STATE-SPACE REPRESENTATION OF VORTEX WAKES USING THE METHOD OF LINES

Roberto Celi
Department of Aerospace Engineering,
University of Maryland,
College Park, Maryland 20742, USA.

Abstract

The method of lines (MOL) is applied to the equations of helicopter rotor vortex wakes, and converts the governing partial differential equations into a system of ordinary differential equations (ODE). These ODE can then be coupled to other ODE modeling helicopter dynamics, for time-marching simulations or to extract linearized models. The MOL is applied to a simplified set of wake equations that has an analytical solution. Because these simplified equations neglect key wake physics, the study is only a first step toward applying MOL to realistic models. Therefore, the conclusions only apply to the simplified problem considered. The results show that the MOL is a suitable method to formulate vortex wake models in state-space form. The solutions are accurate and numerically stable. Refining the space discretization increases the stiffness of the ODE, but explicit solvers can still be used. Computational efficiency increases when the accuracies of space and time discretizations are matched. Formulas of several orders are used in the space discretization. In all cases, the explicit solver DE/STEP is much more computationally efficient than the implicit solver DASSL. Linearized state-space wake models can be easily obtained. The MOL could also provide a systematic methodology to extract state-space models from CFD formulations, and therefore to increase the accuracy of helicopter simulation models.

Notation

$[A_\zeta]$	Linearized matrix of ODE system; also, finite difference matrix
e	Local error tolerance for the ODE solvers
$\mathbf{i}, \mathbf{j}, \mathbf{k}$	Unit vectors of rotor coordinate system
N_ζ	Number of intervals in the ζ-discretization
$\mathbf{r}$	Distance of a vortex point from the hub
R	Rotor radius
r_x, r_y, r_z	Components of $\mathbf{r}$ along $\mathbf{i}, \mathbf{j}, \mathbf{k}$
α_s, β_0	Rotor shaft angle and flapping angle
ζ	Angular distance of a vortex point from the blade
$\Delta\zeta$	Step size of the finite difference discretization along ζ
μ	Advance ratio
ψ	Blade azimuth angle

Abbreviations

BDF	Backward Differentiation Formula(s)
CFD	Computational Fluid Dynamics
DAE	Differential-Algebraic Equation(s)
MOL	Method of Lines
ODE	Ordinary Differential Equation(s)
PDE	Partial Differential Equation(s)
2PCD2	Two-point central difference scheme (2nd order)
2PU1	Two-point upwind difference scheme (1st order)
3PU2	Three-point upwind difference scheme (2nd order)
4PCD4	Four-point central difference scheme (4th order)
5PBU4	Five-point biased upwind difference scheme (4th order)

Introduction

The solution of several practical problems in rotorcraft aeromechanics is much simpler if the mathematical problem is formulated in state-space form, i.e., in the form of systems of ODE. For example, the simplest and most direct way to compute the aeroelastic stability of a helicopter rotor is to perform an eigenvalue analysis of a linearized version of the equations of motion. Also, many quantities of interest in flight dynamics, such as dynamic stability and frequency response to pilot input, can be calculated directly from a linearized mathematical model in state-space form. State-space models are also very important for the design of rotor and flight control systems, because the vast majority of control design techniques relies on the availability of such models. Finally, although time-marching solutions of the equations of motion can be obtained if the equations are a combination of ODE and PDE, they are usually easier to achieve if the equations are all in ODE form.

The structural and inertia portions of the typical equations of motion of a helicopter can usually be written in state-space form with no major difficulty. On the other hand, several components of the aerodynamic portion

Professor, Alfred Gessow Rotorcraft Center; e-mail: celi@eng.umd.edu.

Paper presented at the 30th European Rotorcraft Forum, Marseilles, France, September 14-16, 2004.

are not currently available in ODE form. State-space models are available for the calculation of the aerodynamic properties of airfoils (see, for example, Refs. [1] and [2]). State-space models of rotor inflow are also currently available. The largest body of work in this area has been developed over the last three decades by Peters and his coworkers (Ref. [3] contains an extensive review of such work). These widely used models trace their origin back to the theory developed by Joglekar and Loewy [4], which is based on an unsteady actuator disk theory. Other state-space inflow models are those of Keller and Curtiss [5], Basset [6], and Rosen [7, 8]. Simple but accurate state-space inflow models can also be extracted using frequency domain system identification, either from experimental [9] or simulation [10, 11] data.

All the state-space models just mentioned are of a "global" nature, in the sense that they take the form of actuator disk theories of various degrees of sophistication. Therefore, they have intrinsic limitations in the level of detail of the flowfield that they can resolve. There are several wake theories that model vortex dynamics and do not suffer from this limitation, but most of them is formulated in state-space form (see Ref. [12] for a comprehensive review of these free-vortex filament methods). The only exception is the state-space free wake model developed by Johnson [13]

The time-dependent geometry of a wake vortex is governed by the following PDE [14]

$$\frac{\partial \mathbf{r}(\psi\zeta\ \ )}{\partial \psi} + \frac{\partial \mathbf{r}(\psi\zeta\ \ )}{\partial \zeta} = \frac{1}{\Omega}\mathbf{V}[\mathbf{r}(\psi\zeta\ \ )] \tag{1}$$

where $\mathbf{r}$ is the position vector of a point on the vortex filament (collocation point), ψ is the blade azimuth, ζ is an angular distance measured along the vortex starting from the vortex release point on the blade trailing edge, Ω is the rotor speed, and $\mathbf{V}$ is the local convection velocity of the vortex [15]. Equation (1) has an exact analytical solution [15] for the special case of constant velocity $\mathbf{V}$ throughout the wake, which is equivalent to assuming a constant value of the inflow λ_i, uniform across the disk. For this case it is [15] $\mathbf{V} = \Omega R\mu\mathbf{i} + \Omega R\lambda_i\mathbf{k}$, and therefore

$$\frac{\partial \mathbf{r}(\psi\zeta\ \ )}{\partial \psi} + \frac{\partial \mathbf{r}(\psi\zeta\ \ )}{\partial \zeta} = R\mu\mathbf{i} + R\lambda_i\mathbf{k} \tag{2}$$

which has the exact solution [15]

$$\begin{aligned} \mathbf{r}(\psi\zeta\ \ ) \ &= \ [R\mu\zeta + r_v(\cos\beta_0\cos(\psi-\zeta)\cos\alpha_s + \\ &\qquad\qquad + \sin\beta_0\sin\alpha_s)]\,\mathbf{i} \\ &\quad + r_v\cos\beta_0\sin(\psi-\zeta)\mathbf{j} \\ &\quad + [R\lambda_i\zeta + r_v(\sin\beta_0\cos\alpha_s \\ &\qquad\qquad - \cos\beta_0\cos(\psi-\zeta)\sin\alpha_s]\,\mathbf{k} \end{aligned} \tag{3}$$

The main objective of this paper is to propose a methodology for the reformulation of vortex wake models in state space form, suitable for direct coupling with time-marching simulations, and for linearized stability and response analyses. This methodology is based on the use of the method of lines, which is a technique for the numerical solution of PDE, in which only the dependency on space is discretized in finite difference form [16], with the result that the PDE are converted into a system of coupled ODE. In this paper, the MOL will be applied to the solution of the simplified vortex wake equation, Eq. (2).

The specific objectives of the paper are:

1. To summarize the main features of the method of lines;

2. To discuss the main issues concerning the space discretization of Eq. (2), and the subsequent time-marching solution;

3. To extract a linearized state-space model of the vortex wake in first-order form, suitable for eigenanalysis; and

4. To present results showing the main numerical characteristics of the MOL-based technique, as a function of a variety of discretization and solution parameters.

It is important to point out that the nonlinear term on the right-hand-side of the complete vortex equation, Eq. (1) contains many key aspects of the problem, both from the physical and the computational point of view. It is in the correct formulation and treatment of this term that many of the challenges of accurate free vortex wake analyses lie. Therefore, the results presented in this paper, which are limited to the simplified version of Eq. (2), should be considered as just a first step toward the objective of deriving accurate free wake models in state-space form.

Finally, because the MOL can be applied to any PDE, it can provide a systematic solution to the problem of coupling CFD models to helicopter simulation models composed of systems of ODE.

Method of Lines

The main features of the MOL will be described in this section, and applied to the simplified vorticity transport equation, Eq. (2). The equation, rewritten in scalar form component by component is:

$$\frac{\partial r_x}{\partial \psi} + \frac{\partial r_x}{\partial \zeta} \ = \ R\mu\zeta \tag{4}$$

$$\frac{\partial r_y}{\partial \psi} + \frac{\partial r_y}{\partial \zeta} \ = \ 0 \tag{5}$$

$$\frac{\partial r_z}{\partial \psi} + \frac{\partial r_z}{\partial \zeta} \ = \ R\lambda_i\zeta \tag{6}$$

with $\mathbf{r} = r_x\mathbf{i} + r_y\mathbf{j} + r_z\mathbf{k}$. The azimuth angle ψ and the the angular location ζ take, respectively, the role of time and space. The three component equations are first-order hyperbolic PDE and, in particular, Eq. (5) is the basic advection equation.

Introduction

The MOL consists of discretizing a PDE with respect to only the space or the time variable, typically the former. Considering Eq. (5) as an example, and assuming a central difference approximation to the ζ-derivative, this partial discretization results in:

$$\frac{\partial r_y(\psi\zeta\ _i)}{\partial \psi} = -\frac{1}{2\Delta\zeta}\left[r_y(\psi\zeta\ _i + \Delta\zeta) - r_y(\psi\zeta\ _i - \Delta\zeta)\right] \tag{7}$$

which is an ordinary differential equation in $r_y(\psi\zeta\ _i)$. If the spatial domain ζ is divided into $N-1$ intervals, then the MOL leads to N ODE like Eq. (7) with $i = 1, 2, \ldots, N$. The solution of the system of ODE yields the solution of the original PDE. The quantities $r_y(\psi\zeta\ _i), i = 1, 2, \ldots, N$ become the states of the model. Each $r_y(\psi\zeta\ _i)$ represents the position along the y-axis of a collocation point on the vortex filament. Equations (4) and (6) are treated in the same way.

The vorticity transport equation is first order in ψ and ζ, and therefore it requires an initial condition and a boundary condition. From the exact solution, Eq. (3), the initial conditions for the three components of the equation are

$$\begin{aligned} r_x(0,\zeta) &= R\mu\zeta + \\ &+ r_v(\cos\alpha_s\cos\beta_0\cos\zeta + \sin\alpha_s\sin\beta_0) \quad (8)\\ r_y(0,\zeta) &= -r_v\cos\beta_0\sin\zeta \quad (9)\\ r_z(0,\zeta) &= r_v(\cos\alpha_s\sin\beta_0 - \sin\alpha_s\cos\beta_0\cos\zeta) (10)\end{aligned}$$

When specialized for $\zeta = \zeta_i$, the equations above provide the initial conditions for the corresponding ODE, such as Eq. (7) for $r_y(\psi\zeta\ _i)$. Equation (3) also provides the boundary conditions:

$$\begin{aligned} r_x(\psi\ 0) &= r_v(\cos\alpha_s\cos\beta_0\cos\psi + \sin\alpha_s\sin\beta_0) \ (11)\\ r_x(\psi\ 0) &= r_v\cos\beta_0\sin\psi \quad (12)\\ r_x(\psi\ 0) &= r_v(\cos\alpha_s\sin\beta_0 - \sin\alpha_s\cos\beta_0\cos\psi) \ (13)\end{aligned}$$

which are used in Eq. (7) for $i = 1$ and $\zeta_1 = 0$ (similarly, for r_x and r_z).

Equations (7), written for all points ζ_i, can be grouped together in matrix form as:

$$\dot{\mathbf{r}}_y(\psi) = [A_\zeta]\,\mathbf{r}_y(\psi) \tag{14}$$

after defining $\dot{(\ )} = \partial/\partial\psi$ and

$$\mathbf{r}_y(\psi) = \left[r_y(\psi\zeta\ _1) \quad r_y(\psi\zeta\ _2) \quad \ldots \quad r_y(\psi\zeta\ _N)\right]^T \tag{15}$$

and with the matrix $[A_\zeta]$ given by

$$[A_\zeta] = -\frac{1}{2\Delta\zeta}\begin{bmatrix} -3 & 4 & -1 & & & \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & & \ldots & & \\ & & & -1 & 0 & 1 \\ & & & 1 & -4 & 3 \end{bmatrix} \tag{16}$$

(all the terms outside the diagonal band are equal to zero). The first and last columns of $[A_\zeta]$ come from special three point approximations to the derivative at the two ends of the ζ domain, that is [16]

$$\frac{\partial r_x(\psi\zeta\ \ )}{\partial\zeta} \approx \frac{1}{2\Delta\zeta}\left[-3r_x(\psi\zeta\ _1) + 4r_x(\psi\zeta\ _2) - r_x(\psi\zeta\ _3)\right] \tag{17}$$

$$\frac{\partial r_x(\psi\zeta\ _N)}{\partial\zeta} \approx \frac{1}{2\Delta\zeta}\left[r_x(\psi\zeta\ _{N-2}) - 4r_x(\psi\zeta\ _{N-1}) + 3r_x(\psi\zeta\ _N)\right] \tag{18}$$

All approximations to the first ζ-derivative in Eq. (16) have errors of order $O(\Delta\zeta^2)$. The ODE corresponding to the first row of $[A_\zeta]$, is treated slightly differently from the other equations. In fact, although the derivative $\partial\mathbf{r}_y(\psi\zeta\ _1)/\partial\psi$ is computed anyway, the quantity $\mathbf{r}_y(\psi\zeta\ _1)$ computed by the ODE solver is overridden at each time step by the corresponding value given by the boundary condition, Eq. (12) (similarly, for r_x and r_z).

The complete state-space model for the vorticity transport equation in the form of Eq. (7) is

$$\begin{Bmatrix} \dot{\mathbf{r}}_x(\psi) \\ \dot{\mathbf{r}}_y(\psi) \\ \dot{\mathbf{r}}_z(\psi) \end{Bmatrix} = \begin{bmatrix} [A_\zeta] & 0 & 0 \\ 0 & [A_\zeta] & 0 \\ 0 & 0 & [A_\zeta] \end{bmatrix} \begin{Bmatrix} \mathbf{r}_x(\psi) \\ \mathbf{r}_y(\psi) \\ \mathbf{r}_z(\psi) \end{Bmatrix} + \\ + \begin{Bmatrix} R\mu\boldsymbol{\zeta} \\ 0 \\ R\lambda_i\boldsymbol{\zeta} \end{Bmatrix} \tag{19}$$

where $\mathbf{r}_x$ and $\mathbf{r}_z$ are defined similarly to $\mathbf{r}_y$ in Eq. (15), and $\boldsymbol{\zeta}$ is defined as

$$\boldsymbol{\zeta} = \left[\zeta_0 \quad \zeta_1 \quad \zeta_2 \quad \ldots\zeta_N\right]^T \tag{20}$$

Equation (19) is now ready to be coupled, for example, to a system of rotor-fuselage ODE for a time-marching solution.

Space discretization

An appropriate spatial discretization is clearly a key ingredient for accurate PDE solutions using the MOL. This section summarizes five space discretizations appropriate for the solution of the vorticity transport equation. These discretizations are taken from Ref. [16], which also contains additional details of their derivation. All the formulas are written for a generic function $f(\zeta)$.

1—Two-point central differences (2PCD2)

This is the discretization used in all the equations of the previous sections, and no additional details will be provided here.

2—Four-point central differences (4PCD4)

The general formula is

$$\left.\frac{\partial f}{\partial \zeta}\right|_{\zeta=\zeta_i} = \frac{1}{24\Delta\zeta}\left[2f(\zeta_{i-2}) - 16f(\zeta_{i-1}) + 16f(\zeta_{i+1}) - 2f(\zeta_{i+2})\right] + O(\Delta\zeta^4) \qquad (21)$$

The formulas at the beginning and the end of the ζ-domain are:

$$\left.\frac{\partial f}{\partial \zeta}\right|_{\zeta=\zeta_0} = \frac{1}{24\Delta\zeta}\left[-50f(\zeta_1) + 96f(\zeta_2) - 72f(\zeta_3) + 32f(\zeta_4) - 6f(\zeta_5)\right] \qquad (22)$$

$$\left.\frac{\partial f}{\partial \zeta}\right|_{\zeta=\zeta_1} = \frac{1}{24\Delta\zeta}\left[-6f(\zeta_1) - 20f(\zeta_2) + 36f(\zeta_3) - 12f(\zeta_4) + 2f(\zeta_5)\right] \qquad (23)$$

$$\left.\frac{\partial f}{\partial \zeta}\right|_{\zeta=\zeta_{N-1}} = \frac{1}{24\Delta\zeta}\left[2f(\zeta_{N-4}) + 12f(\zeta_{N-3}) - 36f(\zeta_{N-2}) + 20f(\zeta_{N-1}) + 6f(\zeta_N)\right] \qquad (24)$$

$$\left.\frac{\partial f}{\partial \zeta}\right|_{\zeta=\zeta_N} = \frac{1}{24\Delta\zeta}\left[6f(\zeta_{N-4}) - 32f(\zeta_{N-3}) + 72f(\zeta_{N-2}) - 96f(\zeta_{N-1}) + 50f(\zeta_N)\right] \qquad (25)$$

All the formulas above have an error of order $O(\Delta\zeta^4)$. Using the expressions above, the $[A_\zeta]$ matrix in Eq. (16) becomes

$$[A_\zeta] = -\frac{1}{24\Delta\zeta}\begin{bmatrix} -50 & 96 & -72 & 32 & -6 & & & & \\ -6 & -20 & 36 & -12 & 2 & & & & \\ 2 & -16 & 0 & 16 & -2 & & & & \\ & 2 & -16 & 0 & 16 & -2 & & & \\ & & \cdots & \cdots & \cdots & \cdots & \cdots & & \\ & & & 2 & -16 & 0 & 16 & -2 & \\ & & & & 2 & -16 & 0 & 16 & -2 \\ & & & & -2 & 12 & -36 & 20 & 6 \\ & & & & 6 & -32 & 72 & -96 & 50 \end{bmatrix} \qquad (26)$$

3—First-order two-point upwind approximation (2PU1)
When applied to the solution of the advection equation with a discontinuity in the flow, the two centered approximations just described tend to produce a strong oscillatory behavior in the solution, especially downstream of the discontinuity [16]. This can be partially remedied using upwind approximations, the simplest of which is a first-order, two-point approximation. Equation (2) does not contain the physics necessary to develop discontinuities, and therefore the use of upwind approximations is not expected to produce any improvement in the accuracy of its solution. However, the behavior of these approximations in the simplified model will be studied anyway, as a first step toward the application to more realistic wake models.

Using the same notation as in the previous section above, the $[A_\zeta]$ matrix in Eq. (16) with this type of approximation becomes

$$[A_\zeta] = -\frac{1}{\Delta\zeta}\begin{bmatrix} -1 & 1 & & & & \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & \cdots & \cdots & & \\ & & & -1 & 1 & \\ & & & & -1 & 1 \end{bmatrix} \qquad (27)$$

This approximation has an error of order $O(\Delta\zeta)$.
4—Second-order three-point upwind approximation (3PU2)
A more accurate, second-order upwind approximation can be built using three points. The $[A_\zeta]$ matrix in Eq. (16) with this type of approximation is

$$[A_\zeta] = -\frac{1}{2\Delta\zeta}\begin{bmatrix} -3 & 4 & -1 & & & \\ -1 & 0 & 1 & & & \\ 1 & -4 & 3 & & & \\ & \cdots & \cdots & \cdots & & \\ & & 1 & -4 & 3 & \\ & & & 1 & -4 & 3 \end{bmatrix} \qquad (28)$$

This approximation has an error of order $O(\Delta\zeta^2)$.
5—Fourth-order five-point biased upwind approximation (5PBU4)
The two-point upwind formula eliminates the downstream oscillations of the centered formulas, but introduces numerical diffusion; the three-point upwind formula reduces the diffusion, but reintroduces some oscillation upstream of the discontinuity [16]. A fourth-order five-point biased upwind approximation, as a combination of centered and upwind formulas, helps reduce both the numerical diffusion and the oscillatory behavior [?16]. The $[A_\zeta]$ matrix in Eq. (16) with this type of approximation is

$$[A_\zeta] = -\frac{1}{12\Delta\zeta}\times \times\begin{bmatrix} -25 & 48 & -36 & 16 & 3 & & \\ -3 & 10 & 18 & -6 & 1 & & \\ 1 & -8 & 0 & 8 & -1 & & \\ -1 & 6 & -18 & 10 & 3 & & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & -1 & 6 & -18 & 10 & 3 \\ & & 3 & -16 & 36 & -48 & 25 \end{bmatrix} \qquad (29)$$

This approximation has an error of order $O(\Delta\zeta^4)$.
Other space discretization methods
When the finite difference formulas above are used to solve a problem, the space (or ζ-) discretization is held

fixed throughout the solution. This is not the most desirable strategy when rapid spatial variations occur only in some portions of the problem, because the (fixed) discretization must be appropriately refined to capture these variations, but might be needlessly fine everywhere else. A growing body of research has addressed the issue of *adaptive* space discretizations, and is reviewed in Ref. [18]. Because of the simplicity of the PDE studied in this paper, however, no such discretization scheme will be considered here.

Stiffness

Refining the space discretization, i.e., reducing $\Delta\zeta$, increases the accuracy of the solution, but also increases the stiffness of the resulting system of ODE. The stability of the ODE solution algorithm must also be taken into account. The discussion that follows is based on Ref. [16].

Consider for simplicity, because it is homogeneous, Eq. (5), and assume that a two-point central difference approximation is used. This results in (dropping the dependency on ψ in the notation):

$$\frac{\partial r_y(\zeta_i)}{\partial\psi} = -\frac{1}{2\Delta\zeta}\left[r_y(\zeta_{i+1}) - r_y(\zeta_{i-1})\right] \tag{30}$$

Assuming a solution of the general type

$$r_y(\psi\zeta\quad) = C\eta(\psi)\phi(\zeta) \qquad \text{with } \phi(\zeta) = e^{jkx}, j = \sqrt{-1} \tag{31}$$

and substituting into Eq. (30) gives:

$$\begin{aligned}\frac{\partial\eta}{\partial\psi} &= -\frac{\psi}{2\Delta\zeta}\left(e^{jk\Delta x} - e^{-jk\Delta x}\right)\\ &= -j\frac{\psi}{\Delta\zeta}\sin(k\Delta\zeta)\end{aligned} \tag{32}$$

Rewriting the equation in terms of an eigenvalue λ gives:

$$\frac{\partial\eta}{\partial\psi} = \lambda_k\psi \qquad \text{with } \lambda_k = -\frac{j}{\Delta\zeta}\sin(k\Delta\zeta) \tag{33}$$

Therefore, the ODE resulting from a space discretization with two-point central differences have purely imaginary eigenvalues, which are inversely proportional to the space step size $\Delta\zeta$. This has two potentially important consequences. The first is that the spectrum of eigenvalues λ_k will become broader as $\Delta\zeta$ becomes smaller or, equivalently, the ratio between the largest and the smallest eigenvalue of the solution will increase as $\Delta\zeta$ decreases. In other words, the equations become stiffer as the space discretization is refined. At some point, a stiff ODE solver will be needed. The second consequence is that the ODE solution scheme must be chosen carefully, because for many schemes the stability properties for solutions with purely imaginary, high frequency eigenvalues, are poor.

The preceding comments were based on the discretization with a two-point central difference formula. However, they are generally valid for the other four discretization previously presented, although the details of the derivations are obviously different. Eigenvalues for all five schemes will presented later, in the “Results” section.

Differential equation solvers

Two publicly available ODE solvers, namely DE/STEP and DASSL, will be used in this study to integrate the equations resulting from the MOL. Both are very well-known, reliable, robust solvers.

DE/STEP [19] is a variable-step, variable-order ODE solver based on Adams-Bashforth formulas. This is a predictor-corrector, explicit solver that is not designed to integrate stiff systems of equations. Stiffness is detected indirectly, by monitoring the number of function evaluations required to advance the solution. When the ODE are stiff, the solver issues an error message and stops execution. DE/STEP adjusts step size and order of the integration formula to achieve user-specified absolute and relative error tolerances. Formulas of up to order 12 can be used. The Adams-Bashforth algorithm is based on building a polynomial of sufficient order to interpolate the actual solution function: this polynomial is easily accessible during the integration, and can be used to reconstruct additional values of the solution (i.e., of r_x, r_y, and r_z) beyond those calculated by DE/STEP if desired.

DASSL [20] is a variable-step, variable-order Differential-Algebraic Equations (DAE) solver based on Backward Differentiation Formulas (BDF). Because a system of ODE is simply a special case of system of DAE with no algebraic equations, DASSL can also be used as an ODE solver. Several advantages of using DASSL as the primary solver for rotorcraft aeromechanic problems have been discussed in Ref. [21]. The most important is that it can solve ODE in the general implicit form $\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, t) = \mathbf{0}$ rather than requiring the usual explicit form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$, and this simplifies considerably the formulation of the equations of motion. DASSL is a predictor-corrector, implicit solver that can integrate stiff systems of equations. (Note that the word “implicit” is used here in two different contexts: an implicit solver like DASSL can solve equations in the explicit form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$, and equations in implicit form can be solved by explicit solvers with special iterative procedures [21].) Like DE/STEP, DASSL adjusts step size and order of the integration formula to achieve the desired local error tolerances. BDF formulas of order up to 5 can be used. The formulas are unconditionally stable only for order 1 and 2. The order selection algorithm monitors some stability metrics and lowers order if necessary, even if this requires smaller time steps and more function evaluations [22]. Similarly to DE/STEP, the BDF formulas are available during the integration, and can be used to reconstruct additional values of r_x, r_y, and r_z beyond those calculated by DASSL if desired.

Equation (19) is already in the form required by

DE/STEP. A minor modification is necessary for use with DASSL, which requires the residual when the tentative solution vectors for states and derivatives (both provided by DASSL at each time step) are substituted into the system of ODE. The modified version of Eq. (19) is

$$\begin{Bmatrix} \varepsilon_x(\psi) \\ \varepsilon_y(\psi) \\ \varepsilon_z(\psi) \end{Bmatrix} = \begin{Bmatrix} \dot{\mathbf{r}}_x(\psi) \\ \dot{\mathbf{r}}_y(\psi) \\ \dot{\mathbf{r}}_z(\psi) \end{Bmatrix} - \begin{bmatrix} [A_\zeta] & 0 & 0 \\ 0 & [A_\zeta] & 0 \\ 0 & 0 & [A_\zeta] \end{bmatrix} \begin{Bmatrix} \mathbf{r}_x(\psi) \\ \mathbf{r}_y(\psi) \\ \mathbf{r}_z(\psi) \end{Bmatrix} - \begin{Bmatrix} R\mu\boldsymbol{\zeta} \\ 0 \\ R\lambda_i\boldsymbol{\zeta} \end{Bmatrix} \qquad (34)$$

where the vector on the left-hand-side is the vector of residuals.

Both DE/STEP and DASSL are written so that they can easily provide the solution either at the end of the interval of integration or at user defined intermediate points such as at equispaced time points. If these intermediate points do not correspond to the points selected by the solver for its integration, the solution is reconstructed from the interpolation polynomials (with the same accuracy as the the actual computed solution).

Linearized model

From the system of ODE obtained with the MOL it is obviously possible to extract a linearized state-space model of the wake, either in isolation or as part of a larger helicopter simulation model. The linearized model can then be used for the solution of a variety of problems in rotorcraft aeromechanics and flight dynamics. They include the calculation of aeroelastic stability eigenvalues, the calculation of rigid body dynamic poles, the calculation of the frequency response to pilot inputs, including bandwidth and phase delay, and the design of active rotor control and flight control systems.

The simplified vortex wake model used in this study, Eq. (2), is already linear. Therefore, the matrix $[A_\zeta]$ obtained as described in the previous section is already the linearized state-space wake model.

In a more sophisticated and realistic wake model, the terms on the right-hand-side of Eq. (1) will make the ODE of Eqs. (19) or (34) nonlinear and more complicated. Therefore, the linearized model will have to be obtained by linearizing numerically the ODE about a reference equilibrium position, such as a trimmed flight condition. Clearly, regardless of the complexity of the wake model, the linearized model will depend on the specific space (or ζ-) discretization, and on the specific time (or ψ-) perturbation scheme.

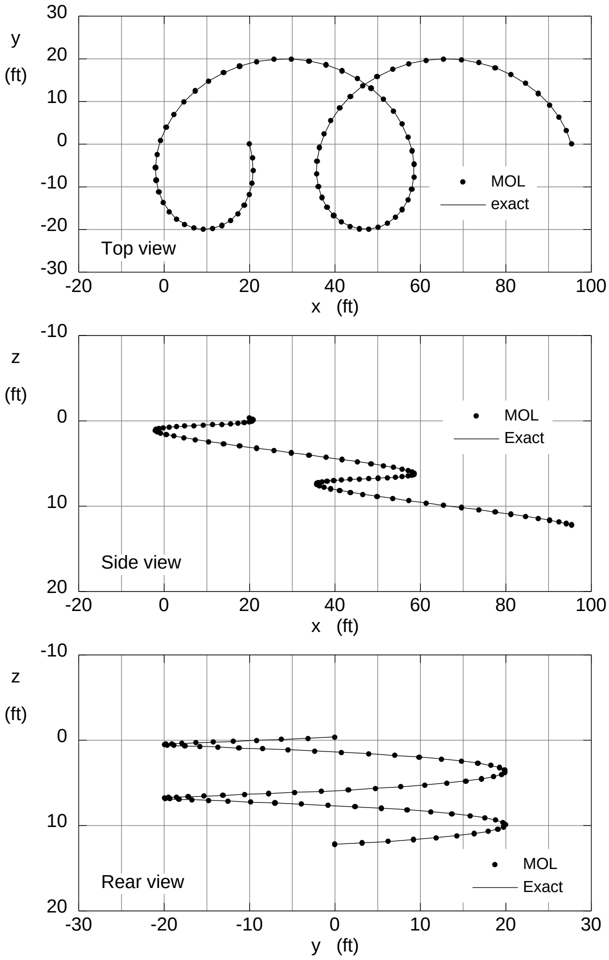

Figure 1: Example of rigid wake geometry; numerical and exact solutions.

Results

All the numerical results presented in this section have been obtained for the following configuration parameters: $\alpha_s = 2$ deg, $\beta_0 = 3$ deg, $\mu = 0.3$, $\lambda = 0.05$, and $r_v = R = 20$ ft. These are reasonable values for helicopter rotor blades, but are otherwise completely arbitrary. Only one rotor blade will be considered. Because Eq. (2) does not contain the physics of interblade aerodynamic interaction, no additional information would be gained by considering more than one blade. Similarly, the length of a vortex filament is set to 720 deg only because this is a reasonable representative value [14], but otherwise the filament length does not have any effect on the solution because Eq. (2) does not model the mutual interaction between wake geometry and rotor inflow. All time (i.e., ψ-) integrations will be carried out from $\psi = 0$ deg to $\psi = 720$ deg. For all the results, DE/STEP and DASSL have been set up so that they return the solution values ten times per rotor revolution, or every $\Delta\psi = 36^o$, for a total of 20 outputs. These numbers are only chosen for convenience, and do not affect in any way the step sizes selected by the solvers to perform the integrations.

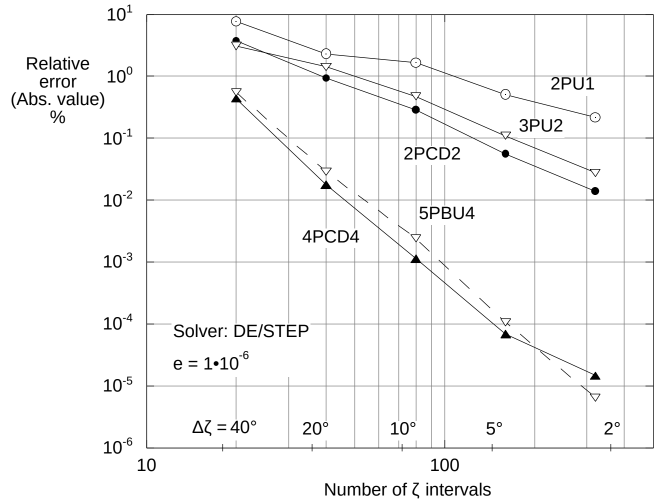

Figure 2: Absolute value of relative error as a function of ζ-discretization; DE/STEP with $e = 10^{-6}$.

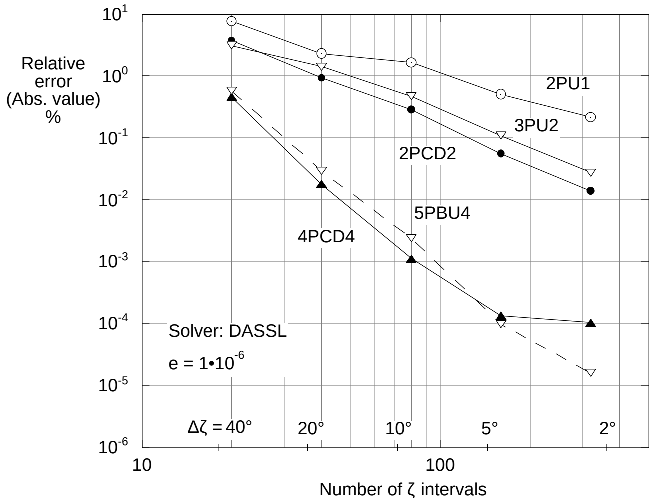

Figure 3: Absolute value of relative error as a function of ζ-discretization; DASSL with $e = 10^{-6}$.

The key parameters that will be explored in the results are the number of intervals N_ζ in which the ζ-domain has been discretized, and the local error tolerances e required from the ODE solvers. The local error tolerances, absolute e_A and relative e_R, will always be set to the same value e. Both ODE solvers and all five ζ-discretization schemes will be used.

An example of graphical representation of the solution is shown in Fig. 1. The wake geometry in the figure was calculated using DE/STEP with $e = 10^{-5}$ and $N_\zeta = 80$. There is clearly an excellent agreement between computed and exact solution.

The relative error of the numerical solution as a function of the number of ζ-intervals N_ζ is presented in Fig. 2. The "relative error" is a RMS measure of the error, and is defined as the square root of the sum of the squares of the relative errors for each component at each ψ output point and at each ζ point, divided by the total number of points. For example, for the case $N_\zeta = 320$, the RMS value consists of the square root of the sum of the squares of $3(\text{components of } \mathbf{r}) \times 20(\psi \text{ output points}) \times 320(N_\zeta) = 19200$ values, further divided by 19200. The relative error is expressed in %. The solver is DE/STEP, with a local error tolerance $e = 10^{-6}$. With this value of e, the error is primarily due to the ζ-discretization, rather than the ψ-discretization.

Figure 2 shows that, as expected, the more accurate formulas are the higher order ones. Central difference and upwind formulas of the same order have similar accuracy, with the central difference formulas being slightly more accurate. The slopes of the curves, when plotted on logarithmic axes as in the figure are consistent with their order of accuracy. The slight slope change for the 4PCD4 formula at $N_\zeta = 320$ may be attributed to the fact that the error has reached the limit of accuracy of the ODE solver for that value of e. The fourth-order formulas produce a relative error of less than 1% even with the coarsest discretization, $N_\zeta = 20$, corresponding to $\Delta\zeta = 36^o$. For $N_\zeta = 80$, or $\Delta\zeta = 9^o$, the relative error is less than 0.01%.

Figure 3 shows the same type of results as Fig. 2, but computed using DASSL. The behavior of the first and second order formulas is virtually identical in both cases, and so is that of the fourth order formulas for the first three values of N_ζ considered. The errors for the two finest discretizations is slightly higher, probably because of the tendency of DASSL to produce somewhat higher errors than DE/STEP for the same value of e [21].

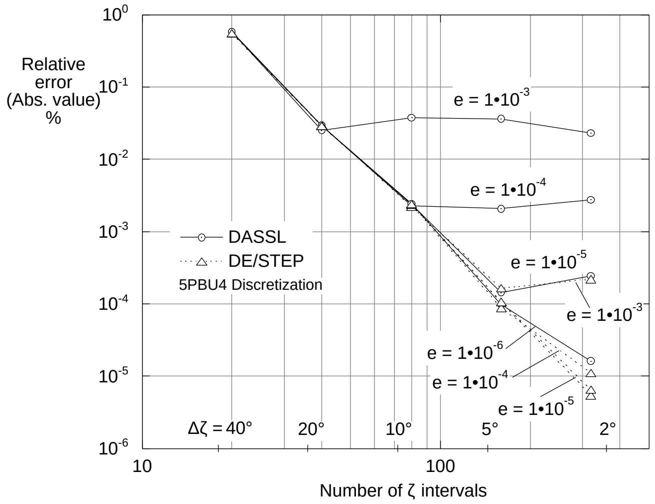

Figure 4: Absolute value of relative error as a function of ζ-discretization, for both ODE solvers and several values of the local error tolerance e; 5PBU4 discretization.

To improve computational efficiency, it is useful to use harmonized ζ- and ψ-discretizations. Figure 4 helps clarify this concept. Again, the figure shows the relative error as a function of N_ζ for both solvers and several values of local error tolerance e. The results

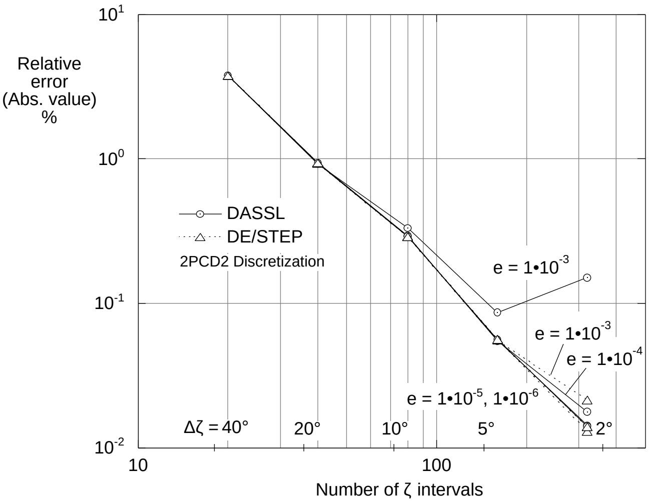

Figure 5: Absolute value of relative error as a function of ζ-discretization, for both ODE solvers and several values of the local error tolerance e; 2PCD2 discretization.

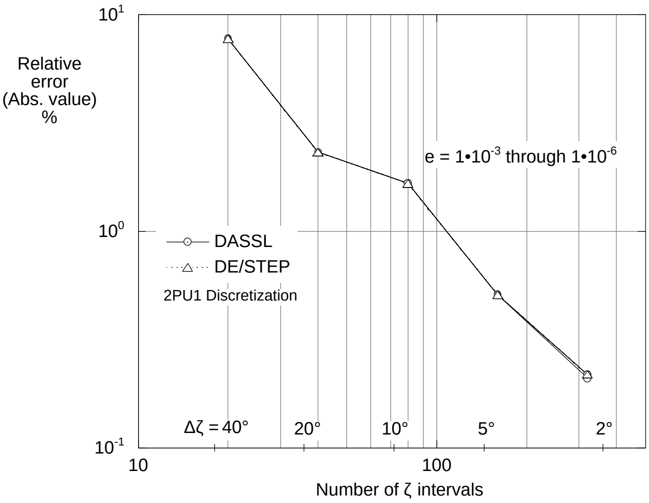

Figure 6: Absolute value of relative error as a function of ζ-discretization, for both ODE solvers and several values of the local error tolerance e; 2PU1 discretization.

for $e = 10^{-6}$ have already been shown in Figs. 2 for DE/STEP and 3 for DASSL: they are representative of the best ψ-accuracy, and the relative error is entirely due to the ζ-discretization. These results essentially lie on a straight line with slope consistent with the fourth order accuracy of the 5PBU4 discretization. First, consider the solid lines in the figure, which refer to the DASSL results. For $e = 10^{-3}$, the error line initially lies on that straight line, but as N_ζ increases the error does not decrease, because the error tolerance e of the ODE solver is not tight enough. In other words, if e is set to 10^{-3}, there is no point in using a $\Delta\zeta$ smaller than about 20^o because the errors in the integration with respect to ψ will nullify any gains in accuracy brought about by a finer ζ mesh. For $e = 10^{-4}$, the error curve leaves the "fourth order slope" line for a $\Delta\zeta$ of about 10^o, which is therefore the appropriate match for that value of e. Similarly, for $e = 10^{-5}$ and $e = 10^{-6}$ there is no accuracy advantage in reducing $\Delta\zeta$ to less than about 5^o and 2^o, respectively.

The same general considerations also apply to the DE/STEP results, shown in Fig. 4 with dashed lines. However, for the same value of e, the DE/STEP results are more accurate than the DASSL results. Therefore, for a tolerance $e = 10^{-3}$ it is justified to use values of $\Delta\zeta$ as small as about 5^o, and the smallest value used in this study $\Delta\zeta = 2.25^o$ is appropriate for all tolerances equal to 10^{-4} and tighter.

The same type of information for the 2PCD2 and the 2PU1 discretizations is presented in Figg. 5 and 6, respectively. For the second-order 2PCD2 the error tolerance e is almost never the driver for the RMS error, with the only exception of the DASSL solution with $e = 10^{-3}$, which is not tight enough to justify values of $\Delta\zeta$ smaller than about 10^o. For the first order 2PU1 formula, only $\Delta\zeta$ determines the error, and $e = 10^{-3}$ is sufficient for every value of $\Delta\zeta$.

Figures 2 through 6 obviously took advantage of the availability of an exact analytical solution. With more realistic and sophisticated wake models, where an analytical solution would not be available, a high accuracy, grid independent solution could be obtained and used as a "truth" model.

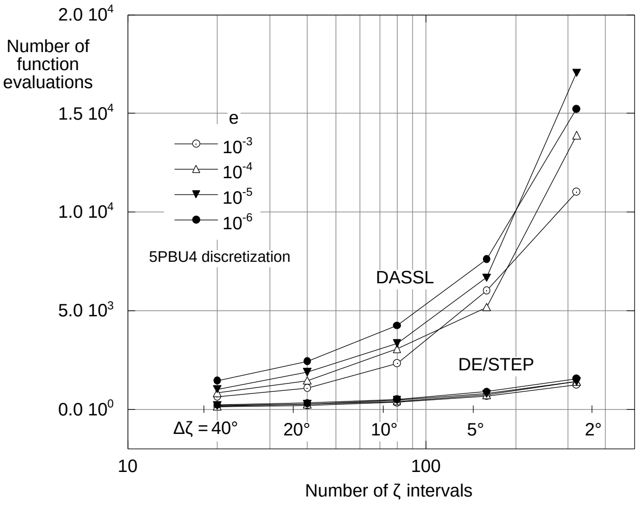

Figure 7: Number of function evaluations as a function of ζ-discretization, for both ODE solvers and several values of the local error tolerance e; 5PBU4 discretization.

The number of function evaluations required is shown in Fig. 7 as a function of N_ζ for both solvers and several values of e. One function evaluation is one evaluation of Eq. (19) for DE/STEP or one evaluation of Eq. (34) for DASSL. Clearly, the computational effort for one function evaluation increases with N_ζ because the number of ODE to be integrated increases. As a consequence, the number of function evaluations is not a completely representative measure of CPU time. On the other hand, CPU time would not be completely representative either, because Eqs. (19) and (34) have a very simple structure that would not be present if instead of Eq. (2) one would solve the more precise and realistic Eq. (1). The y-axis in Fig. 7 does not show a $\Delta\psi$ step size, which could perhaps be more intuitive. In fact, both DASSL and DE/STEP are variable-step and predictor-corrector solvers, which means that the ψ-step size is far from constant, especially at the beginning of the integration, that the solver may try different $\Delta\psi$ in the same step before deciding which one to accept, and that Eqs. (19) and (34) could be evaluated more than once for the same value of ψ. Therefore, even an "equivalent" $\Delta\psi$ obtained by dividing the number of function evaluations by the total integration length would be essentially meaningless.

Figure 7 clearly shows that DASSL is the more expensive solver, requiring about one order of magnitude more function evaluations than DE/STEP for the same local error tolerance e. This is to be expected, because implicit solvers like DASSL are typically more computationally expensive than explicit solvers like DE/STEP. Also, the number of function evaluation for DE/STEP increases more slowly with tighter local tolerances: compared with $e = 10^{-3}$, only about 25% more function evaluations are needed for $e = 10^{-6}$. The corresponding figure for DASSL is of about 55%. The number of function evaluations also increases more slowly with N_ζ for DE/STEP than DASSL. The very gradual increase of function evaluations for DE/STEP shown in Fig. 7 can be interpreted as a sign that the potential increase in stiffness for decreasing $\Delta\zeta$ does not materialize in practice, at least for the range of $\Delta\zeta$ considered. Therefore, DE/STEP is perfectly adequate, despite being an explicit solver.

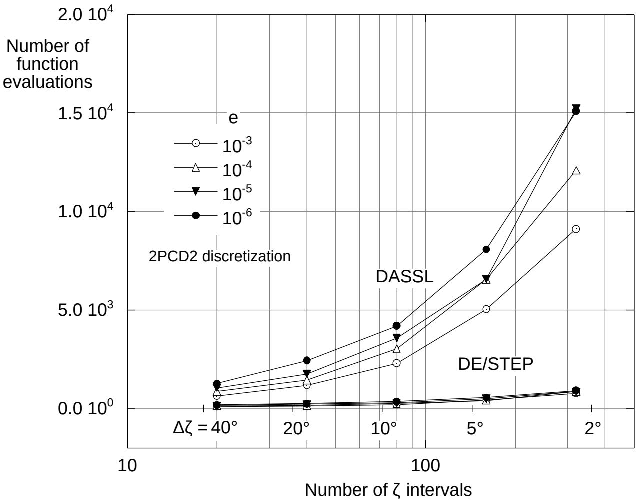

Figure 8: Number of function evaluations as a function of ζ-discretization, for both ODE solvers and several values of the local error tolerance e; 2PCD2 discretization.

The corresponding data for the 2PCD2 and the 2PU1 discretizations are shown in Figg. 8 and 9, respectively.

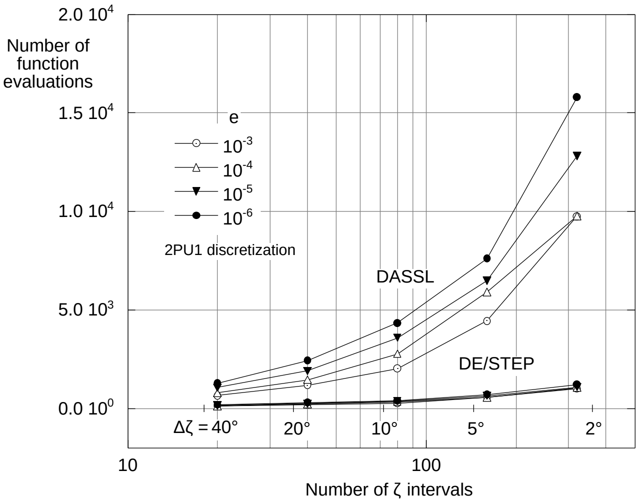

Figure 9: Number of function evaluations as a function of ζ-discretization, for both ODE solvers and several values of the local error tolerance e; 2PU1 discretization.

It is interesting to note that the number of function evaluations does not change noticeably with the type of ζ-discretization. This can be probably explained by the modest amount of coupling among the ODE indicated by the banded structure of the various $[A_\zeta]$ matrices for the simplified Eq. (2). For a more sophisticated wake model, blade-vortex and vortex-vortex interactions may create transients or discontinuities that will be captured in different ways by different ζ-discretizations and this, in turn, could create a stronger connection between number of function evaluations and specific discretizations.

The computational effort using DASSL is not uniformly distributed throughout the integration. Figure 10 shows the cumulative number of function evaluations as the integration progresses, for both solvers, the 5PBU4 discretization, $N_\zeta = 320$, and several values of e. (The DE/STEP data essentially lie on a straight line for all values of e.) The figure shows three interesting features. The first is that a considerable portion of the computational effort, especially for the higher accuracy solutions, is expended at the beginning of the integration. This is the phase in which DASSL determines an appropriate initial step size and BDF formula order. The second feature is that the computational effort shows clear "jumps" in selected phases of the integration. These are caused by the finite-difference recalculation of the Jacobian of the system of ODE [21]. Because Jacobian calculations are expensive, DASSL attempts to perform them as infrequently as possible, and will do so only when the Newton iteration inherent in the implicit method fails to converge [20]. Many Jacobian calculations are needed at the beginning of the integration, and this explains the computational effort there. The third feature apparent in Fig. 10 is that, after the initial step size and order have been determined, and whenever Jacobians are not

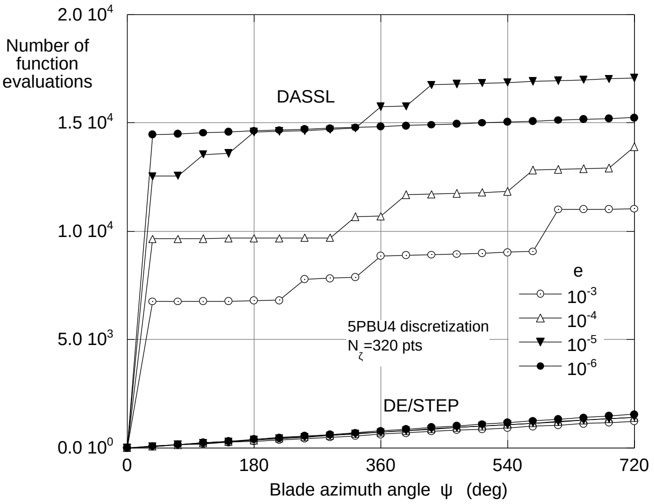

Figure 10: Cumulative number of function evaluations during the ψ-integration, for both ODE solvers and several values of the local error tolerance e; 5PBU4 discretization, N_ζ=320 points.

recalculated, DASSL requires fewer function evaluations than DE/STEP (the slope of the lines in the figure is smaller).

The implications for the use of DASSL in a more realistic situation (e.g., in the solution of Eq. (1) instead of the simpler Eq. (2) are threefold. First, on the positive side, the computational effort can probably be reduced by providing DASSL with better information on initial step size and order. This was not done in the present study. Second, also on the positive side, after the initial overhead the difference in number of function evaluations between DASSL and DE/STEP is not as great as Fig. 7 indicates. This is important if the integration is continued beyond the two rotor revolutions of the present study. Third, and this on the negative side, the computational effort is likely to increase in more realistic situations. Although DASSL can handle discontinuities and sharp transients very well, the stronger the discontinuity the more DASSL needs to adjust step size and order. For a true step change in forcing function, i.e., one with infinite slope, DASSL tends to behave as if it were restarting the integration from the time of the step change, i.e., it computes many Jacobians and it requires many function evaluations [21]. In a typical application to Eq. (1) one should expect, depending on the flight condition, to encounter strong transients associated with vortex-vortex and blade-vortex interaction.

All this attention to the behavior of DASSL, when the results clearly indicate that DE/STEP is the more efficient solver, is for two main reasons. The first, and more important, stems from one key motivation for the conversion of the vortex wake model to state-space form, namely, the consistent coupling with an ODE-based model of helicopter dynamics (the other motivation, i.e., the extraction of a linearized model, does not involve integration). Because the most sophisticated dynamic models of a helicopter end up being of the form $\dot{\mathbf{x}} = \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}; t)$, with no convenient way to convert them to the explicit form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; t)$ required by most ODE solvers, the ability of DASSL to accept equations in the general implicit form $\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}; t) = \mathbf{0}$ can dramatically reduce the effort of formulating or modifying the helicopter equations of motion (see Ref. [21] for a more detailed discussion of this issue). In other words, even if DASSL is not the most efficient solver for a MOL-based solution of the wake equations, its use could still be justified for the benefits in the formulation of the overall helicopter model.

The second reason for the focus on DASSL is that, with a more realistic wake model, the ζ-discretization might have to be refined (whether by static or adaptive remeshing) to capture discontinuities or transients in the ζ-direction. This could make ODE stiffness an issue, even if it is not one in the present study. The computational effort required by an explicit solver like DE/STEP might then become greater than that of an implicit solver like DASSL.

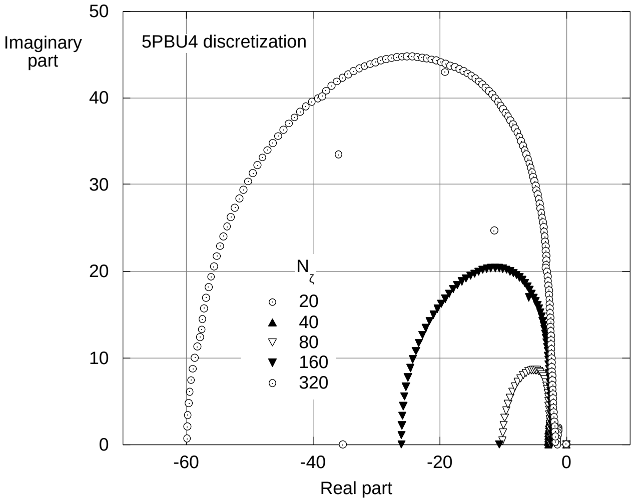

Figure 11: Linearized system poles for increasing number N_ζ of intervals in which the ζ-domain has been discretized; 5PBU4 ζ-discretization.

The eigenvalues (or poles) of the linearized state-space wake model are shown in Fig. 11, for the 5PBU4 discretization, and several values of the number of intervals N_ζ. Obviously, in each case there will be N_ζ eigenvalues, which will appear as real or complex conjugate pairs. The vast majority of the eigenvalues follow a characteristic pattern that starts in the neighborhood of the origin and ends near the real axis, with increasing real parts, and imaginary parts that first increase and then decrease. Frequency and damping ratio increase constantly from the beginning to the end of the pattern.

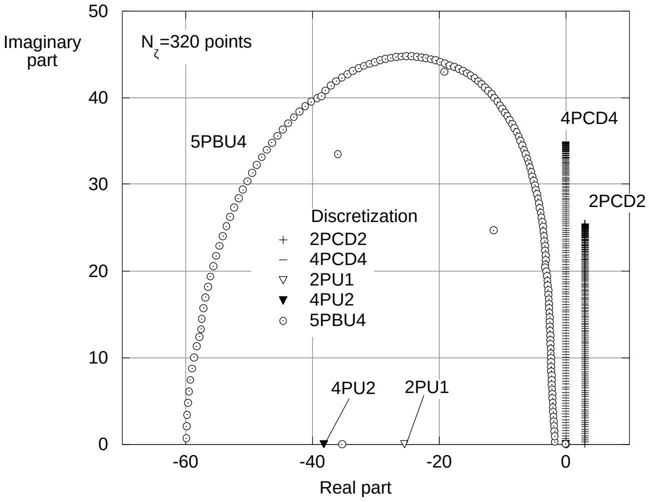

Figure 12: Linearized system poles for the five ζ-discretizations; $N_\zeta = 320$. (Note: the offset with respect to the imaginary axis of the pure imaginary 2PCD2 poles is only for visual clarity.)

The figure clearly shows that as N_ζ increases, i.e., as the ζ-discretization is refined, the maximum frequency of the eigenvalues also increases, i.e., the system of ODE becomes stiffer.

A few eigenvalues appear outside the pattern for every value of N_ζ, and are most likely caused by the fact that the coefficient of first and last rows of the $[A_\zeta]$ matrix are different from those of the rest of the diagonal band. Some of these "spurious" eigenvalues have a frequency that is far lower than the rest and, in turn, some of them have positive real parts, i.e., they are unstable. For example, for $N_\zeta = 20$, there is a real positive eigenvalue at 0.0023 and a complex conjugate pair at $0.0007 \pm 0.00216i$. They both correspond to a frequency of about 0.0023 rad/sec, whereas the next higher eigenvalue has a frequency of 1.44 rad/sec, and the highest has a frequency of 2.27 rad/sec. The corresponding eigenvalues for $N_\zeta = 320$ are 0.0488 and $0.0151 \pm 0.0466i$, corresponding to a frequency of about 0.049 rad/sec, with the rest of the eigenvalues ranging in frequency from 1.77 to 59.9 rad/sec.

To verify that these very low frequency, slightly unstable eigenvalues would not affect negatively the stability of long term wake calculations, the solution for $N_\zeta = 320$ and the 5PBU4 discretization was carried out for 150 revolutions, instead of the 2 used throughout this study. No instability in the solution or degradation of accuracy occurred.

Finally, Fig. 12 shows the eigenvalues of $[A_\zeta]$ for all five discretizations considered in this study, and $N_\zeta = 320$. Except for the "spurious" eigenvalues, which occur for all five discretizations, the eigenvalues appear in clearly defined patterns. If $[A_\zeta]$ is obtained from central difference formulas (2PCD2 and 4PCD4), the eigenvalues are purely imaginary (those corresponding to 2PCD2 are shown in the figure as slightly offset from the imaginary axis, for visual clarity). If upwind formulas are used (2PU1 and 4PU2) the eigenvalues collapse into a single value for each case. The pattern for the biased upwind formula (5PBU4) has already been discussed.

Because of their strong dependence on the discretization scheme, it is clear that the eigenvalues shown in Figs. 11 and 12 reflect more the numerical characteristics of $[A_\zeta]$ than the underlying physics. The results obtained with the upwind formulas 2PU1 and 4PU2 seem to be more in line with physical intuition, because the eigenvalues are all the same (except for a few spurious ones) and correspond to asymptotic, well damped motions. However, a better grounded interpretation will require a careful study of the poles of the more complete wake model, Eq. (1), rather than the simplified model of the present study. Future research should also explore the *eigenvectors* of the linearized model, whether they can be used in a modal coordinate transformation to reduce the size of the system of wake ODE, and the relationship with the wake modes recently identified and studied by Bhagwat and Leishman [23] starting from a perturbation analysis of the governing wake equations.

Summary and Conclusions

This paper has presented the application of the method of lines to the vorticity transport equation, which is the foundation for many mathematical models of helicopter rotor vortex wakes. The method of lines transforms the governing PDE into a system of ODE, and therefore into a state-space model. This model can then be coupled to other ODE modeling helicopter dynamics, or used to extract a linearized model suitable for flight dynamic and aeroelastic stability analyses, frequency response calculations, and rotor and flight control system design applications.

In this paper, the MOL has been applied to a simplified version of the vorticity transport equation, corresponding to a rigid vortex wake and uniform inflow, for which an exact analytical solution exists. Because this simplified equation does not model very important wake physics, such as the interaction between wake geometry and inflow, the study should be considered only as a first step toward the application of the MOL to realistic vortex wake models. Therefore, the conclusions listed below only apply to the simplified problem considered. Future research will determine whether they can also be extended to more sophisticated wake models.

The main conclusions of the present study are:

1. The method of lines is a convenient, rigorous, easy to apply method to formulate vortex wake models in state-space form. The solutions it generates are accurate and numerically stable.

2. Refining the space discretization of the equations increases the stiffness of the resulting system of ODE,

but not to the point that explicit ODE solvers cannot be used. For the finest discretization used in this study, corresponding to a $\Delta\zeta$ of about 2^o, no stiffness was apparent.

3. The best computational efficiency is obtained when the accuracies of the space (or ζ-) and time (or ψ-) discretizations are matched. They can be controlled, respectively, through the step size $\Delta\zeta$ and the local error tolerance e of the ODE solver. First-, second-, and fourth-order formulas were used in the ζ-discretization, and did produce errors consistent with the respective orders of accuracy.

4. For all the cases studied, the explicit, Adams-Bashforth based solver DE/STEP was much more computationally efficient than the implicit, BDF based solver DASSL. However, DASSL greatly simplifies the formulation of the helicopter model to which the wake is likely to be coupled, and therefore it should still be considered as an option for the wake solution itself. Additionally, the required ζ-discretization in more realistic problems could become so fine that stiffness would in fact become a problem, and require implicit solvers like DASSL.

5. The number of required function evaluations for each solver depends primarily on the desired local error tolerance and the step size $\Delta\zeta$, but very little on the type of ζ-discretization.

6. Linearized state-space wake models can be easily obtained. For the cases studied, the eigenvalues reflected the numerics of the problem, rather than the physics. Further research should clarify the connection between these eigenvalues and the wake dynamics.

Finally, because the method of lines can be applied in principle to any PDE, or system of PDE, its potential usefulness is not limited to vortex wakes. In fact, it could provide a systematic methodology to obtain state-space models from a wide variety of CFD formulations, and therefore increase the accuracy of simulation models for helicopter aeromechanics and flight dynamics.

Acknowledgments

This research was supported by the National Rotorcraft Technology Center under the Rotorcraft Center of Excellence Program, Technical Monitor Dr. Y. Yu.

References

[1] Leishman, J. G., and Nguyen, K. Q., "State Space Representation of Unsteady Airfoil Behavior," *AIAA Journal*, Vol. 28, (5), May 1990, pp. 836-844.

[2] Petot, D., "Differential Equation Modeling of Dynamic Stall," *La Recherche Aérospatiale*, 1989-5.

[3] Peters, D. A., Morillo, J. A., and Nelson, A. M., "New Developments in Dynamic Wake Modeling for Dynamics Applications," *Journal of the American Helicopter Society*, Vol. 48, (2), Apr 2003, pp. 120-127.

[4] Joglekar, M., and Loewy, R., "An Actuator-Disc Analysis of Helicopter Wake Geometry and the Corresponding Blade Response," USAAVLABS Technical Report 69-66, Dec 1970.

[5] Arnold, U. T. P. and Keller, J. D. and Curtiss, H. C., Jr., and Reichert, G., "The Effect of Inflow Models on the Predicted Response of Helicopters," *Journal of the American Helicopter Society*, Vol. 43, (1), Jan 1998, pp. 25-36.

[6] Basset, P.-M., "Modeling of the Dynamic Inflow on the Main Rotor and the Tail Components in Helicopter Flight Mechanics," *American Helicopter Society 53rd Annual Forum*, Virginia Beach, VA, Apr 1997, pp. 1643-1656.

[7] Rosen, A., "Approximate Actuator Disk Model of a Rotor in Hover or Axial Flow Based on Vortex Modeling," *Journal of the American Helicopter Society*, Vol. 49, (1), Jan 2004, pp. 66-79.

[8] Rosen, A., "Approximate Actuator Disk Model of a Rotor in Hover or Axial Flow Based on Potential Flow Equations," *Journal of the American Helicopter Society*, Vol. 49, (1), Jan 2004, pp. 80-92.

[9] Fletcher, J. W., and Tischler, M. B., "Improving Helicopter Flight Mechanics Models with Laser Measurements of Blade Flapping," *American Helicopter Society 53rd Annual Forum*, Virginia Beach, VA, Apr 1997, pp. 1467-1994.

[10] Rosen, A., Yaffe, R., Mansur, M. H., and Tischler, M. B., "Methods for Improving the Modeling of Rotor Aerodynamics for Flight Mechanics Purposes," *American Helicopter Society 54th Annual Forum*, Washington, DC, May 1998, pp. 1337-1355.

[11] Mansur, M. H., and Tischler, M. B., "An Empirical Correction for Improving Off-Axes Responses in Flight Mechanics Helicopter Models," *Journal of the American Helicopter Society*, Vol. 43, (2), Apr 1998, pp. 94-102.

[12] Leishman, J. G., Bhagwat, M. J., and Bagai, A., "Free-Vortex Filament Methods for the Analysis of Helicopter Rotor Wakes," *Journal of Aircraft*, Vol. 39, (5), Sep-Oct 2002, pp. 759-775.

[13] Johnson, W., "Time-Domain Unsteady Aerodynamics of Rotors with Complex Wake Configurations," *Vertica*, Vol. 12, No. 1/2, 1988, pp. 83-100.

[14] Bagai, A., and Leishman, J.G., "Rotor Free-Wake Modeling using a Pseudo-Implicit Technique—Including Comparison with Experimental Data," *Journal of the*

American Helicopter Society, Vol. 40, (3), Jul 1995, pp. 29-41.

[15]Bagai, A., "Contribution to the Mathematical Modeling of Rotor Flow-Fields Using a Pseudo-Implicit Free-Wake Analysis," Ph.D. Dissertation, Department of Aerospace Engineering, University of Maryland, College Park, 1995.

[16]Schiesser, W. E., *The Numerical Method of Lines: Integration of Partial Differential Equations*, Academic Press, San Diego, 1991.

[17]Carver, M. B., and Hinds, H. W., "The Method of Lines and the Advection Equation," *Simulation*, Vol. 31, (2), Aug 1978, pp. 59-69.

[18]Vande Wouver, A., Saucez, Ph., and Schiesser, W. E., (eds.), *Adaptive Method of Lines*, Chapman and Hall/CRC, 2001.

[19]Shampine, L. F., and Gordon, M. K., *Computer Solution of Ordinary Differential Equations—The Initial Value Problem*, W. H. Freeman and Company, San Francisco, 1975.

[20]Brenan, K. E., Campbell, S. L., and Petzold, L. R., *The Numerical Solution of Initial Value Problems Using Differential-Algebraic Equations*, Elsevier Science Publishing Co., New York, 1989.

[21]Celi, R., "Implementation of Rotary-Wing Aeromechanical Problems Using Differential-Algebraic Equation Solvers," *Journal of the American Helicopter Society*, Vol. 45, (4), Oct 2000, pp. 253-262.

[22]Skelboe, S., "The Control of Order and Steplength for Backward Differentiation Methods," *BIT*, Vol. 17, (1), Jan 1977, pp. 91-107.

[23]Bhagwat, M. J., and Leishman, J. G., "Stability, Consistency and Convergence of Time Marching Free-Vortex Rotor Wake Algorithms," *Journal of the American Helicopter Society*, Vol. 46, (1), Jan 2001, pp. 59-71.