PARAMETER ESTIMATION IN ONE-DIMENSIONAL TIME-DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS

K. SCHITTKOWSKI*

Department of Mathematics, University of Bayreuth, 95440 Bayreuth, Germany

...

We consider an approach to determine parameters in a system of one-dimensional time-dependent parabolic differential equations and coupled ordinary differential equations. The model allows transmission conditions between separate integration areas for functions and derivatives. Proceeding from given experimental data, e.g. observation times and measurements, the minimum least squares distance of the measured data from the solution of the dynsmical system at designated space values is to be computed. The method of lines is used to discretize the partial differential equation with respect to polynomials of arbitrary odd order, and to transform the original system into a sequence of ordinary differential equations, that can be solved then by any available ODE-solver. Numerical test results are included to show the efficiency of different ODE solvers and optimization routines based on a collection of 20 test models.

KEY WORDS: least squares optimization, nonlinear programming, data fitting, partial differential equations, line method

1 INTRODUCTION

Parameter estimation plays an important role in many natural science and other disciplines. The key idea is to estimate unknown parameters in a mathematical model that describes the real life situation, by minimizing the distance of some known experimental data from the theoretical model data. Thus also model parameters that cannot be measured directly can be estimated by a least squares fit and analysed subsequently.

The purpose of this paper is to describe an algorithmic approach to fit data in a dynamical system. The model functions define one-dimensional time-dependent partial differential equations e.g. of parabolic type. In addition there may be coupled ordinary differential equations with initial values. The model allows arbitrary transition conditions between separate integration areas for functions and derivatives. Thus also non-continuous or non-smooth transitions can be taken into account. Moreover there may be any nonlinear equality and inequality constraints

* The paper was supported by the BMBF-Verbundprojekt *Modellierung, Simulation und Steuerung transdermaler Systeme* under contract number 03-SC7BAY-0

for the parameters to be estimated. In particular upper and lower bounds for the parameters must be defined. The solution of the differential equation is inserted into certain fitting criteria, yielding a least squares objective function of a mathematical optimization problem that is to be solved numerically.

It is assumed that the model functions are differentiable with respect to the parameters to be estimated. The unknown parameters to be estimated, are hidden in right-hand side of the system equations, i.e. the partial or ordinary differential equations, the intial values, the boundary or transition conditions, the constraints, or the fitting criteria.

Only for illustration purposes we denote the independent model variable the *time* variable of the system and the dependend data as *measurement* values of an *experiment*. By this the probably most frequent application is described. On the other hand, these terms may have any other meaning within a model depending on the underlying application problem.

The mathematical model is described in Section 2 in more detail. The least squares formulation is outlined and the mathematical structure of the partial differential equation is described.

The partial differential equation is discretized by the method of lines, see e.g. Schiesser [30], leading to a system of ordinary differential equations. Thus all techniques we know from parameter identification in ordinary differential equations, can be transferred to the solution of one-dimensional partial equations. The discretization procedure is outlined in Section 3.

For the integration of the resulting ordinary differential equation, several different integration routines published by Hairer and Wanner[17] have been taken into account. Since the discretized systems have a tendancy to become stiff, the usage of implicit methods is highly recommended. A couple of alternative numerical least squares optimization methods have been integrated into the system. They are based on the Gauss-Newton method, but differ in algorithmic details. A brief outline of their underlying strategies is also found in Section 3.

Some numerical test results to illustrate the performance of the numerical methods chosen, are presented in Section 4. For a simple heat equation the influence of the differential equation solver is investigated. Then we show some comparative results obtained by 20 different models w.r.t. the different optimization routines implemented. Many of the test cases possess a practical background, and come with *real life* experimental data. In some other cases measurement values are generated artificially to investigate also the question, whether a known solution can be identified exactly. The corresponding partial differential equations are outlined in an appendix.

2 THE MATHEMATICAL MODEL

2.1 Least Squares Formulation

The basic mathematical model is a constrained nonlinear least squares problem, i.e. the minimization of a sum of squares of nonlinear functions of the form

$$\sum_{i=1}^{l} f_i(p)^2 \quad , \tag{1}$$

where $p \in I\!R^n$ has to satisfy constraints of the form

$$\begin{array}{lll} g_j(p) = 0 & , & j = 1, \ldots, m_e \quad , \\ g_j(p) \geq 0 & , & j = m_e + 1, \ldots, m \quad , \\ p_l \leq p \leq p_u & & . \end{array} \tag{2}$$

Here we assume that the parameter vector p is n-dimensional and that all nonlinear functions are continuously differentiable with respect to p.

All least squares parameter estimation algorithms proceed from the above formulation, although in the one or other case different approaches are available to define the objective functions and constraints within a FORTRAN subroutine. The assumption, that all problem functions must be smooth, is required, since the more efficient numerical algorithms in PDEFIT are based more or less on the Gauss-Newton method, that requires at least first derivatives.

In the following we restrict all investigations to parameter estimation problems where one vector-valued model function is available to compute the fitting criteria with respect to an additional variable called *time*. We proceed now from r measurement sets given in the form

$$(t_i, y_i^k) \quad , \quad i = 1, \ldots, s \; , \quad k = 1, \ldots, r \quad , \tag{3}$$

where s time values and rs corresponding measurement values are defined. Together with a vector-valued model function

$$h(p, t) = (h_1(p, t), \ldots, h_r(p, t))^T, \tag{4}$$

we get the above least squares formulation from

$$f_j(p) = h_k(p, t_i) - y_i^k,$$

where j runs from 1 to rs in any order. Then the underlying idea is to minimize the distance between the model function at certain time points and the corresponding measurement values. The individual distances are denoted the residuals of the problem. In the ideal case the residuals are zero indicating a perfect fit of the model function to the measurements.

Since for practical applications additional weighting factors are highly useful, we use instead the notation

$$f_j(p) = w_i^k (h_k(p, t_i) - y_i^k) \tag{5}$$

with suitable non-negative weighting factors w_i^k.

Thus the resulting least squares function to be minimized is the sum of the particular $f_j(p)$ for $j = 1, \ldots, rs$, i.e.

$$f(p) = \sum_{k=1}^{r} \sum_{i=1}^{s} (w_i^k (h_k(p, t_i) - y_i^k))^2 \tag{6}$$

2.2 The Partial Differential Equation

The next step is to outline, how the model function $h(p, t)$ depends on the solution of a system of ordinary differential equations obtained by discretizing a partial differential equation.

First we assume without loss of generality, that the initial time is 0. This assumption facilitates the description of the mathematical model and is easily satisfied in practice by a suitable linear transformation of the time variable.

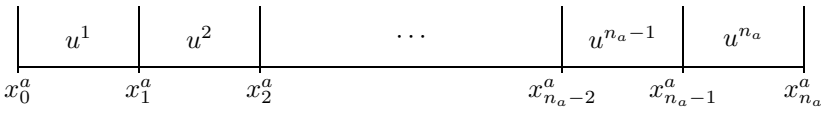
The model we want to investigate now, is defined by a system of n_p one-dimensional partial differential equations in one or more spatial intervals. A similar model was investigated by Lang[23]. These intervals could describe e.g. certain areas with different diffusion coefficients. They are given by the outer boundary values x_{L} and x_{R} that define the total integration interval w.r.t. the space variable x, and optionally some additional internal transition points $x_1^a, \ldots, x_{n_a-1}^a$. Thus we get a sequence of $n_a + 1$ boundary and transition points

$$x_0^a := x_{\text{L}} < x_1^a < \ldots < x_{n_a-1}^a < x_{n_a}^a := x_{\text{R}} \; . \tag{7}$$

For each integration interval, we have to define a partial differential equation of the form

$$u_t^i = f^i(x, t, v, u^i, u_x^i, u_{xx}^i, p) \;\; , \quad i = 1, \ldots n_a \; , \tag{8}$$

where $x \in I\!R$ is the spatial variable with $x_{i-1}^a \leq x \leq x_i^a$ for $i = 1, \ldots, n_a$, $t \in I\!R$ the time variable with $0 < t \leq t_s$, $v \in I\!R^{n_o}$ the solution vector of an optional coupled system of ordinary differential equations, $u^i \in I\!R^{n_p}$ the system variable we want to compute, and $p \in I\!R^n$ the parameter vector to be identified by an outer least squares algorithm. t_s is the final integration time, in general the last experimental time value.



Any solution of the coupled system depends on the spatial variable x, the time variable t, the parameter vector p, and is therefore written in the form $v(t, p)$ and $u^i(x, t, p)$ for $i = 1, \ldots, n_a$.

For any of both end points x_{L} and x_{R} we allow boundary conditions of the form

$$\begin{aligned} u^1(x_{\mathrm{L}},t,p) &= u^L(t,p) \\ u^{n_a}(x_{\mathrm{R}},t,p) &= u^R(t,p) \\ u_x^1(x_{\mathrm{L}},t,p) &= \hat{u}^L(t,u^1(x_L,t,p),p) \\ u_x^{n_a}(x_{\mathrm{R}},t,p) &= \hat{u}^R(t,u^{n_a}(x_R,t,p),p) \end{aligned} \tag{9}$$

for $0 < t \leq t_s$ with given functions $u^L(t,p)$, $u^R(t,p)$, $\hat{u}^L(t,u,p)$ and $\hat{u}^R(t,u,p)$. It is essential to understand, that we do not require the evaluation of all boundary functions. The user has to define an appropriate combination of boundary conditions, so that the resulting PDE is well-defined and uniquely solvable.

Transmission conditions between the different areas may be defined in addition. They are allowed at most at transition points and have the form

$$\begin{aligned} u^i(x_i^a,t,p) &= c_i^R(t,u^{i+1}(x_i^a,t,p),p) \\ u^{i+1}(x_i^a,t,p) &= c_i^L(t,u^i(x_i^a,t,p),p) \\ u_x^i(x_i^a,t,p) &= \hat{c}_i^R(t,u^{i+1}(x_i^a,t,p),u_x^{i+1}(x_i^a,t,p),p) \\ u_x^{i+1}(x_i^a,t,p) &= \hat{c}_i^L(t,u^i(x_i^a,t,p),u_x^i(x_i^a,t,p),p) \end{aligned} \tag{10}$$

with $0 < t \leq t_s$, $i = 1,\ldots,n_a-1$ and given functions on the right-hand side. Again the user may omit any of these functions, if a transition condition does not exist at a given x_i^a-value. To avoid internal access conflicts, either u^i or u^{i+1} may be defined, also either u_x^i or u_x^{i+1}, but never both at the same transition point.

Since the starting time is assumed to be zero, initial conditions must have the form

$$u^i(x,0,p) = u_0^i(x,p) \quad , \quad i = 1,\ldots,n_a \tag{11}$$

and are defined for all $x \in \left[x_{i-1}^a, x_i^a\right]$, $i = 1,\ldots,n_a$. They may depend on the parameters to be estimated.

If ordinary differential equations are to be coupled to the partial differential equations, we proceed from an additional ODE system of the form

$$\dot{v}_j = F_j(t,v,u^{i_j}(x,t,p),u_x^{i_j}(x,t,p),u_{xx}^{i_j}(x,t,p),p) \tag{12}$$

for $j = 1,\ldots,n_o$ with initial values

$$v(0,p) = v_0(p) \tag{13}$$

that may depend again on the parameters to be estimated. The system has n_o components, i.e. $v = (v_1,\ldots,v_{n_o})^T$. Coupling of ordinary differential equations is allowed at any spatial point of the discrete system. The right-hand side of the coupling equation may depend on the corresponding solution of the partial equation and its first and second derivative w.r.t. the space variable at the point under consideration.

To indicate that the fitting criteria $h_k(p,t)$ depend also on the solution of the differential equation at any spatial points, where k denotes the index of a measurement

set, we use the notation

$$h_k(p,t) = \hat{h}_k(p,t,v(t,p),u^i(x,t,p),u^i_x(x,t,p),u^i_{xx}(x,t,p)) \tag{14}$$

and insert $\hat{h}_k$ instead of h_k into the least squares function(6). Again the fitting criteria may depend on solution values at any spatial point w.r.t. an integration interval defined by the index i and r denotes the number of measurement sets. The fitting criterion may depend on the solution of the partial equation and its first and second derivative w.r.t. the space variable at the actual spatial point.

In order to achieve smooth fitting criteria, we assume that all model functions depend continuously differentiable on the parameter vector p. Moreover we assume that the discretized system of differential equations is uniquely solvable for all p with $p_l \leq p \leq p_u$.

3 NUMERICAL IMPLEMENTATION

3.1 The Discretization Scheme

The underlying idea is to transform the partial differential into a system of ordinary differential equations by discretizing the model functions w.r.t. the spatial variable x. This approach is known as the method of lines, see e.g. Schiesser[30].

For the i-th integration interval of the spatial variable, we denote the number of discretization points by n_i, $i = 1,\ldots,n_a$. We proceed from uniform grid points within each interval and get a discretization of the whole space interval from x_{L} to x_{R}.

Now we consider a single interval (x^a_{i-1}, x^a_i). The internal equidistant discretization points possess the form

$$x^i_j := x^a_{i-1} + j \cdot \frac{x^a_i - x^a_{i-1}}{n_i + 1} \quad , \tag{15}$$

for $j = 1,\ldots,n_i$. To approximate the first and second partial derivative of $u(x,t,p)$ w.r.t. the spatial variable at a given point x^i_j, we compute a polynomial interpolation subject to the neighbouring values $u(x^i_{j_1},t,p),\ldots,u(x^i_j,t,p),\ldots,u(x^i_{j_2},t,p)$. Whenever possible, $j_1 < j$ and $j_2 > j$ are chosen, such that j remains in the center. But when approaching the boundary of the integration interval, one has to shift the interpolating interval in a straightforward way.

The number of interpolation points, i.e. $j_2 - j_1 + 1$, depends on the desired polynomial degree as given by the user. The polynomials are computed by Newton's interpolation formula. The order is always even, since only an odd numbers of interpolation points is allowed. We made this restriction to get uniformly distributed grid points.

Whenever a boundary condition in Dirichlet form

$$\begin{aligned} u_k^1(x_{\mathrm{L}},t,p) &= u_k^L(t,p) \\ u_k^{n_a}(x_{\mathrm{R}},t,p) &= u_k^R(t,p) \\ u_k^i(x_i^a,t,p) &= c_{i,k}^R(t,u^{i+1}(x_i^a,t,p),p) \\ u_k^i(x_{i-1}^a,t,p) &= c_{i-1,k}^L(t,u^{i-1}(x_{i-1}^a,t,p),p) \end{aligned} \tag{16}$$

is given for some $1 \leq k \leq n_p$, see (9) or (10), respectively, then we know the value of the boundary function and use it to interpolate the function $u(x,t,p)$ as described above. In other words the corresponding function value in the right-hand side of the discretized system is replaced by the value given.

Alternatively a boundary condition may appear in the form

$$\begin{aligned} u_{k,x}^1(x_{\mathrm{L}},t,p) &= \hat{u}_k^L(t,u^1(x_L,t,p),p) \\ u_{k,x}^{n_a}(x_{\mathrm{R}},t,p) &= \hat{u}_k^R(t,u^{n_a}(x_R,t,p),p) \\ u_{k,x}^i(x_i^a,t,p) &= \hat{c}_{i,k}^R(t,u^{i+1}(x_i^a,t,p),u_x^{i+1}(x_i^a,t,p),p) \\ u_{k,x}^i(x_{i-1}^a,t,p) &= \hat{c}_{i-1,k}^L(t,u^{i-1}(x_{i-1}^a,t,p),u_x^{i-1}(x_{i-1}^a,t,p),p) \end{aligned} \tag{17}$$

for some $1 \leq k \leq n_p$, see (9) and (10). In this case the interpolating polynomial at the boundary is computed by Hermite formula to exploit knowledge about the known spatial derivative. Coupled ordinary differential equations are added to the discretized system without any further modification.

When defining the transmission conditions, it is important to have the underlying flux direction in mind. If, for example, the flux is in the direction of the spatial variable and we want to define a continous transition at x_i^a, then we have to formulate the corresponding transition function in the form $u_k^{i+1}(x_i^a,t,p) = u_k^i(x_i^a,t,p)$ in order to guarantee, that the boundary values w.r.t. x are spread over the interval.

Since we use a uniform grid on each interval, it might be necessary to define artificial transition points in order to get a grid refinement in a certain spatial area. Smooth transitions are achieved in these cases, if function values and spatial derivative coincide, e.g. $u_k^{i+1}(x_i^a,t,p) = u_k^i(x_i^a,t,p)$ and $u_{k,x}^i(x_i^a,t,p) = u_{k,x}^{i+1}(x_i^a,t,p)$ in case of a flux from left to right.

3.2 Numerical Algorithms

Due to the practical importance of parameter estimation, very many numerical codes were developed in the past and are available in form of software tools for minimizing least squares functions. To solve parameter estimation in a flexible way, four different optimization algorithms are taken into account for our implementation:

1. DFNLP: By transforming the original problem into a general nonlinear programming problem in a special way, typical features of a Gauss-Newton and quasi-Newton method are retained, see Schittkowski[33]. The resulting optimization

problem is solved by a standard sequential quadratic programming code called NLPQL, cf. Schittkowski[32].

2. DN2GB: The subroutine is a frequently used unconstrained least squares algorithm and was developed by Dennis, Gay and Welsh[6] and is based also on a Gauss-Newton and quasi-Newton approach.
3. NLSNIP: The code is a special purpose implementation for solving constrained nonlinear least squares problems by a combination of Gauss-Newton, Newton and quasi-Newton techniques, cf. Lindström[24].
4. DSLMDF: The algorithm proceeds from successive line searches along the unit vectors by comparing function values only. After a search cycle the Gauss-Newton-type method DFNLP is executed with a limited number of iterations, see Nickel[27].

DFNLP and NLSNIP are capable to take additional linear or nonlinear constraints into account in form of equality or inequality restrictions. The codes DN2GB and DSLMDF allow at least the definition of upper and lower bounds for the variables.

The method of lines is used to discretize the partial differential equation with respect to polynomials of arbitrary odd order, and to transform the original system into a system of ordinary differential equations, that can be solved then by any available ODE-solver. In the present case, it is possible to select a suitable algorithm out of six different solvers:

1. DOPRI5: Explicit Runge-Kutta-method of Dormand and Prince[9] with order 4/5 and step control, see Hairer, Nørsett and Wanner[16]
2. DOP853: Explicit Runge-Kutta-method of Dormand and Prince[9] with order 8 and step control, see Hairer, Nørsett and Wanner[16]
3. ODEX: Extrapolation method based on GBS algorithm with variable order and step size, see Hairer, Nørsett and Wanner[16]
4. RADAU5: Implicit Runge-Kutta-method of Radau-type with order 5, see Hairer and Wanner[17]
5. SDIRK4: Diagonally implicit Runge-Kutta-method, see Hairer and Wanner[17]
6. SEULEX: Extrapolation method based on implicit Euler method with variable order and step size control, see Hairer and Wanner[17]

The implicit methods RADAU5, SDIRK4 and SEULEX are capable to solve stiff problems. Gradients of fitting functions with respect to the parameters are approximated numerically.

The usage of the resulting computer program with name PDEFIT is outlined in Dobmann and Schittkowski[8]. Basically PDEFIT is a FORTRAN subroutine, that can be inserted into any other program, where the model functions must be defined by a separate subroutine. To facilitate the usage, also a main program is available, where the data, e.g. measurements, solution parameters, initial values, are read in from a file. In the present version, gradients are evaluated numerically by the forward difference formula.

4 NUMERICAL TESTS

4.1 Test Environment

The numerical algorithm to estimate parameters in systems of partial differential equations, was implemented in form of a FORTRAN code with name PDEFIT, see Dobmann and Schittkowski[8]. Moreover an interactive user interface in form of a database (MS-Access) running under Windows 3.11 and Windows 95 is available, see Schittkowski[34] for details. The system is called EASY-FIT and allows also parameter identification in explicit model functions, dynamical systems of nonlinear equations, Laplace equations, ordinary differential equations and differential algebraic equations. A particular advantage of the system is, that model functions may be defined either in form of FORTRAN code or in form of a FORTRAN-similar modelling language called PCOMP, that allows in addition automatic differentiation of nonlinear functions, see Dobmann, Liepelt and Schittkowski[7].

All test runs have been performed under the EASY-FIT system on a PC running under Windows 95 (Pentium processor with 120 MHz). The FORTRAN code was compiled and linked by the Salford FTN77 compiler yielding a 32 bit executable code. Nonlinear model functions are defined in the PCOMP syntax and are interpreted during run time.

The purpose of the numerical tests is to show that the proposed discretization approach leads to an efficient solution method for the class of problems under consideration. Moreover we want to compare parameter estimation methods for some special *real life* and academic models.

The optimization routines are executed always with the same initial parameter set although we know, that in the one or other case these tolerances can be adapted to special situations leading to better individual results. For DFNLP we use the termination tolerance 1.0E-7, for DN2GB we define 1.0E-5 for the relative function and variable convergence. NLSNIP is executed with a tolerance of 1.0E-5 for the main relative termination tolerances EPSREL, EPSX and EPSH, and 1.0E-3 for the absolute stopping value EPSABS. The total number if iterations is bounded by 100 for all three algorithms. The code DSLMDF is not allowed to perform more that 20 outer iteration cycles with a termination accuracy of 1.0E-5 for the local search step performed by DFNLP. The search algorithm needs at most 5 function evaluation for each line search with reduction factor of 2.0 and an initial steplength of 0.2.

In some cases, we consider test examples used in the literature for simulation purposes only. In these cases we define any appropriate constants to become the optimization parameters to be estimated, and generate measurements *by hand*. Starting from the constants given, we compute model function values w.r.t. a series of predetermined time values, and round these numbers to two correct digits. Then we define some arbitrary initial values and try to recalculate the known parameters.

4.2 Testing the Influence of the ODE-Solver

We start with a very simple example, see e.g. Schiesser[30]. Consider *Fourier's first law* for heat conduction

$$u_t = Du_{xx}$$

for $0 < t \leq 0.5$ and $0 < x < 1$ with boundary conditions

$$u(0,t) = u(1,t) = 0$$

for $0 \leq t \leq 0.5$ and initial condition

$$u(x,0) = \sin\left(\frac{\pi x}{L}\right)$$

for $0 < x < 1$. The partial differential equation is discretized w.r.t. 10 equidistant intervals. Parameters to be estimated, are D and L, and we use $u(0.5,t)$ as fitting criterion. Measurements are simulated w.r.t. $D = 1$ and $L = 1$ at time coordinates 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5. Starting values are $D = 0.01$ and $L = 2$.

This academic example can be solved quite easily by all parameter estimation algorithms and all integration routines that have been implemented in PDEFIT. Thus we use the example to compare the efficiency of all possible combinations of available ODE-solvers. For the test runs, we use ACCREL=ACCABS=1.0E-6 for the relative and absolute termination criteria and HINIT=1.0E-4 for the initial stepsize of the integration routines executed. Numerical gradients are computed w.r.t. the stepsize 1.0E-5 by forward differences.

In all cases, the computed optimal solution is reevaluated correctly subject to 4 correct digits and the residual is in the order of 0.1E-6. Table 1 contains the following data:

method	-	parameter estimation method
solver	-	solution method for discretized system of ordinary differential equations
n_f	-	number of objective function evaluations
n_g	-	number of gradient evaluations, i.e number of outer iterations
n_{pde}	-	Total number of all right-hand side evaluations of dicretized ODE-system

The implicit ODE-solver RADAU5 is the fastest one w.r.t. number of function evaluations and calculation time. The discretized ordinary differential equation system is quite stable, since also the explicit solvers DOPRI5, DOP853 and ODEX are applicable and work quite well. However they require more function calls than the implicit methods. Also some of the parameter estimation codes, e.g. DN2BG, are more sensitive w.r.t. the accuracy, by which the subproblem is solved, than others. Especially the combined search algorithm DSLMDF is quite robust subject to the accuracy in function and gradient values.

method	*solver*	n_f	n_g	n_{pde}
DFNLPD	RADAU5	10	8	5,397
	SDIRK4	10	8	9,949
	SEULEX	10	8	8,365
	DOPRI5	10	8	11,888
	DOP853	11	9	17,971
	ODEX	11	9	26,106
DN2GB	RADAU5	9	8	5,307
	SDIRK4	35	11	19,304
	SEULEX	11	10	10,186
	DOPRI5	12	11	16,410
	DOP853	13	11	23,986
	ODEX	44	14	80,404
NLSNIP	RADAU5	8	7	4,847
	SDIRK4	8	7	8,851
	SEULEX	8	7	7,390
	DOPRI5	8	7	10,276
	DOP853	8	7	14,344
	ODEX	9	8	27,692
DSLMDF	RADAU5	33	3	7,722
	SDIRK4	33	3	14,160
	SEULEX	33	3	11,603
	DOPRI5	33	3	15,450
	DOP853	33	3	21,488
	ODEX	33	3	30,885

TABLE 1: Numerical results for heat equation

name	n_{par}	n_{pde}	n_{area}	n_{ode}	n_{disc}	n_{meas}	n_{exp}	*scale*	*data*
HEAT	2	1	1	0	11	1	11	0	S
RICHARDS	5	1	1	0	11	1	20	1	E
G_WATER	5	2	1	0	21	1	82	1	E
SKIN_2A	10	2	2	4	11/11	4	19	1	E
SKIN_3A	10	3	3	4	7/5/7	4	19	1	E
KINETIC	10	1	3	6	7/5/7	3	15	1	E
NAG_1	3	2	1	0	11	2	38	0	S
NAG_2	2	1	2	0	11/11	1	23	0	S
NITROGEN	3	2	1	0	11	2	20	0	S
CONTAMIN	3	4	1	4	11	4	24	0	E
MILL	5	1	1	0	11	1	40	1	S
DISRET	2	2	1	0	11	2	38	0	S
SHEAR	4	3	1	0	21	3	33	0	S
EL_DYN	3	2	1	0	11	2	20	0	S
HOT_SPOT	3	1	1	0	21	1	11	0	S
WAVES	3	2	1	0	29	10	50	0	S
FLAME	2	2	1	0	11	5	15	0	S
BRUSSEL	3	2	1	0	21	12	120	0	S
POLLUTN	8	4	1	0	11	4	28	1	S
INTEG	3	1	1	0	21	5	25	1	S

TABLE 2: Test examples (review)

4.3 Testing the Efficiency of Parameter Estimation Algorithms

We select a number of test cases, where most models possess a practical background and where some of them possess *real life* experimental data. Some basic features are summarized in Table 2 using the notation

- $name$ - test problem name as used in subsequent tables
- n_{par} - number of parameters to be estimated
- n_{pde} - number of PDE's
- n_{area} - number of integration areas
- n_{ode} - number of coupled ODE's
- n_{disc} - number of discretization points
- n_{meas} - number of measurement sets
- n_{exp} - total number of experimental data
- $scale$ - scaling procedure (0 - none, 1 - division by sum of squared measurement values)
- $data$ - type of experimental data (S - simulation, E - experiment)

The intention is to compare the performance of optimization codes. Thus we integrate the differential equations always with the same solver RADAU5 and the same stopping tolerances ACCREL=ACCABS=1.0E-6 and HINIT=1.0E-4. Numerical gradients are computed w.r.t. the stepsize 1.0E-5 by forward differences.

Therefore the numerical results give an impression, how a parameter estimation problem based on a one-dimensional PDE can be solved by the very first approach.

Table 3 contains the numerical results achieved, where we use the following notation:

name	-	test problem name
code	-	name of least squares algorithm used
tr	-	termination reason 0 : stopping conditions satisfied 1 : maximum number of iterations reached 2 : error message of least squares algorithm 3 : error message of ODE solver
residual	-	final residual value, i.e. sum of squared deviations (scaled in some cases)
n_f	-	number of function evaluations
n_g	-	number of gradient evaluations or number of iterations, respectively
t_{calc}	-	calculation time in *min*

Note that the calculation time could be reduced significantly by using FORTRAN code for the nonlinear model functions. But in this case we prefer the interpretation of the model functions in the PCOMP language, see Section 4.1.

In some cases the algorithms compute different answers. Especially for problems with *real life* data, the final convergence speed is sometimes low probably because of large errors in the measurements. Either an existing local minimizer is approximated or the stopping tolerance is not sufficiently small in these cases. More details are found in the appendix.

To summarize the results, see Table 4, we present the number of test cases where an algorithm got the least residual subject to two significant digits. Moreover we show the number of runs where an algorithm performed best subject to number of function and gradient evaluations and subject to calculation time. Here we compare only those algorithms that achieved the lowest residual w.r.t. one significant digit.

Although the number of test examples if by far too low to get statistically relevant results, we get the impression that the codes DN2GB of Dennis, Gay and Welsh[6] and NLSNIP of Lindström[24] behave best w.r.t. reliability and efficiency. However none of the four codes tested was able to solve all problems w.r.t. the required accuracy. Thus it must be expected that the practical solution of a parameter estimation problem based on a partial differential equation, consists of a stepwise adaption of the algorithm, model and data, until a solution is obtained. Surprisingly the heuristic combined search method as implemented in the code DSLMDF, works quite satisfactorily.

5 CONCLUSION

We combine well-known algorithms for least squares optimization and ordinary differential equations to estimate parameters in partial differential equations of

name	*code*	*tr*	*residual*	n_f	n_g	t_{calc}
HEAT	DFNLP	0	0.21E-6	10	8	0:33
	DN2GB	0	0.16E-6	9	8	0:22
	NLSNIP	0	0.16E-6	8	7	0:18
	DSLMDF	0	0.16E-6	33	3	0:18
RICHARDS	DFNLP	0	0.0058	64 28	6	9:27
	DN2GB	3	-	-	-	-
	NLSNIP	3	-	-	-	-
	DSLMDF	3	-	-	-	-
G_WATER	DFNLP	0	0.014	21	4	3:20
	DN2GB	0	0.017	60	22	13:19
	NLSNIP	0	0.015	22	4	3:33
	DSLMDF	1	0.013	3,468	291	418:20
SKIN_2A	DFNLP	0	0.0015	43	34	13:01
	DN2GB	1	0.00079	181	101	50:20
	NLSNIP	0	0.0015	44	19	8:10
	DSLMDF	0	0.0013	374	22	22:10
SKIN_3A	DFNLP	0	0.0031	23	21	8:00
	DN2GB	1	0.0049	176	101	59:30
	NLSNIP	0	0.0055	23	9	4:16
	DSLMDF	0	0.0031	190	4	7:30
KINETIC	DFNLP	2	-	-	-	-
	DN2GB	0	0.039	18	13	17:30
	NLSNIP	3	-	-	-	-
	DSLMDF	0	0.063	340	19	26:10
NAG_1	DFNLP	0	0.14E-5	13	11	3:42
	DN2GB	0	0.14E-5	8	7	2:26
	NLSNIP	0	0.14E-5	8	7	2:22
	DSLMDF	0	0.15E-5	73	6	6:27
NAG_2	DFNLP	0	0.97E-4	39	35	6:57
	DN2GB	0	0.97E-4	33	25	4:50
	NLSNIP	0	0.97E-4	11	11	2:00
	DSLMDF	0	0.84E-3	51	4	2:31
NITROGEN	DFNLP	0	0.11E-3	42	40	9:26
	DN2GB	0	0.41E-5	56	23	8:44
	NLSNIP	0	0.41E-5	19	19	8:55
	DSLMDF	0	0.15E-2	67	1	4:12
CONTAMIN	DFNLP	0	0.0045	68	50	7:70
	DN2GB	0	0.0045	9	8	1:12
	NLSNIP	0	0.0045	21	15	1:39
	DSLMDF	0	0.0046	85	3	2:35

TABLE 3: Comparative results (continued)

name	*code*	*tr*	*residual*	n_f	n_g	t_{calc}
MILL	DFNLP	0	0.14E-6	5	3	1:57
	DN2GB	0	0.14E-6	30	11	7:22
	NLSNIP	0	0.14E-6	19	6	4:26
	DSLMDF	0	0.12E-5	60	1	6:42
DISRET	DFNLP	0	0.00021	42	29	8:51
	DN2GB	0	0.00021	20	13	2:10
	NLSNIP	0	0.00021	16	10	2:14
	DSLMDF	2	-	-	-	-
SHEAR	DFNLP	0	0.13	25	8	40:00
	DN2GB	0	0.052	41	14	63:50
	NLSNIP	0	0.045	41	15	54:40
	DSLMDF	0	0.042	127	20	183:10
EL_DYN	DFNLP	0	0.85E-6	21	19	5:18
	DN2GB	0	0.82E-6	12	10	2:59
	NLSNIP	0	0.82E-6	26	12	3:46
	DSLMDF	0	0.83E-6	75	3	4:18
HOT_SPOT	DFNLP	0	0.57E-4	83	60	18:20
	DN2GB	0	0.58E-4	76	52	13:57
	NLSNIP	0	0.58E-4	101	55	15:91
	DSLMDF	0	0.58E-4	311	33	24:40
WAVES	DFNLP	0	0.35E-4	74	52	24:50
	DN2GB	0	0.35E-4	19	15	8:16
	NLSNIP	3	-	-	-	-
	DSLMDF	0	0.35E-4	45	5	9:19
FLAME	DFNLP	0	0.68E-6	16	12	7:15
	DN2GB	0	0.66E-6	10	8	4:23
	NLSNIP	0	0.66E-6	7	7	3:44
	DSLMDF	0	0.20E-5	23	4	4:54
BRUSSEL	DFNLP	0	0.52E-3	40	31	220:00
	DN2GB	0	0.52E-3	19	16	79:10
	NLSNIP	0	0.52E-3	26	20	124:10
	DSLMDF	0	0.52E-3	125	10	146:40
POLLUTN	DFNLP	0	0.60E-5	93	65	246:40
	DN2GB	0	0.50E-5	46	21	56:30
	NLSNIP	0	0.55E-5	31	18	44:50
	DSLMDF	0	0.45E-4	324	8	137:10
INTEG	DFNLP	0	0.46E-5	79	50	9:26
	DN2GB	0	0.45E-5	15	14	1:19
	NLSNIP	0	0.45E-5	67	45	11:38
	DSLMDF	0	0.45E-5	116	20	6:56

TABLE 3: Comparative results

code	$residual^{best}$	n_f^{best}	n_g^{best}	t_{calc}^{best}
DFNLP	11	4	2	3
DN2GB	15	9	3	10
NLSNIP	11	8	7	6
DSLMDF	6	0	9	2

TABLE 4: Summary of performance results

parabolic type. The one-dimensional system is transformed into a sequence of ordinary differential equations by the method of lines. The model allows coupled ordinary differential equations and transmission conditions between different areas. The numerical implementation is outlined and test results are presented based on a collection of test examples. Some of them are obtained by real life models where experimental data are available.

REFERENCES

1. S. Adjerid, J.E. Flaherty. *A moving finite element method with error estimation and refinement for one-dimensional time dependent partial differential equations*. SIAM Journal on Numerical Analysis, Vol. 23, 778-796, 1986.
2. F. Andersson, B. Olsson (eds). *Lake Gådsjön. An acid forest lake and its catchment*. Ecological Bulletins, Vol. 37, Stockholm, 1985.
3. M. Bakker. *Software for semi-discretization of time-dependent partial differential equations in one space variable*. Report NW 52/77, Mathematisch Centrum, Amsterdam, The Netherlands, 1977.
4. J.G. Blom, P.A. Zegeling. *Algorithm 731: A moving grid interface for systems of one-dimensional time-dependent partial differential equations*. ACM Transactions on Mathematical Software, Vol. 20, No. 2, 194-214, 1994.
5. N. Caassen, S.A. Barber. *Simulation model for nutrient uptake from soil by a growing plant root system*. Agronomy Journal, Vol. 68, 961-964, 1976.
6. J.E. Dennis jr., D.M. Gay, R.E. Welsch. *Algorithm 573: NL2SOL-An adaptive nonlinear least-squares algorithm*. ACM Transactions on Mathematical Software, Vol. 7, No. 3, 369-383, 1993.
7. M. Dobmann, M. Liepelt, K. Schittkowski. *Algorithm 746: PCOMP: A FORTRAN code for automatic differentiation*. ACM Transactions on Mathematical Software, Vol. 21, No. 3, 233-266, 1995.
8. M. Dobmann, K. Schittkowski. *PDEFIT: A FORTRAN code for parameter estimation in partial differential equations - user's guide -*. Report, Department of Mathematics, University of Bayreuth, 1995.
9. J.R. Dormand, P.J. Prince. *High order embedded Runge-Kutta formulae*. Journal on Computational Applied Mathematics, Vol. 7, 67-75, 1981.
10. H.A. Dwyer, B.R. Sanders. *Numerical modeling of unsteady flame propagation*. Report SAND77-8275, Sandia National Laboratories, Livermore, USA, 1978.
11. G. Fairweather, J.C. Lopez-Marcos. *A box method for a nonlinear equation of population dynamics*. IMA Journal on Numerical Analysis, Vol. 11, 525 - 538, 1991.
12. P. Fischer. *Modellierung und Simulation der Amonium- und Stickstoffdynamik in strukturierten Waldböden*. Diplomarbeit, Fakultät für Mathematik und Physik, Universität Bayreuth, 1996.

13. J.E. Flaherty, P.K. Moore. *Integrated space-time adaptive hp-refinement methods for parabolic systems.* Applied Numerical Mathematics, Vol. 16, 317-341, 1995.
14. A. Gaßner. *Parameterschätzung in parabolischen Differentialgleichungen mit Anwendungen in der Goökologie.* Diplomarbeit, Fakultät für Mathematik und Physik, Universität Bayreuth, 1995.
15. A.G. Groch. *Automatic control of laminar flow cooling in continuous and reversing hot strip mills.* Iron and Steel Engineer, 16-20, 1990.
16. E. Hairer, S.P. Nørsett, G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems.* Series Computational Mathematics, Springer, 1991.
17. E. Hairer, G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.* Series Computational Mathematics, Springer, 1991.
18. K.H. Hartge, R. Horn. *Einführung in die Bodenphysik.* Enke, Stuttgart, 1991.
19. C. Hedrich. *Modellierung, Simulation und Parameterschätzung von Kühlprozessen in Walßstraßen.* Diplomarbeit, Fakultät für Mathematik und Physik, Universität Bayreuth, 1996.
20. S.A.M. Hotchkiss. *Skin as a xenobiotic metabolizing organ.* In: Process in Drug Metabolism, G.G. Gibson ed., Taylor and Francis Ltd., London, 217-262, 1992.
21. R. Hoch. *Modellierung von Fließwegen und Verweilzeiten in einem Einzugsgebiet unter stationären Fließbedingungen.* Diplomarbeit, Fakultät für Biologie, Chemie und Geowissenschaften, Universität Bayreuth, 1995.
22. J. Ingham, I.J. Dunn, E. Heinzle, J.E. Prenosil. *Chemical Engineering Dynamics.* VCH, Weinheim, New York, Basel, Cambridge, Tokyo, 1994.
23. J. Lang. *High-resolution selfadaptive computations on chemical reaction-diffusion problems with internal boundaries.* Preprint SC 94-9, ZIB Berlin, 1994.
24. P. Lindström. *A general purpose algorithm for nonlinear least squares problems with nonlinear constraints.* Report UMINF-103.83, Institute of Information Processing, University of Umea, Umea, Sweden, 1983.
25. N.K. Madsen. *MOLAG: A method of lines adaptive grid interface for nonlinear partial differential equations.* in: PDE Software: Modules, Interfaces and Systems, B. Engquist, T. Smedsaas eds., North-Holland, 1984.
26. M.C. MacCracken. *First annual report, DOT-CIAP program.* Rep. UCRL-51336, Lawrence Livermore Lab., Livermore, USA, 1973.
27. B. Nickel. *Parameterschätzung basierend auf der Levenberg-Marquardt-Methode in Kombination mit direkter Suche.* Diplomarbeit, Mathematisches Institut, Universität Bayreuth 1995.
28. U. Nowak. *A fully adaptive MOL-treatment of parabolic 1D-problems with extrapolation techniques.* Preprint SC 95-25, ZIB Berlin, 1995.
29. S.V. Pennington, M. Berzins. *New NAG Library software for first-order partial differential equations.* ACM Transactions on Mathematical Software, Vol. 20, No. 1, 63-99, 1994.
30. W. E. Schiesser. *The Numerical Method of Lines.* Academic Press, Inc., San Diego, 1991.
31. W. B. Pratt, P. Taylor. *Principles of Drug Action.* Churchill Livingstone, New York, 1990.
32. K. Schittkowski. *NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems.* Annals of Operations Research, Vol. 5, 485-500, 1985/86.
33. K. Schittkowski. *Solving nonlinear least squares problems by a general purpose SQP-method.* In: Trends in Mathematical Optimization, K.-H. Hoffmann, J.-B. Hiriart-Urruty, C. Lemarechal, J. Zowe eds., International Series of Numerical Mathematics, Vol. 84, Birkhäuser, 1988.
34. K. Schittkowski. *Parameter estimation in dynamical systems with EASY-FIT - user's guide* -. Report, Dept. of Mathematics, University of Bayreuth, 1996.
35. K. Schittkowski, M. Dobmann, M. Wolf. *Parameteridentifikation in transdermalen Systemen.* submitted for publication, 1996.
36. F. Seredynski. *Prediction of plate cooling during rolling-mill operation.* Journal of the Iron and Steel Institute, 197-203, 1973.

37. R.F. Sincovec, N.K. Madsen. *Software for nonlinear partial differential equations*. ACM Transactions on Mathematical Software, Vol. 1, No. 3, 232-260, 1975.
38. I. Steinsträsser. *The organized HaCaT cell culture sheet: A model approach to study epidermal peptide drug metabolism*. Dissertation, Pharmaceutical Institute, ETH Zürich, 1994.
39. R. Strehlow. *Grundzüge der Physik*. Vieweg-Verlag, Wiesbaden, 1995.
40. A.N. Tychonov, A.A. Samarski. *Differentialgleichungen der mathematischen Physik*. Band 39, VEB Deutscher Verlag der Wissenschaften, Berlin, 1959.
41. M.T. Van Genuchten, P.J. Wierenga. *Mass transfer studies in sorbing porous media. 1. Analytical solutions*. Soil Sci. Soc. Am. Journal, Vol. 44, 892-898, 1976.
42. J.G. Verwer, J.G. Blom, R.M. Furzeland, P.A. Zegeling. *A moving grid method for one-dimensional PDEs based on the method of lines*. in: Adaptive Methods for Partial Differential Equations, J.E. Flaherty, P.J. Paslow, M.S. Shephard, J.D. Vasilakis eds., SIAM, Philadelphia, Pa., 160-175, 1989.
43. M. Wolf. *Mathematisch-physikalische Berechnungs- und Simulationsmodelle zur Beschreibung und Entwicklung therapeutischer Systeme*. Habilitationsschrift, Mathematisch-Naturwissenschaftliche Fakultät, Universität Bonn, 1994.
44. M. Wolf, G.Lee. *Untersuchungen zur Verteilungsfunktion einer Übergangsschicht*. Vorabkurzfassung, Boehringer Ingelheim KG, 1995.
45. - *NAG FORTRAN Library Manual*. Mark 10, Vol. 2, Numerical Algorithms Group, Oxford, England, 1983.

6 APPENDIX: TEST EXAMPLES

All test problems are described in the same format. We give a short introduction into the background and describe the mathematical model in detail.

For the parameters to be estimated, we show the initial values p^0 from where the optimization algorithms are started, the final solution p^{method} obtained by one or several least squares codes, and the exact one $p^\star$, if measurements are simulated proceeding from a given set of parameters. The last line shows the corresponding residual, scaled for some test cases.

Moreover we present all experimental data for completeness. If measurements are simulated, we outline only the time values used for a simulation run by which the data have been computed. These data are rounded w.r.t. two exact digits.

6.1 HEAT

1. **Background:** Simple test example describing a heat diffusion process with constant diffusion parameter, see e.g. Schiesser[30] (*Fourier's first law* for heat conduction).
2. **System equations:**
$$u_t(x,t) = Du_{xx}(x,t)$$
for $0 < t \leq 0.5$, $0 < x < 1$
3. **Initial conditions:**
$$u(x,0) = \sin\left(\frac{\pi x}{L}\right)$$
for $0 < x < 1$
4. **Boundary conditions:** $u(0,t) = u(1,t) = 0$ for $0 \leq t \leq 0.5$

5. **Coupled ODE's:** none
6. **Transmission conditions:** none
7. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
D	0.01	1.0007	1.0007	1.0006	1.0006	1.0
L	2.0	1.0012	1.0010	1.0010	1.0010	1.0
res	3.4	0.16E-6	0.16E-6	0.16E-6	0.16E-6	0.21E-6

8. **Fitting criterion:** $u(0.5, t)$ for $0 < t \leq 0.5$
9. **Measurements:** Simulated at time coordinates 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5.

6.2 RICHARDS

1. **Background:** Vertical diffusion of water in soil, modelled by Richards equation, see e.g. Hartge and Horn[18].
2. **Remarks:** The resulting ODE is very unstable in the sense, that the equation cannot be integrated when using an inappropriate set of inital parameters. Also some of the least squares algorithms are terminated because of numerical difficulties reported by the ODE solver. There is no exact fit of the data indicating that something is wrong with the model, the data or the link between model and data, e.g. the dimension. Although the residual is reduced from 0.12 to 0.0058, the initial parameters are changed only very slightly.
3. **System equations:**

$$\frac{\partial}{\partial t}\theta(u(x,t)) = \frac{\partial}{\partial z}(K(u(x,t))u_x(x,t)) + \frac{\partial}{\partial x}K(u(x,t))$$

with

$$\theta(u) = \begin{cases} \theta_r + (\theta_s - \theta_r)(1 + (-\alpha u)^n)^{1/n-1} & \text{if } u < 0 \\ \theta_s & \text{otherwise} \end{cases}$$

and

$$K(u) = K_s(1 + (-\alpha u)^n)^{(1/n-1)/2}(1 - (-\alpha u))^n(1 + (-\alpha u)^n)^{1/n-1})^2$$

if $u < 0$, and $K(u) = K_s$ otherwise, for $0 < t$, $15 < x < 55$

4. **Initial conditions:** Linear interpolation of the data

x_i	$u(x_i, 0)$
15.0	-14.8941
40.0	-12.4942
70.0	-11.6868

see Gaßner[14]

5. **Boundary conditions:** Linear interpolated data for u at left and right boundary, see Table 5 or Gaßner[14].

6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}
θ_r	0.25	0.227
θ_s	0.4	0.408
α	0.1	0.00912
n	7.1	7.129
K_s	30.0	29.98
res	0.12	0.0058

9. **Fitting criterion:** $u(40,t)$ for $0 < t$
10. **Measurements:** See Gaßner[14]

t_i	y_i	t_i	y_i
0.166	-13.741	1.833	-19.991
0.333	-14.241	1.999	-20.241
0.499	-14.991	2.166	-20.744
0.666	-15.741	2.333	-20.994
0.833	-16.488	2.499	-21.494
0.999	-17.241	2.666	-21.997
1.166	-17.491	2.833	-22.247
1.332	-18.241	2.999	-22.497
1.499	-18.741	3.166	-22.747
1.666	-19.491	3.333	-23.250

6.3 G_WATER

1. **Background:** Tracer experiment to investigate acidification of groundwater pollution, see Andersson and Olsson[2] or Hoch[21].
2. **Remarks:** The initial parameters are quite close to the optimal ones, since they are altered only very slightly and the scaled total residual is reduced from 0.036 to 0.014. The search method DSLMDF computed the best fit, however with a very large number of function evaluations.
3. **System equations:**

$$\begin{aligned} \theta_m \frac{\partial c_m}{\partial t}(x,t) + \theta_{im} \frac{\partial c_{im}}{\partial t}(x,t) &= \theta_m D_m \frac{\partial^2 c_m}{\partial x^2}(x,t) - \theta_m V_m \frac{\partial c_m}{\partial x}(x,t) \\ \theta_{im} \frac{\partial c_{im}}{\partial t}(x,t) &= \alpha \left(c_m(x,t) - c_{im}(x,t) \right) \end{aligned}$$

   for $0 < t \leq 2.55$, $0 < x < 80$, see Van Genuchten and Wierenga[41]
4. **Initial conditions:** $c_m(x,0) = 0$, $c_{im}(x,0) = 0$ for $0 < x < 80$

t_i	$u(15, t_i)$	t_i	$u(55, t_i)$
0.0000	-14.8941	0.0000	-11.6868
0.1661	-15.4185	0.1661	-12.4372
0.3327	-15.9429	0.3324	-12.9375
0.4994	-16.4673	0.4991	-13.6780
0.6661	-16.9755	0.6658	-13.9281
0.8327	-17.5000	0.8325	-14.6686
0.9994	-17.7745	0.9991	-14.9287
1.1661	-18.2908	1.1658	-15.6790
1.3327	-18.5654	1.3324	-15.9192
1.4991	-18.8399	1.4991	-16.1693
1.6661	-19.1063	1.6661	-16.4195
1.8327	-19.3645	1.8327	-17.1598
1.9994	-19.8891	1.9994	-17.1598
2.1661	-20.1556	2.1661	-17.4100
2.3327	-20.4220	2.3325	-17.6601
2.4994	-20.7048	2.4991	-17.9102
2.6661	-20.9631	2.6658	-18.1503
2.8327	-21.2295	2.8324	-18.4104
2.9994	-21.4960	2.9991	-18.6505
3.1661	-21.5124	3.1658	-18.6505
3.3327	-21.7871	3.3324	-19.4008
3.4991	-22.0454	3.4991	-19.4008
3.6661	-22.3201	3.6661	-19.3908
3.8327	-22.3365	3.8327	-19.6409
3.9994	-22.6030	3.9994	-19.6409
4.1661	-22.8778	4.1661	-19.8910
4.3327	-22.8860	4.3327	-20.1411
4.4994	-23.4109	4.4991	-20.1310
4.6661	-23.6692	4.6658	-20.3811
4.8327	-23.9358	4.8324	-20.3811

TABLE 5: Boundary values for RICHARDS

5. **Boundary conditions:**

$$c_m(0,t) - \frac{D_m}{V_m}\frac{\partial c_m}{\partial x}(0,t) = \begin{cases} 5800, & \text{if } t < 0.01042; \\ 0, & \text{otherwise.} \end{cases}$$

$$c_m(80,t) + \frac{D_m}{V_m}\frac{\partial c_m}{\partial x}(80,t) = 0$$

for $0 \le t \le 2.55$

6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}
V_m	70.0	70.1	55.8	78.5	134.7
D_m	270.0	270.0	394.8	280.8	130.8
α	0.8	0.807	1.418	0.704	0.145
θ_m	0.16	0.109	0.138	0.0885	0.00960
θ_{im}	0.17	0.136	0.121	0.134	0.306
res	0.036	0.014	0.017	0.015	0.013

9. **Fitting criterion:**

$$h(t) = c_m(40,t) - \frac{D_m}{V_m}\frac{\partial c_m}{\partial x}(40,t)$$

for $0 < t \le 2.55$

10. **Measurements:** See Table 6, Andersson and Olsson[2] or Hoch[21]

6.4 SKIN_2A

1. **Background:** In vitro experiment with two membranes to describe the diffusion of a substance through human skin and the generation of a metabolite, see Wolf[43], Wolf and Lee[44], and Schittkowski, Dobmann and Wolf[35].
2. **Remarks:** Although we start from a continuous transition, the algorithm computes a non-continuous transmission. Very probably the system is overdetermined because of 10 parameters to be estimated.
3. **System equations:**

$$\begin{aligned} u_t^s(x,t) &= D_1^s u_{xx}^s(x,t) - V_{max} u^s(x,t)/(K_m + u^s(x,t)) \\ u_t^m(x,t) &= D_1^m u_{xx}^m(x,t) + V_{max} u^s(x,t)/(K_m + u^s(x,t)) \end{aligned}$$

for $0 \le x \le l_1$, $0 \le t \le T$, with Michaelis-Menten-effect, see Hotchkiss[20], Pratt and Taylor[31],

$$\begin{aligned} u_t^s(x,t) &= D_2^s u_{xx}^s(x,t) \\ u_t^m(x,t) &= D_2^m u_{xx}^m(x,t) \end{aligned}$$

t_i	y_i	t_i	y_i
0,11	17.80	1.35	29.31
0,12	0.66	1.38	28.40
0.15	11.58	1.41	27.32
0.18	9.54	1.44	26.13
0.24	32.09	1.47	23.95
0.27	23.47	1.50	22.40
0.30	28.30	1.53	23.33
0.33	29.42	1.56	19.56
0.36	32.61	1.59	15.85
0.39	33.16	1.62	22.93
0.42	34.46	1.65	22.10
0.45	36.54	1.68	19.60
0.48	38.34	1.71	20.71
0.51	40.30	1.74	20.45
0.54	41.23	1.77	19.42
0.57	41.57	1.80	18.14
0.60	40.97	1.83	17.48
0.63	41.98	1.86	17.29
0.66	43.71	1.89	17.69
0.69	45.45	1.92	18.41
0.72	44.43	1.95	18.27
0.75	40.39	1.98	17.61
0.78	43.19	2.01	16.57
0.81	43.04	2.04	15.29
0.84	41.13	2.07	13.88
0.87	40.13	2.10	12.53
0.90	40.01	2.13	11.38
0.93	38.87	2.16	10.92
0.96	39.95	2.19	10.82
0.99	37.19	2.22	10.75
1.02	37.10	2.25	10.20
1.05	39.11	2.28	9.53
1.08	39.42	2.31	8.80
1.11	34.03	2.34	8.10
1.14	31.62	2.37	7.44
1.17	34.80	2.40	6.92
1.20	35.88	2.43	6.56
1.23	34.30	2.46	6.25
1.26	31.73	2.49	5.96
1.29	30.36	2.52	5.62
1.32	29.90	2.55	5.21

TABLE 6: Experimental data for G_WATER

for $l_1 \leq x \leq l$, $0 \leq t \leq T$, where $l = 0.01$, $l_1 = 0.005$, $K_m = 0.007$ and $T = 30$, see also Wolf[43], Wolf and Lee[44]

4. **Initial conditions:** $u^s(x,0) = 0$, $u^m(x,0) = 0$ for $x > 0$ and $u^s(0,0) = Y_0 P^s/V_a$, $u^m(0,0) = 0$ otherwise, with $V_a = 3000$
5. **Boundary conditions:**

$$\begin{aligned} u^s(0,t) &= P^s/V_a v^s(t) \\ u^m(0,t) &= P^m/V_a u^m(t) \\ u^s(l,t) &= P^s/V_a u^s(t) \\ u^m(l,t) &= P^m/V_a u^m(t) \end{aligned}$$

for $0 \leq t \leq T$

6. **Coupled ODE's:**

$$\begin{aligned} \dot{v}^s(t) &= P^s/V_a F_a D^s u_x^s(0,t) \\ \dot{v}^m(t) &= P^m/V_a F_a D^m u_x^m(0,t) \\ \dot{w}^s(t) &= -P^s/V_a F_a D^s u_x^s(l,t) \\ \dot{w}^m(t) &= -P^m/V_a F_a D^m u_x^m(l,t) \end{aligned}$$

for $0 \leq t \leq T$, where $v^s(0) = Y_0$, $v^m(0) = 0$, $w^s(0) = 0$, $w^m(0) = 0$, $F_a = 63.6$

7. **Transmission conditions:**

$$\begin{aligned} u_x^s(l_1^-,t) &= T^s u_x^s(l_1^+,t) \\ u_x^m(l_1^-,t) &= T^m u_x^m(l_1^+,t) \\ u_x^s(l_1^+,t) &= D_2^s/D_1^s u_x^s(l_1^-,t) \\ u_x^m(l_1^+,t) &= D_2^m/D_1^m u_x^m(l_1^-,t) \end{aligned}$$

for $0 \leq t \leq T$

8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}
D_1^s	0.01	0.0400	0.00258	0.0423	0.0155
D_1^m	1.0	0.573	1.571	1.93	6.230
D_2^s	1.0	2.83	80.57	2.00	975.8
D_2^m	1.0	1.38	3.753	2.59	2.328
P^s	0.05	0.029	0.459	0.0289	0.0734
P^m	0.001	0.0039	0.0139	0.000965	0.000206
V_{max}	35.0	34.9	4.661	35.0	16.41
Y_0	109.0	106.3	106.3	106.4	106.5
T^s	1.0	0.244	0.322	0.234	0.274
T^m	1.0	1.22	0.529	2.07	9.07
res	0.93	0.0015	0.00079	0.0015	0.0013

9. **Fitting criteria:** $v^s(t)$, $v^m(t)$, $w^s(t)$, $w^m(t)$ for $0 \leq t \leq T$

10. **Measurements:**

t_i	y_i^1	y_i^2	y_i^3	y_i^4
0.0	107	-	-	-
2.0	105	0.8	0.6	0.3
5.0	102	2.2	1.4	0.8
7.0	99	3.0	1.9	1.0
10.0	97	4.5	2.5	1.8
20.0	89	8.9	4.0	3.9
30.0	81	13.1	5.0	6.2

cf. Steinsträsser[38]

6.5 SKIN_3A

1. **Background:** Same experiment as for SKIN_2A, but with full modelling of smooth transition between areas.
2. **Remarks:** The transition between two different diffusion areas is modelled in detail. Very probably the system is overdetermined because of 10 parameters to be estimated.
3. **System equations:**

$$\begin{aligned} u_t^s(x,t) &= D_1^s u_{xx}^s(x,t) - V_{max} u^s(x,t)/(K_m + u^s(x,t)) \\ u_t^m(x,t) &= D_1^m u_{xx}^m(x,t) + V_{max} u^s(x,t)/(K_m + u^s(x,t)) \end{aligned}$$

for $0 \le x \le l_1$, $0 \le t \le T$,

$$\begin{aligned} u_t^s(x,t) &= \left(D_1^s + \tfrac{D_2^s - D_1^s}{l_2 - l_1}(x - l_1)\right)(\mu_s u_x^s(x,t) + u_{xx}^s(x,t)) \\ &\quad + \tfrac{D_2^s - D_1^s}{l_2 - l_1}(\mu_s u^s(x,t) + u_x^s(x,t)) \\ u_t^m(x,t) &= \left(D_1^m + \tfrac{D_2^m - D_1^m}{l_2 - l_1}(x - l_1)\right)(\mu_m u_x^m(x,t) + u_{xx}^m(x,t)) \\ &\quad + \tfrac{D_2^m - D_1^m}{l_2 - l_1}(\mu_m u^m(x,t) + u_x^m(x,t)) \end{aligned}$$

for $l_1 \le x \le l_2$, $0 \le t \le T$,

$$\begin{aligned} u_t^s(x,t) &= D_2^s u_{xx}^s(x,t) \\ u_t^m(x,t) &= D_2^m u_{xx}^m(x,t) \end{aligned}$$

for $l_2 \le x \le l$, $0 \le t \le T$, where $l = 0.01$, $l_1 = 0.0045$, $l_2 = 0.0055$, $K_m = 0.007$, and $T = 30$, see Schittkowski, Dobmann and Wolf[35] or Wolf[43]

4. **Initial conditions:** $u^s(x,0) = 0$, $u^m(x,0) = 0$ for $x > 0$ and $u^s(0,0) = Y_0 P^s / V_a$, $u^m(0,0) = 0$ otherwise
5. **Boundary conditions:**

$$\begin{aligned} u^s(0,t) &= P^s/V_a v^s(t) \\ u^m(0,t) &= P^m/V_a u^m(t) \\ u^s(l,t) &= P^s/V_a u^s(t) \\ u^m(l,t) &= P^m/V_a u^m(t) \end{aligned}$$

for $0 \le t \le T$, $V_a = 3000$

6. **Coupled ODE's:**

$$\begin{aligned} \dot{v}^s(t) &= P^s/V_a F_a D^s u_x^s(0,t) \\ \dot{v}^m(t) &= P^m/V_a F_a D^m u_x^m(0,t) \\ \dot{w}^s(t) &= -P^s/V_a F_a D^s u_x^s(l,t) \\ \dot{w}^m(t) &= -P^m/V_a F_a D^m u_x^m(l,t) \end{aligned}$$

for $0 \le t \le T$, where $v^s(0) = Y_0$, $v^m(0) = 0$, $w^s(0) = 0$, $w^m(0) = 0$, $F_a = 63.6$

7. **Transmission conditions:**
Smooth transition functions, i.e.

$$\begin{aligned} u_x^s(l_1^-,t) &= u_x^s(l_1^+,t) \\ u_x^m(l_1^-,t) &= u_x^m(l_1^+,t) \\ u_x^s(l_1^+,t) &= u_x^s(l_1^-,t) \\ u_x^m(l_1^+,t) &= u_x^m(l_1^-,t) \end{aligned}$$

for $0 \le t \le T$

8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}
D_1^s	0.01	0.0094	0.0022	0.00460	0.00941
D_1^m	1.0	0.354	0.0344	0.137	0.324
D_2^s	1.0	0.723	14.96	0.334	0.720
D_2^m	1.0	1.605	2.121	0.933	1.461
P^s	0.05	0.0519	0.094	0.104	0.0520
P^m	0.001	0.0090	0.0	0.00621	0.0099
V_{max}	35.0	34.99	47.22	17.73	34.94
Y_0	109.0	104.3	102.9	105.7	104.2
μ^s	0.0	-0.361	110.7	0.0	-0.367
μ^m	0.0	0.0831	-661.6	0.0	0.128
res	0.42	0.0031	0.0049	0.0055	0.0031

9. **Fitting criteria:** $v^s(t)$, $v^m(t)$, $w^s(t)$, $w^m(t)$ for $0 \le t \le T$

10. **Measurements:**

t_i	y_i^1	y_i^2	y_i^3	y_i^4
0.0	107	-	-	-
2.0	105	0.8	0.6	0.3
5.0	102	2.2	1.4	0.8
7.0	99	3.0	1.9	1.0
10.0	97	4.5	2.5	1.8
20.0	89	8.9	4.0	3.9
30.0	81	13.1	5.0	6.2

cf. Steinsträsser[38]

6.6 KINETIC

1. **Background:** Diffusion of a substance through skin with additional kinetic model for systemic cycle, see Wolf[43], without metabolism.
2. **Remarks:** The transition between two different diffusion areas is modelled in detail. Ordinary differential equations are coupled at both boundaries. Very probably the system is overdetermined, since 10 parameters are to be estimated and only 15 measurements are available.
3. **System equations:**

   $$u_t(x,t) = D_1 u_{xx}(x,t)$$

   for $0 \le x \le l_1$, $0 \le t \le T$,

   $$u_t(x,t) = \frac{\partial}{\partial x}\left((D_1 + \frac{D_2 - D_1}{l_2 - l_1}(x - l_1))(\mu u(x,t) + u_x(x,t))\right)$$

   for $l_1 \le x \le l_2$, $0 \le t \le T$,

   $$u_t(x,t) = D_2 u_{xx}(x,t)$$

   for $l_2 \le x \le l$, $0 \le t \le T$, where $l = 0.21$, $l_1 = 0.1$, $l_2 = 0.11$, $\mu = -50$, $D_0 = 12$, and $T = 6$
4. **Initial conditions:** $u(x,0) = 0$ for $0 \le x \le l$
5. **Boundary conditions:**

   $$\begin{aligned} u(0,t) &= y_1(t)/V_1 \\ u(l,t) &= y_4(t)/V_4 \end{aligned}$$

   for $0 \le t \le T$, where $V_1 = 0.1$ and $V_4 = 1$
6. **Coupled ODE's:**

$$
\begin{aligned}
\dot{y}_1(t) &= D_0 k \exp(-kt) - (k_{12} + k_{13}) y_1(t) + k_{31} y_3(t) + F_a D_1 u_x(0,t) \\
\dot{y}_2(t) &= -k_{12} y_1(t) \\
\dot{y}_3(t) &= k_{13} y_1(t) - k_{31} y_3(t) \\
\dot{y}_4(t) &= -k_{45} y_4(t) + k_{54} y_5(t) - F_a D_2 u_x(0,t) - k_p(r_0 - c(t)) y_4(t) + k_m c(t) V_4 \\
\dot{y}_5(t) &= -k_{54} y_5(t) + k_{45} y_4(t) - F_a D_2 u_x(l,t) \\
\dot{c}(t) &= k_p(r_0 - c(t)) y_4(t)/V_4 - k_m c(t)
\end{aligned}
$$

for $0 \le t \le T$, where $y_1(0) = 0$, $y_2(0) = 0$, $y_3(0) = 0$, $y_4(0) = 0$, $y_5(0) = 0$, $c(0) = 0$, $F_a = 0.5$, $r_0 = 160$

7. **Transmission conditions:**
   Continuous transition functions, i.e.

$$
\begin{aligned}
u(l_1^+, t) &= u(l_1^-, t) \\
u(l_2^-, t) &= u(l_2^+, t) \\
u_x(l_1^-, t) &= \mu u(l_1^+, t) + u_x(l_1^+, t) \\
u_x(l_2^+, t) &= \mu u(l_2^-, t) + u_x(l_2^-, t)
\end{aligned}
$$

   for $0 \le t \le T$

8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}
k	50.0	-	3315.7	-	0.925
k_{12}	0.0	-	4.84	-	0.0
k_{13}	6.0	-	15.47	-	10.90
k_{31}	0.5	-	1.04	-	0.295
k_{45}	0.0	-	0.497	-	7.85
k_{54}	1.0	-	1.82	-	3.67
D_1	0.001	-	0.00065	-	0.175
D_2	1.0	-	0.726	-	0.513
k_p	0.01	-	0.0602	-	0.0561
k_m	0.01	-	0.113	-	0.0
res	0.34E+5	-	0.039	-	0.063

9. **Fitting criteria:** $y_1(t)/V_1$, $y_4(t)/V_4$, $y_5(t)/V_5$ for $0 \le t \le T$, where $V_5 = 5$

10. **Measurements:**

t_i	y_i^1	y_i^2	y_i^3
0.5	4.72	0.94	0.38
1.0	3.06	1.0	0.62
2.0	2.34	0.72	0.39
4.0	1.77	0.26	0.25
6.0	1.69	0.0	0.06

6.7 NAG_1

1. **Background:** Test problem for subroutine D03PBF of NAG-Library with some applications in electrodynamics, see NAG[45].
2. **Remarks:** The parameter estimation problem is solved easily by all algorithms.
3. **System equations:**

$$\begin{aligned} u_t(x,t) &= a u_{xx}(x,t) - f(c(u(x,t) - v(x,t))) \\ v_t(x,t) &= b v_{xx}(x,t) + f(c(u(x,t) - v(x,t))) \end{aligned}$$

   for $t > 0$ and $0 < x < 1$, where $f(z) = \exp(z) - \exp(-2z)$
4. **Initial conditions:** $u(x,0) = 1$ and $v(x,0) = 0$ for $0 < x < 1$
5. **Boundary conditions:** $u_x(0,t) = 0$, $v(0,t) = 0$, $u(1,t) = 1$, $v_x(1,t) = 0$ for $t > 0$
6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
a	1.0	0.02392	0.02393	0.02393	0.0238	0.024
b	1.0	0.1698	0.1699	0.1699	0.1696	0.17
c	1.0	5.728	5.728	5.728	5.725	5.73
res	6.6	0.14E-5	0.14E-5	0.14E-5	0.15E-5	0.15E-5

9. **Fitting criteria:** $u(0,t)$ and $v(1,t)$ for $0 < t \leq 0.5$
10. **Measurements:** Simulated at 19 time coordinates 0, 0.1, 0.2, 0.3, ..., 1, 1.5, 2, 2.5, ..., 0.45, 0.5.

6.8 NAG_2

1. **Background:** Test problem for subroutine D03PBF of NAG-Library with interface, see NAG[45].
2. **Remarks:** Starting values for the parameters to be estimated, are far away from the optimal ones. Although the resulting least squares problem is difficult to solve, parameters are identified more or less correctly.
3. **System equations:**

$$u_t(x,t) = a(2u_x(x,t)/x + u_{xx}(x,t)) - b\exp(u(x,t))$$

   for $t > 0$ and $0 < x < 0.5$,

$$u_t(x,t) = (2u_x(x,t)/x + u_{xx}(x,t)) - \exp(u(x,t))$$

   for $t > 0$ and $0.5 \leq x < 1$
4. **Initial conditions:** $u(x,0) = 0$ for $x < 1$ and $u(1,0) = 1$ otherwise
5. **Boundary conditions:** $u_x(0,t) = 0$, $u(1,t) = 1$ for $t > 0$

6. **Coupled ODE's:** none
7. **Transmission conditions:** $u_x(0.5^+, t) = a u_x(0.5^-, t)$ for all $t > 0$
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
a	1.0	5.29	5.30	5.30	9.97	5.0
b	1.0	987.6	987.6	987.6	890.0	1000.0
res	169.3	0.97E-4	0.97E-4	0.97E-4	0.25E-2	0.10E-3

9. **Fitting criterion:** $u(0.5, t)$ for $t > 0$
10. **Measurements:** Simulated at 23 time coordinates 0.01, 0.02, ..., 0.12, 0.13, 0.15, 0.2, 0.3, ..., 0.9, 1

6.9 NITROGEN

1. **Background:** Nitrogen and ammonium dynamics in forrest soils, see Caassen, Barber[5] and Fischer[12].
2. **Remarks:** Starting values for the parameters to be estimated, are far away from the optimal ones. Thus the resulting least squares problem is difficult to solve, and none of the algorithms is able to identify parameters correctly.
3. **System equations:**

$$\begin{aligned} u_t(x,t) &= (D u_{xx}(x,t) - 70 u_x(x,t) - m_1 L(t) u(x,t)) \\ &\quad + 1.387(r(t) - v(x,t)))/1.216 \\ v_t(x,t) &= (D v_{xx}(x,t) - 70 v_x(x,t) - m_2 L(t) v(x,t)) + 1.387 u(x,t))/3.8 \end{aligned}$$

for $t > 0$ and $0 < x < 1$, where $r(t) = 4.5 \exp(-1.387t)$ and

$$L(t) = 0.000172 \exp(5t/(6t+1)) + 0.0008 \exp(18t/(13t+30))$$

4. **Initial conditions:** $u(x, 0) = 1$, $v(x, 0) = 0.5$ for $0 \le x \le 1$
5. **Boundary conditions:**

$$\begin{aligned} u_x(0,t) &= (15.79 u(0,t) - 15.84)/D \\ u_x(1,t) &= 0 \\ v_x(0,t) &= (15.79 v(0,t) - 7.921)/D \\ v_x(1,t) &= 0 \end{aligned}$$

for all $t > 0$

6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
m_1	1.0	226.9	202.4	202.4	17.65	182.5
m_2	1.0	15.93	150.7	150.7	0.615	182.5
D	0.01	25.63	29.5	29.5	25.65	31.536
res	1.7	0.11E-3	0.41E-5	0.41E-5	0.15E-2	0.21E-4

9. **Fitting criterion:** $u(0.5,t)$ and $v(0.9,t)$ for $t>0$
10. **Measurements:** Simulated at 10 time coordinates 0.5, 1, ..., 5.

6.10 CONTAMIN

1. **Background:** Contamination of liquids.
2. **Remarks:** Data are very bad and the model is unstable. A reasonable fit cannot be obtained. Either model, data or scaling are wrong.
3. **System equations:**
$$u_t^i(x,t) = Du_{xx}^i(x,t)$$
for $i=1\ldots4$, $t>0$ and $0<x<10.7$,
4. **Initial conditions:** $u^i(x,0)=c_i$, $i=1,\ldots,4$, for $0<x<10.7$ with $c_1=0.5$, $c_2=1$, $c_3=2$, $c_4=5$
5. **Boundary conditions:** $u^i(0,t)=0$, $u^i(10.7,t)=Vy^i(t)/V_a$ for $i=1,\ldots,4$, $V_a=0.00505$, and $t>0$
6. **Coupled ODE's:**
$$\dot{y}^i(t) = -DF_a u_x^i(10.7,t)$$
for $t>0$ with $y^i(0)=c_iV_a/V$, $i=1,\ldots,4$
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}
D	1.0	0.038	0.705	0.0	0.0
V	10.0	95.8	95.8	95.8	93.2
F_a	1.0	1.02	0.00	47.1	0.287
res	0.70	0.0045	0.0045	0.0045	0.0046

9. **Fitting criterion:** $y^i(t)/V_a$ for $i=1,\ldots,4$ and $t>0$
10. **Measurements:**

t_i	y_i^1	y_i^2	y_i^3	y_i^4
2.0	0.00236	0.00375	0.00909	0.01183
8.0	0.01061	0.00375	0.03463	0.02915
24.0	0.00822	0.00938	0.02698	0.06002
48.0	0.00404	0.01125	0.01804	0.08210
72.0	0.02006	0.00837	0.03492	0.06652
96.0	0.01128	0.02424	0.02857	0.05050

6.11 MILL

1. **Background:** Plate cooling in rolling mills with two cooling areas (water and air), see Hedrich[19] or Groch[15].
2. **Remarks:** The optimization parameters appear only in a boundary coefficient where a linear dependency is assumed. The model is ill-posed in the sense that parameters cannot be identified correctly.
3. **System equations:**

$$u_t(x,t) = (k_\lambda u_x(x,t)^2 + \lambda(u(x,t))u_{xx}(x,t))/(p(u(x,t))c_p(u(x,t))$$

for $0 < t \leq 30$ and $0 < x < 10$, where $\lambda(u) = k_\lambda^0 + k_\lambda u$, $p(u) = k_p^0 + k_p u$ and $k_\lambda = 0.01$, $k_\lambda^0 = 15$, $k_p^0 = 7.85$, $k_p = -0.00033$, and $c_p(u)$ is a linear interpolation of the data

u_i	$c_p(u_i)$
0.0	0.68
780.0	1.1
790.0	2.8
840.0	0.72
880.0	0.7
920.0	0.6
1400.0	0.73

see e.g. Tychonoff and Samarski[40]

4. **Initial conditions:** $u(x,0) = 900$ for all x
5. **Boundary conditions:**

$$\begin{aligned} u_x(x_b,t) &= 0.001(\alpha u(x_b,t) - u_r) + E(t)c((u(x_b,t) + 273.15)^4 \\ &\quad -(u_r + 273.15)^4)/\lambda(u(x_b,t)) \end{aligned}$$

where $E(t) = 0.001u(x_b,t)(0.000125u(x_b,t) - 0.38) + 1.1$, $c = 5.67E - 8$, $x_b = 0$ or $x_b = 10$, respectively, and

$$\begin{aligned} u_r &= 20 \\ \alpha &= 1000(1 + \psi) \\ \psi &= a_1 + 40a_2 + 900a_3 + 20a_4 + 690a_5 \end{aligned}$$

for $4 < t \leq 20.5$, and

$$\begin{aligned} u_r &= 50 \\ \alpha &= 60 \end{aligned}$$

otherwise, see e.g. Strehlow[39] and Seredynski[36]

6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
a_1	0.0	-0.00121	-0.0190	-0.0211	-0.406	-0.70
a_2	0.0	-0.000580	-0.0290	-0.0283	0.0	0.0145
a_3	0.0	-0.000960	0.0000576	0.000339	0.0	0.0012
a_4	0.0	0.000610	0.135	0.135	0.0	0.0005
a_5	0.0	0.000676	-0.00288	-0.00329	0.0	-0.002
res	0.011	0.14E-6	0.14E-6	0.14E-6	0.12E-5	0.14E-6

9. **Fitting criterion:** $u(0,t)$ for $t > 0$
10. **Measurements:** Simulated at 40 equidistant time coordinates between 0 and 40

6.12 DISRET

1. **Background:** Non-isothermal turbular reactor with with first-order reaction for axial heat dispersion, see Ingham e.al.[22].
2. **Remarks:** The right-habnd side of the PDE contains an exponential term containing the Arrhenius constants of the substances. Thus the model replects a typical situation arising very often in chemical engineering. Parameters can be identified by three algorithms easily, but the search method DSLMDF failed.
3. **System equations:**

$$\begin{aligned} c_t(x,t) &= c_{xx}(x,t) - p_1 c_x(x,t) - r_1(x,t) \\ T_t(x,t) &= (\alpha/D) T_{xx}(x,t) - p_2 T_x(x,t) - b_2 r_1(x,t) \end{aligned}$$

with

$$\begin{aligned} \alpha &= \lambda/(\rho c_p) \\ b_1 &= z_f l^2/D \\ b_2 &= H_r c_0/(\rho c_p T_0) \\ p_1 &= vl/D \\ p_2 &= vl/\alpha \\ r_1(x,t) &= b_1 c(x,t) \exp(-E_a/(RT_0 T(x,t))) \end{aligned}$$

for $0 < t \leq 0.3$ and $0 < x < 1$, $l = 100$, $c_0 = 0.5$, $T_0 = 637$, $v = 0.3$, $\rho = 60$, $H_r = -36000$, $c_p = 0.8$, $E_a = 40000$, $z_f = 12000$, $R = 3.577$

4. **Initial conditions:** $c(x,0) = 0.0$ and $T(x,0) = 1$ for all x
5. **Boundary conditions:** $c(0,t) = 1$, $T(0,t) = 1$, $c_x(1,t) = 0$, $T_x(1,t) = 0$ for all $t > 0$
6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
D	5.0	0.99996	0.99996	0.99995	-	1.0
λ	1.0	48.14	48.14	48.14	-	48.0
res	3.0	0.00021	0.00021	0.00021	-	0.00028

9. **Fitting criterion:** $c(1,t)$ and $T(1,t)$ for $t > 0$
10. **Measurements:** Simulated at 19 time coordinates 0.01, 0.02, ..., 0.08, 0.1, ..., 0.3

6.13 SHEAR

1. **Background:** Shear band formation with fast moving front, see Nowak[28] and Flaherty, Moore[13].
2. **Remarks:** The optimization algorithms computed different solutions. The best one was obtained by the heuristic search method DSLMDF, but with a large number of function evaluations.
3. **System equations:**

$$\begin{aligned} u_t(x,t) &= v(x,t) \\ v_t(x,t) &= (g(T(x,t))u_x(x,t))_x + v_{xx}(x,t)/R_e \\ T_t(x,t) &= T_{xx}(x,t)/(P_r R_e) + v_x(x,t)^2/R_e \end{aligned}$$

   with

$$g(T) = 0.5((1+g_{inf}) - (1-g_{inf})\tanh((T-T_m)/\Delta T))$$

   for $0 < t \leq 0.3$ and $0 < x < l$, $\Delta T = 0.01$
4. **Initial conditions:** $u(x,0) = 0$, $v(x,0) = 0$ and $T(x,0) = 0$ for all x
5. **Boundary conditions:**

$$v(1,t) = \begin{cases} v_0 t/r, & \text{if } t < r \\ v_0, & \text{if } r \leq t < d-r \\ v_0(d-t)/r, & \text{if } d-r \leq t < d \\ 0, & \text{if } t \geq d \end{cases}$$

   $v(0,t) = 0$, $T(0,t) = 0$, $T(1,t) = 0$ for all $t > 0$, where $v_0 = 0.5$, $d = 1.5$, and $r = 0.05$
6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
R_e	200.0	199.9	81.6	80.36	84.3	100.0
P_r	100.0	99.9	1005.8	0.817	16.6	50.0
g_{inf}	0.1	0.751	0.539	0.00035	0.0	0.05
T_m	0.01	0.0138	0.0246	0.0283	0.0338	0.03
res	0.89	0.13	0.052	0.045	0.042	0.049

9. **Fitting criterion:** $u(0.5,t)$, $v(0.5,t)$ and $T(0.5,t)$ for $t > 0$
10. **Measurements:** Simulated at 11 equidistant time coordinates 0.3, 0.6, ..., 3.3

6.14 EL_DYN

1. **Background:** Electrodynamic model with steep boundary layers, see Blom and Zegeling[4] and Bakker[3].
2. **Remarks:** The resulting parameter estimation problem can be solved more or less easily by all optimization codes.
3. **System equations:**

$$\begin{aligned} u_t(x,t) &= \epsilon p u_{xx}(x,t) - g(u(x,t) - v(x,t)) \\ v_t(x,t) &= p v_{xx}(x,t) + g(u(x,t) - v(x,t)) \end{aligned}$$

   with

$$g(z) = \exp(\eta z/3) - \exp(-2\eta z/3)$$

   for $0 < t \leq 0.3$ and $0 < x < l$,
4. **Initial conditions:** $u(x,0) = 1$ for all x
5. **Boundary conditions:** $u_x(0,t) = 0$, $u(1,t) = 1$, $v(0,t) = 0$, $v_x(1,t) = 0$ for all $t > 0$
6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
ϵ	1.0	0.14311	0.14309	0.14309	0.1430	0.143
p	1.0	0.1751	0.1749	0.17493	0.1748	0.1743
η	1.0	17.21	17.21	17.21	17.20	17.19
res	8.0	0.85E-6	0.82E-6	0.82E-6	0.83E-6	0.20E-5

9. **Fitting criterion:** $u(0,t)$ and $v(1,t)$ for $t > 0$
10. **Measurements:** Simulated at 10 time coordinates 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 1, 2, 3, 4

6.15 HOT_SPOT

1. **Background:** Single step reaction from combustion theory with steep temperature front lines, see Adjerid and Flaherty[1] and Verwer e.al.[42].
2. **Background:** The two parameters in the right-hand side of the PDE to be estimated, can be identified correctly by all optimization algorithms.
3. **System equations:**

$$u_t(x,t) = u_{xx}(x,t) + \frac{r\exp(\delta)}{a\delta}(1 + a - u(x,t))\exp(-\delta/u(x,t))$$

   for $0 < t \leq 0.3$ and $0 < x < 1$,
4. **Initial conditions:** $u(x,0) = 1$ and $v(x,0) = 0$ for all x
5. **Boundary conditions:** $u_x(0,t) = 0$, $u(1,t) = 1$ for all $t > 0$

6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
a	5.0	0.999991	0.999991	0.999991	0.999993	1.0
δ	1.0	20.05	20.05	20.05	20.03	20.0
r	1.0	4.998	4.998	4.998	4.998	5.0
res	1.8	0.57E-4	0.58E-4	0.58E-4	0.58E-4	0.61E-4

9. **Fitting criterion:** $u(0,t)$ for $t > 0$
10. **Measurements:** Simulated at 11 time coordinates 0.05, 0.1, 0.15, 0.2, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29

6.16 WAVES

1. **Background:** Waves travelling in opposite directions in form of a semi-hyperbolic system, see Madsen[25] and Verwer e.al.[42].
2. **Remarks:** Initial conditions are non-continuous and depend on parameters to be estimated. They can be estimated correctly, only NLSNIP broke down because of instabilities during integration.
3. **System equations:**

$$\begin{aligned} u_t(x,t) &= -u_x(x,t) - au(x,t)v(x,t) \\ v_t(x,t) &= v_x(x,t) - au(x,t)v(x,t) \end{aligned}$$

for $t > 0$ and $-0.5 < x < 0.5$

4. **Initial conditions:**

$$u(x,0) = \begin{cases} b(1+\cos(c\pi x)) & \text{for } -0.3 \le x \le -0.1 \\ 0 & \text{otherwise} \end{cases}$$

$$v(x,0) = \begin{cases} b(1+\cos(c\pi x)) & \text{for } 0.1 \le x \le 0.3 \\ 0 & \text{otherwise} \end{cases}$$

5. **Boundary conditions:** $u(-0.5,t) = 0$, $u(0.5,t) = 0$, $v(-0.5,t) = 0$, $v(0.5,t) = 0$ for all $t > 0$
6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
a	70.0	100.08	100.08	-	100.08	100.0
b	1.0	0.5001	0.5001	-	0.5001	0.5
c	5.0	10.001	10.001	-	10.002	10.0
res	8.4	0.35E-4	0.35E-4	-	0.35E-4	0.51E-4

9. **Fitting criterion:** $u(x_i,t)$ and $v(x_i,t)$ for $t > 0$ with $x_1 = -0.2$, $x_2 = -0.1$, $x_3 = 0$, $x_4 = 0.1$ and $x_5 = 0.2$
10. **Measurements:** Simulated at 5 time coordinates 0.1, 0.15, 0.2, 0.25, 0.35

6.17 FLAME

1. **Background:** Flame propagation with fast moving front, see Dwyer, Sanders[10] and Verwer e.al.[42].
2. **Remarks:** The fast moving front does not impose major numerical difficulties. The model equation is integrated w.r.t. a very crude spatial discretization.
3. **System equations:**

$$\begin{aligned} u_t(x,t) &= u_{xx}(x,t) - u(x,t)f(v(x,t)) \\ v_t(x,t) &= v_{xx}(x,t) + u(x,t)f(v(x,t)) \end{aligned}$$

   for $t > 0$ and $0 < x < 1$, where $f(v) = 1.0E + 6a\exp(-4/v)$
4. **Initial conditions:** $u(x,0) = 1$, $v(x,0) = 0.2$ for $0 \le x \le 1$
5. **Boundary conditions:** $u_x(0,t) = 0$, $u_x(1,t) = 0$, $v_x(0,t) = 0$ for all $t > 0$, and $v(1,t) = 0.2 + t/b$ for all $t \le b$ and $v(1,t) = 1.2$ otherwise
6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
a	1.0	3.513	3.519	3.519	3.425	3.52
b	0.001	0.000198	0.000199	0.000199	0.000193	0.0002
res	0.61	0.68E-6	0.66E-6	0.66E-6	0.20E-5	0.67E-6

9. **Fitting criterion:** $u(x_i,t)$ for $t > 0$ with $x_i = 0.6, 0.7, \ldots 1$, $i = 1, \ldots 5$
10. **Measurements:** Simulated at 3 time coordinates 0.002, 0.004, 0.006

6.18 BRUSSEL

1. **Background:** Multi-molecular reaction with diffusion, see Flaherty and Moore[13] and Hairer, Nørsett and Wanner[16] (brusselator).
2. **Remarks:** A large number of measurements was simulated and spread over the whole two-dimensional integration interval. The parameters estimation problem is well-posed, i.e. parameters can be identified correctly by all algorithms. However the calculation times are quite large.
3. **System equations:**

$$\begin{aligned} u_t(x,t) &= \epsilon u_{xx}(x,t) + u(x,t)^2 v(x,t) - au(x,t) + 1 \\ v_t(x,t) &= \epsilon v_{xx}(x,t) - u(x,t)^2 v(x,t) + bu(x,t) \end{aligned}$$

   for $t > 0$ and $0 < x < 1$, where $\epsilon = 0.002$

4. **Initial conditions:** $u(x,0) = 0$,

$$v(x,0) = 1 + 5x - 0.25\tanh(cx) - 0.25\tanh(c(1-x))$$

for $0 \le x \le 1$

5. **Boundary conditions:** $u_x(0,t) = 0$, $u_x(1,t) = 0$, $v_x(0,t) = 0$ and $v_x(1,t) = 0$ for all $t > 0$
6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
a	3.0	4.4002	4.4002	4.4002	4.4002	4.4
b	3.0	3.4004	3.4004	3.4004	3.4004	3.4
c	10.0	19.92	19.91	19.91	19.91	20.0
res	0.69E+4	0.52E-3	0.52E-3	0.52E-3	0.52E-3	0.59E-3

9. **Fitting criterion:** $u(x_i,t)$ and $v(x_i,t)$ for $t > 0$ with $x_i = 0.2, 0.4, 0.6, 0.8, 1$, $i = 1, \ldots, 5$
10. **Measurements:** Simulated at 12 time coordinates 1, 2, ..., 12

6.19 POLLUTN

1. **Background:** Pollution effects of supersonic transports in stratosphere, see Sincovec and Madsen[37] or MacCracken[26].
2. **Remarks:** The numerical integration is very time-consuming, since the model equation is highly stiff and the integration interval is very large. Also in this case, we start the ODE-solver with default solution tolerances, e.g. an initial steplength of 1.0E-4.
3. **System equations:**

$$\begin{aligned}
u_t(x,t) &= Du_{xx}(x,t) + k_1 - k_2u(x,t) + k_3v(x,t) + k_4y(x,t) \\
&\quad -k_5u(x,t)v(x,t) - k_6u(x,t)y(x,t) \\
v_t(x,t) &= Dv_{xx}(x,t) + k_2u(x,t) - k_3v(x,t) + k_5u(x,t)v(x,t) \\
&\quad -k_7v(x,t)w(x,t) \\
w_t(x,t) &= Dw_{xx}(x,t) - k_8w(x,t) + k_4y(x,t) + k_6u(x,t)y(x,t) \\
&\quad -k_7v(x,t)w(x,t) + 800 + s(x) \\
y_t(x,t) &= Dy_{xx}(x,t) - k_4y(x,t) + k_7v(x,t)w(x,t) - k_6u(x,t)y(x,t) \\
&\quad +800
\end{aligned}$$

for $0 \le x \le 1$, $t \ge 0$, where $D = 1.0E-9$, and $s(x) = 3250$, if $0.475 \le x \le 0.575$, and $s(x) = 360$ otherwise.

4. **Initial conditions:** $u(x,0) = 1.306028E{+}6$, $v(x,0) = 1.076508E{+}12$, $w(x,0) = 6.457715E+10$, $y(x,0) = 3.542285E+10$ for $0 \le x \le 1$

5. **Boundary conditions:** $u_x(x_b,t)=0$, $v_x(x_b,t)=0$, $w_x(x_b,t)=0$, $y_x(x_b,t)=0$ for $t>0$ and $x_b=0$ or $x_b=1$, respectively
6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
k_1	1.0E+5	4.478E+5	4.916E+5	4.929E+5	2.536E+5	4.0000E+5
k_2	1.0E+2	0.739E+2	0.122E+2	0.059E+2	0.752E+2	2.7244E+2
k_3	1.0E-4	0.269E-4	0.044E-4	0.021E-4	0.279E-4	1.0002E-4
k_4	1.0E-3	1.902E-3	0.310E-3	0.146E-3	1.929E-3	7.0000E-3
k_5	1.0E-16	1.128E-16	0.368E-16	0.538E-16	2.410E-16	3.6700E-16
k_6	1.0E-12	4.621E-12	5.070E-12	5.082E-12	2.631E-12	4.1300E-12
k_7	1.0E-15	0.974E-15	0.161E-15	0.078E-15	0.988E-15	3.5700E-15
k_8	1.0E-8	1.605E-8	1.604E-8	1.605E-8	1.609E-8	1.6000E-8
res	0.52	0.60E-5	0.50E-5	0.55E-5	0.46E-4	0.17E-4

9. **Fitting criteria:** $u(1,t)$, $v(1,t)$, $w(1,t)$ and $y(1,t)$ for $t>0$
10. **Measurements:** Simulated at 7 time coordinates 1.0E+6, 5.0E+6, 1.0E+7, 5.0E+7, 1.0E+8, 5.0E+8, 1.0E+9

6.20 INTEG

1. **Background:** First-order population dynamics governed by an integro-differential equation, see Pennington and Berzins[29] or Fairweather and Lopez-Marcos[11], with exact boundary condition.
2. **Remarks:** The intention is to show that also problems with integrals over the spatial interval can be identified correctly. The resulting parameter estimation problem is quite stable, all optimization algorithms compute the optimal parameters correctly.
3. **System equations:**

$$u_t(x,t) = -b_1 u_x(x,t) - b_2 u(x,t) \int_0^a u(z,t)dz$$

for $0<t\le a$, $0<x\le a$

4. **Initial conditions:**

$$u(x,0) = \frac{\exp(-x)}{2-\exp(-a)}$$

for $0\le x\le a$

5. **Boundary conditions:**

$$u(0,t) = \frac{1}{1-\exp(-a)+\exp(-t)}$$

for $0 \leq t \leq a$

6. **Coupled ODE's:** none
7. **Transmission conditions:** none
8. **Estimated parameters:**

p	p^0	p^{DFNLP}	p^{DN2GB}	p^{NLSNIP}	p^{DSLMDF}	$p^\star$
b_1	0.01	1.003	1.004	1.004	1.004	1.0
b_2	10.0	1.004	1.005	1.005	1.005	1.0
b_3	10.0	1.0004	1.00002	1.00002	0.9997	1.0
res	4.8	0.46E-5	0.45E-5	0.45E-5	0.45E-5	0.67E-5

9. **Fitting criterion:** $u(i,t)$ for $0 \leq t \leq a$ and $i = 1, \ldots, 5$
10. **Measurements:** Simulated at time coordinates 1, 2, 3, 4, 5