Solving Constrained Differential-Algebraic Systems Using Projections

R. J. Hanson, Visual Numerics, Inc. and F. T. Krogh, Math à la Carte, Inc.

A White Paper by Visual Numerics, Inc.

November 2008

Visual Numerics

Visual Numerics, Inc.

2500 Wilcrest Drive, Suite 200

Houston, TX 77042 USA

www.vni.com

Solving Constrained Differential-Algebraic Systems Using Projections

R. J. Hanson, Visual Numerics, Inc. and F. T. Krogh, Math à la Carte, Inc.

by Visual Numerics, Inc.

Publishing History:

November 2008

TABLE OF CONTENTS

Summary

This is a report on software additions to the Differential-Algebraic Equation (DAE) solver code **DASSL** made during our sporadic researches of 2001-2008. The code modifications are made to deal with "constraint or invariant drift" for DAE systems. These terms are defined below. We also discuss the problem of computing consistent initial conditions for derivative values of DAE systems.

We have installed changes to the basic **DASSL** integration algorithm and provided a modified code **DDASLX, [**6**]**. Our expectation is that **DDASLX** expands the range of application models that can be solved with **DASSL**.

When constraints are present we perturb each step taken by the integrator using a projection algorithm. This change to a step is typically much smaller than the accepted error tolerance. It forces the solution back onto constraints or invariants that the mathematical solution must satisfy. Without this projection step, or some equivalent intervention, the numerical solution may become "non-physical" because certain natural constraints of the system are no longer satisfied. Our primary intent in this report is to give a description of the projection algorithm and draw attention to practical details and limitations.

We also discuss methods for getting started. This is primarily an issue for Index 1 systems where the initial values of the solution are given but the initial derivative values are unknown.

Introduction

The notation and algorithms discussed here are primarily taken from [1], pp. 117. The starting point for use of the DASSL software is the initial value problem

$$F\left(t, y, y'\right) = 0, \quad y' = \frac{dy}{dt},$$
$$y\left(t_0\right) = y_0$$
$$y'\left(t_0\right) = y'_0$$

Equation 1

Each of F, y, y' are $N-$ dimensional vectors. For many dynamical modeling problems the stated system has been transformed using the system mechanics to a first order system, as shown. In addition to substituting for higher order derivatives to obtain a first order system, there may be changes made to the components of F that reduce the *index* of the system to the value one or value zero. In Appendix II we present an algorithm for computing some of the initial values for y'_0 when the index of the system has the value one or zero.

A discussion about expressing a given differential equation in this standard form, involving derivatives of order higher than the value one, is given for example in [2], pp. 5-10. A further change is often made that replaces constraints for a system by their total derivatives. This change is done because it reduces the *index* of the problem. After a sufficient number of replacements of the constraints by their derivatives, a system is obtained such that it has index of value one or zero. Then the **DASSL** software may be used to numerically solve the system.

But reducing the index in this manner does not force the resulting intermediate constraints to be satisfied, [1], p. 141. This has been well-known for a long time and various methods have been proposed so that the constraints remain satisfied. Our results add to this discussion.

An elementary example that illustrates many of these issues is the planar pendulum model in Cartesian coordinates

$$x'' = -\lambda x, \quad x'' = \frac{d^2x}{dt^2},$$
$$y'' = -\lambda y - g$$
$$\frac{1}{2}\left(x^2 + y^2 - L^2\right) = 0$$

Equation 2

The parameters g, L are the gravity constant and the length of the pendulum. The function $\lambda(t)$ is the tension in the wire or thin rod. There is a unit point mass at the end of the rod.

Changing Eq. 2 to standard form is done by defining

$$y_1 = x$$
$$y_2 = y$$
$$y_3 = y_1'$$
$$y_4 = y_2'$$
$$y_5 = \lambda$$

Equation 3

This leads to the first intermediate system of index 3

$$y_1' - y_3 = 0$$
$$y_2' - y_4 = 0$$
$$y_1 y_5 + y_3' = 0$$
$$y_2 y_5 + y_4' + g = 0$$
$$\frac{1}{2}\left(y_1^2 + y_2^2 - L^2\right) = 0$$

Equation 4

We will include the dynamic total energy constraint for this problem – Kinetic energy + Potential energy = Constant. In terms of the problem variables and the initial conditions,

$$\frac{1}{2}\left(y_1'^2 + y_2'^2\right) + \left(L + y_2\right) g = Lg$$
$$y_1(0) = L, \quad y_j(0) = 0, \; j = 2,3,4,5$$

The system total energy equation comes from noting that the amount of work done to raise the unit mass or the bob of the pendulum from rest $(y_1, y_2) = (0, -L)$ to the horizontal position $(y_1, y_2) = (L, 0)$ is Lg . Substituting for the derivatives and canceling the constant term Lg leads to the equivalent total energy constraint and initial conditions:

$$\frac{1}{2}\left(y_3^2 + y_4^2\right) + y_2 g = 0$$
$$y_1(0) = L, \quad y_j(0) = 0,\ j = 2,3,4,5$$

Equation 5

The index of the system reduces from 3 to 2 by replacing the length constraint of Eq. 4 by its derivative

$$\frac{d}{dt}\frac{1}{2}\left(y_1^2 + y_2^2 - L^2\right) =$$
$$y_1 y_1' + y_2 y_2' =$$
$$y_1 y_3 + y_2 y_4 = 0$$

Equation 6

The index reduces from 2 to 1 by replacing the resulting constraint of Eq. 6 by its derivative

$$\frac{d}{dt}\left(y_1 y_3 + y_2 y_4\right) =$$
$$y_1 y_3' + y_1' y_3 + y_2 y_4' + y_2' y_4 =$$
$$-y_1 y_5 y_1 + y_3^2 - \left(y_2 y_5 + g\right) y_2 + y_4^2 =$$
$$-y_5\left(y_1^2 + y_2^2\right) + y_3^2 + y_4^2 - g y_2 =$$
$$-y_5 L^2 + y_3^2 + y_4^2 - g y_2 = 0$$

Equation 7

The index reduces from 1 to 0 by a last differentiation of Eq. 7

$$\frac{d}{dt}\left(-y_5 L^2 + y_3^2 + y_4^2 - g y_2\right) =$$
$$-y_5' L^2 + 2\left(y_3 y_3' + y_4 y_4'\right) - g y_2' =$$
$$-y_5' L^2 + 2\left\{y_3\left(-y_1 y_5\right) + y_4\left(-y_2 y_5 - g\right)\right\} - g y_2' =$$
$$-y_5' L^2 - 2 y_5\left(y_1 y_3 + y_2 y_4\right) - 2 y_4 g - g y_2' =$$
$$-y_5' L^2 - 2 y_4 g - g y_2' =$$
$$-y_5' L^2 - 3 y_4 g = 0$$

Equation 8

The index 1 system in standard form is thus

$$
\begin{aligned}
y_1' - y_3 &= 0 \\
y_2' - y_4 &= 0 \\
y_1 y_5 + y_3' &= 0 \\
y_2 y_5 + y_4' + g &= 0 \\
-y_5 L^2 + y_3^2 + y_4^2 - g y_2 &= 0
\end{aligned}
$$

Equation 9

with 2 constraints

$$
\begin{aligned}
\frac{1}{2}\left(y_1^2 + y_2^2 - L^2\right) &= 0 \\
y_1 y_3 + y_2 y_4 &= 0
\end{aligned}
$$

Equation 10

The index 0 system in standard form is

$$
\begin{aligned}
y_1' - y_3 &= 0 \\
y_2' - y_4 &= 0 \\
y_1 y_5 + y_3' &= 0 \\
y_2 y_5 + y_4' + g &= 0 \\
-y_5' L^2 - 3 y_4 g &= 0
\end{aligned}
$$

Equation 11

with 3 constraints

$$
\begin{aligned}
\frac{1}{2}\left(y_1^2 + y_2^2 - L^2\right) &= 0 \\
y_1 y_3 + y_2 y_4 &= 0 \\
-y_5 L^2 + y_3^2 + y_4^2 - g y_2 &= 0
\end{aligned}
$$

Equation 12

This index 0 system of Eq. 11 is also written as the system of ordinary differential equations

$$
\begin{aligned}
y_1' &= y_3 \\
y_2' &= y_4 \\
y_3' &= -y_1 y_5 \\
y_4' &= -y_2 y_5 - g \\
y_5' &= -\left(3g / L^2\right) y_4
\end{aligned}
$$

Equation 13

In vector form we write Eq. 13 as $y' = F_0(y)$. The value of the initial index - in this case 3 - is defined by noting the number of differentiations it took to arrive at a system of ordinary differential equations.

For purposes of studying the variation equation $\frac{\partial y}{\partial y_0}$, recall, e. g. [3], p. 94, that it satisfies the linear ODE system $\left(\frac{\partial y}{\partial y_0}\right)' = \frac{\partial y'}{\partial y}\frac{\partial y}{\partial y_0}, \quad \left.\frac{\partial y}{\partial y_0}\right]_{t=t_0} = I$. For the pendulum ODE problem we have

$$\frac{\partial y'}{\partial y} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -y_5 & 0 & 0 & 0 & -y_1 \\ 0 & -y_5 & 0 & 0 & -y_2 \\ 0 & 0 & 0 & -3g/L^2 & 0 \end{bmatrix}$$

Equation 14

A direct calculation, e.g. [5], gives the characteristic equation of Eq. 14

$$\det\left(\frac{\partial y'}{\partial y} - \lambda I\right) = -\lambda\left(\lambda^4 + u\lambda^2 + v\right)$$
$$u = 2y_5 + \left(3g/L^2\right)y_2, \quad v = y_5\left(y_5 + \left(3g/L^2\right)y_2\right)$$

Equation 15

The roots of this quintic, from [5], are $\lambda = 0, \quad \lambda^2 = -y_5, \quad \lambda^2 = \left(3g/L^2\right)y_2 - y_5$

Therefore the eigenvalues of $\frac{\partial y'}{\partial y}$ all remain on the imaginary axis. This is true since $y_2 \le 0$ and $y_5 \ge 0$. These inequalities show the characteristic equation has one constant root at the value zero and two dynamic pairs of purely complex-conjugate roots. In terms of the variational equations for the initial conditions, small variations are generally neither magnified nor damped.

There are critical points regarding stability when the pendulum returns to the horizontal: $y_1 = \pm L, \quad y_2 = 0, \quad y_5 = 0$. There the Jacobian matrix $\frac{\partial y'}{\partial y}$ has 0 as an eigenvalue of algebraic multiplicity 5. But the rank is 3. Therefore the Jordan normal form at these critical points has

two blocks, one of size 4 by 4 with a super-diagonal having 3 nonzero entries of value 1, and the other block of dimension 1. At these critical points the local approximate solution to the variation equations will be instantaneously amplified.

A potential defect in the **DASSL** code is that the constraints will not necessarily remain satisfied within the error tolerance required of the solution components. The authors of [1] note this on p. 141. They also mention that reduction of index, by repeated differentiation, should be used with caution. But since **DASSL** requires that the index be 1 or 0, this appears to be a warning without stated alternatives – at least if one wants to use that software.

In [1], p. 157, Table 6.2.2 gives the sizes of the residuals for the three constraints when integrating the index 1 system. The integration is done on the interval $[0,1]$. On this interval the residuals remain acceptably small. Our experiments show that when integrating on longer intervals, say $[0,10000]$, the constraint residuals are much larger than the requested error tolerances. This can lead to simulations that do not track the physical process they are attempting to model.

The DDASLX Problem Statement

Our starting point is a *modified statement of the initial value problem*

$$F(t,y,y') = 0, \quad y' = \frac{dy}{dt},$$
$$y(t_0) = y_0$$
$$y'(t_0) = y'_0$$
$$G(t,y) = 0$$

Equation 16

Each of F, y, y' are $N-$ dimensional vectors, and G is $M-$ dimensional – the constraints of the problem. These constraints are all those meaningful to the physical model of the problem. In the simple pendulum example we will use the derivatives leading to both index 1 and index 0 problems, including the total energy constraint.

The user interface for **DASSL** requires an evaluation subprogram that computes F. Partial derivative matrices $\frac{\partial F}{\partial y} \& \frac{\partial F}{\partial y'}$ are also needed. There is an option to approximate these using divided one-sided differences. For the **DDASLX** problem statement the evaluation program must also provide $G \& \frac{\partial G}{\partial y}$. One can approximate $\frac{\partial G}{\partial y}$ using divided differences, but this step must be performed within the evaluation subprogram.

Our numerical addition to **DASSL** is that the integration first proceeds in a single step without using G. Then after a successful step there is an additional change to that step such that $G(t,y) = 0$ is re-gained. This secondary change normally does not introduce abrupt changes to y that would result in re-starting the integration over this or the next step. But

without this secondary step the required constraints might eventually become unacceptably large, and result in a physically impossible simulation.

The Projection Algorithm

During the integration of the system, changes in the approximate solution y are made in two places for **DDASLX**. This is done first when computing consistent initial conditions and then routinely as the predictor-corrector algorithm proceeds step-by-step. Whenever the integration algorithm has made a move and changed y we move onto the constraints. Writing the desired change or projected solution as $y - dy$, it is required that $G(t, y - dy) = 0$. This is computed with a single Newton method step: Solve the linear system $Cdy = G(t, y), \quad C = \dfrac{\partial G(t, y)}{\partial y}$ and then compute the Newton update $y \leftarrow y - dy$. These linear systems are typically underdetermined and always rank deficient, so a particular solution is chosen. A minimum length solution is computed using the weighted l_2 norm defined by the weights $WT_i = RTOL|y_i| + ATOL, \quad i = 1, \ldots, N$. See [1], p. 131. With $dy = W^{-1/2}du$, where the diagonal matrix W has diagonal terms WT_i, the scaled system $CW^{-1/2}du = G(t, y)$ is solved. In other words we move onto the constraints with a step that minimizes $\left(\sum_{i=1}^{N} WT_i^{-1} du_i^2 \right)^{1/2}$. There were other choices for the diagonal weighting matrix that we tried, namely W^{-1}. Our experiments suggest that using the square-root matrix weight is the most effective.

Any method that solves the weighted linear system associated with this form of the Newton method can be considered. It is desirable that the method be efficient and accurate. Our primary choice consists of an accurate method based on row pivoting and plane rotations. We did not concern ourselves with overall efficiency.

To present this we change the notation and consider a rectangular system $Au = b$. We use row pivoting to maximize the l_2 norm of each row of the partially factored matrix, on and to the right of the pivot row. Then row interchanges are applied to the factored matrix and applied to the right-hand side. Next plane rotations are used to eliminate non-zero values in that row, rotating each non-zero to the main diagonal. It is necessary to save the plane rotations and we use the feature implemented in the BLAS-1 code **DROTG** for this, [4], p. 314. To summarize, we compute a permutation matrix P and a product of plane rotations Q such that $PAQ = [L_{M \times K} : 0]$, where L is lower triangular. We use those diagonal terms of L that are non-zero to determine the rank K of the system. Next we use the first K rows of L and the permuted right-hand side Pb to solve for K non-zero values of v, $u = Qv$. The remaining values of v are given the value zero. Then the weighted minimum length solution is given by $u = Qv$. It is here that the plane rotations are re-constructed and used in reverse order of their construction.

Other methods that could be considered for this projection step, when $K = M$, are to form normal equations and solve $AA^T v = b, \quad u = A^T v$. This coefficient matrix can be factored using Cholesky decomposition. It is advisable to carry more precision for the factorization than is requested in the integration step.

For sparse and large systems, again with $K = M$, one can also solve the compound square system $\begin{bmatrix} A & 0 \\ -I & A^T \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$ for the solution u of minimum l_2 norm. This system is still sparse, but larger in size than $Au = b$. The v vector is not the same as in previous uses. Its role is to force the particular solution of $Au = b$ to be the minimum length solution. This method can be considered if there is a sparse solver required for the solution of (5.2.11), [1], applied to the iteration matrix there called G .

At the initial step both **DASSL** and **DDASLX** solve for consistent initial conditions. Thus for y_0 given but y_0' unspecified, there are initial Newton method solution steps that solve for y_0' . Following this initial phase we project onto the constraints, as described, and accept the move only if $|dy_i| \le WT_i, \quad i = 1, \ldots, N$. Otherwise there is an error condition. After each predictor-corrector step there is a projection onto constraints but with some additional logic to scale the size of the projection step, should this be necessary. This scaling limits the weighted norm of dy to not exceed the weighted norm of the step de that results from the corrector, [1], Section 5.2.

Constraining the Pendulum Model

Figure 1 shows that for the combined relative and absolute tolerance 10^{-NDIG} , NDIG=6,8, and 10, the y_2 value becomes positive. But this is physically impossible if no energy is added to the system. With the initial condition $y_2(0) = 0$, it is clear that $y_2(t) \le 0, \quad t > 0$. Otherwise the pendulum appears to rise above the horizontal without additional work.

We consider the index 1 and index 0 constraints separately. For the index 1 problem we have, ignoring the total energy constraint,

$$G_1(t, y) = \begin{bmatrix} \frac{1}{2}\left(y_1^2 + y_2^2 - L^2\right) \\ y_1 y_3 + y_2 y_4 \end{bmatrix}$$

Equation 17

And with the energy constraint of Eq. 9 included this becomes

$$\hat{G}_1(t,y)=\begin{bmatrix}\frac{1}{2}\left(y_1^2+y_2^2-L^2\right)\\ y_1y_3+y_2y_4\\ \frac{1}{2}\left(y_3^2+y_4^2\right)+y_2g\end{bmatrix}$$

Equation 18

For the index 0 problem, exclusive of the energy constraint, we have

$$G_0(t,y)=\begin{bmatrix}\frac{1}{2}\left(y_1^2+y_2^2-L^2\right)\\ y_1y_3+y_2y_4\\ -y_5L^2+y_3^2+y_4^2-gy_2\end{bmatrix}$$

Equation 19

And with total energy constrained we have

$$\hat{G}_0(t,y)=\begin{bmatrix}\frac{1}{2}\left(y_1^2+y_2^2-L^2\right)\\ y_1y_3+y_2y_4\\ -y_5L^2+y_3^2+y_4^2-gy_2\\ \frac{1}{2}\left(y_3^2+y_4^2\right)+y_2g\end{bmatrix}$$

Equation 20

An apparently natural second set of additional constraints was investigated by noting that the total derivative $\frac{dG_0}{dt}=\frac{\partial G_0}{\partial y}y'=\frac{\partial G_0}{\partial y}F_0(y)=0$. This leads to the matrix-vector product

$$\frac{\partial G_0}{\partial y}F_0=\begin{bmatrix}y_1 & y_2 & 0 & 0 & 0\\ y_3 & y_4 & y_1 & y_2 & 0\\ 0 & -g & 2y_3 & 2y_4 & -L^2\end{bmatrix}\begin{bmatrix}y_3\\ y_4\\ -y_1y_5\\ -y_2y_5-g\\ -\left(3g/L^2\right)y_4\end{bmatrix}$$

This gives

$$\frac{\partial G_0}{\partial y} F_0 = \begin{bmatrix} y_1 y_3 + y_2 y_4 \\ -y_5 L^2 + y_3^2 + y_4^2 - g y_2 \\ 2\left(y_1 y_3 + y_2 y_4\right) \end{bmatrix}$$

Thus adding the additional constraint $\frac{dG_0}{dt} = 0$ is redundant. Each resulting component constraint is already found within the constraints $G_0\left(t, y\right) = 0$. Including the energy constraint in this second order term adds no information either.

Deriving the constraints for an application by repeated differentiation to reduce the index value is straightforward and tedious. Tools such as formula manipulation packages will help here. The results clearly favor including the energy constraint for the pendulum problem. Figure 5 shows that when not using the energy constraint the pendulum integration exhibits a quadratic error growth. Figure 6 shows that when energy is constrained the errors are much smaller and grow linearly. Linear error growth is partially due to rounding errors and the non-diagonal Jordan normal form that arises from the linear Jacobian matrix at extreme points $y = \left(\pm L, 0, 0, 0, 0\right)^T$.

Closing Remarks

- We believe that including constraint information together with the DAE equations will allow a larger class of problems to be successfully integrated than without the constraints.
- There is no requirement that additional constraints be provided, so this feature can be ignored if there is no need for it.
- Integrating an index 0 problem, instead of an index 1 problem, may require a significant amount of preliminary algebraic computation. Yet the results for the pendulum problem, Figures 3-4, show this choice to yield the least amount of work for evaluations of the functions, partial derivatives and some of the linear algebra. We expect this will generally be true.
- In the example considered the energy is an invariant of the system. Other problems may have other invariants. Knowing what these are and modeling them may be difficult in practice. It is not clear this is always useful. But comparing Figure 5 with Figure 6 shows that this is important for the long-term integration of the pendulum model.

- We installed an optional smoothing algorithm [7] in **DDASLX** for the adaptive step-size selection algorithm found in **DASSL**. Our default choice is H211b, with $b = 4$, since this performed well in Söderlind's tests.
- We added functionality to **DDASLX** that allows for the usage of any user-defined linear equation solver. This interface, including the function and derivative evaluations, is provided with "reverse communication."

Finally we find the modified code **DDASLX** is awkward and hard to maintain. The word "shoehorned" is appropriate! There is the opportunity to rework the integration algorithm and maintain the constraints and step-size smoothing. But we did not take the opportunity.

References

1. Brenan, K. E., Campbell, S. L., Petzold, L. R., *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM Publications, Classics in Applied Mathematics, Philadelphia, PA, (1996).
2. Shampine, L. F. *Numerical Solution of Ordinary Differential Equations*, Chapman and Hall, New York, NY, (1994).
3. Hartman, P. *Ordinary Differential Equations*, John Wiley & Sons, New York, NY, (1964).
4. Lawson, C. L., Hanson, R. J., Kincaid, D. R., Krogh, F. T. Basic Linear Algebra Subprograms for Fortran Usage, *ACM. Trans. Math. Soft.* **5**(3), (1979).
5. http://www.quickmath.com QuickMath is an automated on-line service that performs certain symbolic mathematical calculation problems using the internet.
6. http://mathalacarte.com/c/math77_head.html Software Library, Math77, Chapter 14.3.
7. Söderlind, G. "Digital Filters in Adaptive Time-Stepping," *ACM. Trans. Math. Soft.* **29**(1), pp. 1–26.
8. Krogh, F. T. , Hanson, R. J. "Getting Consistent Initial Conditions for a DAE," Math à la Carte, Rept. 2008-2, http://mathalacarte.com/fkrogh.

Appendices

Appendix I

These tables and figures use the tolerances $WT_i = RTOL_k \left| y_i \right| + ATOL_k, \quad i = 1,\ldots,5$, with values $RTOL_k = ATOL_k = 10^{-NDIG_k}, \quad NDIG_k = 5,6,8,10$. The "true" solution, for purposes of stating the errors in Figures 1 and 2, was obtained by integrating with $NDIG = 11$, and constraining total energy.

	Index 1		Index 0	
NDIG	y_1	y_2	y_1	y_2
5	0.1514	-0.3142	-0.0451	0.3555
6	-0.0458	0.3015	-0.0459	0.3126
8	-7.2162D-04	0.0024	-0.0017	0.0059
10	-5.1636D-06	1.7319D-05	-1.2437D-05	4.1718D-05

Figure 1. Long Term Position Errors of the Pendulum, t = 1000

	Index 1		Index 0	
NDIG	y_1	y_2	y_1	y_2
5	0.0033	-0.0108	-0.002	0.0069
6	1.6784D-04	-5.6240D-04	-9.5761D-05	3.2136D-04
8	5.2148D-07	-1.7492D-06	-4.8723D-07	1.6341D-06
10	2.1445D-09	-7.1927D-09	-1.6767D-09	5.6236D-09

Figure 2. Long Term Position Errors of the Pendulum, t = 1000

$NDIG = 10$	#F evaluations	#Decompositions	# Projection Solves	Time (S)
Index 1	12,217,441	24,210	4,638,767	80.3
Index 0	9,277,646	54	4,638,770	80.0

Figure 3. Work for integrating index 1 and index 0 systems, $[0,10000]$
Only Derivative Constraints Used

$NDIG = 10$	#F evaluations	#Decompositions	# Projection Solves	Time (S)
Index 1	12,393,836	24,208	4,638,759	81.0
Index 0	9,281,550	549	4,640,488	79.8

Figure 4. Work for integrating index 1 and index 0 system, $[0,10000]$ Derivative and Total Energy Constraints Used

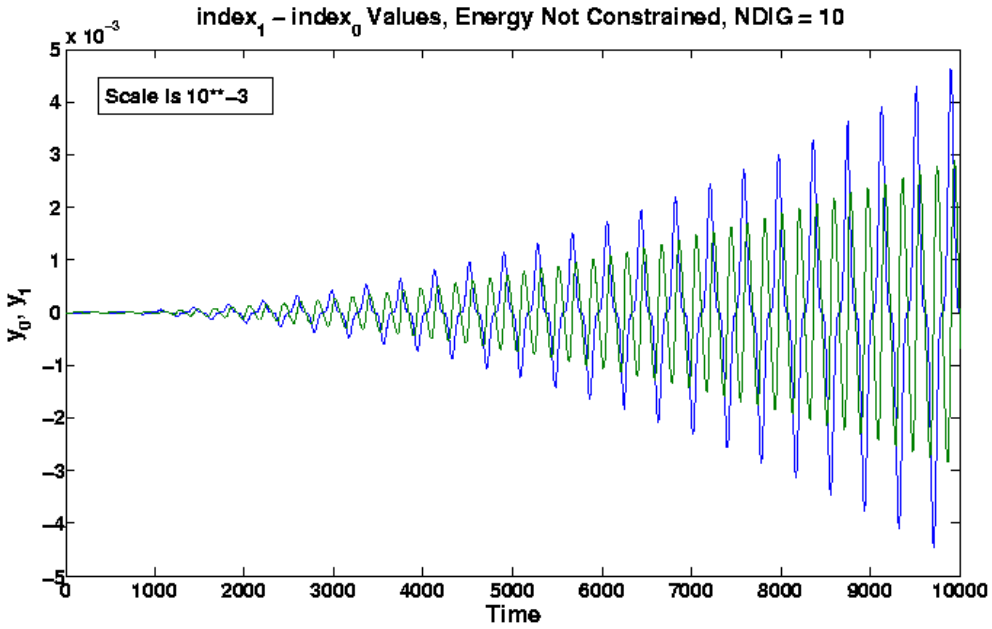


Figure 5. Quadratic Error Growth when Total Energy is Not Constrained

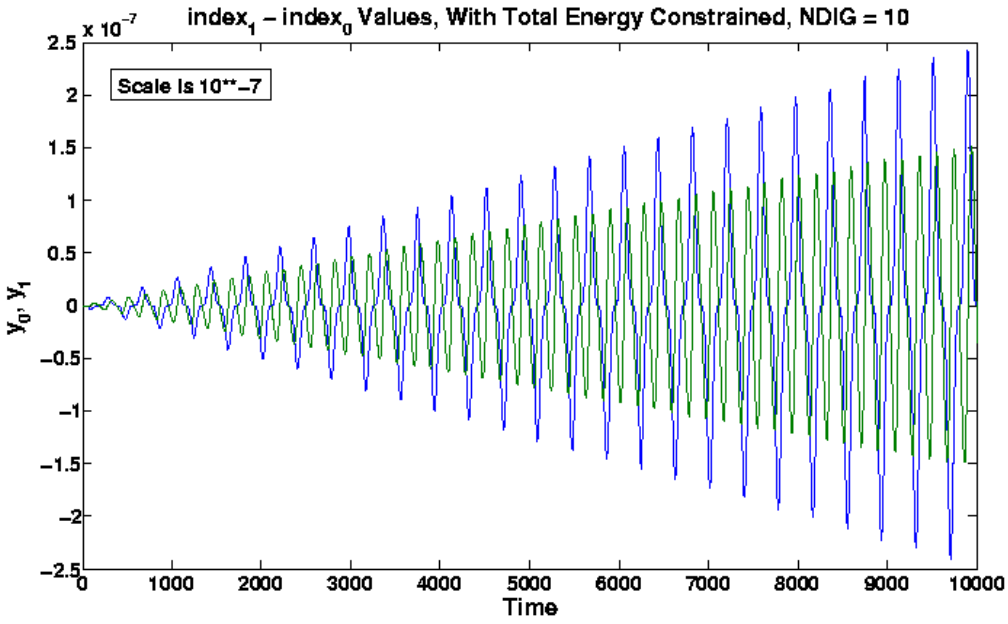


Figure 6. Linear Error Growth with Energy Constrained

Appendix II - Getting Started: Computing Initial Conditions

In Equation 1 the initial conditions may not be completely known. Our basis for presenting an algorithm to compute these values assumes:

1. The initial position values $y(t_0) = y_0$ are known and are to remain fixed.
2. Some or all of the initial derivative values $y'(t_0) = y'_0$ are not known exactly and must be computed numerically. All values of y'_0 can be changed to achieve consistent initial conditions.
3. Estimates for $y'(t_0) = y'_0$ are given such that $F(t_0, y_0, y'_0)$ is defined and has continuous partial derivatives $F_t, F_y, F_{y'}$ in the neighborhood of a solution for y'_0.
4. The *Index* of Equation 1 has the value 0 or 1.

A simple example illustrates part of the problem of finding initial derivative values. Consider two equations of the Index 1 DAE, [1], p. 138:

$$y_1 + y'_1 + y'_2 - g_1(t) = 0$$
$$y_2 - g_2(t) = 0$$

Only the sum of the derivatives is determined by the first equation. None of the derivatives appear in the second equation. So a Newton method, for example, will not work in solving for initial derivative values. It is clear that consistency requires $y'_2 - g'_2(t_0) = 0$.

We present an approach that will likely work in the general case of an Index 0 or 1 DAE. It is a form of Newton's method. The method consists of solving $2N$ equations:

$$F\left(t_0, y_0, y_0'\right) = 0$$
$$\left.\frac{dF}{dt}\right|_{t_0, y_0, y_0', y_0''} = 0$$

To simplify notation abbreviate $A = \frac{\partial F}{\partial y}$, $B = \frac{\partial F}{\partial y'}$. For the inner Newton method we expand around the current iterate y_0'. Then the two equations are approximated by solving the rank-deficient linear systems

$$\min \left\| y'' \right\|^2 \; subject \; to \begin{bmatrix} B & 0_{N\times N} \\ A & B \end{bmatrix} \begin{bmatrix} dy' \\ y'' \end{bmatrix} = \begin{bmatrix} -F\left(t_0, y_0, y_0'\right) \\ -F_t - A y_0' \end{bmatrix} \equiv \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Equation 21

One should start the iteration by attempting to solve the first N equations and complete the update $y_0' \leftarrow y_0' + dy'$. For DAE systems of Index 0 this will determine y_0'. There is a concern of how long to continue iterating this update step. A reduction in the norm of $F\left(t_0, y_0, y_0'\right)$ is required each iteration. This iteration is continued until the norm is as small as the model uncertainty itself. This may be a difficult criterion to apply. If the DAE is actually an ODE, $F\left(t, y, y'\right) = f(t, y) - y'$, then convergence will occur in one update step.

A solution of Equation 21 that yields updated values for y_0' has some novel features for problems of Index 1. To proceed for dense matrices, factor the rank-deficient matrix B: e.g. compute $QBK = \begin{bmatrix} R_{r\times r} & 0 \\ 0 & 0 \end{bmatrix} \equiv \begin{bmatrix} \tilde{R} \\ 0 \end{bmatrix}$, where R is a $k \times k$ upper triangular matrix, and K is an orthogonal matrix. This factorization can be done using Gaussian elimination with full pivoting followed by application of Householder transformations from the right side. Another approach is to apply Householder transformations with column pivoting from the left side and finish the factoring from the right side using Householder transformations.

Change variables $\begin{bmatrix} dy' \\ y'' \end{bmatrix} = \begin{bmatrix} Kx \\ Kz \end{bmatrix}$ and apply the factor Q to both blocks of Equation 21 to get the transformed system

$$\min \left\| z \right\|^2 \; subject \; to \begin{bmatrix} QBK & 0_{N\times N} \\ QAK & QBK \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} Qb_1 \\ Qb_2 \end{bmatrix}$$

Equation 22

Partition the matrix and transformed right-hand sides

$$QAK = \begin{bmatrix} C_{k\times N} \\ D_{(N-k)\times N} \end{bmatrix}, Qb_1 = \begin{bmatrix} d_k \\ e_{N-k} \end{bmatrix}, Qb_2 = \begin{bmatrix} f_k \\ g_{N-k} \end{bmatrix}$$

Then collect those equations involving $x = Ky'$ and rearrange the remaining system:

$$\min \|z\|^2 \; subject \; to \begin{bmatrix} \tilde{R} & 0 \\ D & 0 \\ C & \tilde{R} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} d \\ g \\ f \\ e \end{bmatrix}$$

Equation 23

Note that the condition that the system has Index 1 implies that the system $\begin{bmatrix} \tilde{R} \\ D \end{bmatrix} x = \begin{bmatrix} d \\ g \end{bmatrix}$ is uniquely solvable for every right-hand side. This determines $x = Kdy'$ uniquely. It is also necessary that Equation 23 satisfy the compatibility condition $e = 0$. Otherwise inconsistent initial conditions have been specified. Finally we could solve the remaining equations for the minimum value of $\|z\|^2$:

$$R \begin{bmatrix} z_1 \\ \vdots \\ z_k \end{bmatrix} = f, \quad z_{k+1} = \cdots = z_N = 0$$

[An interesting side note is that for computing a single update to the initial estimates for y_0', it *is not necessary* to compute this update to y_0''. It is also not required that this particular solution for $z = K^T y''$ be calculated.]

A regularization procedure for solving Equation 21 leads to the perturbed linear system

$$\begin{bmatrix} B & \varepsilon I_N \\ A & B \end{bmatrix} \begin{bmatrix} dy' \\ y'' \end{bmatrix} = \begin{bmatrix} -F\left(t_0, y_0, y_0'\right) \\ -F_t - Ay_0' \end{bmatrix} \equiv \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \varepsilon \cong macheps \|B\|$$

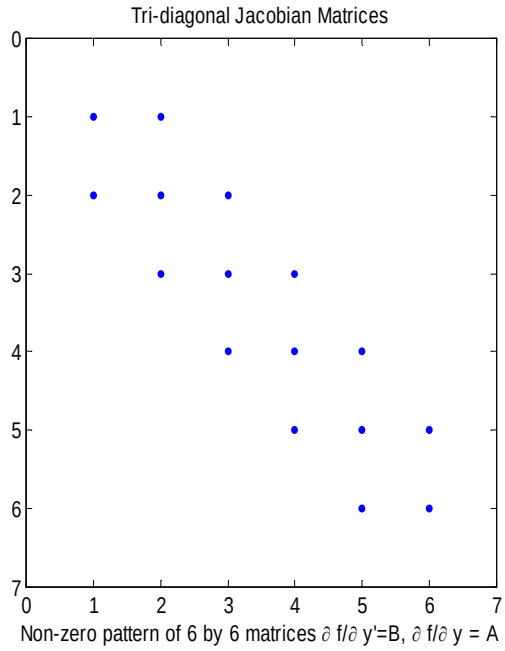
Equation 24

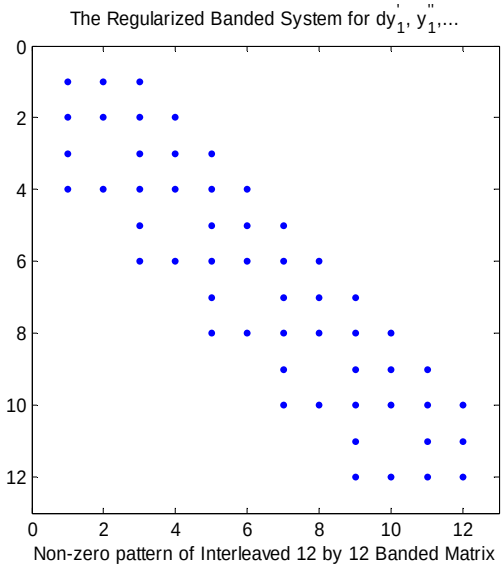
This regularized formulation is perhaps useful when there is a special structure or perhaps sparseness for the matrices A, B, particularly when N is large. The *ad hoc* property of Equation 24 derives from adding additional N regularizing equations $\eta B dy' - y'' \cong 0, \eta$ small, $\doteq$ machine eps to Equation 21. After applying plane rotations to these additional equations, rescaling, and retaining the first $2N$ equations we arrive at Equation 24, with $\varepsilon = -\eta/(1+\eta^2)$. The ignored equations are $-(1+\eta^2)^{-1/2} y'' \cong 0$, terms that directly express smallness in $\|y''\|$. This development may appear convoluted but it follows from the fact that

there are arbitrary choices about what regularizing equations may be adjoined to ill-posed systems. The choice we have made follows from the convenience when computing a solution of Equation 24 with existing linear equation solvers.

An important special case of Equation 24 occurs when A, B are banded matrices. For $\varepsilon \neq 0$ arrange the $2N$ unknowns in the order $x = [dy'_1, y''_1, \ldots, dy'_N, y''_N]^T$. Following this step arrange the $2N$ equations resulting in final row order: $[1, N+1, 2, N+2, \ldots, N, 2N]$. The right-hand side vector must also be correspondingly rearranged. This linear system for x is banded, and the upper and lower bandwidths are effectively doubled. Nevertheless, this is an important fact to use because of the existence of efficient banded matrix solvers in such software collections as LAPACK or LINPACK. Also note that for small values of $|\varepsilon| > 0$ the permuted equation will be ill-conditioned but the uncertainty will be in the second derivative terms which are not used in the Newton step.

Here are details for the bandwidths. Suppose that m_l is the number of sub-diagonals and m_u is the number of super-diagonals of A, B. Then the $2N \times 2N$ interleaved banded matrix has $\hat{m}_l = 2m_l + 1$ sub-diagonals and $\hat{m}_u = \max(2m_u, 1)$ super-diagonals. Thus the bandwidth is slightly more than doubled for the interleaved system. Note that even if A, B were triangular matrices (upper or lower) the interleaved system is not triangular but genuinely banded. Here are illustrations of the non-zero patterns when A, B are tri-diagonal with $N = 6$.





About the Authors

R. J. Hanson

Dr. Richard Hanson, Ph.D. is a Principal Scientist at Visual Numerics. His major project responsibilities include technical supervision, project management, and involvement in the full cycle of software development. Early IMSL Library projects included notable improvements in the IMSL real, symmetric, dense eigenvalue-eigenvector code, from which (given the right environment) an order of magnitude speed-up is obtained compared with LAPACK.

For the past twenty years, Dr. Hanson has also served on the IFIP WG 2.5 Committee on Mathematical Software. He was appointed for significant contributions to the publication of mathematics software and continues to participate in projects, which monitor and improve mathematical software tools, languages and working environments.

Some of his most recent significant work has been to design and implement MPI-enhanced subroutines and functions for scientific applications on distributed network computers.

F. T. Krogh

Dr. Fred Krogh worked at the Jet Propulsion Laboratory for 30 years where he retired as a Principal Mathematician. He consults there and with Visual Numerics. Dr. Krogh has served as Algorithms Editor to the *ACM Transactions on Mathematical Software*. He is best known for work on Adams methods for ordinary differential equations and has also done work on stiff equations, quadrature, nonlinear least squares, FFT's, and interpolation. He is currently working on a new method for linear programming, quadratic programming, and constrained least squares. He is the founder of Math à la Carte, which provides numerical software at http://mathalacarte.com.