Continuation and Collocation for Parameter Dependent Boundary Value Problems *

G. Bader and P. Kunkel

Universität Heidelberg
Inst. f. Angewandte Mathematik
Im Neuenheimer Feld 294
D-6900 Heidelberg
West Germany

Abstract

The numerical approximation of boundary value problems in the context of one-parameter continuation is considered. A collocation discretization of parameter dependent boundary value problems is suggested. The convergence properties are analysed and a stable algorithm is presented. Tangent continuation and steplength control are combined with a Gauss-Newton type corrector, which is shown to be quadratically convergent. Based on this analysis, a general purpose continuation code COLCON has been implemented. Numerical examples demonstrating the efficiency of the approach are presented.

The present work has been supported by the Deutsche Forschungsgemeinschaft.

* This paper is published in: SIAM J. Sci. Stat. Comput., 10, 72-88, (1989)

1 Introduction

The formulation of many computational problems includes physical parameters. Frequently, not only a single solution is of interest but the dependence of the solution on selected parameters. The numerical solution of such problems is usually done by continuation methods.

The present paper deals with the tracing of the solution manifold of boundary value problems in ordinary differential equations depending on a single free parameter. There exists a number of codes for the numerical solution of such problems. Probably, the best known codes in this area are AUTO of Doedel [16] in its recent version of Doedel and Kernevez [17] and BIFPACK of Seydel [27]. A more recent approach is taken in MULCON of Deuflhard, Fiedler and Kunkel [14]. For further codes with continuation features see [1,21,22,24,26].

When tracing solution manifolds, usually a large number of boundary value problems have to be solved. Therefore, the choice of an efficient and reliable boundary value solver is a most important aspect.

The multiple shooting type codes [27,14] inherit their efficiency from the high standard of initial value software available. However, applied to stiff boundary value problems, i.e. problems with large Lipschitz constants, this approach experiences severe limitations originating in the fact that the initial value problems to be solved during the solution process are unstable.

Global methods, on the other hand, are much less affected by stiffness. The crucial point for the efficiency and robustness of codes based on these methods seems to be the effectiveness of the mesh selection procedure implemented (see [4,5,6]). A code of this type is the above cited AUTO which uses Gaussian collocation for discretizing the differential equations. It implements an adaptive mesh selection similar to COLSYS [4,5], but keeps the total number of mesh intervals fixed. The discretization points are only equidistributed with respect to an estimation of the discretization error after a certain number of continuation steps. The overall accuracy of the computed collocation solution, however, is not checked. Therefore it is not clear how well the computed collocation solution approximates the exact solution. Especially in real life applications, the solution profile of the discrete boundary value problem may differ considerably from the exact one. In fact, the whole structure of the solution diagram may be disturbed by the presence of so-called spurious solutions.

The purpose of the present paper is to report on the development of a continuation code aiming at an automatic tracing of solution branches for parameter dependent boundary value problems where the total error of the collocation approximation is kept below a user prescribed error tolerance. To achieve this, an efficient steplength control for the continuation procedure is combined with a collocation discretization using a fully adaptive mesh selection procedure. In section 2, the underlying continuation techniques are outlined. A tangent continuation with theoretically backed steplength control is combined with a quadratically convergent Gauss-Newton type iteration, extending ideas of [13]. A collocation discretization for parameter dependent boundary value problems based on the techniques used in the collocation code

COLNEW of Bader and Ascher [10] is presented and its convergence properties are analysed in section 3. The following section 4 discusses the implementation of these ideas in a general purpose code COLCON. Finally, in section 5, numerical examples demonstrating the performance of the code are given.

2 Tangent continuation with steplength control

For notational convenience we consider in this and the following section one scalar m-th order differential equation depending on a single parameter only

$$\mathcal{L}(y,\tau) \equiv y^{(m)} - f(x, y, y', \ldots, y^{(m-1)}; \tau) = 0 \,, \qquad a < x < b \,, \qquad \tau \in \mathrm{R} \tag{2.1a}$$

with m boundary conditions

$$\mathcal{B}_j(y,\tau) \equiv g_j(\boldsymbol{y}(\xi_j);\tau) = 0 \,, \qquad a \le \xi_j \le b \,, \quad 1 \le j \le m \tag{2.1b}$$

where

$$\boldsymbol{y}(x) = (y(x), y'(x), \ldots, y^{(m-1)}(x))^T$$

denotes the state vector of the corresponding first order system. Note, however, that all subsequent considerations directly carry over to systems of mixed order differential equations depending on several parameters.

Remark. Most discretization schemes require (2.1a) to be rewritten as a system of first order differential equations, cf. [14,17,27]. Collocation, however, allows for a direct treatment of higher order differential equations in a more efficient way, cf. [3,5]. In contrast to [14,17,27] the boundary conditions (2.1b) are assumed to be separated. Nonseparated boundary conditions can be easily reformulated by introducing additional parameters or trivial differential equations, cf. [8].

Assume that the (generally nonlinear) functions f and g_j are sufficiently smooth with respect to the arguments x, $\boldsymbol{y}$, and τ. In particular, require that any solution of (2.1) possesses the (minimal) smoothness $y \in C^{m-1}[a,b] \cap C^m_\pi$, C^m_π requiring the m-th derivative of any solution to be piecewise continuous on a partition π of $[a,b]$ (cf. (3.1)).

If we consider the parameter τ as an additional unknown, problem (2.1) becomes an underdetermined boundary value problem. Following the theoretical treatment in [18], we assume the solution set $\mathcal{M}$ of (2.1) to be a continuously differentiable one dimensional manifold in $\mathcal{C} = (C^{(m-1)}[a,b] \cap C^m_\pi) \times \mathrm{R}$. This assures that the solution manifold $\mathcal{M}$ can be locally parametrized along its tangent direction.

To solve problem (2.1) we apply an iterative method based on linearization. For that purpose define the linearization of (2.1) by

$$\mathcal{L}[y,\tau](u,\delta) \equiv u^{(m)} - \sum_{l=1}^{m} c_l(x) u^{(l-1)} - d(x)\delta \tag{2.2a}$$

where

$$c_l(x) = \frac{\partial f(x, \mathbb{y}, \tau)}{\partial \boldsymbol{y}_l} \,, \qquad l = 1, \ldots, m \,,$$

$$d(x) = \frac{\partial f(x, \mathbb{y}, \tau)}{\partial \tau}$$

with the boundary conditions

$$\mathcal{B}_j[y,\tau](u,\delta) \equiv \sum_{l=1}^{m} b_{jl} u^{(l-1)}(\xi_j) + e_j \delta \,, \qquad 1 \le j \le m \tag{2.2b}$$

where

$$b_{jl} = \frac{\partial g_j(\boldsymbol{y},\tau)}{\partial \boldsymbol{y}_l} \,, \qquad l = 1,\ldots,m \,,$$
$$e_j = \frac{\partial g_j(\boldsymbol{y},\tau)}{\partial \tau} \,.$$

Before outlining the details of the continuation process, problem (2.1) is rewritten in the more compact form

$$F(z) = 0 \,, \qquad z = (y,\tau) \,, \tag{2.3}$$

where F contains both the nonlinear differential operator (2.1a) and the boundary conditions (2.1b). This operator equation is now considered on appropriate Hilbert spaces

$$F{:}\,\Omega \subset Z \to Y \tag{2.4}$$

with F being a $C^r(\Omega)$-mapping, $r \ge 1$, on some open subset $\Omega \subset \mathcal{C} \subset Z$ with $\mathcal{C}$ continuously embedded into Z. Let $W^\nu[a,b]$, $0 \le \nu \le m$ denote the Sobolev spaces

$$W^\nu[a,b] = \{u \mid u^{(\nu)} \in L_2[a,b]\} \tag{2.5a}$$

equipped with the scalar products

$$(u,v)_\nu = \sum_{l=0}^{\nu} \int_a^b u^{(l)}(s) v^{(l)}(s)\, ds \,, \tag{2.5b}$$

and the induced norms $\|u\|_\nu = (u,u)_\nu^{1/2}$. Then, with the canonical scalar products, the spaces Z and Y are naturally identified with the Hilbert spaces

$$\begin{aligned} Z &= W^m[a,b] \times \mathrm{R} \quad , \quad \|(u,\tau)\|_Z = \left(\|u\|_m^2 + \tau^2\right)^{1/2} \,, \\ Y &= W^0[a,b] \times \mathrm{R}^m \,, \quad \|(u,g)\|_Y = \left(\|u\|_0^2 + \sum_{l=1}^{m} g_l^2\right)^{1/2} \,. \end{aligned} \tag{2.5c}$$

Note that for the chosen spaces the Fréchet derivative $DF{:}\,\Omega \to Y$ of F is a bounded linear Fredholm operator of index 1 (cf. [18]). This ensures the existence of the Moore-Penrose pseudoinverse, see chap. 8 of [11]. Thus, in order to find a solution of the underdetermined operator equation (2.3) we apply the Gauss-Newton iteration

$$\begin{aligned} &(a) \qquad \Delta z^k = -DF(z^k)^+ F(z^k) \\ &(b) \qquad z^{k+1} = z^k + \Delta z^k \qquad\qquad k = 0,1,\ldots \end{aligned} \tag{2.6}$$

where $DF(z^k)^+$ denotes the Moore-Penrose pseudoinverse of $DF(z^k)$. The choice of an appropriate starting value $z^0 \equiv z_p \in \Omega$ is described below. For the realization of this method in the case of finite dimensional systems of equations see [13,14]. The convergence properties of the iterative scheme are given by

Theorem 1

Let $F\colon \Omega \to Y$, $F \in C^1(\Omega)$ for some convex subset $\Omega \subset Z$ such that the Fréchet derivative DF of F has the properties

$$\begin{aligned} \dim \text{kernel}\, DF(z) &= 1 \\ DF(z)Z &= Y \end{aligned} \qquad \text{for } z \in \Omega.$$

Then, under the assumptions

(2.7)

$$(a) \quad \|\Delta z^0\|_Z \le \alpha_o, \qquad z^0 \in \Omega$$

$$(b) \quad \|DF(z_1)^+[DF(z_2) - DF(z_3)]\|_Z \le \omega_o \|z_2 - z_3\|_Z$$
$$\text{for any } z_1, z_2, z_3 \in \Omega,\ z_2 - z_3 \in \text{range}\, DF(z_1)^+$$

$$(c) \quad h_o = \frac{1}{2}\alpha_o\omega_o < 1$$

$$(d) \quad S(z^0, r) = \left\{ z \in Z \;\middle|\; \|z - z^0\|_Z \le r \right\} \subset \Omega$$
$$\text{for } r = \frac{\alpha_o}{1 - h_o}$$

iteration (2.6) remains in $S(z^0, r)$ and converges to some solution $z^* \in \mathcal{M}$ with

$$F(z^*) = 0$$

on the solution manifold. The rate of convergence is quadratic

$$\|z^{k+2} - z^{k+1}\|_Z \le \frac{\omega_o}{2}\|z^{k+1} - z^k\|_Z^2 .$$

∎

The proof follows from theorem 4 of [15] realizing that $\kappa(z)$ defined in [15] vanishes identically under the above assumptions (cf. [13]).

Note that the conditions of the above theorem are affine invariant, i.e. they are invariant under bounded linear bijections of appropriate spaces, cf. [15]. Classical formulations of convergence theorems use the Lipschitz constant L of the Fréchet derivative $DF(z)$ in order to bound the radius of convergence for iterative schemes. Applied to problems with large Lipschitz constants, e.g. stiff boundary value problems, this results in a very small convergence radius proportional to $1/L$ being unnecessary restrictive. In contrast to this, the above theorem gives a convergence radius which does not depend on $1/L$.

The Moore-Penrose pseudoinverse is known to find the *least squares solution* Δz^+ of $DF(z)\Delta z = -F(z)$, that is

(2.8)

$$\|\Delta z^+\|_Z = \inf\left\{ \|\Delta z\|_Z \;\middle|\; DF(z)\Delta z = -F(z) \right\}.$$

Furthermore, the solution Δz^+ is orthogonal to the local tangent vector

$$(\Delta z^+, t(z))_Z = 0$$

where $t(z)$ is defined in (2.9). In the above sense iteration (2.6) searches a solution of $F(z) = 0$ closest to the starting value z^0. In practical applications this tends to minimize the number of iteration steps necessary to solve $F(z) = 0$.

From (2.8) we also conclude that the Moore-Penrose pseudoinverse has minimal norm K over all generalized inverses $DF(z)^-$, see [11]. In terms of boundary value problems this leads to a minimal stability or conditioning constant for the underdetermined linear problem (2.2). See [3,10,19] for the importance of a small stability constant when solving linear boundary value problems. In that respect the Moore-Penrose pseudoinverse leads to solutions of the linear problems with optimal stability bounds.

For the following define the tangent vector $t(z)$ as a solution of the linear system

$$DF(z)t(z) = 0 \tag{2.9a}$$

normalized to unit norm

$$\|t(z)\|_Z = 1\,. \tag{2.9b}$$

Let z_0 be a point on the solution manifold $\mathcal{M}$. We predict an estimation for an advanced point on the solution manifold along the tangent direction $t(z_0)$ by

$$z_p = z_0 + s\, t(z_0) \tag{2.10}$$

with an appropriately chosen steplength parameter s. Then, we apply the corrector iteration (2.6) with starting value $z^0 = z_p$ in order to find a nearby solution z^* with $F(z^*) = 0$.

The above approach is closely related to pseudo-arclength continuation, see [23]. Instead of employing the Moore-Penrose pseudoinverse, system (2.3) is enlarged by a normalization condition, e.g. of the form

$$\|z(s) - z_0\|_Z^2 - s^2 = 0$$

where s is a prescribed steplength approximating the arclength of the solution branch. An expansion argument then shows that the two approaches are only identical in the limiting case $s \to 0$. For nonzero stepsizes, especially in the case of high curvature, they may differ considerably.

The approximation property of the tangent predictor is utilized by the following

Theorem 2

Consider the continuation process (2.10) with the iteration scheme (2.6). In addition to the assumptions of theorem 1, let

$$\|DF(z_5)^+[DF(z_4) - DF(z_0)]t(z_0)\|_Z \le \omega_t s_1 \|t(z_0)\|_Z^2 \tag{2.11}$$

for any

$$\begin{aligned} z_4 &= z_0 + s_1 t(z_0) \\ z_5 &= z_0 + s_2 t(z_0) \end{aligned} \qquad 0 \le s_1, s_2 \le s$$

such that $z_{4,5} \in \Omega$ and z_0 is a point on the solution manifold $\mathcal{M}$. Then the iteration (2.6) converges to a solution $z^* \in \mathcal{M}$ for any starting point z_p from (2.10) with

$$s < s_{\max} = 2 \left(\frac{1}{\omega_o \omega_t} \right)^{1/2} . \tag{2.12}$$

∎

For a proof see [13].

Relation (2.12) provides an upper bound for acceptable continuation stepsizes. On this basis a theoretical backed steplength control can be derived (see [13]). For this purpose the unknown global bounds ω_o, ω_t have to be replaced by accessible local estimates $[\omega_o]$ and $[\omega_t]$. To work with these estimates successfully, one has to consider a predictor corrector type steplength strategy. Following [13], a value s_{pred} for the steplength is predicted after a successful continuation step with solution z^* and steplength s_{old} by

$$s_{\text{pred}} = \sqrt{\frac{\theta_{\max}}{\theta_0} \frac{\|\Delta z^0\|_Z}{\|z^0 - z^*\|_Z} \frac{|c_*|^2}{|c_0|^3}} \, s_{\text{old}} \tag{2.13}$$

with the values $c_* := (t(z^*), t(z^0))_Z$, $c_0 := (t(z_0), t(z^0))_Z$ describing the curvature of the solution manifold and

$$\theta_k = \frac{\|\overline{\Delta z}^{k+1}\|_Z}{\|\Delta z^k\|_Z} , \qquad k = 0, 1, \ldots \tag{2.14}$$

where $\overline{\Delta z}^{k+1}$ solves the simplified correction

$$\overline{\Delta z}^{k+1} = -DF(z^k)^+ F(z^{k+1}) .$$

and $\theta_{\max} \leq 1$ is a chosen safety factor. With this prediction a new value of z_p is found and the convergence of the iteration scheme (2.6) is monitored by

$$\theta_k < 1 .$$

If the convergence of iteration (2.6) proves to be insufficient, the continuation steplength is reduced by the correction formula

$$s_{\text{corr}} = \sqrt{\frac{\theta_{\max}}{\theta_k}} \, s_{\text{pred}} \tag{2.15}$$

obtained from (2.12) using updated estimates for $[\omega_o]$ and $[\omega_t]$, and the iteration is repeated. The above steplength control is implicitly based on the assumption that $c_k := (t(z^k), t(z_0))_Z > 0$. To be on the safe side, the validity of $c_k \geq c_{\min}$ for some chosen $c_{\min} > 0$ is monitored during the iteration process.

The above strategy, tangent continuation combined with the steplength control (2.13), (2.15) has proved to be highly efficient in the context of continuation for systems of nonlinear equations. A comparison with a number of different approaches including pseudo-arclength continuation is given in [13]. Solving continuation problems for boundary value problems, discretization yields large systems of nonlinear equations. Thus, it can be expected that the efficiency of this strategy carries over to boundary value problems.

Remark. The above considerations cover the treatment of limit or turning points. However, branch or bifurcation points are not covered because the assumptions of theorem 1 no longer hold. Therefore, a special treatment is required for the calculation of such points. If only tracing a solution path is desired, practical experience has shown that one can expect that the above algorithm usually "jumps" over these critical points. For a justification see [23].

3 Discretization of parameter dependent differential equations

Consider a k-stage collocation scheme with $k \geq m$ for the boundary value problem (2.1). Thus, there is a partition (mesh) π of $[a,b]$

$$\begin{aligned} &\pi : a = x_1 < x_2 < \cdots < x_N < x_{N+1} = b\,, \\ &h_i = x_{i+1} - x_i, \qquad h = \max_{1 \leq i \leq N} h_i \end{aligned} \tag{3.1a}$$

and k points

$$0 \leq \varrho_1 < \varrho_2 < \cdots < \varrho_k \leq 1 \tag{3.1b}$$

which we choose to be Gaussian points.

A piecewise polynomial function $y_\pi(x)$ is sought such that $y_\pi \in \mathcal{P}_{k+m,\pi}[a,b] \cap C^{m-1}[a,b]$ * satisfies the m boundary conditions (2.1b) given on a subset of π, and $y_\pi(x)$ satisfies the differential equations (2.1a) at the collocation points

$$\begin{aligned} \mathcal{L}_\pi(y_\pi,\tau) = y_\pi^{(m)}(x_{ij}) - f(x_{ij}, y_\pi, y_\pi', \ldots, y_\pi^{(m-1)}; \tau)\,, \\ x_{ij} = x_i + h_i \varrho_j \end{aligned} \tag{3.2}$$

for $1 \leq j \leq k$ and $1 \leq i \leq N$. Note that this collocation procedure directly carries over to mixed order systems of differential equations depending on several parameters.

Using local linearization techniques to solve problem (2.1) it is sufficient to apply the above collocation procedure iteratively to linear problems

$$\mathcal{L}[y,\tau](u,\delta) = q(x)$$

corresponding to (2.2). Recall the monomial Runge-Kutta basis representation [10,7]. Let us focus on one subinterval $[x_i, x_{i+1}]$ and express the polynomial $u_\pi(x)$ by

$$u_\pi(x) = \sum_{l=1}^{m} \frac{(x-x_i)^{l-1}}{(l-1)!} u_\pi^{(l-1)}(x_i) + h_i^m \sum_{j=1}^{k} \psi_j\left(\frac{x-x_i}{h_i}\right) u_\pi^{(m)}(x_{ij}) \tag{3.3}$$

with polynomials ψ_j of order $k+m$ defined by

$$\begin{aligned} &\psi_j^{(l-1)}(0) = 0\,, \qquad 1 \leq l \leq m\,, \\ &\psi_j^{(m)}(\varrho_\nu) = \delta_{\nu j}\,, \qquad 1 \leq \nu \leq k\,. \end{aligned} \qquad 1 \leq j \leq k \tag{3.4}$$

Defining the vectors

$$\begin{aligned} \boldsymbol{u}_i &= (u_\pi(x_i), u_\pi'(x_i), \ldots, u_\pi^{(m-1)}(x_i))^T, \\ \boldsymbol{v}_i &= (u_\pi^{(m)}(x_{i1}), u_\pi^{(m)}(x_{i2}), \ldots, u_\pi^{(m)}(x_{ik}))^T \end{aligned}$$

* We say that v is in $\mathcal{P}_{k+m}$ if $v(x)$ is a polynomial of order $k+m$ (degree $< k+m$) on an appropriate interval, and that v is in $\mathcal{P}_{k+m,\pi}$ if $v(x)$ is a piecewise polynomial which is in $\mathcal{P}_{k+m}$ on each subinterval of the mesh π.

and applying the collocation equations now yields k relations

$$\boldsymbol{W}_i \boldsymbol{v}_i - \boldsymbol{V}_i \boldsymbol{u}_i - \boldsymbol{d}_i \delta = \boldsymbol{q}_i \,, \qquad 1 \le i \le N \tag{3.5}$$

where $\boldsymbol{W}_i$ and $\boldsymbol{V}_i$ are matrices of order $k \times k$ and $k \times m$ respectively, with entries

$$\boldsymbol{W}^i_{\nu j} = \delta_{\nu j} - \sum_{l=1}^{m} c_l(x_{i\nu}) h_i^{m+1-l} \psi_j^{(l-1)}(\varrho_\nu) \,, \qquad 1 \le \nu, j \le k \tag{3.6}$$

$$\boldsymbol{V}^i_{\nu j} = \sum_{l=1}^{j} \frac{c_l(x_{i\nu})(h_i \varrho_\nu)^{j-l}}{(j-l)!} \,, \qquad 1 \le \nu \le k \,, \quad 1 \le j \le m \,, \tag{3.7}$$

and

$$\boldsymbol{d}_i = (d(x_{i1}), d(x_{i2}), \ldots, d(x_{ik}))^T, \tag{3.8}$$

$$\boldsymbol{q}_i = (q(x_{i1}), q(x_{i2}), \ldots, q(x_{ik}))^T. \tag{3.9}$$

Next, we obtain m additional relations from the global continuity requirements

$$\boldsymbol{C}_i \boldsymbol{u}_i + \boldsymbol{D}_i \boldsymbol{v}_i = \boldsymbol{u}_{i+1} \,, \qquad 1 \le i \le N \tag{3.10}$$

where $\boldsymbol{C}_i$ is an $m \times m$ upper triangular matrix with entries

$$\boldsymbol{C}^i_{\nu j} = \frac{h_i^{j-\nu}}{(j-\nu)!} \,, \qquad 1 \le \nu \le j \le m \tag{3.11}$$

and $\boldsymbol{D}_i$ is an $m \times k$ matrix with entries

$$\boldsymbol{D}^i_{\nu j} = h_i^{m+1-\nu} \psi_j^{(\nu-1)}(1) \,, \qquad 1 \le \nu \le m \,, \quad 1 \le j \le k \,. \tag{3.12}$$

From equations (3.5), (3.6) it follows immediately that for h_i sufficiently small $\boldsymbol{W}_i$ is nonsingular and

$$\boldsymbol{W}_i^{-1} = \boldsymbol{I} + \mathcal{O}(h_i) \,.$$

Hence, using a standard linear equation solver with pivoting, the local unknowns $\boldsymbol{v}_i$ can be eliminated as long as $\boldsymbol{W}_i$ is nonsingular. The stability of this process with respect to roundoff error propagation has been shown in [3,7,10] where both a theoretical justification and numerical tests are presented. A different elimination process allowing the direct treatment of nonseparated boundary conditions is employed in [17,23].

Using the equations (3.5) and (3.10) for this elimination leads to relations between the solution values at the end points of the i-th subinterval

$$\boldsymbol{\Gamma}_i \boldsymbol{u}_i - \boldsymbol{u}_{i+1} + \boldsymbol{p}_i \delta = \boldsymbol{r}_i \,, \qquad 1 \le i \le N \tag{3.13}$$

where

$$\boldsymbol{\Gamma}_i = \boldsymbol{C}_i + \boldsymbol{D}_i \boldsymbol{W}_i^{-1} \boldsymbol{V}_i \,, \quad \boldsymbol{p}_i = \boldsymbol{D}_i \boldsymbol{W}_i^{-1} \boldsymbol{d}_i \,, \quad \boldsymbol{r}_i = -\boldsymbol{D}_i \boldsymbol{W}_i^{-1} \boldsymbol{q}_i \,. \tag{3.14}$$

Assembling these equations for $1 \le i \le N$ and the corresponding boundary conditions results in a global shooting like system of equations

$$\boldsymbol{G} \begin{pmatrix} \boldsymbol{u} \\ \delta \end{pmatrix} = \boldsymbol{r} \tag{3.15a}$$

with an $n \times (n+1)$-matrix $\boldsymbol{G}$

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{B}_a & & & & & \boldsymbol{p}_a \\ \boldsymbol{\Gamma}_1 & -\boldsymbol{I} & & & & \boldsymbol{p}_1 \\ & \boldsymbol{\Gamma}_2 & -\boldsymbol{I} & & & \boldsymbol{p}_2 \\ & & \ddots & \ddots & & \vdots \\ & & & \boldsymbol{\Gamma}_N & -\boldsymbol{I} & \boldsymbol{p}_N \\ & & & & \boldsymbol{B}_b & \boldsymbol{p}_b \end{pmatrix} \tag{3.15b}$$

where $n = (N+1)m$ and $\boldsymbol{B}_a$, $\boldsymbol{B}_b$, $\boldsymbol{p}_a$, and $\boldsymbol{p}_b$ stand for the entries coming from the boundary conditions (2.2b). For ease of notation, we assume in the above formula to have boundary conditions only at the end points.

The linear system (3.15) is solved by computing its *linear least squares solution*

$$\begin{pmatrix} \boldsymbol{u} \\ \delta \end{pmatrix} = \boldsymbol{G}^{+} \boldsymbol{r} \tag{3.16}$$

such that the norm $\|(u_\pi, \delta)\|_Z$, defined in (2.5), is minimized for the collocation solution (u_π, δ) of the linear problem (2.2). As the Sobolev norm $\|u_\pi\|_m$ is not directly accessible, it is approximated by Gaussian quadrature yielding a discrete norm

$$\|(u,\delta)\|_\pi = \Big(\sum_{i=1}^{N} h_i \sum_{j=1}^{k} b_j \sum_{l=0}^{m} {u^{(l)}}^2(x_{ij}) + \delta^2\Big)^{1/2} \tag{3.17}$$

with b_j, $1 \le j \le k$, denoting the corresponding quadrature weights. This approximation possesses the convergence property

$$\|(u,\delta)\|_\pi - \|(u,\delta)\|_Z = \mathcal{O}(h^{2k}) \,. \tag{3.18}$$

This completes the solution of linear collocation problems.

Returning to the nonlinear problem (2.1) and using the notation of section 2, we rewrite the collocation equations (3.2) in the form

$$F_\pi(z_\pi) = 0 \,. \tag{3.19}$$

Applying the Gauss-Newton iteration (2.6) to these nonlinear collocation equations results in a discretized analogue for iteration (2.6). The solution of the linearized subproblems

$$DF_\pi(z_\pi)\Delta z_\pi = -F_\pi(z_\pi) \tag{3.20}$$

is achieved as specified in (3.5) through (3.16) above.

After these preparations the convergence properties of the discretization can be considered.

Theorem 3

Assume that problem (2.1) has a continuously differentiable solution manifold $\mathcal{M}$ in $\mathcal{C}$ and that its Fréchet derivative has a nullspace of dimension one. Consider the k-stage collocation at Gaussian points for (2.1). Then there exist positive constants ϱ and h_0 such that for all meshes with $h \le h_0$ the following is valid.

a) In a tube S_ϱ of radius ϱ around the solution manifold $\mathcal{M}$ there exists a unique one dimensional solution manifold $\mathcal{M}_\pi$ in $(\mathcal{P}_{k+m,\pi}[a,b] \cap C^{m-1}[a,b]) \times \mathrm{R}$ of the collocation equations (3.2).

b) The solution manifold $\mathcal{M}_\pi$ can be computed by the iteration scheme (2.6) applied to the collocation equations (3.2). The iteration converges quadratically to some point on $\mathcal{M}_\pi$, provided that the initial guess is sufficiently close to the solution manifold $\mathcal{M}$.

c) For the linearized boundary value problem (2.2), there is a stable implementation with stability constant cN, where c depends on the stability constant of the linearized problem.

d) Let (y_π, τ_π) be a solution of the discrete problem (3.2). Then there exists a solution (y, τ) of the exact problem with $y \in C_\pi^{2k+1}$, such that the following superconvergence estimates hold.

$$\begin{aligned} |y^{(l)}(x_i) - y_\pi^{(l)}(x_i)| &= \mathcal{O}(h^{2k})\,, \qquad 0 \le l \le m-1\,, \quad 1 \le i \le N+1\,, \\ |\tau - \tau_\pi| &= \mathcal{O}(h^{2k}) \end{aligned} \tag{3.21a}$$

e) The global errors satisfy uniformly for $x \in [a,b]$

$$\begin{aligned} &y^{(l)}(x) - y_\pi^{(l)}(x) = \\ &\quad h_i^{k+m-l} y^{(k+m)}(x_i) P^{(l)}\left(\frac{x - x_i}{h_i}\right) + \mathcal{O}(h_i^{k+m-l+1}) + \mathcal{O}(h^{2k})\,, \\ &\qquad x_i \le x \le x_{i+1}\,, \quad 1 \le i \le N\,, \quad 0 \le l \le k+m-1 \end{aligned} \tag{3.21b}$$

where

$$P(\xi) = \frac{1}{k!(m-1)!} \int_0^\xi (t-\xi) \prod_{j=1}^{k} (t - \varrho_j)\, dt\,.$$

∎

The proof of the theorem follows along the lines of [2].

4 Implementation

The above considerations led us to implement a continuation code called COLCON. The code is based on the well known general purpose code COLSYS (cf. [4,5]) in its implementation COLNEW described in [10].

The class of problems treated by COLCON allows the following general form: a system of d nonlinear differential equations of orders m_ν, $1 \le m_\nu \le 4$, $1 \le \nu \le d$,

$$y_\nu^{(m_\nu)} = f_\nu(x, \boldsymbol{y}; p) , \qquad a \le x \le b \tag{4.1}$$

depending on the unknown m^*-vector function $\boldsymbol{y}$

$$\boldsymbol{y} = (y_1, y_1', \ldots, y_1^{(m_1-1)}, y_2, \ldots, y_2^{(m_2-1)}, \ldots, y_d^{(m_d-1)})^T \tag{4.2}$$

and the parameter vector of length *npar* is subject to separated nonlinear side conditions

$$\begin{aligned} g_j(\boldsymbol{y}(\xi_j); p) = 0 , \qquad & 1 \le j \le mside , \\ a \le \xi_1 \le \xi_2 \le \ldots \le \xi_{mside} \le b . & \end{aligned} \tag{4.3}$$

The total number of unknowns is $mstate = m^* + npar$ with $m^* = \sum_{\nu=1}^{d} m_\nu$. For continuation purposes we assume $mside = mstate - 1$ boundary conditions, i.e. one free parameter.

Remark. In the case of $mside = mstate$ boundary conditions, this implementation represents a generalization of COLNEW allowing a direct treatment of parameter dependent problems without continuation refered to as COLPAR in the following. Thus, it is no longer necessary to write the parameters p as trivial differential equations $p' = 0$ resulting in an unnecessary enlargement of the size of the boundary value problem. According to this, COLPAR proves to be more efficient than COLNEW in terms of computing time and especially in terms of storage requirements (see example 2 below).

When attempting to implement a general purpose collocation code, the following aspects arise:

1) solution representation
2) linear system solution
3) error estimations and mesh selection
4) nonlinear problem solution

Our aim in implementing the continuation code COLCON has been to write a modular piece of software being compatible with the code COLNEW. As a result the two codes are almost identical with respect to solution representation, error estimation and mesh selection. In fact, the latter one may be used to compute an initial solution to start continuation (see example 2). For continuation problems however, the new code differs considerably with respect to nonlinear problem solving. While in COLNEW a damped Newton method is realized, COLCON realizes a Gauss-Newton

iteration without damping but with a steplength control for the continuation procedure. The actual implementation is done in a predictor corrector mode with a convergence monitor along the lines indicated in section 2. During the continuation process the accuracy of the computed solution approximations is checked in order to satisfy the user prescribed tolerances. Further, a number of additional safeguards are implemented and checked before accepting a continuation step. In this respect the code provides a means to trace regular continuation paths automatically. This property seems to be important especially for the solution of real life problems.

One subject not yet covered is the numerical solution of the least squares equation (3.15) with (3.16). We do not intend to use QR-decomposition on the matrix $\boldsymbol{G}$ because this tends to destroy sparsity. Recall that the solution manifold of (3.15) is known to be of dimension one only. Hence, if we can compute, in a stable way, a tangent vector t

$$\boldsymbol{G}t = 0 \tag{4.4}$$

and a special solution z of

$$\boldsymbol{G}z = \boldsymbol{r} \tag{4.5}$$

with $z = (\boldsymbol{u}, \delta)$, then the least squares solution of (3.15) is easily computed by

$$z^+ = z - \frac{(z,t)_\pi}{(t,t)_\pi}\, t \tag{4.6}$$

where $(\cdot,\cdot)_\pi$ is the scalar product corresponding to (3.17).

In order to solve the linear system (3.15) we modify the code SOLVEBLOK [12] used in COLSYS and also in an updated form in COLNEW. This code solves regular systems of almost block diagonal linear systems, i.e. systems of the form (3.15) but without parameter columns. The extension of this code for parameter dependent problems has been previously considered in [9]. Note, however, that (3.15) is underdetermined. As a consequence, we have to apply some kind of column permutation in order to obtain a stable decomposition of $\boldsymbol{G}$. Unrestricted column permutation, however, would destroy the sparse structure of the matrix. Hence, we consider partial column permutation only. In order to retain numerical stability additional row permutation is applied. Let

$$P\boldsymbol{G}Q = LU \tag{4.7}$$

with $n \times n$ lower and $n \times (n+1)$ upper triangular matrices L and U and let P and Q be permutation matrices of corresponding dimensions. The decomposition by SOLVEBLOK is done in a blockwise manner with full row pivoting. Treating the parameter columns just as special right hand sides, the same strategy is used for matrix $\boldsymbol{G}$ except for the last block. This block, however, is decomposed by LU-decomposition with column pivoting. In contrast to unrestricted column permutation no additional fill-in compared with SOLVEBLOK is created in this way.

For general matrices the above algorithm is not necessarily stable. But, taking into account that $\boldsymbol{G}$ arises from the discretization of a boundary value problem, this procedure gives a decomposition of the form (4.7) with a nonsingular matrix L as $\boldsymbol{G}$ is known to be of full rank (cf. theorem 3,b). To show numerical stability of the decomposition in this context is an open question involving the stability of the underlying boundary value problem. So far, we have not yet experienced any difficulty with the implementation of this procedure.

The desired quantities t and z from (4.4) and (4.5) respectively are now easily computed from decomposition (4.7). Observe that the underdetermined problem has a unique solution, if the last component of the solution permuted by Q^T is prescribed. In this way, one calculates a tangent vector t by

$$\begin{aligned} U\overline{t} &= 0\,, \\ e_{n+1}^T\overline{t} &= 1\,, \\ t &= Q\overline{t} \end{aligned} \tag{4.8}$$

and a special solution z of the inhomogeneous equation by

$$\begin{aligned} L\overline{\boldsymbol{r}} &= P\boldsymbol{r}\,, \\ U\overline{z} &= \overline{\boldsymbol{r}}\,, \\ e_{n+1}^T\overline{z} &= 0\,, \\ z &= Q\overline{z} \end{aligned} \tag{4.9}$$

where e_{n+1} denotes the $(n+1)$-st unit vector. Together with (4.6), this yields the desired least squares solution of the linear subproblems (3.16).

5 Numerical results

In this section, the performance of the new code COLCON is demonstrated on four problems experiencing various kinds of difficulty. All examples were run on an IBM3090 using FORTRAN double precision (56 bit mantissa). For all examples the code was used with 4 collocation points per subinterval, requiring 5 digits accuracy for the solution and additionally 2 digits accuracy for the first derivative in case of second order differential equations.

Example 1. Consider the boundary value problem

$$\begin{aligned} &u'' + \tau \exp(u) = 0 \,, \\ &u(0) = 0 \,, \quad u(1) = 0 \,, \quad \tau \geq 0 \,. \end{aligned}$$

This problem is frequently used as a standard test example, see [17,14]. The solution shows a single turning point and an interior layer of increasing steepness. Starting from the origin, the code takes 1.1 sec CPU-time to complete 20 continuation steps. The corresponding diagram is given in figure 1.

Fig. 1: continuation diagram for example 1
$\times$ marking continuation steps

Example 2. This problem is described in [28,25] and was used in [10] as a test problem for COLNEW. The model describes mass and energy balance of the renal counterflow system. With F_{iv}—the axial volume flow in the i-th tube, J_{iv}—the outward transmural volume flux, $F_{iv}C_{ik}$—the axial flow of the k-th solute in the i-th

tube and J_{ik}—the outward transmural flux per unit length of the k-th solute from the i-th tube, there are 18 differential equations with nonseparated boundary conditions for F_{iv}, C_{ik}, $1 \le i \le 6$, $1 \le k \le 2$, describing the steady state case, see [28].

Now, it can be verified that this boundary value problem can be simplified to have only 11 differential equations depending on three parameters P_1, P_2, and $P_3 \equiv C_{22}$

$$
\begin{aligned}
&C'_{12} = 20h_{1v}C_{12}^2(C_{41} + C_{42} - 21C_{12})\,,\\
&C_{11} = 20C_{12}\,, \quad F_{1v} = \frac{0.05}{C_{12}}\,, \quad F_{2v} = -\frac{0.05}{C_{22}}\,,\\
&C'_{21} = 20C_{22}J_{21}\,, \quad C'_{22} = 0\,,\\
&C'_{31} = \frac{h_{3v}}{P_1}C_{31}^2(C_{41} + C_{42} - C_{31} - C_{32})\,,\\
&C'_{32} = \frac{C_{31}}{P_1}(J_{3v}C_{32} - J_{32})\,,\\
&F'_{4v} = -J_{4v}\,,\\
&P_1 = \frac{C_{31}(0)}{2C_{32}(0)}\,, \quad F_{3v} = \frac{P_1}{C_{31}}\,, \quad F_{5v} = 5\,, \quad F_{6v} = \frac{0.05}{C_{62}}\,,\\
&C'_{41} = \frac{1}{F_{4v}}(J_{4v}C_{41} - J_{41})\,,\\
&C'_{42} = \frac{1}{F_{4v}}(J_{4v}C_{42} - J_{42})\,,\\
&C'_{51} = -200(C_{51} - C_{41})\,, \quad C'_{52} = -200(C_{52} - C_{42})\,,\\
&C'_{61} = 20C_{62}(J_{6v}C_{61} - J_{61})\,,\\
&C'_{62} = 20C_{62}^2J_{6v}\,,
\end{aligned}
$$

with separated boundary conditions

$$
\begin{aligned}
&C_{12}(0) = 0.05\,, && C_{51}(0) = 1\,, && C_{52}(0) = 0.05\,, && C_{31}(0) = P_2\,,\\
&C_{31}(0) = 20P_1C_{32}(0)\,, && C_{62}(0) = C_{22}(0)\,, && C_{61}(0) = C_{21}(0)\,, &&\\
&C_{12}(1) = C_{22}(1)\,, && C_{21}(1) = 20C_{12}(1)\,, && C_{41}(1) = C_{51}(1)\,, && C_{42}(1) = C_{52}(1)\,,\\
&C_{61}(1) = 20P_1C_{62}(1)\,, && F_{4v}(1) = -5\,, && C_{61}(1) = P_2\,. &&
\end{aligned}
$$

Further, the functions J_{i1}, J_{i2} and J_{iv}, $1 \le i \le 6$, are now specified. Transmural volume fluxes are defined as follows:

$$
J_{iv} = \begin{cases} h_{iv}\sum_{k=1}^{2}(C_{4k} - C_{ik})\,, & i = 1,2,3,5\,,\\ -\sum_{k\neq 4,6} J_{kv}\,, & i = 4\,,\\ (1 - C_{i1}) + (0.05 - C_{i2})\,, & i = 6\,, \end{cases}
$$

where $h_{1v} = h_{3v} = 10$ and $h_{iv} = 0$ for $i = 2,5$. Transmural solute fluxes are:

$$
J_{i1} = \begin{cases} 0\,, & i = 1,3\,,\\ 1000(C_{i1} - C_{41})\,, & i = 5\,,\\ 0.75C_{i1}/(1 + C_{i1})\,, & i = 6\,, \end{cases} \qquad J_{i2} = \begin{cases} 0\,, & i = 1,2,6\,,\\ 1000(C_{i2} - C_{42})\,, & i = 5\,, \end{cases}
$$

$$J_{21}(x) = \begin{cases} 1.8\,, & 0 \le x \le 0.4\,, \\ 1.8 + [-18 + 100(C_{21}(x) - C_{41}(x))](x - 0.4)\,, & 0.4 < x < 0.5\,, \\ 10(C_{21}(x) - C_{41}(x))\,, & 0.5 \le x \le 1\,, \end{cases}$$

$$J_{32}(x) = \begin{cases} 0\,, & 0 \le x \le 0.4\,, \\ 0.1(C_{32}(x) - C_{42}(x))(x - 0.4)\,, & 0.4 < x < 0.5\,, \\ 0.01(C_{32}(x) - C_{42}(x))\,, & 0.5 \le x \le 1\,, \end{cases}$$

$$J_{4k}(x) = -\sum_{i \neq 4,6} J_{ik}(x)\,, \qquad 0 \le x \le 1\,, \quad k = 1, 2\,.$$

This completes the specification of the problem.

We treat this example in order to compare the relative performances of COLCON and COLNEW. Using COLNEW, we begin with the simple initial guess that all solution components are set identically to 1 and apply a chain of classical continuation steps. Thus, the problem is solved with continuation parameter $\tau = h_{1v} = h_{3v}$ taking the values $0, 0.8, 1.6, 2.4, 4, 6, 8, 10$ and starting with the solution obtained with each τ as initial guess for the problem with the next τ in the sequence. For running COLNEW as described in [10], trivial differential equations $P_i' = 0$, $i = 1, 2, 3$, have to be added for the parameters P_i. This results in a system of 14 first order equations. To complete the entire sequence of continuation steps takes 17.1 sec CPU-time and 645K storage.

Using COLPAR with direct treatment of parameters as mentioned in section 4, the above system of 11 equations needs no extension. The corresponding results are 12.3 sec CPU-time and 445K storage. The savings both in CPU-time and storage are mainly due to the reduced size of the linear systems solved. Generally, one can expect that the direct treatment of parameters pays off especially for problems with periodic boundary conditions as the effective size of the computational problem is halved.

The continuation code COLCON is constructed to expect a reasonable good solution approximation to start from. In this highly nonlinear example we use COLPAR to compute the solution for fixed parameter $\tau = 0$. Using this as initial guess for COLCON, the complete problem is solved in 19 continuation steps taking 22.2 sec CPU-time and 490K storage. The results indicate that it may be more expensive to trace a solution path by an automatic continuation procedure than to use a well chosen sequence of classical continuation steps.

Example 3. Consider a forced nonlinear oscillator modelled by the differential equation

$$\left(\frac{\tau}{2\pi}\right)^2 u'' + \frac{1}{25}\left(\frac{\tau}{2\pi}\right) u' - \frac{1}{5}u + \frac{8}{15}u^3 = \frac{2}{5}\cos 2\pi x$$

together with the periodic boundary conditions

$$u(0) = u(1)\,, \quad u'(0) = u'(1)\,.$$

Note that COLCON requires the problem to be formulated with separated boundary conditions. Thus, we introduce parameters p_1, p_2 and replace the above boundary conditions by

$$u(0) = p_1 \ , \qquad u'(0) = p_2 \ ,$$
$$u(1) = p_1 \ , \qquad u'(1) = p_2 \ .$$

This problem is suggested as test problem for continuation methods (see e.g. [27]). Difficulties in the numerical solution of this example arise from the presence of 16 turning points and the same number of bifurcation points along the main branch of periodic solutions in the range $\tau \in [0.06, 3]$. This branch is shown in figure 2. Starting the computation from $\tau = 3$, COLCON takes 212 continuation steps and 93.7 sec CPU-time.

Fig. 2: main branch of periodic solutions for example 3

Example 4. This problem describing the dynamic behavior of a tubular reactor is considered in [20]. The steady state equations result after transformation in two differential equations for relative reactant concentration and temperature given by

$$\frac{1}{\mathrm{Pe}_1} u_1'' = u_1' - \mathrm{Da} f(u) \ ,$$
$$\frac{1}{\mathrm{Pe}_2} u_2'' = u_2' - \beta(u_2 - u_{2w}) - \mathrm{Da} B f(u) \ ,$$

with the nonlinearity

$$f(u) = (1 - u_1) \exp\Big(\frac{u_2}{1 + u_2/\gamma}\Big)$$

describing a irreversible first order reaction, and boundary conditions

$$u_i'(0) - \mathrm{Pe}_i u_i(0) = 0\ , \quad u_i'(1) = 0\ , \quad i = 1, 2\ .$$

This nonlinear boundary value problem is known to exhibit solution profiles with extremely steep layers for the following set of parameters:

$$\begin{aligned} \mathrm{Pe}_1 &= 320\ , & \mathrm{Pe}_2 &= 100\ , \\ \beta &= 0.72\ , & u_{2w} &= -9.6\ , \\ B &= 16.8\ , & \gamma &= 16.9\ . \end{aligned}$$

As continuation parameter the Damköhler number $\mathrm{Da} \geq 0$ is chosen. A characteristic of this example is the moving of a steep layer region indicating the reaction zone with varying Damköhler number. This property proves to be a difficult feature for any adaptive mesh selection procedure involved in a global boundary value solver. Figure 3 shows a sample of typical solution profiles for u_1. Starting from $\mathrm{Da} = 0$, COLCON takes 253 continuation steps and 104.8 sec CPU-time to complete the run for the range $\mathrm{Da} \in [0, 1]$.

$\mathrm{Da}_1 = 0.4427$
$\mathrm{Da}_2 = 0.4423$
$\mathrm{Da}_3 = 0.4011$
$\mathrm{Da}_4 = 0.2937$
$\mathrm{Da}_5 = 0.1930$
$\mathrm{Da}_6 = 0.1297$

Fig. 3: typical solution profiles for example 4

References

[1] M. Aluko and H.-C. Chang,
PEFLOQ: An algorithm for the bifurcational analysis of periodic solutions of autonomous systems,
Comp.Chem.Eng. 8, 355-365 (1984).

[2] U. Ascher,
Collocation for two-point boundary value problems revisited,
SIAM J.Numer.Anal. 23, 596-609 (1986).

[3] U. Ascher and G. Bader,
A note on conditioning, stability and collocation matrices,
Tech.Rep. CMA-R16-86, Centre for Mathematical Analysis, Australian National University Canberra (1986).

[4] U. Ascher, J. Christiansen and R. D. Russell,
A collocation solver for mixed order systems of boundary value problems,
Math.Comput. 33, 659-679 (1979).

[5] U. Ascher, J. Christiansen and R. D. Russell,
Collocation software for boundary value ODE's,
ACM Trans.Math.Software 7, 209-222 (1981).

[6] U. Ascher and S. Jacobs,
On collocation implementation for singularly perturbed two-point problems,
Tech.Rep. 86-19, Dept. of Computer Science,
University of British Columbia (1986).

[7] U. Ascher, S. Pruess and R. D. Russell,
On spline basis selection for solving differential equations,
SIAM J.Numer.Anal. 20, 121-142 (1983).

[8] U. Ascher and R. D. Russell,
Reformulation of boundary value problems into "standard form",
SIAM Rev. 23, 238-254 (1981).

[9] G. Bader,
Numerische Behandlung von Randwertproblemen für Funktionaldifferentialgleichungen,
Thesis, University of Heidelberg (1983).

[10] G. Bader and U. Ascher,
A new basis implementation for a mixed order boundary value solver,
SIAM J.Sci.Stat.Comput. 8, 483-500 (1987).

[11] A. Ben-Israel and T.N.E. Greville,
Generalized Inverses: Theory and Applications,
John Wiley & Sons, New York (1974).

[12] C. deBoor and R. Weiss,
SOLVEBLOK: A package for solving almost block diagonal linear systems,
ACM Trans.Math.Softw. 6, 80-87 (1980).

[13] P. Deuflhard, B. Fiedler and P. Kunkel,
Efficient Numerical Pathfollowing beyond Critical Points,
SIAM J.Numer.Anal. 24, 912-927 (1987).

[14] P. Deuflhard, B. Fiedler and P. Kunkel,
Numerical Pathfollowing beyond Critical Points in ODE Models,
in Proc. Large Scale Scientific Computing, P. Deuflhard, B. Engquist (eds.), Birkhäuser, Boston (1987).

[15] P. Deuflhard and G. Heindl,
Affine invariant convergence theorems for Newton's method and extensions to related methods,
SIAM J.Numer.Anal. 16, 1-10 (1979).

[16] E.J. Doedel,
AUTO: A program for the automatic bifurcation analysis of autonomous systems,
Cong.Num. 30, 265-284 (1981).

[17] E.J. Doedel and J.P. Kernevez,
AUTO: Software for continuation and bifurcation problems in ordinary differential equations,
Appl.Math.Rep., California Institute of Technology (1986).

[18] J.P. Fink and W.C. Rheinboldt,
On the discretization error of parametrized nonlinear equations,
SIAM J.Numer.Anal. 20, 732-746 (1983).

[19] F. deHoog and R.M.M. Mattheij,
The role of conditioning in shooting techniques,
in Proc. Workshop on Numerical Boundary Value ODE's, U. Ascher, R.D. Russell (eds.), Birkhäuser, Boston (1985).

[20] K.F. Jensen and W.H. Ray,
The bifurcation behavior of tubular reactors,
Tech.Rep. MRC#2172, Math. Research Center, University of Wisconsin (1981).

[21] C. Kaas-Petersen,
Technique to trace curves of bifurcation points of periodic solutions of dissipative systems,
Tech.Rep., Centre for Nonlinear Studies, University of Leeds (1987).

[22] C. Kaas-Petersen,
PATH — User's Guide,
Tech.Rep., Centre for Nonlinear Studies, University of Leeds (1987).

[23] H.B. Keller,
Numerical solution of bifurcation and nonlinear eigenvalue problems,
in Proc. Applications of Bifurcation Theory, P.H. Rabinowitz (ed.), Academic Press (1977).

[24] M. Lentini and V. Pereyra,
PASVA4: An ODE boundary solver for problems with discontinuous interfaces and algebraic parameters,
Mat.Aplic.Comp. 2, 103-118 (1983).

[25] R. Mejia and J.L. Stephenson,
Numerical solution of multinephron kidney equations,
J.Comp.Phys. 32, 235-246 (1979).

[26] V. Pereyra,
Deferred corrections software and its application to seismic ray tracing,
Comp.Suppl. 5, 211-226 (1984).

[27] R. Seydel,
A continuation algorithm with step control,
in Proc. Numerical Methods for Bifurcation Problems, T. Küpper, H.D. Mittelmann, H. Weber (eds.), Inter.Ser.Numer.Math. 70, Birkhäuser, Basel (1984).

[28] J.L. Stephenson, R.P. Tewarson and R. Mejia,
Quantitative analysis of mass and energy balance in non-ideal models of the renal counterflow system,
Proc.Nat.Acad.Sci. U.S.A. 71, 1618-1622 (1974).