# Exploiting block triangular form for solving DAEs: reducing the number of initial values

J. Pryce, N. Nedialkov, G. Tan, R. McKenzie

**Abstract** The authors have written two codes to solve DAEs by structural analysis (SA). The first is written in C++ (DAETS) and deals with the solution of DAE initial value problems, using SA. Upon seeing how informative the SA could be the authors wrote DAESA (in MATLAB) to do only the structural analysis. These codes rely on exploiting the block triagular form (BTF) of a DAE, this paper explains how.

## 1 Overview of the SA method

Both DAETS and DAESA handle a DAE in $n$ state variables $x_j(t),\ j = 1,\dots,n$, of the general (possibly nonlinear) form

$$f_i(t,\ \text{the } x_j \text{ and derivatives of them}) = 0, \quad i = 1,\dots,n$$

which includes the case of a fully implicit or purely algebraic system. The numerical solution scheme used in DAETS is via Taylor series, in steps over a range, using automatic differentiation, analogous to a Taylor series method for ODEs.

The method starts by forming the $n \times n$ signature matrix $\Sigma = (\sigma_{ij})$, where

$$\sigma_{ij} = \begin{cases} \text{highest order of derivative to which } x_j \text{ occurs in } f_i \\ -\infty \text{ if it does not occur .} \end{cases} \tag{1}$$

A highest value transversal (HVT) is found, which comprises $n$ finite entries, one in each row and column of $\Sigma$, such that the total of these entries is maximised. We assume the problem is structurally well-posed, meaning that such a HVT exists.

J. Pryce
Cardiff University, Cardiff, United Kingdom. e-mail: j.d.pryce@cantab.net

N. Nedialkov
McMaster University, Hamilton, Canada. e-mail: nedialk@mcmaster.ca

G. Tan
McMaster University, Hamilton, Canada. e-mail: tang4@mcmaster.ca

R. McKenzie
Cardiff University, Cardiff, United Kingdom. e-mail: mckenzier1@cardiff.ac.uk

Valid non-negative integer valued offset vectors $\mathbf{c} = (c_1, \dots, c_n)$ and $\mathbf{d} = (d_1, \dots, d_n)$ are found, where valid means

$$d_j - c_i \geq \sigma_{ij} \;\text{ for all } i, j, \text{ with equality on a HVT,} \tag{2}$$

normalised by the constraint $\min_i c_i = 0$. There are unique element-wise smallest vectors $\mathbf{c}$, $\mathbf{d}$, which we call the canonical offsets; these are used from now on. However, any choice of valid offsets specifies a solution scheme by which to find Taylor coefficients in batches. Namely for stage $k = k^*, k^*+1, \dots$ where $k^* = -\max_j d_j \leq 0$, solve the equations

$$f_i^{(k+c_i)} = 0 \quad \forall i \text{ such that } k + c_i \geq 0 \tag{3}$$

for the variables

$$x_j^{(k+d_j)} \quad \forall j \text{ such that } k + d_j \geq 0\,. \tag{4}$$

Consider the simple pendulum DAE, in variables $x(t), y(t), \lambda(t)$ and parameters length $L$ and gravity $G$. We find its $\Sigma$ matrix, HVT (two, one marked by • the other by ∘) and offsets

$$\begin{aligned} 0 = A &= x'' + x\lambda \\ 0 = B &= y'' + y\lambda - G \\ 0 = C &= x^2 + y^2 - L^2 \end{aligned} \qquad \Sigma = \begin{array}{c} \\ A \\ B \\ C \\ d_j \end{array} \!\! \begin{array}{c} \begin{array}{ccc} x & y & \lambda \end{array} \\ \left[\begin{array}{ccc} 2^\bullet & -\infty & 0^\circ \\ -\infty & 2^\circ & 0^\bullet \\ 0^\circ & 0^\bullet & -\infty \end{array}\right] \\ \begin{array}{ccc} 2 & 2 & 0 \end{array} \end{array} \begin{array}{c} c_i \\ 0 \\ 0 \\ 2 \\ \\ \end{array}$$

This specifies a solution scheme of the form

| Stage $k$ | solve | for "batch" | kind |
|---|---|---|---|
| $-2$ | $0 = C = x^2 + y^2 - L^2$ | $x, y$ | 1 by 2 nonlinear |
| $-1$ | $0 = C' = 2xx' + 2yy'$ | $x', y'$ | 1 by 2 linear |
| $0$ | $0 = A, B, C''$ | $x'', y'', \lambda$ | 3 by 3 linear |
| $1$ | $0 = A', B', C'''$ | $x''', y''', \lambda'$ | 3 by 3 linear |

(5)

and so on for later stages. At each stage, treat items found previously as "known".

A key object is the $n \times n$ system Jacobian matrix $\mathbf{J}$, with entries

$$\mathbf{J}_{ij} = \frac{\partial f_i}{\partial x_j^{(d_j - c_i)}} = \begin{cases} \dfrac{\partial f_i}{\partial x_j^{(\sigma_{ij})}} & \text{if } d_j - c_i = \sigma_{ij} \\ 0 & \text{otherwise, including where } \sigma_{ij} = -\infty \end{cases}$$

The solution scheme succeeds [3] iff $\mathbf{J}$ is non-singular. E.g., for the pendulum,

$$\mathbf{J} = \begin{bmatrix} \partial A/\partial x'' & 0 & \partial A/\partial \lambda \\ 0 & \partial B/\partial y'' & \partial B/\partial \lambda \\ \partial C/\partial x & \partial C/\partial y & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 2x & 2y & 0 \end{bmatrix}$$

—nonsingular, since $\det(\mathbf{J}) = -2(x^2+y^2) = -2L^2 \neq 0$ at a consistent point.

## 2 An IV paradox and its explanation

### IVs, naive version

Solution scheme (3), (4) gives a simple recipe, in terms of the offsets, for what initial values (IVs) must be provided: namely these comprise all $x_j^{(r)}$ such that

$$\begin{cases} 0 \le r < d_j, & \text{if the DAE is quasilinear (see below), so } \sum_j d_j \text{ values in all;} \\ 0 \le r \le d_j, & \text{otherwise, so } n + \sum_j d_j \text{ values in all.} \end{cases} \tag{6}$$

E.g., the simple pendulum is quasilinear with $\mathbf{d} = (2,2,0)$, so the recipe is: in scheme (5), give IVs for $x, x'; y, y'$. *Note:* we call them IVs but they are really a set of *trial values* from which a near-by initial consistent point can always be computed in a numerically stable way. Since this DAE has 2 degrees of freedom (DOF) one could specify such a point choosing just 2 of these values, say $x, x'$, instead of 4; but any such choice is numerically unstable for some initial position of the pendulum.

### IVs, exploiting DAE structure

When applied to DAEs having structure, the simple recipe (6) leads to paradoxes. Namely, if one subsystem of the DAE drives another but is itself un-driven, (6) can make the driving subsytem need more IVs than it would, were it stand-alone. An example is the coupled two pendula system

$$\begin{aligned} 0 &= A = x'' + x\lambda, \\ 0 &= B = y'' + y\lambda - G, \\ 0 &= C = x^2 + y^2 - L^2, \\ 0 &= D = u'' + u\mu, \\ 0 &= E = v'' + v\mu - G, \\ 0 &= F = u^2 + v^2 - (L + cx')^2 \,. \\ & \quad c \text{ is a constant} \,. \end{aligned} \qquad \Sigma = \begin{array}{c|ccc|ccc|c} & x & y & \lambda & u & v & \mu & c_i \\ \hline A & 2 & & 0 & & & & 1 \\ B & & 2 & 0 & & & & 1 \\ C & 0 & 0 & & & & & 3 \\ \hline D & & & & 2 & & 0 & 0 \\ E & & & & & 2 & 0 & 0 \\ F & 1 & & & 0 & 0 & & 2 \\ \hline d_j & 3 & 3 & 1 & 2 & 2 & 0 & \end{array}\,, \tag{7}$$

where a blank in the $\Sigma$ means $-\infty$. The first three equations model one pendulum, the second three model a second (coupled) pendulum. Pendulum 1 drives pendulum 2 in that $x'$ appears in equation $F$ (horizontal velocity of pendulum 1 affects length of pendulum 2), see entry in position $(F,x)$ of $\Sigma$; but there's no reverse influence.

Yet, Pendulum 1's offsets have increased by 1 while Pendulum 2's are unchanged. Hence, pendulum 1 now needs IVs for $x'', y''$ and $\lambda$, which as a stand-alone system it did not. Clearly something is wrong here.

To clarify what is going on, consider Table 1, which lists the stages in the uncoupled and coupled system solution processes.

With coupling, you can't find $u, v$ at same time as $x, y$, because $u, v$ satisfy $0 = F = u^2 + v^2 - (L + cx')^2$, which uses $x'$ which hasn't been found yet. And so on. The raised offsets say, in effect, "Shift pendulum 1 a stage earlier, so its derivatives

| Uncoupled pendula | |
|---|---|
| $k$ | find |
| | |
| $-2$ | $x,y, \quad u,v$ |
| $-1$ | $x',y', \quad u',v'$ |
| $0$ | $x'',y'',\lambda, \quad u'',v'',\mu$ |

(a)

| Coupled pendula | |
|---|---|
| $k$ | find |
| $-3$ | $x,y$ |
| $-2$ | $x',y', \quad u,v$ |
| $-1$ | $x'',y'',\lambda, \quad u',v'$ |
| $0$ | $x''',y''',\lambda', \quad u'',v'',\mu$ |

(b)

**Table 1** Stages in solving the uncoupled and coupled 2-pendula systems.

| Uncoupled pendula | | |
|---|---|---|
| | | |
| $k$ | find in parallel | |
| $-2$ | $x,y$ | $u,v$ |
| $-1$ | $x',y'$ | $u',v'$ |
| $0$ | $x'',y'',\lambda$ | $u'',v'',\mu$ |

(a)

| Coupled pendula | | | |
|---|---|---|---|
| local $k^*$ | | global $k$ | |
| $=k+1$ | find | | *then* find |
| $-2$ | $x,y$ | $-3$ | |
| $-1$ | $x',y'$ | $-2$ | $u,v$ |
| $0$ | $x'',y'',\lambda$ | $-1$ | $u',v'$ |
| $1$ | $x''',y''',\lambda'$ | $0$ | $u'',v'',\mu$ |

(b)

**Table 2** Explanation in terms of local stage counter $k^*$ for first pendulum.

are ready when pendulum 2 needs them". Their apparent effect of increasing the number of IVs needed is mistaken, and due to (6) not telling the whole story.

The explanation comes from considering pendulum 1 to have a *local stage counter* $k^* = k+1$. (Pendulum 2 also has one, but it is the same as the global counter $k$.) Introducing $k^*$ into Table 1 gives Table 2.

In the coupled system *each global stage* solves for pendulum 1 data first and then uses this to solve for pendulum 2 data. Relative to its *local* stages, pendulum 1 has local offsets $(\widehat{\mathbf{c}}, \widehat{\mathbf{d}}) = (0,0,2;2,2,0)$, the same as when it is stand-alone. And it is clear from the solution scheme in Table 2(b) that pendulum 1 requires the same IVs $x,y,x',y'$ as when it is stand-alone—which is as it should be.

## 3 The benefits of BTF

The paradox in the previous section arose because the DAE had a *nontrivial block triangular form, BTF*, and the explanation came from recognizing this.

A BTF is a property of a *sparsity pattern*, in this case a subset $S$ of $\{1,2,\dots,n\}^2$, the $n \times n$ positions in the signature matrix or Jacobian. Write a $\times$ in the positions $(i,j)$ that belong to $S$, and leave the others blank. If we can permute rows and columns so $S$ looks like the example opposite, then we have put $S$ in (upper) block triangular form.

$$\left[\begin{array}{cc|ccc|ccc}
\times & \times & & \times & & \times & & \times \\
\times & \times & & \times & \times & & & \\
\hline
& & \times & \times & & & \times & \\
& & \times & \times & \times & & & \\
& & \times & & \times & \times & & \times \\
\hline
& & & & & & \times & \times \\
& & & & & \times & \times & \\
& & & & & \times & & \times
\end{array}\right],$$

Namely, there are square diagonal blocks that are themselves irreducible (cannot be split into a finer BTF) and the below-diagonal blocks are empty. Such a BTF can always be found if $S$ is structurally non-singular, i.e. contains some transversal; and it is unique up to ordering of the diagonal blocks [1].

In the DAE structural analysis context we have two choices of sparsity pattern to use. A natural one is the sparsity pattern of $\Sigma$:

$$S = \{(i,j) \mid \sigma_{ij} > -\infty\}$$

We call the BTF based on this sparsity pattern the *coarse BTF*. A more informative BTF is found by using the sparsity pattern of the Jacobian $\mathbf{J}$:

$$S_0 = S_0(\mathbf{c},\mathbf{d}) = \{(i,j) | d_j - c_i = \sigma_{ij}\}$$

We call the resulting BTF the *fine BTF* since $S_0 \subseteq S$ and it usually gives a strict refinement of the BTF that $S$ generates. Though $S_0$ depends on the $(\mathbf{c},\mathbf{d})$ chosen, the resulting set of blocks is independent of $(\mathbf{c},\mathbf{d})$ up to possible reordering [2].

Each diagonal block of the BTF defines a subsystem of the DAE: its equations (rows) and variables (columns) form a free-standing DAE, if one counts any other variables that occur in these equations as external driving functions.

Let there be $p$ blocks of sizes $N_1,\dots,N_p$ summing to $n$. As we are using upper BTF, each block depends only on those below it, the bottom block being independent of all others. It can be proved that for the *fine* BTF, there is a well defined notion that the $\ell$th block has a *local stage counter*

$$k_\ell = k + K_\ell, \quad \ell = -1,\dots,p,$$

where $k$ is the global stage counter and $K_\ell$ is an integer $\geq 0$, the *lead time* of that block. The *local offsets* given by

$$\widehat{c_i} = c_i - K_\ell, \qquad \widehat{d_j} = d_j - K_\ell$$

are thus the offsets of the $\ell$th block as a free-standing DAE in the sense described above.

As an example, the coupled two pendulum (7) has the following fine BTF:

$$\Sigma = \begin{array}{c|ccc|ccc|cc}
 & u & v & \mu & x & y & \lambda & c_i & \widehat{c_i} \\
\hline
F & 0 & 0 & & 1 & & & 2 & 2 \\
E & & 2 & 0 & & & & 0 & 0 \\
D & 2 & & 0 & & & & 0 & 0 \\
\hline
C & & & & 0 & 0 & & 3 & 2 \\
B & & & & & 2 & 0 & 1 & 0 \\
A & & & & 2 & & 0 & 1 & 0 \\
\hline
d_j & 2 & 2 & 0 & 3 & 3 & 1 & & \\
\widehat{d_j} & 2 & 2 & 0 & 2 & 2 & 0 & &
\end{array}, \qquad
\mathbf{J} = \begin{array}{c|ccc|ccc|}
 & u & v & \mu & x & y & \lambda \\
\hline
F & 2u & 2v & & \xi & & \\
E & & 1 & v & & & \\
D & 1 & & u & & & \\
\hline
C & & & & 2x & 2y & \\
B & & & & & 1 & y \\
A & & & & 1 & & x
\end{array}$$

where $\xi = -2c(L{+}cx')$. Hence we have a lead time $K_1 = 0$ for the first block and $K_2 = 1$ for the 2nd block.

## 4 Initial values revisited

Initial (or trial) values are numbers a user must supply, to specify an initial consistent point of the DAE from which to propagate a numerical solution. Of course one would like to demand as few IVs as possible. The BTF theory outlined above (see [2] for more detail) shows the IVs needed are determined, not by the naive recipe (6) using global offsets, but by the corresponding formula using local offsets $\widehat{d}_j$. Namely within a block they comprise all $x_j^{(r)}$ such that:

$$\begin{cases} 0 \le r < \widehat{d}_j, & \text{if the block is quasilinear (see below);} \\ 0 \le r \le \widehat{d}_j, & \text{otherwise;} \end{cases} \tag{8}$$

where quasilinear (QL) means the equations in the block are jointly linear in their leading derivatives, $x_j^{(d_j-c_i)}$. With blocks of size $N_\ell$ and lead times $K_\ell$, $\ell = 1, \ldots, p$, this needs $\sum_{\ell=1}^{p} N_\ell K_\ell$ fewer IVs than recipe (6). This is before considering an important effect: experience suggests that small blocks are far more likely to be QL than are large ones. As a result, BTF and QL analysis work together to reduce the number of IVs needed.

## 5 Conclusions and future work

Structural analysis based on BTF promises to reduce the work of solving a DAE IVP numerically, especially by simplifying the linear algebra involved. As yet, DAETS does not use such methods, whereas some other simulation tools based on DAE models do use some form of SA to reduce work.

The coarse rather than fine, BTF can be exploited to solve IVPs in parallel by pipelining the solution process block-wise: this also deserves study.

We believe that our theory is the most powerful available, for doing this sort of analysis, but other tools are currently ahead of DAETS in putting SA into practice. Thus we wish to study other simulation tools' use of SA in numerical solution.

## References

[1] Pothen A, Fan CJ (1990) Computing the block triangular form of a sparse matrix. ACM Transactions on Mathematical Software 16(4):303–324, URL `http://doi.acm.org/10.1145/98267.98287`

[2] Pryce J, Nedialkov NS, Tan G (2013) DAESA — a Matlab tool for structural analysis of DAEs: Theory. To appear in ACM Trans. Math. Software

[3] Pryce JD (2001) A simple structural analysis method for DAEs. BIT 41(2):364–394