# Chapter 6

# Optimization methods for IbM parameter estimation

## 6.1 Introduction

Individual-based Modelling and simulation of microbial systems may have two aims. On the one hand, they are used to improve the comprehension of the system's behaviour. On the other hand, once they have been validated, they may be used as virtual experiments in order to predict the behaviour of a system under certain conditions.

The use of IbM to improve understanding of a microbial system is the main utility of this methodology (Dens et al., 2005a; Dens et al., 2005b). An estimation of the input parameters of the simulations is necessary to study the real behaviour in depth, since the use of an inadequate set of input parameters may produce incoherent and artificial results. Therefore, if the behaviour of a microbial system is studied in the framework of the real parameters, the detection of a non-predicted behaviour contributes to the understanding of that system.

This thesis aims to improve understanding of the lag phase through IbM simulations. In the previous chapter, the soundness of INDISIM predictions and, therefore, the validity of the model behind the simulator were qualitatively checked by means of experimental measurements. In order to improve our understanding of the mechanisms that cause the lag phase, further experiments were also proposed. In this chapter we will define a methodology that should allow for the setting up of a quantitative relationship between the experimental work and the IbM simulations.

The predictive side of the IbM requires an accurate setting of the involved parameters, since the IbM simulations are treated as virtual experiments (Hilker et al., 2006). Therefore, specific and numerical results for delimited conditions are expected. Input data must be perfectly fitted to the microorganism strain, the medium and the conditions that are to be reproduced. The parameter estimation must be carried out carefully to obtain proper predictions. However, in this case the continuous models are usually more efficient in terms of time expenditures than IbM simulations.

In general, the input parameters may be obtained from two different sources. The bibliography may give values that are based on experimental data or theoretical analysis. Some of the parameters that are not found in the literature may be estimated by performing *ad-hoc* experiments. The parameters that can not be found in the literature nor measured through experiments must be estimated with optimization methods.

There is a third field, related to the understanding nature of the IbM, where IbM parameter estimation acquires a special significance. It is the use of an IbM simulation to quantify the unknown parameters that are difficult to measure. That is, the IbM parameter estimation may be an aim in itself. Input data in IbM simulations are mainly related to individual bacterial properties that are difficult to determine experimentally. Therefore, estimation may be useful to generate a pool of microbial data. This is the parameter estimation *with* an IbM, which goes beyond mere adjustment of the model.

In previous chapters we have seen that INDISIM simulations qualitatively reproduce the evolution of the bacterial cultures during the transient phases. In this chapter we will see that INDISIM simulations also allow the obtaining of quantitative knowledge.

Several methods for parameter estimation have been developed in the framework of continuous modelling. However, they are not usable with IbM because they are generally based on gradient methods, which are not applicable in IbM because of their discrete nature.

The aim of this chapter is to adapt, test and evaluate the usefulness of different parameterization methods for IbM. The work behind this chapter was carried out at BioTeC (Bioprocess Technology and Control research group, Katholieke Universiteit Leuven), during a stay from January to April 2007. It was a collaboration with Dr. Arnout Standaert and Dr. Kristel Bernaerts, with the supervision of Dr. Jan Van Impe, and the parameterization models were tested with BacSim (Kreft et al., 1998) and INDISIM in parallel, obtaining similar results.

In this chapter, three methods are adapted and tried out for estimating one, two and three parameters. We will also establish a general framework for performing parameter estimation on any IbM set.

The optimization processes require a huge number of simulations. Therefore, a simplified version of INDISIM is used and some slight modifications in the model are made in order to facilitate the comparison with BacSim results. Nevertheless, this chapter presents a methodological development. Neither the INDISIM version used nor the input parameters are significant for the final conclusions. Once the most appropriate method is identified, adapted and implemented for use in INDISIM parameter estimation, it will be used in the future to quantify specific parameters in a thoroughly tested version of INDISIM for the specific case.

### 6.1.1 Input indexes and parameters of an IbM simulation

Individual-based Models are bottom-up approaches. Basically, when the simulations are performed, microscopic data are input and macroscopic results are obtained. The input data may be divided in two groups:

1. Indexes and *non-physical* parameters, which are part of the programming scheme. They are dimensionless parameters, for instance:
   (a) dimension parameters, to set the arrays and matrices;
   (b) *choice indexes*, that allow the user to choose between different options such as to agitate or not, opened or closed culture or the kind of metabolism, among others;
   (c) random seeds, to generate the random numbers during the simulation.
2. *Physical* parameters, which can be also classified in two sub-groups:
   (a) Intrinsic parameters of the spatial or biological model such as uptake constants, mass to initiate the reproduction cycle, cell cycle duration, yields or diffusion constants, among others.
   (b) Characteristic parameters of a certain simulation: initial conditions of the inoculum and the environment (initial biomass distribution or nutrient amount), and other conditions that define the simulation (maximum duration or maximum number of bacteria and nutrient fluxes, among others).

The first class of parameters is inherent to the programming strategies, so there is no need for estimating them. The initial conditions of the inoculum and the environment, as well as the conditions that define the simulation, are also mainly defined by the system to be reproduced. However, some particular conditions may be difficult to evaluate - for

instance, the initial conditions of the inoculum. Finally, the intrinsic parameters of the model must be set. Some of these may be determined either from the bibliography or the experimental measures.

Nevertheless, some of these parameters may be difficult to obtain, due to the essence of the IbM. For instance, the uptake constants or the mass to initiate the reproduction cycle for the specific strain are unlikely to be experimentally measured or to be found in the bibliography. Moreover, in many cases the experimentally measured macroscopic parameters do not have a direct equivalence to the microscopic parameters. Actually, when we relate macroscopic measurements to microscopic parameters we are using a model; the soundness of this relationship depends on the soundness of the model used.

Therefore, when some parameters are difficult or impossible to fix from the literature or the experiments, the parameter estimation of the IbM is necessary to adjust the simulations to the real systems.

### 6.1.2 Basis of the IbM parameter estimation

To estimate the parameters of an IbM, some steps must be taken. First of all, it is necessary to set the values that can be deduced from the literature or induced from experimental data, which are the most arguable sources. The parameters to be estimated should be as low as possible to reduce the parameterization time. So this step concludes with the identification of the parameters to be estimated. It is also useful to know some thresholds of these values or at least they order of magnitude. Since we are talking about IbM models of microbiological systems, the parameters to be estimated must be biological or physical values with their units. Therefore, the literature should allow a delimitation of their possible values.

To carry out the estimation of the chosen parameters, experimental information about the system is necessary. That is, the logical estimation process is based on the fitting of the unknown parameters so that simulation results are as similar as possible to an experimental dataset. So, the second step is the choice of the experimental dataset (or datasets) to be used. The IbM simulations must be adapted to generate results that are comparable to these data.

In a third step, a numerical method to evaluate the soundness of the simulations regarding the experimental data is needed. This numerical evaluation is usually an *objective function* that decreases with the soundness of the fitting. Typically, the sum of squared errors (SSE) or the mean square error are used (MSE) (Standaert, 2007).

Once the three actors are on the table (IbM and parameters to be estimated, experimental dataset and objective function), the parameterization process can start. The aim

is to find the parameter values combination that gives the lowest value for the objective function. At this moment, a method to perform the search for the lowest value in a systematic manner is needed.

## 6.2 Parameter estimation with INDISIM

### 6.2.1 Experimental dataset

The experimental dataset has been taken from Bernaerts (2002). It corresponds to an experiment performed with *Escherichia coli* K12 MG1655, with the conditions detailed below (Bernaerts, 2002).

The strain was stored at $-80\,^{o}C$ in Brain Heart Infusion (BHI) broth (Oxoid) supplemented with 25% glycerol (Acros). Inocula were prepared by transferring a loopful of the stock culture to $20\,ml$ BHI in a flask of $100\,ml$. This inoculated flask was placed on a rotary shaker at $175\,rpm$ in a temperature incubator (Temarks, model KBP6151). The culture was then allowed to grow for $24\,h$ at $18\,^{o}C$. Then, $100\,ml$ of the cell suspension was transferred to $20\,ml$ fresh BHI, and incubated for $18\,h$ under the same conditions. Cells within a late-exponential or early stationary growth phase were obtained and inoculated into a new BHI medium.

The initial pH of the culture medium before inoculation was set to 7.55 by addition of KOH after autoclaving. The initial cell concentration after inoculation was around $10^4\,CFU/ml$. The *E. coli* growth was performed in a bioreactor (BIOFLO III, New Brunswick Scientific Inc., US) that maintained the temperature at $27.5\,^{o}C$. The pH was also kept constant (7.55), and the agitation speed was fixed at $400\,rpm$.

After serial dilution of the cell suspension in BHI broth, the appropriate dilution was surface-plated on BHI agar using a spiral plate (Eddy Jet IUL Instruments S.A., Spain). Plates were incubated for $18-24\,h$ at $37\,^{o}C$ and enumerated to obtain the $CFU/ml$ counts.

The resulting curve is shown in Figure 6.1. The parameter estimation will be carried out in two steps. First of all, only the exponential phase data will be used (Fig. 6.2). Then, the lag and exponential phase data will be used.

### 6.2.2 Objective function

A function that quantifies the soundness of the simulations is needed as an objective function for the parameter estimation methods. This function must reflect whether the simulation results fit the experimental data well. A common function that is used for

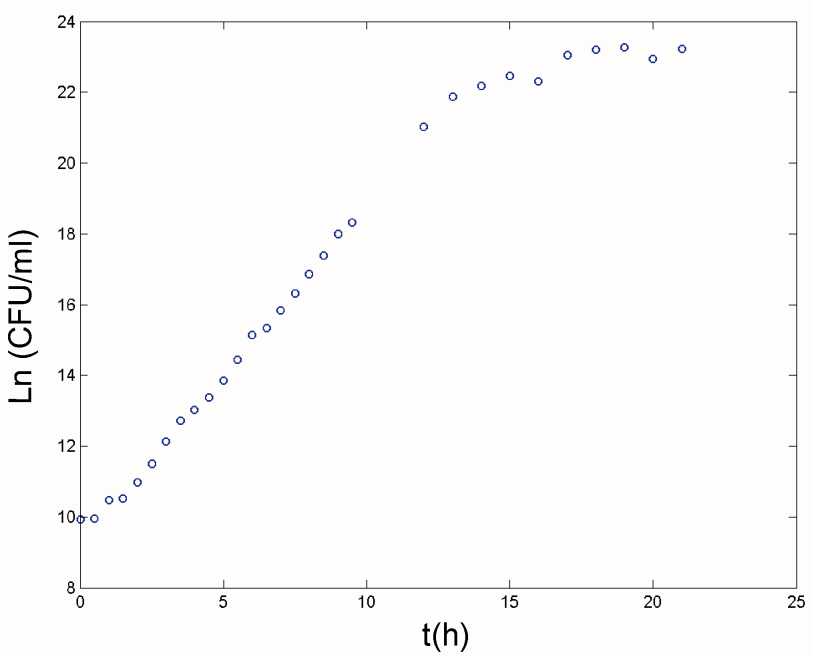



Figure 6.1: *Growth curve for an Escherichia coli strain growing in a BHI broth at* $27.5\,^{o}C$ *(Bernaerts, 2002).*

similar cases is the *Mean Square Error* or MSE. This quantifies the differences between the experimental points and the corresponding points of the simulated curve. Thus, the closer the points, the lower the MSE value.

In this specific case we work with cell concentration. Our experimental data consist of a set of cell concentration measures at certain measure times, $C_{exp}(t_i)$. The simulation gives the cell concentration at each time step ($t_s = 1\,min$). Therefore, we can identify the cell levels resulting from the simulation that correspond to the measure times, $C_{sim}(t_i)$. If $n$ is the total number of data points that is used in the calculation, the MSE will be:

$$MSE = \frac{1}{n-1}\sum_{i=1}^{n}(C_{exp}(t_i) - C_{sim}(t_i))^2 \tag{6.1}$$

### 6.2.3 INDISIM adaptation

The aim of this chapter is to adapt, test and evaluate the usefulness of different parameterization methods. Thus, it is not necessary to work with the complete INDISIM

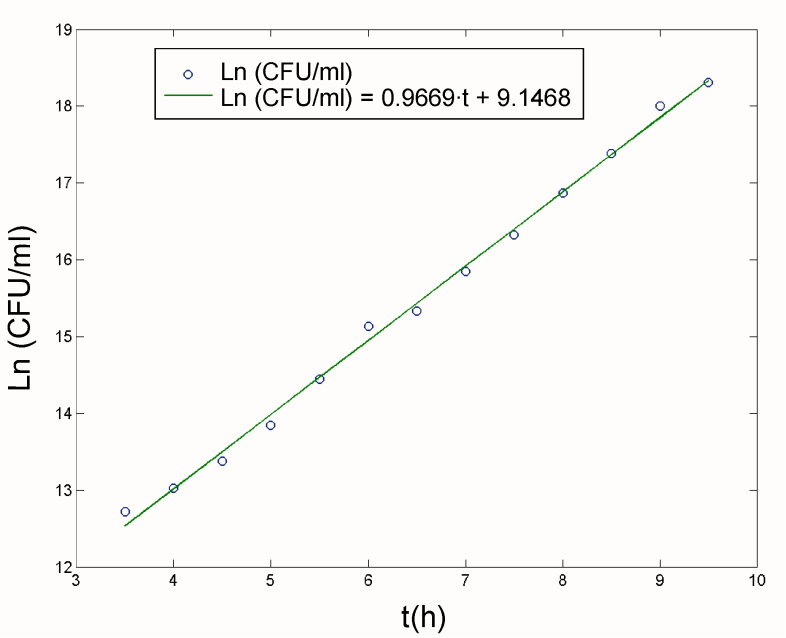



Figure 6.2: *Exponential part of the experimental growth curve that will be used in this study (Bernaerts, 2002).*

version. The version used is a simplification of the one presented in Section 2.3, in order to obtain fast simulations that allow fast multi-evaluations. This optimization of the methods for IbM parameter estimation has been done together with BioTeC research group (Katholieke Universiteit Leuven), who work on BacSim parameter estimation (Kreft et al., 1998; Kreft et al., 2001; Standaert, 2007). Therefore, we need an INDISIM version that is similar to BacSim in order to compare results and extract common conclusions. Some of the main changes and simplifications are detailed below.

**Output:** The experimental dataset is the growth curve presented in Figure 6.1. Therefore, the only output that we need from the simulations is the evolution of the cell concentration. The suppression of graphic output and unnecessary calculations considerably reduces the simulation time.

**Experimental setup:** The dataset comes from a homogeneous batch culture. Therefore, the model is adapted to shuffle around the nutrient particles and the bacteria at each time step. There is no need to introduce randomness when setting the order of

the sequential turns of action, and the nutrient diffusion and bacterial movement can be avoided. These simplifications result in a simulation time reduction.

**Random number generator:** Multi-evaluations and multi-runs must be performed. It is important to use a random number generator with a long cycle. We have implemented with INDISIM a pseudo random number generator based on Mersenne-Twister algorithm (Matsumoto and Nishimuram, 1998).

**Reproduction model:** For the reasons given above, the reproduction model is slightly modified in some simulations: it is used the mass at division $m_d$ (at the end of the cell cycle) instead of the mass to initiate the reproduction cycle $m_R$ (at the beginning of the reproduction cycle). That is, the cell cycle duration is not taken into account, and the reproduction occurs immediately when the cell reaches the $m_d$. The relationship between them is $m_d = m_R \cdot e^{-\mu_{max} \cdot t_R}$. This simplification is made in accord with the aim of this chapter, which is the assay of the optimization methods for carrying out IbM parameter estimation.

**Uptake model:** Regarding the uptake, a double Gaussian is used, in order to be coherent with BacSim's model. On the one hand, we maintain a small variability in the individual mean uptake each time step. On the other hand, a bigger Gaussian is introduced to give individual mean uptake values, as is done in BacSim. After some tests, a $CV = 0.15$ for this Gaussian is chosen. This assumption has to be considered as a temporal artifact to test the methods, since it has serious consequences on the biological model. It implies that there are some bacteria with a natural tendency to be fast growers, while there are other bacteria that tend to grow slowly. But some questions arise from this. Is this hereditary behaviour? If it is, natural selection should favor the fast growers over the slow. If it is not, it is difficult to defend such a strong tendency without a genetic component. But, again, the aim of the chapter arises and the simplifications or small modifications of the model do not have to be strongly considered.

This example shows that different behaviour models are implicit in the parameterization processes. Thus, depending on the chosen model we will obtain different parameters.

With these simplifications and assumptions, an INDISIM simulation with an initial population of $100\,cells$ takes between 35 seconds and 1 minute to reach a population of $10^6\,cells$ in a standard PC. Since the experimental data covers a cell level from $10^4$ to $10^8$CFU/ml, a scaling of the simulation results is performed prior to the MSE evaluation.

The scaling is proportionally done by means of simple fitting procedures appropriate to the purposes of this study (Standaert, 2007).

### Input parameters

The *non-physical* parameters are adjusted to chose the experimental setup conditions, a batch homogeneous culture, and set the particular conditions enumerated above.

The *physical* parameter input file has been modified to introduce the parameters in real units. These values are converted into simulation units in the first subroutine. The choice of these parameters is made according to the values used by BacSim, which are taken from the bibliography (Table 6.1).

Table 6.1: *Overview of the physical input parameters and the bibliography sources.*

| *Parameter* | *Value* | *Source* |
|---|---|---|
| Cellular density, $\rho$ (dry g/l) | 290 | Table 2 of Kreft et al. (1998) |
| Reproduction cycle duration, $t_R$ (min) | 27 | Table 3 of Bremer and Dennis (1996) for *E. coli* growing at 37°C and $\mu = 1h^{-1}$ |
| Mass at division, $m_d$ (dry pg) | 0.426 | Table 1 of Kreft et al. (1998)[1] |
| mass to initiate the reproduction cycle, $m_R$ (dry pg) | 0.276 | From $m_d$ and $\mu_{exp}$, extrapolating the growth during reproduction cycle ($m_R = m_d \cdot e^{-\mu_{exp} \cdot t_R}$) |
| Yield $Y$ (dry g cell/mol glucose) | 78.69 | Table 2 of Kreft et al. (1998) and Pirt's relationship from Domach et al. (1983)[2] |

[1]The volume at division, $V_d$, is calculated as $V_d = V_{d,min}2^g$, where $g$ is the number of generations per hour ($g = \mu/ln2$). The volume $V_{d,min}$ would be the volume at division if $\mu = 0$. It is evaluated with the expression $V_{d,min} = 2\overline{V}_u/1.433$, where $\overline{V}_u$ is the mean cellular volume when $\mu = 0$ (Kreft et al., 1998). The mass at division $m_d$ can be easily calculated with the cellular density.

[2]Pirt's relationship establishes $1/Y = 1/Y_{max} + m/\mu$. Taking $Y_{max} = 0.444\, fg_{drybiom}/fg_{glucose}$ and $m = 0.0004\, fg_{glucose}/fg_{drybiom} \cdot min$ from Kreft et al. (1998) and $\mu_{max} = 0.9669\, h^{-1}$ from the experimental dataset, we obtain the indicated result.

### Parameters to be estimated

The optimization process has been carried out in two phases. In the first step, the three methods were used to estimate one or two parameters to fit the experimental data corresponding to the exponential phase. In the second step, the NEWUOA method was

used to estimate two or three parameters with the data of the lag and exponential phases. This is an overview of the different estimations:

1. Exponential phase data
   (a) One parameter: mean maximum uptake rate ($u_{max}$)
2. Lag and exponential phase data
   (a) Two parameters: mean maximum uptake rate ($u_{max}$) and mean mass at division ($m_d$), fixing an initial biomass distribution among bacteria
   (b) Three parameters: mean maximum uptake rate ($u_{max}$) and initial biomass distribution among bacteria ($A$ and $B$ of a Weibull distribution), fixing the mean mass at division

## 6.3 Parameter estimation methods

### 6.3.1 Classical method: grid search

The classical method for parameter estimations consists in evaluating the objective function (MSE) for different values of the parameter to be estimated, and taking as the best value that one that minimizes the objective function. When more than one parameter is being estimated, the same process must be carried out with different combinations of the parameter values, taking as the best the combination that gives the minimum value for the objective function.

To do this in a systematic way, a grid with the different combinations to be evaluated is generated. Each parameter is discretized in a certain interval where its best value is assumed to be. Therefore, for each parameter $p_i$ a number $n_{p_i}$ of points to be tested are set. If $N$ parameters must be estimated, the grid will consist of $n_{p_1} \times n_{p_2} \times ... \times n_{p_i} \times ... \times n_{p_N}$ points. The value of the objective function is evaluated at each point of the grid, and the best combination of the parameters is given by the grid point where the objective function has the minimum value.

For IbM parameter estimation, every objective function estimate requires a simulation with the corresponding input parameters. IbM simulations always include some randomness that comes either from the programming strategies or from the computing essence. Therefore, the same set of input parameters may result in different output parameters for two independent simulations. This reflects the reality: it is impossible to obtain the

same results from two independent experiments, although the conditions are assumed to be the same.

The grid search method must take into account this phenomenon. Usually several repetitions of the whole search are performed, so that we have a distribution of 'best estimates' as a result. From the histogram of the obtained best estimates for each parameter, the final best estimation can be obtained.

This method does not require a complex implementation. We only need a program that generates the grid and executes the corresponding INDISIM simulations. For each simulation, the value of the objective function must be assessed (Fig. 6.3).

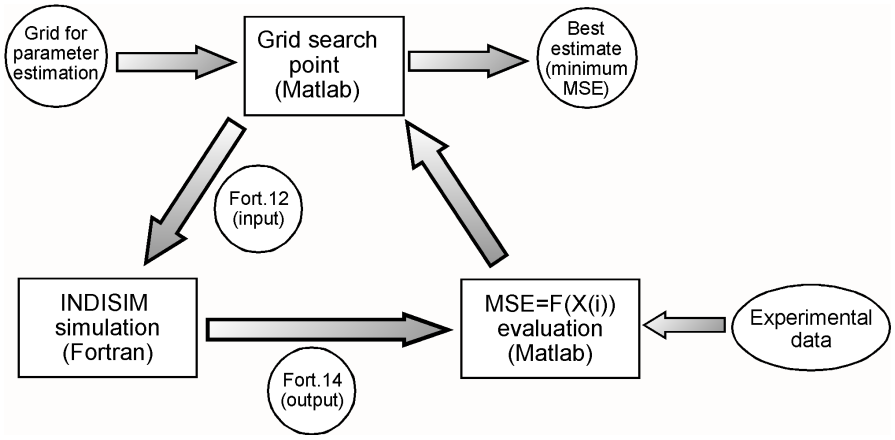



Figure 6.3: *Overview of the implementation of the grid search method for parameterization INDISIM.*

This method is probably the most reliable, but also the most tedious. It requires a lot of simulations and, therefore, it involves a great time expenditure. For instance, let us suppose that two parameters must be estimated. These parameters are discretized in 20 values each one, so a $20 \times 20$ grid is created, resulting in 400 points where the MSE must be assessed. If 100 repetitions of the whole search are performed to solve the problem of the randomness, $400 \times 100 = 40000$ simulations must be carried out. Let us suppose that we are working with simulations that last 5 minutes each on a standard PC. Then, the total amount of time required to perform the grid search is $5 \times 40000 = 200000$ minutes. This is equivalent to 139 days-more than 4 months! Although several simulations may be performed in parallel and the time can vary depending on the computer used, the total time is still great.

Several methods have been developed to find the minimum of a function without

assessing its value at each grid point. We need some methods that optimize the search for the minimum of a non-derivable function. Two methods that do not use the function gradient in their algorithms have been chosen to be adapted and tested. They are the Nelder-Mead Threshold Accepting (Section 6.3.2) and the NEWUOA (Section 6.3.3).

### 6.3.2 Nelder-Mead Threshold Accepting

The Nelder-Mead simplex search method is an algorithm that was first proposed by Nelder and Mead (1965). It has become one of the most used algorithms for nonlinear unconstrained optimization. It is a direct search method, since it does not use derivatives. The basic unit in this method is the *simplex*: a geometric figure in an $n$-dimensional space that is a convex hull of $n+1$ vertexes, with each vertex representing a certain combination of the $n$ parameters to be estimated. The value of the objective function is assessed in each simplex vertex, and a new simplex that is closer to the objective function is constructed in each step. The rules for constructing each new simplex are based on a set of geometric operations: *reflection*, *expansion*, *outside contraction*, *inside contraction* and *shrinkage* (for a graphical description of these operations see Standaert, 2007). The value of the objective function determines whether new vertexes are accepted or rejected. If a new vertex is accepted, a new simplex is constructed by rejecting the worst existing vertex, always according to the objective function values.

Four parameters are defined to determine the impact of the geometric operations. They are the coefficients of reflection ($\rho = 1$), expansion ($\chi = 2$), contraction ($\gamma = 1/2$) and shrinkage ($\sigma = 1/2$) (Standaert, 2007).

As was pointed out in Section 6.3.1, if we perform independent IbM simulations there is an inherent randomness that results in slightly different output for the same input parameters and conditions. This phenomenon can entail the appearance of local minima in the objective function. To solve this obstacle, the Threshold Accepting (TA) algorithm is incorporated (Dueck and Scheuer, 1990). TA performs a local search that escapes local minima by means of accepting solutions that are not worse than the current one by more than a given threshold, $\tau$. That is, a new point is randomly chosen around the existing one. If the objective function value in the new point is not worse than its value in the old point plus $\tau$, the new point replaces the old one. The value of $\tau$ is successively decreased as the real minimum comes closer.

### NMTA for INDISIM parameter estimation

The Nelder-Mead Threshold Accepting (NMTA) algorithm for BacSim one- parameter estimation was implemented in Matlab by Standaert (2007). The original program was adapted to render the INDISIM parameter estimation with the chosen experimental data. Figure 6.4 shows an overview of the structure of the implemented program.

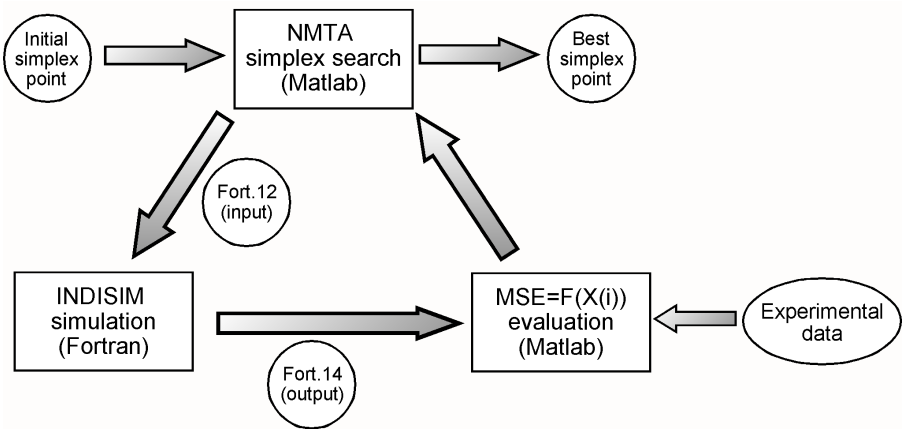



Figure 6.4: *Overview of the implementation of the NMTA method for optimizing the INDISIM parameter estimation.*

In this study, the local minima will be avoided by means of several mechanisms. The most important is the incorporation of the above-mentioned TA algorithm. Then, we have three complementary mechanisms. At each simplex search step, the objective function is recalculated for all the simplex vertexes, even for the existing points. Moreover, a certain number of function evaluations, $R$, is averaged to obtain the objective function value. Finally, at each simplex search step there is a small probability $\xi = 0.15$ of making a random shift of the current simplex, with a random magnitude in a random direction.

An initial simplex must be input to the program to initiate the search. From this initial point, $n_R = 3$ successive rounds will be performed to find the objective function minimum, each one containing a defined number of simplex search steps, $n_{S,r}$. The first round ($r = 1$) will contain $n_{S,1} = 30$ search steps. The second round ($r = 2$) will comprise $n_{S,2} = 20$ simplex search steps, and $n_{S,3} = 10$ for the third one. The number of repetitions, $R$, for averaging the function value also varies from round to round. It increases as the function minimum comes closer in order to increase the final result accuracy: $R_1 = 1$, $R_2 = 5$ and $R_3 = 15$.

The threshold value is calculated for each round according to Eq. 6.2:

$$\tau_r = \Gamma \cdot \alpha \cdot R_r^{\beta} \tag{6.2}$$

The constant $\Gamma$ is a parameter that determines the strictness of the threshold value, and it is set to $\Gamma = 2$ (Standaert, 2007). The constants $\alpha$ and $\beta$ must be set according to the objective function behaviour around the minimum. Therefore, this depends on the specific case to be studied. A summary of the mentioned constants is shown in Table 6.3.

Table 6.3: *Summary of the parameters and constants of the NMTA algorithm to be used in INDISIM parameter estimations (source: Standaert, 2007).*

| *Constant* | *Value* | *Definition* |
|---|---|---|
| $\rho$ | 1 | Coefficient of reflection |
| $\chi$ | 2 | Coefficient of expansion |
| $\gamma$ | 1/2 | Coefficient of contraction |
| $\sigma$ | 1/2 | Coefficient of shrinkage |
| $\xi$ | 0.15 | Probability of randomly changing the current simplex |
| $n_R$ | 3 | Number of successive rounds |
| $n_{S,1}$ | 30 | Number of simplex search steps in the first round |
| $n_{S,2}$ | 20 | Number of simplex search steps in the second round |
| $n_{S,3}$ | 10 | Number of simplex search steps in the third round |
| $R_1$ | 1 | Number of evaluations of the objective function at each simplex vertex in the first round |
| $R_2$ | 5 | Number of evaluations of the objective function at each simplex vertex in the second round |
| $R_3$ | 15 | Number of evaluations of the objective function at each simplex vertex in the third round |
| $\Gamma$ | 2 | Constant for the threshold accepting strictness |

### 6.3.3 The NEWUOA method

The NEWUOA is a method for solving non-linear unconstrained minimization problems that was developed by Powell (2004). The algorithm seeks the minimum of an objective function $F(\overrightarrow{x})$, where $\overrightarrow{x} \in \mathbb{R}^n$. The model proposes the use of the objective function values for building a quadratic model, $Q(\overrightarrow{x}) \approx F(\overrightarrow{x})$, which is assumed to be valid in a neighbourhood of the current iteration, called trust region. The radius of the trust region is iteratively adjusted. Then, the quadratic model is minimized within the trust region, hopefully yielding a point with a low function value.

The complete method is detailed in Powell (2004). In general terms, at each iteration the NEWUOA algorithm fits a quadratic model $Q$ to a set of points $Y$ of the objective

function $F(\overrightarrow{x})$, which is valid inside a trust region. Some specific rules are also used for the quadratic model identification (Powell, 2004). Then, NEWUOA finds the minimum of the quadratic within the trust region. The real cost of the function is evaluated at this point: if the decrease in the real value is less than the decrease predicted by the model, the radius of the trust region is scaled down; otherwise, the radius is not changed. The minimization algorithm by successive quadratic approximations is summarized in Algorithm 1. The algorithm finishes when the radius of the trust region achieves the lower boundary fixed by the user, $\rho_{end}$.

**Algorithm 1** *Overview of the NEWUOA algorithm for each iteration (source: M. Guilbert, personal communication, 2007)*

(i) Initialize the set $Y_{beg}$, the radius of the trust region $\rho_{beg}$ and the first iteration
(ii) Build the quadratic model $Q$
(iii) Minimize this model within the trust region
(iv) Update the set of interpolation $Y$
(v) Update the radius of the trust region $\rho$
(vi) Update the current iteration and go to step (ii), until $\rho = \rho_{end}$.

### NEWUOA for INDISIM parameter estimation

The NEWUOA software can be freely downloaded from the Web[1]. It is programmed in Fortran. Only a few modifications in it need to be made in order to input the necessary data and to link the NEWUOA software with the $F(\overrightarrow{x})$ calculations. Thus, it has been adapted for performing INDISIM parameter estimation. The objective function is, again, $F(\overrightarrow{x}) = MSE(\{p_i\})$. That is, the vector $\overrightarrow{x}$ contains the different combinations of the $N$ parameters to be estimated, $\{p_i\}$. Each 'combination' or vector has to be introduced to the INDISIM input file to obtain the corresponding simulation results and, therefore, the $MSE$. Since three different programs have to be combined, text files are used to pass the information from one to another. An overview of the global implementation of NEWUOA with *MSE* evaluations and INDISIM is shown in Figure 6.5.

The NEWUOA algorithm uses $m = 2n + 1$ points for each interpolation, where $n$ is the dimension of the vector $\overrightarrow{x}$ and, therefore, is equal to the number of parameters to be estimated ($n = N$). The program asks for an initial point, $\overrightarrow{x_0}$, and the radius of the initial trust region, $\rho_{beg}$. The initial group of $m$ interpolation points, $Y_{beg}$, is set by adding and subtracting the radius $\rho_{beg}$ to the initial vector components, $\overrightarrow{x_0} = (\{p_{i,0}\})$.

For instance, let us suppose that two parameters, $p_1$ and $p_2$, are being estimated. In this case, the vector $\overrightarrow{x}$ has two components, and $m = 2N + 1 = 5$ interpolation

[1] http://www.inrialpes.fr/bipop/people/guilbert/newuoa/newuoa.html

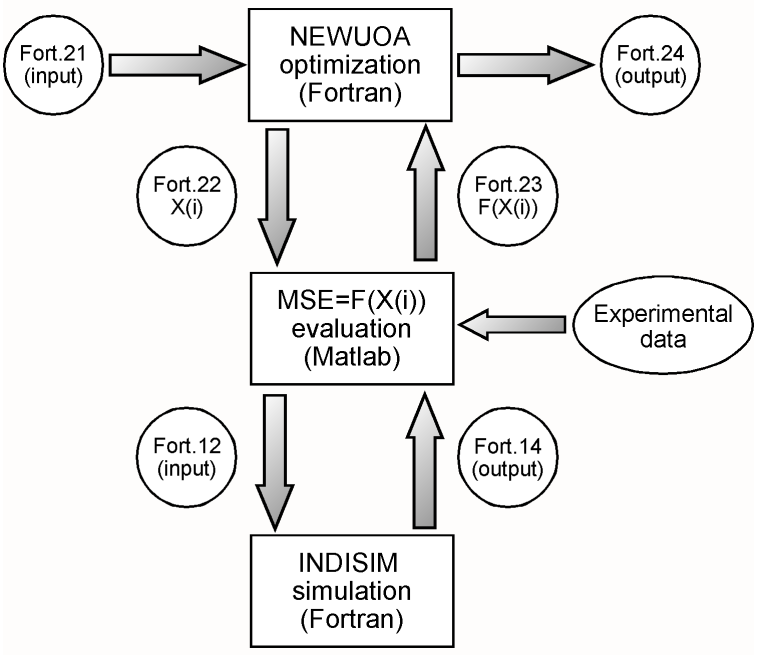



Figure 6.5: *Overview of the NEWUOA implementation for optimizing the INDISIM parameterization.*

points. If the initial point is set to $\overrightarrow{x_0} = (p_{1,0}, p_{2,0})$, and the initial radius is $\rho_{beg}$, the other four interpolation points for the first iteration will be $\overrightarrow{x_1} = (p_{1,0} + \rho_{beg}, p_{2,0})$, $\overrightarrow{x_2} = (p_{1,0} - \rho_{beg}, p_{2,0})$, $\overrightarrow{x_3} = (p_{1,0}, p_{2,0} + \rho_{beg})$ and $\overrightarrow{x_4} = (p_{1,0}, p_{2,0} - \rho_{beg})$.

The use of a unique radius to set the trust region and the interpolation points is the first pitfall in our study. The $N$ parameters that we are estimating may have very different orders of magnitude, and the trust region for each one can be very different from the others. Another problem arises from the nature of the objective function. $F(\overrightarrow{x})$ is assumed to be defined for $\mathbb{R}^N$, while $MSE$ may not be defined in some zones: if the mass to initiate the reproduction cycle is being estimated, negative values for it make no sense, or INDISIM may not be prepared to perform simulations with a mean uptake above an upper limit. Therefore, we may have some constraints, and NEWUOA is a method to solve unconstrained minimization problems.

However, the physical essence of these parameters provides the clues for partially solving these problems. We must be able to identify an interval for each parameter that contains the best estimation. This interval will be the initial trust region, and it should

be as small as possible to have the $MSE$ defined for the whole interval, but great enough to contain the solution. This will be the key factor for the success of the optimization.

The parameters $p_i$ will be re-scaled before being input to the NEWUOA algorithm, taking the values from 0 to 1 in the given interval. If $p_{i,min}$ and $p_{i,max}$ are the limits for the parameter $p_i$, the re-scaled parameter, $\hat{p}_i$, will be (Eq. 6.3):

$$\hat{p}_i = \frac{p_i - p_{i,min}}{p_{i,max} - p_{i,min}} \tag{6.3}$$

Therefore, in the new scale $\hat{p}_{i,min} = 0$ and $\hat{p}_{i,max} = 1$. Working with the parameters in the transformed scale, the radius $\hat{\rho}_{beg}$ can be commonly defined for all of them. Then, when a set of parameters has to be input to INDISIM to evaluate the corresponding $MSE$ value, the original values are retrieved as (Eq. 6.4):

$$p_i = (p_{i,max} - p_{i,min}) \cdot \hat{p}_i + p_{i,min} \tag{6.4}$$

The initial point can be set to the middle point of the interval, $\overrightarrow{x_0} = (\{p_{i,0}\} = \{\hat{p_{i,0}} = 0.5\})$. With an initial radius $\hat{\rho}_{beg} = 0.5$, we would sweep the whole interval at the first iteration.

If the interval is well chosen, we will not have problems regarding the constraints, because the algorithm will almost always be within the interval. If it is not possible to delimit the interval, some penalization function can be added to the objective function to avoid the *forbidden* zones, as has been done in other fields (Guilbert et al., 2006).

## 6.4 Results

### 6.4.1 Parameter estimation with exponential phase experimental dataset

In this section, only the experimental data corresponding to the experimental phase is used (Fig. 6.2). The aim is to implement and test the usefulness of the three methods, so the first step must be done with the simplest case. First of all, the estimation of one parameter is performed in order to implement, adjust, test and improve the three methods.

The first parameter to be estimated is the $u_{max}$. The other input parameters are set according to Table 6.1.

### Classical method

A grid of 20 uniformly distributed $u_{max}$ values is set up, covering the interval [0.011, 0.014]. The entire grid is covered from the minimum to the maximum to evaluate the MSE at each point of the one-dimensional grid. This process is repeated for 149 individual and independent runs. The results are shown in Figure 6.6.

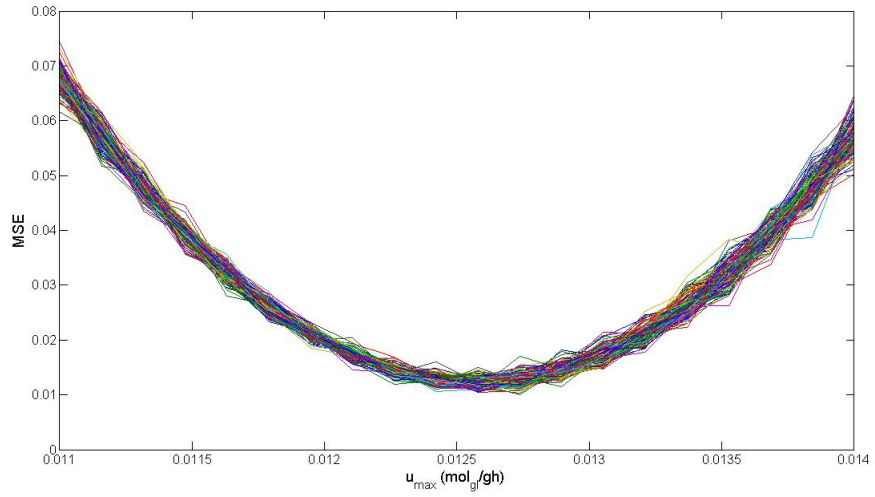



Figure 6.6: *The MSE for a grid of 20 $u_{max}$ values, for 149 runs executed with non-overlapping random number sequences.*

The best estimate for $u_{max}$ is assumed to be the grid point where the MSE reaches its minimum value. A histogram of the best estimates obtained by the 149 runs is constructed (Fig. 6.7a). A normal distribution is fitted to this histogram to evaluate the mean best estimate (Fig. 6.7b), which results in $u_{max} = 0.012589 \pm 0.000108\, mol_{gl}/gh$.

### NMTA method

The constants $\alpha$ and $\beta$ that are used to fix the threshold (Eq. 6.2) must be set for this specific case. They are derived from the behaviour of $MSE$ around the minimum (mean and standard deviation). 300 evaluations of $MSE$ value at the point $u_{max} = 0.0124\, mol_{gl}/gh$, which is near the minimum, are assessed. The values $\alpha$ and $\beta$ are derived from the mean and the standard deviation of the result (Standaert, 2007), which are $\overline{MSE} = 0.0137$ and $StdDev = 0.0009$. The obtained values are $\alpha = 0.0009468$ and $\beta = -0.527$.

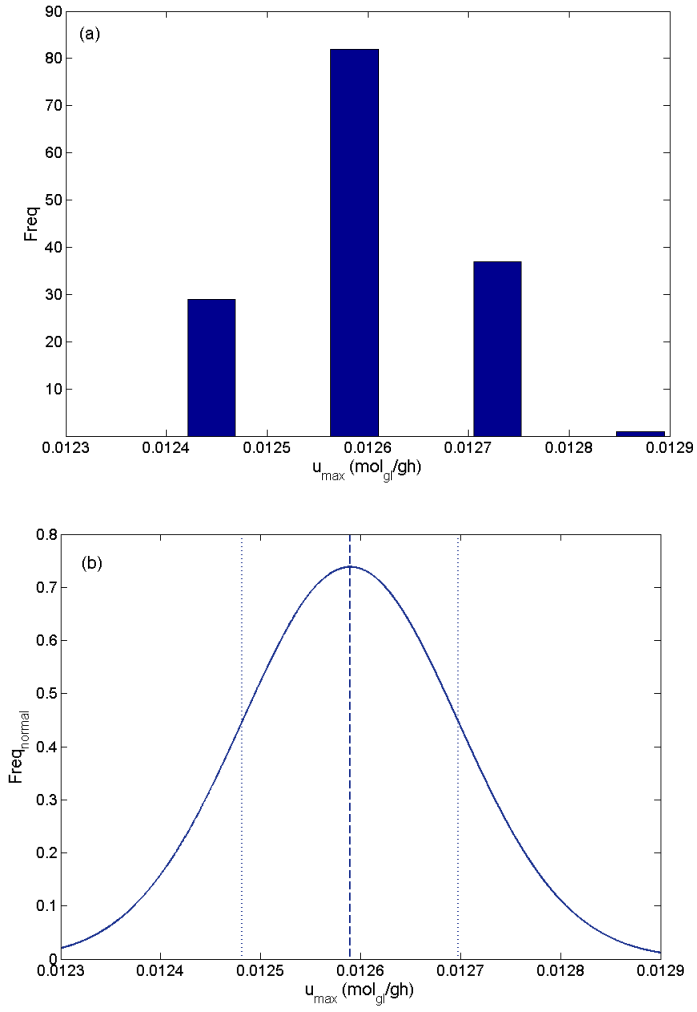
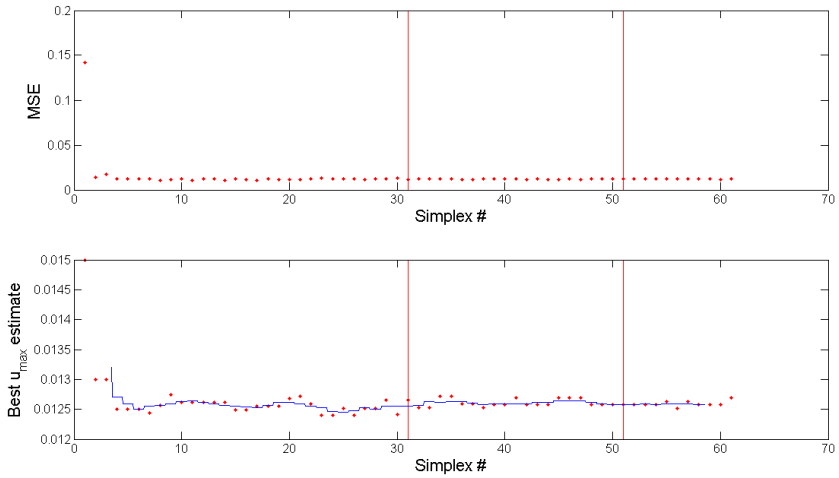


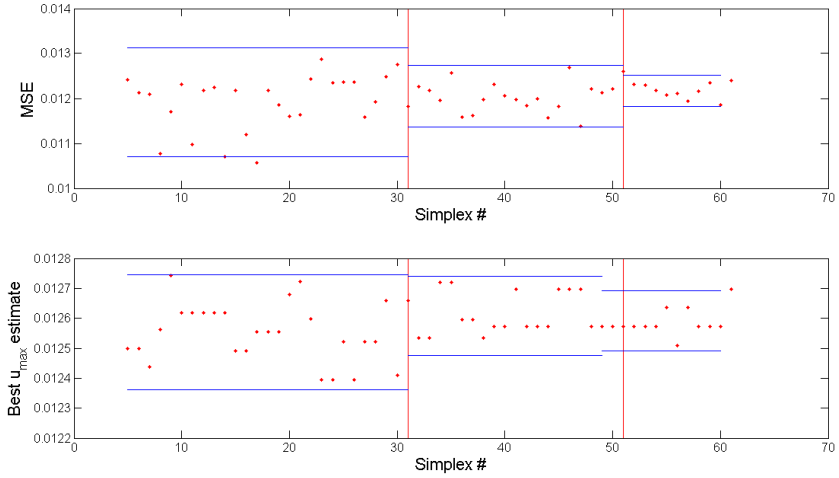

Figure 6.7: *(a) Best estimates histogram for the 149 individual runs. (b) Normal distribution fitted to the best estimates histogram, with the dashed line the mean value and the dotted lines the confidence interval boundaries.*

Several optimizations are done to assay and polish up the NMTA method, each lasting between 6 and 20 hours. In general, no problems of convergence are detected, and the quality of the best estimates are acceptable. Figure 6.8 shows a typical result. The initial simplex was [0.016, 0.015], and the optimization process took 15 $h$. The best estimate was

$u_{max} = 0.0127\, mol_{gl}/gh$, which is in agreement with the grid search result ($\Delta u_{max} < 1\%$).




(a) Complete results for the maximum uptake estimate




(b) Enlargement of the convergence zone

Figure 6.8: *Results of the Nelder-Mead Threshold Accepting method for estimating the best $u_{max}$. The vertical lines indicate the boundaries between subsequent stepping rounds increasing accuracy.*

**NEWUOA method**

The first tests of NEWUOA method for one parameter estimation are done without the re-scaling options, since there are no problems of orders of magnitude or trust regions between parameters. Each optimization process lasts around 50 minutes. 20 independent optimizations have been performed from different initial points, which are randomly chosen. The results are shown in Figure 6.9. As is shown, no problems of convergence are detected. The mean best estimate is $u_{max} = 0.01269\, mol_{gl}/gh$. This result is also in agreement with the grid search and NMTA best estimates.

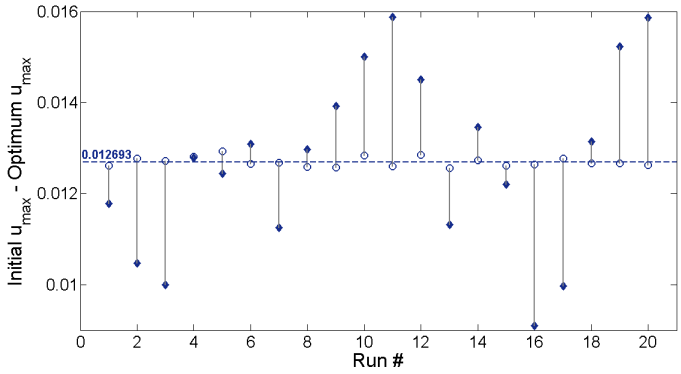



Figure 6.9: *20 independent rounds of NEWUOA optimization method for estimating the $u_{max}$. Blue full rhombus are the randomly chosen initial points, and red open circles are the best estimate for each run. The mean of the 20 best estimates is shown in a dashed line.*

**Conclusions**

The best estimates for $u_{max}$ are similar using any of the presented methods, and the quality of the results is comparable. The classical grid search gives a lot of information about the *MSE* behaviour in the studied interval, but is slow and tedious. NMTA and NEWUOA show good convergence for the studied case, although it is necessary to delimit the interval. The fastest method by far is NEWUOA.

### 6.4.2 Parameter estimation with lag and exponential phases experimental dataset

After the tests with one parameter estimation, the method that seems to work best is the NEWUOA. It is immediately usable for estimating more than one parameter with-

out any adaptation. The only part that must be adjusted is the re-scaling of the input parameters (see Section 6.3.3). However, the NMTA method would require a long re-programming process in order to be able to estimate two parameters. Therefore, the estimation of more than one parameter will be carried out with the NEWUOA method, with some tests with the grid search.

In this part, the experimental data of the lag and exponential phases are used.

**Two-parameter estimation**

The initial distribution of the biomasses among the bacteria of the inoculum is set with a Weibull distribution. It is assumed to be known, and the parameters to be estimated will be the mass at division, $m_d$, and the mean maximum uptake rate, $u_{max}$. The lag phase duration of the simulated growths is mainly given by the distance between the initial mean mass (which is fixed) and the mass at division. The greater the distance, the longer the lag phase. The uptake rate determines the lag duration and the culture growth rate at exponential phase. The greater the uptake rate, the shorter the lag and greater the growth rate.

A Weibull distribution (Eq. 6.5) is fitted to a biomass distribution of a simulated growth at the end of the stationary phase. The parameters for this fitting result in $A = 0.0766\,pg$ (parameter related with the mean mass) and $B = 2.3236$ (parameter related to the shape).

$$f(m; A, B) = \frac{B}{A}\left(\frac{m}{A}\right)^{B-1} e^{-(m/A)^B} \tag{6.5}$$

The distribution is displaced $C = 0.0561$ from the origin. At each simulation, an initial biomass distribution is randomly generated with the fitted Weibull parameters. An example of a biomass distribution of 10,000 individuals, randomly generated with the fitted Weibull parameters, is plotted in Figure 6.10.

With this initial distribution, a NEWUOA optimization is performed in order to estimate $u_{max}$ and $m_d$. They must be re-scaled as indicated in Eq. 6.3. The chosen intervals are detailed in Table 6.4.

Table 6.4: *Intervals for $u_{max}$ and $m_d$ to re-scale the input values in the NEWUOA algorithm.*

| | $u_{max}\,(mol_{gl}/gh)$ | $m_d\,(pg)$ |
|---|---|---|
| *min* | 0.007 | 0.25 |
| *max* | 0.016 | 1.0 |

Forty independent runs are performed in order to validate the good convergence of

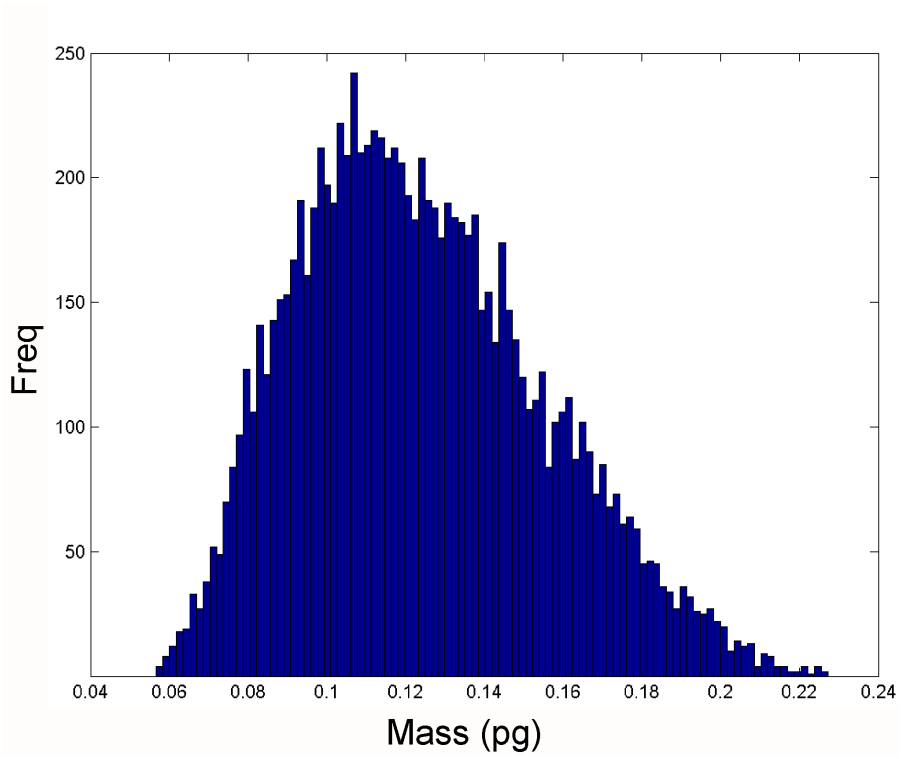



Figure 6.10: *Example of an initial biomass distribution for the estimation of mass at division and maximum uptake rate.*

the optimization. For carrying out these rounds, the starting points are randomly chosen in the interval $\hat{p}_i \in [0.25, 0.75]$ in the re-scaled format (between 0 and 1). The $\hat{\rho}_{beg}$ is set to 0.25, so the whole interval is potentially covered by the randomly chosen initial points, and the $\hat{\rho}_{end} = 0.000001$.

The results, which are plotted in Figure 6.11, give a $u_{max} = 0.0125\, mol_{gl}/gh$ and a $m_d = 0.4662\, pg$. Both of them are in the range of the expected values. The $u_{max}$ is slightly lower than the previous parameter estimations, and the $m_d$ is near the value that was used in previous simulations ($m_{d,bib} = 0.426\, pg$). The optimization processes take around 1 hour each. The convergence in mass at division is not as good as the convergence in maximum uptake rate. The reason for this is that the experimental data have only two points that correspond to lag phase. Therefore, there is a lack of information for carrying out the parameterization process properly. Nevertheless, the method seems to work properly. That is, the problem is in the experimental data, not in the optimiza-

tion method. It has been tested with other datasets with a longer lag phase, and the convergence is considerably better.

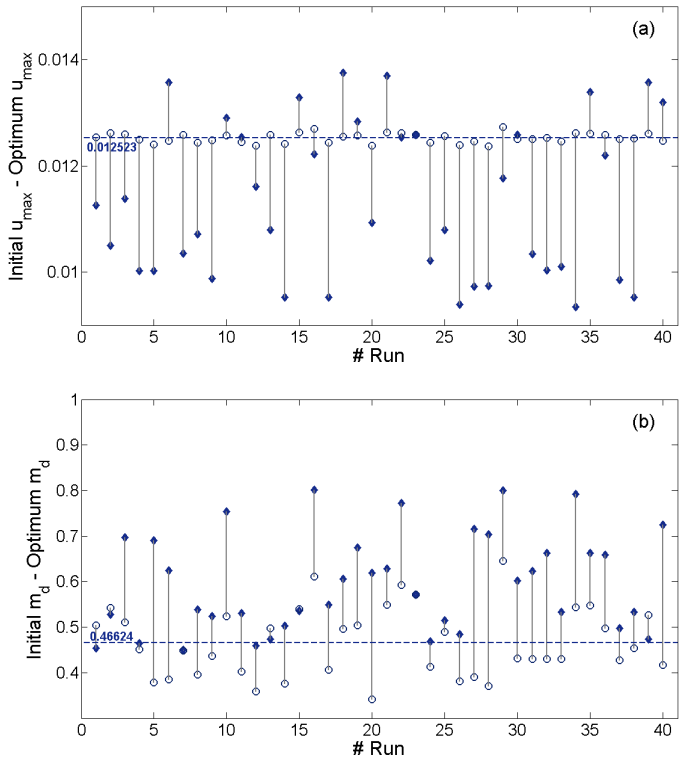



Figure 6.11: *NEWUOA results for optimizing (a) maximum uptake rate ($u_{max}$) and (b) mass at division ($m_d$). Blue full circles are the initial points, and red open circles are the best estimates for each run. The dashed line indicates the mean value of the best estimates.*

A grid search has been built to corroborate this result (Fig. 6.12). It yields a similar best estimate for both parameters ($u_{max} = 0.0124\, mol_{gl}/gh$, $m_d = 0.4\, pg$).

### Three-parameter estimation

Taking the $m_d$ from the bibliography ($m_d = 0.426\, pg$), a third test is carried out to evaluate the usefulness of the NEWUOA method for the estimation of more than one parameter. In this case, a three-parameter evaluation is performed. The chosen parameters are the $u_{max}$ again, and the two Weibull parameters for setting the inoculum

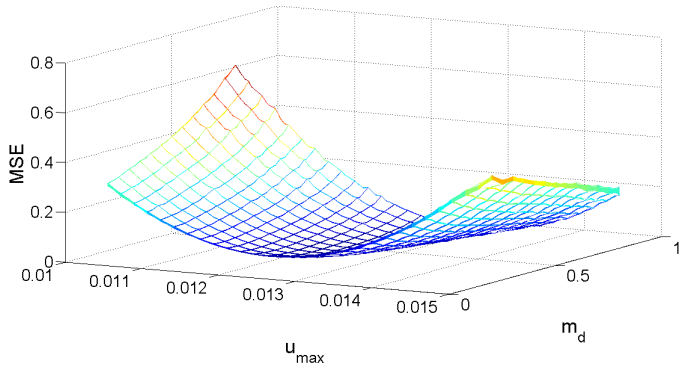



Figure 6.12: *Grid search for mass at division and maximum uptake rate.*

biomass distribution, $A$ and $B$. Thus, the initial biomass distribution is now assumed to be unknown.

The intervals for re-scaling the three parameters are shown in Table 6.5. Again, 40 independent runs are carried out with a random initial point $\hat{p}_i \in [0.25, 0.75]$, and the radii are set to $\hat{\rho}_{beg} = 0.25$ and $\hat{\rho}_{end} = 0.000001$. Each run takes from 1 to 2 hours.

Table 6.5: *Intervals for $u_{max}$, $A$ and $B$ for re-scaling the input values.*

| | $u_{max}\,(mol_{gl}/gh)$ | $A\,(pg)$ | $B$ |
|---|---|---|---|
| *min* | 0.007 | 0.01 | 1.0 |
| *max* | 0.016 | 0.3 | 6.0 |

The mean of the 40 independent runs results in the best estimates $A = 0.10\,pg$, $B = 2.99$ and $u_{max} = 0.0124\,mol_{gl}/gh$ (Fig. 6.13). The convergence is acceptable for $u_{max}$ and $A$, which are mainly related to the growth rate and the lag duration, respectively. But the convergence for the parameter $B$ is not so good. The reason is, again, a lack of information in the experimental data. The shape of the initial distribution determines the shape and duration of the transition between lag and exponential phases, as has been seen previously (Chapter 4). Thus, the lack of experimental points in the transition phase causes a lack of information for estimating the constant $B$.

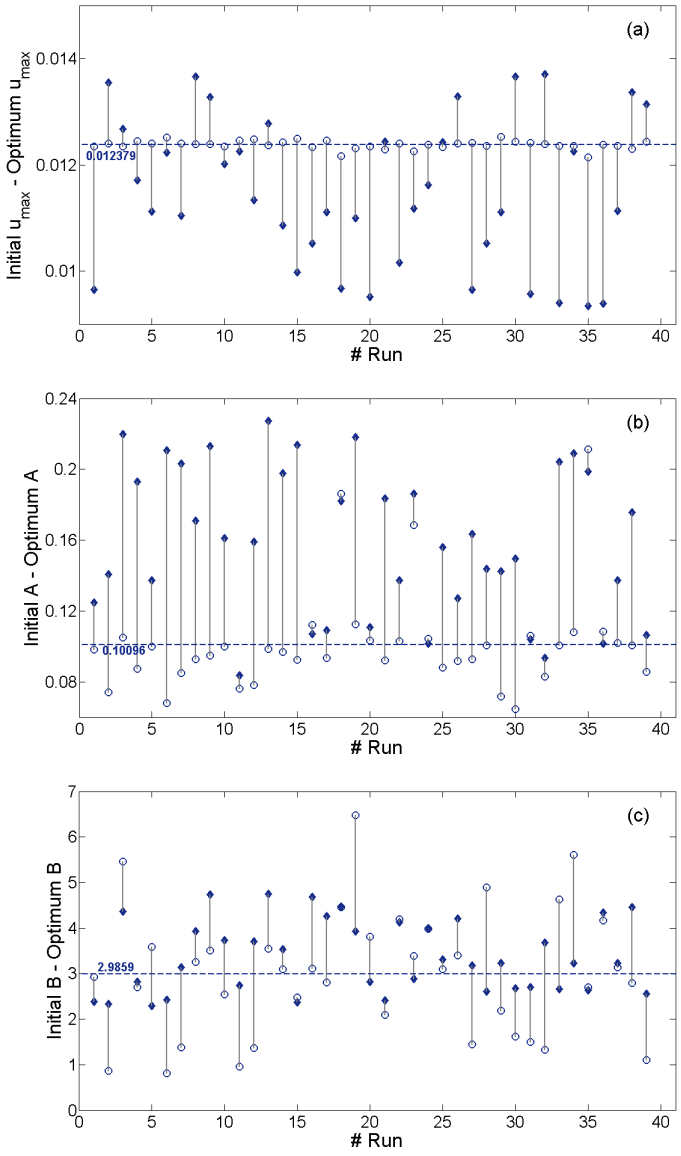



Figure 6.13: *NEWUOA results for optimizing (a) the maximum uptake rate $u_{max}$, (b) the Weibull distribution constant for the inoculum A and (c) the Weibull distribution constant for the inoculum B. Blue full rhombus are the initial points, and red open circles are the best estimates for each run. The dashed line indicates the mean value of the best estimates.*

## 6.5 Discussion

Two methods have been adapted and tested for IbM parameter estimation. NMTA method, which has been shown to be considerably faster than the classic grid search, shows a good convergence for one parameter but some difficulties for the programming. The NEWUOA method is a black box that can be easily adapted for many optimization problems. The adaptation for applying NEWUOA to any IbM parameter estimation problem has been carried out, and a strategy for avoiding the differences in orders of magnitude and trust regions has been implemented with success. One-, two- and three-parameter optimizations have been tested. More than three parameters could be easily evaluated with this method with the appropriate experimental dataset, without any specific adaptation of the algorithm. NEWUOA has proved to be a useful tool for IbM parameterization, although it requires a minimum knowledge of the parameters to be estimated. It is also the fastest method, and the results have sufficient precision. For further and better applications of the NEWUOA method to IbM parameter estimation, the objective function should be slightly modified to incorporate a penalty function that avoids the algorithm of leaving the real trust region. It has also been seen that the experimental data must contain as much information as possible, to obtain reliable results for the parameters.