Parameter identification (PI) techniques have found a rapidly increasing interest in the past years, particularly in biology, chemistry and other sciences, where a quantitative description of a complex process can often not be derived from investigations into isolated subsystems by in vitro experiments, and observations of the complete process in vivo are the only source of information. At the same time, the development of conceptionally new numerical methods has promoted substantial improvements in the treatment of such problems.

The aim of the present paper is to review the essential lines of development of a novel class of PI algorithms and their realization in the multiple shooting code PARFIT [1] and the collocation code COLFIT [2], the common base of which is a Generalized Gauss Newton method for discretized PI boundary value problems. Several features of the implementation, that are considered important for the efficiency, stability and applicability, particularly of PARFIT, are presented.

1. Class of Problems

Both PARFIT and COLFIT are designed for the identification of \( n_p \) parameters \( p \) in a nonlinear model o.d.e. system for the \( n_x \) states \( x(t) \)

\[
\begin{align*}
\dot{x}(t) &= f(t,x,p) + T(t,x,p;Q(t,x,p)) \\
x(t_r) &= x(t_r) + C(t,x(t_r),p) \quad \text{for some } Q_j(t,x(t_r),p) = 0,
\end{align*}
\]

(1)

the right hand side and solution of which may be discontinuous depending on sign changes of the switching functions \( Q \). Observations are given in terms of functions of states and parameters at data points \( t_j \)

\[
n_{ij} = q_i(t_j,x(t_j),p) + e_{ij}
\]

(2)
subject to measurement errors \(c_{ij}\), which are (for ease of presentation) assumed to be independent, normally distributed with zero mean and known variances \((N(0, \sigma_{ij}^2))\). Thus, by minimizing a weighted least squares function

\[
\min_{x,p} \frac{1}{2} \sum_{t=2}^{n} \sigma_{ij}^2 (n_{ij} - g_{i}(t_j, x(t_j), p))^2
\]

a maximum likelihood estimate is obtained. Additionally, side conditions are often given to specify further model properties such as boundary conditions, initial values or parameter restrictions

\[
\begin{align*}
    r_2(x(t_1), \ldots, x(t_n), p) &= 0 \\
    r_3(x(t_1), \ldots, x(t_n), p) &\geq 0,
\end{align*}
\]

whereas a priori information subject to error is included in (3). Note that the switching function formulation (1) allows a convenient treatment of variable data points which are only implicitly given or subject to measurement error.

Although this problem class meets most practical requirements, there are certain extensions which require special attention.

Multiexperiment problems

Frequently, the available data arise from several experiments under different conditions, and the PI problem consists of several independent problems which have possibly only the parameters \(p\) in common.

The structure of such problems is analyzed in [3]. An enhanced version of the multiple shooting procedure PARFIT incorporating the methods developed in [3] is currently being developed, which was used for the numerical solution of two problems in section 8. An important application is e.g. the identification of geophysical structures by the tracing of seismic rays (REINERS [4], using the model of PEREYRA, KELLER and LEE [5]).

(Semi-)Infinite problems

In a number of cases, functions space norms are more appropriate or convenient instead of the discrete norm (3)

\[
\int_0^t \|r_c^s(t, x(t), p)\|^2 dt = \min
\]

sometimes with the additional feature that functions \(u(t)\) must be identified

\[
\dot{x}(t) = f(t, x(t), p, u(t))
\]

In such cases, one may resort to optimal control methods, which in this form can be immediately treated by the algorithm described e.g. in [6], but it should be emphasized that the generalized Gauss-Newton method described below has a natural extension to the (semi-)infinite case [7]. The algorithm of NOWAK, DEUFLHARD [8] is based on the latter approach.

8. The Initial Value Problem Approach

The most intuitive and probably most widespread approach to PI in o.d.e. is the repeated solution of the initial value problem in some kind of iterative procedure to improve the fit. A well known representative of this class of IVP methods is the algorithm of MILSTEIN [9,10], which uses GEAR'S [11] integrator and a refined random search technique [BREMMERMANN [12]].

The IVP approach has two major drawbacks, which are apparent in numerical practice. Firstly, this reinvention of the inverse problem eliminates the states by means of (1) in favour of the parameters, and thus neglects any information on the states which is at hand in an inverse problem. There is rather impressive numerical evidence about the deterioration of efficiency thus caused. Secondly, the elimination of the states can cause a substantial loss of stability for the solution scheme. At least for bad initial guesses of the parameters, the "direct" (nonlinear) IVP may be ill-conditioned and hard to solve, or a solution may not even exist, even when the original PI problem is perfectly well conditioned. To illustrate the latter drawback, we consider

A notorious test problem

which is a modification of a two point boundary value problem described by BULIRSCH [13]. A model o.d.e. for two states and one unknown parameter \(p\) with fixed initial values is given by
\( \dot{x}_1 = x_2 \quad , \quad x_1(0) = 0 \)
\( \dot{x}_2 = \mu x_1 - (\mu^2 + p^2) \sin x_2 \quad , \quad x_2(0) = \pi \quad t \in [0,1] \)  
(7)

The solution of the true parameter value \( p = \pi \) is

\( x_1(t) = \sin \pi t \quad , \quad x_2(t) = \pi \cos \pi t \).
(8)

Measurements are generated by adding pseudo random noise \( (N(0,0.2), \alpha = 0.05) \) at selected data points \( t_j = j \cdot 0.1 \) \( (j=1,\ldots,10) \). The results for single shooting as shown in Fig. 1 are disastrous. Choosing \( p_0 = 1 \), as a starting guess, the integration routine \( \text{DIFFYSYS} \) \( (14) \) terminates at \( t \approx 0.23 \) for \( \mu = 60 \). But even for the true value of \( p \) (correct up to 16 decimals) and highest integration accuracy the solution is properly reproduced only on the first half of the interval.

![Graph](image)

**Fig. 1** Notorious test problem - single shooting \( (\mu = 60) \)

3. The Multiple Shooting Approach to PI

An alternative to the IVP approach is derived by regarding the PI problem \( (1-4) \) as a constrained, overdetermined multipoint boundary value problem - independent of whether the "direct" problem is a BVP or not - as in the PI package PARFIT \( (1,7) \) which is based on an earlier multipoint BVP algorithm of the author \( (6) \) using the multiple shooting technique \( (cf.13,15,16,17,18) \).

This algorithm is briefly summarized. Using a suitable mesh \( \tau \)

\[ \tau^m \ni \tau_1 < \tau_2 < \ldots < \tau_m \quad , \quad \Delta \tau_j := \tau_{j+1} - \tau_j \quad (j=1,\ldots,m-1) \]  
(9)

e.g. a subset of the data points, one computes the solutions \( x(t;\tau_j,p) \) of \( m-1 \) independent IVP

\( \dot{x} = f(t;x,p) \quad , \quad x(\tau_j) = s_j \quad , \quad t \in [\tau_j,\tau_{j+1}] \quad \text{a.e.} \)  
(10)

The additional variables \( (s_1,\ldots,s_m) \) are estimates of the states \( x(\tau_j) \). Formal insertion of this (discontinuous) parametrization of the solution into the original PI problem yields a large, constrained least squares problem for which adequate methods are outlined in the following sections.

**Problem PII:**

\[ \|x_1(s_1,\ldots,s_m,p)\|_2 = \min \quad (s_1,\ldots,s_m,p) \]  
(11)

subject to \( R_2(s_1,\ldots,s_m,p) = 0 \) , \( R_3(s_1,\ldots,s_m,p) \geq 0 \)

with the additional matching conditions

\[ h_j(s_{j+1},s_j,p) := x(\tau_{j+1},s_j,p) - s_j = 0 \]  
(12)

to ensure continuity of the solution.

**Remark:** One obvious advantage of multiple shooting is that any a priori information about the solution trajectory can be utilized to improve the initial guess. Hence, the influence of poor estimates for the parameters is substantially reduced.

**Mesh selection**

The basic idea of multiple shooting is that by choosing a sufficiently fine mesh one can avoid drifting too far away from the solution trajectory \( x(t) \). Quantitatively accessible bounds that do not overestimate the growth behaviour (like global Lipschitz constants) can be derived by a first order perturbation analysis. Defining the propagation matrices by the variational differential equations (VDE)

\[ d(\Phi^P)(t,\tau)/dt = f_x(t,x(t),p)(\Phi^P) + (0 f_p(t,x(t),p)) \]

\[ (\Phi^P)(t,\tau) = (1 0) \]

An error \( \delta_x, \delta x(t) \) at time \( \tau \) is propagated according to...
If $e_j$ is the integration tolerance used, and $e_p$ the accuracy of the parameter, a mesh can be called sufficiently fine if the mesh condition (MC):

$$\gamma_j e_j + e^P P T_j \leq TOL_j$$  

holds, where $TOL_j$ is the required accuracy of the solution $x(t)$ (to be selected carefully, see section 71).

The propagation factors

$$\gamma_j := \max_{t_1, t_2 \in I_j} \|\phi(t_1, t_2)\| \geq 1$$  

$$\gamma^P := \max_{t_1, t_2 \in I_j} \|\phi^P(t_1, t_2)\| \geq 0$$  

are approximated in PARFIT by means of $\phi^P(\tau_{j+1}, \tau_j)$, which is computed anyway.

Remark 1: The mesh condition (MC) is essentially a stability criterion and usually allows very coarse meshes if once a mesh is fixed, the accuracy can be adapted to satisfy (MC). Actually, PARFIT uses a refined strategy which adapts the accuracy componentwise. If a limit accuracy is reached, the mesh must be refined (or $TOL_j$ relaxed). It may be noteworthy that (MC) is also a valuable tool for the automatic mesh selection similar to that used in COLFIT [2], cf. sect. 4.

Remark 2: Throughout this paper, all norms are understood to be scaled due to an internal scaling of the variables in order to equilibrate them (such that $\|s_j\| \leq 1$, $\|p\| \leq 1$). Consequently, all error criteria are absolute (but "pseudoretative") which considerably simplifies calculations.

Remark 3: In case of the test problem (7) one shows that for single shooting $\gamma_j r_j = 10^{-2}$, but $= 10^4$ for multiple shooting with a uniform mesh of $\Delta t = 0.1$ (cf. table 1, fig. 3).

4. Alternative BVP Approaches

The basic ideas of PARFIT can be applied to other BVP approaches as well. Since most methods used in practice lead to a discretized or parameterized differential equation

$$\delta x(t) = \phi(t, x) \delta x(t) + \phi^P(t, x) \delta p$$  

the insertion of the discretized $x^h$ into the PI problem yields a constrained least squares BVP analogous to problem P11. This includes in particular alternate multiple shooting as in [19], finite differences with iterative deferred corrections [20] and collocation [21]. As a representative we shall sketch the collocation approach (cf. [21,22,23,24,26]) which was realized in the code COLFIT in cooperation with SCHLÖDER and BNR [2] (cf. [26]).

For a given mesh $\tau^m$, an additional collocation mesh $\rho = \{\tau^1_j\}$

$$\rho: \tau^1_j := \tau_j + c_j \Delta t_j : 0 \leq c_j < \cdots < c_j \leq 1 \quad (j=1, \ldots, m)$$  

is defined, and from the class $P(1, \tau) := \{y | y$ is polynomial of degree $\leq 1$ on $[\tau_j, \tau_{j+1}]$ an approximation of the o.d.e. solution is determined by collocation on $\rho$, i.e. choosing $y \in P(1, \tau) \cap C[\tau_1, \tau_m]$ such that

$$\hat{y}(t) = f(t, y(t), \rho) \quad \text{for all} \quad t \in \rho.$$  

Technical choices of $\{c_j\}$ are the GAUSS, RADAU or LOBATTO points on $[0,1]$. In COLFIT, LOBATTO-collocation is used, which is especially suitable for singular perturbation problems - see the analysis of ASCHER, WEISS [27] using, e.g., piecewise cubic polynomials in Hermite representation - conditions (19) read

$$s_{j+1} = s_j + \frac{\Delta t_j}{8}(s_j^g + s_j^g + 4f(\tau_j s_j^g + s_j^g + s_j^g + s_j^g + s_j^g)) + \frac{\Delta t_j^g}{8}(s_j^g - s_j^g) + p = 0$$  

$$s_j^g = f(\tau_j s_j^g)$$  

which may be interpreted (insert (20b) into (20a)) as a symmetric A-stable implicit Runge-Kutta discretization

$$s_j^g + \Delta t_j^g (f(s_j^g s_j^g + s_j^g s_j^g) - s_j^g) = h_j^g (s_j^g s_j^g s_j^g).$$  

Note, that (19) can always be represented this way (cf. [24]). Collocation thus yields a discretized PI-BVP of the same type as in multiple shooting, with the matching conditions replaced by (19) or (21) resp.
Mesh selection

Firstly, in order to retain the usual superconvergence properties, and for computational simplification, all data points should belong to the basic mesh \( h^m \). Secondly, the mesh must be sufficiently fine to guarantee the required accuracy. Since the local truncation error \( e_j \) of \( \Psi_j \) is of order \( p = 4 \) a mesh condition for collocation is given by

\[
\Delta x_j e_j = O(\Delta x_j^{p+4}) \leq \text{TOL}_j
\]  

(22)

which is used in COLFIT for mesh selection. Since (21) may be viewed as a special multiple shooting scheme with a one step integration, the number of meshpoints required by the accuracy criterion (22) is typically much higher than that of multiple shooting. Note, however, that the stability condition [MC] must hold here, too.

Remark 1: Since global and local error are of (fixed) order 4 for cubic Lobatto collocation, the method can be effective only for moderate accuracy requirements. The flexibility and efficiency could however be greatly increased by extrapolation or deferred corrections based on the error expansions in even powers of \( \Delta x_j \).

Remark 2: Discontinuities due to switching conditions in (1) are treated in [2] by a modification of the variable meshpoint technique of [28].


The constrained approximation problem P11 is solved in PARFIT and COLFIT by a generalized Gauss-Newton (GGN) method, which appears to be both natural and particularly effective for the considered problem.

Problem P12: \[ \|F_1(x)\|_2 = \min \]

subject to \[ F_2(x) = 0 \quad \text{and} \quad F_3(x) \geq 0. \]

A given iterate \( x_k \) is improved by

\[ x_{k+1} = x_k + \lambda_k \Delta x_k \quad (0 < \lambda_{\min} \leq \lambda_k \leq 1) \]  

(23)

where \( \Delta x_k \) is the solution of the linearized system

Problem P13: \[ \|F_{1k} + J_{1k} \Delta x_k\|_2 = \min \quad (J_{1k} = F_1' (x_k) F_1 = F) \]

subject to \[ F_{2k} + J_{2k} \Delta x_k = 0 \quad \text{and} \quad F_{3k} + J_{3k} \Delta x_k \geq 0 \]

Procedures of this type are considered e.g. in [1, 3, 7, 29, 30, 31]. We briefly summarize some basic results and strategies.

Local convergence: Since under mild regularity assumptions the active inequality constraints of P13 remain unchanged in a solution neighbourhood [7], only the equality constrained case needs to be treated. Defining the composite function \( F^* = (F_1^* F_2^*)^T, J = F^* \), one easily shows that a solution of P13 is given in terms of a generalized inverse \( J^* \),

\[ \Delta x_k = -J_k^* F_k \]  

(24)

If in a region D where \( F \in C^1(D) \)

\[ \|J(y)^T (J(y + t(x-y)) - J(x))(x-y)\| \leq \omega t \|x-y\|^2 \quad \omega < \infty \]  

(25a)

\[ \|J(y)^T - J(x)^T\| R(x) \leq \kappa(x) t \|y-x\| \quad \kappa(x) \leq 1 \quad R(x) = (I - J(x) J(x)^T)F(x) \]  

(25b)

an initial guess \( x_0 \) is given with \( \alpha_0 := \|J(x_0)^* F(x_0)\| \)

\[ \delta_0 := \kappa + \frac{\omega}{2} \alpha_0 < 1 \quad (x, 0, \delta_0, \kappa, \delta_0) \subset D \]  

(26)

the full step GGN method is well-defined and converges to a stationary point \( x^* \), which is a strict local minimum if \( \lambda_{\min} J \) have full rank. The final convergence rate is \( \kappa(x^*) \).

Remark: \( \omega \) and \( \kappa \) are two fundamental (weighted) Lipschitz constants, which characterize the nonlinearity of the model \( \omega \) and the incomparability of the data with the model \( \kappa \). In statistically well-posed problems, \( \kappa \) is small, and the GGN method essentially has the excellent convergence behaviour of a second order method although it requires only first order information.

Global convergence: In order to extend the convergence domain, the
damping factor $\lambda_k$ is monitored by the natural monotonicity test (cf. [1,32,33])

$$
\|\Delta x_{k+1}(\lambda_k)\| = \frac{\nu_k(x_k, \lambda_k) - \nu_k(x_k, \lambda_k)}{\nu_k(x_k, \lambda_k)} \leq \|\Delta x_k\| (\lambda \in [0,1])
$$

For this (local) level function, a simplified line search can be derived from the estimate

$$
\omega_k(\lambda) := \max_{\lambda \in [0,1]} \frac{\|\Delta x_k(\lambda, \lambda)\|}{\|\Delta x_k\|}.
$$

An optimal damping factor in terms of this upper bound is

$$
\lambda^*_k = \min \left(1, \frac{1}{\omega_k(x_k)} \|\Delta x_k\| \right) \quad (\omega_k \text{ mon. increasing})
$$

$\omega_k(\lambda)$ can be safely estimated by the a posteriori formula (cf. [1,7])

$$
\omega_k(\lambda) \approx \frac{2\|\Delta x_k(\lambda)\| - (1 - \lambda)\|\Delta x_k\|}{\lambda^2 \|\Delta x_k\|^2} = \omega_k(\lambda) + O(\lambda)
$$

This estimate is applied in two ways: $\omega_k(\lambda)$ of the last iteration is used as a predictor by means of (29). If the monotonicity test fails, it is used to abort the step length. This procedure is easily shown to be feasible.

Remark 1: The damping strategy (29) is also "optimal" in the sense, that the resulting (nearly constant) step length $\omega_k^{-1}$ characterizes the radius of the domain in which the linearization is valid, and thus optimally exploits the computed derivative information.

Remark 2: Global convergence for the damped GN method can be established by means of an $\mathcal{H}_2$-space level function [1,7]. Although a similar proof for the damping strategy described here is still an open question, its use is strongly emphasized by numerical experience.

6. Solution of Linear Subproblems

The linearization of the discrete problem P11 yields a total Jacobian of a special structure

$$
J = \begin{bmatrix}
P_1 & 0 & \cdots & 0 \\
D_2 & P_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
D_n & P_n & \cdots & P_n
\end{bmatrix}
$$

$$
F = \begin{bmatrix}
\nu_{11} & \nu_{12} & \cdots & \nu_{1n} \\
\nu_{21} & \nu_{22} & \cdots & \nu_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\nu_{n1} & \nu_{n2} & \cdots & \nu_{nn}
\end{bmatrix}
$$

The Conding Algorithm

For multiple shooting, one has $G_j^0 = -I$, $G_j^0 = G_j = \#(\tau_j, \tau_j, \tau_j)$, $G_j^0 = G_j = \#(\tau_j, \tau_j, \tau_j)$. Using the matrices $G_j^0$ as pivots for a block-Gaussian elimination, the backward recursion

$$
u_j = R_j, \quad P_j = D_j^T, \quad E_j = D_j^T \quad (1=1,2,3)
$$

$$
u_j = D_j + E_j \quad (J=1,2,3)
$$

transforms the system (31) equivalently to the condensed problem for initial values and parameters:

$$
C \hat{u}_1 + E_1 \hat{s}_1 + P_1 \hat{a}_1 = 0
$$

s.t. $u_2 + E_2 \hat{s}_2 + P_2 \hat{a}_2 = 0$ and $u_3 + E_3 \hat{s}_3 + P_3 \hat{a}_3 = 0$

which is of much smaller dimension. Having solved (33), the other increments can be computed by the forward recursion

$$
s_{j+1} = G_j \hat{s}_j + G_j \hat{a}_j + h_j \quad (j=1,\ldots,n-1)
$$

For collocation, the same algorithm applies with the replacements

$$
(G_j^0)^{-1}G_j = G_j, \quad (G_j^0)^{-1}P_j = P_j, \quad (G_j^0)^{-1}h_j = h_j
$$

Since nonsingularity of $G_j^0$ is an essential stability requirement for collocation anyway (cf. [24]), this procedure is feasible. Note, that
\begin{equation}
(\delta_j^2)^{1/2} \tilde{z}_j + \delta_j \tilde{T}_j \tau_j + O(h^{n+1}) + (\delta_j^2)^{-1/2} \tilde{p}^2 \tilde{T}_j \tau_j + O(h^{n+1}) \tag{36}
\end{equation}

Solution of the condensed system

For solution of the condensed system, several methods are available (cf. [1, 29, 34]). Efficient techniques for the multirexperiment case are developed in [3].

"Real life" PI-problems, like other inverse problems, are frequently singular or ill-conditioned, when the model is overparametrized (non-unique) with respect to the available experimental data, so that regularisations are needed. Since the backward recursion leaves the rank unaltered, a treatment of the condensed system is sufficient. As all algorithms basically depend on triangularisation by QR decomposition (BUSINGER, GOLUB [35]), pseudoinversion can be applied after appropriate rank selection (DEUFLEHARD, SAUTTER [36]):

Remark: Note that in case of rank-deficiency, the condensed system allows a complete (linear) analysis of redundancies in the parameter space.

Statistical a posteriori analysis of solution

Parameters obtained by a PI procedure are useful only if also some estimate of their statistical reliability is given. The GGM method in combination with the condensing algorithm used in PARTFIT and COLFIT makes this particularly easy. The covariances of the parameters (and the initial values, if they are not fixed) can be computed cheaply and easily from the decomposed condensed system, and from this, first-order approximations to individual confidence intervals can be derived ([1, 7], see SCHLOEDER [3] for the multirexperiment case).

However, more statistical aids are desirable, and presently under consideration.

Conditioning of PI-boundary value problems

With the definitions (considering only constraints active at the solution

\begin{equation}
B_1 = \sum_{j=1}^{n-1} D^j \phi(\tau_j \tau_{j+1}) \times A_1 = \sum_{j=1}^{n-1} D^j \phi(\tau_j \tau_{j+1}) \times E_1 = A_1 + B_1 \times P_1 = \sum_{j=1}^{n-1} A_j G_{j+1}^2 + E_j \times S_1 = (E_1 P_1)^+ \tag{37}
\end{equation}

an explicit representation of the generalized inverse \( J^+ \)

\begin{equation}
(\delta_m)^{1/2} \tilde{z}_j + \delta_j \tilde{T}_j \tau_j + O(h^{n+1}) \tag{38}
\end{equation}

is obtained by means of the condensing algorithm. With the assumptions

\begin{equation}
\| S_1 \| \leq c_1 \leq K_1 < \infty, \quad \| S_1 A_j \| \leq e_{1j} \leq K_2 < \infty, \quad \| S_1 B_j \| \leq b_{1j} \leq K_3 < \infty \tag{39}
\end{equation}

one readily verifies the upper bound for \( | \delta_j^2 | \) (cf. [7, 37, 38]).

Lemma 1: \( | \delta_j^2 | \leq c_j + \sum_{j=1}^{n-1} (a_{1j} + b_{1j}) \| \tilde{T}_j \| \leq K_{1j} + (n-1) K_3 < \infty \tag{40} \)

Asymptotic bounds for collocation are established analogously, making use of (36). Note that it is sufficient that \( c_{1j}, e_{1j}, b_{1j} \) be bounded for just one pair \((1,j)\) to establish the bounds \( K_1, K_2 \) independent of a specific mesh.

The dependence on the number of meshpoints can be avoided using \( 1/\tilde{T}_j \) as a weight for \( | \tilde{T}_j | \) (thus introducing an error-per-unit-interval criterion in the meshconditions (15) and (22)).

Stability of the condensing algorithm

There has been a lot of discussion on the stability of this type of recursive solver in the recent past. Since no pivoting is performed, it can be expected to be stable only if the o.d.e. system is stable and \( C_j^2 \) is "smaller" than \( C_j^2 \) in some sense. In general, however, the errors \( \delta_j^2 \equiv \tilde{T}_j - \tilde{T}_j \) of the increment \( \tilde{T}_j \) computed in the forward recursion, satisfying

\begin{equation}
\delta_j^2 = \delta_j^2 + \delta_j^2 \delta_j + \delta_j \tag{41}
\end{equation}

may be dominating the exact increments \( \delta_j \), since their growth behaviour is that of the possibly unstable IVP. The terms \( \delta_j \) incorporate the rounding errors made in the evaluation of (36), they are (elementwise) bounded by

\begin{equation}
| \delta_j | \leq \| G_j \| | \delta_j | | \tilde{z}_j | + \| G_j^2 \| | \tilde{p}^2 | | \tilde{T}_j | + | h_j | \varepsilon_{\text{mach}} \tag{42}
\end{equation}
For the sake of stability, alternative linear system solvers are often advocated, such as the algorithms of VARAH [39] or de BOOR, WEISS [40], which perform some kind of pivoting but try to utilize the structure for computational and storage economy. (A PI variant of [39] was designed by BR for use in COULFIT [21].)

On the other hand, the condensing algorithm has been successfully used for many sensitive real life problems including PI. Its computing and storage efficiency is outstanding, and it offers significant analytical and statistical insight by means of the condensed system. Hence, the development of modifications that maintain the efficiency of the algorithm under suitable stability assumptions is a challenging problem.

The SHAVING algorithm makes SURE

The SHAVING algorithm was developed in cooperation with KRAMER-EIS after inspiring discussions with B. RUSSELL and B. MATHEIJN. It is a special iterative refinement procedure similar to the one developed independently by DEUFLHARD and BADER [38], who also give a detailed rounding error analysis. Therefore, we may restrict the presentation of our procedure to the main lines.

The basic idea of SHAVING (combined with Successive Refinement) is to perform the forward recursion only as long as the defect of the right hand side

\[ \tilde{h}_j := h_j - G_j \tilde{\delta}_j - G_j \tilde{\delta}_p + \tilde{\delta}_j \]

(43)

can be considered to be zero. This defect is actually a zero, if computed numerically, but it is known to be exactly \( \delta_j \) of (41). From (42), one may thus conclude, that the contribution of the parasitic terms \( \delta_j \) is small, if \( \delta_j \) is small compared to \( |\Delta s_j| \), or \( |\Delta s_j| \), resp.

\[ |\delta_j| \leq \alpha |\Delta s_j| \]  \hspace{1cm} (\alpha < 1).

(44)

An upper bound for \( |\delta_j| \) can be computed a priori by the recursion

\[ d_{j+1} = |G_j| (d_j + |\Delta s_j| \eta |p|) + |G_j| (|\delta_p| + |\Delta p| \eta |p|) + |h_j| \]

(45)

This criterion can however be relaxed and simplified taking into account the special nature of the right hand side. In view of the mesh condition

\[ \tilde{\eta}_j \]

can be considered small compared to the natural perturbation made in the evaluation of \( h_j \), if

\[ ||G_j|| ||\Delta s_j|| \eta |p| + ||G_j|| ||\Delta p|| \eta |p| \leq \delta \cdot \text{TO}_j \]

\( \delta \ll 1 \).

(46)

If (46) is violated, the forward recursion is stopped. Only the defect of this index, \( \tilde{\eta}_j \), has to be computed, whereas the previous values are set to zero - they are considered to be "shaven". Repeated application of the thus modified condensing algorithm (backward recursion for the right hand sides \( u_j \) only) successively refines the computed solution (and eventually removes the parasitic parts). The performance scheme of the algorithm is visualized in Fig. 2.

\[ \log (\Delta s_j) \]

\[ \Delta s_j^{(0)} \quad \Delta s_j^{(1)} \quad \Delta s_j^{(2)} \]

Fig. 2 Performance scheme of the "shaving" algorithm

7. Evaluation of Pencil and Derivatives

The main bulk of computational labour in PARFIT arises from the numerical solution of the o.d.e., and the generation of their derivatives w.r.t. parameters and initial values. Therefore some remarks seem appropriate as to how this can be done most efficiently in a multiple shooting code.

7.1 Solution of IVPs

Two variable order and step extrapolation codes are the standard numerical integrators in PARFIT, for non-stiff systems DIFSYS (BULIRSCH, STÖER [14]) based on an explicit midpoint rule, and for stiff systems METAN1(BADER, DEUFLHARD [41]), based on a semi-implicit midpoint rule.

However, other integrators can be used as well, if the necessary inter-
faced to PARFIT are provided.

**Adaptive accuracy selection**

A lot of computational work can be saved if the integration tolerance is adapted to the requirements of the GGN process [1]. Although this seems straightforward, a computationally effective and safe implementation that works for highly nonlinear as well as for ill-conditioned problems is not trivial.

Up to now, PARFIT and its companion code for optimal control problems OPCON are the only multiple shooting procedures where this concept is realized.

An adaptive accuracy scheme can be derived from the following considerations (details to be presented elsewhere).

First, once an iterate $x_k$, say, is computed, the error $e_k$ of the increment $\Delta x_k$ should not be smaller than $\|\Delta x_{k+1}\|$ to be efficient

$$e_k \approx \|\Delta x_{k+1}\| = \|g_k\|\|x_{k+1}'\| \|\Delta x_k\|.$$  

(47)

Estimates of these convergence rates can be given in terms of the local convergence theorem.

The global accuracy $e_k$ can be achieved by lemma 1 choosing the local error tolerances $TOL_j$ according to

$$\|\Delta x_k^j\| \leq e_k.$$  

(48)

Outside the local convergence domain, condition (47) is replaced in PARFIT by the requirement, that the predictor-corrector strategy (29,30), on which the natural nonmonotonic test is based, be stable!

The efficiency of the overall scheme can also slightly be increased by storing additional information (about optimal initial step sizes and order, etc.) for the integrator to be used in subsequent iterations.

**7.2 Calculation of derivatives**

Being the most time consuming part of the calculations in PARFIT, this must be very efficiently realized. Secondly, if the algorithm shall be of use in practical applications, the chance of human error in these calculations must be strictly excluded.

**External Numerical Differentiation (END)**

A well known way to ensure the latter requirement is the use of a finite difference approximation $\Delta\theta$ to the derivative $\theta$

$$\Delta\theta_h(t;\cdot) := \eta^{-1}(x(t;\cdot + h\cdot) - x(t;\cdot)) - \Delta\theta(t;\cdot + \epsilon\cdot)$$  

(49)

where for abbreviation, $c := (s_j, p)$ and $d$ a specified direction. The derivatives can thus be computed by $(n_j + n_p + 1)$ full integrations with accuracy TOL varying one component of $c$ at a time ($d = e_i$).

Even for an optimal balance between discretization error (which is a discontinuous function of $c$ usually) and cancellation of leading digits the best accuracy $\epsilon$ one can expect is

$$\epsilon = O(\sqrt{TOL}) \quad \text{with} \quad \eta = O(\sqrt{TOL}).$$  

(50)

The efficiency of this scheme is low, since high accuracy integration is needed for low accuracy results. Moreover, for low tolerances as used in the adaptive accuracy scheme, it becomes totally unreliable.

**Internal Numerical Differentiation (IND)**

A remedy is provided by internal numerical differentiation as described in [1]. The basic idea is to compute the derivative of the (variable order and step) discretization scheme - thus getting rid of the TOL-bound - and saving the overhead spent in the integration of $(n_j + n_p + 1)$ nearly identical trajectories. We describe this procedure for the semi-implicit midpoint rule (141)

$$\Delta x_k := P f_0, \quad y_0 := y_0 + h\Delta x_0, \quad h := H/N \quad \text{(Start)}$$  

$$\Delta x_k := \Delta x_{k-1} + 2P(f_k - \Delta x_{k-1}), \quad y_{k+1} = y_k + h\Delta x_k \quad (k = 1, \ldots, N)$$  

(51)

where $P = (I - A)^{-1}$, $A \approx f_x(t, y_0, p)$, $f_j = f(t_j, y_j, x_j, p)$.

The final smoothing step

$$x_h(c) := \frac{1}{2}(y_{N+1} + y_{N+1})$$

then yields an approximation of the initial value problem solution $x(c)$ at $t = t_0 + H$. The discretization error has an asymptotic expansion in
even powers of \( h \), which permits the construction of higher order approximations \( T_{kj} \) by polynomial extrapolation from a sequence of values \( x_h(c) \).

\[
x_h(c) = x(c) + \sum_{i=1}^{M} a_i(c) h^{2i} + O(h^{2M+2})
\]

\[
T_{kj}(c) = x(c) + \sum_{i=1}^{M} a_i(c) h^{2i} + O(h^{2M+2}).
\]

(52)

As a consequence (under suitable assumptions) the finite difference approximation \( \Delta^n \) and the derivative \( \Delta^0 \) have an asymptotic expansion, too.

**Lemma 2:** For \( 0 \leq n < n_0 \):

\[
\Delta^n x_h(c) = \Delta^n x(c) + \sum_{i=1}^{M} a_i(c) h^{2i} + O(h^{2M+2})
\]

\[
\Delta^n T_{kj}(c) = \Delta^n x(c) + \sum_{i=1}^{M} a_i(c) h^{2i} + O(h^{2M+2})
\]

(53)

The proof involves the linearity of basic discretization and interpolation.

(53) allows a convenient control of the discretization error of \( \Delta^n T_{kj} \) (along with that of \( T_{kj} \)), which may serve as an additional stability check for the discretization scheme (51). It also indicates the accuracy bound due to cancellation for \( n>0 \).

As an interesting consequence of lemma 2, one has in comparison to \( \text{END} \) (49) (informally)

\[
e = 0(\text{TOL}) + 0(\sqrt{\frac{\varepsilon}{\text{mach}}}) \quad \text{for} \quad n = 0(\sqrt{\frac{\varepsilon}{\text{mach}}})
\]

(54)

i.e. the accuracy is of the same order as the accuracy TOL of the nominal trajectory.

The analytical limit of \( \text{IND} \)

If correct and cheap analytical derivatives of the right hand side are available, e.g. in case of chemical reaction systems, it can be economic (and it removes the \( 0(\sqrt{\frac{\varepsilon}{\text{mach}}}) \) bound in (54)) to perform the limit analytically and compute \( \Delta^n T_{kj}(c) \). It is easily shown that this is equivalent to the integration of the VDE (13) columnwise by the discretization scheme (51) using the same Padé-approximants, order and

stepsizes as in \( \text{IND} \) - thus reflecting the identical stability behaviour. Note, that as a consequence of lemma 2, forward difference approximation and analytical limit deliver identical results for moderate accuracies (say, \( \text{TOL} > 0(\sqrt{\frac{\varepsilon}{\text{mach}}}) \)).

Summing up, internal numerical differentiation leads to a drastic reduction of computing time (60 - 80 p.c. especially for low tolerances [1]) compared to external numerical differentiation because of substantial overhead savings (e.g. the Padé-approximants \( P(h) \) only have to be computed once for all trajectories), and much lower accuracy requirements for the basic discretization scheme.

Moreover, \( \text{IND} \) is stable especially for low tolerances, where most of the effort is spent in an adaptive accuracy scheme.

8. Numerical Results

This last section summarizes a few numerical results to illustrate some aspects of the performance of the algorithms. For details, and more complex problems, the reader is referred to [117]

Notorious test problem

For multiple shooting with \( \text{PARFIT} \) (using the 10 data points as mesh) the problem is well conditioned. Numerical results for \( 10 \leq \mu \leq 120 \) are summarized in table 8.1. Single shooting works for \( \mu = 10,20 \) only, since the propagation factor (row 4) is larger than \( 10^{19} \) for \( \mu \geq 40 \). With multiple shooting, the stability mesh condition (15) is satisfied for \( \text{TOL} = 10^{-10} \) and all \( \mu \) (row 3).

<table>
<thead>
<tr>
<th>( \mu )</th>
<th>10</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of iterations</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>min. loc. error ( y_{j_{\mu}} \text{mach} )</td>
<td>( 10^{-14} )</td>
<td>( 10^{-13} )</td>
<td>( 10^{-12} )</td>
<td>( 10^{-11} )</td>
<td>( 10^{-10} )</td>
<td>( 10^{-9} )</td>
<td>( 10^{-8} )</td>
</tr>
<tr>
<td>error propagation (</td>
<td></td>
<td>e(1,0)</td>
<td></td>
<td>)</td>
<td>( 10^5 )</td>
<td>( 10^{10} )</td>
<td>( 10^{19} )</td>
</tr>
<tr>
<td>No. of shaving steps</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1 Notorious test problem
Although the initial trajectory for $\mu = 60$ looks rather weird (Fig. 3), convergence is safely obtained after 4 iterations to an accuracy of $10^{-3}$. The modified condensing algorithm needs two refining steps.

![Fig. 3 Notorious test problem - multiple shooting ($\mu = 60, p(0) = 1$)](image)

\textit{Detndrogenation of pyridine}

This problem (due to ZWAGA [42]) was investigated in detail in [1,7]. It describes the identification of 11 unknown parameters (the rate constants) in a system of 7 differential equations

\[
\begin{align*}
\dot{A} &= -p_1 A + p_9 B \\
\dot{B} &= p_1 A - p_2 B - p_3 C B + p_7 D - p_9 B + p_{10} D F \\
\dot{C} &= p_2 B - p_3 C B - 2 p_4 C C - p_6 C + p_8 E + p_{10} D F + 2 p_{11} E F \\
\dot{D} &= p_3 C B - p_5 D - p_7 D - p_{10} D F \\
\dot{E} &= p_4 C C + p_7 D - p_8 E - p_{11} E F \\
\dot{F} &= p_5 C B + p_8 C C + p_6 C - p_{10} D F - p_{11} E F \\
\dot{G} &= p_6 C + p_7 D + p_8 E
\end{align*}
\]

representing a recycling process for pyridine (A), which is transformed into ammonia (F) and pentane (G) by means of three catalysts.

The initial values are assumed to be error-free (A(0) = 1 and all others zero.) 77 Measurements of all seven state variables at times $t_j = j \cdot 0.5$ ($j = 1, \ldots, 77$) are available. The pyridine problem allows some instructive numerical tests and comparisons.

### 8.1 Multiple Shooting vs. Single Shooting (Efficiency)

(A) In a first experiment the initial guess for all parameters is chosen to be $p^{(0)}_1 = 1$ - a rather poor guess, since the solution values vary between $p_2 = 0.0201$ and $p_3 = 29.4$. For multiple shooting, seven meshpoints are selected (0,0.5,1.5,...,5.5). Single shooting needs 18 iterations (42 sec.), with the damping strategy active during the first 13 iterations. Multiple shooting requires only 8 iterations and a computing time of 16 sec. (IBM 370/168), and needs two damping steps only.

It is not only the drastic reduction of computing time that is noticeable: for single shooting, the initial guess is "far" from the local convergence domain, for multiple shooting, it is just on the rim!

(B) A second experiment is very instructive, here, all parameters are set to zero initially.

With multiple shooting, only 7 iterations (11 sec.) are needed. For single shooting, the initial rank of the condensed system is only 1 (excluding initial values), and it increases step by step during 22 iterations (58 sec.).

The explanation is at hand: system (55) is homogeneous, and the initial values represent a very specific steady state. Thus, only parameter $p_1$ is effective in the first iteration, followed by $p_2, p_3$ in the next, etc. Multiple shooting leads to a full rank problem from the start, since the subinterval integrations do not begin in a steady state. Thus, multiple shooting has also some "regularizing" properties.

### 8.2 Multiple Shooting vs. Collocation

| Table 2 | Comparison PARFIT vs. COLFIT - Pyridine problem A (EPS=1D-3) |
|-----------------|-----------------|-----------------|-----------------|
| iterations | comp.time(sec) | F-calls | meshsize | basic storage |
| COLFIT | 11 (+4) | 52 | 19160 | 28 | 3348 + 4900 |
| PARFIT | 8 | 16 | 33099 | 7 | 3348 + 882 |

- Includes 4 mesh refinements, excluding the integrator (+ 782).
- The second figure represents the mesh-dependent storage

Table 2 compares the performance of PARFIT and COLFIT for problem 8.1A. Obviously, PARFIT has distinct advantages over COLFIT with respect to computing time and storage requirements (and for higher accuracies the
difference is even more marked.) According to the number of iterations, PARFIT also seems to have a slightly better convergence behaviour. On the other hand, COLFIT is extremely economical w.r.t. F-evaluations (recall that it is a 4th order method with two F-calls per subinterval only). Although the results for this problem are characteristic according to our experience, it allows only for a rather preliminary conclusion.

A more detailed comparison on the basis of several examples is in preparation [2]. It seems, however, that the real advantages of collocation are in the field of singular perturbation problems.

8.3 Comparison of GGN and Random Search

For quite a while, there has been a discussion in P1 about the relative advantages of the (higher order) GGN method of PARFIT and low order methods such as random optimization. We therefore conclude with two examples treated previously by MMLSTEIN [9, 10] (with random optimization) and the author [1, 7]. Both examples are multieperiment problems and were solved by PARME.

Inducible Enzyme Synthesis

This biochemical problem originally due to HEINMETS [43] is reputed to be extremely ill-conditioned (SCHLITZ-GRITERMANN [44]). For details, the reader is referred to [1, 9]. The P1 problem consists of a model with 12 nonlinear o.d.e. and 14 unknown parameters, which have to be estimated from artificial data generated by adding pseudo-random noise to the true solution.

Three numerical experiments are performed. (A) uses one set of initial values, with 4 data points (46 measurements), (B) is like (A) but with 12 data points (144 measurements), (C) uses a second set of initial values with 4 data points and 48 measurements in addition to (A).

Table 3 summarizes the results of table 8.3 in [1]. The PARMEX parameter estimates are always within the 95% confidence intervals, whereas some of the estimates due to [9] are always outside.

<table>
<thead>
<tr>
<th>Trajectories</th>
<th>Data Points (Measurements)</th>
<th>Parameters within 95% Confidence Intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>4 (48)</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>12 (144)</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>4 (96)</td>
</tr>
</tbody>
</table>

Grasseete Ecosystem

This test problem is based on a model of GARFINKEL [45] for an ecosystem of one herbivore feeding on two types of grass in a two trophic level ecosystem. Details can be found in [7, 10]. The model involves three nonlinear o.d.e. and 8 parameters. Following [10], pseudoeperimental data are generated by adding 9% noise to three trajectories corresponding to three different sets of initial values at 20 equidistant data points each.

Three numerical problems are generated by solving the P1 problem for one, two and all three data sets.

The results for the 3 critical parameters $p_1, p_2, p_7$ documented in [10] are assembled in Table 4 together with the corresponding results of PARMEX. Full solution details for the latter are given in [7].

Table 4 Grasseete ecosystem - comparison of results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PARMEX</td>
<td>PARMEX</td>
<td>[10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[10]</td>
<td>[10]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.0</td>
<td>0.995</td>
<td>0.005</td>
<td>0.028</td>
</tr>
<tr>
<td>4</td>
<td>1.2</td>
<td>1.237</td>
<td>0.037</td>
<td>0.029</td>
</tr>
<tr>
<td>7</td>
<td>1.0</td>
<td>1.015</td>
<td>0.015</td>
<td>0.029</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.93</td>
<td>1.93</td>
<td></td>
</tr>
<tr>
<td>I+II</td>
<td>1.0</td>
<td>0.024</td>
<td>0.016</td>
<td>0.016</td>
</tr>
<tr>
<td>4</td>
<td>1.2</td>
<td>1.215</td>
<td>0.015</td>
<td>0.016</td>
</tr>
<tr>
<td>7</td>
<td>1.0</td>
<td>1.006</td>
<td>0.006</td>
<td>0.014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.22</td>
<td>1.22</td>
<td></td>
</tr>
<tr>
<td>I+II+III</td>
<td>1.0</td>
<td>1.005</td>
<td>0.005</td>
<td>0.011</td>
</tr>
<tr>
<td>4</td>
<td>1.2</td>
<td>1.213</td>
<td>0.013</td>
<td>0.013</td>
</tr>
<tr>
<td>7</td>
<td>1.0</td>
<td>1.995</td>
<td>0.005</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.13</td>
<td>1.13</td>
<td></td>
</tr>
</tbody>
</table>

a to be multiplied by (approx.) 4.1 for 95% confidence interval

b computing time (sec.) [Iterations]
As in the previous problem, one observes that for each problem at least two parameter estimates of the random optimizer are clearly outside the confidence region whereas the estimates of the GGN are always safely within and have an error that is overall smaller by an average factor of ten.

With respect to computing time, PARMEX seems to be more effective, too, since the two machines (IBM 370/168 (PARMEX) and IBM 360/91) appear to be comparable.

From the results one may conclude, that the higher order GGN method has a definitely safer local convergence behaviour than random optimization, and delivers estimates which are statistically reliable. On the other hand it appears that the higher order information also pays off in the earlier iteration stages - which might be explained in terms of the "optimal stepsize" strategies discussed in sect. 5.

9. Conclusion

The present article reviews recent advances in the development of constrained least squares boundary value problem algorithms for o.d.e., in particular an enhanced multiple shooting code, PARFIT, and a collocation code, COLFIT, that are apt for treatment of a broad class of inverse problems in parameter identification.

Several implementational developments are emphasized that have led to a substantial increase in efficiency, stability and applicability of these methods, such as a stabilized recursive linear system solver, adaptive accuracy selection and efficient internal numerical differentiation schemes.

The performance of the new techniques is demonstrated by several numerical test problems, which show significant advantages over previous approaches.

Acknowledgement: This work was supported by the Deutsche Forschungsgemeinschaft (Sonderforschungsbereich 72).

References

[8] U. Nowak, P. Deuflhard: Towards Parameteridentification for Large Chemical Reaction Systems, these proceedings
[18] P. Deuflhard: Recent Advances in Multiple Shooting Techniques, in: Computational Techniques for Ordinary Differential Equations (Gladwell / Sayers, eds.)


[38] P. Deuflhard, G. Bader: Multiple Shooting Techniques Revisited, these proceedings


