A project on parameter identification in reaction kinetics

P.W. Hemker, J. Kok

Department of Numerical Mathematics

Report NM-R9301 January 1993

CWI is the National Research Institute for Mathematics and Computer Science. CWI is part of the Stichting Mathematisch Centrum (SMC), the Dutch foundation for promotion of mathematics and computer science and their applications. SMC is sponsored by the Netherlands Organization for Scientific Research (NWO). CWI is a member of ERCIM, the European Research Consortium for Informatics and Mathematics.

P.O. Box 4079, 1009 AB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

A Project on Parameter Identification in Reaction Kinetics

P.W. Hemker, J. Kok
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Abstract

The purpose of the project was to collect knowledge in parameter estimation, in order to be able to solve real-life parameter estimation problems from chemical engineering. A suitable method to solve these problems has been found and a computer program has been developed.

By means of the program several inverse problems for parameter estimation have been solved. For the solution, not only the parameters and their range of probability were identified, but it was also determined to what extent the information available from the experiments led to the possible identification of the parameters.

A summary of results is given in this report and subjects for further research are identified.

1991 Mathematics Subject Classification: Primary: 65L05. Secondary: 34A55, 65D10, 65H10, 80A30.

1991 CR Categories: G.1.2, G.1.6, G.1.7.

Keywords & Phrases: parameter identification, reaction kinetics, parameter estimation, inverse problems, chemical engineering.

Report NM-R9301
ISSN 0169-0388
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

CONTENTS

1. INTRODUCTION

A primary task in the modeling and understanding of chemical reaction systems is the identification of unknown parameters. These parameters are e.g. reaction constants in the kinetic equations or a priori unknown conditions. Given a description (a model) of the process under consideration, the unknown parameters can be identified when sufficient and suitable experimental data about the course of the reaction are available. Then, if the model is correct, the description with the parameters determined may account for the data and predict future behaviour.

Determining the model parameters is called the "inverse problem" of reaction kinetics, contrary to *simulation* which is the "direct problem". In simulations the course of the reaction is predicted given the reaction equations together with known parameters which describe the initial concentrations and circumstantial conditions.

It often occurs that the inverse problem — in which the parameters have to be determined — is *ill-posed*. This means that *several* choices for the parameters allow a model for which the simulation agrees with the given experimental data. In addition to this basic uncertainty, in practice two other sources for inaccuracies have to be taken into account:

- In the solution of the inverse problem, a number of direct problems have to be solved. For solving these direct problems complicated reaction systems must be integrated by numerical techniques. This process should be efficient and it should not introduce significant additional errors. This requires software implementations of advanced integration techniques.

- The (approximate) solution of a model should "fit" a set of data that is perturbed by experimental errors. The accuracy of these data determines the accuracy to which the parameters can be computed and it should determine the precision (the tolerances) used to control the induced optimisation problem.

For deriving the parameters that determine the qualitative and quantitative behaviour of chemical reactions, beside the model we need sufficient measurements as well as an appropriate mathematical method. Computers are indispensable for performing the simulations of the complex dynamical systems (describing the reactions of many chemical components) and also for fitting the simulations to the measurements. Existing numerical software is still not capable or sufficiently reliable for solving the inverse problem in all generality. Even good numerical mathematical analysis is lacking for providing a sound basis for truly generally applicable software.

A possible approach for treating the problem of parameter identification is to combine a modern simulation technique with a nonlinear least-squares method for data fitting. As this data fitting would require *global* nonlinear minimisation, such a direct combination usually does not yet result in a sufficiently successful method in practice.

Improvements can be expected by introducing the possibility of user interaction with the computer computation. This enables the exploitation of human expertise for the efficient solution of the problem. A prerequisite for such an approach is a suitable user interface, including sufficient means of 'scientific visualisation'[1]. Of course — with or without human interaction — statistical information obtained from the fitting problem (e.g. information about singular values and singular vectors) should be fed back into the simulation to discriminate the predictions resulting from the simulation. Further feedback might be given by the user in the form of suggestions during the computation to restrict the feasible parameter space, to drop certain measurements or to provide possible new measurements.

Here we report on the result of a pilot research project "Parameter Identification in Reaction Kinetics", in which advanced numerical techniques have been investigated for the solution of the inverse problem. An experimental implementation has been constructed (in Fortran) for one successful method which was used for several test problems and for some real-life problems. Further, the applicability has been greatly enhanced by incorporating symbolic manipulation techniques for

[1] For the elaboration of this idea, in Summer 1992 an application was made for STW support.

deriving the Jacobian matrices for the mathematical equations that describe the reaction equations. By modern computer graphics techniques, now available on the new SGI work stations, we are able to get additional insight into possibilities for global optimisation by inspecting the relevance of a problem's parameters through a cunning visualisation of their behaviour. These parts of the project are discussed in the following sections.

2. Objectives

The objectives of the research project were:

- To get insight into the availability of software aimed at chemical process modeling and the solution of chemical inverse problems, characterised by its user-friendliness and the communication language, the employed numerical techniques, and other (system-independent) properties that determine its usefulness for the chemical researcher.

  A comprehensive recent survey of available software for process simulation can be found in [4, 11]. For further useful information in this direction we refer to [28, 15].

- To investigate the numerical techniques that can be used as part of the solution process for inverse problems, in particular solution methods for (stiff) ordinary differential equations or differential-algebraic equations, solution methods for nonlinear least-squares problems and regularisation methods for ill-posed problems. Here we refer to [2, 7, 8, 10, 12].

- To implement pilot software for the solution of inverse problems.

  This software is briefly described in Section 4.

- To carry out case studies for a selection of typical inverse problems from practice.

  This experience is summarised in Section 5.

These activities were to be carried out in continuous interaction with representatives of the industrial project partner. In addition, several other relevant papers [33, 8, 19, 27] were studied and discussed in a small working group together with the industrial project partner.

3. Description of the problem

The mathematical formulation of the problem is the following. The physical model is described by a system of n differential (or differential algebraic equations[2])

$$\frac{d}{dt}\mathbf{y} \;=\; \mathbf{f}(t, \mathbf{y}, \mathbf{p})\,, \tag{3.1a}$$

and a corresponding set of initial conditions

$$\mathbf{y}(t_0, \mathbf{p}) \;=\; \mathbf{y}^0(\mathbf{p})\,. \tag{3.1b}$$

Here $\mathbf{y}$ is the n-vector of unknown functions, and $\mathbf{p}$ represents an m-vector of parameters. In the process considered, $\mathbf{p}$ has the value $\mathbf{p}^\star$, but $\mathbf{p}^\star$ is unknown. Some of the components of the vector $\mathbf{y}(t, \mathbf{p})$ can be measured for different values of t, but these measurements are perturbed by experimental errors. It is assumed that the form of $\mathbf{f}$ is known, together with some statistical properties of the experimental errors. The problem is to find an estimate $\overline{\mathbf{p}}$ for the vector $\mathbf{p}^\star$.

With y_i $(1 \le i \le N)$ we denote the observed value for the c_i-th component of $\mathbf{y}(t_i, \mathbf{p}^\star)$. Thus, we have a set of observations $\{y_i\}$ and for all i the index i identifies an observation y_i for the c_i-th

[2] In order to simplify the notation we restrict ourselves here to differential equations. Without essential differences the same treatment can be generalised to differential algebraic equations. In fact, also the software developed can be used for DAEs.

component of the system ($1 \leq c_i \leq n$) at time t_i, $t_0 < t_i \leq t_N$. Corresponding with these data, for a given $\mathbf{p}$ we can compute a set of theoretical values $\{\mathbf{y}_{c_i}(t_i, \mathbf{p})\}_{i=1}^{N}$.

If we assume a normal distribution for the measurement errors, and if we assume the classical linear statistics theory to hold, then the problem can be formulated as a least squares problem: define the N-vector of defects

$$\mathbf{Y}(\mathbf{p}) \quad = \quad (\mathbf{y}_{c_i}(t_i, \mathbf{p}) - y_i)_{i=1,\cdots,N} \,, \tag{3.2}$$

and introduce the sum of squares of the defects

$$S(\mathbf{p}) \quad = \quad \|\mathbf{Y}(\mathbf{p})\|_2^2 \quad = \quad \sum_{i=1}^{i=N} (\mathbf{y}_{c_i}(t_i, \mathbf{p}) - y_i)^2 \,. \tag{3.3}$$

Using an integration procedure to solve the system (3.1), we can solve the problem by minimising $S(\mathbf{p})$ over the space of all possible $\mathbf{p}$. Of course, the number of data N should be at least equal to the number of unknown parameters m. In principle, any good minimisation procedure could be used to solve this problem. However, a good global minimisation procedure for this problem is hard to find.

Moreover, even if we assume that the function $S(\mathbf{p})$ is the best one to minimise and that the minimum is unique, the question remains as how badly conditioned the problem is, i.e. *how small a perturbation in some values of y_i will cause how large a variation* in the minimising vector $\bar{\mathbf{p}}$. In relation to this question it is clear that not only an estimate for $\mathbf{p}^\star$ should be found, but also an estimate for its reliability.

In this report we assume that the experimental errors are statistically independent and that they have a Gaussian distribution with zero mean value. Further, we assume that linear statistics can be applied indeed, and that the measurements can be scaled such that they have equal variance σ^2. Thus the covariance matrix of the vector of experimental errors η is

$$\mathbf{E}(\eta\eta^T) \quad = \quad \sigma^2 \mathbf{I}, \tag{3.4}$$

and the probability density of η is given by

$$p(\eta) \quad = \quad (2\pi\sigma)^{-N/2} \exp(-\|\eta\|^2 \mathbf{I}/(2\sigma^2)) \,. \tag{3.5}$$

3.1. The dependence of $\mathbf{Y}(\mathbf{p})$ on $\mathbf{p}$.

The solution $\mathbf{y}$ of the system (3.1) can be considered as a function of t as well as a function of $\mathbf{p}$. We consider the difference between two adjacent solutions $\mathbf{y}_1(t, \mathbf{p})$ and $\mathbf{y}_2(t, \mathbf{p}+\delta\mathbf{p})$ of (3.1), both starting with the same initial condition $\mathbf{y}_1(t_0, \mathbf{p}) = \mathbf{y}_2(t_0, \mathbf{p} + \delta\mathbf{p}) = \mathbf{y}^0(\mathbf{p})$. For a small $\delta\mathbf{p}$ and differentiable functions $\mathbf{f}$, the difference between both solutions is, to first order, described by

$$\frac{d}{dt}(\mathbf{y}_2 - \mathbf{y}_1) \quad = \quad \frac{\partial \mathbf{f}(t, \mathbf{y}_1, \mathbf{p})}{\partial \mathbf{y}_1}(\mathbf{y}_2 - \mathbf{y}_1) + \frac{\partial \mathbf{f}(t, \mathbf{y}_1, \mathbf{p})}{\partial \mathbf{p}} \delta\mathbf{p} + \cdots \,. \tag{3.6}$$

With

$$\mathbf{FY}\,(t, \mathbf{y}, \mathbf{p}) \quad = \quad \frac{\partial \mathbf{f}(t, \mathbf{y}, \mathbf{p})}{\partial \mathbf{y}} \,, \tag{3.7}$$

an $n \times n$-matrix, and

$$\mathbf{FP}\,(t, \mathbf{y}, \mathbf{p}) \quad = \quad \frac{\partial \mathbf{f}(t, \mathbf{y}, \mathbf{p})}{\partial \mathbf{p}} \,, \tag{3.8}$$

an $n \times m$-matrix, both matrices in (3.6) depend on $\mathbf{p}$ and $\mathbf{y}_1$, but not on $\delta\mathbf{p}$ or $(\mathbf{y}_2 - \mathbf{y}_1)$.

To know the dependence of the defects (3.2) on $\mathbf{p}$, it is expedient to know also the functions

$$\mathbf{YP} \quad = \quad \frac{\partial \mathbf{y}(t, \mathbf{p})}{\partial \mathbf{p}} \,. \tag{3.9}$$

These m vector functions are determined by the differential equation (3.6), which — together with (3.1) — can be written as the system

$$\begin{aligned} \frac{d}{dt}\mathbf{y} &= \mathbf{f}\,, \\ \frac{d}{dt}\mathbf{YP} &= \mathbf{FY}\,.\,\mathbf{YP} + \mathbf{FP}\,, \end{aligned} \tag{3.10a}$$

with the initial conditions

$$\begin{aligned} \mathbf{y}(t_0) &= \mathbf{y}^0(\mathbf{p})\,, \\ \mathbf{YP}(t_0) &= \frac{\partial \mathbf{y}^0(\mathbf{p})}{\partial \mathbf{p}}. \end{aligned} \tag{3.10b}$$

The second part in this system (3.10) essentially is a system of $n \times m$ differential equations. If we solve this system together with the first system, (3.1), we are able to compute

$$\mathbf{A}(\mathbf{p}) = \left(\frac{\partial}{\partial \mathbf{p}}\mathbf{y}_{c_i}(t_i, \mathbf{p})\right)_{i=1,\cdots,N}. \tag{3.11}$$

This is an $N \times m$-matrix describing the dependence of the defects (3.2) on $\mathbf{p}$.

3.2. Minimising $S(\mathbf{p})$

Consider the function $S(\mathbf{p})$ defined by equation (3.3). The value $\overline{\mathbf{p}}$ that minimises $S(\mathbf{p})$ is an estimate of the true value $\mathbf{p}^\star$. In (3.3) $\mathbf{y}(t, \mathbf{p})$ is nonlinear function of $\mathbf{p}$. Therefore we assume that we may linearise this dependence on $\mathbf{p}$ in a sufficiently small neighbourhood of $\overline{\mathbf{p}}$, and that $\mathbf{p}^\star$ lies in this neighbourhood.

Suppose that $\mathbf{p}$ is a trial vector in the neighbourhood around $\overline{\mathbf{p}}$ and that $\delta\mathbf{p}$ is the required correction: $\mathbf{p} + \delta\mathbf{p} = \overline{\mathbf{p}}$. The residual vector $\mathbf{Y}(\mathbf{p})$ is approximated by a linear function of this parameter

$$\mathbf{Y}(\mathbf{p}) = \mathbf{Y}(\overline{\mathbf{p}} - \delta\mathbf{p}) \approx \mathbf{Y}(\overline{\mathbf{p}}) - \mathbf{A}(\mathbf{p})\,\delta\mathbf{p}\,, \tag{3.12}$$

and for the residual function

$$\begin{aligned} S(\overline{\mathbf{p}}) &= S(\mathbf{p} + \delta\mathbf{p}) = \|\mathbf{Y}(\mathbf{p} + \delta\mathbf{p})\|^2 \\ &\approx \|\mathbf{Y}(\mathbf{p}) + \mathbf{A}(\mathbf{p})\,\delta\mathbf{p}\|^2 \\ &= \|\mathbf{Y}(\mathbf{p})\|^2 + 2\delta\mathbf{p}^T\mathbf{A}^T\mathbf{Y}(\mathbf{p}) + \delta\mathbf{p}^T\mathbf{A}^T\mathbf{A}\delta\mathbf{p}\,. \end{aligned} \tag{3.13}$$

The approximating function to $S(\overline{\mathbf{p}})$ is a quadratic function of $\delta\mathbf{p}$ and it has a minimum at the point given by the normal equations

$$\mathbf{A}^T(\mathbf{p})\mathbf{A}(\mathbf{p})\,\delta\mathbf{p} = -\mathbf{A}^T(\mathbf{p})\,\mathbf{Y}(\mathbf{p})\,. \tag{3.14}$$

If the matrix $\mathbf{A}^T(\mathbf{p})\mathbf{A}(\mathbf{p})$ is non-singular, this equation determines $\delta\mathbf{p}$ from $\mathbf{Y}(\mathbf{p})$.

In the linear case, $\mathbf{p} + \delta\mathbf{p}$ so determined would be the required solution $\overline{\mathbf{p}}$, and the minimum value of $S(\mathbf{p})$ attained would be

$$S(\overline{\mathbf{p}}) = \|\mathbf{Y}(\mathbf{p})\|^2 - \delta\mathbf{p}^T\mathbf{A}^T\mathbf{A}\delta\mathbf{p}\,. \tag{3.15}$$

In the general, nonlinear case $S(\mathbf{p} + \delta\mathbf{p})$ will not be the minimal value of $S(\mathbf{p})$ and the process of approximating $\overline{\mathbf{p}}$ can be iterated.

The process described has the same order of convergence as quasi-linearisation. This process is often called quadratically convergent. In fact, however, the process only converges quadratically if the experimental errors vanish. Otherwise we have first order convergence.

The linear system (3.14) is the starting point for a *Gauss-Newton* method. The disadvantage of this method is well-known: it fails if the matrix $\mathbf{A}(\mathbf{p})$ is (nearly) singular. Therefore, the above Gauss-Newton method is modified and a *Levenberg-Marquardt* method [26] is used instead. Here the step vector is given by

$$\left(\mathbf{A}^T(\mathbf{p})\mathbf{A}(\mathbf{p}) + \lambda\mathbf{I}\right)\,\delta\mathbf{p} \;=\; -\mathbf{A}^T(\mathbf{p})\,\mathbf{Y}(\mathbf{p})\,, \tag{3.16}$$

where λ is some regularising parameter (a nonnegative scalar).

For $\lambda = 0$ the vector $\delta\mathbf{p}$ is equal to the vector defined by equation (3.14). If λ tends to infinity the direction of $\delta\mathbf{p}$ tends to the "steepest descent" direction and the length of $\delta\mathbf{p}$ tends to zero,

$$\delta\mathbf{p} \;\approx\; -\mathbf{A}^T(\mathbf{p})\,\mathbf{Y}(\mathbf{p})/\lambda\,.$$

The problem is now to find a proper choice for λ. In our implementation, equation (3.16) sometimes will be solved for different values of λ. In order to avoid superfluous calculations, and also in order to keep track of the singular values, the singular value decomposition of $\mathbf{A}(\mathbf{p})$ is used:

$$\mathbf{A}(\mathbf{p}) \;=\; \mathbf{U}(\mathbf{p})\,\Sigma(\mathbf{p})\,\mathbf{V}^T(\mathbf{p})\,, \tag{3.17}$$

where $\mathbf{U}(\mathbf{p})$ is a unitary $N \times m$-matrix, $\mathbf{V}(\mathbf{p})$ is a unitary $m \times m$-matrix, and the $m \times m$-matrix $\Sigma(\mathbf{p})$ is diagonal with on the main diagonal the singular values: $\Sigma(\mathbf{p}) = \mathrm{diag}(\sigma_1, \cdots, \sigma_m)$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$. Substituting (3.17) into (3.16) yields the step vector $\delta\mathbf{p}$ by

$$\delta\mathbf{p} \;=\; -\mathbf{V}(\mathbf{p})\left(\Sigma^2(\mathbf{p}) + \lambda\mathbf{I}\right)^{-1}\,\Sigma(\mathbf{p})\,\mathbf{U}^T(\mathbf{p})\,\mathbf{Y}(\mathbf{p})\,. \tag{3.18}$$

After a successful iteration step, i.e. after a computation of $\delta\mathbf{p}$ such that $S(\mathbf{p}+\delta\mathbf{p}) < S(\mathbf{p})$, the vector $\mathbf{p}+\delta\mathbf{p}$ is taken as a new approximation in a new iteration step. This makes a robust and efficient local minimisation procedure. The iteration process is stopped if the change in $S(\mathbf{p})$ is less than an a priori given tolerance. Of course, this tolerance is made dependent on the accuracy of the observations in $\mathbf{Y}(\mathbf{p})$.

3.3. Statistics

Let $\overline{\mathbf{p}}$ be the final estimate of $\mathbf{p}$ so that $S(\mathbf{p}) \geq S(\overline{\mathbf{p}})$ for all $\mathbf{p}$. We again assume that linearisation is allowed in a sufficiently large neighbourhood of $\overline{\mathbf{p}}$.

For the perturbations η_i of the observed values y_i, we assume an $N(0,\sigma^2)$ distribution[3], and so it follows from (3.14) that the estimated value $\overline{\mathbf{p}}$ will also be normally distributed. We define $\delta\mathbf{p} = \overline{\mathbf{p}} - \mathbf{p}^\star$, hence the expectation of $\delta\mathbf{p}$ will be zero when $\mathbf{p} = \overline{\mathbf{p}}$. We are also interested in the covariance matrix of $\delta\mathbf{p}$, i.e. the expected value of $\delta\mathbf{p}\delta\mathbf{p}^T$:

$$\begin{aligned} E(\delta\mathbf{p}\delta\mathbf{p}^T) \;&=\; E((\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{A}^T\mathbf{Y}\,\mathbf{Y}^T\mathbf{A}\,(\mathbf{A}^T\mathbf{A})^{-1}) \\ &=\; (\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{A}^T\,E(\mathbf{Y}\mathbf{Y}^T)\,\mathbf{A}\,(\mathbf{A}^T\mathbf{A})^{-1} \\ &=\; \sigma^2(\mathbf{A}^T\mathbf{A})^{-1}\,. \end{aligned}$$

By equation (3.14) $\delta\mathbf{p}$ is a linear function of $\mathbf{Y}$. Hence its probability density is also Gaussian and given by

$$P(\delta\mathbf{p}) \;=\; \frac{\exp(-\delta\mathbf{p}^T\mathbf{A}^T\mathbf{A}\delta\mathbf{p}/2\sigma^2)}{\sqrt{(2\pi\sigma)^m\,\det((\mathbf{A}^T\mathbf{A})^{-1})}}\,.$$

[3]For a distribution of the error in the experiments with different variances, in Section 3.6 we introduce possible scalings.

From (3.15) follows immediately

$$\|\mathbf{Y}(\bar{\mathbf{p}}+\delta\mathbf{p})\|^2 \quad = \quad S(\mathbf{p}) + \delta\mathbf{p}^T\mathbf{A}^T\mathbf{A}\delta\mathbf{p}\,. \tag{3.19}$$

Now it is clear that $\|\mathbf{Y}\|^2/\sigma^2$, $\delta\mathbf{p}^T\mathbf{A}^T\mathbf{A}\delta\mathbf{p}/\sigma^2$ and $S(\mathbf{p})/\sigma^2$ have a χ^2-distribution with N, m and $N-m$ degrees of freedom respectively. An estimate of σ^2 is given by

$$s^2 = S(\bar{\mathbf{p}})/(N-m) = \|\mathbf{Y}(\bar{\mathbf{p}})\|^2/(N-m)\,. \tag{3.20}$$

The confidence region at level α is the ellipsoidal region

$$\delta\mathbf{p}^T\mathbf{A}^T\mathbf{A}\delta\mathbf{p} \leq \frac{m}{N-m}S(\bar{\mathbf{p}})\,F_\alpha(m, N-m)\,, \tag{3.21}$$

where $F_\alpha(m, N-m)$ is the α-point of the F-distribution with m and $N-m$ degrees of freedom. The principal axes of the ellipsoidal region are given by the eigenvectors of $\mathbf{A}^T\mathbf{A}$ and the length of the axes is $1/\sqrt{\lambda_i}$, with λ_i the eigenvalue of the corresponding eigenvector.

The confidence limits for each estimate, supposing that the other estimates are correct, are

$$\bar{\mathbf{p}}_i \pm \delta\mathbf{p}_i\,,$$

where

$$\delta\mathbf{p}_i = \sqrt{\frac{m}{N-m}S(\bar{\mathbf{p}})F_\alpha/(\mathbf{A}^T\mathbf{A})_{ii}}\,.$$

Other confidence limits for the individual estimates (independently) are

$$\bar{\mathbf{p}}_i \pm \delta\mathbf{p}_i^\star\,,$$

where

$$\delta\mathbf{p}_i^\star = \sqrt{\frac{m}{N-m}S(\bar{\mathbf{p}})F_\alpha\,(\mathbf{A}^T\mathbf{A})^{-1}_{ii}}\,.$$

The geometrical interpretation is that the tangent planes to the ellipsoid with normals in the direction i are at a distance $\delta\mathbf{p}_i^\star$ from the centre of the ellipsoid and that the i^{th} axis cuts the ellipsoid at points $\delta\mathbf{p}_i$ from the centre. Clearly $\delta\mathbf{p}_i \leq \delta\mathbf{p}_i^\star$.

3.4. Integration of the differential equations

The system of differential (algebraic) equations that is to be solved at each iteration step of the minimisation process, is the large system (3.10). This large system can be considered as a family of $m+1$ smaller systems: the original system (3.1):

$$\frac{d}{dt}\mathbf{y} = \mathbf{f}$$

and m additional systems (see (3.10a)):

$$\frac{d}{dt}\mathbf{y}_{\mathbf{p}_j} = \mathbf{f}_{\mathbf{p}_j} + \mathbf{f}_\mathbf{y}\,\mathbf{y}_{\mathbf{p}_j} \quad , \; j = 1, \cdots, m.$$

Here $\mathbf{y}_{\mathbf{p}_j} = \partial\mathbf{y}/\partial\mathbf{p}_j$, $\mathbf{f}_{\mathbf{p}_j} = \partial\mathbf{f}/\partial\mathbf{p}_j$, and $\mathbf{f}_\mathbf{y} = \partial\mathbf{f}/\partial\mathbf{y}$ is the Jacobian matrix of the system (3.1). The Jacobian matrix of the large system (3.10) has the special structure

$$\mathbf{J} = \begin{pmatrix} \mathbf{f}_\mathbf{y} & 0 & \cdots & \cdots & 0 \\ \mathbf{f}_\mathbf{y}\mathbf{p}_1 & \mathbf{f}_\mathbf{y} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \mathbf{f}_\mathbf{y}\mathbf{p}_m & 0 & \cdots & 0 & \mathbf{f}_\mathbf{y} \end{pmatrix}, \tag{3.22}$$

where $\mathbf{f}_{\mathbf{yp}_j} = \partial^2 \mathbf{f}/\partial \mathbf{y} \partial \mathbf{p}_j$. In this Jacobian matrix we recognise a typical one-way coupling of the system. It follows that all the eigenvalues of the large Jacobian matrix coincide with those of the small Jacobian matrix $\mathbf{f}_\mathbf{y}$. Thus, the stability behaviour of the large system is the same as that of the small system and no additional stiffness is introduced.

In order to solve the system of differential (algebraic) equations a routine based on the BDF formulas has been constructed. In the usual way this routine is provided with step length and order control. Of course, this control is applied only for the small subsystem (3.1) because the behaviour of the large system is determined mainly by its eigenvalue distribution and the final accuracy of the minimisation is only dependent on the accuracy in the computation of (3.3).

In the implementation we take full advantage of the one-way coupling in (3.22). In each step of the integration process, the equation (3.1) is solved as an independent subsystem. When this part of the integration has been successfully completed, the m other systems in (3.10) are solved directly (e.g. using the same LU-decomposition in the solution of the implicit equations).

Another interesting feature of the BDF formulas used in the solution of (3.10) is the natural polynomial interpolation that is implied with the integration scheme. This interpolation is used when experimental data are available at t_i-points that are not step points for the BDF formula. As a consequence, the selection of the integration step points can be completely independent of the observation times $\{t_i\}$.

3.5. Implementation of the solution method

The approach in this project for solving the inverse problem has been to investigate the combination of a robust method for integrating (3.1) and a nonlinear minimisation technique for fitting the solution to the data. Based on the theory for linear statistics for the problem stated in (3.3), we apply *least-squares approximation*, which means that we want to find vectors $\mathbf{p}$ which minimise the *residual function*:

$$S(\mathbf{p}) = \mathbf{Y}(\mathbf{p})^{\mathrm{T}} \mathbf{Y}(\mathbf{p}) = \|\mathbf{Y}\|_2^2, \tag{3.23}$$

where $\|..\|_2$ denotes the ℓ_2-norm. In fact, taking into account different possible weights for the different experiments, we — more generally — introduce a *weighted* norm and minimise:

$$S(\mathbf{p}) = \mathbf{Y}(\mathbf{p})^{\mathrm{T}} W \mathbf{Y}(\mathbf{p}) = \|\mathbf{Y}\|_W^2 , \tag{3.24}$$

where W is a diagonal matrix of positive weights. By these weights we can take into account data about the accuracy of the individual measurements. For handling measurements with a fixed relative accuracy, a weights matrix W based on the magnitudes of the components of $\mathbf{y}$ can easily be generated automatically.

For the solution of the modified nonlinear least squares problem we use the *Levenberg-Marquardt* method described in section 3.2. As the minimisation procedure essentially finds local minima, we do not assume that a generally applicable method will result from such a direct combination of an IVP integrator and the local nonlinear minimisation routine. However, the present approach gives a good start and we expect that this method will enable us to analyse typical difficulties. The approach will allow us to eventually derive extensions to solve also the global problem. We foresee that the present technique will be part of a more comprehensive method, in which a different algorithm will take care of global minimisation and the present technique will deal with local problems.

In the project much attention was paid to the influence of scaling either (*i*) the parameters $\mathbf{p}$, or (*ii*) the dependent variables $\mathbf{y}$, or (*iii*) the vector of residuals. The next subsection is devoted to the formulation of the problem when all these possibilities are available.

3.6. Scaling components of the vectors $\mathbf{y}$, $\mathbf{p}$, *and* $\mathbf{Y}$

As a simple and yet sufficient means to deal with the sensitiveness of the problem or the solution method to different magnitudes in the elements of the vectors $\mathbf{p}$, $\mathbf{y}$, or $\mathbf{Y}(\mathbf{p})$, we have introduced facilities to separately scale these 3 vectors.

By scaling the components of the vectors $\mathbf{y}$ and $\mathbf{p}$ in (3.1) and (3.24) as:

$$\mathbf{y} = (\alpha_1 \tilde{y}_1, \ldots, \alpha_n \tilde{y}_n)^{\mathrm{T}}, \tag{3.25a}$$

and

$$\mathbf{p} = (\beta_1 \tilde{p}_1, \ldots, \beta_m \tilde{p}_m)^{\mathrm{T}}, \tag{3.25b}$$

we obtain the differential equations:

$$\alpha_i \frac{d}{dt} \tilde{y}_i(t, \tilde{\mathbf{p}}) = f_i(t, \{\alpha_j \tilde{y}_j\}_{j=1}^{n}, \{\beta_j \tilde{p}_j\}_{j=1}^{m})$$

or:

$$\frac{d}{dt} \tilde{y}_i(t, \tilde{\mathbf{p}}) = \tilde{f}_i(t, \tilde{\mathbf{y}}, \tilde{\mathbf{p}}).$$

This yields the, scaled, initial-value problem:

$$\begin{cases} \dfrac{d}{dt} \tilde{\mathbf{y}}(t, \tilde{\mathbf{p}}) &= \tilde{\mathbf{f}}(t, \tilde{\mathbf{y}}, \tilde{\mathbf{p}}), \\ \tilde{\mathbf{y}}(t_0, \tilde{\mathbf{p}}) &= \tilde{\mathbf{y}}^0(\tilde{\mathbf{p}}) \end{cases} \tag{3.26}$$

When elaborating, we find the following expression for the residual function:

$$\begin{aligned} S(\mathbf{p}) &= \|\mathbf{Y}\|_W^2 \\ &= \sum_{j=1}^{N} w_j \left(\alpha_{c_j} \tilde{y}_{c_j}(t^{(j)}, \tilde{\mathbf{p}}) - y_j\right)^2 \end{aligned} \tag{3.27}$$

in the case that W is a diagonal matrix with diagonal elements w_i. One possibility for W is:

$$w_i = \frac{1}{\alpha_{c_i}^2}$$

This means that the scaling of $\mathbf{y}$, which has its influence in the integration of the IVP, is annihilated by the scaling of the residual for the minimisation problem.

Experimenting with these scaling facilities showed in the first place, that an explicit, user-provided scaling for $\mathbf{p}$ had little influence. It obviously should not influence the solution found, but also the numerical behaviour of the nonlinear minimisation routine was hardly influenced. (This shows robustness of the minimisation procedure.)

As was to be expected, the scaling of $\mathbf{y}$ and $\mathbf{Y}(\mathbf{p})$ both influenced the solution process, as it modified the function to be minimised, and consequently it changed the solution $\mathbf{p}$. It was clear that this facility was indispensable for defining the very function that is to be minimised. Experience was obtained that in many cases a relative accuracy is given for measurements. For this we could use an automatic scaling facility.

In the final phase of the project the scaling facility (3.25) was exploited less frequently. As will be described in section 4.1, we have developed a new tool for defining the problem on a more elementary level than in the Fortran routines that describe the equations (3.1) (viz. in the `Maple` [13, 14] language). This tool provided a much easier way of introducing some scalings. The initially implemented facility for scaling, by (3.25) in the package of Fortran subroutines, was maintained for possible future efficiency improvements.

4. Description of the Parameter Identification software

The parameter identification software for solving inverse problems consists of two parts. The actual solver, called PEIDE, is implemented as a set of Fortran 77 subroutines. It is activated when its topmost subroutine (SUBROUTINE PEIDE) is called, and it solves the parameter identification problem that is defined by a set of four Fortran subroutines. The second part is a Maple program, called A_ENGINE, that can be used for generating the definition of the problem in Fortran. A_ENGINE delivers the four Fortran subroutines that are required for PEIDE, when a mathematical definition of the parameter identification problem is given as input to the Maple program.

We provide four ways for employing this software, depending on whether the user is willing to provide his or her own software for generating Jacobian matrices, monitoring and input and output processing, with user access to all possibilities for setting options. These four levels of access are:

1. **the most computer-assisted level.** The user only has to provide a mathematical model of the chemical reactions, and the data file(s) with measurements. The mathematical model is given as a system of differential (algebraic) equations and the corresponding initial values, as functions of the unknown parameters, in the Maple [13, 14] language[4]. Input for a complete run of PEIDE is generated by the Maple program A_ENGINE which uses advanced algebraic formula manipulation. The user can solve a parameter identification problem completely. The generated set of four Fortran subroutines which define the problem can be re-used in successive calls (possibly with different measurements) on the common technical level. Examples of problems solved are given in Section 5.

2. **the common technical level.** The user has to provide a definition of the parameter identification problem in the form of a set of four subroutines (DERIV, CYSTRT, JADFDY, JADFDP), and data files containing measurements and possibly scale factors. The subroutines DERIV, CYSTRT, JADFDY and JADFDP are required for defining the right-hand side function $\mathbf{f}(t, \mathbf{y}, \mathbf{p})$ of the initial-value problem, initialising the starting vector $\mathbf{y}^0(\mathbf{p})$, and obtaining the Jacobian matrices $\mathbf{FY}\,(t, \mathbf{y}, \mathbf{p})$ and $\mathbf{FP}\,(t, \mathbf{y}, \mathbf{p})$. These subroutines can be provided by an earlier use of the program on the **most computer-assisted level**.

   A driver program PIIRK is available which calls PEIDE in a standard way, using all available standard facilities for reading the measurements and printing the results.

   A lot of flexibility is left to the user, since many of the options for PEIDE are input-driven and are available through the standard data-reading subroutine. Therefore, for users employing many of the available facilities, this can be the usual working level.

3. **the advanced technical level.** As for the previous, higher level the user must provide a definition of the parameter identification problem through the set of four subroutines (DERIV, CYSTRT, JADFDY, JADFDP). In addition, the user also provides a main program calling PEIDE. Several tasks like reading input, monitoring the solver, processing the results, approximating the required Jacobian matrices, and providing scaling can be carried out by additional library subroutines that provide these facilities in a number of standard ways.

4. **the most technical level.** On the lowest level are provided the bare subroutines, PEIDE and its auxiliary library subroutines. Among other tasks, routines are available for solving an IVP, solving a nonlinear minimisation problem, reading input files or writing results. Further routines are available for subproblems such as the solution of a linear system, interpolation and the construction of a singular-value decomposition. Some linear algebra subproblems are solved by LAPACK subroutines (see [1]).

   The parameter identification problem must be defined by the four subroutines DERIV, CYSTRT, JADFDY and JADFDP. On this level, in addition, all other input parameters of PEIDE must be set,

[4] An example can be found in Section 4.1.

such as those for reading the measurements and the possible scalings. In a user-provided main program the user calls PEIDE and proceeds with the results obtained through PEIDE's output parameters. The ordinary user probably will never call PEIDE this way.

In the next two subsections we only describe the first two (highest-level) modes of use. A description of the other levels of use is not intended here as it would unavoidably be a long and very technical description.

4.1. The A_ENGINE *generator*

The highest level for using the solver is through the A_ENGINE program which generates Fortran source code for solving the inverse problem. In addition, it generates a LaTeX [22] input file for type-set output of the mathematical equations, derivatives and Jacobians.

The A_ENGINE process, as represented by Figure 1 (on page 13), consists of the following parts:

1. Let $PROBLEM (a formal parameter name) be the name of a problem. In the example below, $PROBLEM = barnes. Then the file $PROBLEM.model should contain a Maple definition of a problem (3.1).

2. With $PROBLEM.model as input file, A_ENGINE calls mapleV [13, 14]. At present, mapleV is the most recent available version of Maple. This call generates two new files: $PROBLEM.f and aproblem.tex.

3. The file $PROBLEM.f contains the four Fortran subroutines DERIV, CYSTRT, JADFDY and JADFDP. This is the required set of subroutines that define a parameter identification problem in Fortran. Together with the available driver program, and data files containing the measurements, this file contains all that is actually needed by the Fortran solver (PEIDE).

4. The file aproblem.tex is immediately included in a prepared file a_peide.tex and can be used as a LaTeX input file for generating a type-set report on the results of this A_ENGINE call.

As an example of the use of A_ENGINE, we treat here Barnes's problem described in section 5.1.1 (on page 23). This means that, in the description above, we take $PROBLEM = barnes.

The required input file barnes.model reads:

```
\# \begin{verbatim}

# First we fix the number of equations (noq),
# and the number of parameters (nop).

  noq:= 2;
  nop:= 3;

# We describe the differential equations by:
#    variables           Y
#    parameters          p
#    derivatives dY/dp = f(Y,p)

  YY := [seq( Y[m], m=1..noq )];
  PP := [seq( p[m], m=1..nop )];
  FF := [seq( f[m], m=1..noq )];

  x:= Y[1] ;
  y:= Y[2] ;
```

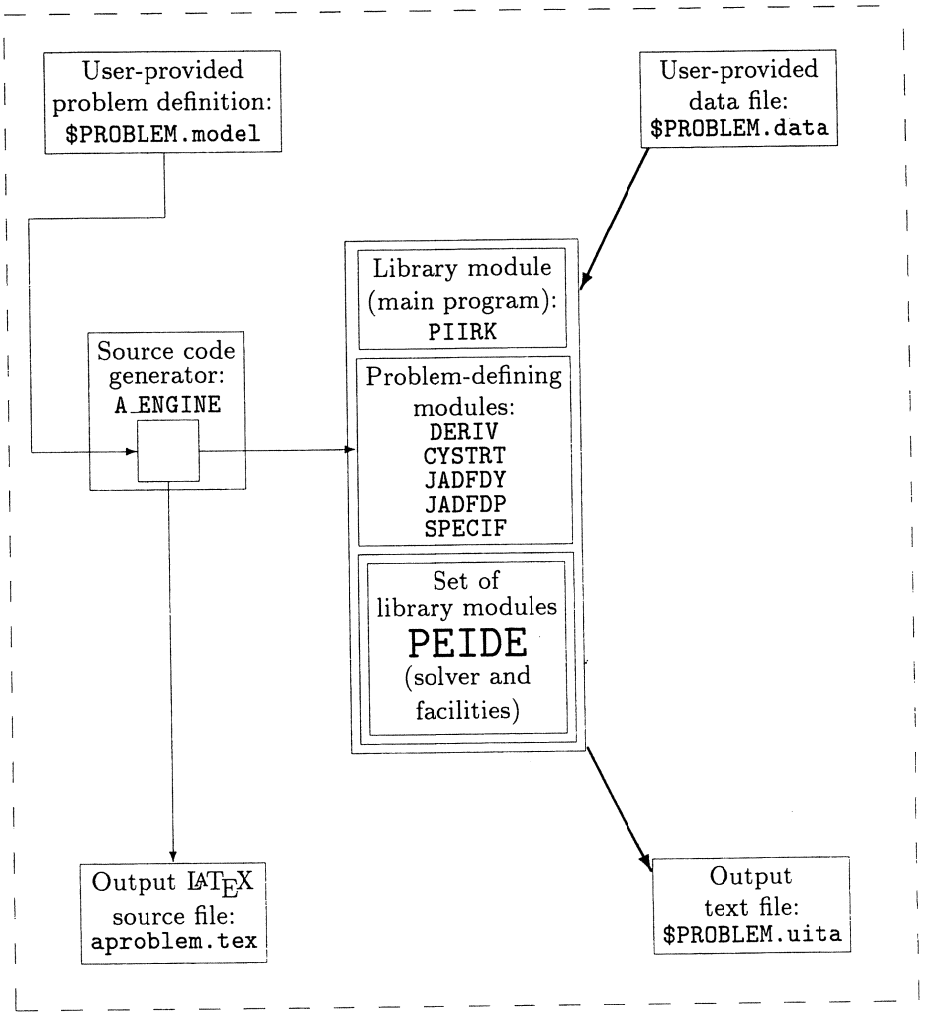

FIGURE 1. Model definition and I/O with A_ENGINE

```
  k1      := p[1] ;
  k2      := p[2] ;
  k3      := p[3] ;

  f[1] := k1 * x - k2 * x * y  ;
  f[2] := k2 * x * y - k3 * y ;

  ymax[1] := 1.0;
  ymax[2] := 1.0;

  ystart[1] := 1.0;
  ystart[2] := 0.3;

#                       required symbols at the end:
#                         noq, nop,
#                         Y,p,f,
#                         YY,PP,FF
#                         ymax, ystart,

\# \end{verbatim}
```

Note, that this file contains comment lines (lines preceded by #) that allow it to be easily included in a LaTeX file.

The Fortran source file `barnes.f`, generated by `A_ENGINE`, reads:

```
C begin of maples FORTRAN
C variable noq = N in PEIDE
C variable nop = M in PEIDE

      subroutine  CYSTRT(p,Y,YMAX,noq,nop)
      implicit double precision(t)
C  External to main program
      integer noq,nop
C     double precision p(nop),Y(noq),YMAX(noq)
C     In general, CYSTRT might use components of Y beyond Y(noq)
C     for initialising  dY/dP, viz. ((Y(5*noq+J*noq+I),I=1,noq),J=1,nop)
C     Therefore:
      double precision p(nop),Y(21300),YMAX(noq)
      YMAX(1) = 0.1D1
      YMAX(2) = 0.1D1
      Y(1) = 0.1D1
      Y(2) = 0.3D0
      Y(13) = 0
      Y(14) = 0
      Y(15) = 0
      Y(16) = 0
      Y(17) = 0
      Y(18) = 0
      return
C     end CYSTRT
      end
```

```
      subroutine SPECIF(POST,RIN,IOUT,ROUT)
      implicit double precision(t)
C  Called in generic main program PIIRK, intended for executing a
C  problem-SPECIFic prelude (POST=1),
C  interlude (POST=2), postlude (POST=3).
C
C  Parameters:
      integer POST,IOUT(10)
      double precision RIN(6),ROUT(4)
C
C  Declarations:
      integer nop, noq, NOBS, ININBP, NCOL, NROW, ITMAX
      integer BP(0:20),COBS(200),SCMODE
      logical GOON
      double precision TOBS(0:200),OBS(200),p(70),
     1   SCALEP(70),SCALEY(50),SCLRES(220)
      double precision LAMBDO, FISHAP
C  String variables for title:
      character TITLE*40
      integer INX
      double precision LOG
C  Commons for initialising in subroutine DAATA:
      common /XDAAT1/ nop,noq,NOBS,ININBP,NCOL,NROW,ITMAX,
     1   SCMODE,COBS,BP,GOON
      common /XDAAT2/ TOBS,OBS,p,SCALEP,SCALEY,SCLRES,
     1   LAMBDO,FISHAP
      common /XDAAT3/ TITLE
C
      intrinsic LOG
C ************************************************************ C
C
C  Replace these lines in the generated .f file by whatever is
C  needed here for the SPECIFic problem.
      return
C     end SPECIF
      end

      logical function    DERIV(p,Y,X,DF,noq,nop)
      implicit double precision(t)
C  External to main program
      integer noq,nop
      double precision X, p(nop),Y(noq),DF(noq)
C  Declaration:
         double precision dexp
         intrinsic dexp
      DF(1) = p(1)*Y(1)-p(2)*Y(1)*Y(2)
      DF(2) = p(2)*Y(1)*Y(2)-p(3)*Y(2)
      DERIV = .true.
      return
C     end DERIV
```

```
      end

      logical function    JADFDY(p,Y,X,FY,noq,nop)
      implicit double precision(t)
C  External to main program
      integer noq,nop
C Note: column length of array FY set to maximum (see main program):
C     double precision X, p(nop),Y(noq),FY(noq,noq)
      double precision X, p(nop),Y(noq),FY(50,noq)
         double precision dexp
         intrinsic dexp
      t1 = p(2)*Y(1)
      t3 = p(2)*Y(2)
      FY(1,2) = -t1
      FY(2,1) = t3
      FY(2,2) = t1-p(3)
      FY(1,1) = p(1)-t3
      JADFDY = .true.
      return
C     end JADFDY
      end

      logical function    JADFDP(p,Y,X,FP,noq,nop)
      implicit double precision(t)
C  External to main program
      integer noq,nop
C Note: column length of array FP set to maximum (see main program):
C     double precision X,p(nop),Y(noq),FP(noq,nop)
      double precision X,p(nop),Y(noq),FP(50,nop)
C  Declaration:
         double precision dexp
         intrinsic dexp
      t1 = Y(1)*Y(2)
      FP(1,2) = -t1
      FP(2,1) = 0
      FP(2,3) = -Y(2)
      FP(2,2) = t1
      FP(1,3) = 0
      FP(1,1) = Y(1)
      JADFDP = .true.
      return
C     end JADFDP
      end

C end of maples FORTRAN
```

The file `aproblem.tex` is intended as an include file for a complete LaTeX input file. When processed by LaTeX, the part `aproblem.tex` results in the following text (which is obtained as part of a more complete document that further consists of a title page, a copy of the `Maple` input file and possibly more information about the particular problem):

The equations are

$$DF_1 = p_1 Y_1 - p_2 Y_1 Y_2$$

$$DF_2 = p_2 Y_1 Y_2 - p_3 Y_2$$

The Jacobian df/dy is

$$\begin{bmatrix} p_1 - p_2 Y_2 & -p_2 Y_1 \\ p_2 Y_2 & p_2 Y_1 - p_3 \end{bmatrix}$$

The Jacobian df/dp is

$$\begin{bmatrix} Y_1 & -Y_1 Y_2 & 0 \\ 0 & Y_1 Y_2 & -Y_2 \end{bmatrix}$$

The maximal values for y are

$$YMAX_1 = 1.0$$

$$YMAX_2 = 1.0$$

The starting values for y are

$$Y_1 = 1.0$$

$$Y_2 = 0.3$$

The starting values for dy/dp are

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Here only the possible non-zero derivatives of the initial conditions with respect to the parameters are mentioned.

4.2. The PIIRK *program*

The second level of use of the software for solving parameter identification problems from reaction kinetics is by directly running a Fortran program, PIIRK. The PIIRK program is employed by providing a set of Fortran 77 subroutines (DERIV, CYSTRT, JADFDY, JADFDP) that together define the problem. PIIRK is a main program acting as a general driver program for PEIDE.

PIIRK makes a call to the subroutine PEIDE with all subroutine parameters correctly set. It arranges that a data file containing measurements (and weights, if user-provided) is read and it calls a library subroutine for printing results.

In Figure 2 (on page 18) we show the relationship between: the library modules, the user-provided modules for defining a problem, and the input and output files.

A complete heading mentioning all parameters of PEIDE is:

```
      subroutine  PEIDE(N,M,NOBS,NBP,PAR,RES,INIBP,JTJINV,L1JTJ,
     1   ITMAX,RIN,IOUT,ROUT,NCOL,NROW,INTOBS,INCOBS,INOBS,
     2   YOUT,L1YOUT,SCMODE,SCALEP,SCALEY,SCLRES,IMONIT)
```

A detailed specification of these parameters, and in particular of the user-provided subroutines that define parts of a parameter identification problem is available, but is not included in this report.

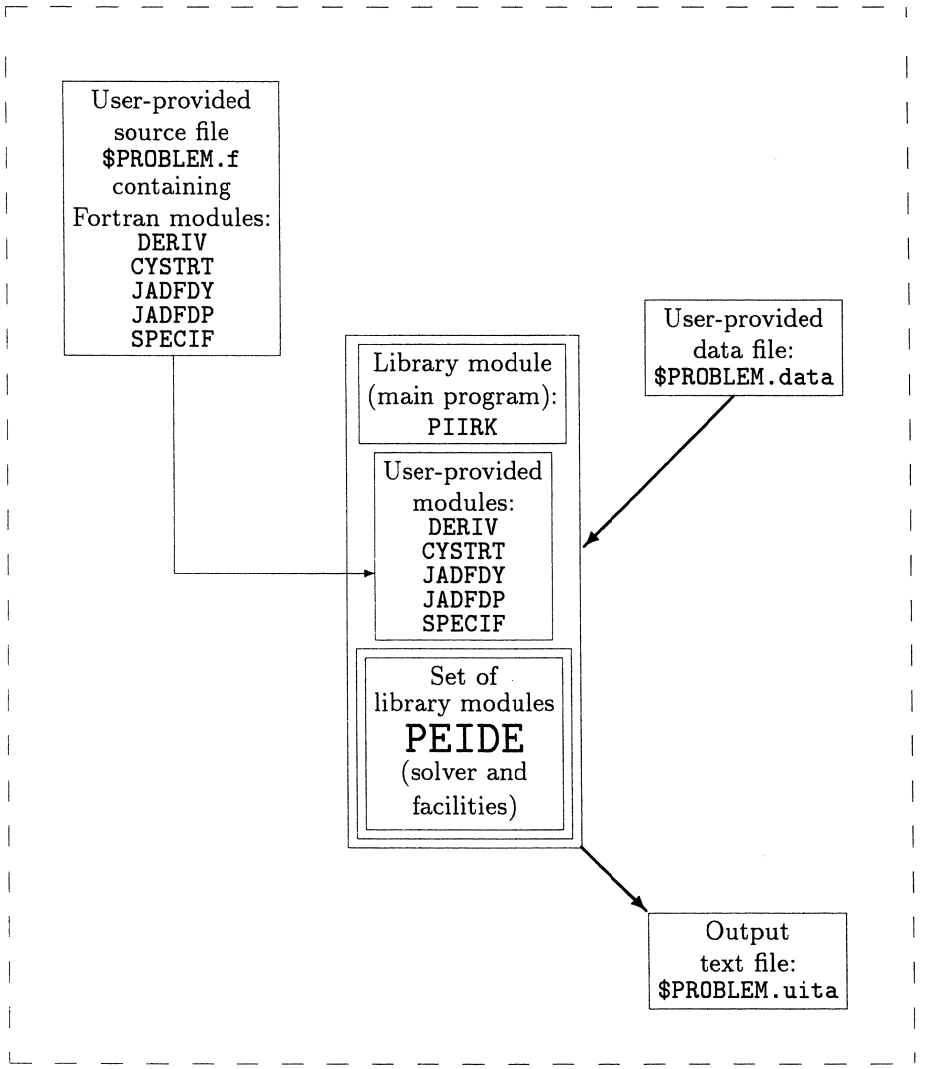

FIGURE 2. Problem definition and I/O with PEIDE

For Barnes's problem, that was used in the example of use of A_ENGINE in the previous section, the only input still needed by the executing Fortran program (after the part barnes.f has been generated by A_ENGINE), is the data file given in Figure 3.

```
B A R N E S - PROBLEM
     3     2    20     0     0    24     0
    0.0100    5.1800
 0    0.0000
 1    0.5000       1     1.1000
 2    0.5000       2     0.3500
 3    1.0000       1     1.3000
 4    1.0000       2     0.4000
 5    1.5000       1     1.1000
 6    1.5000       2     0.5000
 7    2.0000       1     0.9000
 8    2.0000       2     0.5000
 9    2.5000       1     0.7000
10    2.5000       2     0.4000
11    3.0000       1     0.5000
12    3.0000       2     0.3000
13    3.5000       1     0.6000
14    3.5000       2     0.2500
15    4.0000       1     0.7000
16    4.0000       2     0.2500
17    4.5000       1     0.8000
18    4.5000       2     0.3000
19    5.0000       1     1.0000
20    5.0000       2     0.3500
        1.0D0
        1.0D0
        1.3D0
go
```

FIGURE 3. Data file for Barnes's problem

The meaning of this input is:

first line: a title to identify the problem, that will appear in the output file,

second and third line: the following numbers:

m = number of parameters,
n = number of equations,
N = number of measurements,
NBP = number of break points,
a selector for the required scaling,
$ITMAX$ = maximum number of iterations,
a selector for the preferred way of monitoring (choosing the required amount of intermediate output),
λ_0 = initial value for the regularising parameter λ of the *Marquardt* subroutine (see (3.16)),
the α−point of the F-distribution,

next 21 lines: the measurements:

on each line $(i, t_i, COBS_i, OBS_i)$,

next three lines: an initial guess for the unknown parameters,

last line: the command go (nogo if only consistency checking of the input is required).

The standard output for Barnes's problem, obtained when the library-provided driver program is used, is (we suppress the repetition of the input file containing the measurements):

```
                    B A R N E S - PROBLEM


 NUMBER OF EQUATIONS    :  2
 NUMBER OF OBSERVATIONS: 20

 MACHINE PRECISION                                 :0.10E-13
 RELATIVE LOCAL ERROR BOUND FOR INTEGRATION        :0.10E-04
 RELATIVE TOLERANCE FOR RESIDUE                    :0.10E-03
 ABSOLUTE TOLERANCE FOR RESIDUE                    :0.10E-03
 MAXIMUM NUMBER OF INTEGRATIONS TO PERFORM         : 24
 RELATIVE STARTING VALUE OF LAMBDA                 :0.10E-01
 RELATIVE MINIMAL STEPLENGTH                       :0.10E-03


 THERE ARE NO BREAK-POINTS

 THE ALPHA-POINT OF THE F-DISTRIBUTION : 5.18


 NORMAL TERMINATION OF THE PROCESS


 LAST INTEGRATION WAS PERFORMED WITHOUT BREAK POINTS



 EUCL. NORM OF THE LAST RESIDUAL VECTOR  :0.4057437E+00
 EUCL. NORM OF THE FIRST RESIDUAL VECTOR :0.4510949E+01
 NUMBER OF INTEGRATIONS PERFORMED        :  7
 LAST IMPROVEMENT OF THE EUCLIDEAN NORM  :0.4916916E-05
 CONDITION NUMBER OF  J'*J               :0.1338214E+02
 LOCAL ERROR BOUND WAS EXCEEDED (MAXIM.) :  0



  PARAMETERS             CONFIDENCE INTERVAL

 0.8606433E+00            0.2003386E+00
 0.2072371E+01            0.3537576E+00
```

0.1815006E+01 0.3774078E+00

CORRELATION MATRIX

I\ J=	1	2
2	0.60091E+00	
3	0.55704E+00	0.84167E+00

COVARIANCE MATRIX

I\ J=	1	2	3
1	0.25827E-02		
2	0.27405E-02	0.80531E-02	
3	0.27102E-02	0.72312E-02	0.91658E-02

THE LAST RESIDUAL VECTOR:

column 1: OBS(I)
column 2: Y(c_I)
column 3: Y(c_I) - OBS(I)
column 4: sqrt(w_I) * OBS(I)
column 5: sqrt(w_I) * Y(c_I)
column 6: sqrt(w_I) * (Y(c_I) - OBS(I))

	1	2	3	4	5	6
1	0.11000E+01	0.10963E+01	-.36986E-02	0.11000E+01	0.10963E+01	-.36986E-02
2	0.35000E+00	0.36042E+00	0.10424E-01	0.35000E+00	0.36042E+00	0.10424E-01
3	0.13000E+01	0.11051E+01	-.19492E+00	0.13000E+01	0.11051E+01	-.19492E+00
4	0.40000E+00	0.45947E+00	0.59471E-01	0.40000E+00	0.45947E+00	0.59471E-01
5	0.11000E+01	0.10006E+01	-.99414E-01	0.11000E+01	0.10006E+01	-.99414E-01
6	0.50000E+00	0.55665E+00	0.56654E-01	0.50000E+00	0.55665E+00	0.56654E-01
7	0.90000E+00	0.84510E+00	-.54903E-01	0.90000E+00	0.84510E+00	-.54903E-01
8	0.50000E+00	0.58450E+00	0.84500E-01	0.50000E+00	0.58450E+00	0.84500E-01
9	0.70000E+00	0.72573E+00	0.25732E-01	0.70000E+00	0.72573E+00	0.25732E-01
10	0.40000E+00	0.52932E+00	0.12932E+00	0.40000E+00	0.52932E+00	0.12932E+00
11	0.50000E+00	0.67534E+00	0.17534E+00	0.50000E+00	0.67534E+00	0.17534E+00
12	0.30000E+00	0.43877E+00	0.13877E+00	0.30000E+00	0.43877E+00	0.13877E+00
13	0.60000E+00	0.68888E+00	0.88880E-01	0.60000E+00	0.68888E+00	0.88880E-01
14	0.25000E+00	0.35720E+00	0.10720E+00	0.25000E+00	0.35720E+00	0.10720E+00
15	0.70000E+00	0.75438E+00	0.54382E-01	0.70000E+00	0.75438E+00	0.54382E-01
16	0.25000E+00	0.30325E+00	0.53247E-01	0.25000E+00	0.30325E+00	0.53247E-01
17	0.80000E+00	0.85929E+00	0.59292E-01	0.80000E+00	0.85929E+00	0.59292E-01
18	0.30000E+00	0.28155E+00	-.18446E-01	0.30000E+00	0.28155E+00	-.18446E-01
19	0.10000E+01	0.98334E+00	-.16665E-01	0.10000E+01	0.98334E+00	-.16665E-01
20	0.35000E+00	0.29504E+00	-.54958E-01	0.35000E+00	0.29504E+00	-.54958E-01

--- Performance statistics:
8 calls of subroutine FUNCT

```
  7 calls of subroutine FUNCT (where SEC=.false.)
  1 calls of subroutine MARQUR
887 calls of subroutine DERIV
422 calls of subroutine JADFDP
429 calls of subroutine JADFDY
446 calls of subroutine DGETRF.


THE CALCULATION IN PEIDE CONSUMED      0.65  SECONDS
```

5. Submitted problems

5.1. Small example problems

We first have successfully tested the PEIDE package on a set of small, partly artificial, test problems (see Table 1). These problems are described in some detail in the Sections 5.1.1–5.1.7.

problem file	number of			description
	params.	variables	observations	
barnes.f	3	2	20	Barnes's problem
gear.f	4	2	8	Gear's problem
bellman.f	2	1	14	Bellman's problem
escep.f	3	2	23	ESCEP
small.f	1	1	4	param. in initial value
expfit.f	5	2	17	exponential fitting
enzym.f	4	2	27	enzyme effusion

Table 1. Example problems

For the problems run with very bad initial estimates of the parameters, the success of the solution method may have depended on the setting of options of the software that were not explained in detail in Section 4.2, like the tuning of the Marquardt parameter λ and the choice of break points for the system of differential equations. Such technical information is omitted here.

5.1.1. Barnes's problem

Barnes's problem [20, 17]

$$\begin{aligned} A &\xrightarrow{p_1} A + A \\ A + B &\xrightarrow{p_2} B + B \\ B &\xrightarrow{p_3} C\,. \end{aligned} \tag{5.1a}$$

The mathematical problem is given by the equations:

$$\begin{aligned} \frac{dy_1}{dt} &= p_1 y_1 - p_2 y_1 y_2 \\ \frac{dy_2}{dt} &= p_2 y_1 y_2 - p_3 y_2\,. \end{aligned} \tag{5.1b}$$

For this problem the initial values were obtained from the initial concentrations in the reactions. The initial values were:

$$y_1(0) = 1\,, \quad y_2(0) = 0.3 \tag{5.1c}$$

A typical set of input data (files `barnes.model` and `barnes.data`, for both the model and the measurements) was described in Section 4. For the call of PEIDE we used $\mathbf{p}^{(0)} = (1, 1, 1.3)^{\mathrm{T}}$ as starting values for the parameters. From the results obtained we mention only the final values of the parameters $\bar{\mathbf{p}}$ and the confidence interval for each of the parameters:

PARAMETERS	CONFIDENCE INTERVAL
0.8606433E+00	0.2003386E+00
0.2072371E+01	0.3537576E+00
0.1815006E+01	0.3774078E+00

5.1.2. Gear's problem

Gear's problem [20, 17] apparently originated from the following set of chemical reactions:

$$\begin{aligned} A + B &\underset{p_3}{\overset{p_4}{\rightleftarrows}} AB \\ AB + B &\underset{p_1}{\overset{p_2}{\rightleftarrows}} ABB\,. \end{aligned} \tag{5.2a}$$

Taking for the concentrations $[ABB] = y_1$ and $[AB] = y_2$, the mathematical problem is given by the equations:

$$\begin{aligned} \frac{dy_1}{dt} &= -p_1 y_1 + p_2 y_2 (2 - y_2 - 2y_1) \\ \frac{dy_2}{dt} &= -p_3 y_2 + p_4 (2 - y_2 - 2y_1)(1 - y_2 - y_1) - \frac{dy_1}{dt}\,. \end{aligned} \tag{5.2b}$$

The initial values for $\mathbf{y}$ were:

$$y_1(0) = 0.25\,, \; y_2(0) = 0.5 \tag{5.2c}$$

For a submitted set of measurements the solver succeeded in finding a solution for $\mathbf{p}$ with a small residual vector.

5.1.3. Bellman's problem

Bellman's problem [20, 17] is derived from the chemical reaction:

$$2NO + O_2 \rightleftarrows 2NO_2 \,. \tag{5.3a}$$

The mathematical problem is given by the equation:

$$\frac{dy_1}{dt} = p_1(126.2 - y_1)(91.9 - y_1)^2 - p_2 y_1^2 \,. \tag{5.3b}$$

The initial value was:

$$y_1(1) = 0 \tag{5.3c}$$

For a submitted set of measurements the solver succeeded in finding a solution for $\mathbf{p}$ with a small residual vector.

5.1.4. The ESCEP problem

The ESCEP problem [20, 17] is derived from the following set of chemical reactions originating from biochemistry:

$$\begin{aligned} E + S &\underset{k_2}{\overset{k_1}{\rightleftarrows}} C \\ C &\xrightarrow{k_3} E + P \,. \end{aligned} \tag{5.4a}$$

The variables of the mathematical problem are the concentrations $[C]$ and $[S]$. The mathematical problem is given by the equations:

$$\begin{aligned} \frac{dy_1}{dt} &= -(1 - y_2)y_1 + k_2 y_2 \\ \frac{dy_2}{dt} &= k_1\left((1 - y_2)y_1 - (k_2 + k_3)y_2\right) \,, \end{aligned} \tag{5.4b}$$

and the initial values were:

$$y_1(0) = 1 \,,\; y_2(0) = 0 \tag{5.4c}$$

For a submitted set of measurements the solver succeeded in finding a solution for $\mathbf{p}$ with a small residual vector.

5.1.5. A small test problem

The following small test problem was chosen in order to test the performance when the initial value for $\mathbf{y}$ depends on $\mathbf{p}$. The mathematical problem is given by the equation:

$$\frac{dy_1}{dt} = 0 \,, \tag{5.5a}$$

with initial condition:

$$y_1(0) = p_1 \,. \tag{5.5b}$$

For the call of PEIDE we used $p_1^{(0)} = 0.5$ as starting values for the parameter, and the observations chosen for this test problem were:

$y_1(1.0) = 1.0, y_1(2.0) = 1.0, y_1(3.0) = 2.0, y_1(4.0) = 2.0.$

From the results obtained we mention only the final value of the parameter $\overline{\mathbf{p}}$ and the confidence interval for this value:

PARAMETERS	CONFIDENCE INTERVAL
0.1500000E+01	0.1685724E+01

It shows that the solver correctly delivered the best solution $p_1 = 1.5$, although the errors in the data are such that no accuracy can be expected.

5.1.6. The exponential fitting problem

As for the previous, small test problem, the exponential fitting problem [20, 17] was chosen in order to test the performance when the initial value for $\mathbf{y}$ depends on $\mathbf{p}$. The problem was derived from a function

$$y_1(t) = p_5 + p_1 e^{p_2 t} + p_3 e^{p_4 t}, \tag{5.6a}$$

which is known to satisfy the equations:

$$\begin{aligned} \frac{dy_1}{dt} &= y_2 \\ \frac{dy_2}{dt} &= -p_4 p_2 y_1 + (p_4 + p_2) y_2 + p_4 p_2 p_5 . \end{aligned} \tag{5.6b}$$

The initial conditions were:

$$\begin{aligned} y_1(0) &= p_1 + p_3 + p_5 \\ y_2(0) &= p_1 p_2 + p_3 p_4 . \end{aligned} \tag{5.6c}$$

For the call of PEIDE we used $\mathbf{p}^{(0)} = (-5, -10, 5, -0.5, 0.5)^{\mathrm{T}}$ as starting values for the parameters. In agreement with [17] we took as observations values of $y_1(t)$ for t at:

$$0.02(0.02)0.1, 0.2(0.2)1, 2, 3, 4, 5(5)20$$

for the parameters:

$$\mathbf{p}^{\star} = (-3, -20, 2, -1, 1)^{\mathrm{T}}.$$

From the results obtained we mention only the final values of the parameters $\overline{\mathbf{p}}$ and the confidence interval for each of the parameters:

PARAMETERS	CONFIDENCE INTERVAL
-.2999995E+01	0.1503825E-03
-.1999901E+02	0.1335448E-02
0.2000049E+01	0.5382094E-04
-.1000026E+01	0.6881261E-04
0.9999989E+00	0.2185958E-04

This shows that, for the given starting values for $\mathbf{p}$, the solver correctly recovers the values of the parameters used for generating the data.

5.1.7. The enzyme effusion problem

The enzyme effusion problem [16, 17] is involved with fitting a model of enzyme effusion into the blood after a heart infarct. The mathematical problem is given by the equations:

$$\begin{aligned} \frac{dy_1}{dt} &= -p_1 y_1 + \frac{p_4}{v_1}(y_2 - y_1) + p_1 y_{lim} + \frac{q}{t\sqrt{2\pi}} \exp\left(-0.5\left(\frac{\ln(t) - p_2}{p_3}\right)^2\right) \\ \frac{dy_2}{dt} &= \frac{p_4}{v_2}(y_1 - y_2)\,. \end{aligned} \tag{5.7a}$$

The initial conditions were:

$$y_1(0.1) = y_2(0.1) = y_{lim}\,. \tag{5.7b}$$

The quantities v_1, v_2, q, y_{lim} are known variables of the problem for which the values per occurrence of the problem (i.e. per patient) are given.

For a run of PEIDE with 27 measurements from a real-life situation (see [17]) we used $\mathbf{p}^{(0)} = (0.16, 2.6, 0.3, 0.32)^{\mathrm{T}}$ as starting values for the parameters.

From the results obtained we mention only the final values of the parameters $\overline{\mathbf{p}}$ and the confidence interval for each of the parameters:

PARAMETERS	CONFIDENCE INTERVAL
0.2725351E+00	0.7859745E-01
0.2653203E+01	0.1180816E+00
0.3661048E+00	0.9868416E-01
0.2081668E+00	0.2901210E+00

These results roughly agree with those given in [17].

5.2. Case problems

5.2.1. A real-life problem from reaction kinetics

The problem treated in this section was taken from the Master's thesis by K.K. Hong [21]. It arises from the reactions occurring in a reactor vessel containing a couple of reagents and a catalyst. Reagents occur both as liquids and gas. Measurements are obtained from those concentrations in the process that appeared to be measurable, which was the case for only a few of the concentrations.

The reactions are:

$$\begin{aligned} 2y_8 + y_1 &\rightleftarrows y_3 \\ 2y_8 + y_3 &\rightleftarrows y_4 \\ y_3 + y_4 &\rightleftarrows y_5 + y_6 \\ 2y_8 + y_6 &\rightleftarrows y_7 \\ y_2 &\rightleftarrows 2y_8\,. \end{aligned} \tag{5.8a}$$

The mathematical problem is (cf.[21]):

$$
\begin{array}{lcl|lcl}
\dfrac{dy_1}{dt} &=& -r_1 g_c/V_\ell\,, & \dfrac{dy_5}{dt} &=& r_3 - v_1/V_\ell\,, \\
\dfrac{dy_2}{dt} &=& -r_5 g_c/V_\ell + (w_1+w_2)/V_\ell\,, & \dfrac{dy_6}{dt} &=& r_3 - r_4 g_c/V_\ell\,, \\
\dfrac{dy_3}{dt} &=& (r_1-r_2)g_c/V_\ell - r_3\,, & \dfrac{dy_7}{dt} &=& r_4 g_c/V_\ell\,, \\
\dfrac{dy_4}{dt} &=& r_2 g_c/V_\ell - r_3\,, & \dfrac{dy_8}{dt} &=& 2(r_5 - r_1 - r_2 - r_4)\,,
\end{array}
\tag{5.8b}
$$

where

$$
\begin{array}{lcl|lcl}
r_1 &=& k_1(y_1 y_8^2 - K_1 y_3)\,, & w_1 &=& k_{11}(q_1(p_{tot}/(RT) - y_5/q_2) - y_2)\,, \\
r_2 &=& k_2(y_3 y_8^2 - K_2 y_4)\,, & w_2 &=& k_{12} w_1 (q_t(p_{tot}/(RT)) - y_2)\,, \\
r_3 &=& k_3(y_3 y_4 - K_3 y_5 y_6)\,, & q_1 &=& q_{10} + a_1 p_5 + b_1 p_2\,, \\
r_4 &=& k_4(y_6 y_8^2 - K_4 y_7)\,, & q_2 &=& q_{20} + a_2 p_5 + b_2 p_2\,, \\
r_5 &=& k_5(y_2 - K_5 y_8^2)\,, & q_t &=& q_{10} + b_1 p_{tot}\,, \\
v_1 &=& r_3/(q_2/V_g + 1/V_\ell)\,, & & &
\end{array}
\tag{5.8c}
$$

in which $q_{10}, q_{20}, a_1, a_2, b_1, b_2, p_{tot}, R, T$ are physical constants or quantities that are kept constant for all experiments, and $g_c, p_2, p_5, V_g, V_\ell$ are quantities which are kept constant per experiment. Parameters are

$$
\mathbf{p} = (k_1, k_2, \ldots, k_5, K_1, K_2, \ldots, K_5, k_{11}, k_{12})^{\mathrm{T}}\,. \tag{5.8d}
$$

For this problem the initial values were obtained from the initial concentrations in the reactions. The non-zero initial values were:

$$
y_1(0) = 2907.0\,,\ y_5(0) = q_2 p_5/(RT) \tag{5.8e}
$$

Computations were carried out with the sets of measurements listed in an appendix of [21]. The measured concentrations were for $[y_3 + y_6], [y_4]$ and $[y_7]$, and measurements had been obtained in experiments with different values of g_c (the amount of catalyst), and p_2 and p_5 (the partial pressures of y_2 and y_5 in the gas volume).

In order to have the mathematical equations in the right shape for solving the problem by PEIDE, the variable y_6 was replaced by $z_6 = y_3 + y_6$.

We have carried out several solver runs while modifying starting values of the parameters, scaling of components of parameters, variables and residues, and other available options. Our experiments gave a clear indication that the chosen model was unsuitable for the data, as the solution for the resulting parameter values did never come close to the experimental data, and additional statistical information showed that the obtained parameter values could not bear accuracy.

For the first runs with the hong problem, the output also indicated the linear dependencies that existed (and should exist) between several of the not measured concentrations. Taking:

$$
\beta = \frac{q_2 V_\ell}{q_2 V_\ell + V_g}\,,
$$

the obvious dependencies are:

$$\frac{dy_1}{dt} + \frac{dy_3}{dt} + \frac{dy_4}{dt} + \frac{2}{\beta}\frac{dy_5}{dt} = 0 \quad \text{and} \tag{5.9a}$$

$$\frac{dy_5}{dt} - \beta\frac{dy_6}{dt} - \beta\frac{dy_7}{dt} = 0\,. \tag{5.9b}$$

Additional runs were made with a simplified problem obtained by eliminating the known linear dependencies in the variables. Although this had a positive effect on the execution time consumed by PEIDE, the effect on the accuracy of the parameters (shown, e.g., by the values given under CONFIDENCE INTERVAL) was not significant. Several runs have been made, in order to find good starting values for the vector of parameters **p**. The main conclusion was that the model did not fit the measurements, and that this was sufficiently indicated by the diagnostics issued by PEIDE.

5.2.2. A submitted real-life problem of a chemical reaction

A real-life problem submitted by the industrial project partner was treated employing the highest level of use of A_ENGINE together with PEIDE. The mathematical problem is derived from chemical reactions, and in experiments with different (but constant) temperatures all concentrations were measured a couple of times.

The reactions are:

$$\begin{aligned}
2y_1 + y_2 &\xrightarrow{r_1} y_3 + y_4 \\
y_3 + y_5 &\xrightarrow{r_2} y_6 + y_4 \\
y_6 + y_4 &\xrightarrow{r_3} y_7 + y_1 \\
y_7 &\xrightarrow{r_4} y_8 + y_5 \\
y_8 + y_5 &\xrightarrow{r_5} y_7\,.
\end{aligned} \tag{5.10a}$$

The mathematical description is:

$$\begin{aligned}
\frac{dy_1}{dt} &= r_3 - 2r_1\,, & \frac{dy_5}{dt} &= r_4 - r_5 - r_2\,, \\
\frac{dy_2}{dt} &= -r_1\,, & \frac{dy_6}{dt} &= r_2 - r_3\,, \\
\frac{dy_3}{dt} &= r_1 - r_2\,, & \frac{dy_7}{dt} &= r_3 - r_4 + r_5\,, \\
\frac{dy_4}{dt} &= r_1 + r_2 - r_3\,, & \frac{dy_8}{dt} &= r_4 - r_5\,,
\end{aligned} \tag{5.10b}$$

where

$$\begin{aligned}
r_1 &= k_1 y_1 y_2\,, & r_4 &= k_4 y_7\,, \\
r_2 &= k_2 y_3 y_5\,, & r_5 &= k_5 y_5 y_8\,. \\
r_3 &= k_3 y_4 y_6\,,
\end{aligned} \tag{5.10c}$$

For this problem the initial values, which were obtained from the initial concentrations in the reactions, were different per experiment. Both experiments with (non-zero) initial values:

$$y_1(0) = 100.0\,,\ y_2(0) = 150.0\,,\ y_5(0) = 100.0 \tag{5.10d}$$

and

$$\mathbf{y}(0) = (100.0, 142.89, 4.34, 0, 123.41, 3.49, 5.55, 9.13)^{\mathrm{T}} \tag{5.10e}$$

were made.

In the formulas given below we have introduced parameters p_i according to $k_i = \exp(p_i)$, to force the parameters k_i to be positive and to obtain relative instead of absolute accuracy. The use of `A_ENGINE` resulted in the following partial LaTeX output file:

The equations are

$$DF_1 = e^{p_3} Y_4 Y_6 - 2.0 e^{p_1} Y_1 Y_2$$

$$DF_2 = -e^{p_1} Y_1 Y_2$$

$$DF_3 = e^{p_1} Y_1 Y_2 - e^{p_2} Y_3 Y_5$$

$$DF_4 = e^{p_1} Y_1 Y_2 + e^{p_2} Y_3 Y_5 - e^{p_3} Y_4 Y_6$$

$$DF_5 = e^{p_4} Y_7 - e^{p_5} Y_5 Y_8 - e^{p_2} Y_3 Y_5$$

$$DF_6 = e^{p_2} Y_3 Y_5 - e^{p_3} Y_4 Y_6$$

$$DF_7 = e^{p_3} Y_4 Y_6 - e^{p_4} Y_7 + e^{p_5} Y_5 Y_8$$

$$DF_8 = e^{p_4} Y_7 - e^{p_5} Y_5 Y_8$$

The Jacobian df/dy is

$$\begin{bmatrix}
-2.0e^{p_1}Y_2 & -2.0e^{p_1}Y_1 & 0 & e^{p_3}Y_6 & 0 & e^{p_3}Y_4 & 0 & 0 \\
-e^{p_1}Y_2 & -e^{p_1}Y_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
e^{p_1}Y_2 & e^{p_1}Y_1 & -e^{p_2}Y_5 & 0 & -e^{p_2}Y_3 & 0 & 0 & 0 \\
e^{p_1}Y_2 & e^{p_1}Y_1 & e^{p_2}Y_5 & -e^{p_3}Y_6 & e^{p_2}Y_3 & -e^{p_3}Y_4 & 0 & 0 \\
0 & 0 & -e^{p_2}Y_5 & 0 & -e^{p_5}Y_8 - e^{p_2}Y_3 & 0 & e^{p_4} & -e^{p_5}Y_5 \\
0 & 0 & e^{p_2}Y_5 & -e^{p_3}Y_6 & e^{p_2}Y_3 & -e^{p_3}Y_4 & 0 & 0 \\
0 & 0 & 0 & e^{p_3}Y_6 & e^{p_5}Y_8 & e^{p_3}Y_4 & -e^{p_4} & e^{p_5}Y_5 \\
0 & 0 & 0 & 0 & -e^{p_5}Y_8 & 0 & e^{p_4} & -e^{p_5}Y_5
\end{bmatrix}$$

The Jacobian df/dp is

$$\begin{bmatrix}
-2.0e^{p_1}Y_1Y_2 & 0 & e^{p_3}Y_4Y_6 & 0 & 0 \\
-e^{p_1}Y_1Y_2 & 0 & 0 & 0 & 0 \\
e^{p_1}Y_1Y_2 & -e^{p_2}Y_3Y_5 & 0 & 0 & 0 \\
e^{p_1}Y_1Y_2 & e^{p_2}Y_3Y_5 & -e^{p_3}Y_4Y_6 & 0 & 0 \\
0 & -e^{p_2}Y_3Y_5 & 0 & e^{p_4}Y_7 & -e^{p_5}Y_5Y_8 \\
0 & e^{p_2}Y_3Y_5 & -e^{p_3}Y_4Y_6 & 0 & 0 \\
0 & 0 & e^{p_3}Y_4Y_6 & -e^{p_4}Y_7 & e^{p_5}Y_5Y_8 \\
0 & 0 & 0 & e^{p_4}Y_7 & -e^{p_5}Y_5Y_8
\end{bmatrix}$$

The maximal values for y are

$$YMAX_1 = 200.0$$

$$YMAX_2 = 200.0$$

$YMAX_3 = 200.0$

$YMAX_4 = 200.0$

$YMAX_5 = 200.0$

$YMAX_6 = 200.0$

$YMAX_7 = 200.0$

$YMAX_8 = 200.0$

The starting values for y are

$Y_1 = 100.0$

$Y_2 = 150.0$

$Y_3 = 0$

$Y_4 = 0$

$Y_5 = 100.0$

$Y_6 = 0$

$Y_7 = 0$

$Y_8 = 0$

The starting values for dy/dp are

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Here only the possible non-zero derivatives of the initial conditions with respect to the parameters are mentioned.

Several runs of PEIDE were made with different starts for $\mathbf{p}$ and with different sets of measurements. For the following measurements:

```
0     0.0000
1    25.0000        1     32.3600
2    25.0000        2    109.1800
3    25.0000        3      6.4600
4    25.0000        4     61.1800
5    25.0000        5     67.5800
6    25.0000        6     20.3500
7    25.0000        7     12.0700
```

```
 8   25.0000      8     1.9400
 9   56.0000      1    24.4100
10   56.0000      2    94.9800
11   56.0000      3     3.3400
12   56.0000      4    72.2500
13   56.0000      5    51.9300
14   56.0000      6    17.2300
15   56.0000      7    30.8400
16   56.0000      8     3.6100
17   96.0000      1    13.3600
18   96.0000      2    78.8300
19   96.0000      3     6.7100
20   96.0000      4    79.9300
21   96.0000      5    42.6800
22   96.0000      6     8.7600
23   96.0000      7    48.5700
24   96.0000      8     7.1300
```

typical results obtained were:

```
                              A -92 /01 - PROBLEM - run 1a


          STARTING VALUES OF THE PARAMETERS
                    -.7505600E+01
                    -.5684000E+01
                    -.6502300E+01
                    +.1098600E+01
                    +.0000000E+00


 NUMBER OF EQUATIONS    :  8
 NUMBER OF OBSERVATIONS: 24

 MACHINE PRECISION                                  :0.10E-13
 RELATIVE LOCAL ERROR BOUND FOR INTEGRATION         :0.10E-04
 RELATIVE TOLERANCE FOR RESIDUE                     :0.10E-03
 ABSOLUTE TOLERANCE FOR RESIDUE                     :0.10E-03
 MAXIMUM NUMBER OF INTEGRATIONS TO PERFORM          : 50
 RELATIVE STARTING VALUE OF LAMBDA                  :0.10E+00
 RELATIVE MINIMAL STEPLENGTH                        :0.10E-03


 THERE ARE NO BREAK-POINTS

 THE ALPHA-POINT OF THE F-DISTRIBUTION : 4.17

 *** MONMAR called in MARQUR before exit.
 Singular values and vectors :
            1           2           3           4           5
      0.51486E+02 0.29348E+02 0.14948E+02 0.12693E+02 0.18703E-02
            1           2           3           4           5
```

```
  1  0.95424E+00 -.26423E+00 0.13737E+00 -.27091E-01 -.91869E-05
  2  0.15086E+00 0.21531E+00 -.46738E+00 0.84406E+00 -.34554E-04
  3  0.20443E+00 0.91797E+00 0.32804E+00 -.89062E-01 -.21983E-05
  4  -.11154E+00 -.14349E+00 0.57246E+00 0.37350E+00 -.70694E+00
  5  0.11147E+00 0.14341E+00 -.57216E+00 -.37336E+00 -.70728E+00


NORMAL TERMINATION OF THE PROCESS


LAST INTEGRATION WAS PERFORMED WITHOUT BREAK POINTS


EUCL. NORM OF THE LAST RESIDUAL VECTOR  :0.6974293E+01
EUCL. NORM OF THE FIRST RESIDUAL VECTOR :0.6455052E+02
NUMBER OF INTEGRATIONS PERFORMED        :  7
LAST IMPROVEMENT OF THE EUCLIDEAN NORM  :0.7022356E-04
CONDITION NUMBER OF  J'*J               :0.7577978E+09
LOCAL ERROR BOUND WAS EXCEEDED (MAXIM.) :  0


 PARAMETERS             CONFIDENCE INTERVAL

-.8430087E+01             0.1694125E+00
-.6231400E+01             0.5565611E+00
-.7398311E+01             0.2854291E+00
0.1532019E+01             0.2761478E+04
-.4346094E+00             0.2762815E+04


CORRELATION MATRIX

  I\ J=    1           2           3           4
  2  -.19828E+00
  3  0.15903E-01 -.29902E+00
  4  0.21186E+00 0.24255E+00 0.30116E-01
  5  0.21179E+00 0.24250E+00 0.30052E-01 0.10000E+01


COVARIANCE MATRIX

  I\ J=    1           2           3           4           5
  1  0.13765E-02
  2  -.89669E-03 0.14857E-01
  3  0.36882E-04 -.22783E-02 0.39074E-02
  4  0.47537E+01 0.17879E+02 0.11385E+01 0.36574E+06
  5  0.47545E+01 0.17884E+02 0.11366E+01 0.36592E+06 0.36610E+06
```

```
       THE LAST RESIDUAL VECTOR:
 column 1: OBS(I)
 column 2: Y(c_I)
 column 3: Y(c_I) - OBS(I)
 column 4: sqrt(w_I) * OBS(I)
 column 5: sqrt(w_I) * Y(c_I)
 column 6: sqrt(w_I) * (Y(c_I) - OBS(I))
          1           2           3           4           5           6
 1  0.32360E+02 0.32272E+02 -.88229E-01 0.32360E+02 0.32272E+02 -.88229E-01
 2  0.10918E+04 0.11126E+03 0.20832E+01 0.10918E+03 0.11126E+02 0.20832E+00
 3  0.64600E+01 0.68587E+01 0.39867E+00 0.64600E+01 0.68587E+01 0.39867E+00
 4  0.61180E+03 0.60870E+02 -.31044E+00 0.61180E+02 0.60870E+01 -.31044E-01
 5  0.67580E+02 0.69034E+02 0.14543E+01 0.67580E+02 0.69034E+02 0.14543E+01
 6  0.20350E+02 0.22133E+02 0.17827E+01 0.20350E+02 0.22133E+02 0.17827E+01
 7  0.12070E+02 0.88330E+01 -.32370E+01 0.12070E+02 0.88330E+01 -.32370E+01
 8  0.19400E+01 0.91245E+00 -.10275E+01 0.19400E+01 0.91245E+00 -.10275E+01
 9  0.24410E+02 0.22248E+02 -.21623E+01 0.24410E+02 0.22248E+02 -.21623E+01
10  0.94980E+03 0.93313E+02 -.16672E+01 0.94980E+02 0.93313E+01 -.16672E+00
11  0.33400E+01 0.47042E+01 0.13642E+01 0.33400E+01 0.47042E+01 0.13642E+01
12  0.72250E+03 0.73048E+02 0.79807E+00 0.72250E+02 0.73048E+01 0.79807E-01
13  0.51930E+02 0.52297E+02 0.36670E+00 0.51930E+02 0.52297E+02 0.36670E+00
14  0.17230E+02 0.16361E+02 -.86914E+00 0.17230E+02 0.16361E+02 -.86914E+00
15  0.30840E+02 0.31342E+02 0.50244E+00 0.30840E+02 0.31342E+02 0.50244E+00
16  0.36100E+01 0.42797E+01 0.66968E+00 0.36100E+01 0.42797E+01 0.66968E+00
17  0.13360E+02 0.16340E+02 0.29799E+01 0.13360E+02 0.16340E+02 0.29799E+01
18  0.78830E+03 0.78947E+02 0.11711E+00 0.78830E+02 0.78947E+01 0.11711E-01
19  0.67100E+01 0.37352E+01 -.29748E+01 0.67100E+01 0.37352E+01 -.29748E+01
20  0.79930E+03 0.79925E+02 -.51390E-02 0.79930E+02 0.79925E+01 -.51390E-03
21  0.42680E+02 0.41301E+02 -.13794E+01 0.42680E+02 0.41301E+02 -.13794E+01
22  0.87600E+01 0.88720E+01 0.11197E+00 0.87600E+01 0.88720E+01 0.11197E+00
23  0.48570E+02 0.49827E+02 0.12574E+01 0.48570E+02 0.49827E+02 0.12574E+01
24  0.71300E+01 0.86183E+01 0.14883E+01 0.71300E+01 0.86183E+01 0.14883E+01


--- Performance statistics:
      8 calls of subroutine FUNCT
      7 calls of subroutine FUNCT (where SEC=.false.)
      1 calls of subroutine MARQUR
    827 calls of subroutine DERIV
    405 calls of subroutine JADFDP
    412 calls of subroutine JADFDY
    412 calls of subroutine DGETRF.


    THE CALCULATION IN PEIDE CONSUMED      2.62  SECONDS
```

The results in this case clearly indicate that parameters p_4 and p_5 cannot be determined. However, as the singular values and the 4^{th} and 5^{th} elements of the singular vectors indicate, it should be possible with these data to determine either $p_4 + p_5$ or $p_4 - p_5$. With the current state of the software it is now fairly easy to modify the model as provided to `A_ENGINE` in a file `$PROBLEM.model`. In the

original model we replaced $p_4 = \ln k_4$ and $p_5 = \ln k_5$ by p'_4 and p'_5 such that $p'_4 + p'_5 = \ln k_4$ and $p'_4 - p'_5 = \ln k_5$. Below, we give a summary of the new results, containing the singular values and singular vectors, the new results for the parameters and their confidence intervals, and the covariance matrix.

```
                    A -92 /01 - PROBLEM - run 1a


        STARTING VALUES OF THE PARAMETERS
                  -.7505600E+01
                  -.5684000E+01
                  -.6502300E+01
                  +.5493000E+00
                  +.5493000E+00

Singular values and vectors :
          1           2           3           4           5
    0.52190E+02 0.30129E+02 0.19634E+02 0.13132E+02 0.26460E-02
          1           2           3           4           5
  1 0.93406E+00 -.32116E+00 0.13885E+00 -.71446E-01 -.12997E-04
  2 0.15303E+00 0.20486E+00 -.59125E-01 0.96494E+00 -.48890E-04
  3 0.21257E+00 0.85104E+00 0.44212E+00 -.18730E+00 -.31096E-05
  4 -.79050E-04 -.95985E-04 0.21377E-03 -.46541E-05 -.10000E+01
  5 -.24274E+00 -.36142E+00 0.88417E+00 0.16940E+00 0.24210E-03


EUCL. NORM OF THE LAST RESIDUAL VECTOR  :0.6974294E+01
CONDITION NUMBER OF  J'*J               :0.3890497E+09


 PARAMETERS             CONFIDENCE INTERVAL

-.8430087E+01             0.1694105E+00
-.6231393E+01             0.5566160E+00
-.7398308E+01             0.2854298E+00
0.5481539E+00             0.2761179E+04
0.9833143E+00             0.7568458E+00


COVARIANCE MATRIX

  I\ J=    1           2           3           4           5
  1 0.13765E-02
  2 -.89677E-03 0.14860E-01
  3 0.36879E-04 -.22786E-02 0.39074E-02
  4 0.47524E+01 0.17878E+02 0.11371E+01 0.36566E+06
  5 -.40067E-03 -.24924E-02 0.93378E-03 -.88526E+02 0.27473E-01
```

These results show that beside p_1, p_2, p_3, the new parameter p'_5 could also be determined with some accuracy.

6. Related work

During the project we have been cooperating with CWI's department of Interactive Systems (IS) with the aim of exploiting modern visualisation technology for discovering new ways to solve the described class of problems. The work done in this direction has been described in a paper by Robert van Liere [23] from which we give the following summary.

The purpose of scientific visualisation is to enhance existing scientific and numerical methods by increasing the scientist's ability to see data and comprehend the results of computations. A typical solution process might consist of four phases, a *modeling phase* when a mathematical model is developed describing some physical phenomenon, and a subsequent cycle of three phases, the *simulation*, *data mapping*, and *presentation* phases, respectively. In these three phases a simulation of the physical reality is made given the model and some initialisation. The data from this simulation computation are collected and used for the generation of geometric primitives, and the results are made available for visual presentation in many ways in order to illuminate the characteristics of the numerical solution space. The user can gain much insight by steering the visual presentations when changing perspective and angle, size and highlighting of an image of one set of data as well as by modifying the data in a new cycle of the three phases with different initial values provided to the model.

The paper [23] gives a description of Barnes's problem (section 5.1.1) for which a complete image of the 3-D solution space for the parameter $\mathbf{p}$ has been computed, which can subsequently be inspected interactively in a large variety of ways. By looking at isosurfaces of the residue function $S(\mathbf{p})$ the user can interactively navigate through this space, examine the areas where local minima occur and call for a better resolution of the data set. This visualisation is a convenient tool for identifying parameter ranges that could be of special interest, and this information can immediately be fed back into the simulation. Similarly, the experimenter can also identify areas in the solution space that are unlikely to become of interest for further generation of experimental data.

It is expected that this synergy of two research disciplines provides a category of powerful tools for scientists and engineers in which modeling and simulation are tightly linked with interactive computer graphics.

7. Conclusion

In this report we have described the development and implementation of software for solving parameter identification problems as they are originating in e.g. (chemical) reaction kinetics. These problems consist of a system of ordinary differential equations (ODEs) or differential-algebraic equations (DAEs) containing parameters (such as reaction constants) to be determined when experimental values (measurements) are available that should fit the solution of the system of ODEs.

The implementation was derived from an existing Algol 60 code that was documented in [17]. The present implementation, called `PEIDE`, consists of a set of Fortran subroutines joining advanced numerical mathematics methods for integrating the systems of ODEs and solving the nonlinear minimisation problem. The solver is provided with many options for modifying its behaviour regarding convergence, which allows much space for further experimentation to find a fruitful balance of the available options for solving a class of problems that is as wide as possible.

For testing purposes the performance of the present implementation has been compared with the performance of the older Algol 60 implementation for a set of small testing problems (see [17]) with satisfactory result, as is described in section 5.1.

The real merits of the implementation have been shown by two series of experiments with measurements from practical situations, viz. the Hong problem (results are described in section 5.2.1) and the `A-92` problem (results are described in section 5.2.2). With both problems several solver runs have been carried out while modifying starting values of the parameters, scaling of components of parameters, variables and residues, and other available options.

For the Hong problem, our experiments with the `PEIDE` solver sufficiently demonstrated that the chosen model was unsuitable to fit the data. The solution for the computed parameter values was

never a good approximation of the experimental data, and the additional statistical information clearly indicated the indeterminacy of the obtained parameters.

The computations with the A-92 problem showed that some of the unknown parameters could quite well be determined and that these parameters yield a solution (of the ODEs) that pretty well fits the measurements. Moreover, it was clear from the additional statistical information for the undetermined parameters that some linear combination of these parameters should also be determinable. This linear combination and its value ($p_4 - p_5$ in section 5.2.2) could subsequently be determined after a minor change of the model.

The results described in this report show that the present software package PEIDE together with the accompanying Maple program for transforming mathematically defined models into Fortran subroutines is a powerful tool for solving parameter identification problems. It deserves a deeper investigation of its possibilities and a further development by applying it to a wider class of models for real-life problems.

ACKNOWLEDGEMENTS

The research was partially made possible by a contract with AKZO. We are grateful to Dr R. van der Hout en Ir P. den Decker (AKZO Research Laboratories Arnhem) for their stimulating discussions and for providing the information and the data for the real-life problems.

Further we want to acknowledge Robert van Liere (Department of Interactive Systems, CWI) for his interest in the project and for showing the new possibilities that are available for parameter estimation by computer visualisation.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenny, and D. Sorensen. *Preliminary LAPACK Users' Guide*, 1991.

[2] U. Ascher and L.Petzold. Projected collocation for higher-order higher-index differential-algebraic equations. *J. Comput. Appl. Math.*, 43.1-2:243–259, 1992.

[3] L. T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers and Chem. Engng*, 8:243–248, 1984.

[4] L. T. Biegler. Chemical process simulation. *Chemical Engineering Progress*, pages 50–61, 1989.

[5] L. T. Biegler. Strategies for simultaneous solution and optimization of differential-algebraic systems. In J. J. Siirola, I. E. Grossmann, and G. Stephanopoulos, editors, *Foundations of Computer-aided Process Design*, pages 155–179, 1990.

[6] L. T. Biegler and R. R. Hughes. Process optimization: A comparative study. *Computers and Chem. Engng*, 7:645–661, 1983.

[7] G. E. Blau, R. R. Klimpel, and E. C. Steiner. Nonlinear parameter estimation and model distinguishability of physiochemical models at chemical equilibrium. *Canadian J. of Chem. Engng*, 50:399–409, 1972.

[8] H. G. Bock. Recent advances in parameter identification techniques for O.D.E. In P. Deuflhard and Hairer, editors, *Progress in Scientific Computing*, volume 2, pages 95–121. Birkhäuser, 1983.

[9] H. G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen.* PhD thesis, Rheinische Friedrich-Wilhelms-Universität zu Bonn, 1985.

[10] H. G. Bock, E. Eich, and J. P. Schlöder. Numerical solution of constrained least squares boundary value problems in differential-algebraic equations. Technical Report No. 440, Universität Heidelberg, 1987.

[11] H. G. Bock and J. P. Schlöder. Fit, fitter, the fittest: Methods for modeling and identification of dynamical systems. In Möller, editor, *Systems Analysis of Biological Processes.* Vieweg, 1987.

[12] H. G. Bock and J. P. Schlöder. Recent progress in the development of algorithms and software for large scale parameter estimation problems in chemical reaction systems. Technical Report No. 441, Universität Heidelberg, 1987.

[13] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, and S. M. Watt. *Maple V Language Reference manual.* Springer Verlag, 1991.

[14] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, and S. M. Watt. *Maple V Library Reference manual.* Springer Verlag, 1991.

[15] P. Deuflhard and U. Nowak. Efficient numerical simulation and identification of large chemical reaction systems. *Ber. Bunsenges. Phys. Chem.*, 90:940–946, 1986.

[16] B. van Domselaar. Een mathematische analyse van het hartinfarct (*Dutch*) (A mathematical analysis of the heart infarct). Technical Report NN 4, Mathematical Centre, 1974.

[17] B. van Domselaar and P. W. Hemker. Nonlinear parameter estimation in initial value problems. Technical Report NW 18, Mathematical Centre, 1975.

[18] P. Eykhoff. *System Identification — Parameter and State Estimation.* Wiley, Chichester, 1974.

[19] P. W. Hemker. Numerical methods for differential equations in system simulation and in parameter estimation. In H.C. Hemker and B. Hess, editors, *Analysis and Simulation of biochemical systems*, pages 59–80. North Holland Publ. Comp., 1972.

[20] P. W. Hemker. Parameter estimation in nonlinear differential equations. Technical Report MR 134, Mathematical Centre, Amsterdam, 1972.

[21] K. K. Hong. Parameter estimation in chemical reactions. Master's thesis, Technical University Eindhoven, 1990.

[22] Leslie Lamport. LaTeX: *A document preparation system.* Addison-Wesley, 1986.

[23] R. van Liere. Computational steering: a case study. *CWI Quarterly*, 5.3:207–217, 1992.

[24] Th. Lohmann, H. G. Bock, and J. P. Schlöder. Numerical methods for parameter estimation and optimal experiment design in chemical reaction systems. Technical Report No. 283, Universität Augsburg, 1987.

[25] S. Macchietto. Solution techniques for processes described by mixed sets of equations and procedures, 1985.

[26] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *J. SIAM*, 11:431–441, 1963.

[27] J. Molenaar. Estimation of kinetical parameters in chemical initial value problems. Technical Report IWDE report 90–08, Technical University Eindhoven, 1990.

[28] U. Nowak and P. Deuflhard. Numerical identification of selected rate constants in large chemical reaction systems. *Applied Numerical mathematics*, 1:59–75, 1985.

[29] J. L. Robertson. The ideal process simulator. *Chemical Engineering Progress*, pages 62–64, 1989.

[30] Johannes P. Schlöder. *Numerische Methoden zur Behandlung hochdimensionaler Aufgaben der Parameteridentifizierung.* PhD thesis, Rheinische Friedrich-Wilhelms-Universität zu Bonn, 1985.

[31] F. C. Schweppe. *Uncertain Dynamic Systems.* Prentice-Hall, London, 1973.

[32] M. Shacham, S. Macchietto, L. F. Stutzman, and P. Babcock. Equation oriented approach to process flowsheeting. *Computers and Chem. Engng*, 6:79–95, 1982.

[33] S. Vajda, P. Valkó, and A. Yermakova. A direct-indirect procedure for estimation of kinetic parameters. *Computers and Chem. Engng*, 10:49–58, 1986.

[34] S. Vasantharajan and L. T. Biegler. Simultaneous strategies for optimization of differential algebraic systems with enforcement of error criteria. *Computers and Chem. Engng*, 10:1083–1100, 1990.

[35] Jack Williams. Approximation and parameter estimation in ordinary differential equations. Technical Report No.171, University of Manchester, 1988.