# Appendix B

# Auto-Tuning Algorithm

The optimal potentials for the lenses and deflectors of the electron monochromator were set under computer control using a Nelder-Mead Simplex Algorithm (NMSA), a technique which is similar to one reported in a previous study [95]. The NMSA is a 'direct search' algorithm for numerically optimising an objective function in N-dimensional space. Direct search algorithms optimise an objective function based solely on function evaluations at different points in parameter space, with no reliance on information about the derivative of the objective function. The auto-tuning algorithm could simultaneously vary up to 12 monochromator potentials in order to maximise the current recorded at the either the Faraday Cup or at one of the internal current pickups within the monochromator.

The logic of the NMSA is described in detail elsewhere [96]. However, optimising the monochromator in real time required some additional logic to that used for optimising an analytical function. The optimal monochromator potentials were found to vary somewhat with time. Further, in some cases the optimal settings actually changed significantly during the course of an autotune. This was particularly prevalent when simultaneously tuning a large number of monochromator elements (more than six), or if the filament current had only recently been turned on. The NMSA therefore needed to incorporate logic that was able to recognise when the optimal setting had 'drifted', and adjust accordingly. To facilitate a discussion of the logic that was incorporated into the NMSA, the logic of a standard maximising NMSA is briefly described below. A flow diagram illustrating these steps is also given in Figure B.0.1:

1. **Generate Simplex**: Given an initial 'guess' vector $\mathbf{x_0}$ (here the elements of $\mathbf{x_0}$ are the monochromator potentials) in N-dimensional parameter space, generate a closed, non-zero volume of N+1 vertices (hereafter referred to as 'the simplex'). This was done simply by adding values $(\lambda_i, i = 0...N-1)$ to the *ith* component of $\mathbf{x_0}$, where each $\lambda_i$ is a characteristic scale.

2. **Order Simplex**: Order and re-label each vertex of the simplex $\mathbf{x_0}$,...,$\mathbf{x_n}$, such that $F(\mathbf{x_0}) > ... > F(\mathbf{x_n})$. Here F($\mathbf{x}$) is the objective function to be maximised. Denote $\mathbf{x_0}$ as the best point, and $\mathbf{x_n}$ as the worst point.

3. **Compute Centroid**: Compute the Centroid ($\overline{\mathbf{x}}$) as:

$$\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=0}^{n-1}\mathbf{x_i}. \tag{B.0.1}$$

4. **Reflect**: Compute the reflection point $\mathbf{x_r}$ as:

$$\mathbf{x_r} = \overline{\mathbf{x}} - \alpha\left(\overline{\mathbf{x}} - \mathbf{x_n}\right), \tag{B.0.2}$$

   where $\alpha$ is the reflection coefficient. Evaluate F($\mathbf{x_r}$). If $F(\mathbf{x_0}) > F(\mathbf{x_r}) > F(\mathbf{x_n})$, replace $\mathbf{x_n}$ with $\mathbf{x_r}$ and go to step 8. Otherwise, if $F(\mathbf{x_0}) \leq F(\mathbf{x_r})$, go to step 5. If neither of these criteria are met, go to step 6.

5. **Expand**: Compute the expansion point $\mathbf{x_e}$ as:

$$\mathbf{x_e} = \mathbf{x_r} - \beta\left(\mathbf{x_r} - \overline{\mathbf{x}}\right) \tag{B.0.3}$$

   where $\beta$ is the expansion coefficient. Evaluate F($\mathbf{x_e}$). If $F(\mathbf{x_e}) > F(\mathbf{x_r})$, replace $\mathbf{x_n}$ with $\mathbf{x_e}$ and go to step 8. Otherwise, replace $\mathbf{x_n}$ with $\mathbf{x_r}$ and go to step 8.

6. **Contract**: Compute the contraction point ($\mathbf{x_c}$) as:

$$\mathbf{x_c} = \overline{\mathbf{x}} - \zeta\left(\overline{\mathbf{x}} - \mathbf{x_n}\right), \tag{B.0.4}$$

   where $\zeta$ is the contraction coefficient. Evaluate F($\mathbf{x_c}$). If $F(\mathbf{x_c}) > F(\mathbf{x_n})$, replace $\mathbf{x_n}$ with $\mathbf{x_c}$ and go to step 8. Otherwise, go to step 7.

7. **Shrink**: Replace all the vertices $\mathbf{x_1}$,...,$\mathbf{x_n}$ by new vertices evaluated as:

$$\mathbf{x_i} = \mathbf{x_0} + \eta\left(\mathbf{x_i} - \mathbf{x_0}\right), \tag{B.0.5}$$

   where $\eta$ is the shrinking coefficient. Now go to step 8.

8. **Check Terminal Condition**: Check to see if the function values at the vertices of the current simplex satisfy a predetermined condition. If yes, return $\mathbf{x_0}$ as the optimal point with F($\mathbf{x_0}$) as the optimal value. Otherwise, return to step 2.

The values for the reflection, expansion, contraction and shrinking coefficient were chosen to minimise the number of iterations of the NMSA, and were determined experimentally. For this research the values chosen were $\alpha = 1$, $\beta = 2$, $\zeta = 0.5$ and $\eta = 0.5$. The initial guess vector for the NMSA was the manually tuned monochromator settings, obtained prior to commencement of auto-tuning.

To evaluate the objective function value at a given vertex, those potentials that formed the vertex in question were applied to their corresponding monochromator elements, and the current to be optimised was recorded. The routine was terminated when the standard deviation of the function values at each vertex was lower than a certain tolerance, usually set as 2% of the best value, with the best vertex taken to be the optimal setting for those monochromator elements. Alternatively, the routine would abort if a predetermined number of iterations was exceeded (typically 50), or the routine could be manually aborted at any time. If an autotune was terminated without locating an optimal setting, the returned value would be the best value when the search was aborted. Each monochromator element had an individual characteristic scale when generating the simplex, to reflect that the performance of the monochromator had different sensitivies to different elements.

To allow the NMSA to monitor for real time drifts, a number of cross-checks were added. In the above logic, the NMSA only changes its best point when it finds a better one. If the optimal settings changed during an optimisation,

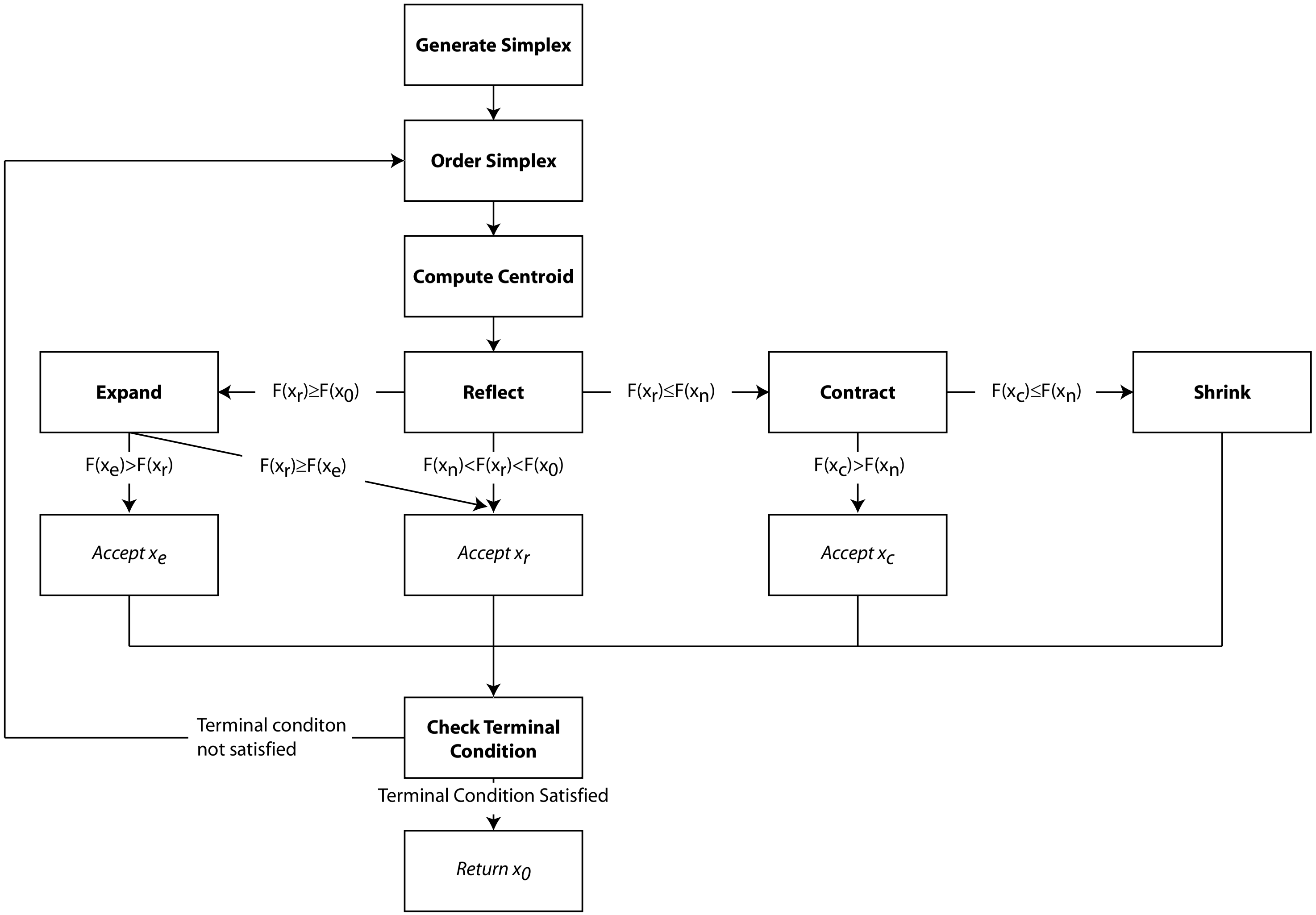

Figure B.0.1: Flow diagram of the logic for a maximising NMSA. The symbols and operations are explained in the text.

the simplex might simply get stuck in the vicinity of the old 'best' point, which would never be re-evaluated. To prevent this from happening, the best point was re-evaluated when the routine reached the 'Shrink' step. A second cross check that was added was to evaluate the standard deviations of the individual potentials across each vertex when checking the terminal condition. If the terminal condition was not satisfied, but the range of potentials over each vertex was less than 2% of the characteristic scale for every monochromator element, the simplex was regenerated using the initial weights and the current best point. This condition prevented the volume of the simplex from becoming too small, which would restrict the NMSA's ability to search quickly through parameter space. The final precaution was to sample the objective function value a number of times when evaluating each point. To obtain the value of any sample, the objective function was logged at 10 kHz, for 500 ms, and all values averaged. Subsequent samples were then taken 500 ms apart. The function value was not accepted until a number of successive samples (usually 5), had a standard deviation lower than some limit (usually 5% of the mean value). This precaution ensured that the current had stabilised before being accepted.

The tuning time was critically important, as problems such as drifts were exacerbated when the software took longer than necessary to find an optimal setting. A judicious choice of the $\lambda_i$'s for each monochromator element was crucial to ensure that the NMSA could tune the monochromator as quickly as possible. By setting the value for any $\lambda_i$ too large, the simplex could search a region of parameter space that was too distant from the optimal value. The NMSA would typically then reach the 'Shrink' step at a number of successive iterations, in order to reduce the scale. Autotuning was much slower if the routine was frequently required to 'Shrink', since all vertices were then re-evaluated, making a Shrink the slowest step. Alternatively, if a $\lambda_i$ value was too small, the NMSA would tend optimise on small local maxima, rather than effectively search parameter space for the global maxima.

Auto-tuning by the computer was only possible with a non-zero current, either from the Faraday cup or some other current pickup to be optimised. Therefore, the monochromator needed to be manually tuned in part before using the computer to find the optimal setting. However, once partially tuned, the computer was usually able to find the optimal setting for 10-12 monochromator elements within about 20 minutes. This compared with a period of several hours to arrive at optimal settings when tuning manually. The currents produced by autotuning also tended to be larger, in some cases by up to a factor of 4, than those produced by manual tuning. This reflects the advantage in having the computer simultaneously varying multiple potentials, as opposed to manually varying one at a time.