MULTI-PARAMETRIC NONLINEAR PROGRAMMING AND THE EVALUATION OF IMPLICIT OPTIMIZATION MODEL ADEQUACY

Elaine T. Hale [*,1] **S. Joe Qin** [*]

[*] *The University of Texas at Austin*
Department of Chemical Engineering
Austin, Texas, USA

Abstract: An algorithmic framework for numerically approximating multi-parametric nonlinear programming (mp-NLP) solutions is given, along with a method that uses mp-NLP for evaluating the adequacy of the nominal model used in Implicit Optimization. The mp-NLP algorithm builds on numerical methods for single parameter nonlinear programming and for the approximation of implicit manifolds. An example problem is presented.

Keywords: parametric optimization, implicit optimization, real-time optimization, dynamic optimization, implicit manifolds

1. INTRODUCTION

A multi-parametric nonlinear program (mp-NLP) is a nonlinear program whose solution is a function of some finite number of uncertain model parameters. In particular, let $u \in \mathbb{R}^n$ be the vector of variables to be minimized over and $\alpha \in \mathbb{R}^d$ be the set of uncertain parameters. Then the mp-NLP problem is to find

$$u^*(\alpha) = \arg \left\{ \begin{array}{l} \min\limits_u f(u, \alpha) \\ s.t. \\ \quad h(u, \alpha) = 0 \\ \quad g(u, \alpha) \leq 0 \end{array} \right\}, \qquad (1)$$

where

$$\begin{aligned} &f \in C^2 : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}, \\ &h \in C^2 : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}^q, \\ &\text{and } g \in C^2 : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}^p. \end{aligned} \qquad (2)$$

No numerical methods for solving this problem have been given in the literature, however, methods for solving various simplifications of this problem have, see for example (Gal and Nedoma, 1972; Tøndel *et al.*, 2003; Guddat *et al.*, 1990). This work extends the predictor-corrector methods given for single parameter nonlinear programming (Lundberg and Poore, 1993; Guddat *et al.*, 1990) to mp-NLP using the multi-dimensional predictor-corrector algorithm for general parameterized nonlinear equations described by Rheinboldt and Brodzik (Rheinboldt, 1988; Brodzik, 1996).

There are many applications for which having a solution to the problem given in Equation 1 would be quite useful. For example, extending Tøndel's work, nonlinear MPC problems can be thought of in this light, with the initial state as the set of uncertain parameters. Then an mp-NLP algorithm would make it possible to do the bulk of the calculations necessary to implement nonlinear MPC offline, and in a manner that is more likely to be able to deal with the complications introduced by

[1] Supported by the American Association of University Women, the University of Texas at Austin, and the National Science Foundation.

non-convexity. Also, process synthesis problems often contain highly uncertain parameters such that one might want to look at the whole space of possible solutions, and multi-objective optimization problems can be considered by introducing parameters that indicate the relative importance of the various objectives. This work, however, concentrates on the application of mp-NLP to the question of whether or not a process model is accurate enough for use in *Implicit Optimization*.

Implicit Optimization is a method for optimizing batch processes proposed by Srinivasan, Bonvin, and their colleagues in a recent paper (Srinivasan *et al.*, 2003). The assumption that must be satisfied in order for this method to be successful is that the optimum of the nominal model of the process, as compared to the actual process optimum, must be qualitatively correct. Specifically, a *qualitative optimum* is defined by parsing the nominal optimum into a sequence of sets of active constraints, and is implemented in the plant via control loops that are designed to follow these constraints and/or set-points, switching at predetermined times which are updated using run-to-run control. The method is appealing in its online simplicity and potential to be robust to plant-model mismatch. However, one question looms: how does one know if the nominal model is close enough to the actual process to produce the correct qualitative optimum?

A three step process for answering this question is proposed. First, the nominal model is augmented with a) any structural elements that are not in the nominal model, but are suspected of being present and significant in the actual process and b) a list of uncertain model parameters. The parameter list should include parameters directly related to whether or not each of the structural elements in a) are present. Secondly, the parametric programming problem that results from using the augmented model in the original optimization problem is solved via discretization of any ODE's and the algorithm discussed in this paper. Thirdly, the region in the parameter space where the nominal qualitative optimum is valid is identified by inspection, and the likelihood of the process being in this region is assessed using process knowledge and data.

Sections 2 and 3 of this paper provide some background and outline a method for solving multi-parametric nonlinear programming (mp-NLP) problems. The fourth section then discusses the three part process for validating an Implicit Optimization result using an example problem. Finally, conclusions and future work are given.

2. MATHEMATICAL FOUNDATIONS

2.1 mp-NLP Problem Formulation

This work solves the mp-NLP problem using the Fritz-John first-order necessary conditions combined with an active set strategy (Lundberg and Poore, 1993; Guddat *et al.*, 1990). In particular, once the set, A, of active inequality constraint indices is defined:

$$A = \{i : g_i(u, \alpha) = 0\}, \tag{3}$$

solving the mp-NLP problem (Equation 1) is equivalent to solving the following set of parameterized nonlinear equations combined with appropriate active set switching:

$$F_A(z, \alpha) = \begin{bmatrix} \nabla_u \mathcal{L}(z, \alpha) \\ h(u, \alpha) \\ g_i(u, \alpha),\ \forall i \in A \\ \lambda^T\lambda + \mu^T\mu + \lambda_0^2 - \beta_0^2 \end{bmatrix} \tag{4}$$

where $\mathcal{L}$ is the Lagrangian (Mangasarian and Fromovitz, 1967):

$$\mathcal{L}(z, \alpha) = \lambda_0 f(u, \alpha) + \lambda^T h(u, \alpha) + \mu^T g(u, \alpha) \tag{5}$$

and

$$z = (u, \lambda, \mu, \lambda_0). \tag{6}$$

2.2 Implicit Manifolds

Parameterized nonlinear equations of the form

$$\begin{aligned} &F(z, \alpha) = 0 \\ &F : \mathbb{R}^N \times \mathbb{R}^d \to \mathbb{R}^N \end{aligned} \tag{7}$$

have been studied quite extensively in their own right. In this setting, a point

$$x = (z, \alpha) \tag{8}$$

is *regular* if $DF(x)$ $(= (\nabla F(x))^T)$ has maximal rank N. Then, given sufficient smoothness of F, the set

$$M = \{x : x \text{ is regular and } F(x) = 0\} \tag{9}$$

is an *implicit, d-dimensional differentiable manifold* (Brodzik, 1996). Also, Sard's theorem tells us that if $F \in C^k, k > d$ then almost all (in the sense of Lebesgue measure) points are regular (Guddat *et al.*, 1990); therefore M encompasses almost all solutions of $F(x) = 0$.

Numerical methods for approximating the manifold M are based on local parameterizations and tangent spaces. Given a regular point $x_0 \in M$, the tangent space of x_0, $T_{x_0}M$, is a d-dimensional linear subspace, is naturally associated with the function F:

$$\begin{aligned} T_{x_0}M &= \ker DF(x_0), \\ &= \left\{y \in \mathbb{R}^{N+d} : DF(x_0)\, y = 0\right\}, \end{aligned} \tag{10}$$

and intuitively represents the directions one can travel from x_0 and expect to remain close to

the solution manifold M. The tangent space of x_0 also induces a local parameterization once an orthonormal basis for $T_{x_0}M$ has been introduced. Let U be a $(N+d) \times d$ matrix whose columns form such an orthonormal basis. Then, within an open neighborhood of x_0, $\mathcal{U}_2$, all points on M are uniquely identified with coordinates in this basis, t, that are within a given open neighborhood of the origin, $\mathcal{U}_1$:

$$M \cap \mathcal{U}_2 = \left\{ x : x = x_0 + Ut + \left(DF\left(x_0\right)\right)^T \Phi\left(t\right) \right. \quad (11)$$
$$\left. , \; t \in \mathcal{U}_1 \right\}$$

where

$$\Phi : \mathbb{R}^d \supset \mathcal{U}_1 \rightarrow \mathbb{R}^N \quad (12)$$

is a unique, smooth function (Fink and Rheinboldt, 1987).

2.3 Triangulations

An approximation of M can be obtained using the above ideas by considering a neighborhood of the origin of $T_{x_c}M$, which has coordinates in $\mathbb{R}^d$ and known point x_c at its origin. New points, t_i's, are predicted in this space, and then projected onto M using Equation 11 and a nonlinear equation solver to approximate $\left(DF\left(x_0\right)\right)^T \Phi\left(t\right)$. The overall structure of M is easier to analyze, however, if connectivity is defined between these approximating points, such that a piecewise linear approximation of M is obtained. In this work, that connectivity is defined for the t_i's in relation to the origin of $T_{x_c}M$ and other relevant points using a *triangulation*.

A triangulation is a set of simplices. In $\mathbb{R}^d$, a *simplex* is the convex hull of $d+1$ points, which are called *vertices* or *nodes* of the simplex. Therefore, if $d = 1$, a simplex is a line segment, if $d = 2$, a simplex is a triangle, if $d = 3$, a simplex is a tetrahedron, etc. Also, the convex hull of any number of vertices of a simplex is a *face* and the convex hull of d vertices is a $facet$.

Besides being a set of simplices, a triangulation of some set of distinct points, $\mathcal{P}$, also satisfies several additional properties (Lawson, 1986):

(1) All vertices of the simplices are members of $\mathcal{P}$.
(2) The interiors of the simplices are pairwise disjoint.
(3) Each facet of a simplex is either on the boundary of the convex hull of $\mathcal{P}$ or is a common facet of exactly two simplices.
(4) Each simplex contains no points of $\mathcal{P}$ other than its vertices.
(5) The union of the simplices in the triangulation is the convex hull of $\mathcal{P}$.

In this work, local triangulations in $T_{x_c}M$ will satisfy all of these properties, but the d-dimensional triangular approximation of M only satisfies the first four properties and only if "convex hull of $\mathcal{P}$"in Property 3 is replaced by "current approximation of M".

2.4 Singular Points and Qualitative Solutions

The developments in the previous two subsections provide the basis for numerically approximating M (Equation 9) as defined by F_A (Equation 4). However, our real desire is to elucidate important qualitative features of this manifold. With respect to our optimization problem, this means identifying when the sought optimum is no longer near the neighborhood of M being considered, which happens when the need to change active sets is detected or the classification of the current optimum (minimum, maximum, or saddle) changes. Regions within such boundaries are considered to have the same *qualitative solution*, and the adjacent qualitative solutions are found by either changing A (and therefore F_A) or by looking at another region on M which has the correct classification and values of α. Poore et al. have shown and exploited the connection between the boundaries of qualitative solutions and properties of solutions of $F_A\left(x\right) = 0$ with active set switching, however, to present the main theorem that demonstrates this connection, some additional notation is needed.

First of all, Equation 4 can be written without using the active set strategy:

$$F\left(z, \alpha\right) = \begin{bmatrix} \nabla_u \mathcal{L}\left(z, \alpha\right) \\ h\left(u, \alpha\right) \\ \mu_i g_i\left(u, \alpha\right), \; \forall i \\ \lambda^T\lambda + \mu^T\mu + \lambda_0^2 - \beta_0^2 \end{bmatrix} = 0. \quad (13)$$

This is the original form of the Fritz-John necessary conditions (still leaving off the sign conditions on μ, g and λ_0), but includes the non-smooth complementarity conditions $\mu_i g_i = 0$, which are difficult to solve numerically.

Secondly, we will use the concept of the eigenvalues of a matrix restricted to a subspace. In particular, let $B \in \mathbb{R}^{n \times n}$ be a square matrix, and V be a k-dimensional subspace of $\mathbb{R}^n$. Then if $Z \in \mathbb{R}^{n \times k}$ is an orthonormal basis for V, the eigenvalues of B restricted to V, $\lambda\left(B|_V\right)$ are just the eigenvalues of $Z^T B Z$, and $B|_V$ is singular if and only if zero is an eigenvalue of $Z^T B Z$.

Now we present the theorem of interest without proof (Lundberg and Poore, 1993):

Theorem 1. Let $(z_0, \alpha_0) = (u_0, \lambda_0, \mu_0, \lambda_{00}, \alpha_0)$ be a solution to Equation 13. Assume that f, g and $h \in C^2$, the original definition of the set A

(Equation 3) holds, and define a tangent space T as:

$$T = \{y \in \mathbb{R}^n : D_u h_i(u_0, \alpha_0)\, y = 0,\ \forall i \\ D_u g_j(u_0, \alpha_0)\, y = 0,\ \forall j \in A\} \quad (14)$$

Then $D_z F(z_0, \alpha_0)$ is nonsingular *if and only if* the following three conditions hold:

(i) $\forall j \in A$, $\mu_j \neq 0$;

(ii) $S := \{\nabla_u h_i(u_0, \alpha_0)\}_{\forall i} \cup \{\nabla_u g_j(u_0, \alpha_0)\}_{\forall j \in A}$ is a linearly independent collection of $q + |A|$ vectors;

(iii) The Hessian of the Lagrangian restricted to tangent space T, $\nabla_u^2 \mathcal{L}\big|_T(z_0, \alpha_0)$, is nonsingular.

Therefore, detecting the transition from one qualitative solution to another is equivalent to monitoring the signs of μ and g (the active set), and detecting singularities in $D_z F_A$, since the characterization of non-singular points is completely based on the active set, the signs of the active μ_i's, and the signs of the eigenvalues of $\nabla_u^2 \mathcal{L}\big|_T(z_0, \alpha_0)$ (Guddat *et al.*, 1990).

3. A MULTI-PARAMETRIC NONLINEAR PROGRAMMING ALGORITHM

An algorithmic framework for solving mp-NLP problems is now given. The basic structure is taken from Brodzik's work on approximating implicitly defined manifolds (of any finite dimension) (Brodzik, 1996), the contributions of this work being the customization of Brodzik's algortihm to the mp-NLP problem (largely using the results of Lundberg and Poore), and the addition of auto-scaling, adaptive step-size adjustment, and parameter bounds following. The algorithm currently does not switch active sets, but that feature is forthcoming. (Switching active sets is related to singularity detection, approximation, and following, as indicated by Theorem 1. The current algorithm only detects singularities.) I will now give an overview of the entire algorithm. Details are excluded due to space constraints.

The proposed algorithm is best described as a multi-dimensional predictor-corrector. Given a starting point x_0 (which is a non-singular solution of Equation 13), the active set A of x_0 is determined, and the algorithm seeks to approximate the manifold

$$M = \{x : x \text{ is regular and } F_A(x) = 0\}. \quad (15)$$

To initialize the algorithm, a reference triangulation of the origin in $\mathbb{R}^d$, $\mathcal{T}_{ref}$, is calculated using triangulations by reflections (Allgower and Georg, 1978), x_0 is placed at its origin, and for each other vertex in the triangulation a new point is predicted. This new point is considered to be in $T_{x_0}M$ and is represented in the full space by

$$x_i^0 = x_0 + h_{x_0} U t_i, \quad (16)$$

where U is an orthonormal basis for $T_{x_0}M$, h_{x_0} is an initial step size, and t_i is a point in $\mathcal{T}_{ref}$. These points, x_i^0, are then projected onto the solution manifold, M, using a nonlinear equation solver to solve the augmented equations

$$G(x_i) = \begin{bmatrix} F_A(x_i) \\ U^T\left(x_i - x_i^0\right) \end{bmatrix} = 0, \quad (17)$$

characterize the new point with regards to the optimization problem, and detect any singularities in $D_z F$ (Lundberg and Poore, 1993). Finally, x_0, the new points and the triangulation structure specified by $\mathcal{T}_{ref}$ are stored in the database structure described in (Brodzik, 1998).

Once initialized, the algorithm enters the main loop. The first step is to find, if it exists, the next node in the database which is on the boundary of the current approximation of M, is nonsingular, and is feasible. This node is labeled x_c (the center point), and analogous to the algorithm initialization, x_c becomes the origin of its tangent space in $\mathbb{R}^d$. Any points in the database which are already in a simplex with x_c are projected onto this tangent space:

$$y_j = U^T(x_j - x_c), \quad (18)$$

with special note being made of points that are on an *open facet*, which is a facet with only one adjacent simplex. Then $\mathcal{T}_{ref}$ is rotated such that the direction of one of its vertices (from the origin) lines up with one of the open facet vertices' directions, and any points in $\mathcal{T}_{ref}$ lying between the open facets and not too close to any of them are used to predict new points in the approximation. These new points, the y_i's on the open facets, and the origin are then triangulated using Joe's algorithm (Joe, 1993). Finally, the new points are projected onto the solution manifold (using the same procedure as the initialization phase), and the algorithm checks to see if any of the new simplices overlap with any simplices that are already in the database. If there is no overlap, the new points, simplices and facets are added to the database. Otherwise, this iteration of the loop is ignored and x_c is marked as temporarily unusable as a center point. The main loop is repeated as long as potential center nodes exist.

4. IMPLICIT OPTIMIZATION EXAMPLE

In order to demonstrate the three part process for establishing Implicit Optimization model adequacy discussed in the introduction, a hypothetical scenario is considered. Imagine that an engi-

neer is charged with starting up a new reaction process based on the following chemical reaction:

$$A \rightarrow B,\ r_1 = k_1 c_A,\ k_1 = 0.05 s^{-1}. \tag{19}$$

Since this is the reaction that produces the product, B, the chemists who studied it were thorough and are sure that the reaction rate constant, k_1, is very accurate. However, they did not estimate the rate constant of the following known side reaction as precisely:

$$B \rightarrow C,\ r_2 = k_2 c_B,\ k_2 = \alpha_1 k_1,\ \alpha_1 \approx 0.5, \tag{20}$$

and suspect a second side reaction, but are not sure if it is significant:

$$A \rightarrow C,\ r_3 = k_3 c_A,\ k_3 = \alpha_2 k_1,\ \alpha_2 \approx 0. \tag{21}$$

The reactions are to take place in a batch reactor with capacity $V_{max} = 1000L$ that is equipped with an automatic feed and emptying system, with minimum and maximum flowrates $F_{min} = 0.01L/s$ and $F_{max} = 1L/s$. The process objective is to maximize the amount of product B produced per unit time, where the total time of a batch is the amount of time it takes to fill the reactor to its initial volume, V_0, plus the duration of the batch, T, plus the amount of time it takes to empty the reactor at the end of the batch. In order to simplify the problem, we will assume that the reactor will be operated in batch or semi-batch mode. More specifically, the feed to the reactor is pure A at a concentration of $c_{A0} = 2M$, and the reaction is either started with the reactor full of feed ($V_0 = V_{max}$), which is batch mode, or it is started only partially full ($V_0 < V_{max}$), and is filled the rest of the way during the first portion of the batch, using a constant flowrate, F, until time τ, when $V(\tau) = V_{max}$ and the flow is shut off, which is semi-batch mode. This process can be modeled by two sets of differential equations, one that is valid for $t < \tau$ ($\tau = 0$ for batch mode and $\tau = \frac{V_{max} - V_0}{F}$ for semi-batch mode):

$$\begin{aligned}
\frac{d\hat{n}_A}{dt} &= c_{A0}F - (1+\alpha_2)\, k_1 \hat{n}_A, \quad \hat{n}_A(0) = c_{A0}V_0 \\
\frac{d\hat{n}_B}{dt} &= k_1 \hat{n}_A - \alpha_1 k_1 \hat{n}_B, \qquad \hat{n}_B(0) = 0 \\
\frac{d\hat{V}}{dt} &= F, \qquad \hat{V}(0) = V_0
\end{aligned} \tag{22}$$

and another that is valid for $t \geq \tau$:

$$\begin{aligned}
\frac{dn_A}{dt} &= -(1+\alpha_2)\, k_1 n_A, \quad n_A(\tau) = \hat{n}_A(\tau) \\
\frac{dn_B}{dt} &= k_1 n_A - \alpha_1 k_1 n_B, \quad n_B(\tau) = \hat{n}_B(\tau) \\
\frac{dV}{dt} &= 0, \qquad V(\tau) = V_{max}.
\end{aligned} \tag{23}$$

Let us assume that the Implicit Optimization method is to be used to optimize this process. The first step is to maximize the objective function using the nominal model ($\alpha_1 = 0.5$ and $\alpha_2 = 0$), and to characterize the resulting solution. That optimization problem can be written as follows, noting that the above differential equations are simple enough that we can obtain an analytic expression for the number of moles of B present at the end of the batch, $n_B(V_0, F, T)$:

$$\begin{aligned}
&\max_{V_0,F,T} \frac{n_B(V_0,F,T)}{\frac{V_0}{F_{max}} + T + \frac{V_{max}}{F_{max}}} \\
&\text{s.t.} \\
&\quad 0 \leq V \leq V_{max} \\
&\quad F_{min} \leq F \leq F_{max} \\
&\quad \tau \leq T.
\end{aligned} \tag{24}$$

(In a more complex problem, the discretized ODE's would be equality constraints in the NLP.) The solution to this problem is $V_0 = 978.7L$, $F = 1L/s$ and $T = 27.7s$, which is characterized by noting that V_0 and T are at unconstrained values and that F is at its upper bound. If we implemented this Implicit Optimization solution directly we would use the given values as a starting point and adjust V_0 and T online using run-to-run control. However, it may very well be that this qualitative optimum is incorrect since α_1 and α_2 are so uncertain. Therefore, the three-step method outlined in the introduction should be used to evaluate the likelihood that this qualitative optimum is correct.

Step one (augmenting the nominal model with suspected structural elements and identifying uncertain model parameters) was done implicitly during the presentation of the problem. Specifically, the nominal model does not include the third reaction, Equation 21, and the uncertain parameters are α_1 and α_2. We will allow fairly wide bounds on these parameters for the purpose of solving the mp-NLP, namely

$$0 \leq \alpha_1, \alpha_2 \leq 5.0. \tag{25}$$

The second step is to solve the parametric optimization problem

$$\begin{aligned}
&(V_0, F, T)^*(\alpha_1, \alpha_2) = \\
&\quad \arg \left\{ \begin{array}{l} \max\limits_{V_0,F,T} \dfrac{n_B(V_0,F,T,\alpha_1,\alpha_2)}{\frac{V_0}{F_{\max}} + T + \frac{V_{\max}}{F_{\max}}} \\ s.t. \\ \quad 0 \leq V \leq V_{\max} \\ \quad F_{\min} \leq F \leq F_{\max} \\ \quad \tau \leq T \end{array} \right\}
\end{aligned} \tag{26}$$

using the method outlined in the second and third sections of this paper. The corresponding numerical results are shown in Figure 4, which depicts the approximation of M projected onto $\alpha_1 \times \alpha_2 \times V_0$. As you can see, the results show no qualitative change in the solution in the region about x_0, as no singularities were detected, however, the algorithm cannot yet fill in gaps that are left in order

to avoid overlap or follow singular paths. The addition of these features should both speed up the algorithm and allow more global exploration of the solution manifold. As the results stand now, there is no reason to believe that Implicit Optimization will not be successful for values of α near nominal, but the more global search forthcoming in future work may find that a different solution is the global optimum for some values of α near enough to the nominal values that the alternative solution(s) should be considered further.

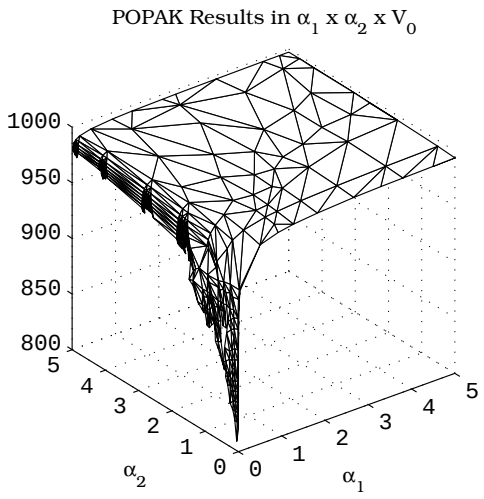

Fig. 1. Numerical results for the Implicit Optimization example. All of the points shown are feasible minimums.

5. CONCLUSIONS

Two novel methods have been presented: an algorithm for solving multi-parametric nonlinear programming (mp-NLP) problems, and a method for evaluating the adequacy of the nominal model used in Implicit Optimization (Srinivasan *et al.*, 2003), which relies heavily on solving an mp-NLP problem. The mp-NLP algorithm is designed to provide qualitative sensitivity information about the underlying NLP problem, particularly by defining regions in the parameter space where a single qualitative optimum holds. This emphasis is particularly useful for evaluating Implicit Optimization model adequacy since the main crux of this method is its dependance on the nominal optimum being qualitatively correct. This paper presents preliminary results for both of these methods, demonstrating their potential. Improvements to the mp-NLP algorithm that will allow more global exploration of the solution space are forthcoming in future work.

REFERENCES

Allgower, E. L. and K. Georg (1978). Triangulations by reflections with applications to approximation. In: *Numerische Methoden der Approximationstheorie* (L. Collatz, G. Meinardus and H. Werner, Eds.). Vol. 4. Birkhäuser Verlag. Basel, Switzerland.

Brodzik, M. L. (1996). Numerical Approximation of Manifolds and Applications. PhD thesis. University of Pittsburgh.

Brodzik, M. L. (1998). The computation of simplical approximations of implicitly defined p-dimensional manifolds. *Comput. Math. Appl.* **36**(6), 93–113.

Fink, J. P. and W. C. Rheinboldt (1987). A geometric framework for the numerical study of singular points. *SIAM J. Numer. Anal.* **24**(3), 618–633.

Gal, T. and J. Nedoma (1972). Multiparametric linear programming. *Manage. Sci.* **18**, 406.

Guddat, J., F. Guerra Vazquez and H. Th. Jongen (1990). *Parametric Optimization: Singularities, Pathfollowing and Jumps*. B. G. Teubner. Stuttgart.

Joe, B. (1993). Construction of k-dimensional delaunay triangulations using local transformations. *SIAM J. Sci. Comput.* **17**(6), 1415–1436.

Lawson, C. L. (1986). Properties of n-dimensional triangulations. *Computer Aided Geometric Design* **3**, 231–246.

Lundberg, B. N. and A. B. Poore (1993). Numerical continuation and singularity detection methods for parametric nonlinear programming. *SIAM J. Opt.* **3**(1), 134–154.

Mangasarian, O. L. and S. Fromovitz (1967). The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. *J. Math. Anal. Appl.* **17**, 37–47.

Rheinboldt, W. C. (1988). On the computation of multi-dimensional solution manifolds of parameterized equations. *Numer. Math.* **53**, 165–181.

Srinivasan, B., D. Bonvin, E. Visser and S. Palanki (2003). Dynamic optimization in batch processes: II. Role of measurements in handling uncertainty. *Comput. Chem. Eng.* **27**(1), 27–44.

Tøndel, P., T. A. Johansen and A. Bemporad (2003). An algorithm for mulit-parametric quadratic programming and explicit MPC solutions. *Automatica* **39**, 489–497.