Bicubic B-Spline Surface Approximation of Invariant Tori

Sita Ramamurti*
David E. Gilsinn†

August 11, 2010

Abstract

The invariant torus of a coupled system of Van der Pol oscillators is approximated using bicubic B-splines. The paper considers the case of strong nonlinear coupling. In particular, the shapes of invariant torii for the Van der Pol coupling parameter λ are computed in the range [0.1, 2.0]. Comparisons are given with results obtained by the MATLAB differential equation solver ode45. Very good normed residual errors of the determining equations for the approximate tori for the cases $\lambda = 0.1,\ 0.6$ are shown. At the upper limit of $\lambda = 2.0$ memory errors occured during the optimization phase for solving the determining equations so that full optimization for some knot sets was not achieved, but a visual comparison of the resulting invariant torus figure showed a close similarity to the solution using ode45.

Keywords: bicubic B-aplines, determining equations, invariant torus, large parameter case, optimization, Van der Pol oscillators

1 Introduction

Physical problems involving coupled vibrating systems often lead to ordinary differential equations of the form

$$\ddot{\mathbf{x}} + \mathbf{A}\mathbf{x} = \lambda \mathbf{N}(\mathbf{x}) \tag{1}$$

where $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{A}$ is diagonal, and $\mathbf{N}$ is a nonlinear function of $\mathbf{x}$. λ is a parameter that scales the nonlinearity. There are many forms of solutions to systems like (1), but we are interested in approximating a particular class of solutions. In particular, we are interested in systems that exhibit an invariant surface of solutions. That is a surface on which, if a soltion passes through a poit on the surface, then the entire solution remains on the surface. In the case of periodic solutions the surface is sometimes referred to as an invariant torus. In the case of small parameters there is a large literature on the existence and approximation of invariant torii (see e.g. Bogoliubov and Mitropolsky [1], Hale [9, 10], and Gilsinn [5]). For the case of a large parameter there is not much in the way of literature. In this study we hope to introduce a computational algorithm that, at least for a certain class of nonlinear systems, does seem to compute invariant tori to a high degree of accuracy for a larger range of parameters. In particular, following the work of Gilsinn [6, 7], we shall consider the system of coupled Van der Pol oscillators

*Department of Mathematics, Trinity Washington University, Washington D. C. 20017
†Mathematical and Computational Sciences Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg MD 20899-8910, david.gilsinn@nist.gov.

$$\begin{aligned} \ddot{x_1} + \mu_1^2 x_1 &= \lambda(1 - x_1^2 - \alpha_1 x_2^2)\dot{x_1} \\ \ddot{x_2} + \mu_2^2 x_2 &= \lambda(1 - \alpha_2 x_1^2 - x_2^2)\dot{x_2} \end{aligned} \tag{2}$$

Under the change of coordinates $x_1 = r_1^{1/2}\sin(\mu_1\omega_1)$, $\dot{x_1} = \mu_1 {r_1}^{1/2}\cos(\mu_1\omega_1)$, $x_2 = r_2^{1/2}\sin(\mu_2\omega_2)$, and $\dot{x_2} = \mu_2 {r_2}^{1/2}\cos(\mu_2\omega_2)$ this system becomes

$$\begin{aligned} \dot{\omega_1} &= 1 + \lambda\left(-\frac{1}{\mu_1}\right)(\sin(\mu_1\omega_1)\cos(\mu_1\omega_1) - r_1\sin^3(\mu_1\omega_1)\cos(\mu_1\omega_1) \\ &\quad -\alpha_1 r_2 \sin(\mu_1\omega_1)\cos(\mu_1\omega_1)\sin^2(\mu_2\omega_2)), \\ \dot{\omega_2} &= 1 + \lambda\left(-\frac{1}{\mu_2}\right)(\sin(\mu_2\omega_2)\cos(\mu_2\omega_2) \\ &\quad -\alpha_2 r_1 \sin^2(\mu_1\omega_1)\sin(\mu_2\omega_2)\cos(\mu_2\omega_2) - r_2\sin^3(\mu_2\omega_2))\cos(\mu_2\omega_2), \\ \dot{r_1} &= 2\lambda r_1(\cos^2(\mu_1\omega_1) - r_1\sin^2(\mu_1\omega_1)\cos^2(\mu_1\omega_1) \\ &\quad -\alpha_1 r_2\cos^2(\mu_1\omega_1)\sin^2(\mu_2\omega_2)), \\ \dot{r_2} &= 2\lambda r_2(\cos^2(\mu_2\omega_2) - \alpha_2 r_1\sin^2(\mu_1\omega_1)\cos^2(\mu_2\omega_2) \\ &\quad -r_2\sin^2(\mu_2\omega_2)\cos^2(\mu_2\omega_2)). \end{aligned} \tag{3}$$

which is of the form

$$\begin{aligned} \dot{\mathbf{\Omega}} &= \mathbf{d} + \lambda\mathbf{\Theta}(\mathbf{\Omega}, \mathbf{r}), \\ \dot{\mathbf{r}} &= \lambda\mathbf{R}(\mathbf{\Omega}, \mathbf{r}), \end{aligned} \tag{4}$$

with $\mathbf{\Omega} = \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix}$, $\mathbf{r} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$, $\mathbf{d} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$,

$$\mathbf{\Theta}(\mathbf{\Omega}, \mathbf{r}) = \begin{pmatrix} \Theta_1(\mathbf{\Omega}, \mathbf{r}) \\ \\ \Theta_2(\mathbf{\Omega}, \mathbf{r}) \end{pmatrix} = \begin{pmatrix} \left(-\frac{1}{\mu_1}\right)(\sin(\mu_1\omega_1)\cos(\mu_1\omega_1) \\ -r_1\sin^3(\mu_1\omega_1)\cos(\mu_1\omega_1) \\ -\alpha_1 r_2\sin(\mu_1\omega_1)\cos(\mu_1\omega_1)\sin^2(\mu_2\omega_2)) \\ \left(-\frac{1}{\mu_2}\right)(\sin(\mu_2\omega_2)\cos(\mu_2\omega_2) \\ -\alpha_2 r_1\sin^2(\mu_1\omega_1)\sin(\mu_2\omega_2)\cos(\mu_2\omega_2) \\ -r_2\sin^3(\mu_2\omega_2))\cos(\mu_2\omega_2) \end{pmatrix},$$

$$\mathbf{R}(\mathbf{\Omega}, \mathbf{r}) = \begin{pmatrix} R_1(\mathbf{\Omega}, \mathbf{r}) \\ \\ R_2(\mathbf{\Omega}, \mathbf{r}) \end{pmatrix} = \begin{pmatrix} 2r_1(\cos^2(\mu_1\omega_1) \\ -r_1\sin^2(\mu_1\omega_1)\cos^2(\mu_1\omega_1) \\ -\alpha_1 r_2\cos^2(\mu_1\omega_1)\sin^2(\mu_2\omega_2)) \\ 2r_2(\cos^2(\mu_2\omega_2) \\ -\alpha_2 r_1\sin^2(\mu_1\omega_1)\cos^2(\mu_2\omega_2) \\ -r_2\sin^2(\mu_2\omega_2)\cos^2(\mu_2\omega_2)) \end{pmatrix}.$$

For the current study we will assume that an invariant torus exists for (3) in the form

$$
\begin{aligned}
S &= \{(\mu_1\omega_1, \mu_2\omega_2, r_1, r_2) : r_1 = f(\mu_1\omega_1, \mu_2\omega_2), r_2 = g(\mu_1\omega_1, \mu_2\omega_2), \\
&\quad (\mu_1\omega_1, \mu_2\omega_2) \in T^2\}. \qquad (5)
\end{aligned}
$$

Using the invariance of the graphs of $f(\mu_1\omega_1, \mu_2\omega_2)$ and $g(\mu_1\omega_1, \mu_2\omega_2)$ under the dynamics generated by (3) we have

$$
\begin{aligned}
\dot{\mathbf{r}} &= D_{\omega_1}\mathbf{F}(\mu_1\omega_1, \mu_2\omega_2)\dot{\omega}_1 + D_{\omega_2}\mathbf{F}(\mu_1\omega_1, \mu_2\omega_2)\dot{\omega}_2 \\
&= \lambda\mathbf{R}(\mathbf{\Omega}, \mathbf{F}(\mu_1\omega_1, \mu_2\omega_2)). \qquad (6)
\end{aligned}
$$

where $\mathbf{F} = \begin{pmatrix} f \\ g \end{pmatrix}$. Now, from (4),

$$
\begin{aligned}
\dot{\omega_1} &= 1 + \lambda\Theta_1(\mathbf{\Omega}, \mathbf{F}(\mu_1\omega_1, \mu_2\omega_2)), \\
\dot{\omega_2} &= 1 + \lambda\Theta_2(\mathbf{\Omega}, \mathbf{F}(\mu_1\omega_1, \mu_2\omega_2)); \qquad (7)
\end{aligned}
$$

We therefore substitute (7) into (6) to get the following quasilinear partial differential equation that $\mathbf{F}(\mu_1\omega_1, \mu_2\omega_2)$ must satisfy in order for its graph to be an invariant torus (See [8], [13]).

$$
\begin{aligned}
N(\mathbf{F}(\mu_1\omega_1, \mu_2\omega_2)) &= D_{\omega_1}\mathbf{F}(\mu_1\omega_1, \mu_2\omega_2)(1 + \lambda\Theta_1(\mathbf{\Omega}, \mathbf{F}(\mu_1\omega_1, \mu_2\omega_2))) \\
&+ D_{\omega_2}\mathbf{F}(\mu_1\omega_1, \mu_2\omega_2)(1 + \lambda\Theta_2(\mathbf{\Omega}, \mathbf{F}(\mu_1\omega_1, \mu_2\omega_2))) \\
&- \lambda\mathbf{R}(\mathbf{\Omega}, \mathbf{F}(\mu_1\omega_1, \mu_2\omega_2)) = 0. \qquad (8)
\end{aligned}
$$

For the remainder of the discussion, we make the following substitutions in order to simplify the notation in the equations.

$$
\begin{aligned}
\theta &= \mu_1\omega_1, \\
\phi &= \mu_2\omega_2.
\end{aligned}
$$

2 Bicubic Spline Collocation Method

We now use the method of collocation with B-splines to approximate the functions $r_1 = f(\theta, \phi)$ and $r_2 = g(\theta, \phi)$. We let

$$
\begin{aligned}
0 &= \theta_0 < \theta_1 < \ldots < \theta_n = 2\pi \\
0 &= \phi_0 < \phi_1 < \ldots < \phi_m = 2\pi
\end{aligned}
$$

be a uniform partition of the rectangle ranging over $[0, 2\pi]$ in the θ direction and $[0, 2\pi]$ in the ϕ direction where n and m are the number of knots in the θ and ϕ directions, respectively and $\Delta h = \theta_{i+1} - \theta_i = 2\pi/n,\ i = 0, \ldots, n-1$ and $\Delta k = \phi_{j+1} - \theta_j = 2\pi/m,\ j = 0, \ldots, m-1$. We seek bicubic spline surfaces of the form

$$
\begin{aligned}
r_{1,nm}(\theta, \phi) &= \sum_{i=-1}^{n+1}\sum_{j=-1}^{m+1} c_{ij}B_i(\theta)B_j(\phi) \\
r_{2,nm}(\theta, \phi) &= \sum_{i=-1}^{n+1}\sum_{j=-1}^{m+1} d_{ij}B_i(\theta)B_j(\phi) \qquad (9)
\end{aligned}
$$

where $\{B_{-1}(\mu_1\theta), B_0(\mu_1\theta), \ldots, B_{n+1}(\mu_1\theta)\}$ are the B spline basis functions in the θ direction with knots $\theta_h,\ h = 0, \ldots, n$ and $\{B_{-1}(\mu_2\phi), B_0(\mu_2\phi), \ldots, B_{m+1}(\mu_2\phi)\}$ are the B spline basis functions in the ϕ direction with knots $\phi_k,\ k = 0, \ldots, m$. The functions $B_i(\theta)$ are defined in Prenter[12] as follows by introducing four additional knots in each direction satisfying

$$\begin{aligned} \theta_{-2} < \theta_{-1} < \theta_0 = 0 \quad &\text{and} \quad \theta_{n+2} > \theta_{n+1} > \theta_n = 2\pi \\ \phi_{-2} < \phi_{-1} < \phi_0 = 0 \quad &\text{and} \quad \phi_{n+2} > \phi_{n+1} > \phi_n = 2\pi \end{aligned} \tag{10}$$

$$B_i(\theta) = \frac{1}{h^3} \begin{cases} (\theta - \theta_{i-2})^3 & \text{if } \theta \in [\theta_{i-2}, \theta_{i-1}] \\ h^3 + 3h^2(\theta - \theta_{i-1}) + 3h(\theta - \theta_{i-1})^2 - 3(\theta - \theta_{i-1})^3, & \\ & \text{if } \theta \in [\theta_{i-1}, \theta_i] \\ h^3 + 3h^2(\theta_{i+1} - \theta) + 3h(\theta_{i+1} - \theta)^2 - 3(\theta_{i+1} - \theta)^3, & \\ & \text{if } \theta \in [\theta_i, \theta_{i+1}] \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The function $B_j(\phi)$ is defined similarly. These functions are twice continuously differentiable on the real line. We can therefore define $B_i'(\theta)$ and $B_i''(\theta)$ using the equations for $B_i(\theta)$. To compute the approximations $r_{1,nm}$ and $r_{2,nm}$ in (9) at the knots, we note that by substituting the knots θ_h and ϕ_k into the definitions of $B,\ \ B',\ \ B''$,

$$B_i(\theta_h)B_j(\phi_k) = \begin{cases} 16 & \text{if } h = i, k = j \\ 4 & \text{if } h = i-1, k = j \quad \text{or} \quad h = i, k = j-1 \\ & \quad \text{or} \quad h = i, k = j+1 \quad \text{or} \quad h = i+1, k = j \\ 1 & \text{if } h = i-1, k = j-1 \\ & \quad \text{or} \quad h = i-1, k = j+1 \\ & \quad \text{or} \quad h = i+1, k = j-1 \\ & \quad \text{or} \quad h = i+1, k = j+1 \end{cases} \tag{12}$$

and $B_i(\theta_h)B_j(\phi_k) \equiv 0 \quad \text{for} \quad h \geq i+2, k \geq j+2 \quad \text{and} \quad h \leq i-2, k \leq j-2.$
The partial derivatives of the interpolants $r_{1,nm}(\theta, \phi)$,$r_{2,nm}(\theta, \phi)$ at the knots can be easily found using the first derivatives and the products $B_i'(\theta_h)B_j(\phi_k)$ and $B_i(\theta_h)B_j'(\phi_k)$ of the spline functions in the tables below.

	$(\theta_{i-1}, \phi_{j-1})$	(θ_{i-1}, ϕ_j)	$(\theta_{i-1}, \phi_{j+1})$
$B_i'(\theta)B_j(\phi)$	$\frac{3}{\Delta h}$	$\frac{12}{\Delta h}$	$\frac{3}{\Delta h}$
$B_i(\theta)B_j'(\phi)$	$\frac{3}{\Delta k}$	0	$-\frac{3}{\Delta k}$

	(θ_i, ϕ_{j-1})	(θ_i, ϕ_j)	(θ_i, ϕ_{j+1})
$B_i'(\theta)B_j(\phi)$	0	0	0
$B_i(\theta)B_j'(\phi)$	$\frac{12}{\Delta k}$	0	$-\frac{12}{\Delta k}$

	$(\theta_{i+1}, \phi_{j-1})$	(θ_{i+1}, ϕ_j)	$(\theta_{i+1}, \phi_{j+1})$
$B_i'(\theta)B_j(\phi)$	$-\frac{3}{\Delta h}$	$-\frac{12}{\Delta h}$	$-\frac{3}{\Delta h}$
$B_i(\theta)B_j'(\phi)$	$\frac{3}{\Delta k}$	0	$-\frac{3}{\Delta k}$

Using the above information, the values of the approximations $r_{1,nm}$ and $r_{2,nm}$ at the knots can be expressed in terms of the parameters c_{ij} and d_{ij} as

$$\begin{aligned} r_{1,nm}(\theta_h, \phi_k) &= 16c_{h,k} + 4(c_{h-1,k} + c_{h,k-1} + c_{h,k+1} + c_{h+1,k}) \\ &\quad + c_{h-1,k-1} + c_{h-1,k+1} + c_{h+1,k-1} + c_{h+1,k+1} \\ r_{2,nm}(\theta_h, \phi_k) &= 16d_{h,k} + 4(d_{h-1,k} + d_{h,k-1} + d_{h,k+1} + d_{h+1,k}) \\ &\quad + d_{h-1,k-1} + d_{h-1,k+1} + d_{h+1,k-1} + d_{h+1,k+1} \end{aligned} \tag{13}$$

A similar computation shows that the derivatives $\frac{\partial r_{1,nm}}{\partial \theta}(\theta_h, \phi_k)$, $\frac{\partial r_{1,nm}}{\partial \phi}(\theta_h, \phi_k)$, $\frac{\partial r_{2,nm}}{\partial \theta}(\theta_h, \phi_k)$ and $\frac{\partial r_{2,nm}}{\partial \phi}(\theta_h, \phi_k)$ at the knots have the following representation in terms of the parameters.

$$\begin{aligned} \frac{\partial r_{1,nm}}{\partial \theta}(\theta_h, \phi_k) &= \frac{3}{\Delta h}(c_{h+1,k+1} + 4c_{h+1,k} + c_{h+1,k-1} \\ &\quad - c_{h-1,k+1} - 4c_{h-1,k} - c_{h-1,k-1}) \\ \frac{\partial r_{1,nm}}{\partial \phi}(\theta_h, \phi_k) &= \frac{3}{\Delta k}(c_{h+1,k+1} - c_{h+1,k-1} + 4c_{h,k+1} \\ &\quad - 4c_{h,k-1} + c_{h-1,k+1} - c_{h-1,k-1}) \\ \frac{\partial r_{2,nm}}{\partial \theta}(\theta_h, \phi_k) &= \frac{3}{\Delta h}(d_{h+1,k+1} + 4d_{h+1,k} + d_{h+1,k-1} \end{aligned}$$

$$-d_{h-1,k+1} - 4d_{h-1,k} - d_{h-1,k-1})$$

$$\frac{\partial r_{2,nm}}{\partial \phi}(\theta_h, \phi_k) = \frac{3}{\Delta k}(d_{h+1,k+1} - d_{h+1,k-1} + 4d_{h,k+1} - 4d_{h,k-1} + d_{h-1,k+1} - d_{h-1,k-1}) \tag{14}$$

To apply the collocation method, we choose the collocation points to coincide with the knots and then substitute the nodal values $r_{1,nm}(\theta_h, \phi_k)$,$r_{2,nm}(\theta_h, \phi_k)$ and their derivatives obtained in equations (13) and (14) into the partial differential equation (8). This results in the following system of $(n+1) \times (m+1)$ coupled nonlinear equations in $(n+3) \times (m+3)$ parameters c_{ij} and d_{ij}:

$$\begin{aligned}
&\left[\frac{3}{\Delta h}(c_{h+1,k+1} + 4c_{h+1,k} + c_{h+1,k-1} - c_{h-1,k+1} - 4c_{h-1,k} - c_{h-1,k-1})\right]((1 + \lambda\Theta_1(\mathbf{\Omega}, r_{1,nm}(\theta_h, \phi_k)))) \\
&+ \left[\frac{3}{\Delta k}(c_{h+1,k+1} - c_{h+1,k-1} + 4c_{h,k+1} - 4c_{h,k-1} + c_{h-1,k+1} - c_{h-1,k-1})\right](1 + \lambda\Theta_2(\mathbf{\Omega}, r_{1,nm}(\theta_h, \phi_k))) \\
&- \lambda\mathbf{R}(\mathbf{\Omega}, r_{1,nm}(\theta_h, \phi_k)) = 0.
\end{aligned} \tag{15}$$

for $h = 0, \ldots, n$ and $k = 0, \ldots, m$.

3 Periodicity Conditions

The surface we are approximating is a closed one generated by two periodic closed curves. The spline model is therefore chosen so that it is closed in both the θ and ϕ directions. This is accomplished by repeating both the knots and the control vertices as follows.

$(\theta_n, \phi_k) = (\theta_0, \phi_k), \quad 0 \le k \le m, \qquad (\theta_h, \phi_m) = (\theta_h, \phi_0), \quad 0 \le h \le n$

$c_{-1,j} = c_{n-1,j}$
$c_{n+1,j} = c_{1,j}, \qquad -1 \le j \le m+1$

$c_{i,-1} = c_{i,m-1}$
$c_{i,m+1} = c_{i,1}, \qquad -1 \le i \le n+1$

$c_{n,k} = c_{0,k}, \qquad 0 \le k \le m$
$c_{h,m} = c_{h,0}, \qquad 0 \le h \le n$

Application of the above periodic boundary conditions reduces equation (13) to a nm $\times$ nm block-pentadiagonal matrix equation of the form

$$\begin{bmatrix} r_{1,nm}(0,0) \\ r_{1,nm}(0,1) \\ r_{1,nm}(0,2) \\ \vdots \\ r_{1,nm}((n-1),(m-2)) \\ r_{1,nm}((n-1),(m-1)) \end{bmatrix} = \begin{bmatrix} D & B & 0 & \dots & 0 & B \\ B & D & B & 0 & \dots & 0 \\ 0 & B & D & B & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & B & D & B \\ B & 0 & \dots & 0 & B & D \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{0,2} \\ \vdots \\ c_{(n-1),(m-2)} \\ c_{(n-1),(m-1)} \end{bmatrix} \tag{16}$$

where D and B are $m \times m$ pentadiagonal matrices with entries

$$D = \begin{bmatrix} 16 & 4 & 0 & \dots & 0 & 4 \\ 4 & 16 & 4 & 0 & \dots & 0 \\ 0 & 4 & 16 & 4 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 4 & 16 & 4 \\ 4 & 0 & \dots & 0 & 4 & 16 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 4 & 1 & 0 & \dots & 0 & 1 \\ 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 1 & 4 & 1 \\ 1 & 0 & \dots & 0 & 1 & 4 \end{bmatrix}$$

with a similar matrix equation obtained for the approximation $r_{2,nm}$ by replacing the control points $c_{i,j}$ with $d_{i,j}$.

Similarly the system of partial derivatives of $r_{1,nm}$ in (14)reduce, after applying the periodic boundary conditions, to a block diagonal form

$$\begin{bmatrix} \frac{\partial r_{1,nm}}{\partial\theta}(0,0) \\ \frac{\partial r_{1,nm}}{\partial\theta}(0,1) \\ \frac{\partial r_{1,nm}}{\partial\theta}(0,2) \\ \vdots \\ \frac{\partial r_{1,nm}}{\partial\theta}((n-1),(m-2)) \\ \frac{\partial r_{1,nm}}{\partial\theta}((n-1),(m-1)) \end{bmatrix} = \begin{bmatrix} 0 & B & 0 & \dots & 0 & -B \\ -B & 0 & B & 0 & \dots & 0 \\ 0 & -B & 0 & B & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & -B & 0 & B \\ B & 0 & \dots & 0 & -B & 0 \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{0,2} \\ \vdots \\ c_{(n-1),(m-2)} \\ c_{(n-1),(m-1)} \end{bmatrix} \tag{17}$$

where B is a $m \times m$ diagonal matrix with entries

$$B = \frac{3}{\Delta h} \begin{bmatrix} 4 & 1 & 0 & \dots & 0 & 1 \\ 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 1 & 4 & 1 \\ 1 & 0 & \dots & 0 & 1 & 4 \end{bmatrix}$$

and

$$
\begin{bmatrix}
\frac{\partial r_{1,nm}}{\partial \phi}(0,0) \\
\frac{\partial r_{1,nm}}{\partial \phi}(0,1) \\
\frac{\partial r_{1,nm}}{\partial \phi}(0,2) \\
\vdots \\
\frac{\partial r_{1,nm}}{\partial \phi}((n-1),(m-2)) \\
\frac{\partial r_{1,nm}}{\partial \phi}((n-1),(m-1))
\end{bmatrix}
=
\begin{bmatrix}
D & B & 0 & \dots & 0 & B \\
B & D & B & 0 & \dots & 0 \\
0 & B & D & B & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & 0 & B & D & B \\
B & 0 & \dots & 0 & B & D
\end{bmatrix}
\begin{bmatrix}
c_{0,0} \\
c_{0,1} \\
c_{0,2} \\
\vdots \\
c_{(n-1),(m-2)} \\
c_{(n-1),(m-1)}
\end{bmatrix}
\tag{18}
$$

where D and B are $m \times m$ diagonal matrices with entries

$$
D = \frac{12}{\Delta k}
\begin{bmatrix}
0 & 1 & 0 & \dots & 0 & -1 \\
-1 & 0 & 1 & 0 & \dots & 0 \\
0 & -1 & 0 & 1 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & 0 & -1 & 0 & 1 \\
1 & 0 & \dots & 0 & -1 & 0
\end{bmatrix}
$$

and

$$
B = \frac{3}{\Delta k}
\begin{bmatrix}
0 & 1 & 0 & \dots & 0 & -1 \\
-1 & 0 & 1 & 0 & \dots & 0 \\
0 & -1 & 0 & 1 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & 0 & -1 & 0 & 1 \\
1 & 0 & \dots & 0 & -1 & 0
\end{bmatrix}
$$

with similar expressions obtained for the partial derivative approximations $\frac{\partial r_{2,nm}}{\partial \theta}$, $\frac{\partial r_{2,nm}}{\partial \phi}$ by replacing the control points $c_{i,j}$ with $d_{i,j}$.

4 MATLAB solution

We now describe the algorithmic details involved in computing the bicubic B-spline coefficients c_{ij} and d_{ij} by setting up an appropriate objective function to be mimimized using MATLAB's nonlinear minimization subroutine. Our objective function simply computes the "residual error": the difference between the derivative of the approximations $\dot{r_{1,nm}}$, $\dot{r_{2,nm}}$ and the derivatives $\dot{r_1}$, $\dot{r_2}$ respectively as given by the system (3)which is equivalent to the partial differential equation given by (8). Substituting equations (16),(17), and (18) into the partial differential equation system (15), we reduce it to a 2nm system of coupled nonlinear equations in 2nm parameters c_{ij} and d_{ij}. The determining equations can now be written as

$$
\begin{aligned}
R_{1,hk}(A) = \frac{\partial r_{1,nm}}{\partial \theta}(h,k)((1+\lambda\Theta_1(\mathbf{\Omega},\mathbf{r}(h,k)))) + \frac{\partial r_{1,nm}}{\partial \phi}(h,k)(1+\lambda\Theta_2(\mathbf{\Omega},\mathbf{r}(h,k))) \\
-\lambda\mathbf{R_1}(\mathbf{\Omega},\mathbf{r}(h,k)) = 0 \\
R_{2,hk}(A) = \frac{\partial r_{2,nm}}{\partial \theta}(h,k)((1+\lambda\Theta_1(\mathbf{\Omega},\mathbf{r}(h,k)))) + \frac{\partial r_{2,nm}}{\partial \phi}(h,k)(1+\lambda\Theta_2(\mathbf{\Omega},\mathbf{r}(h,k)))
\end{aligned}
$$

$$-\lambda \mathbf{R_2}(\mathbf{\Omega}, \mathbf{r}(h,k)) = 0 \tag{19}$$

for $h = 0, 1, \ldots, n,\ k = 0, 1, \ldots, m,$ where $\mathbf{A} = [\mathbf{c}; \mathbf{d}]$with $\mathbf{c} = (c_{-1,-1}, c_{-1,0}, \ldots, c_{n+1,m+1})^T$ and $\mathbf{d} = (d_{-1,-1}, d_{-1,0}, \ldots, d_{n+1,m+1})^T$. Equation (19)can then be used to find an approximation for the vector $\mathbf{A}$ by a minimization process, such as

$$\min_{c,d} \sum_{k=1}^{2nm} |R_k(A)|^2 \tag{20}$$

We use MATLAB's `fsolve` command to solve the system of nonlinear equations for the parameters. For an initial guess, we fix the following parameters for this case study:

$\alpha_1 = 0.2$, $\alpha_2 = 0.4$, $\mu_1 = 1$,$\mu_2 = 1.414$

and take the first approximation to the radii as

$r_{1,nm} = 3.53$, $r_{2,nm} = 1.18$

We note that the matrix form of equation (16) is

$$Ac = r \tag{21}$$

with the matrix A being nonsingular. That is, system (16) has a unique solution and consequently, the collocation approximate (9)satisfying the periodic boundary conditions is uniquely defined. We can therefore solve (21) by substituting the initial guess for $r_{1,nm}$ and $r_{2,nm}$ as 3.53 and 1.18 respectively into the right hand side of (21)to obtain the vector c of the unknown parameters c_{ij} as

$$c = A^{-1}r \tag{22}$$

A similar calculation yields the initial vector d of the unknown parameters d_{ij}.

5 Optimization Issues

For the first few runs, we coded the objective function and called up the `fsolve` optimization routine without supplying the sparsity pattern of the Jacobian. This worked well for small values of λ but as the strength of coupling increased, we had to increase the size of the knot domain for getting a good approximation as well as for the optimization to successfully terminate. Increasing the size of the knot domain for bigger values of λ slowed down the optimization considerably. Computationally, the model proved expensive even for $\lambda = 0.5$ which appeared to require a 40 by 40 knot domain for a reasonably good approximation.

The residual error for any point in the knot domain depends only on the nine surrounding parameters and therefore has very few non-zero partial derivatives. The Jacobian is sparse with well-structured blocks of non-zero elements. The computational cost of finite differencing which MATLAB uses to approximate the Jacobian, can therefore be significantly reduced if we can supply the structure of the Jacobian to Matlab. This is done by setting `JacobPattern` to `'on'` in `options`. This was our first improvement to the code. With this enhancement the results were encouraging for λ between 0.1 and 1.0: The number of function evaluations per iteration dropped significantly (for example, for $\lambda = 0.5$ with a 40 by 40 knot domain, the number of evaluations dropped from 3202 to just 32 per iteration!),and the time required for the optimization to successfully terminate reduced considerably. The following table summarizes these results. Unfortunately, supplying the

$\lambda = 0.5$, knots=40		
Implementation	Normalized Residual Error	Time for Optimization
Jacobian Structure Unspecified	2.26395e-007	4.0188e+003 secs
Jacobian Structure Specified	2.33638e-007	41.3750 secs

Jacobian Sparsity Pattern appears to be insufficient for λ bigger than or equal to 1.0 which require bigger knot domains for achieving the required accuracy. For these coupling strengths, running the code generates an 'Out of Memory' output in MATLAB. The figures below for $\lambda = 1.0$ & 2.0 were generated with a 50 by 50 knot domain and the optimization was incomplete. Although unoptimized, the bicubic spline surface appears to capture the surface generated by numerical integration of the full system as shown in figures (3) and (4)below.

6 Results

We present here the results of running our code for four different values of the coupling parameter. In each case, the error trends were computed between the numerically integrated angle-radial equations, equation (4)using the MATLAB ode45 solver, and the approximate system of phase equations on the torus, given by

$$\begin{aligned} \dot{\mathbf{\Omega}} &= \mathbf{d} + \lambda \mathbf{\Theta}(\mathbf{\Omega}, \mathbf{r_{nm}}(\theta, \phi)), \\ \mathbf{r} &= \mathbf{r_{nm}}(\theta, \phi) \end{aligned} \tag{23}$$

We set the error tolerance in these computations to 10^{-8} and tabulated the results. First, we consider the case $\lambda = 0.1$. We ran the code for three different sizes of the knot domain starting with 20 knots in each of the θ and ϕ directions. Figure (1)shows the full and approximate system integration for the case $\lambda = 0.1$.

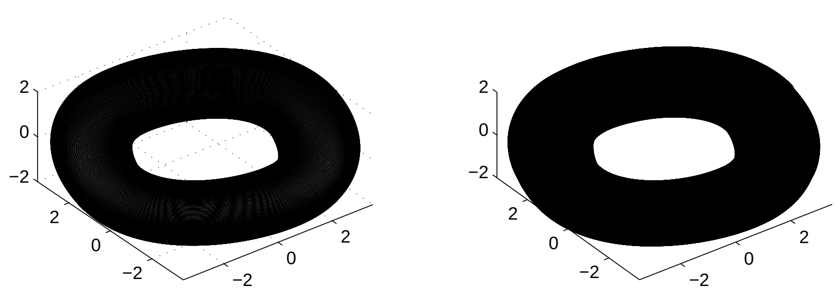

Figure 1: Bicubic approximation (left)and numerically integrated Van der Pol (right) for $\lambda = 0.1$ with a 30 by 30 knot domain

Table (2) shows the results of changing the number of knots keeping $\lambda = 0.1$ fixed. The third row shows the time T1 which represents the sum of the seconds it takes to create the necessary matrices in the main program plus the time it takes to evaluate once the function called by the optimization subroutine. T2 in seconds is the sum of T1 and the time it takes to run the optimization subroutine `fsolve` in MATLAB. The last row gives the normed residual error in each case.

Table 1: Summary table for $\lambda = 0.1$ showing timings and residual error.

$\lambda = 0.1$			
Knots	20	30	40
T1(seconds)	0	0.0160	0.0630
T2(seconds)	7.0620	13.3290	38.8750
Normed Residual Error	1.92718e-010	9.35666e-010	6.38028e-009

Figure (2)represents the $\lambda = 0.6$ case. We observe some deforming of the torus but the bicubic surface appears to be consistent with that generated by the MATLAB's ode45 solver.

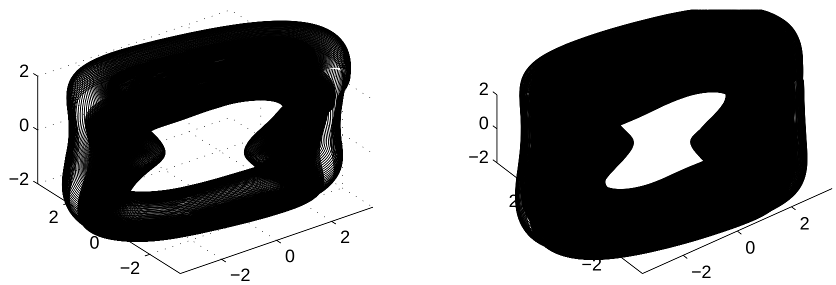

Figure 2: Bicubic approximation (left)and numerically integrated Van der Pol (right) for $\lambda = 0.6$ with a 45 by 45 knot domain

Table 2: Summary table for $\lambda = 0.6$ showing timings and residual error.

$\lambda = 0.6$			
Knots	40	45	50
T1(seconds)	0.0620	0.0780	0.1410
T2(seconds)	68.1250	80.7190	265.6710
Normed Residual Error	8.00207e-011	9.18135e-010	7.72686e-008

As mentioned in the previous section, for values of $\lambda \geq 1.0$ the optimization did not terminate successfully due to insufficient number of knots. Figures (3) and (4) depict two such cases. We note that as the strength of coupling increase, the inner sides of the invariant torus get closer and eventually connect with each other. Again, the results appear to be consistent.

7 Conclusions

This case study presents an alternate approach to approximating the invariant torus of the coupled van der Pol oscillator. The bicubic B-spline collocation method in conjunction with the vectorization capabilities offered by the MATLAB processing system provides us with the tools for developing a significant time saving optimization algorithm for efficiently computing the approximate periodic surface. Comparisons of the present approach with the MATLAB numerical integration of the full

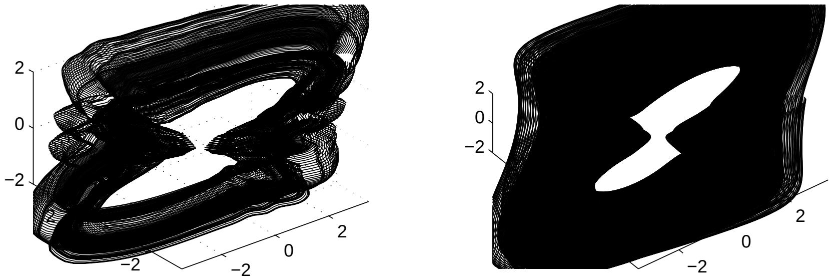

Figure 3: Bicubic approximation (left)and numerically integrated Van der Pol (right) for $\lambda = 1.0$ with a 50 by 50 knot domain

system of coupled oscillators indicate that the method demonstrated in this paper is promising. For strong nolinear coupling, computer memory appears to be the only limitation.

8 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 endoresement 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.

References

[1] Bogoliubov, N. N., Mitropolsky, Y. A., *Asymptotic Methods in the Theeory of Non-Linear Oscillations*, Gordon and Breach, New York, 1961.

[2] Dag, I., Irk, D.,and Sahin, A., ‘B-spline Collocation Methods for Numerical Solutions of the Burger’s Equation’,*Mathematical Problems in Engineering* **5**, 2005, 521-538.

[3] Dyksen, W., Houstis, E., Lynch, R., and Rice, J.,‘The performance of the collocation and Galerkin methods with Hermite bicubics’,*SIAM Journal of Numerical Analysis* **21**, 1984, 695-715.

[4] Ge, T., and Leung, A. Y. T.,‘Construction of Invariant Torus Using Toeplitz Jacobian Matrices/Fast Fourier Transform Approch’,*Nonlinear Dynamics* **15**, 1998, 283-305.

[5] Gilsinn, D. E., ‘Asymptotic Approximations of Integral Manifolds’, *SIAM J. Appl. Math.*, **47**, 5, 1987, 929-940.

[6] Gilsinn,D.E., ‘Constructing Invariant Tori for Two Weakly Coupled van der Pol Oscillators’, *Nonlinear Dynamics* **4**, 1993, 289-308.

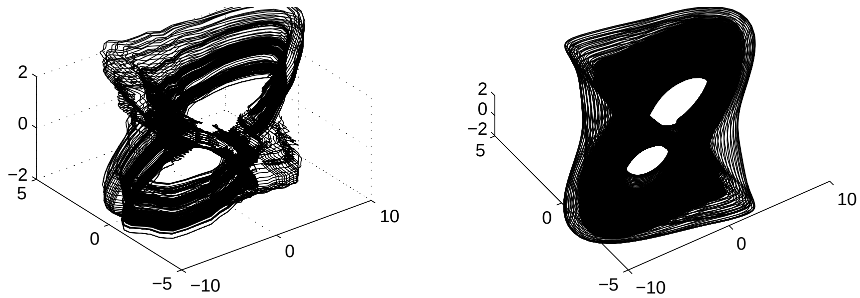

Figure 4: Bicubic approximation (left)and numerically integrated Van der Pol (right) for $\lambda = 2.0$ with a 50 by 50 knot domain

[7] Gilsinn,D.E., ‘Constructing Galerkin’s Approximations of Invariant Tori Using MACSYMA’, *Nonlinear Dynamics* **8**, 1995, 269-305.

[8] Guckenheimer, J., and Holmes, P., *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Springer-Verlag, New York, 1983, 130 -138.

[9] Hale, J. K., ’Integral manifolds of perturbed differential equations’, *Ann. of Math.*, **73**, 1961, 496-531.

[10] Hale, J. K., *Ordinary Differential Equations*, Wiley-Interscience, New York, 1969.

[11] Lu,F., and Milios, E. E., ‘Optimal spline fitting to planar shape’,*Signal Processing* **37**, 1994,129-140.

[12] Prenter,P.M., *Splines and Variational Methods*, John Wiley & Sons, 1975, 78 - 87.

[13] Wiggins,S., *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, Springer-Verlag, New York, 1990, 193 - 204.