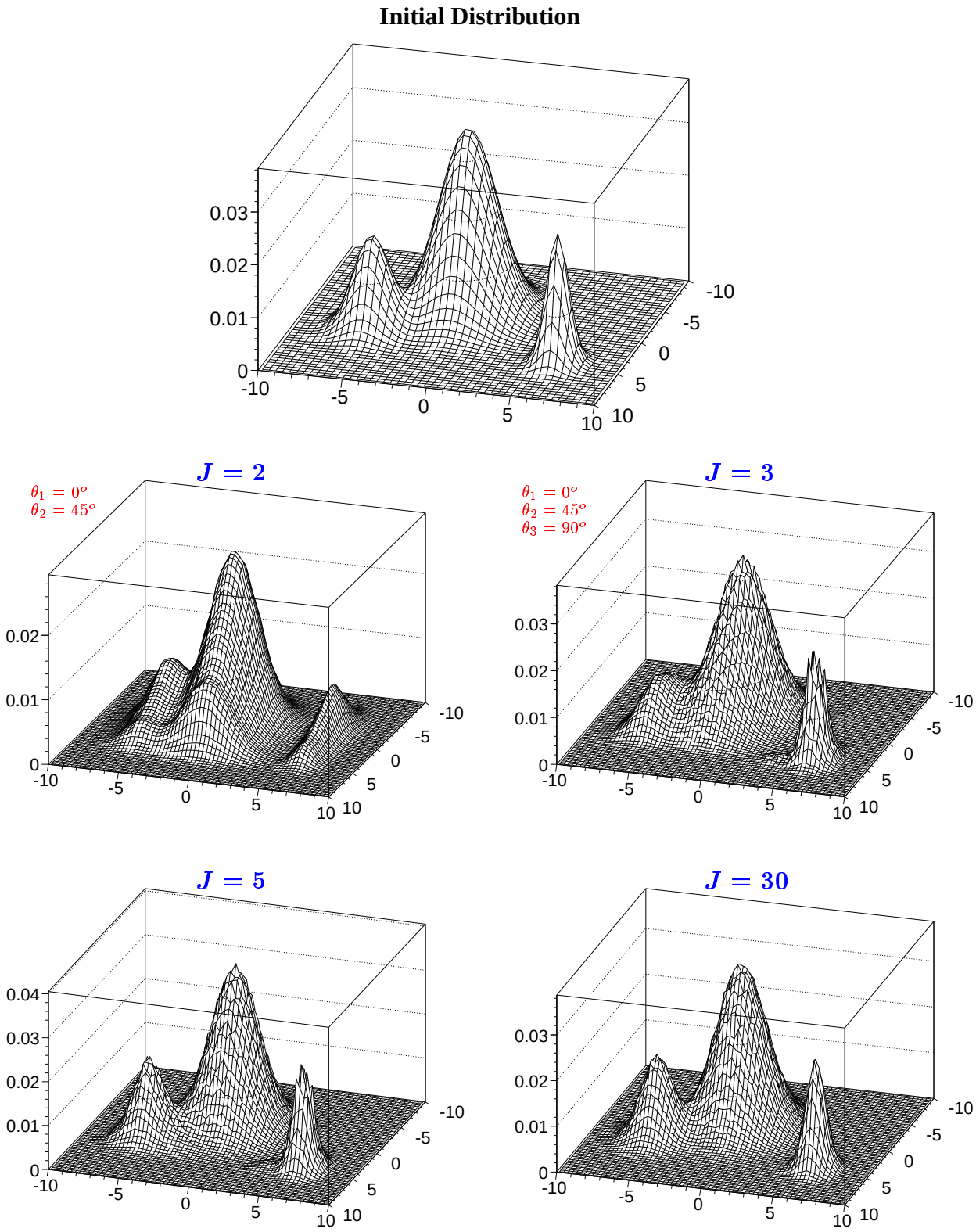

Figure 11: The overlay of three gaussian distributions with different center of mass, width and amplitudes (top) pose as benchmark of the reconstruction efficiency; the reconstructed distributions are shown for $J = 2, 3, 5, 30$ projections in use.

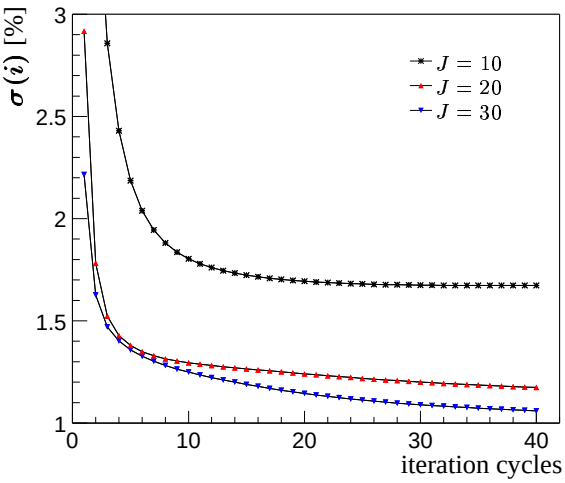

Figure 12: Deviation of the reconstructed distribution with respect to the composed gaussian distribution (fig. 11; top) according to eq. (42) as function of the number of iteration cycles; three curves for different numbers of applied projections $J = 10, 20, 30$ are shown. The observed deviations decrease with increasing amount of input information, i.e. increasing number of projections.

($J = 2, 3, 5, 30$) where possible error sources, e.g. noise, have been ignored. Each projection has been subdivided into 50 equidistant bins. Using $J = 30$ simulated projections in the reconstruction an impressing congruence is achieved. Overlapping regions, widths and amplitudes are nicely reproduced. A remarkable observation is that only 5 projections are sufficient for a reasonable reconstruction of the initial distribution. If the number of used projections is only 2 or 3 significant artefacts appear in the reconstruction, but these cases would anyway have to be regarded as incomplete measurements.

Figure 12 shows the deviation of the reconstructed distribution from to the initial distribution (according to eq. (42)) as function of the number of iteration cycles. Similar to the previous example the growing number of available projections is accompanied by a reduction of the observed deviations. Again the three curves for $J = 10, 20, 30$ rapidly decrease within the first $\mathcal{O}(10)$ iteration cycles. In contrast to the previous example slightly more iteration cycles are needed for $J = 30$ to get numerically stable results. Note, that for equal conditions, i.e. equal binnings and number of projections, the observed deviation for the composed gaussian distribution is an order of magnitude smaller than for the flat distribution of the previous example. This demonstrates that the Maximum Entropy algorithm is especially well suited to reconstruct gaussian shaped profiles.

The achievable accuracy depends on both, the number of available projections and the number of bins per projection. This is illustrated in figure 13. For the gaussian example the maximum gain in accuracy is obtained for when going form 6 to 7 projections (left) whereas the improvements for additional projections are only moderate. Also the number of bins of each applied projections massively affects the level of precision (right). Strong degeneration is correlated to small numbers of bins per projection. Therefore the used binning within the projections should be adjusted close to the resolution limits of the measuring devices to optimise the reconstruction efficiency.

2.3.3 Calculation Time

An important limitation for usage of tomographic reconstruction methods is often given by the restricted computer resources. However over the last years the development on computer

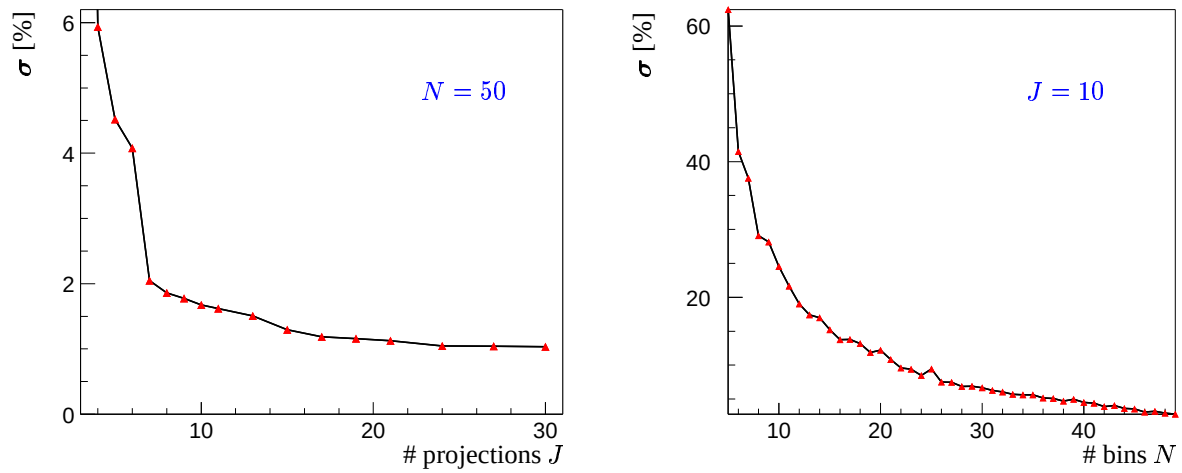

Figure 13: Deviation of the reconstructed distribution from the composed gaussian initial distribution (eq. (42)) as function of the number of projections with $N = 50$ bins per projections (left) and as function of the number of bins per projection with fixed number of projections to $J = 10$ (right)

hardware brought overwhelming improvements. All presented results have been calculated on a standard Linux PC with 1.7 GHz Intel Pentium IV processor and 256 MB memory.

The needed CPU time for a reconstruction based on the Maximum Entropy method depends mainly on the number of contributing polygons. Figure 14 illustrates the quadratical dependence of the number of polygons on the number of projections. Here 50 equidistant bins per projection have been used. With increasing number of polygons also the complexity of the polygons, i.e the number of constituting points, increases. Consequently the calculation time for all envolved polygons increases quadratically with number of projections or the number of bins per projection respectively. Figure 15 shows the consumed CPU time for the geometrical calculations of all polygons which depends on the number of projections J (left) and the number of bins per projection N. For example, using $J = 15$ projections with $N = 50$ bins each

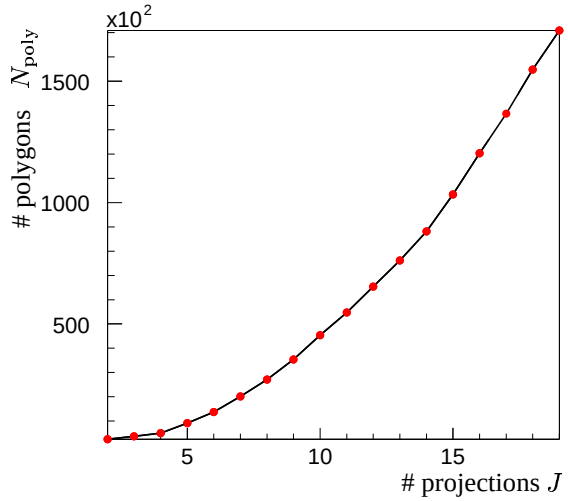

Figure 14: Number of calculated unique polygons as function of the number of different projections; the numbers have been evaluated from the gaussian example in sec. 2.3.2 where 50 bins per projection have been used.

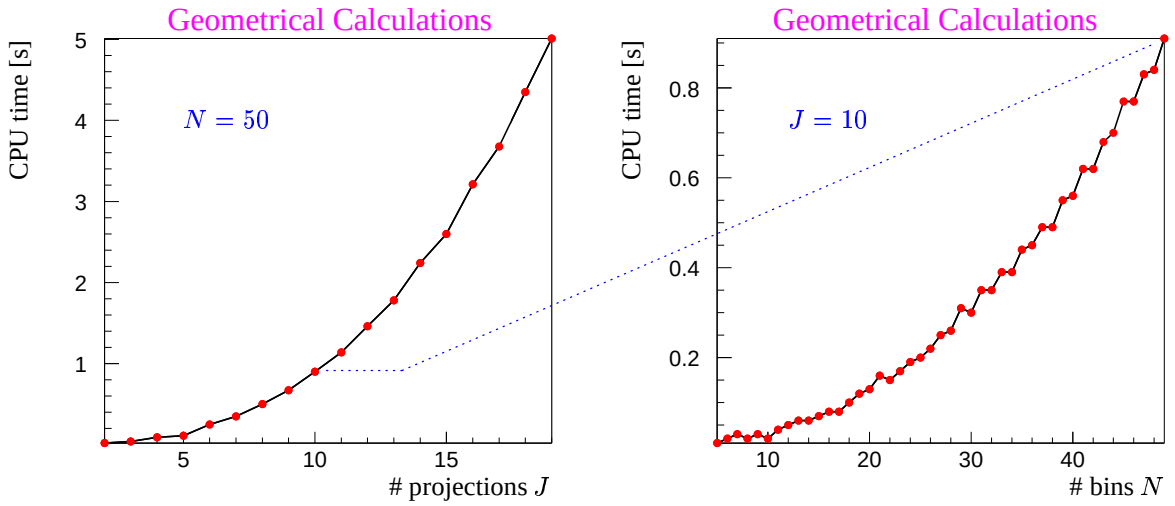

Figure 15: Needed CPU time in seconds for the geometrical calculations of all involved polygons as function of the number of used projections J (left) and the number of bins per projection N (right). The left diagram has been evaluated for a fixed number of bins per projection with $N = 50$ whereas the right diagram has been evaluated for a fixed number of projections with $J = 10$. The dotted line marks the common point of both diagrams.

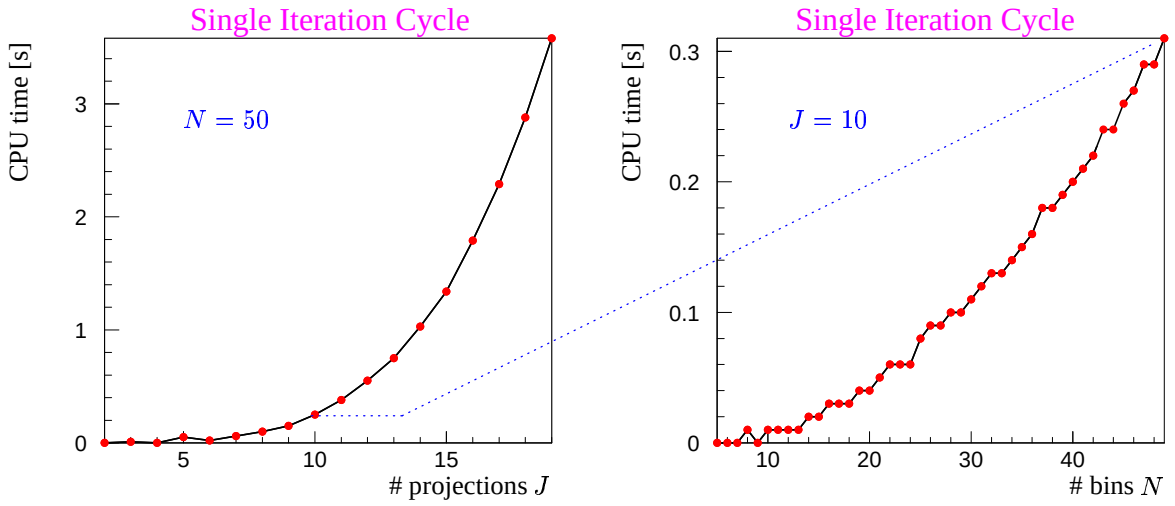

Figure 16: Needed CPU time in seconds for a single iteration cycle including all sub-iterations and using the memory-resident geometrical calculations; the plots illustrate the dependence on the number of used projections J (left) and the number of bins per projection N (right). The left diagram has been evaluated for a fixed number of bins per projection with $N = 50$ whereas the right diagram has been evaluated for a fixed number of projections with $J = 10$. The dotted line marks the common point of both diagrams.

then $\mathcal{O}(10^5)$ polygons have to be calculated which needs roughly 3 seconds CPU time.

Thus for the chosen configuration one complete iteration cycle normally would take roughly $J \times 3\,\mathrm{s} = 45\,\mathrm{s}$ CPU time due to the necessary execution of sub-iteration cycles. However, as mentioned in section 2.2.2 the geometrical calculations have to be performed only once, because the results, i.e. polygon sizes F_ν and corresponding bin configurations $\mathsf{M}^{(\nu)} = \{m_1^{(\nu)}, ..., m_J^{(\nu)}\}$, are stored residently in memory. Figure 16 shows the effective CPU time needed for a single iteration cycle including all sub-iteration cycles when the geometrical calculations are stored in memory. Also here the CPU time dependence on the number of projections resp. number of bins per projection evolves quadratically in parallel with the set of polygons. However the needed absolute CPU time is dramatically reduced. For the above scenario using $J = 15$ projections with $N = 50$ bins each the total CPU time takes roughly 1.5 seconds which means a factor of 30 reduction of reconstruction time. The only limiting factor could be the available memory space.

3 Phase Space Tranformations for Accelerator Beams

3.1 Basics

The motion of a charged particle in an accelerator can be described by a trajectory in the six-dimensional phase space. The beam line as combination of magnetic elements and accelerating structures defines the boundary conditions for the propagating particle. Now, if the initial parameters

$$\vec{X} = (x, x', y, y', l, \delta) \tag{43}$$

and the beam line elements are known, the full trajectory $\vec{X}(s)$ in the phase space can be calculated. The coordinates in eq. (43) refer to a reference trajectory. The variables x and y denote the horizontal and vertical displacement of the particle, x' and y' the horizontal and vertical angle of the particle trajectory with respect to the reference trajectory. The variables l and δ refer to the longitudinal phase space and denote the longitudinal displacement and the fractional momentum deviation ($\delta = \Delta p/p_o$) with respect to the reference particle. Instead of the longitudinal displacement l the corresponding time difference τ can be used.

3.2 Linear Transfer Matrix Formalism

The main magnets in an accelerator are dipoles and quadrupoles where the field depends linearly on the displacement x resp. y. Therefore in the "linear optics" approximation the beam line elements can be characterised by matrices and the particle trajectory results from matrix multiplications

$$\vec{X}_\mathrm{f} = \mathbf{M} \cdot \vec{X}_\mathrm{i} \tag{44}$$

where $\mathbf{M}$ is a 6×6 square matrix, called beam transfer matrix. The effective transformation for the entire beam line can easily be obtained by multiplying the transfer matrices $\mathbf{M}_\xi$ of all beam line elements $\xi = [1, ...n]$ sequentially from the end to the beginning of the beam line

$$\mathbf{M}_\mathrm{eff} = \mathbf{M}_n \cdot \mathbf{M}_{n-1} \cdots \mathbf{M}_2 \cdot \mathbf{M}_1 \quad . \tag{45}$$

The physical properties of the different possible beam line elements find their mathematical correspondence in special transfer matrices:

- **Drift Space:**
  The simplest beam line element is a drift space of length L where no forces influence the particles and only the transverse displacements do change according to x' resp. y'. The transfer matrix looks as follows:

$$\mathbf{M}_{\mathrm{D}} = \begin{pmatrix} 1 & L & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & L & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{46}$$

- **Quadrupole Magnet:**
  A quadrupole magnet focuses the beam in one plane and defocuses in the orthogonal plane. The length L and the strength k of the quadrupole are the basic parameters of a quadrupole. The strength again depends on the gradient g of quadrupole magnet and the momentum p of the particle

$$k = e\,g\,\frac{1}{E} \qquad\qquad [k] = \frac{1}{\mathrm{m}^2} \qquad . \tag{47}$$

  With $\Omega = L\sqrt{k}$ the focusing resp. defocusing sub-matrices

$$\mathbf{Q}_{\mathrm{F}} = \begin{pmatrix} \cos(\Omega) & \frac{1}{\sqrt{k}}\sin(\Omega) \\ -\sqrt{k}\sin(\Omega) & \cos(\Omega) \end{pmatrix} \tag{48}$$

$$\mathbf{Q}_{\mathrm{D}} = \begin{pmatrix} \cosh(\Omega) & \frac{1}{\sqrt{k}}\sinh(\Omega) \\ \sqrt{k}\sinh(\Omega) & \cosh(\Omega) \end{pmatrix} \tag{49}$$

  give the full 6×6 transformation matrix via

$$\mathbf{M}_{\mathrm{Q}} = \begin{pmatrix} \mathbf{Q}^{x}_{\mathrm{F\,(D)}} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{Q}^{y}_{\mathrm{D\,(F)}} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{1} \end{pmatrix} \quad \text{with} \quad \mathbf{O} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} . \tag{50}$$

  The matrix describes a quadrupole magnet aligned according to the x-y axis.

- **Acceleration:**
  The acceleration of relativistic electron bunches in a radio frequency (RF) cavity affects

the longitudinal phase space. The longitudinal displacement l and the fractional energy deviation δ transform in linear approximation according to the following transfer matrix:

$$\mathbf{M}_{\text{cav}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & M_{65} & M_{66} \end{pmatrix} \tag{51}$$

Here the transverse and longitudinal phase space are decoupled so that the relevant transformations simplify to

$$\begin{pmatrix} l' \\ \delta' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ M_{65} & M_{66} \end{pmatrix} \begin{pmatrix} l \\ \delta \end{pmatrix} \ . \tag{52}$$

Let Φ_0 be the RF phase and $E^{\text{ref}}_{\text{initial}}$ the initial energy corresponding to the reference particle. The phase difference $\Delta\Phi$ is explicitly related to the longitudinal displacement l via

$$\Delta\Phi = \Phi - \Phi_0 = \frac{2\pi}{\lambda} l \ . \tag{53}$$

Now, with the voltage U_{acc} of the cavity the matrix elements can be written as follows [8]

$$M_{66} = \frac{E^{\text{ref}}_{\text{initial}}}{E^{\text{ref}}_{\text{initial}} + eU_{\text{acc}} \cos\Phi_0} \qquad \text{and} \qquad M_{65} = -\frac{eU_{\text{acc}} \sin\Phi_0}{E^{\text{ref}}_{\text{initial}} + eU_{\text{acc}} \cos\Phi_0} \frac{2\pi}{\lambda} \ . \tag{54}$$

Besides the linear beam matrix formalism, higher order effects [8], i.e. the non-linearity of the energy gain within the longitudinal dimension, can be taken into account by extending the linear equations (52) to

$$\delta' = M_{66}\,\delta + M_{65}\,l + M_{\text{non-lin}}\,l^2 \tag{55}$$

with

$$M_{\text{non-lin}} = -\frac{\frac{1}{2}eU_{\text{acc}} \cos\Phi_0}{E^{\text{ref}}_{\text{initial}} + eU_{\text{acc}} \cos\Phi_0} \left(\frac{2\pi}{\lambda}\right)^2 \ . \tag{56}$$

The linear beam transfer formalism is easy to handle for the needs of tomography. However, higher order effects such as higher multipoles in the magnets or internal Coulomb forces in the bunches are neglected here. In addition experimental errors, e.g. imperfections of the magnetic structures or misalignment cause systematic uncertainties on the knowledge of the true transfer matrices. Therefore the quantitative results of tomographic phase space reconstruction could be distorted according to these uncertainties. More details of the transfer matrix formalism can be found elsewhere [12].

3.3 Beam Moments and Beam Matrix

Instead of tracing individual particles along the beam line normally it is more convenient to consider the dynamics of whole particle bunches to study the propagation of the beam. Due to the large number of particles within one bunch, up to 10^{10} electrons in case of the TESLA Test Facility, the bunches are characterised by the density distribution $\varrho(x, x', y, y', l, \delta) \equiv \varrho(x_1, x_2, x_3, x_4, x_5, x_6)$ in the six-dimensional phase space where $\varrho(x_1, ..., x_6)\, dV$ gives the number of particles in the volume element dV at the phase space coordinates $\mathcal{P}$=$(x_1, x_2, x_3, x_4, x_5, x_6)$.

The density distribution ϱ is used to compute the first and second moments

$$\sigma_i \equiv \int x_i\, \varrho(x_1, x_2, x_3, x_4, x_5, x_6)\; dV = < x_i > \tag{57}$$

$$\sigma_{ij} = \sigma_{ji} \equiv \int (x_i - \sigma_i)(x_j - \sigma_j)\, \varrho(x_1, x_2, x_3, x_4, x_5, x_6)\; dV\,. \tag{58}$$

The σ_i define weighted average of the corresponding coordinate and the second moments σ_{ij} form the covariance matrix $\boldsymbol{\sigma}$ of the density distribution

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} & \sigma_{15} & \sigma_{16} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} & \sigma_{25} & \sigma_{26} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} & \sigma_{35} & \sigma_{36} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} & \sigma_{45} & \sigma_{46} \\ \sigma_{51} & \sigma_{52} & \sigma_{53} & \sigma_{54} & \sigma_{55} & \sigma_{56} \\ \sigma_{61} & \sigma_{62} & \sigma_{63} & \sigma_{64} & \sigma_{65} & \sigma_{66} \end{pmatrix} \tag{59}$$

which is called the beam matrix. The diagonal elements describe the transverse spatial beam width in the horizontal plane σ_{11} and vertical plane σ_{33}, the transverse angular divergence in the horizontal plane σ_{22} and the vertical plane σ_{44}, the bunch length σ_{55} and the fractional momentum deviation σ_{66}. The off-diagonal elements define the covariances between the different phase space coordinates. In total 21 beam parameters are independent. For a six-dimensional gaussian density distribution ϱ_g the first and second moments are sufficient to define ϱ_g unambigiously. Then the density function is given by

$$\varrho_g(x_1, ..., x_6) = \sqrt{\frac{|\widehat{\boldsymbol{\sigma}}|}{(2\pi)^6}} \exp\left(-\frac{1}{2} \sum_{i=1}^{6} \sum_{j=1}^{6} \widehat{\sigma}_{ij} (x_i - \sigma_i)(x_j - \sigma_j) \right) \tag{60}$$

with $\widehat{\sigma}_{ij}$ as the matrix elements of $\widehat{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^{-1}$ and $|\widehat{\boldsymbol{\sigma}}| \equiv \det(\widehat{\boldsymbol{\sigma}})$. Often measurements of beam parameters base on the assumption of gaussian distributed profiles, e.g. quadrupole scans to determine the emittance. Here the capability of tomography becomes clear. Besides the determination of RMS beam parameters the tomography allows to quantify the deviations from gaussian profiles.

For any linear coordinate transformation $\mathbf{M}$ according to eq. (44) it can be shown [13] that the covariance matrix $\boldsymbol{\sigma}_\mathbf{i}$ for a given density distribution $\varrho \geq 0$ undergoes a transformation as follows

$$\boldsymbol{\sigma}_\mathrm{f} = \mathbf{M}\, \boldsymbol{\sigma}_\mathrm{i}\, \mathbf{M}^\mathrm{T}\;. \tag{61}$$

This transformation law of the variances and covariances is not restricted to gaussian distributions, but holds for arbitrary density distributions. Consequently, the evolution of the beam matrix along the beam line is determined by the single transfer matrix $\mathbf{M}$ according to eq. (45). The beam parameters can be calculated at any position if they are measured at one place. However, it must be noted that the magnetic elements lead to dispersive distortions due to the energy dependence of the transfer matrix. These effects will be ignored for our considerations concerning the transverse phase space reconstruction, i.e. the longitudinal and transverse phase space are reconstructed independently ignoring correlations among each other (see also next section). Therefore chromaticity represents one relevant source of systematic errors for tomographic reconstruction of the phase space distribution.

3.4 Simplifying Approximations (Decoupling)

Probing the full the six-dimensional phase space of a particle beam is a highly complex problem. Different measuring techniques have to supplement each other. To reduce the complexity resonable approximations have to be made which allow us to determine important beam parameters without knowledge of the full six-dimensional phase space distribution.

From a fundamental point of view the determination of phase space distributions by tomographic reconstruction techniques (but also for various other methods) has an intrinsic difficulty. The precise propagation of the beam along the beam line depends on the exact phase space distribution in the full **six**-dimensional phase space. However the determination of the phase space distribution bases on the knowledge of the predicted beam propagation thus having a kind of vicious circle. For example, the beam propagation within the transverse phase space also depends on the energy spectrum due to the effects of chromaticity, i.e. transverse and longitudinal phase space are not entirely decoupled. In many cases the correlations among the different dimensions of the phase space can be neglected in good approximation which allows a factorisation of the phase space distribution as follows

$$\varrho(x_1, x_2, x_3, x_4, x_5, x_6) = \varrho_{\mathrm{trans,X}}(x_1, x_2) \cdot \varrho_{\mathrm{trans,Y}}(x_3, x_4) \cdot \varrho_{\mathrm{long}}(x_5, x_6) \quad . \tag{62}$$

Under this assumption the beam matrix formalism provides a simple description of the beam propagation which can be exploited for two-dimensional phase space tomography[11]. All non-diagonal elements of the beam matrix eq. (59) are assumed to vanish, except the elements $\sigma_{12} = \sigma_{21}$, $\sigma_{23} = \sigma_{32}$ and $\sigma_{56} = \sigma_{65}$. The different sub-phase spaces are then decoupled and can be evaluated separately by tomographic reconstruction.

The assumption of *decoupled* phase space distributions according to eq. (62) is only justified if also the transfer matrices leave the x_1-x_2 phase space and x_3-x_4 phase space decoupled. In particular, skew quadrupole components must be absent. Then the transfer matrix simplifies to

$$\mathbf{M}_{\mathrm{trans}} = \begin{pmatrix} \mathbf{M}_{\mathrm{x}} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\mathrm{y}} \end{pmatrix} \qquad \text{with} \tag{63}$$

[11] Strictly speaking the two-dimensional tomographic reconstruction of $\varrho_{\mathrm{trans,X}}(x_1, x_2)$ resp. $\varrho_{\mathrm{trans,Y}}(x_3, x_4)$ also operates for a true transverse phase space distribution $\varrho_{\mathrm{trans}}(x_1, x_2, x_3, x_4)$ if the requirement of eq. (63) holds. Then the distributions $\varrho_{\mathrm{trans,X(Y)}}$ represent the two-dimensional projections of the true distribution.

$$\mathbf{M}_{\mathrm{x}} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \ , \ \mathbf{M}_{\mathrm{y}} = \begin{pmatrix} M_{33} & M_{34} \\ M_{43} & M_{44} \end{pmatrix} \text{ and } \boldsymbol{O} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \ . \tag{64}$$

Here *a priori* possible couplings to the longitudinal phase space, i.e. especially energy dependencies are ignored. For the decoupled phase spaces one defi nes 2×2 beam matrices

$$\boldsymbol{\sigma}_{\mathrm{x}} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} \text{ and } \boldsymbol{\sigma}_{\mathrm{y}} = \begin{pmatrix} \sigma_{33} & \sigma_{34} \\ \sigma_{43} & \sigma_{44} \end{pmatrix} \ . \tag{65}$$

The evolution of the tranverse horizontal beam parameters $\boldsymbol{\sigma}_{\mathbf{x}}$ resp. transverse vertical beam paramters $\boldsymbol{\sigma}_{\mathbf{y}}$ along the beam line is again described via eq. (61), but now for the case of 2×2 matrices where the transformations in x_1-x_2 phase space with $\mathbf{M}_{\mathbf{x}}$ resp. x_3-x_4 phase space with $\mathbf{M}_{\mathbf{y}}$ do not depend on each other.

3.5 Phase Ellipses and Beam Parameters

The simplifying approximations described in section 3.4 allow a simple interpretation of the (remaining) beam parameters. For example, if we consider the decoupled two-dimensional density distribution in the x-x' plane, the corresponding density function looks as follows[12]

$$\varrho_{\mathrm{x}}(x_1, x_2) = \tilde{N} \exp\left(-\frac{x_1^2}{2\sigma_{11}} + x_1 x_2 \frac{\sigma_{12}}{2\sigma_{11}\sigma_{22}} - \frac{x_2^2}{2\sigma_{22}} \right) \qquad \text{with } \ x_1 \equiv x \, , \ \ x_2 \equiv x' . \tag{66}$$

The ϱ_{x} distribution can be intuitively interpreted in terms of iso-lines in the x_1-x_2 plane for $\varrho_{\mathrm{x}}(x_1, x_2) = \tilde{C}$ which defi nes ellipses of the form

$$\sigma_{22} x_1^2 - 2\sigma_{12} x_1 x_2 + \sigma_{11} x_2^2 = C \tag{67}$$

and the 1σ phase ellipse is given for $C = 1$. The ellipse is determined in shape and orientation by three parameters, i.e. the three second order moments $(\sigma_{11}, \sigma_{22}, \sigma_{12})$ or alternatively the so-called Twiss parameters $(\beta_{\mathrm{x}}, \alpha_{\mathrm{x}}, \gamma_{\mathrm{x}})$ related by $\beta_{\mathrm{x}}\gamma_{\mathrm{x}} - \alpha_{\mathrm{x}}^2 = 1$ together with the beam emittance ϵ_{x} [15]. The relations between the phase space ellipse and beam parameters are shown in fi gure 17. For the y-y' phase space analogous relations exist. The Twiss parameter notation is related to the beam moments via

$$\boldsymbol{\sigma}_{\mathbf{x(y)}} = \begin{pmatrix} \sigma_{11(33)} & \sigma_{12(34)} \\ \sigma_{21(43)} & \sigma_{22(44)} \end{pmatrix} = \epsilon_{\mathrm{x(y)}} \begin{pmatrix} \beta_{\mathrm{x(y)}} & -\alpha_{\mathrm{x(y)}} \\ -\alpha_{\mathrm{x(y)}} & \gamma_{\mathrm{x(y)}} \end{pmatrix} \ . \tag{68}$$

The beam emittance $\epsilon_{\mathrm{x(y)}}$ is a measure of the beam quality. It is one of the essential parameters to characterise the particle density in the beam for example in terms of specifc luminosity for colliding beams or in terms of radiation power of an SASE free electron laser. The beam emittance is defi ned as the phase space volume occupied by a certain fraction of the beam particles. Suppose we have gaussian distributions the beam emittance ϵ is typically given by the area of the 1σ level phase ellipse which correspondingly surrounds 38% of the beam particles. If the energy is conserved (no decelerating/accelerating structures) due to Liouville's theorem the occupied volume resp. emittance is a conserved quantity during the beam propagation. Small

[12] Here possible displacements with respect to the point of origin are ignored assuming $\sigma_1 = 0$ and $\sigma_2 = 0$.

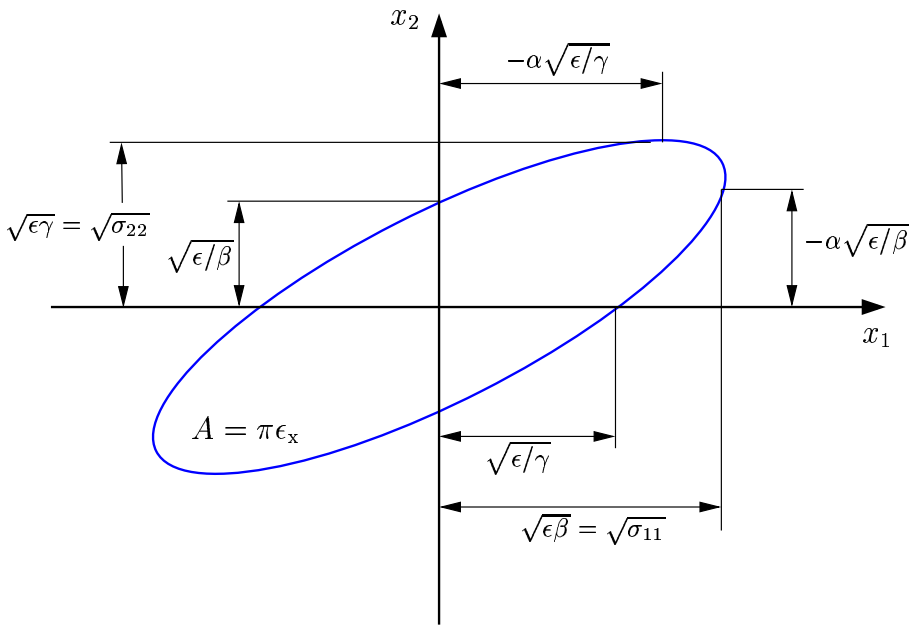

Figure 17: Relation between the phase space ellipse and the beam parameters in the x_1-x_2 plane of the tranverse phase space; analogous relations exist for the x_2-x_3 phase space.

occupied phase space volumes are related to small spatial beam sizes and small angular divergence. The statistical definition of the emittance for the transverse horizontal, transverse vertical and longitudinal phase space are given by the beam parameters as follows

$$\epsilon_{\mathrm{x}} = \sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}^2} \qquad \epsilon_{\mathrm{y}} = \sqrt{\sigma_{33}\sigma_{44} - \sigma_{34}^2} \qquad \epsilon_{\mathrm{long}} = \sqrt{\sigma_{55}\sigma_{66} - \sigma_{56}^2}\ . \tag{69}$$

For varying particle energies the above definition of the emittances is not invariant. Instead one defines the so-called normalised emittance

$$\epsilon^{\mathrm{N}} = \gamma\,\epsilon \qquad \text{with} \qquad \gamma = \frac{p}{m_0 c} \tag{70}$$

which is conserved during acceleration.

4 Transverse Phase Space Tomography

Advanced and dedicated beam diagnostic methods represent a major ingredient in the understanding and the operation of particle accelerators like the TESLA Test Facility (TTF). The beam of the TTF linear accelerator is used to drive a SASE free-electron laser (FEL) in the VUV wavelength regime. Therefore stable and reproduceable beam conditions have to be provided. This will become particularly important in the near future when TTF enters the next stage of extension (TTF Phase II) with the final goal to have a user facility of the free-electron laser. Therefore beam properties, e.g. energy spread, bunch length, transverse and longitudinal emittances, have to be measured and monitored with high accuracy to be able to optimise the FEL performance. Several methods have been developed to measure relevant beam parameters. In this context tomographic image reconstruction of the phase space distributions of the electron beam represents one essential technique to gain basic information about the beam properties.

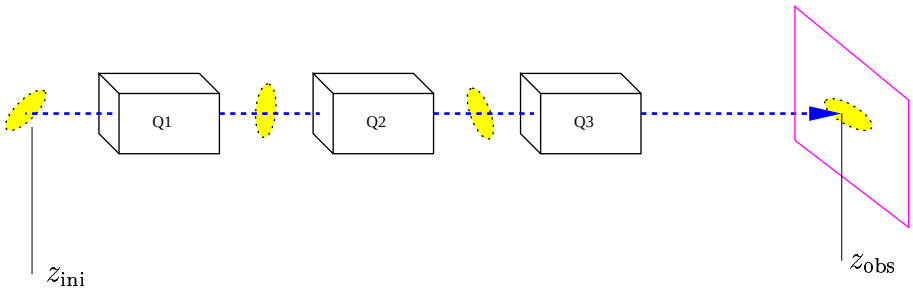

Figure 18: Sketch of a beam line with three quadrupole magnets (triplett) evoking transformations of the transverse phase space distribution between the initial position z_{ini} and the position of oberva-tion z_{obs}; the applied transformation depends on the geometrical setup of the beam line and the adjusted strength of each quadrupol magnet. The transfer matrix formalism allows the simple calculation of the transformation matrices (see section 3.2).

4.1 Experimental Setups

The transverse phase space distribution of the electron beam can be reconstructed by applying a quadrupole scan which causes rotations of the phase space distribution. A set of quadrupole magnets is used to rotate the phase space distribution between the point z_{ini} before the quadrupole magnets and the position of observation z_{obs} behind the set of quadrupoles (fi gure 18). Alterna-tively a fi xed optics can be used and the measurements of the beam profi les are done at different locations along the beam line. Both possibilities generate rotations of the transverse phase space. The intensity profi les can be measured by different types of devices.

- **Wirescanners:**
  Wirescanners [18] provide one-dimensional profi les by moving a thin wire along a fi xed axis through the beam. Thus the scattering intensity can be measured as function of the wire position. However, to get a full profi le many particle bunches have to used; each bunch provides only one measurement at a certain spatial position of the wire. The resolution is limited by the thickness of the wire and the precision of the wire positioning.

- **Optical Transition Radiation Devices:**
  Inserting a metallic screen into the beam line optical transition radiation (OTR) can be recorded which provides two-dimensional spatial images of the charge distribution. The OTR is emitted due to the passage of the charged particles from one dielectric medium into another. The radiation can be imaged by an optical system and digitised with a CCD camera. Physical details about optical transition radiation can be found for exam-ple in [8]. Compared to wirescanners OTR stations yield full two-dimensional spatial profi les already for single bunches[13]. The two-dimensional OTR profi les are numerically integrated in either x or y coordinate and the obtained one-dimensional profi les can be used to reconstruct the x–x' resp. y–y' phase space distribution. Basic limitations are re-lated for example to imaging errors, depth of fi eld, sensitivity and pixel sizes of the CCD chip.

[13] Of course, restrictions are given due to the intensity of the emitted radiation which is often too low for small bunch charges. Then CCD cameras can integrate the radiation over multiple bunches to get resonable intensity.

4.2 Transformations within the Transverse Phase Space

The transfer matrix formalism introduced in section 3.2 describes linear effects of the beam propagation along the beam line and allows the quantitative evaluation of the transfer matrix from a fixed starting point z_{ini} to the position of the measuring device z_{obs}. Using only transfer beam lines according to eq. (63) which explicitly do not couple x_1-x_2 phase space with x_3-x_4 phase space (see section 3.4) admit the independent tomographic reconstruction of $\varrho_{\mathrm{trans,X}}(x_1, x_2)$ and $\varrho_{\mathrm{trans,Y}}(x_3, x_4)$. We are interested in rotations of the phase space covering an effective range between zero and 180^o to obtain an optimised reconstruction accuracy. Exemplarily we consider the horizontal phase space, but analogous considerations are valid for the vertical phase space.

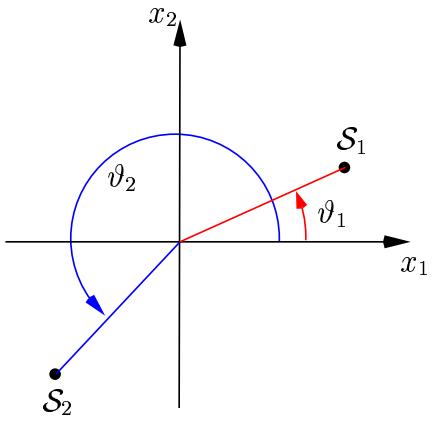

Figure 19: Assignment of the angle to a given point $\mathcal{S}$ in the x_1-x_2 plane using counterclockwise notation accordingly to eq. (72)

Tranformations are given by the 2×2 transfer matrix according to eq. (64)

$$\mathbf{M}_{\mathrm{x}} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \quad . \tag{71}$$

Each transformation can be understood as superposition of rotation and shearing. In the x_1-x_2 plane any given point $\mathcal{S}$ can be assigned to a certain angle ϑ in counterclockwise notation covering the range of values from zero to 2π as follows (figure 19)

$$\vartheta(x_1, x_2) \equiv \begin{cases} \arctan \frac{x_2}{x_1} & x_1 > 0,\ x_2 > 0 \\ \frac{\pi}{2} & x_1 = 0,\ x_2 > 0 \\ \pi + \arctan \frac{x_2}{x_1} & x_1 < 0,\ x_2 \neq 0 \\ \frac{3}{2}\pi & x_1 = 0,\ x_2 < 0 \\ 2\pi + \arctan \frac{x_2}{x_1} & x_1 > 0,\ x_2 < 0 \\ \text{undefined} & x_1 = 0,\ x_2 = 0 \end{cases} \quad . \tag{72}$$

Thus an effective, relative rotation angle $\Delta\vartheta_{\mathbf{M}}$ induced by a given transformation $\mathbf{M}_{\mathrm{x}}$ can be defined for each point $\mathcal{S} = (x_1, x_2)$ as the difference of the assigned angles for $\mathcal{S}$ and the transformed point $\mathcal{S}' = \mathbf{M}_{\mathrm{x}}\mathcal{S}^{\mathrm{T}}$ according to eq. (72):

$$\Delta\vartheta_{\mathbf{M}} = \vartheta(M_{11}\, x_1 + M_{12}\, x_2, M_{21}\, x_1 + M_{22}\, x_2) - \vartheta(x1, x2) \quad . \tag{73}$$

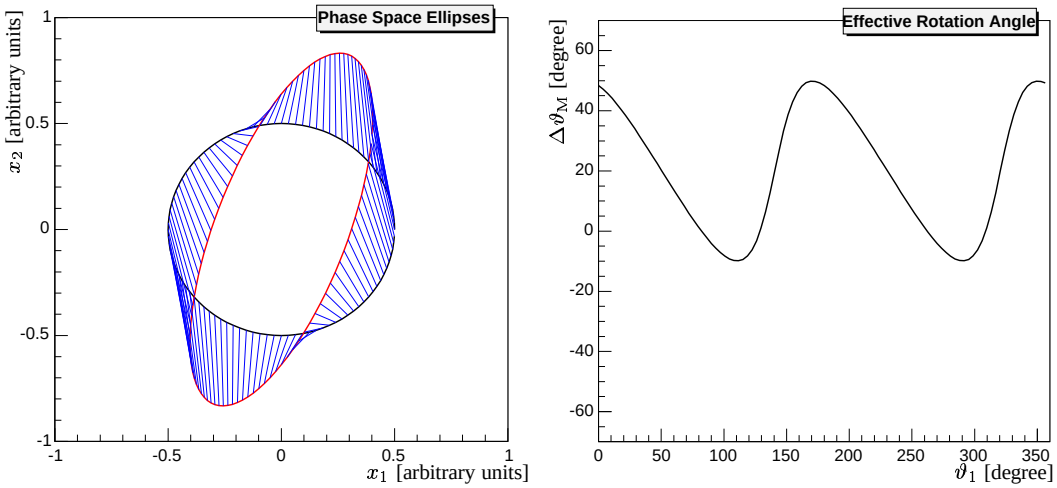

Figure 20: Two phase ellipses in the x_1-x_2 phase space (left); the transition from the circular line to the elliptical line shape results from the linear transformation according to eq. (74). The straight lines hint the changes of corresponding pairs of points. The right plot shows the induced effective rotation according to eq. (73) for each point on the circular line as function of the assigned angle from eq. (72). The shearing causes a coordinate dependent variation of the effective rotation.

For the simple case of regular rotations as given by eq. (1) and eq. (2) the effective rotation angle $\Delta\vartheta_{\mathbf{M}}$ corresponds to the inducing angle θ and is independent of the initial coordinates of the chosen point. The independence on coordinates does not hold for generalised transformations where all transformation matrix elements M_{ij} are independent. Then also shearing occurs leading to variations of the assigned relative rotation angle. This situation is sketched in fi gure 20. The left plot shows two phase ellipses in the x_1-x_2 phase space. Each point lying on the circular line has been transformed using the following matrix

$$\mathbf{M}_{\text{example}} = \begin{pmatrix} 0.8 & 0.1 \\ 0.9 & 1.4 \end{pmatrix} \quad . \tag{74}$$

The transformation leads to the elliptical deformation of the initial line shape. In the fi gure the added straight lines hint the changes of corresponding pairs of points. The variation of the effective rotation angles is evaluated in the right plot. It shows the induced rotation according to eq. (73) for each point on the circular line as function of the assigned angle from eq. (72). The induced deformation, i.e. shearing, is related to coordinate dependent variations of the effective rotation. These variations increase with the degree of deformation. Consequently the 'projection angle' is not uniquely defi ned for 'generalised' 2×2 matrices.

The described characteristics can be observed within the the transverse phase space. In this context the implemented MENT algortihm takes generalised, linear transformations correctly into account! However, the criterion of completness of input data must be reconceived. It becomes not suffi cient to consider only the effective projection angles of one single point in the phase space[14]. Systematic uncertainties certainly vary with the set of applied transformations

[14] As already mentioned for ordinary rotations according to eq. (1) we would adjust the projections to cover the range between zero and 180^o for an optimised sample of measurements.

resp. the degree of ‘deformation’. Thus transformations evoking minor effects of deformation are preferable to reduce the systematic uncertainties with respect to the issue of completness.

4.3 Error Sources

The application of the transverse phase space tomography contains various experimental problems which have to be considered to get reliable results. The following list figures out some of the dominant error sources which can falsify the reconstruction potentially:

- On the one hand all considerations and examples have assumed a stable point of reference, i.e. for each measured profile the point of origin is known. This would allow to attach all profiles to the same point of reference. On the other hand due to practical reasons the beam orbit is often not well aligned along the corresponding quadropole axis thus causing systematic shifts of the orbit during a quadropole scan. Here calculating the center of mass for each profile defines a ”pseudo” point of origin which can fix the problem but introduces uncertainties to the tomographic reconstruction.
- Possible deviations from the nominal elliptical distribution give hints to the existence of correlations[15] among the x-x' and y-y' phase space which would influence the beam propagation. In addition the transverse beam propagation is energy dependent due to the effects of chromaticity and dispersion.
- The usage of wirescanners introduce bunch-to-bunch variations into the reconstruction which also increases the systematic deviations.
- The bunch charge fluctuates from bunch-to-bunch. So all measured profiles have to be normalised to the same value.
- Bunch remnants which lie out of the timing window of interest and dark currents enter and falsify the profile measurements.

5 Longitudinal Phase Space Tomography

The longitudinal phase space of a particle bunch denotes the two-dimensional parameter space in the energy domain and the longitudinal spatial coordinate[16] domain. The determination of the longitudinal phase space distribution of a particle beam is a challenging task in accelerator physics. Present accelerator designs, e.g. the Tesla Test Facility, make time resolutions in the femto second regime necessary which lies at the frontier of existing technologies. Recently dedicated, new methods are in the focus, e.g. Electro Optic Sampling [14], to tackle this difficult task. In this context tomographic reconstruction supplements the choice of possible tools.

The measurement of energy distributions of particle bunches with a magnetic spectrometer is a standard technique in accelerator physics whereas the time domain is not directly observable

[15] Indeed such kind of deviations have beem observed at Tesla Test Facility during phase I.

[16] Of course, in the relativistic case the longitudinal spatial coordinate translates to time domain equivalently.

with these kind of measurements. However with the tomographic reconstruction technique also the time domain and the correlations among the energy domain become deducible (with some intrinsic limitations; see also [11]). Projections of the longitudinal phase space distributions onto the energy coordinate serve as input for tomography.

5.1 Transformations within the Longitudinal Phase Space

Rotations of the longitudinal phase space evolve for example from the off-crest acceleration of particle bunches within RF cavities. In principle the longitudinal density profile of the bunch converts into characteristic energy distortions during the propagation through the cavities. These distortions can be determined experimentally from the measurement of the energy spectrum. Compared to the transverse phase space the transformations within the longitudinal phase space appear slightly more complicated. Additional physical effects like coherent synchrotron radiation in the bunch compressor magnets influence the beam profile.

5.1.1 Linear Approximation

The applicable rotations depend on the phase Φ of the RF field at which the bunch traverses the cavity. The matrix formalism as introduced in sec. 3.2 describes such transformations. In the linear approximation eq. (52) the transformation represents a shearing of the longitudinal phase space with the phase Φ as adjustable parameter. However this experimental setup holds an intrinsic limitation. For any resonable phase shift the time profile cannot fully be mapped onto the energy domain, hence longitudinal transformations cannot generate full rotations up to angles of 180^o. So the experimental data remain incomplete which in turn has severe implications for the reconstruction[17]. Therefore tomographic algorithms are essential which minimise reconstruction artefacts due to the intrinsic lack of information. Here the Maximum Entropy Algorithm appears inevitable.

5.1.2 Non-linear transformations

To improve the accuracy of the reconstruction the non-linear term eq. (56) can be taken into account for all longitudinal transformations. These specific transformations, namely

$$\begin{aligned} \delta' &= M_{66}\,\delta + M_{65}\,l + M_{\mathrm{non-lin}}\,l^2 \\ l' &= l \end{aligned} \tag{75}$$

resp. for the inverse transformation

$$\begin{aligned} \delta &= M_{66}\,\delta' - M_{65}\,l' - M_{\mathrm{non-lin}}\,\left(l'\right)^2 \\ l &= l' \end{aligned} \tag{76}$$

are taken into account for the longitudinal phase space reconstruction (sec. 5.3).

[17] In a synchrotron or with a magnetic chicane the full range of rotations becomes accessible [17].

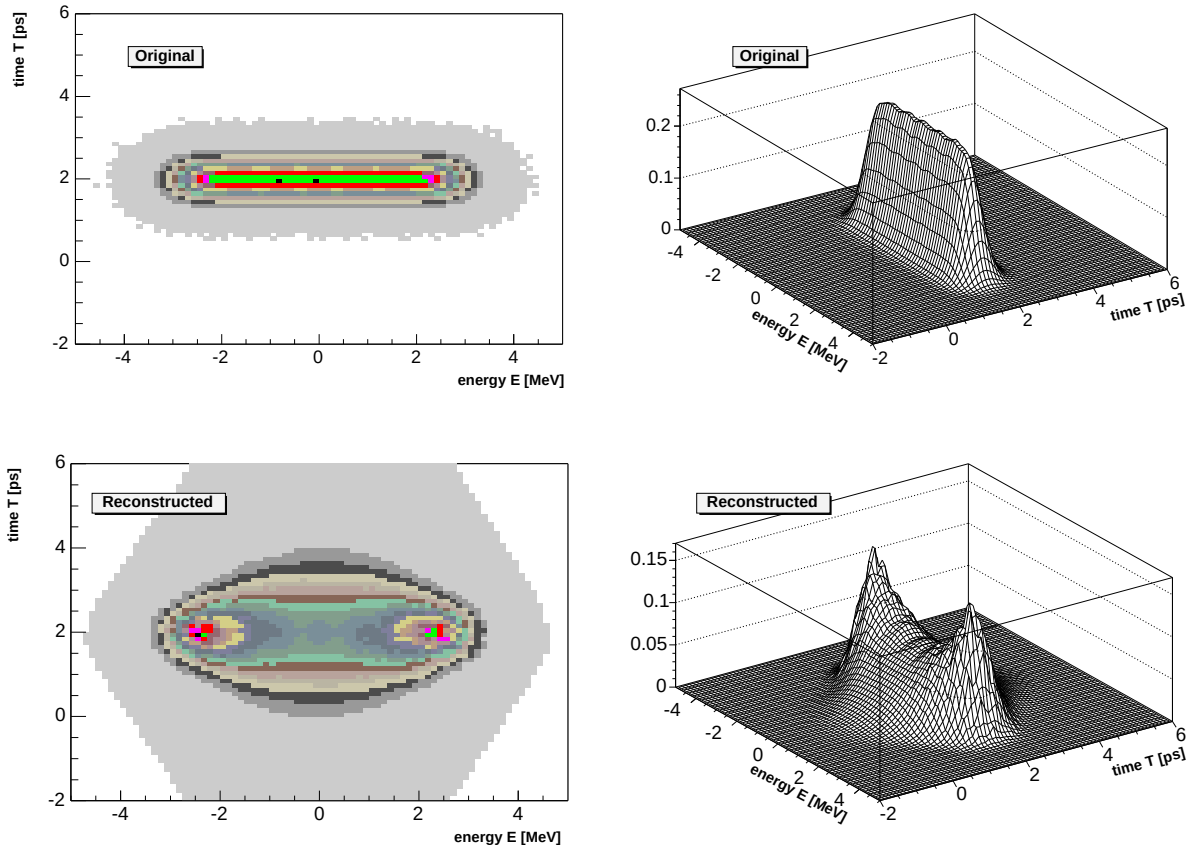

Figure 21: Demonstration of the capability to reconstruct longitudinal phase space distributions. The upper plots show the original distribution with a narrow peak in the time domain and an evenly spread energy spectrum over a larger range as contour plot (left) and isometric plot (right). The lower plots show the reconstructed distribution respectively which results from seven different simulated projection profiles for RF phases between -40^o and 40^o.

5.2 Example

Figure 21 gives an example of the capability to reconstruct longitudinal profiles for incomplete coverage of the rotation angle. The upper plots show the underlying phase space distribution which was set with a rather narrow Gaussian time profile but with an evenly spread energy profile over a larger energy range. The reconstructed profile (bottom) was calculated using seven simulated projection profiles[18] with adjusted RF phases between -40^o and 40^o.

The the two-peak structure of the reconstructed profile results from the limitation of the available projection angles. The best resolution in time is achieved at the edges of the distibution and degenerates towards the center. Futher discussions about these issues can be found for example in [11].

[18]The simulated profiles can be simply obtained using the method of class `ProjectionElement_Long`; for details see section A.3.2.

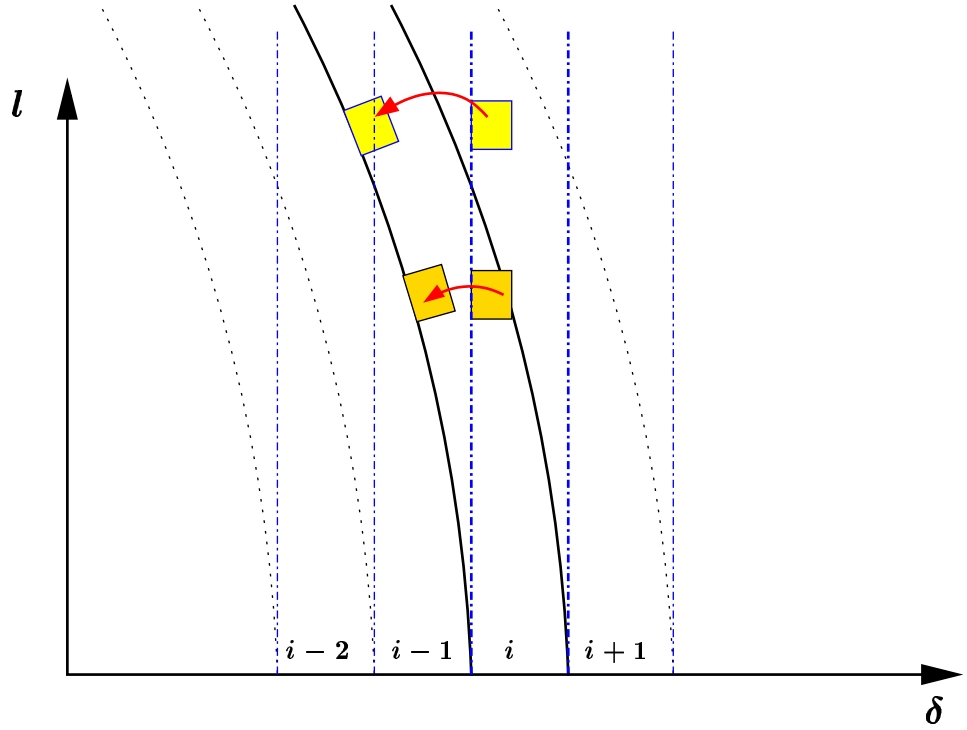

Figure 22: Sketch of phase space tranformations within the longitudinal phase space; non-linear effects cause bin boundaries with specific curvatures. Therefore the simple assumption of overlayed bins with "straight line" boundaries does not hold any longer. The curvatures of the bins under transformations can be taken into account by subdividing the energy bins into sub-bins along the time coordinate; then the small bin elements which are transformed (arrows) follow closely the curvature of the "true" bin boundaries.

5.3 Technical Adaptation of the Maximum Entropy Algorithm

The introduced non-linear transformations eq. (75) resp. eq. (76) within the longitudinal phase space cause bin boundaries with specific curvatures as illustrated in figure 22. The total contents of each curved bin represents a part of the observable energy profile, i.e. a projection of the curved bin onto the energy coordinate which can be measured by a spectrometer. However curved bins do not fit to the standard implementation of the MENT algorithm (sec. 2) which requests straight lines as bin boundaries.

However the numerical deviations become negligible if small sub-bins are introduced. Then the non-linear contributions do not accumulate over larger phase space regions. The evaluation of all sub-bins separately and adding up the resulting phase space fragments take the non-linear effects correctly into account! Exemplarily figure 22 shows two different sub-bins which have been transformed to the new frame of reference with a specific curvature. The sub-bins fit very precisely to the curved "true" bin boundaries. Without the sub-binning the deviations would integrate to sizeable effects for sufficiently strong curvatures. The C++ classes `MaxEntropy_Long` (sec. A.2) and `ProjectionElement_Long` (sec. A.4) have been especially adopted and make use of the sub-binning technique above to avoid numerical inaccuracies.

A Class Descriptions

The developed user interface provides simple access to the previously described functionality of the Maximum Entropy Algorithm. The code was written in C++ under a *LINUX* operating system on a standard PC. The general system requirements are rather poor. Only an installed library of the ROOT physics analysis package [1] is needed. The software can either be used in standalone mode or in a more flexible manner as interactive ROOT session which exploits the dictionary formalism of ROOT (see also ROOT reference manual). Additional implementation details can be found in the corresponding header files[19] of the provided C++ classes.

A.1 Class `MaxEntropy` [Iteration]

The basic class `MaxEntropy` implements the two-dimensional (2D) Maximum Entropy Algorithm which is feasible for **all** kinds of linear transformations conserving the size of any chosen area under these transformations. So not only rotational transformations with their *SO*(2) group properties but generalised transformations inducing also sheering are applicable. Consequently the class is suitable to transverse phase space tomography and allows the reconstruction of two-dimensional density distributions of the x-x' phase space or y-y' phase space respectively. The measured/simulated data are accessible through instances of the class `ProjectionElement` (sec. A.3) to the iterative reconstruction.

A.1.1 Public Methods

- `void set_projection_element(ProjectionElement *proj_elm)`
  Check in one projection element, i.e. single measurement, to be used within the MENT algorithm for the reconstruction (sec. A.3).
- `void delete_all_projections(void)`
  Delete list of all stored projection elements, i.e. data, and re-initialse the MENT algorithm for new data.
- `void Iterate_from_storage(int n_iteration)`
  Start one iteration of the MENT algorithm with all checked in projection elements. The first iteration call **must** start with the value `n_iteration=0`. Then all needed initialisations are executed. All geometrical calculations are stored residently in the tree structure before the first iteration starts. So repeated, identical calculations of geometrical elements are avoided and save CPU time for subsequent iterations.
- `void Iterate(int n_iteration)`
  In principle it does the same as method `Iterate_from_storage()`. However, the calculated polygons are not stored in memory. In fact this results in significant increase of the calculation time because of necessary geometrical calculations for each iteration. This (slow) iteration option represents an alternative in case of exhausted memory using method `Iterate_from_storage()`. Note, exhausted memory would kill the program automatically by the system kernel without comment!

[19] All header files of the software package are tagged with the extension ".hh".

➪ `TH2D_my* reconstruct_histo2D_from_Hspace (int nbinsX, int nbinsY, char *name, int n_smooth)`
This method provides two-dimensional ROOT histograms[20] of the reconstructed distribution. It is calculated from the actual values of the Lagrange Multipliers of the recent iteration. The user defines the number of bins in the both coordinate axis. The parameter `n_smoothing` can optionally be defined. The default `n_smoothing=0` gives the precise reconstructed value in the user-defined rectangular grid, i.e. each bin contains the exact integral over the occupied bin area divided by its size (fig. 8). Visualisation artefacts due to the chosen binning are excluded. For `n_smoothing>0` it defines the number of sub-bins within each bin where the reconstructed function is sampled.

➪ `void test(void)`
Complete test example of the MENT algorithm with simple, simulated distributions using ordinary rotations from 0 degree to 180 degree. The code of this method documents the logical order of all relevant method calls for the reconstruction.

A.2 Class `MaxEntropy_Long` [Iteration]

The class `MaxEntropy_Long` contains few specific code adaptations to cope also with the **non-linear** transformations occuring in Longitudinal Phase Space Tomography (sec. 5). Basic difference compared to class `MaxEntropy` is the additional sub-binning within each data bin. Non-linear effects can be regarded as linear in the examined sub-bins. So each bin can accurately be treated under longitudinal phase space transformations provided that the sub-bins are chosen reasonably small. The non-linear effects do not accumulate to sizeable deviations over larger ranges of the phase space (see also sec. 5).

For the user the handling does not differ compared that of class `MaxEntropy` (sec. A.1). Only the data members are slightly modified and instances of class `ProjectionElement_Long` have to be used instead of class `ProjectionElement`. The class `ProjectionElement_Long` contains additional information about the non-linear matrix elements of longitudinal phase space transformations (sec. A.4).

A.3 Class `ProjectionElement` [Data Representation]

This class defines the data interface to the class `MaxEntropy`. Each class instance makes a measurement under fixed transformation conditions available to the reconstruction. A specific instance contains a one-dimensional intensity distribution. Also the transformation conditions refering to the profile measurement have to be given. A two-dimensional matrix defines the transformation of the initial distribution (to be reconstructed) to the observed distribution at the position of the measuring device.

For simulation purposes an initial two-dimensional distribution can be defined which is used to analytically calulate the one-dimensional projection profile. The profile is treated analogously to measurements. It enters the MENT algorithm and the unfolded 2D distribution can be compared to the known original distribution. The simulation features have extensivly been used for tests of the correctness of the software.

[20] Class TH2D_my inherits from the ROOT class TH2D but has additional functionality (sec. A.7).

A.3.1 Puplic Methods for Data Access

- `void set_matrix_X(Matrix M)`
  `void set_matrix_Y(Matrix M)`
  Define a 2×2 transformation matrix extracted from the 6×6 resp. 4×4 beam transfer matrix in the x-x' phase space or y-y' phase space respectively (sec. 3.2).

- `void set_rotation_matrix(double angle_degree)`
  Define a 2×2 rotation matrix as transformation matrix only by giving the rotation angle.

- `void set_id(int id)`
  Set an identification code (integer value) of the instance. The ID is necessary to distinguish different measurements for the geometrical calculations. The value is arbitray but must be unique! Consecutive numbering of data sets is recommended.

- `void set_projection_data(Matrix M, TH1D* histo1D, int id)`
  Define a complete (single) measurement by giving all relevant components simultaneously, namely the transformation matrix, the measured data represented by 1D ROOT histogram and a unique identification code.

- `double projection_normalisation(void)`
  Calculate the normalisation of the 1D intensity distribution.

- `void clip(void)`
  Skip all bins of the 1D input distribution at the left/right border which are 0 (<1E-10) to reduce calculation effort without affecting the reconstruction results.

- `int number_of_bins(void)`
  Return the actual number of valid bins.

- `double get_xmax(void)`
  `double get_xmin(void)`
  Return the valid range of the measured data.

- `double get_ymax(void)`
  `double get_ymin(void)`
  Return the valid range of the unobservable coordinate within 2D space! In principle these values are unknown for a single measurement. However, the overlay of different measurements restricts the 2D space also in the complement coordinate. Class MaxEntropy calculates the range and set the values automatically via methods `set_ymax()` and `set_ymin()` before reconstructing the given profiles.

- `double get_G_data(int index)`
  Return the data value of a specific bin of projection data (`G_data`). The valid range of `index` goes from 0 to (`number_of_bins()-1`).

- `void reset_G_data(int index)`
  Return the data value of a specific bin of projection data (`G_data`). The valid range of `index` goes from 0 to (`number_of_bins()-1`).

- `double get_xlow(int index)`
  `double get_xup(int index)`
  `double get_xwidth(int index)`
  Return the bin boundary properties of specific data bin. The valid range of `index` goes from 0 to (`number_of_bins()-1`).

- `TH1D* get_projection_histo1D(void)`
  Create a new 1D ROOT histogram of the projection data (`G_data`) for purposes of visualisation. The ownership of the returned histogram goes beyond the scope of the producing instance! In addition method `TH1D* get_projection_histo1D(TAxis* axis)` creates a rebinned histogram. The ROOT object TAxis defines the binning.

- `void show_projection(void)`
  Show the 1D ROOT histogram of the projection data (`G_data`) within a instance's private canvas.

A.3.2 Puplic Methods concerned with Simulation

The following methods have been implemented to realise simulations as verification of the MENT algorithm capabilities.

- `void set_initial_histo2D(TH2D_t* histo2D)`
  Define a 2D histogram containing a global INITIAL 2D distribution. It can be used to "simulate" the measured data by calculating its projection with respect to the defined transformation of the instance. The instance copies the histogram and will not be owner of the offered histogram (`histo2D`).

- `void calc_transformation_from_initial_histo2D`
  `(int nbinsX, int nbinsY)`
  Transform the initial 2D histogram (defined with method `set_initial_histo2D()`) to the final position using the prevoiusly defined transfer matrix. The values of `nbinsX` and `nbinsY` define the number of bins in both coordinates for the transformed histogram. For the transformation the exact bin boundaries are taken into account, i.e. the analytical calculations avoid numerical inaccuracy. From the simulated 2D histogram the projection is calculated and filled into the projection data structure (`G_data`) used for the reconstruction.

- `TH2D_t* get_transformed_histo2D(void)`
  Return the 2D histogram calculated by `calc_transformed_2Dhisto_from_initial_histo2D()`. The ownership of the returned histogram goes beyond the scope of the producing instance!

A.4 Class `ProjectionElement_Long` [Data Representation]

This class defines the data interface for the baisc class `MaxEntropy_Long` (sec. A.2). Compared to class `ProjectionElement` it contains some minor adoptions related to the specific properties of longitudinal phase space transformations (sec. 5) and the data handling does only slightly differ with respect to the definition in section A.3. The modified public methods are described below:

- `void set_trafo(Matrix M, double A13)`
  Define the transformation ensemble consisting of a 2×2 transformation matrix and an additional matrix element A_{13} which describes a specifc longitudinal phase space transformation corresponding to eq. (75). The basic adoptions are mainly related to the methods
  - `polygone_trafo()`
  - `get_point_transformed()`
  - `get_point_transformed_invers()`.
- `void set_projection_data (Matrix M, double A13, TH1D* hist1D, int id)`
  Define a complete (single) measurement by giving all relevant components simultaneously, namely the transformation matrix, the additional matrix element, the measured data represented by 1D ROOT histogram and a unique identification code.

A.5 Class `Polygon` [Geometry Calculations]

The class `Polygon` is implemented as chained list of 2D points. After constructor initialisation the list is empty. Then the polygon can be constructed by successively adding points to a class instance. The order of the points within the list determines the frequency polygon, i.e. which point has a link to the subsequent point. To get a *closed* polygon the last point in the chained list is assumed to be related to the first point of the list. All declarations and detailed comments can be found in the corresponding header file. The MENT algorithm intensively uses the class features, but there is no need to directly access class `Polygon` for the standard reconstruction.

A.5.1 Puplic Methods

- `void add_at_first(double X, double Y)`
  Add an additional point with coordinates `(X,Y)` at the beginning of the chained list of points.
- `Polygon* clone(void)`
  Provide an exact copy of the polygon without changing the original polygon.
- `int get_npoint(void)`
  Get the number of points of the polygon. Note, for `get_npoint()`<3 the polygon representation is degenerated.

- `double contents(void)`
  Calculate the area of the full polygon by summing up all sub-triangles.

- `Polygon* split(double X0)`
  Split the current polygon along the line with `X=X0` and `Y=`$(-\infty, \infty)$. The part of the polygon with `X<X0` remains in the current instance. The part of the polygon with `X>X0` forms an independent instance which returns. Note, the returned pointer to the new instance can also be `NULL` and the current instance can contain zero points (degeneration) after execution of method `split()`. This is important for the consecutive exception handling.

- `Polygon* split_Y(double x0)`
  Does the same as `split()` but for the complementary coordinate.

- `void shrink_Xrange(double Xmin, double Xmax)`
  `void shrink_Yrange(double Ymin, double Ymax)`
  Shrink the polygon to the given values in `X` resp. `Y` coordinate. The methods have no effect, if the defined ranges are larger than the occupied range of the polygon.

- `int count_points(void)`
  Move explicitly through the chained list of points and count points.

- `void polygon_trafo(Matrix* M)`
  Transform polygon to new coordinates using the defined 2×2 transformation matrix.

- `void get_extrema`
  `(double &Xmin, double &Xmax, double &Ymin, double &Ymax)`
  Return the dimensions of the current polygon.

- `TPolyLine* get_polyline(int Color)`
  Create and return pointer of an instance of the ROOT class `TPolyLine` containing all points of the point list. `TPolyLine` can be drawn and visualises the geometry of current polygon.

A.6 Class `Matrix`

This class was developed to handle simple operations on any kind of $n \times n$ matrices (e.g. matrix multiplications, determinant calculation, etc.). Details about the class functionality can be found in the header file `Matrix.hh`.

A.7 Class `TH2D_my`

This class inherits and extends the functionality of the ROOT class `TH2D` which allows the handling of 2D histograms. The following useful methods have been added:

- `void normalise()`
  Normalise the 2D histogram to unit (value of 1).

- `double get_normalisation()`
  Calculate the current normalisation.

- `double integral_over_polygon(Polygon* &P)`
  Calculate the exact (!) integral of the 2D histogram over the area occupied by the given polygon. All relevant histogram bins are weighted with the area fractions intersecting with the polygon. All products of bin content times area fraction are summed up.

- `void add_volume_element(Polygon* &P, double function_value)`
  Add to each occupied bin the value of `function_value` $\times A_{frac}$

- `void clear(void)`
  Reset all bins of the histogram.

- `TH1D* projection_X(void)`
  `TH1D* projection_Y(void)`
  Create a 1D histogram containing the projection to the X axis resp. Y axis.

- `TH2D_my* rebinned_histo(int nbinsX, int nbinsY)`
  Rebin the 2D histogram within an equidistant grid of `nbinsX` resp. `nbinsY` bins.

- `TH2D_my* transformed`
  `(Matrix* M, char *name, int nbinsX, int nbinsX)`
  Transform complete histogram with respect to defined 2×2 transformation matrix and rebin the transformed 2D histogram. Exact bin boundaries are taken into accout, i.e. numerical artefacts are avoided!

- `TH2D_my* rotate`
  `(double theta_degree, char *name, int nbinsX, int nbinsY)`
  As method above, however introduce a simple rotation with an angle of `theta_degree`. Exact bin boundaries are taken into accout, i.e. numerical artefacts are avoided.

- `TH2D_my* exchanged_XY(void)`
  Exchange X and Y coordinates of the current histogram.

- `void calculate_moments(void)`
  Calculate the first and second moments of 2D histogram distribution. The values can be printed with method `print_moments(void)`.

- `double deviations(TH2D_my* histo2D)`
  `double deviations(TH2D *histo2D)`
  Calculate deviations with respect to reference histogram corresponding to eq. (42).

- `void draw(void)`
  Draw the current 2D histogram onto private canvas.

References

[1] R. Brun, F. Rademakers, CERN division, `http://root.cern.ch`

[2] J. Radon, *Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten*, Ber. Verh. Saechs. Akad. Leipzig, Math. Phys. Kl. **69** (1917) 262.

[3] B. Jähne, *Digitale Bildverarbeitung*, Springer-Verlag Berlin, 1997, ISBN 3-540-61379-X.

[4] D.W. Townsend, M.Defrise, *Image Reconstruction Methods in Positron Tomography*, CERN Yellow Report **93-02** (1993) 1-43.

[5] F. Natterer, *Tomography*, New York, John Wiley & Sons Inc., 1993

[6] G. Minerbo, *MENT: A Maximum Entropy Algorithm for Reconstruction a Source from Projection Data*, Computer Graphics and Image Processing, **10** (1979), p. 48-68

[7] R. Bowley, M. Sanchez, *Introductory Statistical Mechanics*, Oxford Science Publications, ISBN 0-19-851193-9;
W. Greiner, L. Neise, H. Stöcker, *Thermodynamik und Statistische Mechanik*, Band 9, Verlag Harri Deutsch

[8] M. Geitz, *Investigation of the Transverse and Longitudinal Beam Parameters at the TESLA Test Facility Linac*, DESY-THESIS-1999-003, Deutsches Elektronen Synchrotron, Nov. 1999

[9] H. Vogel, Gerthsen Physik, 18. Auflage (1995), Springer Verlag, ISBN 3-540-59278-4

[10] B.R. Frieden, *Restoring with maximum likelihood and maximum entropy*, J. Opt. Soc. Amer. (1972), p.511-518;
E.T. Jaynes, *Prior Probabilities*, Trans. IEEE, SSC4, (1968)

[11] M. Hüning, *Analysis of Surface Roughness Wake Fields and Longitudinal Phase Space in a Linear Electron Accelerator*, DESY-THESIS-2002-029, Deutsches Elektronen Synchrotron, Aug. 2002

[12] P. Schmüser, J. Rossbach, *Basic Course on Accelerator Optics*, DESY M-93-02, Internal Report, Feb. 1993

[13] H. Wiedemann, Particle Accelerator Physics, 1993, Springer Verlag

[14] M. Brunken *et.l.*, *Electro-Optic Sampling at the TESLA Test Accelerator: Experimental Setup and First Results*, TESLA Report 2003-11, Deutsches Elektronen Synchrotron, Feb. 2003

[15] H. Schlarb, *Collimation System for the VUV Free-Electron Laser at the TESLA Test Facility*, DESY-THESIS-2001-055, Deutsches Elektronen Synchrotron, Nov. 2001

[16] N.V. Denisova, Maximum-entropy-based tomography for gas and plasma diagnostics, J. Phys. D: Appl. Phys.**31** (1998) 1888-1895

[17] S. Hancock, M. Lindroos, *Longitudinal phase space tomography with space charge*, Phys. Rev. ST Accel. Beams (2000) 3;
K.N. Ricci, T.I. Smith, *Longitudinal electron beam and free electron laser microbunch measurement using off-phase acceleration*, Phys. Rev. ST Accel. Beams (2000) 3

[18] N. von Bargen *et. al.*, *Prototyp eines Wirescanners für TTF II*, TESLA Report 2002-08, Deutsches Elektronen Synchrotron, Aug. 2002