APPROXIMATING PERIODIC SOLUTIONS OF AUTONOMOUS DELAY DIFFERENTIAL EQUATIONS

DAVID E. GILSINN

Mathematical and Computational Sciences Division
National Institute of Standards and Technology
100 Bureau Drive, Stop 8910
Gaithersburg, MD 20899-8910 [1]

ABSTRACT. Machine tool chatter has been characterized as isolated periodic solutions or limit cycles of delay differential equations. Determining the amplitude and frequency of the limit cycle is sometimes crucial to understanding and controlling the stability of machining operations. In Gilsinn [9] a result was proven that says that, given an approximate periodic solution and frequency of an autonomous delay differential equation that satisfies a certain non-criticality condition, there is an exact periodic solution and frequency in a computable neighborhood of the approximate solution and frequency. The proof required the estimation of a number of parameters and the verification of three inequalities. In this paper the details of the algorithms will be given for estimating the parameters required to verify the inequalities and to compute the final approximation errors. An application will be given to a Van der Pol oscillator with delay in the nonlinear terms. A MATLAB m-file implementing the algorithms discussed in the paper is given in the appendix.

AMS (MOS) Subject Classification. 34K11, 34K13, 34K28.

1. Introduction

Machine tool dynamics has been modeled using delay differential equations for a number of years as is clear from the vast literature associated with it. For a detailed review of machining dynamics see Tlusty [32]. For a discussion of dynamics in milling operations see Balchandran [1] and Zhao and Balachandran [34]. For drilling operations see Stone and Askari [29] and Stone and Campbell [30]. For an analysis of chatter occurring in turning operations see Hanna and Tobias [16], Marsh et al. [21], and Nayfeh et al. [22]. Machine tool chatter is undesirable self-exited periodic oscillations during machining operations. It has been identified as a Hopf bifurcation of limit cycles from steady state solutions. For a way of estimating the critical Hopf bifurcation parameters that lead to machine tool chatter see Gilsinn [8].

In studying the effects of chatter it is sometimes desirable to compute the amplitude and frequency of the limit cycle generating the chatter. This entails solving the delay differential equations that model the machine tool dynamics. There is a large literature on numerically solving delay differential equations. Some representative methods are described in Banks and Kappel [2], Engelborghs et al. [5], Kemper [20], Paul [23], Shampine and Thompson [25], and Willé and Baker [35]. Although these methods generate solution vectors that can be studied by harmonic and power spectral methods to estimate the frequency of periodic cycles, they do not directly generate a representative model of a limit cycle such as a Fourier series representation.

It is also desirable to know whether a representation of an approximate limit cycle is close to a true limit cycle. In other words we wish to answer the question as to whether the approximate solution represents sufficiently well a true solution. This is answered with a set of test criteria by Gilsinn [9], who showed that, given a representative approximate solution and frequency for a periodic solution to the autonomous delay differential equation

$$\dot{x} = X(x(t), x(t-h)), \tag{1}$$

where $x, X \in \mathbb{R}^n$, $h > 0$, X sufficiently differentiable, there are conditions, depending on a number of parameters, for which (1) has a unique exact periodic solution and frequency in a computable neighborhood of the approximate solution and frequency. This result was first established in a very general manner for functional differential equations by Stokes [28] who extended an earlier result for ordinary differrential equations in Stokes [27]. A crucial aspect in applying the result involves verify a certain "non-criticality" condition. However, no computable algorithms were given in the case of functional differential equations to estimate the various parameters or verify the "non-criticality" condition. Only recently have algorithms been developed to computationally verify these conditions in the fixed delay case. A preliminary announcement of algorithms for computing these parameters and verifying the "non-criticality" condition was given by Gilsinn [7]. In this paper we include a more detailed discussion of the algorithms and apply them to a Van der Pol equation with delay in its nonlinear terms.

The result of Stokes [28] for functional differential equations depends on verifying certain conditions that require computing various parameters. In order to apply Stokes' result a proof in the case of equation (1) will be given here since certain inequalities that are developed within the proof are necessary for proving the fixed point contraction mapping conditions and rely on the specific form of (1).

The notation used in the paper is described in Section 2. The non-criticality condition is defined in Section 3. In Section 4 we construct an exact frequency and 2π-periodic solution of (1) as a perturbation problem. In Section 5 we define a map that is used to prove, by a contraction argument, the existence of an exact frequency and 2π-periodic solution of (1). The main contraction theorem is proven in Section 6. A Galerkin algorithm to compute a 2π-periodic solution to a nonlinear autonomous delay differential equation is given in Section 7. The Floquet Theory for delay differential equations is discussed in Section 8. An algorithm for computing the characteristic multipliers of the variational equation of (1) with respect to the approximate 2π-periodic solution, is outlined in Section 9. An algorithm to determe the solution to the formal adjoint equation with respect to the variational equation of (1) with respect to the approximate 2π-periodic solution, is outlined in Section 10. An algorithm for estimating a critical parameter is given in Section 11. An application of these algorithms to the Van der Pol equation with delay is given in Section 12. Conclusions are given in Section 13 and a disclaimer is given in Section 14. The derivation of the differentiation matrix (86) is given in Appendix 1, Section 15. Certain bounds and Lipschitz conditions used in the fixed point theorem are proven in Appendices 2 and 3 (Section 16 and Section 17). The MATLAB code and associated support functions implementing the algorithms are given in Appendix 4, Section 18.

2. **Notation**

Let C_ω denote the space of continuous functions from $[-\omega, 0]$ to $\mathbf{C}^n$ with norm in C_ω given by $|\phi| = \max|\phi(s)|$ for $-\omega \le s \le 0$, where

$$|\phi(s)| = \left(\sum_{i=1}^{n} |\phi_i(s)|^2\right)^{1/2}. \tag{2}$$

C_ω is a Banach space with respect to this norm. Let $\mathcal{P}$ be the space of continuous 2π-periodic functions with sup norm, $|\cdot|$ on $(-\infty, \infty)$. Let $\mathcal{P}_1 \subset \mathcal{P}$ be the subspace of continuously differentiable 2π-periodic functions with the sup norm. Let $X(x, y)$ be continuously differentiable in some domain $\Omega_n \subset \mathbf{C}^n \times \mathbf{C}^n$ with bounded derivatives where

$$|X_i(x, y)| \le \mathcal{B}, \tag{3}$$

for $i = 1, 2$, $(x, y) \in \Omega_n$. The subscripts of X indicate derivatives with respect to the first and second variables of X respectively. We further assume that the first partial derivatives satisfy Lipschitz conditions given by

$$|X_i(x_1, y_1) - X_i(x_2, y_2)| \leq \mathcal{K}(|x_1 - x_2| + |y_1 - y_2|), \tag{4}$$

for $(x_1, y_1),\ (x_2, y_2) \in \Omega_n$.

In order to simplify the notation for (1) we will first normalize the delay h to unity by setting $s = t/h$. Then, (1) becomes

$$\frac{dy}{ds}(s) = h\, X(y(s), y(s-1)) \tag{5}$$

where $y(s) = x(s\, h)$. Therefore we will assume $h = 1$ in (1). We will also make one further transformation. Since the period $T = 2\pi/\omega$ of a periodic solution for (1) is unknown we can normalize the period to $[0, 2\pi]$ by introducing the substitution of t/ω for t and rewriting (1), with $h = 1$, in the form

$$\omega\, \dot{x} = X(x(t), x(t - \omega)). \tag{6}$$

For $\psi_1, \psi_2 \in \mathcal{P}$ we denote the total derivative of $X(x, y)$ by

$$dX(x, y; \psi_1, \psi_2) = X_1(x, y)\psi_1 + X_2(x, y)\psi_2. \tag{7}$$

Let $A(t),\ B(t)$ be continuous 2π-periodic matrices. Then a characteristic multiplier is defined as follows.

Definition 2.1. *ρ is a* characteristic multiplier *of*

$$\dot{y} = A(t)y(t) + B(t)y(t - \omega) \tag{8}$$

if there is a nontrivial solution $y(t)$ of (8) such that $y(t + 2\pi) = \rho\, y(t)$. Note that if $\rho = 1$ then $y(t)$ is 2π-periodic.

To simplify some of the notation we will suppress the t and write, for example, $x = x(t),\ x_\omega = x(t - \omega)$, but in other cases we will maintain the t, especially when describing computational steps. We will also at times use the notation

$$|x|_2 = \left[\int_0^{2\pi} |x(t)|^2\ dt\right]^{1/2}. \tag{9}$$

3. Non-criticality Condition

Galerkin and harmonic balance methods can be used to develop 2π-periodic approximate solutions for (6). A fast discrete Fourier series algorithm for computing an approximate series solution and frequency, $(\hat{\omega}, \hat{x})$, has been given by Gilsinn [8]. See Section 7 below for a brief discussion of a Galerkin method for approximating a solution. At this point, then, we assume that we have developed a 2π-periodic approximate solution and frequency, $(\hat{\omega}, \hat{x})$ for (6), where $\hat{x}$ is 2π-periodic and

$$\hat{\omega}\dot{\hat{x}} = X(\hat{x}, \hat{x}_{\hat{\omega}}) + k \tag{10}$$

where $k(t)$ is a 2π-periodic residual bounded by

$$|k| \leq r. \tag{11}$$

The required size of the residual error, r, will become clear based upon estimates later in this paper. These estimates will indicate in particular situations how accurately an approximate solution and frequency would need to be computed.

The variational equation with respect to the approximate solution and frequency is given by

$$\hat{\omega}\dot{z} = dX\left(\hat{x}, \hat{x}_{\hat{\omega}}; z, z_{\hat{\omega}}\right). \tag{12}$$

Let $\hat{A} = X_1\left(\hat{x}, \hat{x}_{\hat{\omega}}\right),\ \hat{B} = X_2\left(\hat{x}, \hat{x}_{\hat{\omega}}\right)$. The formal adjoint of (12) is given in row form by

$$\hat{\omega}\dot{v} = -v\hat{A} - v_{-\hat{\omega}}\hat{B}. \tag{13}$$

The next lemma, proven in Halanay [14], relates the number of independent 2π-periodic solutions of (12) to those of (13).

Lemma 3.1. *System (12) and (13) have the same finite number of independent 2π-periodic solutions.*

We will not give the proof of the next lemma, since it is also proven in Halanay [14]. The result, however, will be critical to the main approximation theorem.

Lemma 3.2. *The nonhomogeneous system*

$$\hat{\omega}\dot{x} = \hat{A}x + \hat{B}x_{\hat{\omega}} + f \tag{14}$$

has a unique 2π-periodic solution if and only if

$$\int_0^{2\pi} v_0^T f dt = 0 \tag{15}$$

for all independent solutions v_0 of period 2π of (13). Furthermore there exists an $M > 0$, independent of f, such that

$$|x| \leq M|f|. \tag{16}$$

We will give the proof of the next lemma. Although it is stated in Hale [15] and in Halanay [14], the proof is not generally available. The result, however, motivates the definition of a non-critical approximate solution.

Lemma 3.3. *Let $\rho = 1$ be a simple (i.e. multiplicity one) characteristic multiplier of (12) Let p be a non-trivial solution of (12) associated with ρ. Define*

$$J(p, \hat{\omega}) = p + \hat{B}p_{\hat{\omega}}, \tag{17}$$

then

$$\int_0^{2\pi} {v_0}^T J(p, \hat{\omega}) dt \neq 0 \tag{18}$$

for all independent solutions v of the adjoint (13).

Proof. Let $y(t)$ be any 2π-periodic solution of (12) and write

$$z = y + tp \tag{19}$$

Then, substituting (19) into (12), write

$$\begin{aligned} \hat{\omega}\dot{z} &= \hat{A}z + \hat{B}z_{\hat{\omega}} + \hat{\omega}\left(p + \hat{B}p_{\hat{\omega}}\right) \\ &= \hat{A}z + \hat{B}z_{\hat{\omega}} + \hat{\omega}J(p, \hat{\omega}) \end{aligned} \tag{20}$$

We will suppose

$$\int_0^{2\pi} v^T J(p, \hat{\omega}) dt = 0 \tag{21}$$

and show a contradiction. Since ρ is a simple characteristic multiplier then p is the only non-trivial 2π-periodic solution associated with ρ and so there is only one solution of (13) associated with $1/\rho$. Lemma 3.2 and (21) imply that there is a unique 2π-periodic solution z of (20). With z and p both 2π-periodic let $w = z - tp$. w cannot be a multiple of p for there would be a t_0 such that $t_0 p = z - tp$ or $z = (t_0 + t)\, p$. Since z, p are 2π-periodic we have $z = (t_0 + t + 2\pi)\, p$. But then we would have $t_0 = t_0 + 2\pi$ or $2\pi = 0$, a contradiction. QED

Lemma 3.2 will imply, in the case of a non-critical 2π-periodic approximate solution of (6), that there is only one v_0 in (22).

We can now give the definition of a non-critical approximate solution of (6).

Definition 3.4. *The pair $(\hat{\omega}, \hat{x})$, where $\hat{x}$ is at least twice continuously differentiable, is said to be non-critical with respect to (6) if (1) the variational equation about the approximate solution $\hat{x}$, given by (12), has a simple characteristic multiplier ρ_0 with all of the other characteristic multipliers not equal to one. (2) If v_0, $|v_0|_2 = 1$, is the solution of (13) corresponding to ρ_o, i.e. with multiplier $1/\rho_0$, then*

$$\int_0^{2\pi} {v_0}^T J(\dot{\hat{x}}, \hat{\omega}) dt \neq 0, \tag{22}$$

where

$$J(\dot{\hat{x}}, \hat{\omega}) = \dot{\hat{x}} + \hat{B}\dot{\hat{x}}_{\hat{\omega}} \tag{23}$$

4. A Perturbation Problem

In this paper we will look for an exact 2π-periodic solution, x, and an exact frequency, ω, for (6) as a perturbation of the 2π-periodic approximate solution , $\hat{x}$, and approximate frequency, $\hat{\omega}$, of (6). In particular, let

$$\begin{aligned} \omega &= \hat{\omega} + \beta \\ x &= \hat{x} + \frac{\hat{\omega}}{\omega} z \end{aligned} \tag{24}$$

Then, substituting (24) into (6) and using (10), we can write the equation for z and β as

$$\hat{\omega}\dot{z} = dX\left(\hat{x}, \hat{x}_{\hat{\omega}}; z, z_{\hat{\omega}}\right) + R(z, \beta) - \beta J\left(\dot{\hat{x}}, \hat{\omega}\right) - k \tag{25}$$

where

$$R(z, \beta) = \left[X\left(\hat{x} + \frac{\hat{\omega}}{\omega} z, \hat{x}_\omega + \frac{\hat{\omega}}{\omega} z_\omega\right) - X\left(\hat{x}, \hat{x}_{\hat{\omega}}\right)\right] - dX\left(\hat{x}, \hat{x}_{\hat{\omega}}; z, z_{\hat{\omega}}\right) + \beta \hat{B}\dot{\hat{x}}_{\hat{\omega}}. \tag{26}$$

and $J\left(\dot{\hat{x}}, \hat{\omega}\right)$ is given by (23).

In the next lemma we establish bounds and Lipschitz conditions for $R(z, \beta)$.

Lemma 4.1. *There exist functions $\mathcal{R}_0(z, \beta) > 0$, $\mathcal{R}_i(z, \beta, \tilde{z}, \tilde{\beta}) > 0$, $i = 1, 2$, such that $\mathcal{R}_0 \to 0$ as $(z, \beta) \to 0$ and $\mathcal{R}_i \to 0$ as $(z, \beta, \tilde{z}, \tilde{\beta}) \to 0$ and*

$$\begin{aligned} |R(z, \beta)| &\leq \mathcal{R}_0(z, \beta) \\ \left|R(z, \beta) - R(\tilde{z}, \tilde{\beta})\right| &\leq \mathcal{R}_1(z, \beta, \tilde{z}, \tilde{\beta})\, |z - \tilde{z}| + \mathcal{R}_2(z, \beta, \tilde{z}, \tilde{\beta}) \left|\beta - \tilde{\beta}\right| \end{aligned} \tag{27}$$

Proof: Appendix 2.

Since we will be considering $|\beta|$ small, we will begin by restricting β, which could be negative, so that

$$\hat{\omega} + \beta \geq \frac{\hat{\omega}}{2}. \tag{28}$$

We can select $|\beta| \leq \hat{\omega}/2$.

As a first step to establishing the existence of a 2π-periodic solution of (25) we first study the existence of a 2π-periodic solution of

$$\hat{\omega}\dot{z} = dX\left(\hat{x}, \hat{x}_{\hat{\omega}}; z, z_{\hat{\omega}}\right) + g - \beta J\left(\dot{\hat{x}}, \hat{\omega}\right) - k \tag{29}$$

where $g \in \mathcal{P}$. For this we have the following lemma

Lemma 4.2. *If $(\hat{\omega}, \hat{x})$ are non-critical with respect to (6), then (a) there exists a unique β such that*

$$g - \beta J\left(\dot{\hat{x}}, \hat{\omega}\right) - k \perp v_0 \tag{30}$$

where v_0 is the solution of (13) corresponding to the characteristic multiplier ρ_0 of (12), and (b) there exists a unique 2π-periodic solution of (29) that satisfies

$$|z| \leq M \left|g - \beta J\left(\dot{\hat{x}}, \hat{\omega}\right) - k\right| \tag{31}$$

for some $M > 0$.

Proof: Take

$$\beta = \alpha \left[\int_0^{2\pi} {v_0}^T (g - k)\, dt \right] \tag{32}$$

where

$$\alpha = \left[\int_0^{2\pi} {v_0}^T J\left(\dot{\hat{x}}, \hat{\omega}\right)\, dt \right]^{-1} \tag{33}$$

and apply Lemma 3.2.

We can now establish bounds on β, z and $\dot{z}$. For notation, designate the unique β and z in Lemma 4.2 by $\beta(g)$ and $z(g)$ respectively, and $\dot{z}$ by $\dot{z}(g)$.

Lemma 4.3. *There exist three constants, designated by* $\lambda_i,\ i = 0,\ 1,\ 2$, *such that*

$$\begin{aligned} |\beta(g)| &\leq \lambda_0(|g| + r) \\ |z(g)| &\leq \lambda_1(|g| + r) \\ |\dot{z}(g)| &\leq \lambda_2(|g| + r) \end{aligned} \tag{34}$$

Proof: From

$$\begin{aligned} |g|_2 &\leq \sqrt{2\pi}\ |g| \\ |k|_2 &\leq \sqrt{2\pi}\ |k| \end{aligned} \tag{35}$$

and the Cauchy-Schwarz inequality applied to (32)

$$|\beta(g)| \leq |\alpha| \left| {v_0}^T \right|_2 |g - k|_2 \leq \sqrt{2\pi} |\alpha| (|g| + r) \tag{36}$$

from the bound $|k| \leq r$.

From (31)

$$\begin{aligned} |z(g)| &\leq M \left[|g| + |k| + |\beta(g)| \left| J\left(\dot{\hat{x}}, \hat{\omega}\right) \right| \right], \\ &\leq M \left[1 + \sqrt{2\pi} |\alpha| \left| J\left(\dot{\hat{x}}, \hat{\omega}\right) \right| \right] (|g| + r) \end{aligned} \tag{37}$$

From (29)

$$|\hat{\omega}|\, |\dot{z}(g)| \leq |dX\left(\hat{x},\, \hat{x}_{\hat{\omega}}; z(g),\, z(g)_{\hat{\omega}}\right)| + \left| g - \beta J\left(\dot{\hat{x}}, \hat{\omega}\right) - k \right|$$

$$\leq \left[\mathcal{B}\left(|z(g)| + |z(g)_{\hat{\omega}}|\right)\right] \tag{38}$$

$$\leq (2M\mathcal{B}) \left[1 + \sqrt{2\pi} |\alpha| \left| J\left(\dot{\hat{x}}, \hat{\omega}\right) \right| \right] (|g| + r) \tag{39}$$

Therefore, from (36), (37), (38),

$$\begin{aligned} \lambda_0 &= \sqrt{2\pi} |\alpha|, \\ \lambda_1 &= M \left[1 + \sqrt{2\pi} |\alpha| \left| J\left(\dot{\hat{x}}, \hat{\omega}\right) \right| \right] \\ \lambda_2 &= \frac{\lambda_1}{|\hat{\omega}|\, M} (1 + 2M\mathcal{B}) \end{aligned} \tag{40}$$

5. A Map and its Properties

In the main approximation theorem we will show that the solution of the perturbation problem (25) is the fixed point of a particular contraction map. In this section we will define the map and establish some properties.

We begin by defining a subset of $\mathcal{P}$, designated by $\mathcal{N}_\delta$, as

$$\mathcal{N}_\delta = \{g \in \mathcal{P} : |g| \leq \delta\}, \tag{41}$$

where $\delta > 0$. Following Stokes [28] we will define a map $S : \mathcal{N}_\delta \to \mathcal{P}$ in terms of two mappings

$$\begin{aligned} L &: \mathcal{N}_\delta \to R \times \mathcal{P}_1, \\ T &: R \times \mathcal{P}_1 \to \mathcal{P}. \end{aligned} \tag{42}$$

To define L, let $g \in \mathcal{N}_\delta$, then Lemma 4.2 assures us of the existence of a unique $\beta(g)$ satisfying (30) and a unique solution $z(g)$ satisfying (29). Thus, define $L : \mathcal{N}_\delta \to R \times \mathcal{P}_1$ by

$$L(g) = (\beta(g), z(g)). \tag{43}$$

Now define $T : R \times \mathcal{P}_1 \to \mathcal{P}$ by

$$T(\beta,\ z) = R(z,\ \beta). \tag{44}$$

Finally, define $S : \mathcal{N}_\delta \to \mathcal{P}$ by

$$S(g) = T(L(g)) = R(z(g), \beta(g)). \tag{45}$$

Lemma 5.1. *For $g \in \mathcal{N}_\delta,\ \tilde{g} \in \mathcal{N}_\delta$ there exist two functions $E_1(\delta),\ E_2(\delta)$ and two positive constants $F_1,\ F_2$ so that*

$$\begin{aligned} |S(g)| &\le E_1(\delta), \\ |S(g) - S(\tilde{g})| &\le E_2(\delta)\,|g - \tilde{g}|\,, \end{aligned} \tag{46}$$

where

$$\begin{aligned} E_1(\delta) &\le F_1\delta^2, \\ E_2(\delta) &\le F_2\delta. \end{aligned} \tag{47}$$

Proof: From (45) and (27) we have

$$\begin{aligned} |S(g)| &\le \mathcal{R}_0(z(g), \beta(g)), \\ |S(g) - S(\tilde{g})| &\le \mathcal{R}_1(z(g), \beta(g), z(\tilde{g}), \beta(\tilde{g}))\,|z(g) - z(\tilde{g})| \\ &\quad + \mathcal{R}_2(z(g), \beta(g), z(\tilde{g}), \beta(\tilde{g}))\,|\beta(g) - \beta(\tilde{g})|\,. \end{aligned} \tag{48}$$

By Cauchy-Schwarz, the fact that $\left|{v_0}^T\right|_2 = 1$, and (40)

$$\begin{aligned} |\beta(g) - \beta(\tilde{g})| &\le |\alpha| \int_0^{2\pi} \left|{v_0}^T (g - \tilde{g})\right| dt, \\ &\le |\alpha| \left[\int_0^{2\pi} |g - \tilde{g}|^2\, dt\right]^{1/2}, \\ &\le \lambda_0\,|g - \tilde{g}|\,. \end{aligned} \tag{49}$$

From Lemma 4.2 and the definition of $\beta(g),\ \beta(\tilde{g})$ we have

$$\begin{aligned} &\int_0^{2\pi} {v_0}^T \left(g - \beta(g) J\left(\dot{\hat{x}}, \hat{\omega}\right) - k\right) dt = 0, \\ &\int_0^{2\pi} {v_0}^T \left(\tilde{g} - \beta(\tilde{g}) J\left(\dot{\hat{x}}, \hat{\omega}\right) - k\right) dt = 0. \end{aligned} \tag{50}$$

Then, by subtracting,

$$\int_0^{2\pi} {v_0}^T \left[(g - \tilde{g}) - (\beta(g) - \beta(\tilde{g}))\, J\left(\dot{\hat{x}}, \hat{\omega}\right)\right] dt = 0. \tag{51}$$

Lemma 4.2 also shows that there exists a unique $\bar{z}$ such that there exists a $\bar{z}$ such that

$$\hat{\omega}\dot{\bar{z}} = dX(\hat{x}, \hat{x}_{\hat{\omega}}; \bar{z}, \bar{z}_{\hat{\omega}}) + \left[(g - \tilde{g}) - (\beta(g) - \beta(\tilde{g}))\, J\left(\dot{\hat{x}}, \hat{\omega}\right)\right]. \tag{52}$$

But from (29), $z(g) - z(\tilde{g})$ also satisfies (52), so that $\bar{z} = z(g) - z(\tilde{g})$ and from (31)

$$\begin{aligned} |z(g) - z(\tilde{g})| &\le M\left|(g - \tilde{g}) - (\beta(g) - \beta(\tilde{g}))\, J\left(\dot{\hat{x}}, \hat{\omega}\right)\right|, \\ &\le \lambda_1\,|g - \tilde{g}|\,. \end{aligned} \tag{53}$$

Then (46) follows from (48) through (53). QED

The specific forms for $E_1(\delta)$ and $E_2(\delta)$ are given in Appendix A.2 as equations (201) and (204), respectively, as well as the selection of F_1 and F_2 as (202) and (205) respectively. As functions of the other parameters $E_1(\delta)$ and $E_2(\delta)$ depend linearly on $\mathcal{K}$ and $\mathcal{B}$, but non-linearly on $\lambda_0,\ \lambda_1$, and λ_2 and thus non-linearly on M.

6. Main Approximation Theorem

In the main theorem the constants $F_1,\ F_2$ are those from Lemma 4.2.

Theorem 6.1. *If (a) $(\hat{\omega}, \hat{x})$ is non-critical with respect to (6) in the sense of Definition 3.4, (b) δ is selected so that*

$$\delta \le \min\{1/F_1,\ 1/2F_2,\ \hat{\omega}/4\lambda_0\} \tag{54}$$

and (c) $r \le \delta$, then there exists and exact frequency, ω^*, and solution, x^*, of (6) such that

$$\begin{aligned} |x^* - \hat{x}| &\le 4\lambda_1\delta, \\ |\omega^* - \hat{\omega}| &\le 2\lambda_0\delta, \end{aligned} \tag{55}$$

where λ_0, λ_1 are defined in (40) and δ is defined in (41).

Proof: Let β and z be defined as in (24). By substituting (24) into (6) we have (25). Associated with (25) we consider (29). We then define the set $\mathcal{N}_\delta$ in (41) and consider the map $S : \mathcal{N}_\delta \to \mathcal{P}$ defined in (45). From (200), (201), and (202) we have $|S(g)| \le F_1\delta^2$ for $g \in \mathcal{N}_\delta$. Furthermore, from (203), (204), and (205) we have, for $g,\ \tilde{g} \in \mathcal{N}_\delta$, that $|S(g) - S(\tilde{g})| \le F_2\delta\,|g - \tilde{g}|$. Now, if we select δ as in (54) then $F_1\delta^2 \le \delta$ and $F_2\delta \le 1/2$, S maps $\mathcal{N}_\delta$ to itself and is a contraction. The last inequality that δ satisfies in (54) assures that $\beta(g)$ satisfies (28) by way of Lemma 4.3, provided r satisfies $r \le \delta$. Therefore, S has a fixed point $g^* \in \mathcal{N}_\delta$. This implies that there exists a unique (β^*, z^*), z^* is 2π-periodic, satisfying (25). Then, from (24), there exists a unique (ω^*, x^*), x^* is 2π-periodic, satisfying (6). From (24), with $r \le \delta$,

$$\begin{aligned} |\omega^* - \hat{\omega}| \le |\beta^*| \le \lambda_0\left(|g^*| + r\right) \le 2\lambda_0\delta, \\ |x^* - \hat{x}| \le \left|\frac{\hat{\omega}}{\hat{\omega} + \beta^*}\right| |z^*| \le 2\lambda_1\left(|g^*| + r\right) \le 4\lambda_1\delta. \end{aligned} \tag{56}$$

QED

We need to introduce a note here on the relationship between r and δ. In practice the process of determining them is iterative. We start by determining an approximate solution and the residual r. We then compute all of the parameters that involve the approximate solution and compute δ from (54). We then compare r against δ. If $r \le \delta$ we are finished, otherwise we have to return and recompute another approximate solution with possibly smaller residual r and iterate the process. The author is not familiar with any result that guarantees that at some point $r \le \delta$, although he suspects that this will eventually happen in most practical problems.

7. **Approximating a Solution and Frequency**

An approximate solution and frequency for (6) can be developed by assuming a finite trigonometric polynomial of the form

$$\hat{x}_m = a_2\cos t + \sum_{n=2}^{m}\left[a_{2n}\cos nt + a_{2n-1}\sin nt\right] \tag{57}$$

where the $\sin t$ term has been dropped so that we can estimate $a_1 = \hat{\omega}$, the frequency. Note that we have centered the approximate solution about the origin, since we assumed $X(0,0) = 0$. If we set $\bar{\mathbf{a}} = (a_1,\ a_2, \cdots,\ a_{2m})$, and

$$E_m(t, \bar{\mathbf{a}}) = a_1\dot{\hat{x}}_m(t) - X\left(\hat{x}_m(t),\ \hat{x}_m(t - a_1)\right) \tag{58}$$

then for a sufficiently fine mesh, specified by $\{t_i : i = 1,\ 2, \cdots,\ 2N\}$, in $[0,\ 2\pi]$, where

$$t_i = \frac{2i-1}{2N}\pi, \tag{59}$$

the determining equations for $\bar{\mathbf{a}}$ can be written as (see Urabe and Reiter [33])

$$\begin{aligned}
F_1(\bar{\mathbf{a}}) &= \frac{1}{N}\sum_{i=1}^{2N} E_m\left(t_i, \bar{\mathbf{a}}\right)\sin t_i = 0 \\
F_2(\bar{\mathbf{a}}) &= \frac{1}{N}\sum_{i=1}^{2N} E_m\left(t_i, \bar{\mathbf{a}}\right)\cos t_i = 0 \\
F_{2n-1}(\bar{\mathbf{a}}) &= \frac{1}{N}\sum_{i=1}^{2N} E_m\left(t_i, \bar{\mathbf{a}}\right)\sin nt_i = 0 \\
F_{2n}(\bar{\mathbf{a}}) &= \frac{1}{N}\sum_{i=1}^{2N} E_m\left(t_i, \bar{\mathbf{a}}\right)\cos nt_i = 0
\end{aligned} \tag{60}$$

for $n = 2, \cdots,\ m$.

These equations give $2m$ equations in $2m$ unknowns. Standard numerical solvers, using, for example, Newton's method, for nonlinear equations can be used to solve for $\bar{\mathbf{a}}$. The number of harmonics, m, and the quadrature index, N, can be selected independently.

8. **Floquet Theory for DDEs**

The analysis of the stability of an approximate periodic solution for (1) usually involves the following considerations. If $\widehat{x}(t),\ \widehat{x} \in \mathbb{R}^n$ is an approximate periodic solution of (1) of period 2π, and $\widehat{\omega}$ an approximate frequency, then the linear variational equation about $\widehat{x}(t)$ can be written

$$\dot{z}(t) = \widehat{A}(t)z(t) + \widehat{B}(t)z(t-\widehat{\omega}), \tag{61}$$

where $\widehat{A}(t)$ and $\widehat{B}(t)$ were defined previously and are periodic, with period 2π.

We now define the period map $U : C_{\widehat{\omega}} \to C_{\widehat{\omega}}$ with respect to (61) by

$$\left(U\phi\right)(s) = z(s+2\pi) \tag{62}$$

where $z(s)$ is a solution of (61) satisfying $z(s) = \phi(s)$ for $s \in [-\widehat{\omega}, 0]$. In this paper we assume $\widehat{\omega} < 2\pi$. U is then a compact operator on $C_{\widehat{\omega}}$, whose spectrum is at most countable with 0 as the only possible limit point (Halanay [14]).

A Floquet theory for (61) has been developed by Stokes [26] . In particular, if $\sigma\left(U\right)$ represents the spectrum of U, then for each $\lambda \in \sigma\left(U\right)$, $U\phi = \lambda\phi$. That is, the spectrum consists of eigenvalues. Furthermore, the space $C_{\widehat{\omega}}$ can be decomposed as the direct sum of two invariant subspaces

$$C_{\widehat{\omega}} = E(\lambda) \oplus K(\lambda) \tag{63}$$

$E(\lambda)$ is finite dimensional and composed of the eigenvectors with respect to λ. Furthermore, $\sigma\left(U|_K\right) = \sigma\left(U\right) - \{\lambda\}$. If $\{\psi_i\}, i = 1, \cdots, d$ is a basis for $E(\lambda)$ and we let Ψ be the matrix with columns ψ_j for $j = 1, \cdots, d$, then there is a matrix $G(\lambda)$ such that

$$U\Psi = \Psi G(\lambda) \tag{64}$$

Thus we can think of $C_{\widehat{\omega}}$ as being a countable direct sum of the invariant subspaces $E(\lambda_i)$ plus a possible remainder subspace, R. That is

$$C_{\widehat{\omega}} = E(\lambda_1) \oplus E(\lambda_2) \oplus \cdots \oplus R \tag{65}$$

where R is a "remainder" set in which any solution of (61) with initial condition in R decays faster than any exponential.

For each of the $E(\lambda_i)$ there is a basis set Ψ_i, and a matrix $G(\lambda_i)$. If we define an at most countable basis set $\{\Psi_i\}$, $i = 1, 2, \cdots$, then we can think about U operating on $\bigoplus_{i=1}^{\infty} E(\lambda_i)$ as being represented by an infinite matrix G_∞. This matrix is referred to as the monodromy matrix. Its eigenvalues are called the Floquet or characteristic multipliers. The periodic solution $\widehat{x}(t)$ of (1) is stable if all of the eigenvalues of U are within the unit circle and unstable if there is at least one with a positive real part. We note that if $\widehat{x}(t)$ is an exact periodic solution of (1) then one of the characteristic multipliers is exactly one.

9. **Estimating Characteristic Multipliers**

In this section we assume that the variational equation with respect to the approximate solution, $\hat{x}(t)$, can be written in the form

$$\dot{z}(t) = \hat{A}(t)z(t) + \hat{B}(t)z(t - \hat{\omega}) \tag{66}$$

where $\hat{A}(t) = \hat{A}(t + 2\pi), \hat{B}(t) = \hat{B}(t + 2\pi)$ and we have reintroduced t to make the operator definitions more transparent. Let $Z(t, s)$ be the solution of (66) such that $Z(s, s) = I_n, Z(t, s) = 0$ for $t < s$ where I_n is the n^2 identity matrix on $\mathbf{C}^n$. The solution $Z(t, s)$ is sometimes referred to as the "Fundamental Solution". Using the variation of constants formula for (66), Halanay [14] shows that the solution of (66) for the initial function $\phi \in C_{\hat{\omega}}$ is given by

$$z(t) = Z(t, 0)\phi(0) + \int_{-\hat{\omega}}^{0} Z(t, \alpha + \hat{\omega})\hat{B}(\alpha + \hat{\omega})\phi(\alpha)\, d\alpha \tag{67}$$

Define the operator

$$(U\phi)(s) = z(s + 2\pi) \tag{68}$$

where $\phi \in C_{\hat{\omega}}$, $s \in [-\hat{\omega},\ 0]$. If there is a non-trivial solution $z(t)$ of (66) such that $z(t+2\pi) = \rho z(t)$ then ρ is a characteristic multiplier of (66). If we combine (67) with (68) and note that $z(\alpha) = \phi$ for $\alpha \in [-\hat{\omega}, 0]$, then characteristic multipliers are the eigenvalues of

$$(U\phi)(s) = Z(s + 2\pi, 0)\phi(0) + \int_{-\hat{\omega}}^{0} Z(s + 2\pi, \alpha + \hat{\omega})\hat{B}(\alpha + \hat{\omega})\phi(\alpha)\, d\alpha \tag{69}$$

where $\phi \in C_0$. Halanay [14] shows that we can restrict $s \in [-\hat{\omega}, 0]$. This operator is sometimes referred to as the *Monodromy Operator*.

9.1. **Approximating the Fundamental Solution by Spectral Collocation.**

In this section we will use spectral methods to compute the fundamental solution of the linear homogeneous delay differential equation (66). These methods are well known for collocating solutions to partial differential equations and boundary value problems. See, for example, Gottlieb [11] and Gottlieb and Turkel [12]. They are not as well known in delay differential equations. In this section we use a spectral method suggested by Bueler [3] and Trefethen [31]. The method has been reported earlier in Gilsinn and Potra [10].

The computation of the fundamental matrix used in the monodromy operator (69) requires the computation of a solution $z(t)$ of (66) on some interval $[a, b]$. This will be done in a stepwise manner. We first find a positive integer q such that $a + q\widehat{\omega} \geq b$. Then we solve, at the first step, $t \in [a,\ a + \widehat{\omega}]$,

$$\dot{z}_1(t) = \widehat{A}(t)z_1(t) + \widehat{B}(t)z_1(t - \widehat{\omega}), \tag{70}$$

where $z_1(t - \widehat{\omega}) = \phi(s)$ for some function $\phi \in C_{\widehat{\omega}}(a)$ and $s = t - \widehat{\omega}$. Thus the initial problem becomes an ordinary differential equation. Then, on $[a + \widehat{\omega}, a + 2\widehat{\omega}]$ we solve

$$\dot{z}_2(t) = \widehat{A}(t)z_2(t) + \widehat{B}(t)z_2(t - \widehat{\omega}), \tag{71}$$

where $z_2(a+\widehat{\omega}) = z_1(a+\widehat{\omega})$, $z_2(t-\widehat{\omega}) = z_1(s)$ for $s \in [a, a+\widehat{\omega}], s = t-\widehat{\omega}$. Again we solve (5) as an ordinary differential equation. The process is continued so that on $[a+(i-1)\widehat{\omega}, a+i\widehat{\omega}]$, for $i = 1, 2, \cdots, q$,

$$\dot{z}_i(t) = \widehat{A}(t)z_i(t) + \widehat{B}(t)z_i(t - \widehat{\omega}), \tag{72}$$

with $z_i(a + (i-1)\widehat{\omega}) = z_{i-1}(a + (i-1)\widehat{\omega})$. We then define $z(t)$ on $[a, b]$ as the concatenation of $z_i(t)$ for $t \in [a + (i-1)\widehat{\omega}, a + i\widehat{\omega}]$ and $i = 1, 2, \cdots, q$.

Since we wish to use a Chebyshev collocation method, we will shift each interval $[a+(i-1)\widehat{\omega}, a+i\widehat{\omega}]$ to the interval $[-1,1]$. For $t \in [a+(i-1)\widehat{\omega}, a+i\widehat{\omega}]$, for $i = 1,2,\cdots,q$, we have $z \in [-1,1]$ provided

$$z = \frac{2}{\widehat{\omega}}t - \frac{(2a+(2i-1)\widehat{\omega})}{\widehat{\omega}}. \tag{73}$$

For $z \in [-1,1]$ we have $t \in [a+(i-1)\widehat{\omega}, a+i\widehat{\omega}]$ provided

$$t = \frac{\widehat{\omega}}{2}z + \frac{(2a+(2i-1)\widehat{\omega})}{2}. \tag{74}$$

We note that the point $t \in [a+(i-1)\widehat{\omega}, a+i\widehat{\omega}]$ and $t-\widehat{\omega} \in [a+(i-2)\widehat{\omega}, a+(i-1)\widehat{\omega}]$ are translated to the same $z \in [-1,1]$. This is clear from

$$\frac{2}{\widehat{\omega}}(t-\widehat{\omega}) - \frac{(2a+(2i-3)\widehat{\omega}}{\widehat{\omega}} = \frac{2}{\widehat{\omega}}t - \frac{(2a+(2i-1)\widehat{\omega}}{\widehat{\omega}} \tag{75}$$

Therefore we can shift the iterated delay problems

$$\dot{z}_i(t) = \widehat{A}(t)z_i(t) + \widehat{B}(t)z_i(t-\widehat{\omega}), \tag{76}$$

for $t \in [a+(i-1)\widehat{\omega}, a+i\widehat{\omega}]$ and $i = 1,2,\cdots,q$, into iterated ordinary differential equations

$$u_i'(z) = \frac{\widehat{\omega}}{2}\widetilde{A}_i(z)u_i(z) + \frac{\widehat{\omega}}{2}\widetilde{B}_i(z)u_{i-1}(z) \tag{77}$$

where, for $t \in [a+(i-1)\widehat{\omega}, a+i\widehat{\omega}]$ and associated $z \in [-1,\ 1]$,

$$\begin{aligned} u_i(-1) &= u_{i-1}(1) \\ u_i(z) &= z_i(t) \\ \widetilde{A}_i(z) &= \widehat{A}(t) \\ \widetilde{B}_i(z) &= \widehat{B}(t) \\ u_{i-1}(z) &= z_i(t-\widehat{\omega}) \end{aligned} \tag{78}$$

The initial function is

$$u_0(z) = z_1(t-\widehat{\omega}) = \phi(t-\widehat{\omega}) \tag{79}$$

for $t-\widehat{\omega} \in [a-\widehat{\omega}, a]$.

We can now approximate the fundamental solution for (66) on $[a,b]$ by first solving the iterated differential equations (76) subject to

$$\begin{aligned} u_i(-1) &= u_{i-1}(1) \\ u_0(z) &= 0, \quad z \in [-1,1] \\ u_1(-1) &= I_n \end{aligned} \tag{80}$$

where I_n is the $n \times n$ identity matrix. We follow the spectral method given in Bueler [3] in that the fundamental solution is solved for in n passes of the iteration process with $u_1(-1) = e_j$, where $e_j = (0,\cdots,1,\cdots,0)^T$ with 1 in the jth element, $j = 1,2,\cdots,n$.

To begin the solution process we take, for some positive integer N, the Chebyshev points

$$\eta_k = \cos\left(\frac{k\pi}{N}\right) \tag{81}$$

on $[-1,1]$, for $k = 0,1,\cdots,N$. The benefits of using these points has been discussed by Salzer [24]. The Lagrange interpolation polynomials at these points are given by

$$l_j(z) = \prod_{\substack{k=0 \\ k\neq j}}^{N} \left(\frac{z-\eta_k}{\eta_j-\eta_k}\right). \tag{82}$$

We have $l_j(\eta_k) = \delta_{jk}$. Then on $[-1,1]$ we set

$$u_i(z) = \sum_{j=0}^{N} u_i(\eta_j)\, l_j(z) \tag{83}$$

We also need to form

$$u_i'(z) = \sum_{j=0}^{N} u_i\left(\eta_j\right) l_j'(z) \tag{84}$$

At the Chebyshev points we will designate

$$D_{kj} = l_j'(\eta_k) \tag{85}$$

The values for these derivatives are given in Gottlieb and Turkel [12] or Trefethen [31] but we state the values for D here for completeness. The derivations are given in Section 15, Appendix 1.

$$\begin{aligned} D_{00} &= \frac{2N^2+1}{6} \\ D_{NN} &= -D_{00} \\ D_{jj} &= \frac{-\eta_j}{2(1-\eta_j^2)}, j = 1, 2, \cdots, N-1 \\ D_{ij} &= \frac{c_i(-1)^{i+j}}{c_j(\eta_i - \eta_j)} \end{aligned} \tag{86}$$

for $i \neq j, i, j = 0, \cdots, N$ where

$$c_i = \begin{cases} 2, & i = 0 \text{ or } N; \\ 1, & \text{otherwise.} \end{cases} \tag{87}$$

For notation, let

$$\begin{aligned} u_i(z) &= \left(u_{i1}, \cdots, u_{in}\right)^T, \\ \widetilde{A}_i(z) &= \left[\widetilde{A}_{pq}^{(i)}(z)\right]_{p,q=1,\cdots,n} \\ \widetilde{B}_i(z) &= \left[\widetilde{B}_{pq}^{(i)}(z)\right]_{p,q=1,\cdots,n} \end{aligned} \tag{88}$$

We then write the collocation polynomial of $u_{ir}, r = 1, \cdots, n$, as

$$u_{ir}(z) = \sum_{k=0}^{N} w_{rk}^{(i)} l_k(z) \tag{89}$$

at the Chebyshev points (81) to get

$$\begin{aligned} u_{ir}\left(\eta_j\right) &= w_{rj}^{(i)} \\ u_{ir}'\left(\eta_j\right) &= \sum_{k=0}^{N} w_{rk}^{(i)} D_{jk} \\ u_{i-1,r} &= w_{rj}^{(i-1)} \end{aligned} \tag{90}$$

The initial conditions for the iterated differential equations are

$$u_{ir}\left(\eta_N\right) = u_{i-1,r}\left(\eta_0\right), \tag{91}$$

or

$$w_{rN}^{(i)} = w_{r0}^{(i-1)} \tag{92}$$

for $r = 1, \cdots, n$.

The discretized differential equations are then given by

$$\left(\sum_{k=0}^{N} w_{rk}^{(i)} D_{jk}\right)_{r=1,n} = \frac{\widehat{\omega}}{2}\left[\widetilde{A}_{rp}^{(i)}(z)\right]_{r,p=1,n}\left(w_{rj}^{(i)}\right)_{r=1,n} + \frac{\widehat{\omega}}{2}\left[\widetilde{B}_{rp}^{(i)}(z)\right]_{r,p=1,n}\left(w_{rj}^{(i-1)}\right)_{r=1,n} \tag{93}$$

for $j = 0, 1, \cdots, N-1$. These provide nN equations but $n(N-1)$ unknowns. The other n equations come from the initial conditions. We define the following vectors

$$\begin{aligned} w_i &= \left(w_{10}^{(i)} \cdots w_{1N}^{(i)} w_{20}^{(i)} \cdots w_{2N}^{(i)} \cdots w_{n0}^{(i)} \cdots w_{nN}^{(i)}\right)^T \\ w_{i-1} &= \left(w_{10}^{(i-1)} \cdots w_{1N}^{(i-1)} w_{20}^{(i-1)} \cdots w_{2N}^{(i-1)} \cdots w_{n0}^{(i-1)} \cdots w_{nN}^{(i-1)}\right)^T \end{aligned} \tag{94}$$

Then we can write the iterated differential equation as

$$\widetilde{D} w_i = \frac{\widehat{\omega}}{2} \widetilde{A}_i w_i + \frac{\widehat{\omega}}{2} \widetilde{B}_i w_{i-1} \tag{95}$$

where $\widetilde{D} = D \otimes I_n$, the Kronecker product, and each D is given by

$$D = \begin{bmatrix} D_{00} & \cdots & D_{0N} \\ \vdots & \vdots & \vdots \\ D_{N-1,0} & \cdots & D_{N-1,N} \\ 0 & \cdots & 1 \end{bmatrix} \tag{96}$$

The unit in the lower right introduces the initial condition, $w_{rN}^{(i)}, r = 1, \cdots, n$, equation. Thus $\widetilde{D}$ is formed by n blocks of D down the diagonal.

The matrix $\widetilde{A}_i$ is given by

(97)

$$\widetilde{A}_i = \begin{bmatrix} \widetilde{A}_{11}^{(i)}(\eta_0) & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \widetilde{A}_{1n}^{(i)}(\eta_0) & 0 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \ddots & 0 & \cdots & 0 \\ 0 & 0 & \widetilde{A}_{11}^{(i)}(\eta_{N-1}) & 0 & \cdots & 0 & \cdots & 0 & 0 & \widetilde{A}_{1n}^{(i)}(\eta_{N-1}) & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \widetilde{A}_{n1}^{(i)}(\eta_0) & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \widetilde{A}_{nn}^{(i)}(\eta_0) & 0 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \ddots & 0 & \cdots & 0 \\ 0 & 0 & \widetilde{A}_{n1}^{(i)}(\eta_{N-1}) & 0 & \cdots & 0 & \cdots & 0 & 0 & \widetilde{A}_{nn}^{(i)}(\eta_{N-1}) & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$\widetilde{B}_i$ is structured in a similar manner except every $(N+1)$th row includes an element $2/\widehat{\omega}$ to take care of the initial condition. Thus

(98)

$$\widetilde{B}_i = \begin{bmatrix} \widetilde{B}_{11}^{(i)}(\eta_0) & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \widetilde{B}_{1n}^{(i)}(\eta_0) & 0 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \ddots & 0 & \cdots & 0 \\ 0 & 0 & \widetilde{B}_{11}^{(i)}(\eta_{N-1}) & 0 & \cdots & 0 & \cdots & 0 & 0 & \widetilde{B}_{1n}^{(i)}(\eta_{N-1}) & 0 & \cdots & 0 \\ \frac{2}{\widehat{\omega}} & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \widetilde{B}_{n1}^{(i)}(\eta_0) & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \widetilde{B}_{nn}^{(i)}(\eta_0) & 0 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \ddots & 0 & \cdots & 0 \\ 0 & 0 & \widetilde{B}_{n1}^{(i)}(\eta_{N-1}) & 0 & \cdots & 0 & \cdots & 0 & 0 & \widetilde{B}_{nn}^{(i)}(\eta_{N-1}) & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \frac{2}{\widehat{\omega}} & 0 & 0 & \cdots & 0 \end{bmatrix}$$

The linear equation (95) can be solved for w_i by setting

$$M_i = \left(\widetilde{D} - \frac{\widehat{\omega}}{2} \widetilde{A}_i\right)^{-1} \frac{\widehat{\omega}}{2} \widetilde{B}_i \tag{99}$$

and

$$w_i = M_i w_{i-1} \tag{100}$$

for $i = 2, 3, \cdots, q$.

To solve for w_1 for the fundamental solution we need to solve

$$u_1'(z) = \frac{\widehat{\omega}}{2}\widetilde{A}_1(z)u_1(z) \tag{101}$$

for $z \in (-1, 1]$ and

$$u_1(-1) = I_n \tag{102}$$

That is, we solve n problems at each iteration, one for each of the initial conditions e_i, where e_i is the standard basis vector with a unit in the ith element and zero elsewhere. For the moment we set the initial vector as

$$w_0 = (0 \cdots u_{01} 0 \cdots u_{02} 0 \cdots u_{0n})^T \tag{103}$$

where $u_{0r}, r = 1, \cdots, n$, is placed in each of the $(N+1)$th elements and zero elsewhere. Then from the previous construction of $\widetilde{D}$ and $\widetilde{A}_1$ we have

$$w_1 = \left(\widetilde{D} - \frac{\widehat{\omega}}{2}\widetilde{A}_1\right)^{-1} w_0 \tag{104}$$

Given that we have computed

$$u_{ir}(z) = \sum_{k=0}^{N} w_{rk}^{(i)} l_k(z) \tag{105}$$

on $[-1, 1]$ for $r = 1, \cdots, n$ we can compute the result for $t \in [a + (i-1)h, a + ih]$ by setting

$$z_{ir}(t) = u_{ir}(z) \tag{106}$$

for $r = 1, \cdots, n$, where

$$z = \frac{2}{\widehat{\omega}}t - \frac{(2a + (2i-1)\widehat{\omega})}{\widehat{\omega}} \tag{107}$$

or

$$z_{ir}(t) = \sum_{k=0}^{N} w_{rk}^{(i)} l_k(\frac{2}{\widehat{\omega}}t - \frac{(2a + (2i-1)\widehat{\omega})}{\widehat{\omega}}) \tag{108}$$

The initial condition is

$$u_{ir}(\eta_N) = u_{i-1,r}(\eta_0). \tag{109}$$

But on $[a + (i-1)\widehat{\omega}, a + i\widehat{\omega}]$, $z_N = -1$ corresponding to $t = a + (i-1)\widehat{\omega}$ and on $[a + (i-2)\widehat{\omega}, a + (i-1)\widehat{\omega}]$, $z_0 = 1$ corresponding to $t = a + (i-1)\widehat{\omega}$, so that

$$z_{ir}(a + (i-1)\widehat{\omega}) = z_{i-1,r}(a + (i-1)\widehat{\omega}) \tag{110}$$

9.2. **Estimating Monodromy Operator Eigenvalues.** To approximate the monodromy operator (69) we will require a quadrature rule that satisfies

$$\sum_{k=1}^{P+1} v_k f(s_k) \to \int_{-\hat{\omega}}^{0} f(s)\, ds \tag{111}$$

as $P \to \infty$ for each continuous function $f \in C_{\hat{\omega}}$. The rule is satisfied if

$$\sum_{k=1}^{P+1} |v_k| \leq Q, \tag{112}$$

for some $Q > 0$ and $P = 1,\ 2, \cdots$. This is satisfied by, for example, Trapezoidal or Simpson rules.

Let $-\hat{\omega} = s_1 < s_2 < \cdots < s_{P+1} = 0$, and define

$$(U\phi)(s) = Z(s + 2\pi, 0)\phi(0) + \sum_{k=1}^{P+1} v_k Z(s + 2\pi, s_k + \hat{\omega}) B(s_k + \hat{\omega})\phi(s_k) \tag{113}$$

for $\phi \in C_{\hat{\omega}}$.

Then, for each $s_i \in [-\hat{\omega}, 0]$,

$$(U\phi)(s_i) = Z(s_i + 2\pi, 0)\phi(0) + \sum_{j=1}^{P+1} w_j Z(s_i + 2\pi, s_j + \hat{\omega})B(s_j + \hat{\omega})\phi(s_j) \tag{114}$$

Since $s_{P+1} = 0$, (114) can be rewritten as

$$\begin{aligned}(U\phi)(s_i) &= \sum_{j=1}^{P} w_j Z(s_i + 2\pi, s_j + \hat{\omega})B(s_j + \hat{\omega})\phi(s_j) \\ &+ \left(Z(s_i + 2\pi, 0) + w_{P+1}Z(s_i + 2\pi, \hat{\omega})B(\hat{\omega})\right)\phi(s_{P+1}),\end{aligned} \tag{115}$$

where $Z(s, \alpha)$ is the fundamental matrix of (66). Equation (115) can be put in matrix form

$$\begin{pmatrix}(U\phi)(s_1)\\ \vdots \\ (U\phi)(s_i) \\ \vdots \\ (U\phi)(s_{P+1})\end{pmatrix} = \begin{bmatrix} U_{1,1} & \cdots & U_{1,j} & \cdots & U_{1,P+1} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ U_{i,1} & \cdots & U_{i,j} & \cdots & U_{i,P+1} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ U_{P+1,1} & \cdots & U_{P+1,j} & \cdots & U_{P+1,P+1}\end{bmatrix}, \tag{116}$$

where the block elements for $i = 1, \cdots, P+1, j = 1, \cdots, P$ are $U_{i,j} = w_j Z(s_i + 2\pi, s_j + \hat{\omega})B(s_j + \hat{\omega})$. The block elements in the last column of the matrix are given by $U_{i,P+1} = Z(s_i+2\pi, 0)+w_{P+1}Z(s_i+2\pi, \hat{\omega})B(\hat{\omega})$ for $i = 1, \cdots, P+1$. The relevant eigenvalue problem becomes

$$\begin{bmatrix} U_{1,1} & \cdots & U_{1,j} & \cdots & U_{1,P+1} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ U_{i,1} & \cdots & U_{i,j} & \cdots & U_{i,P+1} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ U_{P+1,1} & \cdots & U_{P+1,j} & \cdots & U_{P+1,P+1}\end{bmatrix} = \lambda \begin{pmatrix}(\phi)(s_1)\\ \vdots \\ (\phi)(s_i) \\ \vdots \\ (\phi)(s_{P+1})\end{pmatrix} \tag{117}$$

10. **Determining Solutions of the Adjoint Equation Associated with Multipliers of the Variational Equation**

In order to estimate α in (33), let $t \in [0, 2\pi]$ and ψ be the initial function defined on $[2\pi, 2\pi + \widehat{\omega}]$. The formal adjoint equation from (13) is given by

$$\dot{v_0}(t) = -v_0(t)\widehat{A}(t) - v_0(t + \widehat{\omega})\widehat{B}(t + \widehat{\omega}), \tag{118}$$

where $v_0(t)$ is a row vector. Ordinarily solving the adjoint equation would require a "backward" integration. However, Halanay [14] showed that the solution of the formal adjoint (118) on $[0, 2\pi]$ is given in row vector form by

$$v_0(t) = \psi(2\pi)Z(2\pi, t) + \int_{2\pi}^{2\pi+\widehat{\omega}} \psi(\alpha)\widehat{B}(\alpha)Z(\alpha - \widehat{\omega}, t)\, d\alpha. \tag{119}$$

The significance of this representation is that only a "forward" integration is required to solve for the fundamental solution Z. Let $\widetilde{\phi}(s)$ be a continuous row vector function defined on $[-\widehat{\omega}, 0]$. Halanay [14] then defined the operator

$$\left(\widetilde{U}\widetilde{\phi}\right)(s) = \widetilde{\phi}(-\widehat{\omega})Z(2\pi, s + \widehat{\omega}) + \int_{-\widehat{\omega}}^{0} \widetilde{\phi}(\alpha)\widehat{B}(\alpha + \widehat{\omega})Z(2\pi + \alpha, s + \widehat{\omega})\, d\alpha. \tag{120}$$

Note the relationship to the monodromy operator (**??**). He also gave an associated operator $\widetilde{V}$, defined on $[2\pi, 2\pi + \widehat{\omega}]$, as

$$\left(\widetilde{V}\psi\right)(s) = y(s-2\pi, \psi) = \psi(2\pi)Z(2\pi, s-2\pi) + \int_{2\pi}^{2\pi+\widehat{\omega}} \psi(\alpha)\widehat{B}(\alpha)Z(\alpha-\widehat{\omega}, s-2\pi)\, d\alpha. \tag{121}$$

Again, note the relationship to (119). He further showed that an eigenvalue ρ_0 of $\widetilde{V}$ is associated with a $1/\rho_0$ multiplier of the formal adjoint equation (118), the eigenvalues of

$U, \widetilde{U}, \widetilde{V}$ are all the same, and the eigenvectors of $\widetilde{U}, \widetilde{V}$ are related by $\widetilde{\phi}(s) = \psi(s + 2\pi + \widehat{\omega})$, $s \in [\widehat{\omega}, 0]$. Although $\widetilde{V}$ is the operator associated with (119), the fact that the eigenvalues and eigenvectors of $\widetilde{V}$ and $\widetilde{U}$ are the same allows algorithms developed for the characteristic multipliers in Section 9 to be easily modified to compute the eigenvalues and eigenvectors for (120). In particular we again partition the interval $[-\widehat{\omega}, 0]$ into P equal intervals of length $\Delta = \widehat{\omega}/P$ by

$$-\widehat{\omega} = s_1 < s_2 < \cdots < s_{P+1} = 0 \tag{122}$$

We use a Simpson integration method to write, with the same weights as previously,

$$\begin{aligned}\left(\widetilde{U}\widetilde{\phi}\right)(s) \quad &= \quad \widetilde{\phi}(s_1)\left(Z(2\pi,\ s+\widehat{\omega}) \ + w_1\widehat{B}(s_1+\widehat{\omega})\, Z(2\pi + s_1,\ s+\widehat{\omega})\right) \\ &\quad + \sum_{j=2}^{P+1} \widetilde{\phi}(s_j)\left(w_j\widehat{B}(s_j+\widehat{\omega})\, Z(2\pi + s_j,\ s+\widehat{\omega})\right)\end{aligned} \tag{123}$$

To solve the adjoint equation in row form on $[0, 2\pi]$, we need only compute the eigenvector of $\widetilde{U}$ associated with the characteristic multiplier of U. The eigenvector is then substituted into equation (119). From the previous section there are likely to be two complex conjugate eigenvalues, ρ_0 and $\overline{\rho}_0$, associated with complex conjugate eigenvectors v_0 and $\overline{v}_0$ of (119), so by linearity of (118) the real part forms a discretized solution of (118) and thus a single real independent solution. We will continue to call the real part of this eigenvector $\widetilde{\phi}$ so that for $t \in [0,\ 2\pi]$

$$\begin{aligned} v_0(t) \quad &= \quad \widetilde{\phi}(s_1)\left(Z(2\pi,\ t) + w_1\widehat{B}(s_1+\widehat{\omega})\, Z(s_1 + 2\pi,\ t)\right) \\ &\quad + \sum_{j=2}^{P+1} \widetilde{\phi}(s_j)\left(w_j\widehat{B}(s_j+\widehat{\omega})\, Z(s_j + 2\pi,\ t)\right)\end{aligned} \tag{124}$$

The j-th block column of $\widetilde{U}\widetilde{\phi}$ is given by

(125)

$$\left(\tilde{U}\tilde{\phi}\right)(s_j) = \left[\tilde{\phi}(s_1), \cdots, \tilde{\phi}(s_i), \cdots, \tilde{\phi}(s_{P+1})\right] \begin{bmatrix} Z(2\pi, s_j+\widehat{\omega}) + w_1\widehat{B}(s_1+\widehat{\omega})X(s_1+2\pi, s_j+\widehat{\omega}) \\ \vdots \\ w_i\widehat{B}(s_i+\widehat{\omega})Z(s_i+2\pi, s_j+\widehat{\omega}) \\ \vdots \\ w_{P+1}\widehat{B}(s_{N+1}+\widehat{\omega})Z(s_{N+1}+2\pi, s_j+\widehat{\omega}) \end{bmatrix}.$$

The eigenvector $\tilde{\phi}$ of the matrix on the right associated with the multiplier of the variational equation is computed and substituted into equation (124) to give the value of $v_0(t)$ on $[0,\ 2\pi]$.

To compute α we need to estimate (13). Again we use a Simpson rule. We partition $[0,\ 2\pi]$ by equidistant intervals $0 = t_1 < t_2 < \cdots < t_{O+1} = 2\pi$ and set $h = 2\pi/O$, where O is an even integer. Again the weights will be set as $u_1 = u_{O+1} = h/3$, otherwise $u_k = 4h/3$ if k is even, $u_k = 2h/3$ if k is odd. We then set

$$\alpha = \left(\sum_{k=1}^{O+1} u_k v_0(t_k)\, J\left(\dot{\widehat{x}}, \widehat{\omega}\right)(t_k)\right)^{-1} \tag{126}$$

where v_0 is a row vector and J is a column vector.

Using the same partition of $[0,\ 2\pi]$ we can compute $v_0(t)$ at each t_k as

(127)

$$v_0(t_k) = \left[\tilde{\phi}(s_1), \cdots, \tilde{\phi}(s_i), \cdots, \tilde{\phi}(s_{P+1})\right] \begin{bmatrix} Z(2\pi, t_k) + w_1 \widehat{B}(s_1 + 2\pi + \widehat{\omega}) Z(s_1 + 2\pi, t_k) \\ \vdots \\ w_i \widehat{B}(s_i + 2\pi + \widehat{\omega}) Z(s_i + 2\pi, t_k) \\ \vdots \\ w_{P+1} \widehat{B}(s_{P+1} + 2\pi + \widehat{\omega}) Z(s_{P+1} + 2\pi, t_k) \end{bmatrix}.$$

We finally normalize v_0 so that $|v_0|_2 = 1$.

11. Estimating the M Parameter

From Halanay [14] the variation of constants formula for

$$\dot{z}(t) = \hat{A}(t)z(t) + \hat{B}(t)z(t - \hat{\omega}) + f(t), \tag{128}$$

where $t \in [0, 2\pi]$, is given by

$$z(t) = Z(t, 0)\phi(0) + \int_{-\hat{\omega}}^{0} Z(t, \alpha + \hat{\omega})\hat{B}(\alpha + \hat{\omega})z(\alpha)\, d\alpha + \int_0^t Z(t, \alpha)f(\alpha)\, d\alpha. \tag{129}$$

The 2π periodic initial function condition with $s \in [-\hat{\omega}, 0]$ is

$$\begin{aligned} \phi(s) &= Z(s + 2\pi, 0)\phi(0) \\ &\quad + \int_{-\hat{\omega}}^{0} Z(s + 2\pi, \alpha + \hat{\omega})\hat{B}(\alpha + \hat{\omega})\phi(\alpha)\, d\alpha + \int_0^{s+2\pi} Z(s + 2\pi, \alpha)f(\alpha)\, d\alpha. \end{aligned} \tag{130}$$

The first step in computing M involves relating ϕ to f. Let $|\phi| = \sup_{-\hat{\omega} \le s \le 0} |\phi(s)|$ and similarly for $|f|$ on $[0, 2\pi]$. To eliminate $\phi(0)$ from (130), set $s = 0$ in (130) and solve for $\phi(0)$ as

$$\begin{aligned} \phi(0) &= \int_{-\hat{\omega}}^{0} (I - Z(2\pi, 0))^{-1} Z(2\pi, \alpha + \hat{\omega})\hat{B}(\alpha + \hat{\omega})\phi(\alpha)\, d\alpha \\ &\quad + \int_0^{2\pi} (I - Z(2\pi, 0))^{-1} Z(2\pi, \alpha)f(\alpha)\, d\alpha. \end{aligned} \tag{131}$$

Substitute (131) into (130) and combine terms as

$$\begin{aligned} \phi(s) &= \int_{-\hat{\omega}}^{0} \left[Z(s + 2\pi, 0)(I - Z(2\pi, 0))^{-1} Z(2\pi, \alpha + \hat{\omega}) \right. \\ &\quad \left. + Z(s + 2\pi, \alpha + \hat{\omega})\right] \hat{B}(\alpha + \hat{\omega})\phi(\alpha)\, d\alpha \\ &\quad + \int_0^{2\pi} \left[Z(s + 2\pi, 0)(I - Z(2\pi, 0))^{-1} Z(2\pi, \alpha) + Z(s + 2\pi, \alpha)\right] f(\alpha)\, d\alpha. \end{aligned} \tag{132}$$

where $s \in [-\hat{\omega}, 0]$.

Let $-\hat{\omega} = s_1 < s_2 < \cdots < s_{P+1} = 0,\ ds = \frac{\hat{\omega}}{P}$, and $0 = t_1 < t_2 < \cdots < t_{O+1} = 2\pi,\ dt = \frac{2\pi}{O}$. We can discretize (132) by setting

$$\phi(s_i) = \sum_{j=1}^{P+1} H_1(i, j)\phi(s_j) + \sum_{k=1}^{O+1} H_2(i, j)f(t_k), \tag{133}$$

where

$$\begin{aligned} H_1(i, j) &= v_j \Big[Z(s_i + 2\pi, 0)(I - Z(2\pi, 0))^{-1} Z(2\pi, s_j + \hat{\omega}) \\ &\quad + Z(s_i + 2\pi, s_j + \hat{\omega})\Big] \hat{B}(s_j + \hat{\omega}) \\ H_2(i, j) &= u_k \Big[Z(s_i + 2\pi, 0)(I - Z(2\pi, 0))^{-1} Z(2\pi, t_k) \\ &\quad + Z(s_i + 2\pi, t_k)\Big] \end{aligned} \tag{134}$$

In vector matrix form (133) can be written

$$\begin{pmatrix} \phi(s_1) \\ \vdots \\ \phi(s_{P+1}) \end{pmatrix} = H_1 \begin{pmatrix} \phi(s_1) \\ \vdots \\ \phi(s_{P+1}) \end{pmatrix} + H_2 \begin{pmatrix} f(t_1) \\ \vdots \\ f(t_{O+1}) \end{pmatrix} \tag{135}$$

Using a generalized inverse we can solve for the ϕ vector with minimum norm by

$$\begin{pmatrix} \phi(s_1) \\ \vdots \\ \phi(s_{P+1}) \end{pmatrix} = (I - H_1)^+ H_2 \begin{pmatrix} f(t_1) \\ \vdots \\ f(t_{O+1}) \end{pmatrix} \tag{136}$$

In the second step the value of $\phi(0)$, given by equation (131), is substituted into equation (129) and terms combined to give

$$\begin{aligned} z(t) &= \int_{-\hat{\omega}}^{0} \left[Z(t,0)(I - Z(2\pi,0))^{-1} Z(2\pi, \alpha + \hat{\omega}) \right. \\ &\quad \left. + Z(t, \alpha + \hat{\omega}) \right] \hat{B}(\alpha + \hat{\omega}) \phi(\alpha)\, d\alpha \\ &\quad + \int_{0}^{2\pi} \left[Z(t,0)(I - Z(2\pi,0))^{-1} Z(2\pi, \alpha) + Z(t,\alpha) \right] f(\alpha)\, d\alpha. \end{aligned} \tag{137}$$

This can be discretized by setting

$$z(t_k) = \sum_{i=1}^{P+1} H_3(k,i)\phi(s_i) + \sum_{j=1}^{O+1} H_4(k,j) f(t_k), \tag{138}$$

where

$$\begin{aligned} H_3(k,i) &= v_i \Big[Z(t_k, 0)(I - Z(2\pi,0))^{-1} Z(2\pi, s_i + \hat{\omega}) \\ &\quad + Z(t_k, s_i + \hat{\omega}) \Big] \hat{B}(s_i + \hat{\omega}) \\ H_4(k,j) &= u_j \Big[Z(t_k, 0)(I - Z(2\pi,0))^{-1} Z(2\pi, t_j) \\ &\quad + Z(t_k, t_j) \Big] \end{aligned} \tag{139}$$

In vector matrix form (138) can be written

$$\begin{pmatrix} z(t_1) \\ \vdots \\ z(t_{O+1}) \end{pmatrix} = H_3 \begin{pmatrix} \phi(s_1) \\ \vdots \\ \phi(s_{P+1}) \end{pmatrix} + H_4 \begin{pmatrix} f(t_1) \\ \vdots \\ f(t_{O+1}) \end{pmatrix} \tag{140}$$

By substituting (136) into (140) we have

$$\begin{pmatrix} z(t_1) \\ \vdots \\ z(t_{O+1}) \end{pmatrix} = \left[H_3 (I - H_1)^+ H_2 + H_4 \right] \begin{pmatrix} f(t_1) \\ \vdots \\ f(t_{O+1}) \end{pmatrix} \tag{141}$$

Therefore

$$|z| \le M|f|, \tag{142}$$

where $M = \left\| H_3 (I - H_1)^+ H_2 + H_4 \right\|_\infty$.

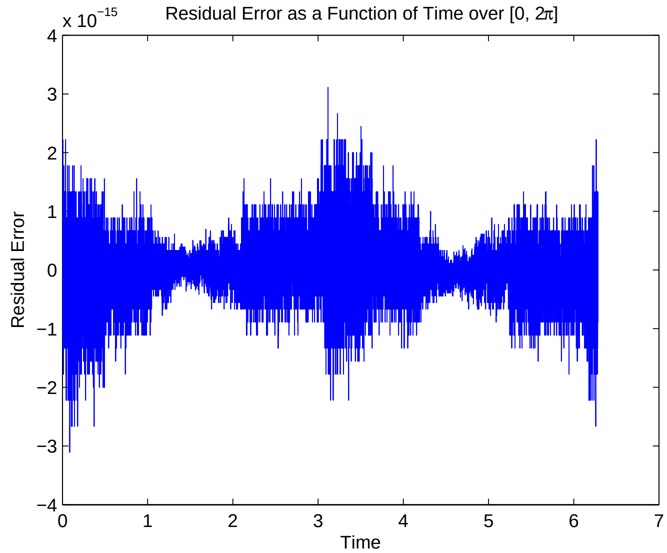

FIGURE 1. Residual Error of Approximate Solution for the Van der Pol Equation.

12. Application to a Van der Pol Equation with Delay

In this section we will apply the main theorem to approximate the limit cycle of the Van der Pol equation with unit delay, given by

$$\ddot{x} + \lambda \left(x(t-1)^2 - 1 \right) \dot{x}(t-1) + x = 0. \tag{143}$$

Since the period of the limit cycle is unknown we introduce an unknown frequency by substituting t/ω for t to obtain

$$\omega^2 \ddot{x} + \omega\lambda \left(x(t-\omega)^2 - 1 \right) \dot{x}(t-\omega) + x = 0, \tag{144}$$

for $t \in [0, 2\pi]$. To compare with an approximation result obtained for ordinary differential equations in Stokes [27], we take $\lambda = 0.1$.

The first step was to estimate an approximate 2π-periodic solution, frequency and residual to (144). By using Galerkin's method described in Section 7 the following approximate solution was obtained

$$\begin{aligned} \hat{x}(t) &= 2.0185 \cos(t) \\ &\quad + 2.5771 \times 10^{-3} \sin(2t) + 2.5655 \times 10^{-2} \cos(2t) \\ &\quad + 1.0667 \times 10^{-4} \sin(3t) - 5.2531 \times 10^{-4} \cos(3t) \\ &\quad - 7.1780 \times 10^{-6} \sin(4t) - 2.2043 \times 10^{-6} \cos(4t), \\ \hat{\omega} &= 1.0012. \end{aligned} \tag{145}$$

where we have displayed only the first few harmonics. This solution was estimated based on 11 harmonics, 40,000 sampled points over $[0, 2\pi]$, and 100 Chebyshev extreme points (81). The residual was estimated by substituting $(\hat{\omega}, \hat{x})$ from equation (145) into equation (144) and finding the maximum of the absolute values of the residuals obtained in the interval $[0, 2\pi]$. The result was $r = 3.1086 \times 10^{-15}$. This residual is significantly better than the one given in Stokes [27]. The distribution of the residuals for the current case is shown in Figure 1. The phase plot of the approximate solution is shown in Figure 2. For $t \in [0, 2\pi]$ we can then immediately estimate $|\hat{x}| \le 2.0436$, $|\dot{\hat{x}}| \le 2.0279$, $|\ddot{\hat{x}}| \le 2.1165$.

In the second step, the values of the constants $\mathcal{B}$ and $\mathcal{K}$ were obtained in a straightforward manner from the variational equation about the approximate frequency and solution given

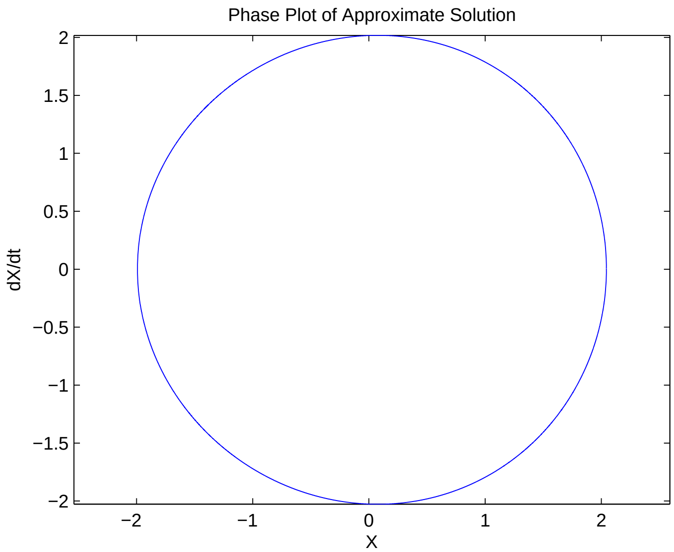

FIGURE 2. Phase Plot of Approximate Solution for the Van der Pol Equation.

by

$$\dot{z}(t) = \hat{A}(t)z(t) + \hat{B}(t)Z(t-\hat{\omega}), \tag{146}$$

where

$$z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}, \quad \hat{A}(t) = \begin{pmatrix} 0 & 1 \\ -1/\hat{\omega}^2 & 0 \end{pmatrix},$$

$$\hat{B}(t) = \begin{pmatrix} 0 & 0 \\ -2(\lambda/\hat{\omega})\hat{x}_1(t-\hat{\omega})\hat{x}_2(t-\hat{\omega}) & (\lambda/\hat{\omega})\left(1-\hat{x}_1(t-\hat{\omega})^2\right) \end{pmatrix}.$$

We use the fact that the natural norm of a matrix, H, associated with a vector norm $|x| = \max_{1\le i\le n}|x_i|$ is $|H| = \max_{1\le i\le n}\sum_{j=1}^{n}|h_{ij}|$. With this definition it is not hard to show that

$$\begin{aligned} |dX(\hat{x};\phi)| &\le \\ &\left|\begin{matrix} 0 & 1 \\ -1/\hat{\omega}^2 - 2(\lambda/\hat{\omega})\hat{x}_1(t-\hat{\omega})\hat{x}_2(t-\hat{\omega}) & (\lambda/\hat{\omega})\left(1-\hat{x}_1(t-\hat{\omega})^2\right) \end{matrix}\right| |\phi|, \\ &\le 2.3776|\phi|. \end{aligned} \tag{147}$$

Therefore, for $\lambda = 0.1$, $\mathcal{B} = 2.3776$. Working conservatively within the domain $D = \{x \in C[0,2\pi] : |x-\hat{x}| \le 1\}$ it is not hard to show that

$$\begin{aligned} &|dX(\hat{x}_{\hat{\omega}} + \psi_1;\phi_{\hat{\omega}}) - dX(\hat{x}_{\hat{\omega}} + \psi_2;\phi_{\hat{\omega}})| \\ &\qquad \le (6\lambda/\hat{\omega})\left(1+|\hat{x}|\right)|\psi_1-\psi_2|\,|\phi|. \end{aligned} \tag{148}$$

Then from (145) and (148) we can estimate $\mathcal{K} = 1.8157$ and, from (23), we can estimate $\left|J(\dot{\hat{x}},\hat{\omega})\right| \le 2.7546$.

Next, we can estimate the characteristic multipliers of the variational equation relative to the function $\hat{x}(t)$. For the quadrature steps in Sections 9 and 10 P and O were taken as 200 and 1200 respectively. These gave mesh widths of about 1/200 on both $[-\hat{\omega}, 0]$ and $[0, 2\pi]$. Using the method of Section 9 we computed two simple conjugate eigenvalues with magnitude 1.0430. All of the other eigenvalues have magnitudes near zero. These are, of course, the eigenvalues of the monodromy operator U. The fundamental matrix Z in (69) is computed using the collocation method of Section 9.1 (See Figure 3). The monodromy operator is formulated as in Section 9. The eigenvalues of the monodromy operator U are plotted in Figure 4. Note that the significant complex conjugate eigenvalues are near the

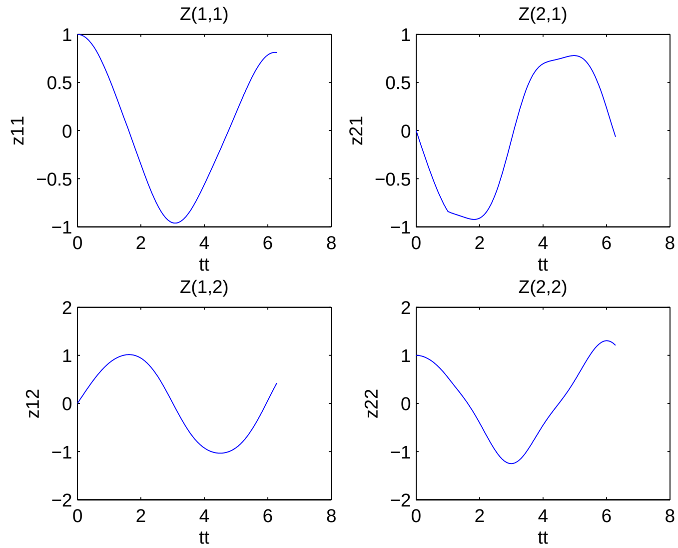

FIGURE 3. Fundamental Matrix for the Variational Equational relative to the Approximate Solution for the Van der Pol Equation.

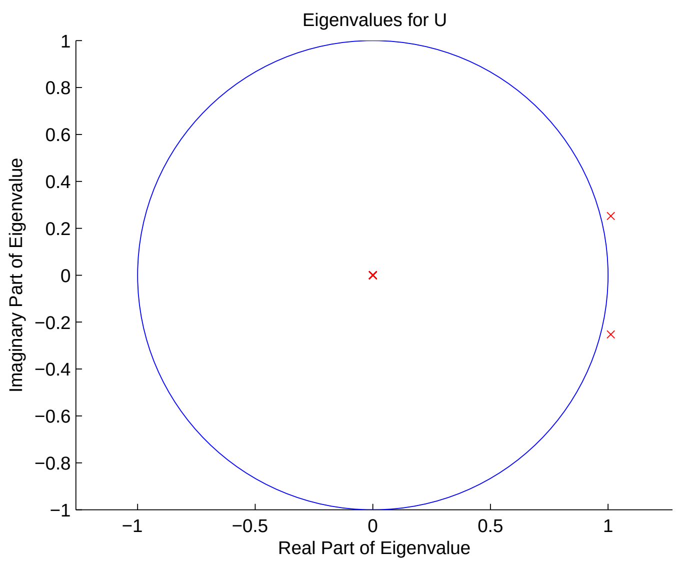

FIGURE 4. Eigenvalues for the Monodromy Operator

unit circle but are not exactly on it. This is due to the fact that (145) is only an approximate solution. The eigenvalues are complex conjugates because the left hand matrix in (117) is real and non-symmetric since the fundamental solution Z is non-symmetric (See Figure 3). We can confirm that the eigenvalues of the operator $\tilde{U}$ are the same as those of U. Graphically this is shown in Figure 5.

In the next step we estimate the parameter α using the methods of Section 10. The solution of the adjoint to the variational equation was computed using equation (127) and the parameter α in (33) was estimated by simple quadrature, with $\Delta = 2\pi/O$ for a sufficiently

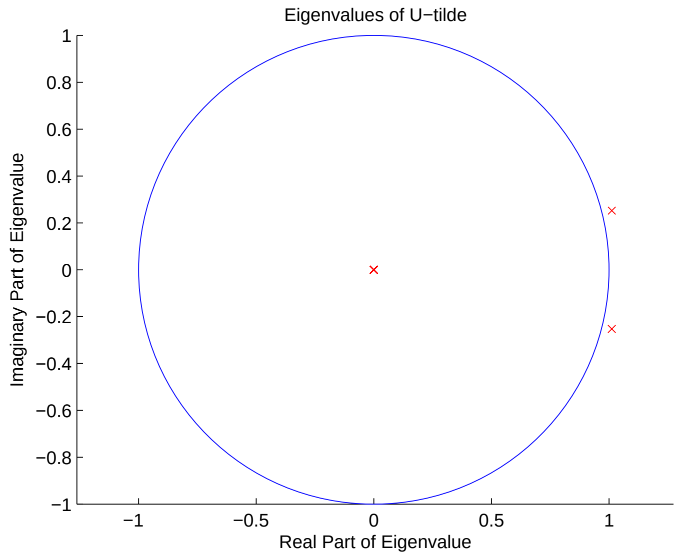

FIGURE 5. Eigenvalues for $\tilde{U}$

large mesh, $0 = t_1 < t_2 < \cdots < t_{O+1} = 2\pi$, as

$$\alpha = \left[\Delta \left|\sum_{i=1}^{O+1} y(t_i) J(\dot{\hat{x}}, \hat{\omega})(t_i)\right|\right]^{-1}. \tag{149}$$

The absolute value of α is estimated as 3.3547.

If we now apply the methods of Section 11, using $\hat{A}(t)$ and $\hat{B}(t)$ defined in equation (146), we can estimate $M = 2.7618 \times 10^2$. These results allow us to estimate λ_0, λ_1 and λ_2 in Lemma 4.3 as $\lambda_0 = 8.4091$, $\lambda_1 = 6.6736 \times 10^3$, and $\lambda_2 = 3.1720 \times 10^4$. Note the magnitude of the parameters.

With the estimates above we can compute $F_1 = 2.5941 \times 10^9$, $F_2 = 1.0798 \times 10^{10}$ from (202) and (205) respectively. Then we compute $\delta = 4.6305 \times 10^{-11}$ from (54). Then $F_1\delta^2 = 5.5623 \times 10^{-12}$ is less than δ and $F_2\delta = 0.5$. Furthermore $r < \delta$. Therefore, the conditions of the main theorem are satisfied and we can conclude from Theorem 6.1 that there exists an exact solution x^* and an exact frequency ω^* of equation (144) such that $|x^* - \hat{x}| \leq 1.2361 \times 10^{-6}$ and $|\omega^* - \hat{\omega}| \leq 7.7877 \times 10^{-10}$.

13. CONCLUSIONS

Although there seem to be a large number of parameters to be computed and inequalities to be tested in order to produce the final error estimates the process is feasible. All of the steps can be completed within a single code. The current code in Appendix 3, Section 18, has also been built around the example in Section 12 and would have to be generalized for other applications, but the code provides a template on which to proceed. From the computational point of view the longest compute times involve the construction of the block matrices (116) and (125). Computing the approximate solution and the fundamental solution of the variational equation is relatively fast compared to these matrix constructions. It behooves anyone wishing to apply the methods of this paper to spend some effort vectorizing the matrix construction algorithms in Sections 9.1 and 10 as much as possible.

The parameter M in the Fredholm Lemma 3.2 is a significant parameter. From the example above, it is clear that it would be desirable to obtain as small a value for M as possible, since its magnitude affects the λ_i, $i = 1, 2$ parameters and λ_1 appears in the final error estimates. In particular, in the example above, the effect of M causes a very fine residual r for the approximate solution (145) to produce a pessimistic error estimate between the approximate solution and the exact solution in the end. From (40) the critical parameter λ_1 is linearly dependent on M.

14. Disclaimer

Certain commercial software products are identified in this paper in order to adequately specify the computational procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology nor does it imply that the software products identified are necessarily the best available for the purpose.

15. Appendix 1

In this appendix we present the derivation of the differentiation matrix (86). The derivation is based on a discussion of pseudospectral Chebyshev methods given in Gottlieb et al. [13], although a full derivation of the differentiation matrix is not given.

Lemma 15.1. *For some positive integer N let the Chebyshev points be given by*

$$\eta_k = \cos\left(\frac{k\pi}{N}\right) \tag{150}$$

on $[-1,1]$, for $k = 0, 1, \cdots, N$. The Lagrange interpolation polynomials at these points are given by

$$l_j(z) = \prod_{\substack{k=0 \\ k\neq j}}^{N} \left(\frac{z-\eta_k}{\eta_j-\eta_k}\right). \tag{151}$$

We have $l_j(\eta_k) = \delta_{jk}$. At the Chebyshev points designate

$$D_{kj} = l_j'(\eta_k) \tag{152}$$

The values for these derivatives are then given as

$$\begin{aligned} D_{00} &= \frac{2N^2+1}{6} \\ D_{NN} &= -D_{00} \\ D_{jj} &= \frac{-\eta_j}{2(1-\eta_j^2)}, j = 1, 2, \cdots, N-1 \\ D_{ij} &= \frac{c_i(-1)^{i+j}}{c_j(\eta_i-\eta_j)} \end{aligned} \tag{153}$$

for $i \neq j, i, j = 0, \cdots, N$ where

$$c_i = \begin{cases} 2, & i = 0 \text{ or } N; \\ 1, & \text{otherwise.} \end{cases} \tag{154}$$

Proof: The Chebyshev polynomial of degree N is given by

$$T_N(z) = \cos\left(N\cos^{-1}\right) \tag{155}$$

for $z \in [-1,\ 1]$.

Define the polynomial

$$g_j(z) = \frac{\left(1-z^2\right) T_N'(z)(-1)^{j+1}}{c_j N^2 \left(z-z_j\right)} \tag{156}$$

for $j = 0, \cdots, N$ and $c_0 = c_N = 2,\ c_j = 1$ for $1 \leq j \leq N-1$. Since $T_N'\left(z_j\right)$ will be shown below to equal zero, $T_N'(z)/\left(z-z_j\right)$ is a polynomial of degree $N-2$ so $g_j(z)$ is a polynomial of degree N. Thus, if we can show that $g_j\left(z_k\right) = \delta_{jk}$ for $k = 0, \cdots, N$, then by uniqueness $g_j(z) = l_j(z)$.

We first need to compute the following derivatives.

$$
\begin{aligned}
T_N'(z) &= \frac{-N\sin\left(N\cos^{-1}z\right)}{\sqrt{1-z^2}} \\
T_N''(z) &= \frac{-N^2\left(1-z^2\right)^{1/2}\cos\left(N\cos^{-1}z\right) - Nz\sin\left(N\cos^{-1}z\right)}{\left(1-z^2\right)^{3/2}} \\
T_N'''(z) &= -N^2\left[-\sin\left(N\cos^{-1}z\right)N\left(1-z^2\right)^{-1/2}\left(1-z^2\right)^{-1}\right. \\
&\quad \left.+\cos\left(N\cos^{-1}z\right)(-1)\left(1-z^2\right)^{-2}(-2z)\right] \\
&\quad -N\left[\sin\left(N\cos^{-1}z\right)\left(1-z^2\right)^{-3/2}\right. \\
&\quad +z\cos\left(N\cos^{-1}z\right)N\left(1-z^2\right)^{-1/2}\left(1-z^2\right)^{-3/2} \\
&\quad \left.+z\sin\left(N\cos^{-1}z\right)\left(\frac{-3}{2}\right)\left(1-z^2\right)^{-5/2}(-2z)\right] \qquad (157)\\
g_j'(z) &= \frac{(-1)^{j+1}}{c_jN^2}\left[\frac{(-2z)T_N'(z)}{z-z_j} + \frac{\left(1-z^2\right)T_N''(z)}{z-z_j}\right. \\
&\quad \left.+\frac{\left(1-z^2\right)T_N'(z)}{\left(z-z_j\right)^2}\right] \\
&= \frac{(-1)^{j+1}}{c_jN^2}\left[\frac{Nz\sin\left(N\cos^{-1}z\right)}{\left(z-z_j\right)\left(1-z^2\right)^{1/2}} - \frac{N^2\cos\left(N\cos^{-1}z\right)}{\left(z-z_j\right)}\right. \\
&\quad \left.+\frac{N\left(1-z^2\right)^{1/2}\sin\left(N\cos^{-1}z\right)}{\left(z-z_j\right)^2}\right]
\end{aligned}
$$

We will first establish that $g_j(z) = l_j(z)$. Clearly, since $\cos^{-1} z_k = k\pi/N$, $T_N(z_k) = 0$, and therefore, for $k \neq j,\ k \neq 0,\ N,\ j \neq 0,\ N,\ g_j(z_k) = 0$. For $k = j,\ j \neq 0,\ N$, using $T_N'(z)$ and L'Hospital's rule for

$$\lim_{z\to z_j}\frac{\sin\left(N\cos^{-1}z\right)}{z-z_j} = \frac{N(-1)^j}{\left(1-z_j^2\right)^{1/2}}, \qquad (158)$$

we have $g_j(z_j) = 1$. For $j = 0,\ z_0 = 1$ so that

$$g_0(z) = \frac{(-1)^{j+2}\left(1-z^2\right)^{1/2}\sin\left(N\cos^{-1}z\right)}{2N(z-1)}. \qquad (159)$$

For $z = z_k,\ k \neq 0,\ g_0(x_k) = 0$. Again apply L'Hospital's rule to show

$$g_0(z_0) = \frac{(-1)^2}{2N}\lim_{z\to 1}\left[N\cos\left(N\cos^{-1}z\right) - \frac{z\sin\left(N\cos^{-1}z\right)}{\left(1-z^2\right)^{1/2}}\right] = 1. \qquad (160)$$

For $j = N, z_N = -1$ and $g_N(z_k) = 0$ for $k = 0,\ 1, \cdots,\ N-1$. For $k = 0$, use L'Hospital's rule to show

$$g_N(z_N) = \frac{(-1)^{N+2}}{2N}\lim_{z\to -1}\left[-\frac{z\sin\left(N\cos^{-1}z\right)}{\left(1-z^2\right)^{1/2}} + N\cos\left(N\cos^{-1}z\right)\right] = 1. \qquad (161)$$

Therefore, $g_j(z) = l_j(z)$.

We now construct the entries in the differentiation matrix (177). These are given by $D_{jk} = g_k'(z_j)$ for $j,\ k = 0,\ 1, \cdots,\ N$. For $k \neq j,\ k \neq 0,\ N$, since $\sin(k\pi) = 0$ and $\cos(k\pi) = (-1)^k$,

$$g_j'(z_k) = \frac{c_k(-1)^{j+1}}{c_j\left(z_k - z_j\right)} \qquad (162)$$

where $c_k = 1$. For $j \neq 0,\ N,\ k = 0$, we have $z_0 = 1$ and, by L'Hospital's rule,

$$g_j'(z_0) = \frac{(-1)^{j+1}}{c_j N^2}\left[\frac{N}{1-z_j}\lim_{z\to 1}\left(\frac{\sin\left(N\cos^{-1}z\right)}{\left(1-z^2\right)^{1/2}}\right) - \frac{N^2}{1-z_j}\right] = \frac{c_0(-1)^j}{c_j\left(1-z_j\right)}, \tag{163}$$

where $c_0 = 2$. For $j \neq 0,\ N,\ k = N$, we have $z_N = -1$ and, by L'Hospital's rule,

$$g_j'(x_N) = \frac{(-1)^{j+1}}{c_j N^2}\left[\frac{N}{1+z_j}\lim_{z\to -1}\left(\frac{\sin\left(N\cos^{-1}z\right)}{\left(1-z^2\right)^{1/2}}\right) + \frac{N^2(-1)^N}{1+z_j}\right] = \frac{c_N(-1)^{j+N}}{c_j\left(z_N-z_j\right)} \tag{164}$$

where $c_N = 2$. For $j = 0,\ k \neq 0,\ N$,

$$g_0'(z_k) = \frac{-1}{c_o N^2}\left[(1+z_k)T_N''(z_k)\right] = \frac{c_k(-1)^k}{c_0\left(z_k-1\right)} \tag{165}$$

where $c_k = 1,\ c_0 = 2$. For $j = 0,\ k = 0$ we start with

$$g_0'(z) = \frac{1}{2N^2}\left[(1+z)T_N'(z)\right] \tag{166}$$

so that

$$g_0'(z) = \frac{1}{2N^2}\left[T_N'(z) + (1+z)T_N''(z)\right]. \tag{167}$$

Since $g_0'(z_0) = \lim_{z\to 1} g_0'(z)$ we need to find $T_N'(1)$ and $T_N''(1)$. From the construction of $T_N'(z)$ and L'Hospital's rule,

$$T_N'(1) = -N\lim_{z\to 1}\left(\frac{\sin\left(N\cos^{-1}z\right)}{\left(1-z^2\right)^{1/2}}\right) = N^2. \tag{168}$$

Also

$$\begin{aligned} T_N''(1) &= -N\lim_{z\to 1}\left[\frac{N\left(1-z^2\right)^{1/2}\cos\left(N\cos^{-1}z\right) + z\sin\left(N\cos^{-1}z\right)}{\left(1-z^2\right)^{3/2}}\right] \\ &= \frac{N\left(1-N^2\right)}{3}\lim_{z\to 1}\left(\frac{\sin\left(N\cos^{-1}z\right)}{\left(1-z^2\right)^{1/2}}\right) = \frac{N^4-N^2}{3}. \end{aligned} \tag{169}$$

Therefore

$$g_0'(z_0) = g_0'(1) = \frac{2N^2+1}{6}. \tag{170}$$

For $j \neq 0,\ N$, we use

$$\begin{aligned} T_N''(z_j) &= \frac{(-1)^{j+1}N^2}{1-z_j^2}, \\ T_N'''(z_j) &= \frac{3(-1)^{j+1}N^2 z_j}{\left(1-z_j^2\right)^2}, \end{aligned} \tag{171}$$

$c_j = 1$, and L'Hospital's rule to show

$$\begin{aligned} g_j'(z_j) &= \frac{(-1)^{j+1}}{N^2}\lim_{z\to z_j}\left[\frac{-2zT_N'(z)}{(z-z_j)} + \frac{\left(1-z^2\right)T_N''(z)}{(z-z_j)} - \frac{\left(1-z^2\right)T_N'(z)}{\left(z-z_j\right)^2}\right] \\ &= \frac{(-1)^{j+1}}{2N^2}\left[-4z_jT_N''(z_j) + \left(1-z_j^2\right)T_N'''(z_j)\right] = -\frac{z_j}{2\left(1-z_j\right)^2} \end{aligned} \tag{172}$$

Finally, for $j = N,\ k = N,\ c_N = 2$,

$$g_N'(z_N) = \frac{(-1)^{N+1}}{2N^2}\lim_{z\to -1}\left[-T_N'(z) + (1-z)T_N''(z)\right]. \tag{173}$$

By L'Hospital's rule

$$
\begin{aligned}
T_N'(-1) &= -N \lim_{z \to -1} \frac{\sin\left(N \cos^{-1} z\right)}{\left(1-z^2\right)^{1/2}} \\
&= -N^2(-1)^N \qquad (174)
\end{aligned}
$$

Also, by L'Hospital's rule,

$$
\begin{aligned}
T_N''(-1) &= \lim_{z \to -1} \left[\frac{-N^2\left(1-z^2\right)^{1/2} \cos\left(N \cos^{-1} z\right) - Nz \sin\left(N \cos^{-1} z\right)}{\left(1-z^2\right)^{3/2}} \right] \\
&= \frac{N^3-N}{3} \lim_{z \to -1} \left(\frac{\sin\left(N \cos^{-1} z\right)}{\left(1-z^2\right)^{1/2}} \right) \\
&= \frac{N^4-N^2}{3}(-1)^N \qquad (175)
\end{aligned}
$$

Therefore

$$
g_N'\left(z_N\right) = -\frac{2N^2+1}{6} = -g_0'\left(z_0\right). \qquad (176)
$$

16. Appendix 2: Bounds and Lipschitz Condition for $R(z,\beta)$

In this section we give a proof of Lemma 4.1. A lengthy, but direct, calculation shows

$$
\begin{aligned}
R(z,\beta) &= \int_0^1 \left[X_1\left(\hat{x} + s\frac{\hat{\omega}}{\omega}z, \hat{x}_{\hat{\omega}} + s\left(\hat{x}_\omega - \hat{x}_{\hat{\omega}}\right) + s\frac{\hat{\omega}}{\omega}z_\omega\right) - X_1\left(\hat{x}, \hat{x}_{\hat{\omega}}\right)\right] \frac{\hat{\omega}}{\omega} z ds \\
&+ \int_0^1 \left[X_2\left(\hat{x} + s\frac{\hat{\omega}}{\omega}z, \hat{x}_{\hat{\omega}} + s\left(\hat{x}_\omega - \hat{x}_{\hat{\omega}}\right) + s\frac{\hat{\omega}}{\omega}z_\omega\right) - X_2\left(\hat{x}, \hat{x}_{\hat{\omega}}\right)\right] \frac{\hat{\omega}}{\omega} z_\omega ds \\
&+ \int_0^1 \left[X_2\left(\hat{x} + s\frac{\hat{\omega}}{\omega}z, \hat{x}_{\hat{\omega}} + s\left(\hat{x}_\omega - \hat{x}_{\hat{\omega}}\right) + s\frac{\hat{\omega}}{\omega}z_\omega\right) - X_2\left(\hat{x}, \hat{x}_{\hat{\omega}}\right)\right] \left(\hat{x}_\omega - \hat{x}_{\hat{\omega}}\right) ds \qquad (177) \\
&+ \left(\frac{\hat{\omega}}{\omega} - 1\right) X_1\left(\hat{x}, \hat{x}_{\hat{\omega}}\right) z + \left(\frac{\hat{\omega}}{\omega} - 1\right) X_2\left(\hat{x}, \hat{x}_{\hat{\omega}}\right) z_{\hat{\omega}} \\
&+ \left[X_2\left(\hat{x}, \hat{x}_{\hat{\omega}}\right)\left(\hat{x}_\omega - \hat{x}_{\hat{\omega}}\right) + \beta X_2\left(\hat{x}, \hat{x}_{\hat{\omega}}\right) \dot{\hat{x}}_{\hat{\omega}} \right] \\
&+ \frac{\hat{\omega}}{\omega} X_2\left(\hat{x}, \hat{x}_{\hat{\omega}}\right)\left(z_\omega - z_{\hat{\omega}}\right).
\end{aligned}
$$

From

$$
\hat{x}_\omega - \hat{x}_{\hat{\omega}} = \int_0^1 \dot{\hat{x}}\left(t - \hat{\omega} - s\beta\right)(-\beta) ds \qquad (178)
$$

we have

$$
\left|\hat{x}_\omega - \hat{x}_{\hat{\omega}}\right| \leq |\beta| \left|\dot{\hat{x}}\right|. \qquad (179)
$$

Similarly

$$
\left|z_\omega - z_{\hat{\omega}}\right| \leq |\beta| \left|\dot{z}\right|. \qquad (180)
$$

Also, from

$$
\begin{aligned}
\left(\hat{x}_\omega - \hat{x}_{\hat{\omega}}\right) + \beta \dot{\hat{x}}_{\hat{\omega}} &= -\beta \int_0^1 \left[\dot{\hat{x}}_{\hat{\omega}}\left(t - \hat{\omega} - s\beta\right) - \dot{\hat{x}}_{\hat{\omega}}\left(t - \hat{\omega}\right)\right] ds \\
&= \beta^2 \int_0^1 \int_0^1 \ddot{\hat{x}}\left(t - \hat{\omega} - us\beta\right) s \; du \, ds \qquad (181)
\end{aligned}
$$

we have

$$
\left|\left(\hat{x}_\omega - \hat{x}_{\hat{\omega}}\right) + \beta \dot{\hat{x}}_{\hat{\omega}}\right| \leq \frac{\beta^2}{2} \left|\ddot{\hat{x}}\right| \qquad (182)
$$

Using (177) through (182), along with (3) and (4), we have

$$|R(z,\beta)| \le \mathcal{R}(z,\beta), \tag{183}$$

where

$$\begin{aligned} \mathcal{R}(z,\beta) &= 2\mathcal{K}\left|\frac{\hat{\omega}}{\omega}\right|^2 |z|^2 + 2\frac{|\beta||z|}{|\omega|}\left(|\hat{\omega}| + |\beta|\right) \\ &\quad + \frac{\beta^2}{2}\left(\mathcal{K}\left|\dot{\hat{x}}\right|^2 + \mathcal{B}\left|\ddot{\hat{x}}\right|\right) + \mathcal{B}\left|\frac{\hat{\omega}}{\omega}\right| |\beta|\,|\dot{z}| \end{aligned} \tag{184}$$

To establish the Lipschitz condition we start with the inequality

$$\begin{aligned} &|dX\left(\hat{x} + a_{11}, \hat{x}_{\hat{\omega}} + a_{12}; b_{11}, b_{12}\right) - dX\left(\hat{x} + a_{21}, \hat{x}_{\hat{\omega}} + a_{22}; b_{21}, b_{22}\right)| \\ &\le \mathcal{K}\left(|b_{11}| + |b_{12}|\right)\left(|a_{11} - a_{21}| + |a_{12} - a_{22}|\right) \\ &+\mathcal{K}\left(|a_{12}| + |a_{22}|\right)\left(|b_{11} - b_{21}| + |b_{12} - b_{22}|\right) \\ &+\mathcal{B}\left(|b_{11} - b_{21}| + |b_{12} - b_{22}|\right) \end{aligned} \tag{185}$$

We need to define some functions that will help simplify the relations somewhat. Let

$$\begin{aligned} \gamma &= s\beta + (1-s)\tilde{\beta} \\ q &= sz + (1-s)\tilde{z} \\ \dot{q} &= s\dot{z} + (1-s)\dot{\tilde{z}} \end{aligned} \tag{186}$$

for $0 \le s \le 1$, and define

$$\begin{aligned} \psi_1(q,\gamma) &= \hat{x} + \frac{\hat{\omega}}{\hat{\omega}+\gamma} q \\ \psi_2(q,\gamma) &= \hat{x}_{\hat{\omega}+\gamma} + \frac{\hat{\omega}}{\hat{\omega}+\gamma} q_{\hat{\omega}+\gamma} \\ \phi_1(q,\gamma) &= -\frac{\hat{\omega}}{(\hat{\omega}+\gamma)^2} q \\ \phi_2(q,\gamma) &= -\left[\dot{\hat{x}}_{\hat{\omega}+\gamma} + \frac{\hat{\omega}}{\hat{\omega}+\gamma} q_{\hat{\omega}+\gamma}\right] - \frac{\hat{\omega}}{(\hat{\omega}+\gamma)^2}\dot{q}_{\hat{\omega}+\gamma} \end{aligned} \tag{187}$$

Since we have earlier chosen $\beta, \tilde{\beta}$ so that

$$\begin{aligned} \hat{\omega} + \beta &\ge \frac{\hat{\omega}}{2} \\ \hat{\omega} + \tilde{\beta} &\ge \frac{\hat{\omega}}{2} \end{aligned} \tag{188}$$

it is easy to see that

$$\left|\frac{\hat{\omega}}{\hat{\omega}+\gamma}\right| \le 2 \tag{189}$$

From (186) we have the following integrals

$$
\begin{aligned}
\int_0^1 |q|^2 ds &\leq \frac{1}{3}\left(|z|+|\tilde{z}|\right)^2 \\
\int_0^1 |\gamma| ds &\leq \frac{1}{2}\left(|\beta|+\left|\tilde{\beta}\right|\right) \\
\int_0^1 |q| ds &\leq \frac{1}{2}\left(|z|+|\tilde{z}|\right) \\
\int_0^1 |\gamma||q| ds &\leq \frac{1}{3}\left(|\beta|+\left|\tilde{\beta}\right|\right)\left(|z|+|\tilde{z}|\right) \\
\int_0^1 |\dot{q}||q| ds &\leq \frac{1}{3}\left(|\dot{z}|+|\dot{\tilde{z}}|\right)\left(|z|+|\tilde{z}|\right) \\
\int_0^1 |\dot{q}||\gamma| ds &\leq \frac{1}{3}\left(|\dot{z}|+|\dot{\tilde{z}}|\right)\left(|\beta|+\left|\tilde{\beta}\right|\right) \\
\int_0^1 |\dot{q}| ds &\leq \frac{1}{2}\left(|\dot{z}|+|\dot{\tilde{z}}|\right)
\end{aligned}
\tag{190}
$$

Define the function

$$
F(z,\beta) = X\left(\hat{x}+\frac{\hat{\omega}}{\hat{\omega}+\beta}z, \hat{x}_{\hat{\omega}+\beta}+\frac{\hat{\omega}}{\hat{\omega}+\beta}z_{\hat{\omega}+\beta}\right). \tag{191}
$$

Taking partial derivatives of (191),

$$
\begin{aligned}
d_1F(z,\beta;y) &= \frac{\hat{\omega}}{\hat{\omega}+\beta}dX\left(\hat{x}+\frac{\hat{\omega}}{\hat{\omega}+\beta}z, \hat{x}_{\hat{\omega}+\beta}+\frac{\hat{\omega}}{\hat{\omega}+\beta}z_{\hat{\omega}+\beta}; y, y_{\hat{\omega}+\beta}\right) \\
d_2F(z,\beta;\eta) &= \eta dX\left(\hat{x}+\frac{\hat{\omega}}{\hat{\omega}+\beta}z, \hat{x}_{\hat{\omega}+\beta}+\frac{\hat{\omega}}{\hat{\omega}+\beta}z_{\hat{\omega}+\beta};\right. \\
&\qquad \left. -\frac{\hat{\omega}}{(\hat{\omega}+\beta)^2}z, -\left[\dot{\hat{x}}_{\hat{\omega}+\beta}+\frac{\hat{\omega}}{\hat{\omega}+\beta}\dot{z}_{\hat{\omega}+\beta}\right]-\frac{\hat{\omega}}{(\hat{\omega}+\beta)^2}z_{\hat{\omega}+\beta}\right) \\
d_1F(0,0;y) &= dX\left(\hat{x}, \hat{x}_{\hat{\omega}}; y, y_{\hat{\omega}}\right) \\
d_2F(0,0;\eta) &= \eta dX\left(\hat{x}, \hat{x}_{\hat{\omega}}; 0, -\dot{\hat{x}}_{\hat{\omega}}\right)
\end{aligned}
\tag{192}
$$

From the definition of $R(z,\beta)$ and (191) we have

$$
R(x,\beta) - R\left(\tilde{z},\tilde{\beta}\right) = F(z,\beta) - F\left(\tilde{z},\tilde{\beta}\right) - d_2F\left(0,0;\beta-\tilde{\beta}\right) - d_1F\left(0,0;z-\tilde{z}\right) \tag{193}
$$

From the definition of γ and q in (186) we define the derivative with respect to s as

$$
d_sF(q,\gamma;ds) = \left[d_1F\left(q,\gamma;z-\tilde{z}\right) + d_2F\left(q,\gamma;\beta-\tilde{\beta}\right)\right] ds. \tag{194}
$$

By the Fundamental Theorem of Calculus

$$
\int_0^1 d_sF(q,\gamma;ds) = F(z,\beta) - F\left(\tilde{z},\tilde{\beta}\right) \tag{195}
$$

We can the write, using (187) and (192)

$$
\begin{aligned}
&R(z,\beta) - R\left(\tilde{z},\tilde{\beta}\right) \\
&= \int_0^1 \left[d_1F\left(q,\gamma;z-\tilde{z}\right) - d_1F\left(0,0;z-\tilde{z}\right)\right] ds \\
&\qquad + \int_0^1 \left[d_2F\left(q,\gamma;\beta-\tilde{\beta}\right) - d_2F\left(0,0;\beta-\tilde{\beta}\right)\right] ds \\
&= \int_0^1 \left[dX\left(\psi_1(q,\gamma),\psi_2(q,\gamma);\psi_1(z,\gamma)-\psi_1(\tilde{z},\gamma),\psi_2(z,\gamma)-\psi_2(\tilde{z},\gamma)\right)\right. \\
&\qquad \left. -dX\left(\psi_1(0,0),\psi_2(0,0);\psi_1(z,0)-\psi_1(\tilde{z},0),\psi_2(z,0)-\psi_2(\tilde{z},0)\right)\right] ds \\
&\qquad + \int_0^1 \left[dX\left(\psi_1(q,\gamma),\psi_2(q,\gamma);\left(\beta-\tilde{\beta}\right)\phi_1(q,\gamma),\left(\beta-\tilde{\beta}\right)\phi_2(q,\gamma)\right)\right. \\
&\qquad\qquad \left. -dX\left(\psi_1(0,0),\psi_2(0,0);\left(\beta-\tilde{\beta}\right)\phi_1(0,0),\left(\beta-\tilde{\beta}\right)\phi_2(0,0)\right)\right] ds
\end{aligned} \tag{196}
$$

From (187) we note that $\psi_1(0,0) = \hat{x}$ and $\psi_2(0,0) = \hat{x}_{\hat{\omega}}$.

Then, using (185) through (196) it is possible to show with some effort that

$$
\begin{aligned}
\mathcal{R}_1\left(z,\beta,\tilde{z},\tilde{\beta}\right) &= 8\mathcal{K}\left(|z|+|\tilde{z}|\right) + \left(2\mathcal{K}\left|\dot{\hat{x}}\right| + \frac{\mathcal{B}}{|\hat{\omega}|}\right)\left(|\beta|+\left|\tilde{\beta}\right|\right) \\
\mathcal{R}_2\left(z,\beta,\tilde{z},\tilde{\beta}\right) &= \frac{\mathcal{K}}{3}\left(\frac{16}{|\hat{\omega}|}+4\right)\left(|z|+|\tilde{z}|\right)^2 \\
&\quad + \left(2\mathcal{K}\left|\dot{\hat{x}}\right| + \mathcal{B}\left(1+\frac{2}{|\hat{\omega}|}\right)\right)\left(|z|+|\tilde{z}|\right) \\
&\quad + \frac{2\mathcal{B}}{|\hat{\omega}|}\left(|\dot{z}|+\left|\dot{\tilde{z}}\right|\right) \\
&\quad + \frac{16\mathcal{K}}{3|\hat{\omega}|}\left(|z|+|\tilde{z}|\right)\left(|\dot{z}|+\left|\dot{\tilde{z}}\right|\right) \\
&\quad + \frac{\mathcal{B}\left|\ddot{\hat{x}}\right|}{2}\left(|\beta|+\left|\tilde{\beta}\right|\right)
\end{aligned} \tag{197}
$$

17. Appendix A.3: Bounds and Lipschitz Conditions for $S(g)$

Let $g \in \mathcal{N}$ and let $r = \delta$. Then from Lemma 5.1 and the selection of β so that $\hat{\omega}+\beta(g) \geq \frac{\hat{\omega}}{2}$, we have

$$
\left|\frac{\hat{\omega}}{\hat{\omega}+\beta(g)}\right| \leq 2 \tag{198}
$$

and

$$
\begin{aligned}
|S(g)| &= |\mathcal{R}_0(z(g),\beta(g))| \\
&\leq 2\mathcal{K}\left|\frac{\hat{\omega}}{\hat{\omega}+\beta(g)}\right|^2 |z(g)|^2 \\
&\quad + 2\frac{|\beta(g)||z(g)|}{|\hat{\omega}+\beta(g)|}\left(|\hat{\omega}|+|\beta(g)|\right) \\
&\quad + \frac{\beta(g)^2}{2}\left(\mathcal{K}\left|\dot{\hat{x}}\right|^2 + \mathcal{B}\left|\ddot{\hat{x}}\right|\right) \\
&\quad + \mathcal{B}\left|\frac{\hat{\omega}}{\hat{\omega}+\beta(g)}\right| |\beta(g)||\dot{z}(g)|
\end{aligned} \tag{199}
$$

If we combine (40), (198), and (199) we have

$$\begin{aligned} |S(g)| \leq & \left\{ 32\mathcal{K}{\lambda_1}^2 + \frac{16\lambda_0\lambda_1}{|\hat{\omega}|}\left(|\hat{\omega}| + 2\lambda_0\delta\right)\right. \\ & \left. +2{\lambda_0}^2\left(\mathcal{K}\left|\dot{\hat{x}}\right|^2 + \mathcal{B}\left|\ddot{\hat{x}}\right|\right) + 8\mathcal{B}\lambda_0\lambda_2\right\}\delta^2. \end{aligned} \tag{200}$$

Set

$$\begin{aligned} E_1(\delta) = & \left\{ 32\mathcal{K}{\lambda_1}^2 + \frac{16\lambda_0\lambda_1}{|\hat{\omega}|}\left(|\hat{\omega}| + 2\lambda_0\delta\right)\right. \\ & \left. +2{\lambda_0}^2\left(\mathcal{K}\left|\dot{\hat{x}}\right|^2 + \mathcal{B}\left|\ddot{\hat{x}}\right|\right) + 8\mathcal{B}\lambda_0\lambda_2\right\}\delta^2. \end{aligned} \tag{201}$$

and let F_1 be a positive constant such that

$$\begin{aligned} F_1 \geq & \; 32\mathcal{K}{\lambda_1}^2 + \frac{16\lambda_0\lambda_1}{|\hat{\omega}|}\left(|\hat{\omega}| + 2\lambda_0\delta\right) \\ & +2{\lambda_0}^2\left(\mathcal{K}\left|\dot{\hat{x}}\right|^2 + \mathcal{B}\left|\ddot{\hat{x}}\right|\right) + 8\mathcal{B}\lambda_0\lambda_2. \end{aligned} \tag{202}$$

Now let $g,\ \tilde{g} \in \mathcal{N}$ and again set $r = \delta$. Then, from (40), (46), and (197) and ,choosing $|g| \leq \delta$, we have, with some algebra,

$$\begin{aligned} |S(g) - S(\tilde{g})| \leq & \left[\lambda_1\left\{8\mathcal{K}\left(|z(g)| + |z(\tilde{g})|\right)\right.\right. \\ & \left. + \left(2\mathcal{K}\left|\dot{\hat{x}}\right| + \frac{\mathcal{B}}{|\hat{\omega}|}\right)\left(|\beta(g)| + |\beta(\tilde{g})|\right)\right\} \\ & +\lambda_0\left\{\frac{\mathcal{K}}{3}\left(\frac{16}{|\hat{\omega}|} + 4\right)\left(|z(g)| + |z(\tilde{g})|\right)^2\right. \\ & + \left(2\mathcal{K}\left|\dot{\hat{x}}\right| + \mathcal{B}\left(1 + \frac{2}{|\hat{\omega}|}\right)\right)\left(|z(g)| + |z(\tilde{g})|\right) \\ & + \frac{2\mathcal{B}}{|\hat{\omega}|}\left(|\dot{z}(g)| + |\dot{z}(\tilde{g})|\right) \\ & + \frac{16\mathcal{K}}{3|\hat{\omega}|}\left(|z(g)| + |z(\tilde{g})|\right)\left(|\dot{z}(g)| + |\dot{z}(\tilde{g})|\right) \\ & \left.\left. + \frac{\mathcal{B}\left|\ddot{\hat{x}}\right|}{2}\left(|\beta(g)| + |\beta(\tilde{g})|\right)\right\}\right]|g - \tilde{g}| \\ \leq & \left[\lambda_1\left\{32\mathcal{K}\lambda_1 + 4\lambda_0\left(2\mathcal{K}\left|\dot{\hat{x}}\right| + \frac{\mathcal{B}}{|\hat{\omega}|}\right)\right\}\right. \\ & +\lambda_0\left\{\frac{16{\lambda_0}^2}{3}\left(\frac{16}{|\hat{\omega}|} + 4\right)\delta + 4\lambda_1\left(2\mathcal{K}\left|\dot{\hat{x}}\right| + \mathcal{B}\left(1 + \frac{2}{|\hat{\omega}|}\right)\right)\right. \\ & + \frac{8\mathcal{B}\lambda_2}{|\hat{\omega}|} + \frac{256\mathcal{K}\lambda_1\lambda_2}{3|\hat{\omega}|}\delta \\ & \left.\left. +2\mathcal{B}\left|\ddot{\hat{x}}\right|\lambda_0\right\}\right]\delta\,|g - \tilde{g}| \end{aligned} \tag{203}$$

Finally, we set

$$
\begin{aligned}
E_2(\delta) \quad = \quad & \Bigg[\lambda_1\Bigg\{32\mathcal{K}\lambda_1 + 4\lambda_0\left(2\mathcal{K}\left|\dot{\hat{x}}\right| + \frac{\mathcal{B}}{|\hat{\omega}|}\right)\Bigg\} \\
& +\lambda_0\Bigg\{\frac{16{\lambda_0}^2}{3}\left(\frac{16}{|\hat{\omega}|}+4\right)\delta + 4\lambda_1\left(2\mathcal{K}\left|\dot{\hat{x}}\right| + \mathcal{B}\left(1+\frac{2}{|\hat{\omega}|}\right)\right) \\
& \quad +\frac{8\mathcal{B}\lambda_2}{|\hat{\omega}|} + \frac{256\mathcal{K}\lambda_1\lambda_2}{3\,|\hat{\omega}|}\delta \\
& \quad +2\mathcal{B}\left|\ddot{\hat{x}}\right|\lambda_0\Bigg\}\Bigg]\delta
\end{aligned} \tag{204}
$$

and let F_2 be a positive constant such that

$$
\begin{aligned}
F_2 \quad \geq \quad & \lambda_1\Bigg\{32\mathcal{K}\lambda_1 + 4\lambda_0\left(2\mathcal{K}\left|\dot{\hat{x}}\right| + \frac{\mathcal{B}}{|\hat{\omega}|}\right)\Bigg\} \\
& +\lambda_0\Bigg\{\frac{16{\lambda_0}^2}{3}\left(\frac{16}{|\hat{\omega}|}+4\right)\delta + 4\lambda_1\left(2\mathcal{K}\left|\dot{\hat{x}}\right| + \mathcal{B}\left(1+\frac{2}{|\hat{\omega}|}\right)\right) \\
& \quad +\frac{8\mathcal{B}\lambda_2}{|\hat{\omega}|} + \frac{256\mathcal{K}\lambda_1\lambda_2}{3\,|\hat{\omega}|}\delta \\
& \quad +2\mathcal{B}\left|\ddot{\hat{x}}\right|\lambda_0\Bigg\}
\end{aligned} \tag{205}
$$

18. Appendix 3: Main Matlab Script

This section includes the main script and supporting functions, except for "cheb.m", which is available in Trefethen [31]. These scripts are included as is. They are not necessarily the most efficient and are specifically oriented towards the Van der Pol equation example in Section 12. A user will have to modify the scripts for their particular problem.

```
global m N CS M0 M2 V0 V1 V2 T lambda DM
global a_bar
global startt endt
global D_hat
global A_hat
global NC
global zj
global m
global piinvN
%****************************************************\\
%User input\\
%****************************************************\\
cal_B = input('Bound on derivatives for right hand side of DDE. cal_B = ');
cal_K = input('Lipschitz condition on right hand side derivatives of DDE. cal_K = ');
lambda = input('Van der Pol Equation parameter lambda. lambda = ');
m = input('Approximate Solution Harmonics. m = ');
N = input('1/2 number of integration points for init. cond. function. N = ');
NC = input('Enter NC for NC + 1 Chebyshev points for collocation. NC = ');
P = input('Enter number of points for Trapezoidal integration on [-omega,0]. P = ');
O = input('Enter number integration points on [0,2pi], P/O = 1/6, e.g. 250/1500. O = ');

%******************************************************\\
%Computing the initial approximation function\\
%******************************************************\\
```

```
%initialize arrays
T = zeros(2*N,1);
i = zeros(2*N,1);
V0 = zeros(2*m-1,1);
V1 = zeros(2*m,1);
V2 = zeros(2*m,1);
M0 = zeros(2*N,2*m-1);
M2 = zeros(2*N,2*m-1);
A = zeros(2*m,1);
A0 = zeros(2*m,1);
DM = zeros(2*N,2*m);
CS = zeros(2*m,2*N);

%Open I/O file
fid = fopen('Est_Periodic_Sol_err_Output.txt','w');

%Projection integration steps
i = (1:2*N)';
T = (pi/(2*N))*(2*i-1);

%Set up fixed arrays
for n = 1:m
    CS(2*n-1,:) = cos((2*n-1)*T)';
    CS(2*n,:) = sin((2*n-1)*T)';
end
M0(:,1) = cos(T);
M2(:,1) = -cos(T);
for n = 1:m-1
    M0(:,2*n) = cos((2*n+1)*T);
    M0(:,2*n+1) = sin((2*n+1)*T);
    M2(:,2*n) = (-(2*n+1)\^2)*cos((2*n+1)*T);
    M2(:,2*n+1) = (-(2*n+1)\^2)*sin((2*n+1)*T);
end
for n = 1:m
    DM(:,2*n-1) = cos((2*n-1)*T);
    DM(:,2*n) = sin((2*n-1)*T);
end

%Initialize A0
A(1) = 1.0;
A(2) = 2.0;
a_bar = fsolve('Numerical_Galerkin',A)
fprintf(fid,'Frequency and Approximate Solution Coefficients \n$');
nabar = length(a_bar);
for io = 1:nabar
    fprintf(fid,'%15.8e \n',a_bar(io));
end

%calculate the residual

r = Van_der_Pol(a_bar);

norm\_r = max(abs(r))
fprintf(fid,'\n\nResidual error r = %15.8e \n',norm_r);
```

```
Lgt\_T = length(T);
xt = zeros(Lgt_T,1);
xdt = zeros(2,Lgt_T);
xdt\_temp = zeros(2,1);
for i = 1:2*N
    xt(i,1) = Galerkin_series(T(i),a_bar,m);
    xdt_temp = Derivative_series(T(i),a_bar,m);
    xdt(1,i) = xdt_temp(1,1);
    xdt(2,i) = xdt_temp(2,1);
end
norm_abs_xt = max(abs(xt(:,1)))
norm_abs_dxt = max(abs(xdt(1,:)))
norm_abs_ddxt = max(abs(xdt(2,:)))
fprintf(fid,'\n\nMax absolute value x(t) on [0,2pi] = %15.8e\n',norm_abs_xt);
fprintf(fid,'\n\nMax absolute value dx/dt(t) on [0,2pi] = %15.8e\n',norm_abs_dxt);
fprintf(fid,'\n\nMax absolute value d2xdt2(t) on [0,2pi] = %15.8e\n',norm_abs_ddxt);
%*************************************************************
%Plot residual error
%*************************************************************
figure;
plot(T,r);
title('Residual Error as a Function of Time over [0, 2\pi]');
xlabel('Time');
ylabel('Residual Error');
disp('press enter to continue')
pause;

%*************************************************************
%Phase plot
%*************************************************************
nt = 10000;
t = zeros(19999,1);
ii = (1:nt)';
t = (ii-1)*2*pi/nt;
x_hat = Phase_series(t,a_bar,m);
figure;
plot(x_hat(:,1),x_hat(:,2));
title('Phase Plot of Approximate Solution');
xlabel('X');
ylabel('dX/dt');
axis equal;
disp('press enter to continue')
pause;
%***************************************************************
% End of appoximation section
%
% Output from this section a_bar, norm_r
%***************************************************************
%***************************************************************
% Begin collocation section
%***************************************************************
%***************************************************************
% Get NC Chebyshev points, zj, and Create Chebyshev differentiation
% matrix, D. See Trefethen book.
```

```
%************************************************************
[D,zj]=cheb(NC);
%************************************************************
% Reset last row of D to account for initial condition row
%************************************************************
D(NC+1,1:NC) = 0.0;
D(NC+1,NC+1) = 1.0;
D_hat = zeros(2*(NC+1),2*(NC+1));
D_hat(1:NC+1,1:NC+1) = D(1:NC+1,1:NC+1);
D_hat(NC+2:2*(NC+1),NC+2:2*(NC+1)) = D(1:NC+1,1:NC+1);

%*************************************************************
% Set up fixed arrays for the Van der Pol problem
% Create the A_hat matrix in the Van der Pol case. It's constant.
% Initialize the B array and a temporary array
%*************************************************************
A_hat = zeros(2*(NC+1),2*(NC+1));
A = [0 1; -1/(a_bar(1,1)^2) 0];
for i = 1:NC
    A_hat(i,i) = A(1,1);
    A_hat(i,NC+1+i) = A(1,2);
    A_hat(NC+1+i,i) = A(2,1);
    A_hat(NC+1+i,NC+1+i) = A(2,2);
end
B = zeros(2,2);  % coefficients of linear delay term for Van der Pol
Temp = zeros(2,2);
*************************************************************
%Compute the fundamental matrix at several points from 0 to 2pi
%for plotting only
%get Lagrange weights from 0 to 2*pi for Fundamental Solution
%*************************************************************

disp('Plotting Fundamental matrix from 0 to 2pi');
%Weights = colloc(0,2*pi);
a = 0;
b = 2*pi;
%debug ***********************************************
Q = fix((b-a)/a_bar(1,1))+1;
B = zeros(2,2); tj = zeros(NC+1,1); g = zeros(2,1);
W0 = zeros(2*(NC+1),1); Weights = zeros(2*(NC+1),2,Q);
B_hat = zeros(2*(NC+1),2*(NC+1),Q);
%solve for weights on first step interval
omega = a_bar(1,1);
M = ((D_hat - (omega/2)*A_hat)^(-1));
W0(NC+1,1) = 1; W0(2*(NC+1),1) = 0;
Weights(:,1,1) = M*W0;
W0(NC+1,1) = 0; W0(2*(NC+1),1) = 1;
Weights(:,2,1) = M*W0;
%other intervals
for i = 2:Q
    B_hat(NC+1,1,i) = 2/omega; B_hat(2*(NC+1),NC+2,i) = 2/omega;
    tj = (omega/2)*zj + (1/2)*(2*a + (2*i-1)*omega);
    for k = 1:length(tj)-1
        g(1,1,1) = a_bar(2,1,1)*cos(tj(k));
```

```
        g(2,1,1) = -a_bar(2,1,1)*sin(tj(k));
        for n = 2:m
            g(1,1,1) = g(1,1,1) + a_bar(n*2,1)*cos(n*tj(k)) + a_bar(n*2-1,1)*sin(n*tj(k));
            g(2,1,1) = g(2,1,1) - n*a_bar(n*2,1)*sin(n*tj(k)) + n*a_bar(n*2-1,1)*cos(n*tj(k));
        end
        B(2,1,1) = (-2*lambda/a_bar(1,1,1))*g(1,1,1).*g(2,1,1);
        B(2,2,1) = (lambda/a_bar(1,1))*(1-g(1,1,1).^2);
        %fill up the B_hat_i matrix
        B_hat(k,k,i) = B(1,1,1);
        B_hat(k,(NC+1)+k,i) = B(1,2,1);
        B_hat((NC+1)+k,k,i) = B(2,1,1);
        B_hat((NC+1)+k,(NC+1)+k,i) = B(2,2,1);
    end
    Weights(:,1,i) = M*(omega/2)*B_hat(1:2*(NC+1),1:2*(NC+1),i)*Weights(:,1,i-1);
    Weights(:,2,i) = M*(omega/2)*B_hat(1:2*(NC+1),1:2*(NC+1),i)*Weights(:,2,i-1);
end
%end debug ******************************************
[max_row, max_col, max_plane] = size(Weights);
%interpolate the fundamental solution
pp = 1000;
tt = zeros(pp+1,1);
zz = zeros(pp+1,1);
yy = zeros(NC+1,1,1);
del = 2*pi/pp;
for ii = 1:pp+1
    %get a time value betwee 0 and 2*pi
    tt(ii) = (ii-1)*del;
    Z = vdp_interp(0,tt(ii),Weights);
    yz11(ii,1) = Z(1,1);
    yz21(ii,1) = Z(2,1);
    yz12(ii,1) = Z(1,2);
    yz22(ii,1) = Z(2,2);
end
figure;
subplot(2,2,1);
plot(tt(1:pp+1),yz11(1:pp+1,1));
title('Z(1,1)');
xlabel('tt');
ylabel('z11');
subplot(2,2,2);
plot(tt(1:pp+1),yz21(1:pp+1,1));
title('Z(2,1)');
xlabel('tt');
ylabel('z21');
subplot(2,2,3);
plot(tt(1:pp+1),yz12(1:pp+1,1));
title('Z(1,2)');
xlabel('tt');
ylabel('z12');
subplot(2,2,4);
plot(tt(1:pp+1),yz22(1:pp+1,1));
title('Z(2,2)');
xlabel('tt');
ylabel('z22');
```

```
disp('press enter to continue')
pause;
% clear out the arrays not needed
clear pp tt yy zz Z Weights

%*****************************************************************
% Computing the monodromy matrix. This is a 2*(P+1) x 2*(P+1) matrix
% Eigenvalues by Collocation. Integration by Trapezoidal rule.
%*****************************************************************
omega = a_bar(1,1);
delta = omega/P; % P is user selected number of Trapezoidal points
iii = 1:P+1;
s = -omega + (iii-1)*delta; %integration from -omega to 0 for monodromy matrix
U = zeros(2*(P+1),2*(P+1)); % the 2 is for the 2 x 2 blocks for VdP equat.
Z = zeros(2,2); %redefine Z again
endt = 2*pi;
B(1,1) = 0.0;
B(1,2) = 0.0;
delta1 = delta;
% Load by column
for jj = 1:P+1
    disp(sprintf('Creating Block column %d for U\n',jj))
    startt = s(jj) + omega;
    %get the weights for the number of step intervals between
    %startt = s(jj)+omega and endt = 2*pi
    Weights_eig = colloc(startt,endt);
    % Set up B matrix for integration step using trapezoidal rule
    x_hat = Vdp_series(s(jj),a_bar,m);
    B(2,1) = (-2*lambda/omega)*x_hat(1)*x_hat(2);
    B(2,2) = (lambda/omega)*(1.0 - x_hat(1)^2);
    % form row blocks for column j
    for ii = 1:P+1
        tt(ii) = s(ii) + 2*pi;
        Z = vdp_interp(startt,tt(ii),Weights_eig);
        if ((ii == 1)|(ii == P+1))
            delta1 = delta/2;
        end
        Temp = delta1*Z*B;
        % Fill in the i row block of the current column
        U(2*ii-1,2*jj-1) = Temp(1,1);
        U(2*ii-1,2*jj) = Temp(1,2);
        U(2*ii,2*jj-1) = Temp(2,1);
        U(2*ii,2*jj) = Temp(2,2);

    end
    clear weights_eig %need to clear since startt changes
end
%add the initial condition blocks to the last column
disp('Adding the initial condition blocks to the last column')
jj = P+1;
startt = 0;
Weights_eig = colloc(startt,2*pi);
for ii = 1:P+1
    tt(ii) = s(ii) + 2*pi;
```

```
    Z = vdp_interp(startt,tt(ii),Weights_eig);
    U(2*ii-1,2*(P+1)-1) = U(2*ii-1,2*(P+1)-1) + Z(1,1);
    U(2*ii,2*(P+1)-1) = U(2*ii,2*(P+1)-1) + Z(2,1);
    U(2*ii-1,2*(P+1)) = U(2*ii-1,2*(P+1)) + Z(1,2);
    U(2*ii,2*(P+1)) = U(2*ii,2*(P+1)) + Z(2,2);

end
eig(U)
disp('Operator matrix U is filled, now computing eigenvalues')
[V,Diag] = eig(U);
[row,col] = size(Diag);
abs_diag = zeros(row,1);
disp('Eigenvalues of U by colloc and max absolute value')
fprintf(fid,'\n\nEigenvalues of U\n');
for i = 1:row
    eigen(i,1) = Diag(i,i);
    re = real(eigen(i,1));
    im = imag(eigen(i,1));
    fprintf(fid,'%15.8e + i %15.8e\n',re,im);
end
for i = 1:row
    abs_diag(i,1) = abs(Diag(i,i));
end
max_abs_diag = max(abs_diag)
fprintf(fid,'\n\nMaximum Absolute Value of Eigenvalue for U\n')
fprintf(fid,'%15.8e\n',max_abs_diag );
%plot the first 20 eigenvalues of U
figure
hold on;
ang = 0:pi/100:2*pi;
plot(sin(ang),cos(ang),'b-')
plot(eigen,'rx')
hold off;
title('Eigenvalues for U');
xlabel('Real Part of Eigenvalue');
ylabel('Imaginary Part of Eigenvalue');
axis equal;
%disp('press enter to continue')
%pause;
%*****************************************************************
%Computing solution of the adjoint associated with characteristic
%multiplier near the unit circle
%*****************************************************************
%First fill the matrix U_tilde. According to Halanay this should have
%the same eigenvalues as U. Trapezoidal integration from -omega to 0.
%*****************************************************************
U_tilde = zeros(2*(P+1),2*(P+1));
delta = omega/P;
iii = 1:P+1;
s = -omega + (iii-1)*delta;
delta1 = delta;
B(1,1) = 0.0;
B(1,2) = 0.0;
%load by columns
```

```
for jj = 1:P+1
    disp(sprintf('Creating Block column %d for U_tilde\n',jj))
    startt = s(jj) + omega;
    Weights_adj_eig = colloc(startt, 2*pi);
    %Load row blocks for colum jj
    for ii = 1:P+1
        %B Matrix for Van der Pol
        x_hat = Vdp_series(s(ii),a_bar,m);
        B(2,1) = (-2*lambda/omega)*x_hat(1)*x_hat(2);
        B(2,2) = (lambda/omega)*(1.0-x_hat(1)^2);
        %Interpolate at row ii Z(s(ii)+2pi,s(jj)+omega)
        Z = vdp_interp(startt,s(ii)+2*pi,Weights_adj_eig);
        if ((ii == 1)|(ii == P+1))
            delta1 = delta/2;
        end
        Temp = delta1*B*Z;
        %Fill in the row blocks ii of the current column jj
        U_tilde(2*ii-1,2*jj-1) = Temp(1,1);
        U_tilde(2*ii-1,2*jj) = Temp(1,2);
        U_tilde(2*ii,2*jj-1) = Temp(2,1);
        U_tilde(2*ii,2*jj) = Temp(2,2);
    end
    clear Weights_adj_eig
end
%add initial condition block to first row
disp('Adding the initial condition blocks to the first row')
endt = 2*pi;
for jj = 1:P+1
    startt = s(jj) + omega;
    Weights_adj_eig = colloc(startt,endt);
    %Interpolate Z(2pi,s(jj)+omega)
    Z = vdp_interp(startt,endt,Weights_adj_eig);
    U_tilde(1,2*jj-1) = U_tilde(1,2*jj-1) + Z(1,1);
    U_tilde(2,2*jj-1) = U_tilde(2,2*jj-1) + Z(2,1);
    U_tilde(1,2*jj) = U_tilde(1,2*jj) + Z(1,2);
    U_tilde(2,2*jj) = U_tilde(2,2*jj) + Z(2,2);
    clear Weights_adj_eig
end
eig(U_tilde)
disp('Operator matrix U_tilde is now filled,');
disp('computing eigenvalues and eigenvectors of transpose');
disp('Transposing U_tilde is necessary in order for Matlab');
disp('to get column eigenvectors');
[PhiT,Dadj] = eig(U_tilde');
[row,col] = size(Dadj);
abs_diag = zeros(row,1);
disp('Eigenvalues by Colloc of U-tilde transpose and max abs val.');
fprintf(fid,'\n\nEigenvalues of U-tilde\n');
for i = 1:row
    eadj(i,1) = Dadj(i,i);
    re = real(eadj(i,1));
    im = imag(eadj(i,1));
    fprintf(fid,'%15.8e + i %15.8e\n',re,im);
end
```

```
for i = 1:row
    abs_diag(i,1) = abs(Dadj(i,i));
end
max_abs_diag = max(abs_diag);
fprintf(fid,'\n\nMaximum Absolute Value of Eigenvalue for U-tilde\n')
fprintf(fid,'%15.8e\n',max_abs_diag );
I_max_eig = find(abs_diag == max_abs_diag)
figure;
hold on;
ang = 0:pi/100:2*pi;
plot(sin(ang),cos(ang),'b-');
plot(eadj,'rx');
title('Eigenvalues of U-tilde');
xlabel('Real Part of Eigenvalue');
ylabel('Imaginary Part of Eigenvalue');
axis equal;
%disp('press enter to continue')
%pause;
%****************************************************************
%The first eigenvector in PhiT is associated with the first eigenvector of
%largest magnitude. Transpose it to make it a row vector.
%****************************************************************
pp(:,1) = PhiT(:,I_max_eig(1));
lgth = length(pp);
phi = zeros(1,lgth);
phi(1,1:lgth) = pp(:,1)';
U_adj = zeros(2*(P+1),2*(O+1));
%N+1 blocks
iii = 1:P+1;
s = -omega + (iii-1)*delta;
sp2pi = s + 2*pi;
endt = 2*pi;
kk = 1:O+1;
t = (kk-1)*2*pi/O;
for k = 1:O+1
    %Compute adjoint solution at the t(k)
    %This will be the k-th column
    disp(sprintf('Computing adjoint at y(t(%d))\n',k));
    %Load up the column k
    startt = t(k);
    if (k == 1)
        startt = 0.0;
    end
    %Get weights for Z(2pi,t(k))
    Weights_adj = colloc(startt,endt);
    dela1 = delta;
    B(1,1) = 0.0;
    B(1,2) = 0.0;
    % Load row block for column
    for ii = 1:P+1
        %disp(sprintf('Creating Block column %d for adjoint\n',ii))
        x_hat = Vdp_series(s(ii),a_bar,m);
        B(2,1) = (-2*lambda/omega)*x_hat(1)*x_hat(2);
        B(2,2) = (lambda/omega)*(1.0-x_hat(1)^2);
```

```
        %Check relation between start and end time of fundamental solution
        if (startt > sp2pi(ii))
            Z = zeros(2,2);
        elseif (startt == sp2pi(ii))
            Z = eye(2);
        else
            %Interpolate Z(s(ii)+2pi,t(k))
            Z = vdp_interp(startt,sp2pi(ii),Weights_adj);
        end
        if ((ii == 1) | (ii == P+1))
            delta1 = delta/2;  %Trapezoidal rule end weights
        end
        Temp = delta1*B*Z;
        % Fill in the row blocks of the current column
        U_adj(2*ii-1,2*k-1) = Temp(1,1);
        U_adj(2*ii-1,2*k) = Temp(1,2);
        U_adj(2*ii,2*k-1) = Temp(2,1);
        U_adj(2*ii,2*k) = Temp(2,2);
    end
    clear Weights_adj
end
endt = 2*pi;
%add initial condition block to first block row of column k
for k = 1:O+1
    startt = t(k);
    %Interpolate Z(2pi,t(k))
    if (k == 1)
        startt = 0.0;
    end
    Weights_adj = colloc(startt,endt);
    Z = vdp_interp(startt,endt,Weights_adj);
    U_adj(1,2*k-1) = U_adj(1,2*k-1) + Z(1,1);
    U_adj(2,2*k-1) = U_adj(2,2*k-1) + Z(2,1);
    U_adj(1,2*k) = U_adj(1,2*k) + Z(1,2);
    U_adj(2,2*k) = U_adj(2,2*k) + Z(2,2);
    clear Weights_adj
end
%Apply row eigenvector to U_adj
%v_0 is 1 row by 2*(O+1) columns since phi is 1 row by 2*(P+1) columns
%and U_adj is P+1 rows by O+1 columns
%v_0 might be complex
v_0 = phi*U_adj;
%normalizing v_0
norm_v_0 = norm(v_0,2);
v_0 = v_0/norm_v_0;

%*******************************************************************
%Computing the alpha parameter
%*******************************************************************
% Now compute the alpha parameter in Stokes theorem
disp('Computing alpha');
% First compute the J(omega,x_dot) function at the mesh points in [0,2*pi].
% compute alpha. First comput J(i)
B(1,1) = 0.0;
```

```
B(1,2) = 0.0;
alpha = 0.0;
J1 = zeros(2,O+1);
J1_temp = zeros(2,1);
w_trap = zeros(O+1,1);
for k = 1:O+1
    w_trap(k,1) = 2*pi/O;
    if ((k==1)|(k==O+1))
        w_trap(k,1) = pi/O;
    end
    %first set up B(t(k))
    x_hat = Vdp_series(t(k)-omega,a_bar,m);
    B(2,1) = (-2*lambda/omega)*x_hat(1)*x_hat(2);
    B(2,2) = (lambda/omega)*(1.0-x_hat(1)^2);
    %Multiply by the derivative at t(k) - omega
    x_hat_dot_delay = Derivative_series(t(k)-omega,a_bar,m);
    %add derivative at t(k)
    x_hat_dot_t = Derivative_series(t(k),a_bar,m);
    %compute J as 2 rows, 1 column
    J = x_hat_dot_t + B*x_hat_dot_delay;
    J_n(k,1) = norm(J,2);
    v0(1,1) = v_0(1,2*k-1);
    v0(1,2) = v_0(1,2*k);
    %form alpha
    alpha = alpha + w_trap(k,1)*v0*J;
end
alpha = 1/alpha
norm_J = max(J_n)
fprintf(fid,'\n\nPrinting estimate of norm of J on [0,2pi] = %15.8e\n',norm_J);
fprintf(fid,'\n\nPrinting alpha\n');
re = real(alpha);
im = imag(alpha);
fprintf(fid,'%15.8e + i %15.8e\n',re,im);
%This alpha might be complex but we really only use abs(alpha)
disp('Absolute value of alpha');
abs_alpha = abs(alpha)
fprintf(fid,'\n\nAbsolute Value of alpha\n');
fprintf(fid,'%15.8e\n',abs_alpha);
%disp('Press Enter to Continue');
%pause;
%*********************************************************************
%Now compute the M bound
%*********************************************************************

% get integral discretization steps

ds = omega/P;
dt = 2*pi/O;

% Identify points

iP = 1:P+1;
sP = -omega + (iP -1)*ds;
iO = 1:O+1;
```

```
t0 = (i0 -1)*dt;

%Trapezoidal weights for integration

w = zeros(P+1);
u = zeros(O+1);
w(1:P+1) = ds;
u(1:O+1) = dt;
w(1) = ds/2;
w(P+1) = ds/2;
u(1) = dt/2;
u(O+1) = dt/2;

% Set up some fixed matrices

% Get collocation weights between 0 and 2pi

Weights1 = colloc(0,2*pi);

% Interpolate to form fundamental solution at 2pi, i.e. Z(2*pi,0)

Z = vdp_interp(0,2*pi,Weights1);

% Form (i - Z(2*pi,0))^(-1)

I2 = eye(2);
ImZ_inv = (I2 - Z)^(-1);

% Form (i - Z(2*pi,0))^(-1)*Z(2*pi,0)

ImZ_inv_Z = ImZ_inv * Z;

% Set up some cell arrays
disp('Setting up cell arrays');
B(1,1) = 0;
B(1,2) = 0;
for iPP = 1:P+1
    Z = vdp_interp(0,sP(iPP)+2*pi,Weights1);
    Z_cell(iPP,1) = {Z*ImZ_inv};
    x_hat = Vdp_series(sP(iPP),a_bar,m);
    B(2,1) = (-2*lambda/omega)*x_hat(1)*x_hat(2);
    B(2,2) = (lambda/omega)*(1.0 - x_hat(1)^2);
    B_cell(iPP,1) = {B};
end
clear Weights1;
% Initialize the H arrays
IP = eye(2*(P+1));
H1 = zeros(2*(P+1), 2*(P+1));
H2 = zeros(2*(P+1), 2*(O+1));
H3 = zeros(2*(O+1), 2*(P+1));
H4 = zeros(2*(O+1), 2*(O+1));
H5 = zeros(2*(O+1), 2*(O+1));
% Compute H1
disp('Computing H1');
```

```
for j1 = 1:P+1
    disp(sprintf('Column block %d of %d columns of H1\n',j1,P+1));
    Weights2 = colloc(sP(j1)+omega,2*pi);
    Z1 = vdp_interp(sP(j1)+omega,2*pi,Weights2);
    for i1 = 1:P+1
        Temp1 = Z_cell{i1,1}*Z1;
        Z2 = vdp_interp(sP(j1)+omega, sP(i1)+2*pi, Weights2);
        Temp2 = (Temp1 +Z2)*B_cell{j1,1};
        Temp2 = w(j1)*Temp2; %Trapezoidal rule weights
        H1(2*i1-1,2*j1-1) = Temp2(1,1);
        H1(2*i1-1,2*j1) = Temp2(1,2);
        H1(2*i1,2*j1-1) = Temp2(2,1);
        H1(2*i1,2*j1) = Temp2(2,2);
    end
    clear Weights2;
end

% Compute H2
disp('Computing H2');
for k2 = 1:O+1
    disp(sprintf('Column block %d of %d columns of H2\n',k2,O+1));
    Weights3 = colloc(tO(k2), 2*pi);
    Z1 = vdp_interp(tO(k2),2*pi,Weights3);
    for i2 = 1:P+1
        if (tO(k2) > sP(i2)+2*pi)
            Z2 = zeros(2,2);
        elseif (tO(k2) == sP(i2)+2*pi)
            Z2 = eye(2);
        else
            Z2 = vdp_interp(tO(k2),sP(i2)+2*pi,Weights3);
        end
        Temp3 = Z_cell{i2,1}*Z1 + Z2;
        Temp3 = u(k2)*Temp3;
        H2(2*i2-1,2*k2-1) = Temp3(1,1);
        H2(2*i2-1,2*k2) = Temp3(1,2);
        H2(2*i2,2*k2-1) = Temp3(2,1);
        H2(2*i2,2*k2) = Temp3(2,2);
    end
    clear Weights3;
end
Weights1 = colloc(0,2*pi);
% Compute H3
disp('Computing H3');
for k3 = 1:O+1
    disp(sprintf('Row block %d of %d rows of H3\n',k3,O+1));
    Temp1 = vdp_interp(0,tO(k3),Weights1);
    for j3 = 1:P+1
        Weights2 = colloc(sP(j3)+omega,2*pi);
        if (j3 == 1)
            Temp4 = (Temp1*ImZ_inv_Z + Temp1)*B_cell{1,1};
            Temp4 = w(1)*Temp4;
            H3(2*k3-1,1) = Temp4(1,1);
            H3(2*k3-1,2) = Temp4(1,2);
            H3(2*k3,1) = Temp4(2,1);
```

```
            H3(2*k3,2) = Temp4(2,2);
        else
            if (t0(k3) < sP(j3) + omega)
                Temp2 = zeros(2,2);
            elseif (t0(k3) == sP(j3) + omega)
                Temp2 = eye(2);
            else
                Temp2 = vdp_interp(sP(j3)+omega,t0(k3),Weights2);
                Temp3 = vdp_interp(sP(j3)+omega,2*pi,Weights2);
                Temp4 = (Temp1*ImZ_inv*Temp3 + Temp2)*B_cell{j3,1};
                Temp4 = w(j3)*Temp4;
                H3(2*k3-1,2*j3-1) = Temp4(1,1);
                H3(2*k3-1,2*j3) = Temp4(1,2);
                H3(2*k3,2*j3-1) = Temp4(2,1);
                H3(2*k3,2*j3) = Temp4(2,2);
            end
        end
        clear Weights2;
    end
end

%Computing H4
disp('Computing H4');
for l4 = 1:O+1
    disp(sprintf('Row block %d of %d rows of H4\n',l4,O+1));
    Weights3 = colloc(t0(l4),2*pi);
    if (l4 == 1)
        Temp1 = vdp_interp(0,2*pi,Weights1);
        for k4 = 1:O+1
            Temp2 = vdp_interp(0,t0(k4),Weights1);
            Temp3 = Temp2*ImZ_inv*Temp1 + Temp2;
            Temp3 = u(1)*Temp3;
            H4(2*k4-1,1) = Temp3(1,1);
            H3(2*k4-1,2) = Temp3(1,2);
            H3(2*k4,1) = Temp3(2,1);
            H3(2*k4,2) = Temp3(2,2);
        end
    else
        Temp1 = vdp_interp(t0(l4),2*pi,Weights3);
        for k4 = 1:O+1
            if (t0(l4) > t0(k4))
                Temp2 = zeros(2,2);
            elseif (t0(l4) == t0(k4))
                Temp2 = eye(2);
            else
                Temp2 = vdp_interp(t0(l4),t0(k4),Weights3);
                Temp3 = vdp_interp(t0(l4),2*pi,Weights3);
                Temp4 = vdp_interp(0,t0(k4),Weights1);
                Temp5 = Temp4*ImZ_inv*Temp3 + Temp2;
                Temp5 = u(l4)*Temp5;
                H4(2*k4-1,2*l4-1) = Temp5(1,1);
                H4(2*k4-1,2*l4) = Temp5(1,2);
                H4(2*k4,2*l4-1) = Temp5(2,1);
                H4(2*k4,2*l4) = Temp5(2,2);
```

```
            end
        end
    end
    clear Weights3;
end
%Computing final bound
H5 = H3*pinv(IP - H1)*H2 + H4;
M = norm(H5,inf);
fprintf(fid,'\n\nPrinting M bound\n');
fprintf(fid,'%15.8e\n',M);
disp('Final M bound')
M
%disp('Press Enter to Continue');
%pause;
%*******************************************************
%Final error estimates
%*******************************************************
abs_omega = abs(a_bar(1));
disp('lambda_0');
lambda_0 = sqrt(2*pi)*abs_alpha
fprintf(fid,'\n\n lambda_0 = ');
fprintf(fid,'%15.8e\n',lambda_0);
disp('om_min_4L0r must be >= 0');
om_min_4L0r = omega-4*lambda_0*norm_r
fprintf(fid,'\n\n om_min_4L0r = ');
fprintf(fid,'%15.8e\n',om_min_4L0r);
lambda_1 = M*(1+sqrt(2*pi)*abs_alpha*norm_J)
fprintf(fid,'\n\n lambda_1 = ');
fprintf(fid,'%15.8e\n',lambda_1);
lambda_2 = (lambda_1/(abs_omega*M))*(1+2*M*cal_B)
fprintf(fid,'\n\n lambda_2 = ');
fprintf(fid,'%15.8e\n',lambda_2);
E11 = 32*cal_K*lambda_1^2;
E12 = (16*lambda_0*lambda_1/abs_omega)*(abs_omega+2*lambda_0*norm_r);
E13 = 2*lambda_0^2 *(cal_K*norm_abs_dxt^2 + cal_B*norm_abs_ddxt);
E14 = 8*cal_B*lambda_0*lambda_2;
disp('E1 must be <= 1')
E1 = (E11 + E12 + E13 + E14)*norm_r
fprintf(fid,'\n\n E1 = ');
fprintf(fid,'%15.8e\n',E1);
E21 = 32*cal_K*lambda_1^2;
E22 = 4*lambda_0*lambda_1*(2*cal_K*norm_abs_dxt + cal_B/abs_omega);
E23 = (16*lambda_1^2*lambda_0/3)*((16/abs_omega)+4)*norm_r;
E24 = 4*lambda_1*lambda_0*(2*cal_K*norm_abs_dxt + cal_B*(1 +2/abs_omega));
E25 = 8*cal_B*lambda_2*lambda_0/abs_omega;
E26 = 256*cal_K*lambda_1*lambda_2*lambda_0*norm_r/(3*abs_omega);
E27 = 2*cal_B*norm_abs_ddxt*lambda_0^2;
disp('E2 must be strictly < 1')
E2 = (E21 + E22 + E23 + E24 + E25 + E26 + E27)*norm_r
fprintf(fid,'\n\n E2 = ');
fprintf(fid,'%15.8e\n',E2);
disp('Final errors')
Final_error_x_star = 4*lambda_1*norm_r
Final_error_omega_star = 2*lambda_0*norm_r
```

```
fprintf(fid,'\n\n Final_error_x_star = ');
fprintf(fid,'%15.8e\n',Final_error_x_star);
fprintf(fid,'\n\n Final_error_omega_star = ');
fprintf(fid,'%15.8e\n',Final_error_omega_star);
%*****************************************************
fclose(fid);
```

18.1. **colloc.m.**

```
function Weights = colloc(a,b)
global D_hat
global A_hat
global a_bar
global NC
global zj
global lambda
global m


Q = fix((b-a)/a_bar(1,1))+1;
B = zeros(2,2,1); tj = zeros(NC+1,1); g = zeros(2,1);
W0 = zeros(2*(NC+1),1); Weights = zeros(2*(NC+1),2,Q);
B_hat = zeros(2*(NC+1),2*(NC+1),Q);
%solve for weights on first step interval
omega = a_bar(1,1);
M = ((D_hat - (omega/2)*A_hat)^(-1));
W0(NC+1,1) = 1; W0(2*(NC+1),1) = 0;
Weights(:,1,1) = M*W0;
W0(NC+1,1) = 0; W0(2*(NC+1),1) = 1;
Weights(:,2,1) = M*W0;
%other intervals
for i = 2:Q
    B_hat(NC+1,1,i) = 2/omega; B_hat(2*(NC+1),NC+2,i) = 2/omega;
    tj = (omega/2)*zj + (1/2)*(2*a + (2*i-1)*omega);
    for k = 1:length(tj)-1
        g(1,1,1) = a_bar(2,1,1)*cos(tj(k));
        g(2,1,1) = -a_bar(2,1,1)*sin(tj(k));
        for n = 2:m
            g(1,1,1) = g(1,1,1) + a_bar(n*2,1)*cos(n*tj(k)) + a_bar(n*2-1,1)*sin(n*tj(k));
            g(2,1,1) = g(2,1,1) - n*a_bar(n*2,1)*sin(n*tj(k)) + n*a_bar(n*2-1,1)*cos(n*tj(k));
        end
        B(2,1,1) = (-2*lambda/a_bar(1,1,1))*g(1,1,1).*g(2,1,1);
        B(2,2,1) = (lambda/a_bar(1,1))*(1-g(1,1,1).^2);
        %fill up the B_hat_i matrix
        B_hat(k,k,i) = B(1,1,1);
        B_hat(k,(NC+1)+k,i) = B(1,2,1);
        B_hat((NC+1)+k,k,i) = B(2,1,1);
        B_hat((NC+1)+k,(NC+1)+k,i) = B(2,2,1);
    end
    Weights(:,1,i) = M*(omega/2)*B_hat(1:2*(NC+1),1:2*(NC+1),i)*Weights(:,1,i-1);
    Weights(:,2,i) = M*(omega/2)*B_hat(1:2*(NC+1),1:2*(NC+1),i)*Weights(:,2,i-1);
end
```

18.2. **Derivative_series.m.**

```
function g=Derivative_series(t,a_bar,m)
g(1,:) = -a_bar(2,1)*sin(t);
g(2,:) = -a_bar(2,1)*cos(t);
for n = 2:m
    g(1,:) = g(1,:) - n*a_bar(n*2,1)*sin(n*t) + n*a_bar(n*2-1,1)*cos(n*t);
    g(2,:) = g(2,:) - (n^2)*a_bar(n*2,1)*cos(n*t) - (n^2)*a_bar(n*2-1,1)*sin(n*t);
end
```

18.3. **Galerkin_series.m.**

```
function g=Galerkin_series(t,a_bar,m)
g = a_bar(2,1)*cos(t);
for n = 2:m
    g = g + (a_bar(n*2,1)*cos(n*t) + a_bar(n*2-1,1)*sin(n*t));
end
```

18.4. **lagrint.m.**

```
function yi = lagrint(x,y,xi)
% lagrint  Interpolation with Lagrange polynomials of arbitrary degree
%
% Synopsis:  yi = lagrint(x,y,xi)
%
% Input:   x,y = tabulated data
%          xi  = point where interpolation is to be evaluated
%
% Output:    yi = value of y at x = xi obtained via interpolation with
%                 polynomial of degree n-1, where length(y) = length(x) = n

dxi = xi - x;        %  vector of xi - x(1), xi - x(2), ... values
n = length(x);       %  degree of polynomial is n-1
L = zeros(size(y));  %  preallocate L for speed

%  Refer to section 10.2.2 in text for explanation of vectorized code
%  used to compute Lagrange basis functions, L(j)
L(1) = prod(dxi(2:n))/prod(x(1)-x(2:n));       %  j = 1
L(n) = prod(dxi(1:n-1))/prod(x(n)-x(1:n-1));   %  j = n
for j=2:n-1
  num = prod(dxi(1:j-1))*prod(dxi(j+1:n));
  den = prod(x(j)-x(1:j-1))*prod(x(j)-x(j+1:n));
  L(j) = num/den;
end
yi = sum(y.*L);      %  Evaluate Polynomial: sum of y(j)*L(j), j=1..n
```

18.5. **Phase_series.m.**

```
function x_hat=Phase_series(t,a_bar,m)
x_hat(:,1)=a_bar(2,1)*cos(t);
x_hat(:,2)=-a_bar(2,1)*sin(t);
for n = 2:m
    x_hat(:,1) = x_hat(:,1) + a_bar(n*2,1)*cos(n*t) + a_bar(n*2-1,1)*sin(n*t);
    x_hat(:,2) = x_hat(:,2) - n*a_bar(n*2,1)*sin(n*t) + n*a_bar(n*2-1,1)*cos(n*t);
end
```

18.6. Van_der_Pol.m.

```
function vdp = Van_der_Pol(A)
global m N CS M0 M2 V0 V1 V2 T lambda DM
    vdp = (A(1)^2)*xdd(A) + A(1)*lambda*(delay_x(A).^2 - 1).*delay_xd(A) + x(A);
```

18.7. vdp_interp.m.

```
function Z = vdp_interp(startt,eval_pt,Weights)
%VDP_INTERP Lagrange interpolation using Chebyshev Points to generate a 2 x
%2 matrix Z  of the fundamental solution for Van der Pol equation. Frequency in a_bar(1).
global a_bar
global NC
global zj

omega = a_bar(1,1);
kk = fix((eval_pt - startt)/omega )+1;
zz = (2/omega)*eval_pt - (2*startt + (2*kk-1)*omega)/omega ;
yy(1:NC+1,1,1) = Weights(1:NC+1,1,kk);
Z(1,1) = lagrint(zj,yy,zz);
yy(1:NC+1,1,1) = Weights(NC+2:2*(NC+1),1,kk);
Z(2,1) = lagrint(zj,yy,zz);
yy(1:NC+1,1,1) = Weights(1:NC+1,2,kk);
Z(1,2) = lagrint(zj,yy,zz);
yy(1:NC+1,1,1) = Weights(NC+2:2*(NC+1),2,kk);
Z(2,2) = lagrint(zj,yy,zz);
```

18.8. Vdp_series.m.

```
function x_hat=Vdp_series(t,a_bar,m)
%set up for dde23
x_hat(1,:)=a_bar(2,1)*cos(t);
x_hat(2,:)=-a_bar(2,1)*sin(t);
for n = 2:m
    x_hat(1,:) = x_hat(1,:) + a_bar(n*2,1)*cos(n*t) + a_bar(n*2-1,1)*sin(n*t);
    x_hat(2,:) = x_hat(2,:) - n*a_bar(n*2,1)*sin(n*t) + n*a_bar(n*2-1,1)*cos(n*t);
end
```

References

[1] B. Balachandran, Nonlinear dynamics of milling processes', Philosophical Transactions of the Royal Society of London, Series A, 359 (2001) 793-819.

[2] H. T. Banks and F. Kappel, Spline Approximations for Functional Differential Equations, Journal of Differential Equations, 34 (1979) 496–522.

[3] E. Bueler, Chebyshev Collocation for Linear, Periodic Ordinary and Delay Differential Equations: *A Posteriori* Estimates, http://arxiv.org/abs/math.NA/0409464

[4] E. A. Butcher, H. Ma, E. Bueler, V. Averina, and Z. Szabo, Stability of linear time-periodic delay-differential equations via Chebyshev polynomials, Int. J. Numer. Meth. Engng, 59 (2004), 895–922.

[5] K. Engelborghs, T. Luzyanina, K. J. In 'T Hout, and D. Roose, Collocation Methods for the Computation of Periodic Solutions of Delay Differential Equations, SIAM J. Sci. Comp., 22 (2000) 1593–1609.

[6] D. E. Gilsinn, Discrete Fourier Series Approximation to Periodic Solutions of Autonomous Delay Differential Equations, DETC 2005-84038, Proceedings of IDETC/CIE 2005: ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, September 24-28, Long Beach, CA.

[7] D. E. Gilsinn, Approximating Limit Cycles of a Van der Pol Equation with Delay, Proceedings of Dynamic Systems and Applications 4, Ed. G. S. Ladde, N. G. Medhin, M. Sambandham, Dynamic Publishers, Inc., 2004, 270-276

[8] D. E. Gilsinn, Estimating Critical Hopf Bifurcation Parameters for a Second-Oder Delay Differential Equation with Application to Machine Tool Chatter, Nonlinear Dynamics 30 (2002) 103-154.

[9] D. E. Gilsinn, Computable Error Bounds for Approximate Periodic Solutions of Autonomous Delay Differential Equations, Nonlinear Dynamics, accepted for publication

[10] D. E. Gilsinn and F. A. Potra, Integral Operators and Delay Differential Equations, Journal of Integral Equations and Applications, 18, 3 (2006) 297–336.

[11] D. Gottlieb, The Stability of Pseudospectral-Chebyshev Methods, Math. Comp., 36 (1981) 107–118.

[12] D. Gottlieb and E. Turkel, Topics in Spectral Methods, Lecture Notes in Mathematics: 1127 A. Dold and B. Eckmann (Ed.), Springer-Verlag, Berlin, 1985.

[13] D. Gottlieb, M. Y. Hussaini, and S. A. Orszag, Theory and Applications of Spectral Methods, in Spectral Methods for Partial Differential Equations, R. G. Voigt, D. Gottlieb, and M. Y. Hussaini (Ed.), SIAM, Philadelphia, (1984) 1–54.

[14] S. Halanay, *Differential Equations: Stability, Oscillations, Time Lags*, Academic Press, New York, 1966.

[15] J. Hale, *Functional Differential Equations*, Springer-Verlag, New York, 1971.

[16] N. H. Hanna and S. A. Tobias, A theory of nonlinear regenerative chatter, ASME Journal of Engineering for Industry, 96 (1974) 247–255.

[17] T. Insperger and G. St*épá*n, Stability of the Damped Mathieu Equation with Time Delay, Journal of Dynamic Systems, Measurement, and Control, 125 (2003) 166–171.

[18] T. Luzyanina and K. Engelborghs, Computing Floquet Multipliers for Functional Differential Equations, International Journal of Bifurcation and Chaos, 22 (2002 2977–2989.

[19] T. Luzyanina and D. Roose, Numerical stability analysis and computation of Hopf bifurcation points for delay differential equations, Journal of Computational and Applied Mathematics 72 (1996) 379–392.

[20] G. A. Kemper, Spline Function Approximation for Solutions of Functional Differential Equations, SIAM J. Num. Anal. 12 (1975) 73–88.

[21] E. R. Marsh, D. S. Yantek, M. A. Davies, and D. E. Gilsinn, Simulation and measurement of chatter in diamond turning, Journal of Manufacturing Science and Engineering, 120 (1995) 230–235.

[22] A. H. Nayfeh, C. Chin,and J. Pratt, Perturbation Methods in Nonlinear Dynamics - Applications to Machining Dynamics, Journal of Manufacturing Science and Engineering, 119 (1997) 485–493.

[23] C. A. H. Paul, Developing a delay differential equation solver, Applied Numerical Mathematics, 9 (1992) 403–414.

[24] H. E. Salzer, Lagrangian interpolation at the Chebyshev points $x_{n,\nu} \equiv cos\,(\nu\pi/n)\,, \nu =)(1)n$; some unnoted advantages, The Computer Journal, 15 (1972) 156-159.

[25] L.F. Shampine and S. Thompson, Solving DDE's in MATLAB, Applied Numerical Mathematics, 37 (2001) 441–458.

[26] A. Stokes, A Floquet Theory for Functional Differential Equations, Proc. Nat. Acad, Sci., 48 (1962) 1330–1334.

[27] A. Stokes, On the Approximation of Nonlinear Oscillations, Journal of Differential Equations, 12 (1972) 535–558.

[28] A.P. Stokes, On the Existence of Periodic Solutions of Functional Differential Equations, Journal of Mathematical Analysis and Applications, 54 (1976) 634–652.

[29] E. Stone and A. Askari, Nonlinear models of chatter in drilling processes. Dynamical Systems 17 No. 1 (2002) 65–85.

[30] E. Stone and Sue Ann Campbell, Stability and bifurcation analysis of a nonlinear DDE model of drilling, Journal of Nonlinear Science 14 No. 1 (2004) 27–57.

[31] L. N. Trefethen, *Spectral Methods in Matlab*, Society for Industrial and Applied Mathematics, 2000.

[32] J. Tlusty, Machine dynamics, in Handbook of High-speed Machine Technology, Robert I. King (ed.), Chapman and Hall, New York, (1985) 49–153.

[33] M. Urabe and A. Reiter, Numerical Computation of Nonlinear Forced Oscillations by Galerkin's Procedure, Journal of Mathematical Analysis and Applications, 14 (1966) 107–140.

[34] M. X. Zhao and B. Balachandran, Dynamics and stability of milling process, International Journal of Solids and Structure, 38 (2001) 2233–2248.

[35] D. R. Will*é* and C. T. H. Baker, DELSOL - a numerical code for the solution of systems of delay-differential equations, Applied Numerical Mathematics, 9 (1992) 223–234.