Computational Methods for Continuation and Bifurcation of Nonlinear Dissipative Dynamical Systems

Chr. Kaas-Petersen

lic.techn.-thesis:
Computational Methods for Continuation and Bifurcation of Nonlinear Dissipative Dynamical Systems

by

Chr. Kaas-Petersen

May 1985

Laboratory of Applied Mathematical Physics,
The Technical University of Denmark, Bldg. 303,
DK-2800 Lyngby,
Denmark.

Til min familiee

Contents

Words of Thanks

"The eternal problem of the human being is how to structure his waking hours. (....) The most common, convenient, comfortable, and utilitarian method of structuring time is by a project designed to deal with the material of external reality: what is commonly known as work." (Berne, [2]). Part of my time has been used to structure the time of other persons; they in turn have partly structured my time. However, most of my time (alone or in company with others) I have tried to design some robust computational methods which could handle equilibrium points of nonlinear dissipative dynamical systems. The methods I have used are presented in this thesis; my results for some specific examples are presented in a number of papers.

This thesis has been written in order to fulfil part of the requirements for the lic.techn. degree (the danish PhD-degree). The papers have been written in order to distribute my ideas and my results.

Sincere thanks to my supervisors: associate professor lic.techn. Hans True, and associate professor lic.techn. Ove Skovgaard, both at the Laboratory of Applied Mathematical Physics (LAMF). They took me under their wings and allowed me to fly my own ways. Hans True told me the pros and cons of railway traffic and Ove Skovgaard supported me with excellent computing facilities.

Underneath the computational methods exist a world of topology, to which everybody refer. However, topology was not an easy matter for me. Then in a course held by Professor Ph.D. Vagn Lundsgaard Hansen (Mathematical Institute) all the basic things were cleared up. I therefore owe many thanks to him.

The software package PATH developed by me has been used by many persons. I thank them all for their criticism and suggestions.

Algorithms are not only used in computational methods. Algorithms are also used in cook book recipes. I thank the secretaries Lis Kaufmann and Lise Gudmandsen (LAMF) for the excellent way they establish the cake-tasting environment at tea-time.

Last I thank my family for support through all the years. I dedicate this thesis to them.

Preface

This PhD-study began with railway bogies and vehicles [32]. And the study will end with railway bogies and vehicles [26, 27]. In the meantime I have studied railway bogies and vehicles [22,33]. The dynamical equations of railway bogies and vehicles are nonlinear and fairly large (4-14 coupled autonomous ordinary differential equations). Computational methods are therefore a must. The software package PATH [25] was originally designed to autonomous ordinary differential equations. However, it turned out to be an easy task to include a number of other types of dynamical systems - both ordinary differential equations and difference equations. The advantages were twofold: first of all PATH became generally more applicable, second the difference equations were both easier and cheaper to study. Therefore I could study these simple systems and gain some useful experience before attacking the autonomous ordinary differential equations. By cheaper I point to the fact that at this University a CPU-time-unit on a computer has a price. I would be a poor man today if I had to pay that myself.

In this thesis the methods are presented, but no details - neither the systems nor the results - are given. The details are contained in the papers accepted for publication [20,22, 27,32,33], the papers (to be) submitted [21,23,24,26], and the programs documented in [25].

May 6, 1985 Chr. Kaas-Petersen

List of Frequently Used Symbols

c	control parameter of dynamical system, Sec.2
f	function describing the dynamical system, Sec.2
g	stability functions, Sec.6
p	equilibrium point, Sec.3
s	arclength along path of equilibrium points, Sec.6
t	time, Sec.2
v	eigenvector corresponding to a critical eigenvalue, Sec.6
x	state of dynamical system, Sec.2
x_0	initial state of solution φ at time t=0, Sec.2
y	= (p;c), i.e. equilibrium point on the path, Sec.5
z	= (x;c), Sec. 3
C	control space for dynamical system, Sec.2
E^c	(linear) approximation to bifurcating path, Sec.6
I	identity map, Sec.4
JORB	indicator for type of orbit, Sec.3
JSYS	indicator for type of system, Sec.3
O	orbit, i.e. graph of solution φ, Sec.3
P	Poincaré map, Sec.3
Q	residual map, Sec.3
T ,T'	periods of orbit, Sec.2
T_s,T'_s	periods of system, Sec.2
X	state space, Sec.2
φ	solution of dynamical system, Sec.3
λ	eigenvalue in linear stability analysis, Sec.4
$(\dot{})$	derivative with respect to time, Sec.2
$(')$	time increment of one time-unit, Sec.2
$( )^*$	value at bifurcation point, Sec.6
$d( )$	ordinary derivative
$\partial( )$	partial derivative
$D( )$	total derivative (a matrix of partial derivatives)
[ ]	number in square bracket is literature reference

1 Introduction

A wheel axle, see Fig. 1a, running with high speed along a track will make small but violent motions from side to side, i.e. motions perpendicular to the track direction. This motion, called the lateral motion, entails wear on the wheels and the rails, and therefore maintenance costs. The small, violent, lateral motions are transmitted to the car body, Fig. 1b, through the springs and the dampers in the bogie, see Fig. 1b, and through the springs and the dampers between bogie and car body. The motions of the car body are unwanted since they reduce the passenger comfort. The springs and dampers trigger other motions; yaw, i.e. small rotations of the wheel axles around a vertical axis, and roll, i.e. small rotations of the car body around a horizontal axis parallel to the track.

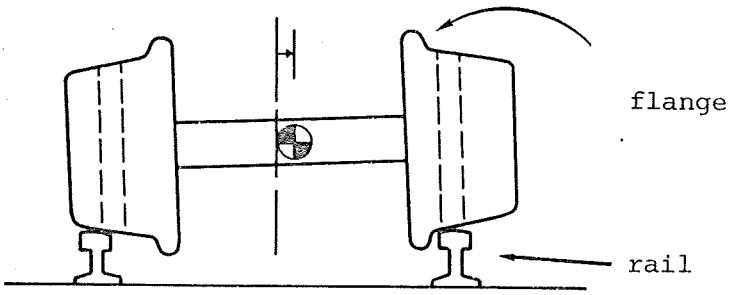


Fig. 1a. Wheel axle on rails. The wheel axle is shown in a position lateral to the centered position.

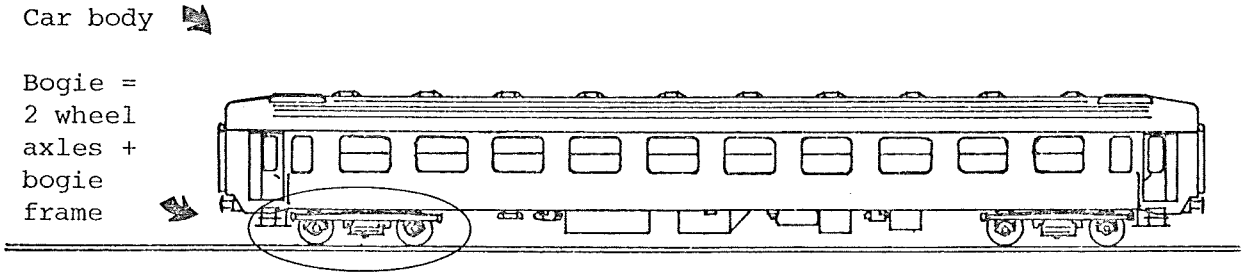


Fig. 1b. A railway vehicle consisting of two railway bogies and a car body.

We have examined this coupled motion in

M1) a simple bogie by which we mean a bogie with two wheel axles stiffly connected to a bogie frame,

M2) a complex bogie by which we mean a bogie with two wheel axles connected with springs and dampers to a bogie frame,

M3) a vehicle by which we mean a car body resting on two simple bogies.

We assume the speed is constant, and the track is straight, level, perfect, and horizontal. We have used the mathematical models of these systems due to Cooperrider [3]. We shall briefly describe the models.

The wheel axles, the wheels, the bogie frame, and the car body are treated as stiff elements. These elements are connected with springs and dampers, the forces of which we assume to be linear. The force between the wheel and the rail in the contact point is calculated according to Johnson's nonlinear theory, cf. [3]. This contact-force depends on the relative velocity of the wheel to the rail in the contact point; this relative velocity is denoted creep. The creep depends on the speed of the wheel axle. The wheel-flange, see Fig. 1a, and the rail may also contact each other. The flange-rail force is described by a socalled dead band spring with constant spring stiffness. This also introduces a nonlinear force. Invoking Newton's second law the dynamical equations can be formulated as ordinary differential equations (ODEs). Since we assume no forces depend explicitly on time, the ODEs are called autonomous. The equations are nonlinear due to the two wheel-rail forces being nonlinear. We shall only study the equations in dependence of one parameter, namely the speed. When the ODEs are formulated as a system of first order equations, the number of equations is 4 for M1, and 14 for M2 & M3. (The number of equations for a vehicle with a car body and two complex bogies will be 34. The computational cost for examination of this latter system will be very large.)

Calculations in our previous papers [22,32,33] show, that the ODEs have the stationary solution zero for all speeds. This corresponds to the bogie/vehicle running centered on the track. The stationary solution is stable for low speeds, which means that small disturbances from the zero eventually are damped away. The stationary solution is unstable for high speeds. The critical speed, where the stability changes a periodic solution is generated - one says that the periodic solution bifurcates from the stationary solution [32]. The bifurcating periodic solution is unstable (so it cannot be realized physically), but a stable periodic solution does exist, and that solution may be realized. More periodic solutions exist for other values of the speed - we say that all the periodic solutions make a path. The stable and the unstable periodic solutions above are on the same path. These results are common for all models.

In the case of the simple bogie (M1), only periodic solutions have been found [22].

In the case of the complex bogie (M2), we have also found chaotic behaviour, by which we mean behaviour very sensitive to small perturbations of the initial data [22,26].

In the case of the vehicle (M3), we have also found bi-periodic behaviour, that is two periodic motions with different periods coupled to each other [27,33].

In this thesis we shall review the methods we have used. The results can be found in the papers we refer to. All methods require a computer. We have examined other systems of autonomous ODEs as well as ODEs with explicit time dependences, namely periodic and bi-periodic. Furthermore we have also examined difference equations with explicit time dependences, namely periodic and bi-periodic. Difference equations are also denoted iterated maps [20,21,23]. These systems do not stem from railway engineering applications. The precise definitions of the dynamical systems are given in Sec. 2. We shall study stationary, periodic, bi-periodic,

and chaotic solutions; Sec. 3. The stability of the solution is studied in Sec. 4. How the solution depends on a parameter is examined in Sec. 5. We say, that each of the solutions trace a path. From this path other paths may branch off. This takes place in bifurcation points where we apply bifurcation theory to find the direction of the bifurcating path, Sec. 6. The computational methods have been implemented in a software package PATH [25].

2 Dissipative Dynamical Systems

The states x_i , $i=1,\ldots,n$ of a dynamical system changes with time t . All allowed states $x = (x_1,\ldots,x_n)$ of the system make up the state space X which is a subset of R^n . How the states x change with time is specified in the dynamical equations. We assume the dynamical equations are controlled by some parameters $c_1,\ldots,c_m$ which do not change with time. We shall restrict ourselves to one-parameter families of dynamical systems, i.e. $m=1$. The allowed values of the control parameter c of the system make up the control space C , which is a subset of R .

When the rate of change of the state x is known in any point of X at any time t for any value of the control parameter c in C , then the dynamical equations are formulated as ordinary differential equations (ODEs):

$$\text{rate of change of } x = dx/dt = \dot{x} = f(t,x;c),$$

where

$t \geqslant 0$,

$x = (x_1,\ldots,x_n) \in X \subset R^n$, $n \geqslant 1$,

$c \in C \subset R$,

$f = (f_1,\ldots,f_n)$, where f_i , $i=1,\ldots,n$ are real functions defined for all $t \geqslant 0$, $x \in X$, and $c \in C$.

f is then denoted a vector field.

When the state at time t, $x(t)$, determines the state at time $t+1$, $x(t+1)$, the dynamical equations are formulated as iterated maps (IMs):

$$\text{state } x \text{ one time unit later} = x(t+1) = \dot{x} = f(t,x;c),$$

where

$t = 0,1,2,3,\ldots,$

$x = (x_1,\ldots,x_n) \in X \subset R^n,\ n \geq 1,$

$c \in C \subset R,$

$f = (f_1,\ldots,f_n)$, where f_i, $i=1,\ldots,n$ are real functions defined for all $t=0,1,2,\ldots$, $x \in X$, and $c \in C$.

We have used the discrete time t and we have introduced the symbol $\dot{x}$ to stress the similarity between IMs and ODEs. Iterated maps are also denoted difference equations.

We assume f is smooth. When the $n \times n$ matrix $Df = (\partial f_i / \partial x_j)$ is not a diagonal matrix, the equations are said to be coupled.

We assume f is dissipative. This means, that the volume of any small volume element of the state space X is not constant in time. The technical definition is (Liouville's Theorem [29]) :

$$\begin{aligned} \operatorname{div}(\dot{x}) = \operatorname{div}(f) = \operatorname{trace}(Df) &\neq 0 \quad \text{for an ODE,} \\ |\det(Df)| &\neq 1 \quad \text{for an IM ,} \end{aligned}$$

for almost all t.

The dynamical system is said to be linear, when f is linear in x ; then we can write

$$f(t,x;c) = Df(t;c)\ x\ .$$

The dynamical system is said to be nonlinear, when f is not linear in x .

We shall deal with three cases of time dependence of the function f :

Autonomous systems

$\dot{x} = f(x;c)$: f does not depend explicitly on time.

Periodic systems

$\dot{x} = f(t,x;c)$: $f(\theta+T_s,x;c) = f(\theta,x;c)$

$\dot{x} = f(t,x;c)$: for all θ, x, and c, where

: $0<T_s$ is the period of the system,

: and T_s is an integer for an IM.

Bi-periodic systems

$\dot{x} = f(t,t,x;c)$: $f(\theta+T_s,\theta',x;c) = f(\theta,\theta',x;c)$,

$\dot{x} = f(t,t,x;c)$: $f(\theta,\theta'+T'_s,x;c) = f(\theta,\theta',x;c)$

: for all θ, θ', x, and c, where

: $0<T_s$, $0<T'_s$ are the periods of the

: system, and T_s is an

: integer for an IM.

We will introduce the indicator JSYS. For an ODE +JSYS shall be the number of periods in the function f ; for an IM -JSYS shall be the number of periods in the function f . Thus

JSYS = 0 for autonomous ODEs,

JSYS = +1 for periodic ODEs,

JSYS = -1 for periodic IMs,

JSYS = +2 for bi-periodic ODEs,

JSYS = -2 for bi-periodic IMs.

We note, that the scheme above can be generalized to tri-, quadra-, quinta-periodic,... systems.

3 Solutions of Dissipative Dynamical Systems

We let $\varphi(t;x_0;c)$ denote the solution of a dynamical system starting in the point x_0 of the state space X at time $t=0$, and the parameter c is given a value in C. The graph of the solution in the state space X is denoted the orbit O; thus

$$O(x_0;c) = \{x\in X: \; x=\varphi(t;x_0;c),\; t\geqslant 0\} \qquad \text{for an ODE,}$$

$$O(x_0;c) = \{x\in X: \; x=\varphi(t;x_0;c),\; t=0,1,2,...\} \quad \text{for an IM .}$$

We shall first deal with autonomous ODEs, the solutions of which are classified according to their geometry. When the orbit is a point in the state space X, see Fig. 2, we have a stationary solution. When the orbit is a closed curve in the state space X, see Fig. 3, we have a periodic solution.

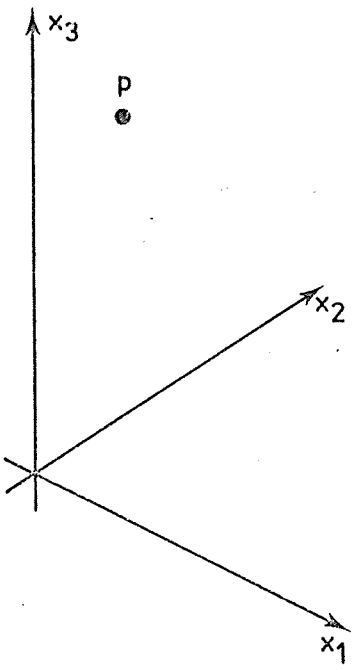


Fig. 2. Illustration of a stationary solution p in $X=R^3$ of an autonomous ODE.

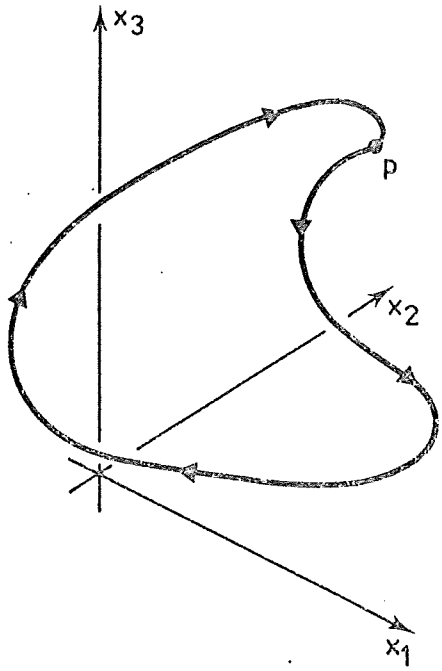


Fig. 3. Illustration of a periodic solution in $X=R^3$ of an autonomous ODE. The arrows on the curve points in the direction of increasing time; at time $t=0$ the solution starts in p, and returns to p after a time interval of length T, the period.

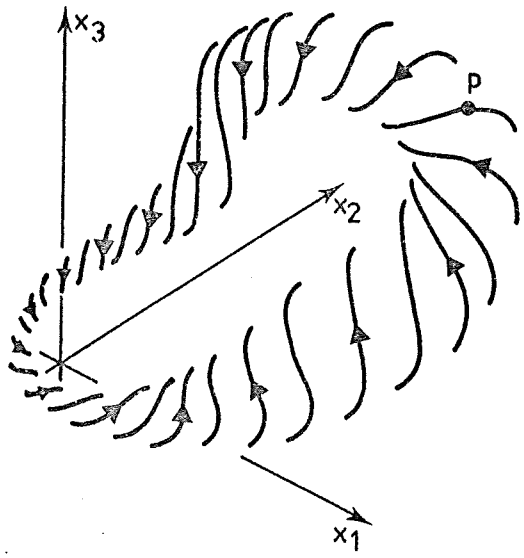


Fig. 4. Illustration of a bi-periodic solution in $X=R^3$ of an autonomous ODE. The arrows on the curve points in the direction of increasing time; at time $t=0$ the solution starts in p. The solution winds round and round on the torus. The solution may or may not return to the point p.

When the orbit is lying on a torus in the state space X, see Fig. 4, we have a bi-periodic solution.

Let us state the precise definitions of the orbits mentioned. The orbit of a stationary solution can be expressed in this way - cf. Fig. 2:

$$O(p;c) = \{x\in X: \quad x=\varphi(t;p;c)=p, \quad t\geqslant 0\} .$$

The orbit of a periodic solution - cf. Fig. 3 - is

$$O(p;c) = \{x\in X: \quad x=\varphi(t;p;c)=\varphi(t+T;p;c), \quad t\geqslant 0, \ T>0\} .$$

After a time interval T - the period - the solution repeats itself. A periodic solution should correctly be called a uni-periodic solution, since the solution $\varphi(\cdot;x_0;c)$ is periodic in one argument (here the first). A bi-periodic solution is periodic in two arguments. Therefore if we formally write the solution $\varphi(\cdot,\cdot;x_0;c)$ then φ is periodic with period T in the first argument and with period T' in the second argument. The orbit can then be expressed in this way:

$$O(p;c) = \{x\in X: \; x=\varphi(t,t;p;c)=\varphi(t+T,t\;\;;p;c) \;\&$$
$$x=\varphi(t,t;p;c)=\varphi(t\;\;,t+T';p;c) \;,$$
$$t\geq 0, \quad T>0, \quad T'>0 \qquad\qquad \} \;.$$

In Fig. 4 one can see, that the core of the torus is a closed curve around which the orbit winds. The first period T is the time it takes to do one turn along the core of the torus, the second period T' is the time it takes to wind one turn around the core of the torus.

We shall collectively refer to p as an equilibrium point. In this way we identify all of the orbit with one point on the orbit. This identification is described a little more detailed later in this section; we there define a map which has that single point p as a zero point.

The scheme of solutions above can be generalized to tri-, quadra-, quinta-periodic,... solutions.

Until now we have dealed with autonomous ODEs. Periodic ODEs may have periodic and bi-periodic solutions; however, the period T of the solution can be any integer multiple of the period T_s of the system. Bi-periodic ODEs may have bi-periodic solutions; however, the periods T and T' of the solution can be any integer multiple of the periods T_s and T'_s respectively of the system.

The orbits of the IMs differ from the orbits of the ODEs in one respect: the time t is discrete for the IMs. This means, that a periodic orbit is a finite set of points in the state space X (not a closed curve) which is repeated after one period T, which therefore must be an integer. Periodic IMs may have periodic and bi-periodic solutions; however, the (integer) period T of the solution can be any integer multiple of the (integer) period T_s of the system. Bi-periodic IMs may have bi-periodic solutions; however, the (integer) period T and the (real) period T' of the solution can be any integer multiple of the (integer) period T_s and the (real) period T'_s respectively of the system.

We note that a nonlinear dynamical system may for one fixed value of the parameter c have either different types

of solutions or it may have different solutions of the same type (or both) in the state space X .

We will introduce the indicator JORB. JORB is the number of periods in the solution. JORB therefore characterizes the orbit $O(p;c)$. In the scheme below, we have condensed the text of Sec. 2 and this section. In the scheme a '+' means that the dynamical system (JSYS indicator) may have that orbit (JORB indicator).

JORB \ JSYS	..	-3	-2	-1	0	+1	+2	+3	..
0					+				
1				+	+	+			
2			+	+	+	+	+		
3		+	+	+	+	+	+	+	
:	:	:	:	:	:	:	:	:	:

This scheme illustrates that

$$|JSYS| \leqslant JORB \ .$$

The limit $JORB \to \infty$ corresponds to a chaotic solution [8,12,29].

We shall describe computational methods for the 9 cases

$$-2 \leqslant JSYS \leqslant +2 \quad \& \quad 0 \leqslant JORB \leqslant 2 \ .$$

First of all, we must compute the orbit $O(x_0;c)$ for any given initial condition x_0 at $t=0$. For IMs we just insert $(t,x;c)$ in the function f and obtain the new state. The ODEs must be solved numerically. We have used the IMSL-routine DVERK [18] and the LSODA-routine [31]. DVERK is a 5th-6th order Runge-Kutta method good for socalled nonstiff equations. LSODA uses 2nd-12th order Adams-Bashforth methods for nonstiff equations and 2nd-5th order Backward Differentiation Formula methods for stiff equations - furthermore LSODA switches automatically between these methods. Interestingly enough, an ODE-solver generates an IM from an ODE. "Bad ODE-solvers" make IMs with spurious orbits which do not stem from orbits of the ODEs. These spurious orbits are denoted ghost solutions [34]. We have not observed ghost solutions with DVERK and LSODA.

Now we will define the maps, which have the equilibrium points p as zero points.

A stationary solution (JORB=0) is a zero point of the vector field f(x;c) [9,10,12,14,28,29].

A periodic solution (JORB=1) is a zero point of a residual map Q(x;c) [23,24]. This residual map can always be defined near the zero point. The definition of Q depends on the three values of JSYS. The symbol Q is used in all three situations to stress the anology.

A bi-periodic solution (JORB=2) is a zero point of a residual map Q(x;c) [20,21,23]. This residual map can be defined when the ratio T/T' of the two periods T and T' is either irrational or rational with a long continued fraction expansion [21]. When the continued fraction expansion of T/T' is short, the bi-periodic solution is better studied as a periodic solution (as described earlier). The definition of Q depends on the five values of JSYS. The symbol Q is used in all five situations to stress the anology.

The way in which we define the residual map Q for bi-periodic solutions can be generalized to tri-, quadra-, quinta-periodic,... solutions.

We compute the zero points with Newton-Raphson's method:

$$z_{j+1} = z_j - [Df(z_j)]^{-1} f(z_j) , \quad \text{for JORB=0,}$$

$$z_{j+1} = z_j - [DQ(z_j)]^{-1} Q(z_j) , \quad \text{for JORB=1,2,}$$

$$z_j = (x_j;c_j), \quad j=0,1,2,..., \quad z_0 \text{ is given.}$$

If c_j is constant for $j=0,1,2,3,...$ then Df resp. DQ is the jacobian. We allow c_j to vary, but then we have to introduce a bond to make Df resp. DQ a square matrix - that is done in Sec. 5. The bond is that the vectors $z_j - z_0$ must be orthogonal to the tangent of the path.

Newton-Raphson's method is stopped when $f(z_j)$ resp. $Q(z_j)$ is zero (within error level of computing accuracy), or if too many iterations have been used.

We note that

i) Newton-Raphson's method is guaranteed to converge to the equilibrium point p when we can supply a z_0 close to p.

ii) Newton-Raphson's method is said to converge, when the map (f resp. Q) was zero within computational accuracy of the map. That accuracy was determined by the algorithm in [7].

iii) Newton-Raphson's method needs the derivative Df resp. DQ of the map (f resp. Q). The derivative is computed numerically using forward differences [6,7]. The accuracy of the map influences the steplength to be used in the difference approximation.

iv) If we think of the subscript j in Newton-Raphson's method as time, then the method is itself an iterated map, and could represent some dynamical system. When the initial condition z_0 is outside the domain of guaranteed convergence, the orbit $\{z_0,z_1,z_2,...\}$ may show chaotic behaviour [17].

The package PATH [25] is an implementation of these methods.

The orbit $O(x_0;c)$ is said to be chaotic, when the orbit is very sensitive to the initial data x_0. By "very sensitive" we mean, that if u is close to x_0 then $O(u;c)$ and $O(x_0;c)$ separate from each other. As long as they are close to each other they separate exponentially. However, the orbits remain bounded in the state space X. The technical definition is that the orbit $O(x_0;c)$ has at least one positive Liapunov exponent [26,29].

We always have to supply an initial state x_0 close to the solution we want. However, for a given parameter value a nonlinear system may have more than one equilibrium point. We have not studied how to obtain all equilibrium points, but methods exist for this purpose [11,15,16].

4 Stability of Equilibrium Points of Dissipative Dynamical Systems

In this section we consider the stability of an equilibrium point p subject to small perturbations in p for a fixed value of the parameter c. If the solution $\varphi(t;p+u;c)$ is attracted to the orbit $O(p;c)$ for $t \to +\infty$ for any small perturbation u, then p is said to be stable; otherwise p is unstable.

A stationary solution p (JORB=0) of an autonomous ODE (JSYS=0) is a zero point of a vector field $f(x;c)$ for some fixed value of the control parameter c; thus $f(p;c)=0$. Near this zero point (for the fixed value of c), the linear approximation to the vector field $f(x;c)$ is

$$Df(p;c)(x-p) ,$$

where x is close to p. The zero point p is said to by hyperbolic, if all eigenvalues λ_i of $Df(p;c)$ are not lying on the imaginary axis in the complex plane [9]. In a one-parameter family of vector fields $f(x;c)$ almost all zero points are hyperbolic [10]. If all eigenvalues have negative real parts (i.e. $Re(\lambda_i) < 0$ for all λ_i) the zero point is stable; otherwise the zero point is unstable.

Periodic (JORB=1) and bi-periodic (JORB=2) solutions p are zero points of a properly defined residual map $Q(x;c)$ (JSYS=-2,-1,0,+1,+2) for some fixed value of the control parameter c; thus $Q(p;c)=0$. We define

$$P(x;c) = Q(x;c) + I(x;c) , \quad I(x;c) = x ,$$

where P is denoted the Poincaré map and I is the identity map. The equilibrium point p will then be a fixed point of P, by which we mean, that $P(p;c)=p$. Near this fixed point

(for a fixed value of c), the linear approximation to the Poincaré map $P(x;c)$ is

$$P(p;c) + DP(p;c)(x-p) ,$$

where x is close to p. The fixed point p is said to be hyperbolic if all eigenvalues λ_i of $DP(p;c)$ are not lying on the unit circle in the complex plane [9]. In a one-parameter family of Poincaré maps $P(x;c)$ almost all fixed points are hyperbolic [10]. If all eigenvalues are lying inside the unit circle (i.e. $|\lambda_i|-1 < 0$ for all λ_i) the fixed point is stable; otherwise the fixed point is unstable.

The matrices $Df(p;c)$ and $DP(p;c)$ are in PATH [25] computed numerically using forward difference approximations, cf. Sec. 3. The eigenvalues are found with the IMSL-routine EIGRF [18].

The stability analyses above are linear stability analyses since we only consider the linear approximations to the vector field resp. Poincaré map near the equilibrium point. The stability of a periodic solution of an ODE is sometimes found using the Floquet theory [19,30]. However, the Floquet theory requires the solution of the linearized variational equations, which is a system of n^2+n coupled ODEs. Furthermore the Floquet theory cannot be generalized to bi-periodic solutions.

5 Continuation of Equilibrium Points of Dissipative Dynamical Systems

A curve of equilibrium points $(p;c)$ of a dynamical system is called a path, see Fig. 5. The path can be determined with a path following method, also called a continuation method. We have used an Euler-Newton method, which is a method of the predictor-corrector type [1,5]; see Fig. 6. Details of the algorithm used can be found in [23,24]. The package PATH [25] determines a path. Note, that the tangent of the path (see Fig. 6) expresses the sensitivity of an equilibrium point p subject to small perturbations in the control parameter. Computation of the path-tangent is therefore sometimes called a sensitivity analysis.

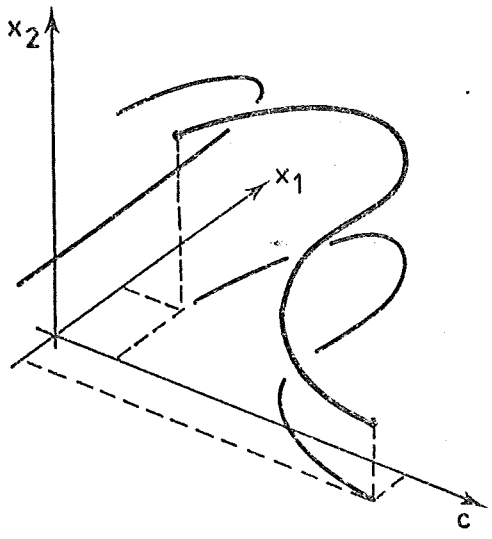


Fig. 5. Path (thick line) of stationary solutions of an autonomous ODE (where $X=R^2$). The projections (thin lines) of the path on the x_1-c-plane and on the x_1-x_2-plane are shown. The projection on the x_2-c-plane is not shown.

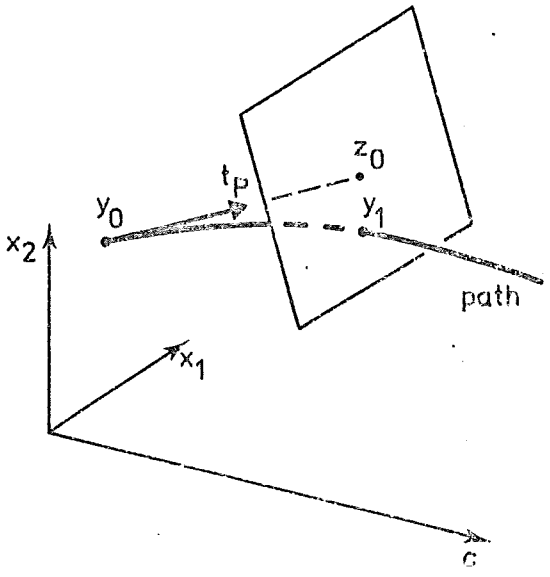


Fig. 6. One step with an Euler-Newton continuation method to follow a path of stationary solutions of an autonomous ODE. Let $y_0=(p_0;c_0)$ be an equilibrium point on the path where t_p is the tangent of the path in y_0. We take an Euler step and reach z_0. We let the hypersurface through z_0 orthogonal to t_p be the domain for the function $f(x;c)$, on which we use Newton-Raphson's method (see Sec. 3) to bring us back to the equilibrium point $y_1=(p_1;c_1)$ on the path. The distance from y_0 to y_1 is an approximation to the arclength Δs traversed.

6 Bifurcations in Dissipative Dynamical Systems

A bifurcation point is an equilibrium point where two or more paths of equilibrium points intersect each other. A necessary condition for an equilibrium point to be a bifurcation point is that the equilibrium point is not hyperbolic (Sec. 4). Since equilibrium points almost always are hyperbolic, the bifurcation points are almost always isolated. An isolated bifurcation point can be located with a continuation method in the following way. A continuation method determines a path of equilibrium points $(p;c)$, which we here will denote the basic path. We will parameterize the basic path with the arclength s along the path, thus the basic path is given by $(p(s);c(s))$. The linear stability analysis amounts to the computation of Df or DP (where D denotes differentiation with respect to the state variable x) in the points $(p(s);c(s))$, and then to the evaluation of all the eigenvalues $\lambda_i = \lambda_i(s)$ of Df or DP; see Fig. 7. Let us define

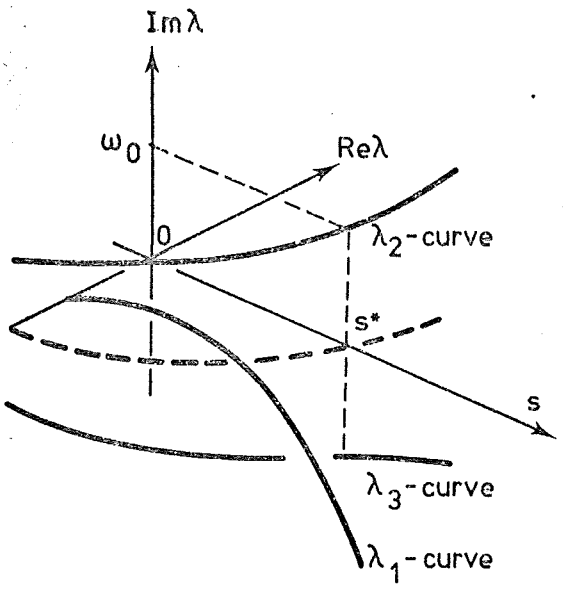


Fig. 7. A real eigenvalue (the λ_1-curve) and a complex conjugated pair or eigenvalues (the λ_2- & λ_3-curves) in dependence of the arclength s. The dashed curve is the real parts of λ_2 and λ_3. When this picture corresponds to a path of stationary solutions (JORB=0) of an autonomous ODE (JSYS=0), then a Hopf bifurcation takes place at $s=s^*$, where the critical eigenvalues are $\pm i\omega_0$. The arclength $s=s^*$ corresponds to a point $(x;c)=(x^*;c^*)$. Since $\frac{d}{ds}(\mathrm{Re}\ \lambda_2)$ in $s=s^*$ is non-zero the strict crossing condition is satisfied.

the real functions $g_i(s)$ - the stability functions - on the path:

$$g_i(s) = \mathrm{Re}(\lambda_i(s)) \quad \text{when } \lambda_i \text{ is an eigenvalue of Df,}$$
$$g_i(s) = |\lambda_i(s)|-1 \quad \text{when } \lambda_i \text{ is an eigenvalue of DP.}$$

If an $s=s^*$ exists, such that one or more of the g_i-functions are zero, then the system is non-hyperbolic and the point $(p(s^*);c(s^*))$ on the basic path is a bifurcation point. Therefore we can determine a bifurcation point on the basic path as a zero point of one of the stability functions. We have used a modified regula falsi technique to find the zero point s^* [23]. The eigenvalues $\lambda_i(s^*)$ for which $g_i(s^*)=0$ will we denote the critical eigenvalues.

We have only dealed with simple bifurcation points, i.e. bifurcation points which satisfy

SB1) the bifurcation point is isolated

SB2) there is only one real critical eigenvalue or one complex conjugated pair of critical eigenvalues

SB3) the derivative of the g_i-function corresponding to the critical eigenvalue is non-zero. This is called the strict crossing condition [19].

The simple bifurcation points are typical (generic) in one-parameter families of dynamical systems [8,10], i.e. if a bifurcation point exists in the system then it is almost always a simple bifurcation point. Some of the simple bifurcation points have been given special names:

Hopf bifurcation:

Df has a complex conjugated pair of critical eigenvalues

Saddle node bifurcation, or tangent bifurcation, or a limit point

Df has the eigenvalue 0 and $\frac{dc}{ds}(s=s^*) = 0$

DP has the eigenvalue +1 and $\frac{dc}{ds}(s=s^*) = 0$

Exchange of stability

Df has the eigenvalue 0 and $\frac{dc}{ds}(s=s^*) \neq 0$

DP has the eigenvalue +1 and $\frac{dc}{ds}(s=s^*) \neq 0$

Orbit doubling bifurcation, or pitchfork bifurcation, or flip bifurcation

DP has the eigenvalue -1

We shall first deal with the simple bifurcation points $(x^*;c^*)$ with a real eigenvalue. Corresponding to that eigenvalue there is a real eigenvector v. That eigenvector span a linear manifold E^c through $(x^*;c^*)$. v is an approximation to the tangent of the bifurcating path. However, at exchange of stability, v may be a poor approximation when the bifurcating path is two-sided [19] (one talks about a transcritical bifurcation [19]). In all other cases the bifurcating path is one-sided [19] and v is tangent of the bifurcating path.

We shall next deal with the simple bifurcation points $(x^*;c^*)$ with a complex conjugated pair of eigenvalues. Corresponding to these eigenvalues there is a complex conjugated pair of eigenvectors v and $\bar{v}$. The real part $Re(v)$ of v and the imaginary part $Im(v)$ of v span a two dimensional linear manifold E^c through $(x^*;c^*)$. All bifurcation points with a complex eigenvalue are one-sided bifurcations, so we can use any vector in E^c as tangent to the bifurcating path. A vector in E^c linearly independent to the tangent chosen is used to define the residual map of which the bifurcating path is a zero. This does also mean, that

(JORB for the bifurcating path) =
= 1 + (JORB for the basic path) .

The package PATH [25] determines the linear manifold E^c for any of the simple bifurcation points mentioned for any of the systems we can examine.

The package BIFOR2 [13] finds a second order approximation to the bifurcating path in the case of a Hopf bifurcation. The package AUTO [4] examines two-parameter families of autonomous ODEs, where it can find stationary, periodic, and homoclinic bifurcations. (We have not examined homoclinic bifurcations.) The package DERPAR [28] determines bifurcation points in one-parameter families of autonomous ODEs.

Usually we do not talk about bifurcation to chaos; instead we talk about routes to chaos [8,12,29]. These routes may involve one or more simple bifurcation points.

7 Conclusions

A railway bogie or a railway vehicle moving with high speed along a straight, horizontal, and level track may perform small lateral oscillations. Three mathematical models describing that lateral motion have been studied. The models consist of a system of nonlinear ordinary differential equations where the speed is control parameter. Details of the equations and the results obtained have been presented in [22,32,33], and will be presented in [26,27].

We have considered general nonlinear dynamical systems described by iterated maps or ordinary differential equations. In such systems we have examined stationary, periodic, and bi-periodic solutions in dependence of one control parameter, and demonstrated, that these solutions may bifurcate from each other. In order to determine the solution, to follow the solution in dependence of a parameter, and to find the bifurcations we have used computational methods. These methods have been implemented in a package PATH coded in fortran [25]. That package is available from the author. Details of the dynamical systems examined and the results obtained have been presented in [20,21,23,24].

References

[1] E. Allgower and K. Georg: Simplicial and continuation methods for approximating fixed points and solutions to systems of equations, SIAM Review, Vol.22, No.1, 1980, pp. 28-85.

[2] E. Berne: Games people play, Ballantine Books, New York, 1964.

[3] N.K. Cooperrider: The hunting behavior of conventional railway trucks, ASME J. Eng. Industry, Vol.94, 1972, pp. 752-762.

[4] E. Doedel: AUTO, Version May 1984, Computer Science Department, Concordia University, Montreal, Canada, 1984.

[5] C.B. Garcia and W.I. Zangwill: Pathways to solutions, fixed points, and equilibria, Prentice-Hall, Inc., N.J., 1981.

[6] P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright: Computing forward-difference intervals for numerical optimization, SIAM J. Sci. Stat. Comput., Vol.4, No.2, 1983, pp. 310-321.

[7] P.E. Gill, W. Murray, and M.H. Wright: Practical optimization, Academic Press Inc. (London) Ltd., 1982.

[8] R. Gilmore. Catastrophe theory for scientists and engineers, John Wiley & Sons, Inc., New York, 1981.

[9] J. Guckenheimer: A brief introduction to dynamical systems, in F.C. Hoppenstaedt (ed.): Nonlinear oscillations in biology, Am. Math. Soc., Providence, 1979.

[10] J. Guckenheimer and P. Holmes: Nonlinear oscillations, dynamical systems, and bifurcations of vector fields, Springer-Verlag: New York-Berlin-Heidelberg-Tokyo, 1983.

[11] R.S. Guttalu and C.S. Hsu: Index evaluation for nonlinear systems of order higher than two, ASME J. App. Mech., Vol.45, 1982, pp. 241-243.

[12] H. Haken: Advanced synergetics, Springer-Verlag, Berlin-Heidelberg-New York-Tokyo, 1983.

[13] B.D. Hassard, N.D. Kazarinoff, and Y-H. Wan: Theory and applications of Hopf bifurcation, London Mathematical Society Lecture Note Series no. 41, Cambridge University Press, Cambridge, 1981.

[14] M.W. Hirsch: The dynamical systems approach to differential equations, Bul. Am. Math. Soc., Vol.11, No.1, 1984, pp. 1-64.

[15] C.S. Hsu: Theory of index for dynamical systems of order higher than two, ASME J. App. Mech., Vol.47, 1980, pp. 421-427.

[16] C.S. Hsu and R.S. Guttalu: Index evaluation for dynamical systems and its applications to locating all the zeros of a vector function, ASME J. App. Mech., Vol.50, 1983, pp. 858-862.

[17] M. Hurley and C. Martin: Newton's algorithm and chaotic dynamical systems, SIAM J. Math. Anal., Vol.15, No.2, 1984, pp. 238-252.

[18] IMSL Library Edition 9, IMSL, Inc., Houston, Texas, 1982.

[19] G. Iooss and D.D. Joseph: Elementary stability and bifurcation theory, Springer-Verlag, New York-Heidelberg-Berlin, 1980.

[20] Chr. Kaas-Petersen: Computation of quasi-periodic solutions of forced dissipative systems, (to appear in J. Comput. Phys.).

[21] Chr. Kaas-Petersen: Computation of quasi-periodic solutions of forced dissipative systems II, (submitted for pub. to J. Comput. Phys., 1985).

[22] Chr. Kaas-Petersen: Dynamical behaviour of railway bogies, Proc. of EUROMECH colloquium 190, October 1-4, 1984 at Hamburg-Harburg, West Germany.

[23] Chr. Kaas-Petersen: Computation, continuation, and bifurcation of torus solutions for dissipative maps and ordinary differential equations, (submitted for pub. to Physica D, 1985).

[24] Chr. Kaas-Petersen: Continuation methods as the link between perturbation analysis and asymptotic analysis, (submitted for pub. to SIAM Review, 1985).

[25] Chr. Kaas-Petersen: PATH - User's Guide, Lab. App. Math. Phys., The Technical University of Denmark, 1985.

[26] Chr. Kaas-Petersen: Chaos in a railway bogie (in preparation for Acta Mechanica).

[27] Chr. Kaas-Petersen and H. True: Periodic, bi-periodic and chaotic dynamical behavior of railway vehicles, (to appear in Proc. of 9th IAVSD Symposium on Dynamics of Vehicles on roads and tracks, Linköping, Sweden, June 24-28, 1985).

[28] M. Kubíček and M. Marek: Computational methods in bifurcation theory and dissipative structures, Springer-Verlag, New York-Berlin-Heidelberg-Tokyo, 1983.

[29] A.J. Lichtenberg and M.A. Lieberman: Regular and stochastic motion, Springer-Verlag, New York-Heidelberg-Berlin, 1983.

[30] A.H. Nayfeh: Introduction to perturbation techniques, John Wiley & Sons, Inc., New York, 1981.

[31] L. Petzold: Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations, SIAM J. Sci. Stat. Comput., Vol.4, No.1, 1983, pp. 136-148.

[32] H. True and C. Kaas-Petersen: A bifurcation analysis of nonlinear oscillations in railway vehicles, Proc. 8th IAVSD Symposium, Vehicle System Dynamics, Swets and Zeitlinger, 1984, pp. 655-665.

[33] H. True and C. Kaas-Petersen: Quasi-periodic oscillations in railway vehicles, Xth International Conference on Nonlinear Oscillations (ICNO), September 12-17, 1984 at Varna, Bulgaria.

[34] M. Yamaguti and S. Ushiki: Chaos in numerical analysis of ordinary differential equations, Physica 3D, 1981, pp. 618-626.

Sammendrag på dansk

Svingninger kan optræde i jernbane bogier når de kører med stor fart ud ad et retlinet, perfekt og vandret spor. Disse svingninger foregår på tværs af køreretningen og giver anledning til slid på hjul og skinne og dermed store omkostninger til vedligeholdelse. Svingninger af bogien inducerer via fjedre og dæmpere svingninger af vognkassen; disse svingninger er uønskede da kørekomforten nedsættes. Svingningerne kan være periodiske, bi-periodiske (dvs. svingningerne består af to koblede periodiske svingninger) eller kaotiske (dvs. svingningerne afhænger kritisk af hvordan den anslås). Vi har benyttet Cooperrider's modeller af jernbane bogier og jernbane vogne. Modellerne er beskrevet ved sædvanlige differential ligninger. Kræfterne mellem hjulene og skinnerne antages at være ikke-lineære, at være dissipative, og at afhænge af farten. Endvidere er kræfterne mellem hjulflangerne og skinnerne antaget at være ikke-lineære. Fjeder- og dæmper-kræfterne antages at være lineære. Da kræfterne ikke afhænger explicit af tiden siges systemet af sædvanlige differential ligninger at være autonomt. Ligningerne og de opnåede resultater er blevet præsenteret i [22,32,33], og vil blive præsenteret i [26,27].

Vi har udviklet beregningsmetoder til at undersøge ligningerne ovenfor. Metoderne kan anvendes på alle autonome sædvanlige differential ligninger. Metoderne har også været benyttet til at behandle sædvanlige differential ligninger, der er periodiske eller bi-periodiske i tiden samt differens ligninger, der er periodiske eller bi-periodiske i tiden. Alle disse systemer kaldes under ét dynamiske systemer, og vi begrænser os til de dissipative dynamiske systemer.

Ud over at bestemme løsningen, finder vi løsningens stabilitet. Vi følger løsningen i afhængighed af en parameter i systemet - en såkaldt sti af løsninger - med en stifølger

metode. Løsningerne kan være stabile på en del af stien og ustabile på resten af stien. Når stabiliteten ændres, vil en ny sti af løsninger afgrene fra den sti vi følger. Sådanne afgreningspunkter kaldes bifurkations punkter og vi foretager en bifurkations analyse for at finde retningen af den afgrenende sti. Et dynamisk system kan således have mere end én sti af løsninger - en nødvendig betingelse herfor er at systemet er ikke-lineært.

Beregningsmetoderne er konstruerede til og anvendt på en regnemaskine. En regnemaskine er nødvendig når (som det er tilfældet her) de dynamiske systemer består af mange sammenhørende ligninger, er meget ikke-lineære, og skal løses over længere tidsintervaller. De dynamiske systemer vi har betragtet og de resultater vi har opnået er præsenterede i [20,21, 23,24]. Metoderne er implementerede i fortran og udgør én programpakke [25].

Summary in English

Oscillations may be seen in railway bogies when they run with high speed along a straight, perfect, and horizontal track. These oscillations take place across the running direction and cause wear on wheels and rails and therefore large maintenance costs. Oscillations of the bogie introduce through springs and dampers oscillations of the car body; these oscillations are unwanted since the passenger comfort is reduced. The oscillations can be periodic, bi-periodic (i.e. the oscillations consist of two coupled periodic oscillations), or chaotic (i.e. the oscillations depend critically on how they are triggered). We have used Cooperrider's models of railway bogies and railway vehicles. The models are described by ordinary differential equations. The forces between the wheels and the rails are assumed to be nonlinear, to be dissipative, and to depend of the running speed. Furthermore are the forces between the wheel-flanges and the rails assumed to be nonlinear. Spring- and damper-forces are assumed to be linear. Since no forces depend explicitly on time, the system of ordinary differential equations is said to be autonomous. The equations and the results we have obtained have been presented in [22,32,33], and will be presented in [26,27].

We have developed computational methods to examine the equations mentioned. The methods are applicable to all autonomous ordinary differential equations. These methods have also been used to examine ordinary differential equations periodic or bi-periodic in time and difference equations periodic or bi-periodic in time. All these systems are denoted dynamical systems, and we restrict ourselves to dissipative dynamical systems.

Besides determining the solution, we also determine the stability of the solution. We trace the solution in dependence of a parameter in the system - a socalled path of solutions - with a path following method. The solutions can be stable on a part of the path and unstable on the other part of the path. When the stability changes, a new path of solutions branches off from the path we are following. Such branch points are called bifurcation points and we perform a bifurcation analysis in order to determine the direction of the bifurcating path. A dynamical system can therefore have more than one path of solutions - a necessary condition for this is that the system is nonlinear.

The computational methods have been developed and used on a computer. A computer is necessary when the dynamical system (as is the case here) consists of many coupled equations, is strongly nonlinear, and have to be solved over long time intervals. The systems we have considered and the results we have obtained have been presented in [20,21,23,24]. The methods are implemented in a software package coded in fortran [25].

Journal of Computational Physics - 2057.

JOURNAL OF COMPUTATIONAL PHYSICS 58, 000–000 (1985)

Computation of Quasi-Periodic Solutions of Forced Dissipative Systems

CHR. KAAS-PETERSEN

Laboratory of Applied Mathematical Physics,
The Technical University of Denmark,
DK-2800Lyngby, Denmark

Received December 20, 1983; revised April 19, 1984

The problem of finding a T-periodic solution of a forced dissipative system of ordinary differential equations is most conveniently reformulated as a fixed point problem of a Poincaré map mapping the phase space at $t=0$ into the phase space at $t=T$, and the stability of the periodic response is equivalent to the stability of the fixed point. It is shown how the problem of determining quasi-periodic solutions of forced systems with two forcing frequencies may be reformulated as a fixed point problem of a new type of Poincaré map that also opens the possibility of treating quasi-periodic solutions with more than two frequencies, as well as treating bifurcating quasi-periodic solutions of autonomous equations. Three examples are considered: one in which the exact solution is known, and two others where the quasi-periodic solutions have been determined by other means.

1. INTRODUCTION

In this section we first state the problem. Second, we review a method used to solve this problem, and third we describe the new approach to the solution of the problem.

We consider a dissipative system of forced ordinary differential equations (ODE):

$$\dot{x} = f(t, x, \omega_1, \omega_2, \ldots, \omega_p)$$

$$(\dot{}) = \frac{d}{dt}(\ )$$

$$x \in R^n,\ f: R \times R^n \times R_+^p \to R^n \qquad (1.1)$$

$$n \geqslant 1,\ p \geqslant 1$$

f is sufficiently smooth.

The p frequencies $\{\omega_1, \ldots, \omega_p\}$ are fixed. It is assumed that the steady state is a quasi-periodic function q where

$$q = q(\omega_1 t, \ldots, \omega_p t) \qquad (1.2)$$

0021-9991/85 $3.00

0021-9991/85 $3.00

Copyright © 1985 by Academic Press, Inc.
All rights of reproduction in any form reserved.

and q is 2π-periodic in each of its arguments. If q shall be a true quasi-periodic function the numbers $\omega_1, \ldots, \omega_p$ must be incommensurable, which means that no vanishing linear combination $c_1\omega_1 + \cdots + c_p\omega_p$ exists with rational coefficients $c_1, \ldots, c_p$.

The number of frequencies can be reduced if $\{\omega_1, \ldots, \omega_p\}$ is not incommensurable. In our algorithm, however, we permit commensurable as well as incommensurable frequencies. For $p=1$ we have the problem of determining a periodic solution of (1.1). Here shooting methods are efficient, which try to find a solution that take on the same value in both ends of the interval $[0, T]$, where $T = 2\pi/\omega_1$.

Let us reformulate this by introducing the Poincaré map. The Poincaré map has domain in state space at $t=0$ and range in state space at $t=T$. Let $x(t)$ be the solution of (1.1) for which $x(0) = x_0$. Then the Poincaré map P maps x_0 into $x(T)$, that is, $x(T) = P(x_0)$. The periodic solution $q(t)$ is then a fixed point for the map P, that is, $q(T) = q(0)$, and the stability of the periodic solution is equivalent to the stability of the fixed point. For $p=2$ we seek a quasi-periodic solution, $q(\omega_1 t, \omega_2 t)$. The only method known to the author is due to Chua and Ushida [1]. They assume a generalized Fourier series

$$x = a_0 + \sum_{i=1}^{\infty} [a_{2i-1} \cos \nu_i t + a_{2i} \sin \nu_i t]$$

$$\nu_i = m_{1i}\omega_1 + \cdots + m_{pi}\omega_p \qquad (1.3)$$

$m_{1i}, \ldots, m_{pi}$ are integers such that $\nu_i > 0$.

They truncate the series at a certain number M, and derive a system of equations for the coefficients (vectors) $a_0, a_1, a_2, \ldots$.

We shall instead use a Poincaré map. First we have to define a stroboscopic function, s. If $x(t)$ is the solution of (1.1) with $x(0) = x_0$, we let the stroboscopic function take on the values

$$x(0), x(T_1), x(2T_1), x(3T_1), \ldots, T_1 = \frac{2\pi}{\omega_1}$$

in the points $\tau_0, \tau_1, \tau_2, \tau_3, \ldots$ where

$$\tau_k = kT_1 \text{ modulo } T_2, \qquad k = 0, 1, 2, \ldots, T_2 = \frac{2\pi}{\omega_2}.$$

We so to speak eliminate one of the 2π-periodic arguments of q.

The Poincaré map P then maps $s(0)$ $(= x(0))$ into $s(T_2)$, thus $s(T_2) = P(s(0))$. We shall in fact use a slightly different definition of τ_k which allow us to compute $s(T_2)$ by interpolation. We have found the quasi-periodic solution when

$$s(T_2) = s(0).$$

When three periods T_1, T_2, and T_3 are given, we may define a stroboscopic function s of two variables. The first variable is $\tau^{[1]} = t \bmod T_2$ and the second is $\tau^{[2]} = t \bmod T_3$. The Poincaré map then maps $s(0, 0) = x(0)$ into $s(T_2, T_3)$. We have found the quasi-periodic solution when $s(T_2, T_3) = s(0, 0)$.

2. Method

In this section we first consider the Poincaré map when one frequency ω is given and show how to use this map in obtaining periodic solutions. Next we describe the new Poincaré map which makes it possible to determine the quasi-periodic solution. In both cases the Poincaré map transforms the problem of finding periodic and quasi-periodic solutions to that of finding fixed points. In the next section we describe how to determine the stability of the fixed point. Poincaré maps for autonomous equations are well known; they are described in [4].

We first describe the Poincaré map when $p = 1$ by considering the equation

$$\dot{x} = f(t, x, \omega), \qquad x \in R^n \tag{2.1}$$

where f is periodic with period $T = 2\pi/\omega$, thus $f(t, x, \omega) = f(t + T, x, \omega)$. We want to find the T-periodic solution $q(t)$ of (2.1) for which

$$\begin{aligned} \dot{q} &= f(t, q, \omega) \\ q(0) &= q(T). \end{aligned} \tag{2.2}$$

Now let $u(0)$ be an approximation to $q(0)$.

We define the Poincaré map P as follows:

$$\begin{aligned} P&: R^n \to R^n \\ u(T) &= P(u(0)) \end{aligned} \tag{2.3}$$

where $u(t)$ is the solution of (2.1) with $u(0)$ as the initial value. What does P do? It maps a point $u(0)$ in phase space at $t = 0$ into that point $u(T)$ in phase space at $t = T$ where the trajectory through $u(0)$ ends at $t = T$. This map may not be defined in all of the phase space, but it is certainly defined in some neighbourhood of the periodic solution. The periodic solution is seen to be a fixed point of P.

The map P can be used to define a map Q as follows:

$$Q = P - I \tag{2.4}$$

where I is the identity. Of course the domain and the range of Q equal the domain and the range of P.

We see that Q maps the periodic solution into a zero. Newton's method is very useful for the purpose of finding zeros. The values of $Q(x)$ and the derivative $DQ(x)$ of $Q(x)$ needed in Newton's method are computed numerically. As an example we find the jth column of $DQ(x)$ when

$$y = Q(x).$$

We perturb x by the vector δ_j in the jth coordinate direction, compute

$$y_j = Q(x + \delta_j)$$

and use the difference approximation:

$$j\text{th column of } DQ = \frac{y_j - y}{\|\delta_j\|}.$$

When all of $DQ(x)$ has been computed the Newton method yields

$$x_{\text{new}} = x - [DQ(x)]^{-1} y, \qquad y = Q(x) \tag{2.5}$$

and x_{new} shall be used in the next iteration.

This process is repeated until $\|Q(x)\| < \varepsilon$ for some preassigned ε, or if too many iterations have been used. This could be an indication of a too poor initial guess.

Now let us take $p = 2$, and we have the quasi-periodic solution $q(\omega_1 t, \omega_2 t)$ of

$$\dot{x} = f(t, x, \omega_1, \omega_2) \tag{2.6}$$

where q is 2π-periodic in both arguments. Take $t_k = kT_1 = k \cdot 2\pi/\omega_1$, $k \in Z_+$. Define

$$\begin{aligned} \tau_k &= 1 + \hat{t}_k \qquad \text{if} \quad 0 \leqslant \hat{t}_k < \tfrac{1}{2} \\ &= \hat{t}_k \qquad \text{if} \quad \tfrac{1}{2} \leqslant \hat{t}_k < 1 \end{aligned} \tag{2.7}$$

where $\hat{t}_k = (t_k/T_2) \bmod 1$.

Then we may define a (vector) function $s(\cdot)$, $s: \{0\} \cup [\frac{1}{2}, \frac{3}{2}] \to R^n$, as follows:

$$\begin{aligned} s(0) &= q(0, 0) \\ s(\tau_k) &= q(\omega_1 t_k, \omega_2 t_k) = q\left(\omega_1 \cdot k \frac{2\pi}{\omega_1}, \omega_2 t_k\right) \\ &= q\left(0, 2\pi \frac{t_k}{T_2}\right) = q(0, 2\pi \hat{t}_k) = q(0, 2\pi\tau_k) \qquad \text{for} \quad k \geqslant 1. \end{aligned} \tag{2.8}$$

We shall call s the stroboscopic function. Equations (2.8) show that if q is quasi-periodic, then s is periodic with period 1 on any interval of length 1, so $s(0) = s(1)$. However, no k can be chosen to make $\tau_k = 1$, but let K be a finite set of k's ordered increasingly such that

$$1 - \varepsilon_q \leqslant \tau_k \leqslant 1 + \varepsilon_q, \qquad \varepsilon_q > 0, \quad k \in K \tag{2.9}$$

and then interpolate each coordinate of s on the net τ_k, $k \in K$, to obtain $s(1)$.

Let $u(0)$ be some approximation to $q(0, 0)$, and let $u(t)$ be the solution of (2.6) with $u(0)$ as initial value. Then we set

$$s(\tau_k) = u(kT_1) \qquad \text{for} \quad k \in K; \qquad s(0) = u(0). \tag{2.10}$$

We interpolate the stroboscopic function to find $s(1)$. Therefore we define the Poincaré map by

$$P: R^n \to R^n$$
$$s(1) = P(s(0)) = P(u(0)). \tag{2.11}$$

It should be added that s is not periodic, not even continuous, unless $u(0)$ is on the quasi-periodic solution. s can be made continuous if we restrict k to a subset $\tilde{K}$ of K, where τ_k is a monotonic increasing or decreasing sequence for $k \in \tilde{K}$.

The problem of finding the quasi-periodic solution is hereby reduced to the problem of finding the fixed point of (2.11).

We shall also indicate how the case $p = 3$ is treated in a similar way. This may easily be extended to higher values of p. We seek the solution $q(\omega_1 t, \omega_2 t, \omega_3 t)$ of

$$\dot{x} = f(t, x, \omega_1, \omega_2, \omega_3). \tag{2.12}$$

q is 2π-periodic in each argument. Take $t_k = k \cdot T_1$, $k \in Z_+$. Define

$$\begin{aligned} \tau_k^{[l]} &= 1 + \hat{t}_k^{[l]} \qquad \text{if} \quad 0 \leqslant \hat{t}_k^{[l]} < \tfrac{1}{2}, \\ &= \hat{t}_k^{[l]} \qquad \text{if} \quad \tfrac{1}{2} \leqslant \hat{t}_k^{[l]} < 1, \end{aligned} \qquad l = 1, 2 \tag{2.13}$$

where

$$\hat{t}_k^{[1]} = \frac{t_k}{T_2} \bmod 1$$

$$\hat{t}_k^{[2]} = \frac{t_k}{T_3} \bmod 1.$$

Equation (2.13) implies that the points $(\tau_k^{[1]}, \tau_k^{[2]})$, $k \in Z_+$, are inside a square in R^2 with center in $(1, 1)$ and side 1. When $t = t_k$, $k \in Z_+$, we consider q on this square, because the 2π-periodicity in each argument of q yields

$$q(\omega_1 t_k, \omega_2 t_k, \omega_3 t_k) = q\left(0, \frac{2\pi}{T_2} t_k, \frac{2\pi}{T_3} t_k\right) = q(0, 2\pi\tau_k^{[1]}, 2\pi\tau_k^{[2]}).$$

We define the stroboscopic (vector) function $s(\cdot, \cdot)$, $s: \{(0, 0)\} \cup [\tfrac{1}{2}, \tfrac{3}{2}] \times [\tfrac{1}{2}, \tfrac{3}{2}] \to R^n$, in the following way:

$$s(0, 0) = q(0, 0, 0)$$
$$s(\tau_k^{[1]}, \tau_k^{[2]}) = q(0, 2\pi\tau_k^{[1]}, 2\pi\tau_k^{[2]}) \qquad \text{for} \quad k \geqslant 1. \tag{2.14}$$

We see that s is periodic in each argument with period 1, so $s(0, 0) = s(1, 1)$.

Let K be a finite subset of k-values ordered increasingly such that $(\tau_k^{[1]}, \tau_k^{[2]})$ is inside a small square with center in $(1, 1)$ and side $2\varepsilon_q$. Thus

$$1 - \varepsilon_q \leqslant \tau_k^{[l]} \leqslant 1 + \varepsilon_q, \qquad l = 1, 2, \text{ for } k \in K. \tag{2.15}$$

Let $u(0)$ be some approximation to $q(0, 0, 0)$ and $u(t)$ the solution of (2.12) with $u(0)$ as initial value. Then we set

$$s(\tau_k^{[1]}, \tau_k^{[2]}) = u(kT_1) \qquad \text{for} \quad k \in K; s(0, 0) = u(0). \tag{2.16}$$

Each coordinate of the stroboscopic function is interpolated on the net $(\tau_k^{[1]}, \tau_k^{[2]})$, $k \in K$, to find $s(1, 1)$. The Poincaré map is now defined by:

$$P: R^n \to R^n$$
$$s(1, 1) = P(s(0, 0)) = P(u(0)). \tag{2.17}$$

The quasi-periodic problem is hereby again reduced to a fixed point problem.

3. Stability of the Fixed Point

Stability of periodic solutions of forced ODE with one forcing frequency is usually determined from the Floquet multipliers [2], but could equally well have been determined as the stability of the fixed point. Here we shall use the latter approach, since this also can be used in the case of quasi-periodic functions.

Let x be the fixed point of the map P, so

$$x = P(x). \tag{3.1}$$

If ε is any disturbance, then by Taylor's theorem (since f is sufficiently smooth)

$$P(x + \varepsilon) = P(x) + DP(x)\,\varepsilon + O(\|\varepsilon\|^2). \tag{3.2}$$

Let $x + \varepsilon$ be mapped into $x + \delta$, then

$$x + \delta = P(x + \varepsilon) = P(x) + DP(x)\,\varepsilon + O(\|\varepsilon\|^2). \tag{3.3}$$

Retaining only the lowest-order terms we get

$$\delta = DP(x)\,\varepsilon. \tag{3.4}$$

x is stable when any disturbance ε yields a δ such that $\|\delta\| < \|\varepsilon\|$. This is fulfilled if all eigenvalues of $DP(x)$ are inside the unit circle.

Another formulation is that P should be a contraction in a neighbourhood of x, and we can get the initial point by merely iterating the map P, starting with some x_0 near to x. This we shall call the brute force method.

Note that since $Q = P - I$ we get

$$DP = DQ + I. \tag{3.5}$$

4. Comments and Practicalities

In this section we will comment on the following issues:

(a) choice of initial conditions
(b) choice of interpolating functions
(c) choice of delta in computation of DQ
(d) applied software
(e) relation to bifurcation problems.

(a) *Choice of Initial Conditions*

Some good guess on the initial conditions must be known. This requirement arises from the nature of the Newton method. For linear ODEs the Newton method will determine the solution in one iteration, and therefore we may use any initial values, as long as the solution of the ODE can be determined numerically.

(b) *Choice of Interpolating Functions*

Here we shall make some comments on the interpolation in the case $p=2$. By construction of the Poincaré map, we have to interpolate the stroboscopic function s in some interval around 1.

The τ_k's cannot be chosen at will and the derivative of s is not available.

On the other hand we can take a smaller ε_q or include more points. To obtain a reasonable interpolation we have used natural cubic splines which are preferable when s is not continuous, and some jumps in the values of s may be observed.

(c) *Choice of Delta*

The delta is used in numerical approximation of the derivative. Let $g: R^n \to R$ be one of the functions in the map Q. Then the derivative in the direction of v, $\|v\|=1$, is

$$\frac{\partial g}{\partial v} = \frac{g(x+\delta v)-g(x)}{\delta} + O(\delta) \tag{4.1}$$

and the error is therefore proportional to δ. The statement follows from the Taylor series expansion of g. So δ must be small in some sense. But g is computed with some error, say, proportional to ε_g and the numerator of (4.1) is therefore in error proportional to ε_g, so that

$$\text{error}\left(\frac{g(x+\delta v)-g(x)}{\delta}\right) = O\left(\frac{\varepsilon_g}{\delta}\right)$$

according to which we shall choose δ large. The optimal δ should minimize the total error in $\partial g/\partial v$, that is, minimize

$$E(\delta) = \frac{\varepsilon_g}{\delta} + \delta.$$

The optimal δ is $\delta = \varepsilon_g^{1/2}$ and the error in $\partial g/\partial v$ is proportional to $\varepsilon_g^{1/2}$. We must therefore make an estimate of ε_g. First let us focus on the interpolation error. Let n_i be the number of knots in an interval of length $2 \cdot \varepsilon_q \cdot T_1$. The knots are randomly distributed. Let h be the maximal distance between two neighbouring knots. Then Kershaw [3] has shown that the maximal approximation error

$$e(\tau) \leqslant Kh^4$$

for τ in the subinterval

$$\tau_a \leqslant \tau \leqslant \tau_b$$

where $\tau_a - \min_{k\in K}\{\tau_k\} = O(h \ln h)$ and $\max_{k\in K}\{\tau_k\} - \tau_b = O(h \ln h)$ for $h \to 0$. As an approximation of h, place the n_i knots equidistant in the interval, so

$$h = \frac{2 \cdot \varepsilon_q \cdot T_1}{n_i - 1}. \tag{4.2}$$

Second, an error arises from the numerical solution of the ODE. We have not tried to estimate it. But the error will be significant when n_i is large. So when n_i is small the interpolation is error-determining, and when n_i is large the numerical solution of the ODEs is error-determining. For n_i small, ε_g is proportional to h^4 and a proper δ may be taken to be proportional to h^2, where h is given by (4.2). We have used $\delta = 10^{-4}$ based on some experimentation.

(d) *Applied Software*

All calculations were performed in double precision on an IBM 3033 machine. Solution of the ODE was performed by the IMSL-routine DVERK using 5th- and 6th-order Runge–Kutta methods with variable steplength, which is efficient when the equations are non-stiff. The local tolerance was set to 10^{-8}. The interpolation by splines was performed by the IMSL-routines ICSCCU and ICSEVU.

(e) *Relation to Bifurcation Theory*

Equation (1.1) for which we want the quasi-periodic solution can be generalized by introducing some parameter v. Thus we consider

$$\dot{x} = f(t, x, \omega_1, \ldots, \omega_p, v). \tag{4.3}$$

Then the quasi-periodic solution is also a function of v, and we can "follow" the solution as a function of v. Bifurcation then takes place when the stable solution becomes unstable as a result of one or more of the eigenvalues of DP crossing the unit circle.

In the above formulation the quasi-periodic solution was present from the very beginning. We will call it the basic solution. But quasi-periodic solutions may also arise as a result of bifurcation, for instance, in autonomous equations, where a quasi-periodic solution may bifurcate from a periodic solution. Also in a forced

system with one period T, the basic T-periodic solution may bifurcate into a quasi-periodic solution.

In order to investigate bifurcation it is necessary to determine the eigenvalues of DP. The method described in Section 2 can be modified to handle the case of quasi-periodic solutions in autonomous ODEs. This will be the subject of a forthcoming paper. A problem arises due to the fact that the two periods T_1 and T_2 are not explicitly given in the equation. All that is known (or can be computed) is the return time for the Poincaré map. The return time will in general depend on the initial condition.

EXAMPLE 1. We consider the linear differential equation (3)

$$\ddot{x}+2\alpha\dot{x}+\beta x=(\beta-2)\cos\sqrt{2}\,t-2\sqrt{2}\,\alpha\sin\sqrt{2}\,t+(\beta-1)\cos t-2\alpha\sin t$$

where α, β are real, $\alpha^2>\beta$. Thus $\omega_1=\sqrt{2}$ and $\omega_2=1$, so $T_1=\sqrt{2}\,\pi$ and $T_2=2\pi$. The complete solution is

$$x=c_1e^{r_+t}+c_2e^{r_-t}+\cos\sqrt{2}\,t+\cos t$$

where $r_\pm=-\alpha\pm\sqrt{\alpha^2-\beta}$, $\alpha^2-\beta>0$. c_1, c_2 are real constants. The transient part, which is the solution of the homogeneous equation, is

$$x_{\text{trans}}=c_1e^{r_+t}+c_2e^{r_-t}$$

and the steady-state part, which is the particular solution, is

$$q(\omega_1 t, \omega_2 t)=\cos\sqrt{2}\,t+\cos t.$$

Since this equation is linear, and the right-hand side consists of a T_1-periodic part added to a T_2-periodic part, it could have been solved using the superposition principle.

We use $n_i=4$ points in the interpolation and $\varepsilon_q=0.2$ and find that $t_k=kT_1$ for $k=4, 7, 10, 13$ will give an increasing sequence of τ_k.

For $\alpha=2$, $\beta=1$ we have $r_\pm<0$ and with $(x, \dot{x})=(0, 0)$ as initial condition we obtain the solution in one Newton iteration

$$x=1.992720023$$

$$\dot{x}=-0.0003792152.$$

It compares favourably with the exact solution $(x, \dot{x})=(2, 0)$. The eigenvalues of the Poincaré map are

$$\lambda_1=-0.00038726$$

$$\lambda_2=0 \qquad \text{(within computing accuracy)}$$

thus confirming the stability of the solution.

For $t_k=kT_1=k\sqrt{2}\,\pi$, $k=0, 1,\ldots$, the stroboscopic function is

$$s(\hat{t}_k)=\begin{bmatrix}x(\hat{t}_k)\\ \dot{x}(\hat{t}_k)\end{bmatrix}=\begin{bmatrix}1+\cos(2\pi\hat{t}_k)\\ -\sin(2\pi\hat{t}_k)\end{bmatrix}$$

where

$$\hat{t}_k=\frac{t_k}{T_2}\bmod 1.$$

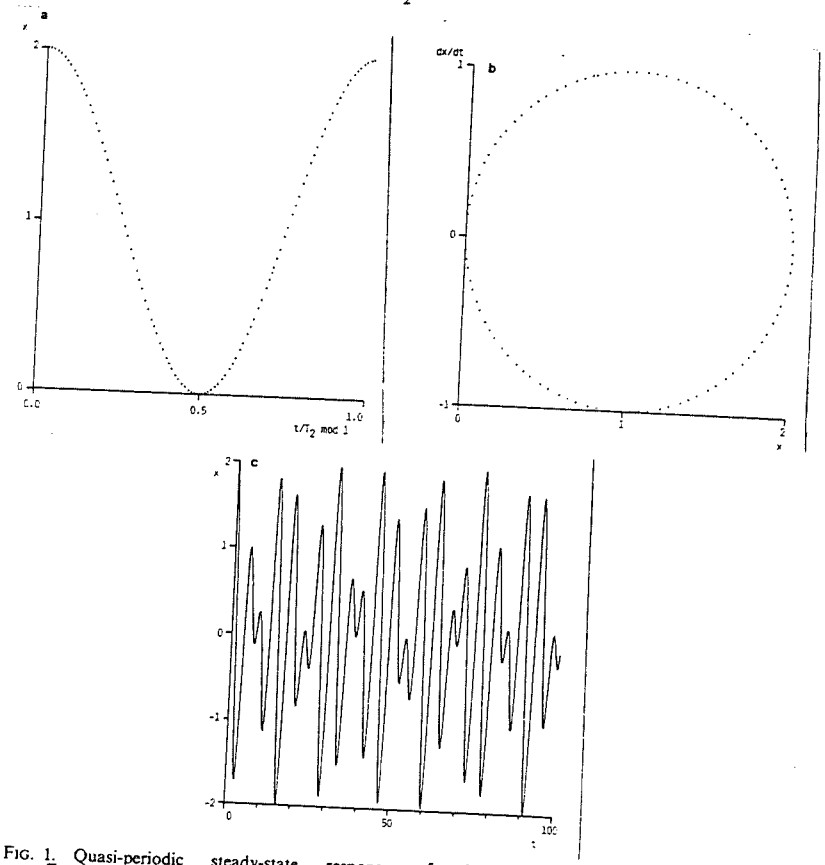


FIG. 1. Quasi-periodic steady-state response of the linear equation $\ddot{x}+4\dot{x}+x=-\cos\sqrt{2}\,t-4\sqrt{2}\sin\sqrt{2}\,t-4\sin t$. Initial conditions $(x, \dot{x})=(1.99272002, -0.0003792)$ at $t=0$. (a) First component of the stroboscopic function $s(\hat{t}_k)$ for $k=0,\ldots, 100$. (b) Second component of $s(\hat{t}_k)$ versus first component of $s(\hat{t}_k)$ for $k=0,\ldots, 100$. This we call the strobed trajectory. (c) Time evolution of x, $0\leqslant t\leqslant 100$.

In Fig. 1a we have plotted the first component of s versus $\hat{t}_k$, and in Fig. 1b we have shown $\dot{x}(\hat{t}_k)$ as a function of $x(\hat{t}_k)$. This is denoted the strobed trajectory. A plot of x as a function of time t is shown in Fig. 1c for $0 \leqslant t \leqslant 100$. For $\alpha = 1/20$ and $\beta = -1/100$ we obtain $r_{\pm} = (1 \pm \sqrt{5})/20$ and $r_+ > 0$ so we have an unstable solution. In one Newton iteration we get

$$x = 1.963136492$$

$$\dot{x} = 0.000305991$$

and the eigenvalues

$$\lambda_1 = 0.73403979$$

$$\lambda_2 = 61.71872713$$

thus confirming the instability of the solution.

EXAMPLE 2. This example is discussed in [1]. We consider the Duffing equation

$$\ddot{x} + 0.05\dot{x} + x + x^3 = 0.3 \cos t + 1.5 \cos 0.115t.$$

We see that $\omega_1 = 1.0$ and $\omega_2 = 0.115$, so that the forcing function is not a true quasi-periodic solution, since it is periodic with period $T = 200 \cdot 2\pi$. However, T is too large for use of the Poincaré map for the periodic case. Using (0, 0) as initial value the brute force method converged to the fixed point in 5 iterations using a 7-knot spline interpolation, where $K = \{8, 9, 17, 18, 26, 34, 35\}$ (so $\varepsilon_q = 0.1$).

$$(x, \dot{x}) = (1.219273582, 0.3004775330) \tag{5.1}$$

and the eigenvalues are

$$\lambda_{1,2} = -0.02604 \pm i\, 0.05943$$

using (5.1) as initial values.

(0. 0) is not close to the fixed point, and therefore Newton's method used 4 iterations, and determined

$$(x, \dot{x}) = (1.219294186, 0.3005114519). \tag{5.2}$$

Using (5.2) as initial values the eigenvalues are

$$\lambda_{1,2} = -0.02894 \pm i\, 0.05801.$$

In both cases $|\lambda_{1,2}| < 1$, so the solution is stable. Otherwise the brute force method would not converge.

The stroboscopic function, the strobed trajectory and the evolutional behaviour of x are plotted in Figs. 2a,b,c using (5.2) as initial values.

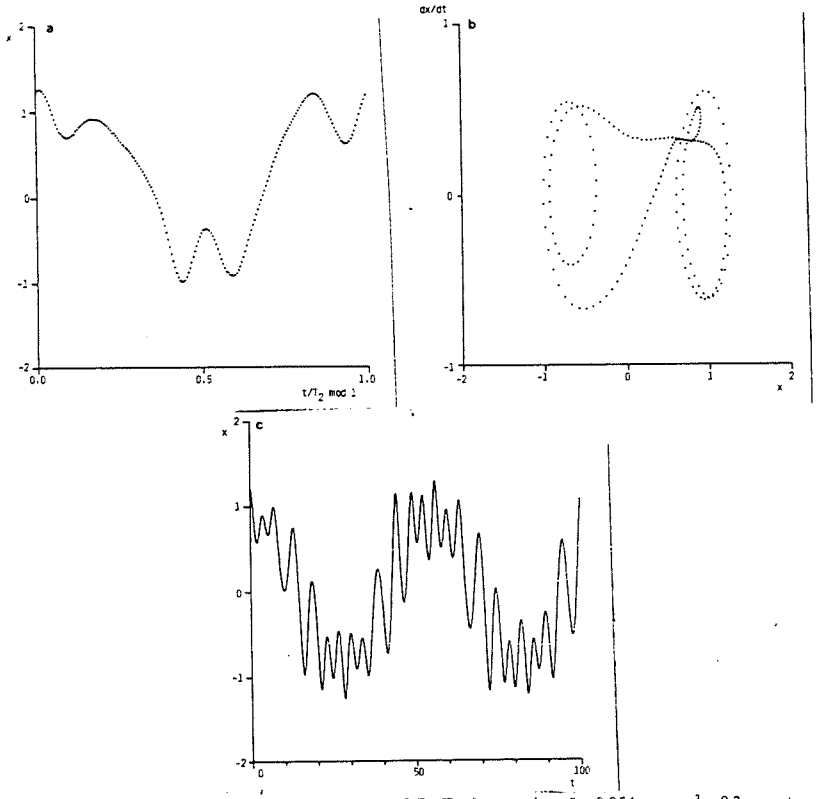


FIG. 2. Quasi-periodic steady-state response of Duffing's equation $\ddot{x} + 0.05\dot{x} + x + x^3 = 0.3 \cos t + 1.5 \cos 0.115t$. Initial conditions $(x, \dot{x}) = (1.219294, 0.30051145)$ at $t = 0$. (a) First component of stroboscopic function, i.e., $x(kT_1)$, $k = 0, 1, \ldots, 200$, $T_1 = 2\pi$, versus the independent variable, $\hat{t}_k$. (b) The strobed trajectory, $k = 0, 1, \ldots, 200$. (c) Time evolution of x, $0 \leqslant t \leqslant 100$.

In [1] Chua and Ushida obtained the initial values

$$(x, \dot{x}) = (1.21332, 0.33872)$$

which compares very well with our results.

EXAMPLE 3. This example is discussed in [1]; we have another Duffing equation

$$\ddot{x} + 0.1\dot{x} + x + x^3 = (1 + \cos 0.115t) \cos t.$$

T_1 and T_2 are as in Example 2.

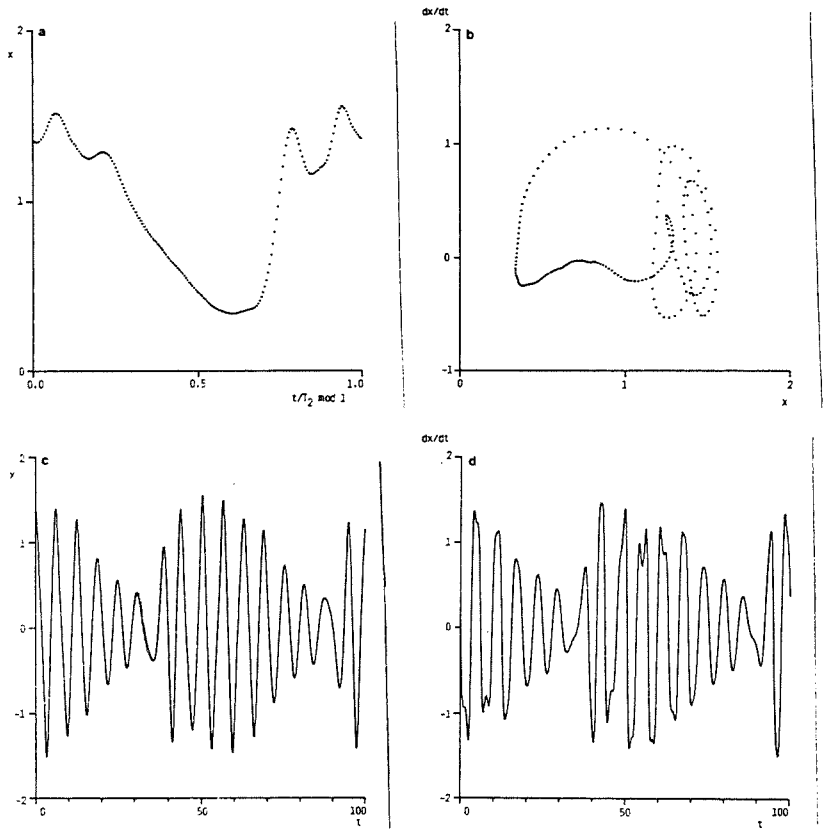


FIG. 3. Quasi-periodic steady-state response of Duffing's equation $\ddot{x}+0.1\dot{x}+x+x^3=(1+\cos 0.115t)\cos t$. Initial conditions $(x, \dot{x})=(1.354439, 0.1390661)$ at $t=0$. (a) First component of stroboscopic function, $k=0, 1,\ldots, 200$. (b) The strobed trajectory, $k=0, 1,\ldots, 200$. (c) Time evolution of x, $0\leqslant t\leqslant 100$. (d) Time evolution of $\dot{x}$, $0\leqslant t\leqslant 100$.

Using $\tilde{K}=\{8, 17, 26, 35\}$ (so $\varepsilon_q=0.2$) and taking $(x, \dot{x})=(0, 0)$ as initial point, Newton's method reaches the initial values as the steady-state solution in two iterations

$$(x, \dot{x})=(1.3494521, 0.12085215).$$

The eigenvalues of the Poincaré map were found to be

$$\lambda_{1,2}=0.0018798\pm i\,0.0012263.$$

If we instead use $K=\{1, 7, 8, 9, 10, 16, 17, 18, 19, 25, 26, 27, 34, 35, 36\}$, $\varepsilon_q=0.2$, we obtain

$$(x, \dot{x})=(1.354439, 0.1390661)$$

in two Newton iterations.

The plots in Fig. 3 have been produced with this initial condition.

In [1] Chua and Ushida found

$$(x, \dot{x})=(1.35403, 0.15168)$$

which compares favourably with our result.

CONCLUSIONS

The Poincaré map, which has already proved useful for the determination of periodic solutions of autonomous ODEs, is also applicable to the determination of periodic solutions of forced ODEs with one forcing frequency. Its applicability is due to the fact that only one point on the trajectory is considered. In this way the periodic solution becomes a fixed point in phase space, and the stability of a fixed point is equivalent to the stability of the periodic solution. Furthermore the method permits us to calculate stable and unstable solutions with equal ease. In this article these merits are extended to the case of quasi-periodic solutions.

Generalized Fourier series can be used to find quasi-periodic solutions as well as Fourier series can be used to find periodic solutions. In neither case, however, can they be used to determine the stability of the solutions.

In addition the Poincaré map yields a clear picture of important qualitative aspects of the quasi-periodic solutions.

ACKNOWLEDGMENT

Sincere thanks to dr. Hans True for discussions and encouragement during the period of work.

REFERENCES

1. L. O. CHUA AND A. USHIDA, *IEEE Trans. Circuits and Systems* CAS-28 (1981), 953–971.
2. G. IOOSS AND D. D. JOSEPH, "Elementary Stability and Bifurcation Theory," Springer-Verlag, New York/Berlin, 1980.
3. D. KERSHAW, *SIAM J. Numer. Anal.* 8 (1) (1971), 67–74.
4. J. E. MARSDEN AND M. F. MCCRACKEN, "The Hopf Bifurcation and Its Applications," Springer-Verlag, New York/Berlin, 1976.

Computation of Quasi-Periodic Solutions of Forced Dissipative Systems II

21

by

Chr. Kaas-Petersen,
Laboratory of Applied Mathematical Physics,
The Technical University of Denmark,
DK-2800 Lyngby. Denmark.

This manuscript consists of 17 pages (text, table, and figure captions) and 4 pages of drawings.

This is a COPY of the original

Submitted for publication to Journal of Computational Physics, date 85.04.01.
AMS Classification 58F & 70K .

Running title: QUASI-PERIODIC SOLUTIONS

Chr. Kaas-Petersen
Laboratory of Applied Mathematical Physics
Building 303,
The Technical University of Denmark,
DK-2800 Lyngby,
Denmark.

ABSTRACT

Quasi-periodicity with two periods we call bi-periodicity. We examine bi-periodic solutions of bi-periodically forced dissipative systems. The systems are described either by ordinary differential equations or by difference equations (iterated maps). When the ratio between the two periods is irrational or rational with many terms in its continued fraction expansion we can find the bi-periodic solution as a fixed point of a Poincaré map. The Poincaré map used to study bi-periodic solutions is a generalization of the Poincaré map used to study periodic solutions. We study a linear difference equation where the exact solution is known. Next we study a non-linear ordinary differential equation of the Duffing type, where we perform a continuation in the coefficient of the cubic term.

INTRODUCTION

We consider dynamical systems which are forced bi-periodically. The dynamic is assumed to be governed either by ordinary differential equations (ODEs) or difference equations (also called iterated maps (IMs)); thus

$$dx/dt = \dot{x} = f(t,t,x) \ , \quad t \geq 0 \qquad , \ x \in R^n \ , \ n \geq 1$$

or

$$x(t+1) = \overset{!}{x} = f(t,t,x) \ , \quad t = 0,1,2,\cdots \ , \ x \in R^n \ , \ n \geq 1$$

where f is known, and

$$f: R_+ \times R_+ \times R^n \to R^n$$

$$f(\theta + T, \quad \theta' \quad , x) = f(\theta,\theta',x)$$

$$f(\quad \theta \quad , \theta' + T', x) = f(\theta,\theta',x)$$

for all θ,θ', and x; T and T' are known periods.

Thus f is bi-periodic. We assume f to be smooth. We assume the system to be dissipative, and to have a bi-periodic orbit; thus

$$x = b(t,t) \ , \ 0 \leq t$$

where

$$b: \ R_+ \times R_+ \to R^n$$

$$b(\theta + T,\theta') = b(\theta,\theta' + T') = b(\theta,\theta') \ , \text{ for all } \theta,\theta'$$

is a solution of the governing equations.

In order to find b(t,t), we note, that $b(0,\tau)$ is a periodic function. We define a Poincaré map P which has $b(0,\tau)$ as a fixed point [6,10]. In order to obtain $b(0,\tau)$ we observe b(t,t) at times t = 0, T, 2T, 3T, ... and define $\tau = t/T' \pmod 1$. The linearized stability of the fixed point is determined from the eigenvalues of the derivative of the Poincaré map DP in that fixed point.

The theory of linear difference equations is equivalent to the theory of linear ordinary differential equations. This is demonstrated in [2]. We

stress this equivalence by denoting the discrete times 0,1,2,3, ... with t, as well as by introducing the symbol ('): $\dot{x}$ denotes the x-vector one time-step later.

Numerical solution methods for ODEs are based on difference equations [7]. Thus an ODE-solver transforms the dynamical system continuous in time into a dynamical system discrete in time.

The literature on bi-periodically forced ODEs is large. In [1,3,10] an approximation to the bi-periodic solution is performed using a generalized Fourier series. There the problem of small denominators arise. It turns out that our method do not have small denominators. In [11] perturbation methods are applied to multifrequency excitations of ODEs. On the other hand the literature on bi-periodically forced iterated maps is sparse. In [12] a linear system of ODEs forced with Dirac delta-functions have been reformulated as an iterated map.

The method discussed here was first announced in [8]. However, in that paper the following two questions were only vaguely answered:

i) When can we define a Poincaré map?

ii) How accurately can we compute the fixed point of the Poincaré map?

The answer of i) is, that a Poincaré map can be defined when the ratio T/T' is irrational, or rational with a long continued fraction expansion. In order to answer ii) we consider systems where the exact solution is known, and we use the algorithms in [4].

Bi-periodic solutions of both periodically forced systems of ODEs and IMs have been reported in [9]. The difficulty in these cases is T' is unknown. Bi-periodic solutions of autonomous ODEs was also studied in [9]; in that case both T and T' are unknown.

With two examples we demonstrate the application of the theory; an IM with known solution, and an ODE where the solution is known for one value of the parameter.

This paper is computational in spirit, and our statements are based on numerical evidence.

THE METHOD

The system of ordinary differential equations

$$\dot{x} = f(t,t,x) \ , \quad x(0) = x_0 \ , \quad x \in R^n \ , \quad 0 \leq t$$

defines an orbit $O(x_0)$ described by the solution curve $\phi(t;x_0)$, $0 \leq t$, where $\phi(0;x_0) = x_0$. Similarly the system of iterated maps

$$x' = f(t,t,x) \ , \ x(0) = x_0 \ , \ x \in R^n \ , \ t = 0,1,2, \ \ldots$$

defines an orbit $O(x_0)$ starting in x_0.

The right hand sides are bi-periodic functions, thus

$$f : R_+ \times R_+ \times R^n \to R^n \ ,$$

$$f(\theta + T,\theta',x) = f(\theta,\theta' + T',x) = f(\theta,\theta',x) \quad \text{for all } \theta,\theta' \ .$$

We assume $0 < T < T'$. This can be assumed without loss of generality in the ODEs. However, for the IMs, T must be an integer and we may then take kT', an integer multiple of T', for the second period to satisfy $0 < T < kT'$.

We assume that the systems have a bi-periodic orbit described by $b(t,t)$, $b(0,0) = x_b$. Thus

$$b : R_+ \times R_+ \to R^n \ ,$$

$$\phi(t;x_b) = b(t,t) \quad \text{with} \quad \phi(0;x_b) = b(0,0) = x_b \ .$$

$$b(\theta + T,\theta') = b(\theta,\theta' + T') = b(\theta,\theta') \quad , \quad 0 < T < T' \ , \ \text{for all } \theta,\theta'.$$

For a moment, let us assume we know the solution x_b. If we look on the solution ϕ at times T, 2T, 3T, ..., jT, ... we have

$$\phi(jT;x_b) = b(jT,jT) = b(0,jT) = b(0,j \cdot \frac{T}{T'} \cdot T') \ .$$

We define the winding number $w = T/T'$, $0 < w < 1$. Then

$$\phi(jT;x_b) = b(0,jw \cdot T') = b(0,\tau_j T') \ ,$$

where $\tau_j = jw \ (\text{mod }]-\frac{1}{2}, \frac{1}{2}])$.

Let J be a set of integers (which we determine in the next section),

such that τ_j, $j \in J$ are in a small interval around zero. We interpolate between the points $\phi(jT;x_b)$ parameterized by τ_j, where $j \in J$. We can then evaluate the interpolating curve in $\tau = 0$. Within computing accuracy the interpolating curve is in the point x_b.

The procedure above can be applied to the situation where x_0 is in some neighbourhood of x_b. Interpolation between the points $\phi(jT;x_0)$, $j \in J$ yields the point $p \in R^n$. We define $q = p - x_0$, and denote it the residual vector.

We now define the residual map Q which to any input x_0 yields the output q, i.e.

$$Q : R^n \to R^n , \quad Q(x_0) = q .$$

The bi-periodic solution x_b is then a zero point of the residual map.

We also define the Poincaré map P :

$$P : R^n \to R^n , \quad P(x_0) = p .$$

The bi-periodic solution x_b is a fixed point of P. The Poincaré map defined here is a generalization of the Poincaré map or return map used in the study of periodic solutions [10]. The linearized stability of the bi-periodic solution is equivalent to the linearized stability of the fixed point of P.

The set of points $\Sigma(x_0) = \{\phi(jT;x_0) \in R^n : j = 0,1,2, \ldots\}$ is denoted the strobed orbit, and each point is denoted a strobe. We have $\Sigma(x_0) \subset O(x_0)$. On the bi-periodic solution $\Sigma(x_b)$ is a point set on a closed curve σ. If w is irrational, then $\Sigma(x_b)$ is dense on σ. If w is rational with a long continued fraction expansion, then the points $\Sigma(x_b)$ are sufficiently close for our use. σ may cross itself in double points and is therefore not invariant.

THE WINDING NUMBER

We define the winding number as the ratio between the two periods

$$w = T/T' \quad , \quad 0 < w < 1 \ .$$

The continued fraction expansion of w is written [1,5,10]

$$w = [w_1, w_2, w_3, \ldots] = \cfrac{1}{w_1 + \cfrac{1}{w_2 + \ldots}}$$

where w_i are positive numbers. When this expansion is truncated at w_k then we obtain the rational numbers

$$\frac{r_k}{s_k} = [w_1, w_2, \ldots, w_k] \quad , \quad k = 1,2,3, \ \ldots \ ,$$

which we denote the successive convergents to w [5]. The numbers can be computed recursively by

$$\left.\begin{aligned} r_k &= w_k \cdot r_{k-1} + r_{k-2} \\ s_k &= w_k \cdot s_{k-1} + s_{k-2} \end{aligned}\right\} \quad k = 2,3, \ \ldots \ ,$$

with

$$r_1 = 1 \quad , \quad r_0 = 0$$

$$s_1 = w_1 \quad , \quad s_0 = 1$$

The sequence of successive convergents alternate about w and converges to w [10]. Thus the distances

$$d_k = s_k w - r_k \quad , \quad k = 1,2,3,\ldots$$

alternate about zero and converge to zero.

For any w we can determine a sequence of numbers n_ℓ, $\ell = 1,2,3, \ \ldots$ such that

$$\delta_\ell < 0 < \delta_\ell + w$$

$$\delta_\ell = n_\ell \cdot w - \ell \quad , \quad \ell = 1,2,3,\ldots$$

When d_k is positive for a chosen s_k, let us find a n_ℓ for which $-\frac{1}{2} < \delta_\ell < 0$, and let m satisfy $0 < \delta_\ell + md_k < \frac{1}{2}$. Then the set of integers

$$J = \{ n_\ell , n_\ell + s_k , \ldots, n_\ell + ms_k \}$$

leads to a set of strobes $S \subset \Sigma(x_0)$,

$$S = \{\phi(jT; x_0) \in R^n : j \in J\}$$

which can be parameterized by $\tau_j = jw \pmod{]-\frac{1}{2}, \frac{1}{2}]}$, $j \in J$. The point on the interpolating curve at $\tau = 0$ we denote p , $p \in R^n$.

When d_k is negative for a chosen s_k, let us find a n_ℓ for which $0 < \delta_\ell + w < \frac{1}{2}$, and let m satisfy $-\frac{1}{2} < \delta_\ell + w + md_k < 0$. Then the set

$$J = \{ n_\ell + 1 , n_\ell + 1 + s_k , \ldots, n_\ell + 1 + ms_k \}$$

can be used to obtain a set of strobes.

If the continued fraction of w breaks off after a certain w_k, then $w = r_k/s_k$. We may either use $J = \{s_k\}$ - and thus study the bi-periodic solution as a periodic solution with period $s_k T$ - or we can determine a set J by the procedure above.

The computational work is essentially equal to the work in computing $\phi(jT;x_0)$ where $j = \max J$. If therefore $1 \ll w_k$ for a certain k, then $s_{k-1} \ll s_k$ and since the set J is based on s_k, the elements of J may become too large for practical computations. A large w_k means that the rational number r_k/s_k is very close to w. We want small values of w_k to have flexibility in the definition of J. Therefore the slower r_k/s_k converge to w, the easier we may find J. The slowest convergence of r_k/s_k will take place for the winding number $[1,1,1,\ldots] = \dfrac{2}{\sqrt{5}-1}$ = the golden mean inverse, where the successive convergents are ratios of the Fibonacci numbers 1,1,2,3,5,8,... each being the sum of its two predecessors.

NUMERICAL DETAILS

The interpolation was done with cubic splines. We used the IMSL-routines ICSCCU to set up the interpolating curve, and ICSEVU to evaluate the curve at $\tau = 0$.

The accuracy of the values of Q was determined with the algorithm in [4]. We called x_0 a zero point of Q when the norm of $Q(x_0)$ was below that accuracy. Then the initial condition on the bi-periodic solution is $x_b = x_0$.

The zero of the residual map Q was found with Newton-Raphson's method. The derivative DQ was approximated with forward differences. The steplength in the difference was found with the algorithm in [4].

The stability of the fixed point x_b of P was determined from the eigenvalues of $DP(x_b) = (DQ + I)(x_b)$. When all eigenvalues are inside the unit circle in the complex plane, the fixed point is stable. The eigenvalues λ_i of DP depend on the values of (n_ℓ, s_k, m). However, if $|\lambda_i| < 1$ for one set (n_ℓ, s_k, m) then λ_i will be inside the unit circle for any other (n_ℓ, s_k, m)-set. Similarly for $|\lambda_i| > 1$.

The governing system of equations may depend on a control parameter. Thus for the ODEs we have

$$\dot{x} = f(t,t,x;c) \quad , \quad c \in R \ .$$

Then the zero point x_b of the residual map depends on c. The path of zeros is implicitly defined, and can be followed from any given point on the path. This will be demonstrated in the example with Duffing's equation.

AN EXAMPLE WITH AN ITERATED MAP

We shall study the linear difference equation

$$\ddot{u} + \dot{u} + \frac{3}{4}u = \cos(\sqrt{2}\,\pi\, t) \ ,$$

$$\ddot{u} = u(t+2) \ , \quad \dot{u} = u(t+1) \ , \quad u = u(t) \ , \quad t = 0,1,2,\ldots .$$

We use the substitution $x_1 = u$, $x_2 = \dot{u}$, and obtain the IM

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_2 - \frac{3}{4}\, x_1 + \cos(\sqrt{2}\,\pi\, t) \ . \end{cases}$$

The determinant of the jacobian of the right hand side is $\frac{3}{4}$, thus the system is dissipative.

We see, that $T = 1$ and $T' = \sqrt{2}$, thus the winding number is $w = 1/\sqrt{2}$ with the continued fraction expansion $w = [1,2,2,2,\ldots]$. We choose $k = 6$ for which $r_k = 70$, $s_k = 99$, such that $d_k = 3.57133 \cdot 10^{-3}$. Then we fix $\ell = 29$ for which $n_\ell = 41$ and $\delta_\ell = -8.62197 \cdot 10^{-3}$. With $m = 4$ we have $J = \{41,140,239,338,437\}$. When the norm of the residual map Q was below 10^{-9} we stopped the Newton scheme. We found the zero point of Q to be

$$x_1 = u(0) = -1.09085\ 67897\ 37628$$

$$x_2 = \dot{u}(0) = \ \ 1.55572\ 71067\ 22602 \ .$$

The eigenvalues of $DP = DQ + I$ in this point were inside the unit circle. Thus the bi-periodic solution is stable. A picture of the solution is seen in Fig. 1.

The difference equation can be solved exactly [2]. The solution is $u(t) = A(-\frac{1}{2} + i\frac{\sqrt{2}}{2})^t + B(-\frac{1}{2} - i\frac{\sqrt{2}}{2})^t + C\cos(\sqrt{2}\,\pi\, t) + D\sin(\sqrt{2}\,\pi t)$. Here $i = \sqrt{-1}$; A and B are arbitrary real constants, whereas C and D are the solutions of the 2×2 system

$$\begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \ ,$$

$$\alpha = \cos(2\pi\sqrt{2}) + \cos(\pi\sqrt{2}) + \frac{3}{4} \ ,$$

$$\beta = \sin(2\pi\sqrt{2}) + \sin(\pi\sqrt{2}) \ .$$

The solution is bi-periodic when $A = B = 0$, and because $\left| -\frac{1}{2} \pm i\frac{\sqrt{2}}{2} \right| < 1$ the solution is stable. We have

$$u(0) = C = -1.09085\ \ 67916\ \ 34538$$

$$\dot{u}(0) = C\cos(\sqrt{2}\pi) + D\sin(\sqrt{2}\pi) = 1.55572\ \ 71093\ \ 23350\ ,$$

which compares very well with our computations above.

AN EXAMPLE WITH AN ORDINARY DIFFERENTIAL EQUATION

We shall study an equation of Duffing's type

$$\ddot{x} + 0.05\dot{x} + x + cx^3 = 0.3\cos t + 1.5\cos(0.115\,t),$$

where c is a control parameter. The case $c = 1$ was examined in [8] originally due to Chua and Ushida [3]. The case $c = 0$ can be solved exactly. The transient is damped, and the steady state solution is

$$x(t) = 6\sin t + A\cos(0.115\,t) + B\sin(0.115\,t)\ ,\quad t \geq 0\ .$$

This solution is bi-periodic, and A and B are the solutions of the 2×2 system

$$\begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix}\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.0 \end{bmatrix}\ ,$$

$$\alpha = 1 - 0.115^2$$

$$\beta = 0.05 \cdot 0.115\ .$$

We obtain

$$x(0) = \qquad A = \ \ 1.52005\ 17541\ 85237$$

$$\dot{x}(0) = 6 + 0.115\ B = \ \ 6.00101\ 86052\ 77247\ .$$

We use the substitution $y_1 = x$, $y_2 = \dot{x}$, and obtain the system

$$\begin{cases} \dot{y}_1 = y_2 \\ \dot{y}_2 = -0.05y_2 - y_1 - cy_1^3 + 0.3\cos t + 1.5\cos(0.115\,t)\ . \end{cases}$$

The divergence of the right hand side is -0.05 , thus the system is dissipative. The two periods are $T = 2\pi$ and $T' = 2\pi/0.115$. Therefore the winding number is

$w = 0.115$, whose continued fraction expansion is $[8,1,2,3,2]$. The successive convergents are $\frac{1}{8}$, $\frac{1}{9}$, $\frac{3}{26}$, $\frac{10}{87}$, and $\frac{23}{200}$. From this we see, that the right hand side is periodic with period $200 \cdot 2\pi$. We choose $k = 2$ and $\ell = 3$ such that $s_k = 9$ and $n_\ell = 17$. With $m = 3$ we then obtain $J = \{17,26,35,44\}$.

We have performed a continuation in the parameter c . The initial values for the bi-periodic solutions are given in Table I. The corresponding strobed orbits can be seen in Fig. 2. The influence of the cubic non-linearity is seen to be very strong. In all cases the solutions are stable.

We used the IMSL-routine DVERK to solve the ODEs with a local error tolerance of 10^{-8}. When the norm of the residual map Q was below 10^{-7} the point was accepted as a zero point. The steplength used in the computation of DQ was $\sqrt{10^{-7}}$. The algorithm in [4] indicated, that $4 \cdot 10^{-7}$ was an optimal steplength. The reason may be, that the solution is very stable, i.e. all eigenvalues are close to zero.

CONCLUSIONS

Given a bi-periodically forced system. When such a system have a bi-periodic solution, we have described a method to determine it. The method assumes the ratio of the two periods T and T' to be either irrational or rational with a long continued fraction expansion. If the continued fraction expansion is short the solution is better studied as a periodic solution. We have examined a system of IMs where we could compare our method with the exact solution. We have also examined a system of ODEs where other methods have been used earlier. In both cases with good results.

REFERENCES

[1] V.I. Arnold: Geometrical methods in the theory of ordinary differential equations, Springer-Verlag, New York-Heidelberg-Berlin, 1983.

[2] C.M. Bender and S.A. Orszag: Advanced mathematical methods for scientists and engineers, McGraw-Hill, 1978.

[3] L.O. Chua and A. Ushida: Algorithms for computing almost periodic steady-state response of non-linear systems to multiple input frequencies, IEEE Tansactions on circuits and systems, Vol. cas-28, No. 10, 1981, pp. 953-971.

[4] P.E. Gill, W. Murray, and M.H. Wright: Practical optimization, Academic Press, 1981.

[5] H.B. Griffiths and P.J. Hilton: A comprehensive textbook of classical mathematics, Springer-Verlag, New York-Heidelberg-Berlin, 1970.

[6] J. Guckenheimer and P. Holmes: Nonlinear oscillations, dynamical systems, and bifurcations of vector fields, Springer-Verlag, New York-Berlin-Heidelberg-Tokyo, 1983.

[7] P. Henrici: Discrete variable methods in ordinary differential equations, Wiley, 1962.

[8] Chr. Kaas-Petersen: Computation of quasi-periodic solutions of forced dissipative systems, J. Comput. Phys. (to appear).

[9] Chr. Kaas-Petersen: Computation, continuation, and bifurcation of torus solutions for dissipative maps and ordinary differential equations, (submitted to Physica D).

[10] A.J. Lichtenberg and M.A. Lieberman: Regular and stochastic motion, Springer-Verlag, New York-Heidelberg-Berlin, 1983.

[11] A.H. Nayfeh and D.T. Mook: Nonlinear oscillations, Wiley, 1979.

[12] A. Salat and J. Tataronis: Stochasticity and order in a linear quasi-periodic differential equation, in C.W. Horton jr., L.E. Reichl, and V.G. Szebehely (eds.): Long-time prediction in dynamics, Wiley, 1983.

Table I.

Initial conditions $(x,\dot{x})$ at t = 0 for bi-periodic orbits

c	x	$\dot{x}$
0.0	1.5199 84759 91978 1	6.0010 18801 48837 8
0.01	3.9109 25574 43250 1	1.5296 40281 25913 2
0.05	2.8089 25069 83854 5	0.4044 34179 38561 66
0.1	2.4496 99465 33074 2	0.0847 64578 43474 747
0.5	0.8456 97528 20074 02	-0.0482 33694 28735 620
1.0	1.2018 82039 60392 9	0.2822 35981 73299 29

FIGURE CAPTIONS

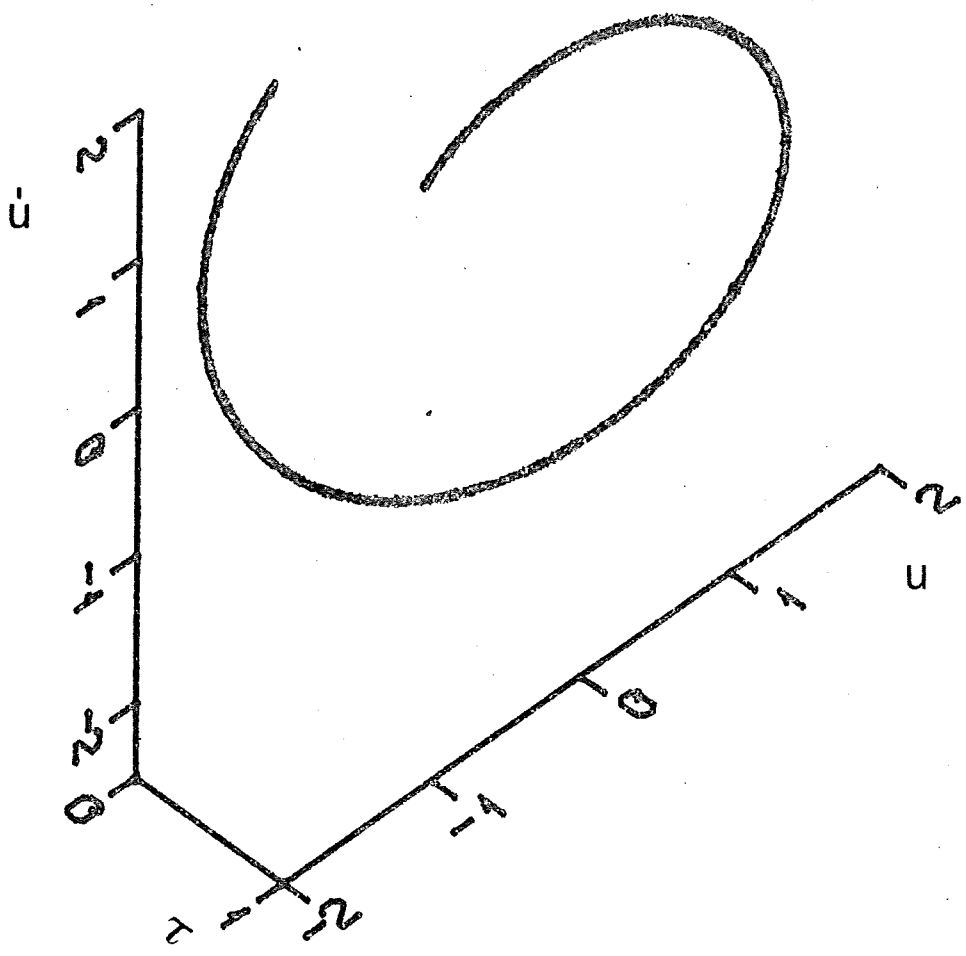
Figure 1. Steady state solution of the difference equation $\ddot{u} + \dot{u} + \frac{3}{4}u = \cos(t\pi\sqrt{2})$. The initial condition at $t = 0$ is $(u,\dot{u}) = (-1.09085\ 67897\ 37628\ ,\ 1.55572\ 71067\ 22602)$. The τ-axis is $\tau = t/T' \pmod{]0,1]}$, $t = 0,1,2,\ldots,500$, and $T' = \sqrt{2}$.

Figure 2. Strobed orbits of the bi-periodic steady state solutions of $\ddot{x} + 0.05\,\dot{x} + x + cx^3 = 0.3\cos t + 1.5\cos(0.115\,t)$ for different values of c . The x and $\dot{x}$ axes have different scales, whereas the τ-axis is $\tau = t/T' \pmod{]0,1]}$, $t = 0,\ 2\pi,\ 2\cdot 2\pi,\ldots,\ 200\cdot 2\pi$, and $T' = 2\pi/0.115$.

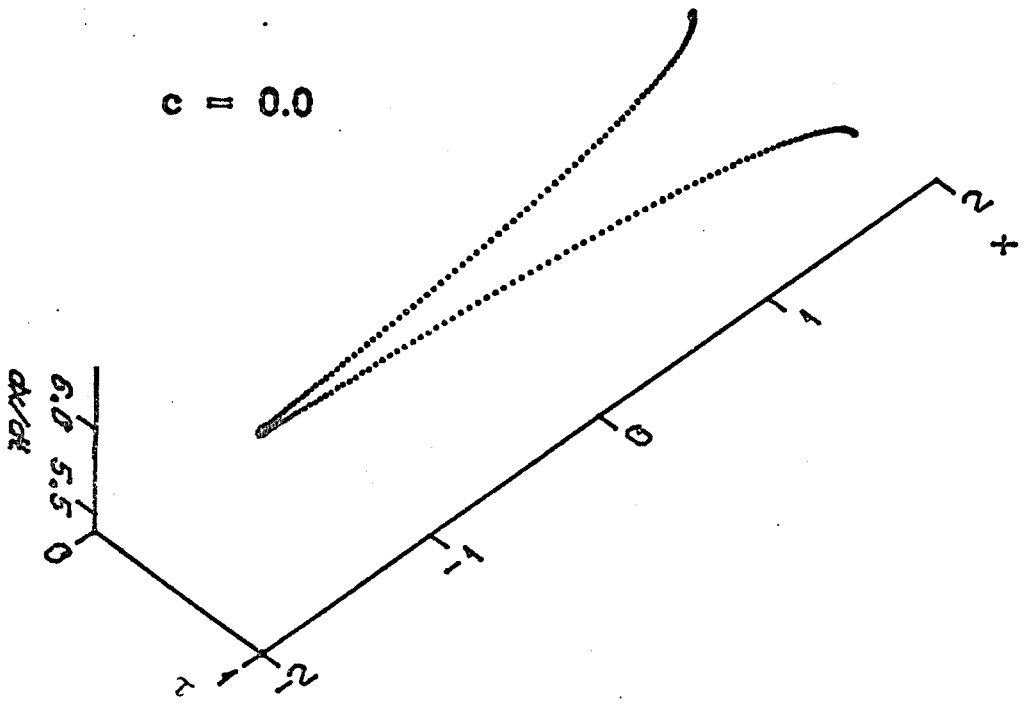
Journal of Computational Physics

Chr. Kaas-Petersen

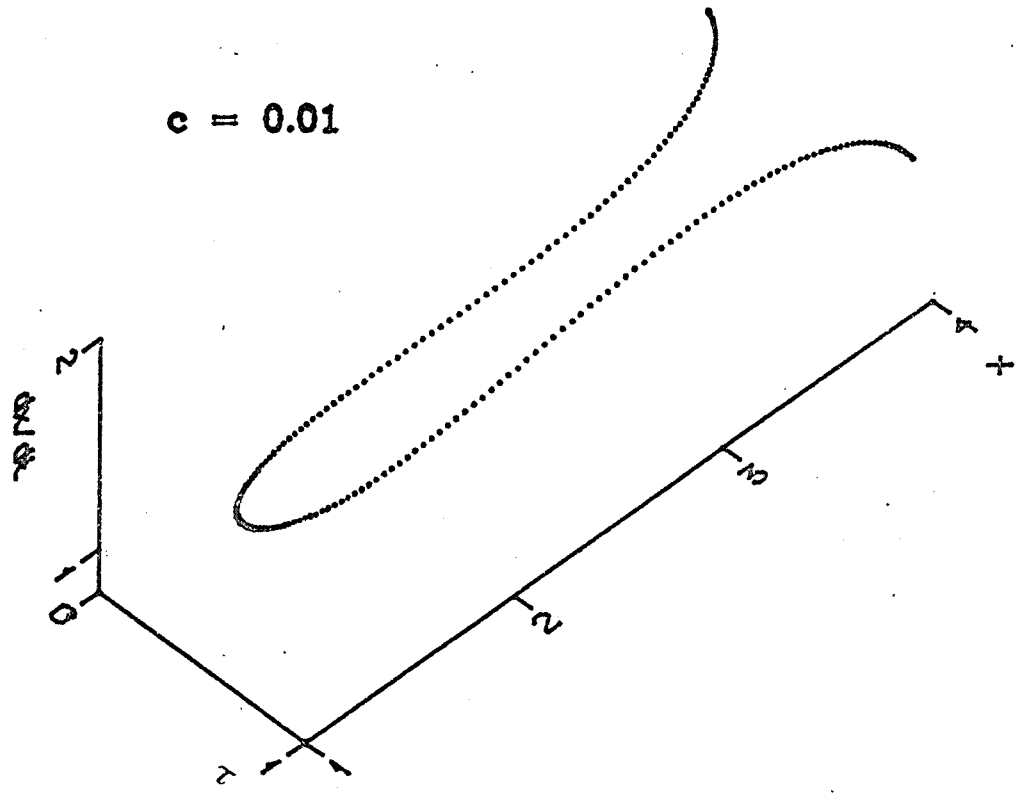
Figure 1



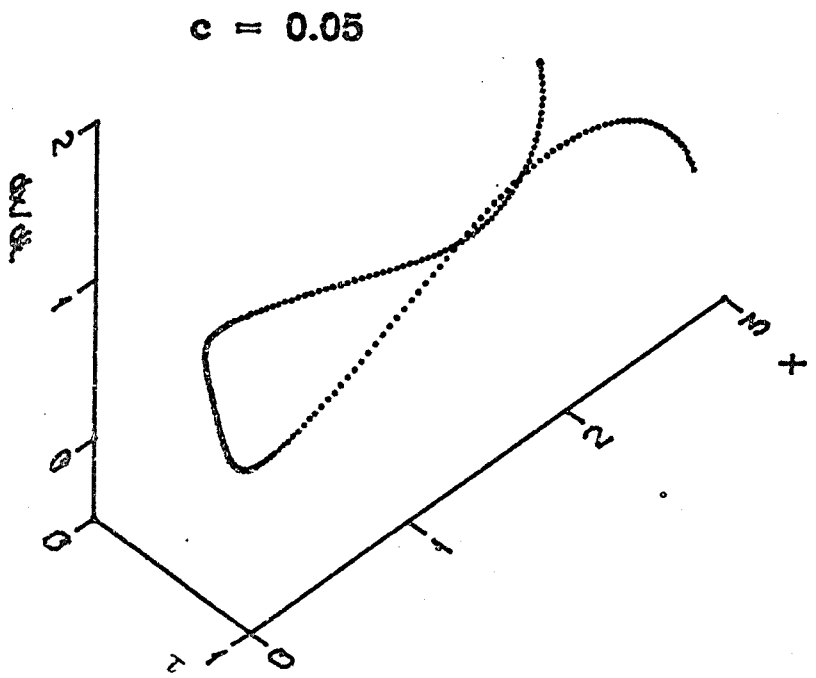
Journal of Computational Physics
Chr. Kaas-Petersen
Figure 2a



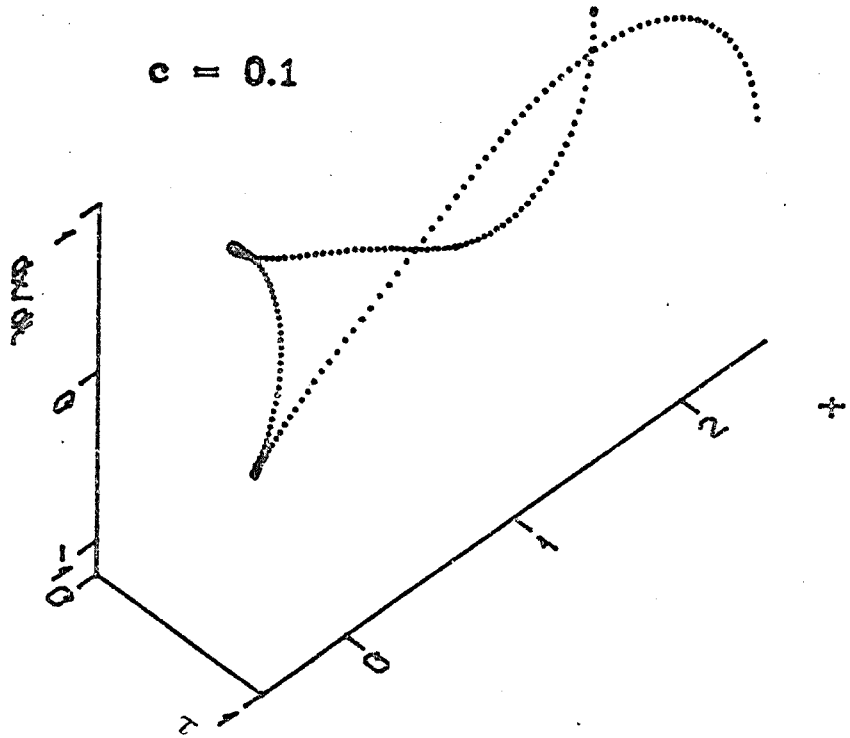
Journal of Computational Physics
Chr. Kaas-Petersen
Figure 2b



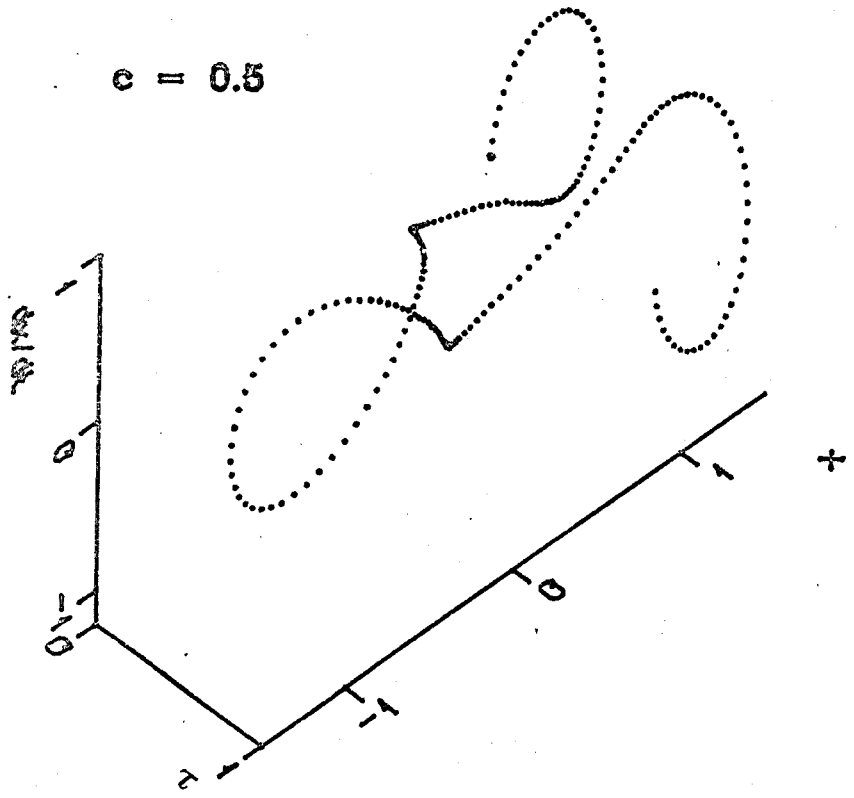
Journal of Computational Physics
Chr. Kaas-Petersen
Figure 2c



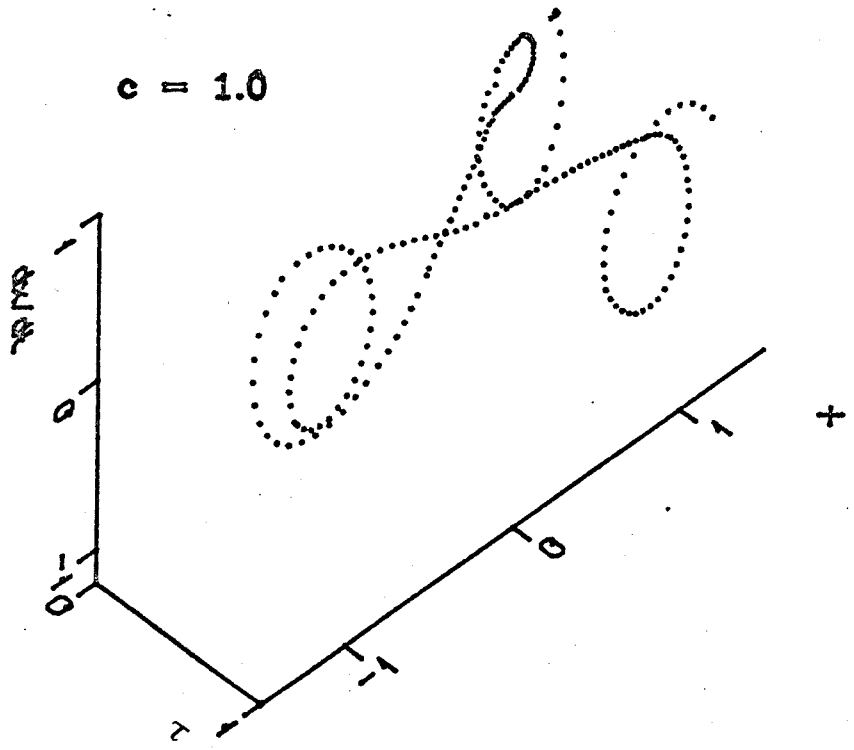
Journal of Computational Physics
Chr. Kaas-Petersen
Figure 2d



Journal of Computational Physics
Chr. Kaas-Petersen
Figure 2e



Journal of Computational Physics
Chr. Kaas-Petersen
Figure 2f



Dynamical behaviour of railway bogies

Chr. Kaas-Petersen,
Laboratory of Applied Mathematical Physics,
The Technical University of Denmark, Bldg. 303,
DK-2800 Lyngby, Denmark

We shall examine two mathematical models of a railway bogie: a simple and a complex; both due to Cooperrider [1]. The models describe the dynamical behaviour of the two-axle-bogie as it moves with constant velocity V along the straight horizontal track. The bogie is connected to the carbody by linear springs and dampers, and the bogieframe is supported by linear springs and dampers on the wheel-axles. These linear forces we denote F_ℓ. The wheel/rail contact forces F_c, which depend on the velocity, are calculated according to Johnson's non-linear theory. The flanges on the wheels introduce flange forces F_f. We model F_f as a dead band spring with linear spring constant. The length of the dead band is 0.018 m and equals the double clearance between the flange and the rail head. The equations governing the dynamics of the bogie can be formulated as a system of non-linear ordinary differential equations on the form

$$dx/dt = F_\ell(x) + F_c(x;V) + F_f(x), \quad t \in R,\ x \in R^n,\ 0 \le V . \qquad (*)$$

Details of (*) can be found in [2], and will appear elsewhere.

The equations (*) have $x=0$ as a solution for all values of V. This stationary solution is stable until a value V_c, above which the solution is non-stable. At $V=V_c$ a Hopf bifurcation takes place, see [2]. Periodic solutions are generated in the Hopf bifurcation point. We construct numerically a Poincaré map, see [3], to transform the periodic solution into a fixed point. The fixed point depends on the velocity V and therefore traces a path, that has been computed by means of a path following algorithm, see [4].

In the simple bogie we allow 2 degrees of freedom: motion in the horizontal plane perpendicular to the track (lateral motion) and rotation in the horizontal plane around a vertical axis (yaw motion). We have studied two situations: $F_f \equiv 0$ (i.e. no flange forces) and $F_f \not\equiv 0$ (i.e. flange forces as described above). In both cases it is in fig. 1a-b seen, that the periodic solution bifurcates at $V_c \approx 70$ m/s and is non-stable (dashed curve). However, the path of periodic solutions turns back in a saddle-node bifurcation [3] and becomes stable (solid curve), and it remains stable

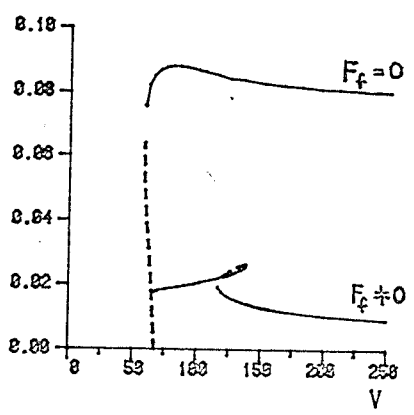


Fig. 1a. Simple bogie: Amplitude (peak-to-peak value in meters) in the lateral motion in dependence of V (in m/s).

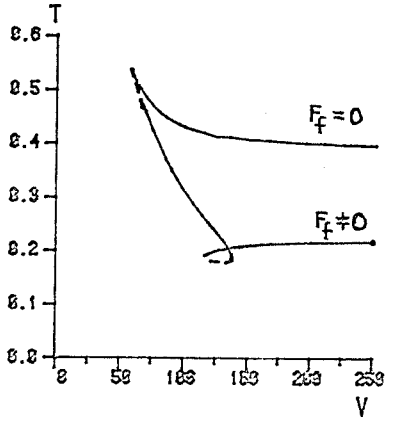


Fig. 1b. Simple bogie: Period (in seconds) of the oscillation in dependence of V (in m/s).

when $F_f \equiv 0$. When $F_f \not\equiv 0$ two more saddle-node bifurcations are seen. Thus 3 periodic solutions - 2 stable and 1 non-stable - exist in a certain velocity-interval. We have yet no explanation of the two extra saddle-node bifurcations.

In the complex bogie we allow 7 degrees of freedom: lateral and yaw motion of each wheel-axle and of the bogie frame, and rotation in a vertical plane around an axis parallel to the track (roll motion) of the bogie frame. The results for the case $F_f \equiv 0$ are similar to the corresponding results for the simple bogie. When flange forces are present, we do not get a loop on the path of periodic solutions; instead we find chaotic behaviour, see [5], confirmed by the calculation of a positive Liapunov exponent. The chaotic response is seen in fig. 2a-b. We have yet no explanation of the part of the motion, that seems to be periodic.

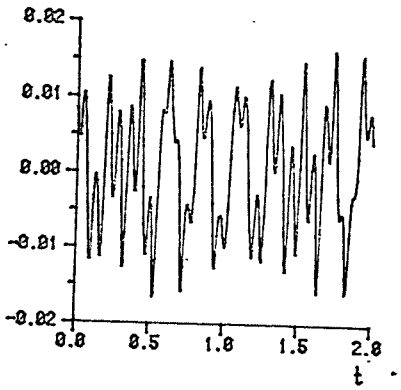


Fig 2a. Complex bogie: Lateral motion of rear wheel axle (in meters), time t in seconds.

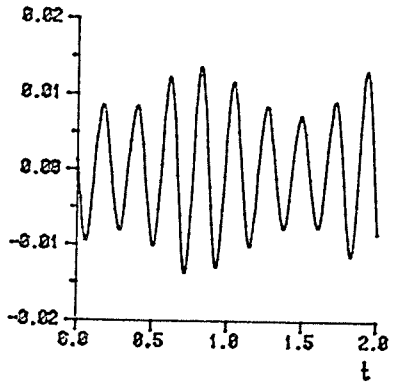


Fig. 2b. Complex bogie: Roll motion of bogie frame (in radians), time t in seconds.

References

[1] N.K. Cooperrider: The hunting behavior of conventional railway trucks, ASME J.Eng.Industry, Vol.94, 1972, pp. 752-762.

[2] H. True and C. Kaas-Petersen: A bifurcation analysis of non-linear oscillations in railway vehicles, Proceedings of 8th IAVSD Symposium, Boston, Mass., 1983, pp. 655-665.

[3] J. Guckenheimer and P. Holmes: Nonlinear oscillations, dynamical systems, and bifurcations of vectorfields, Springer-Verlag, 1983.

[4] C.B. Garcia and W.I. Zangwill: Pathways to solutions, fixed points, and equilibria, Prentice-Hall, 1981.

[5] A.J. Lichtenberg and M.A. Lieberman: Regular and stochastic motion, Springer-Verlag, 1983.

CKP
85 FEB 04

Computation, continuation, and bifurcation of torus solutions for dissipative maps and ordinary differential equations

by

Chr. Kaas-Petersen

Laboratory of Applied Mathematical Physics,
The Technical University of Denmark,
DK-2800 Lyngby, Denmark

23

This manuscript consists of 42 pages (text, tables, and figure captions) and 8 pages of drawings.

Running title: Computation of torus solutions

Submitted for publication to Physica D.

Abstract

A periodic solution can be considered as a fixed point of a Poincaré map. We shall demonstrate, how we can generalize this fixed-point-equivalence to quasi-periodic solutions with two periods. Thus the quasi-periodic solution or bi-periodic solution will be a fixed point of a more general Poincaré map. The stability of the bi-periodic solution is then equivalent to the stability of the fixed point. We demonstrate, how we can follow the fixed point in dependence of any control parameter in the system. Furthermore we demonstrate both how the bi-periodic solution may bifurcate to more complicated solutions and how it may arise as a bifurcation from a periodic solution. Our work is algorithmic in spirit, and our statements are based on numerical evidence.

Introduction

Let $S : X \to R^n$, $X \subseteq R^n$ be a smooth, dissipative map. We call it the stroboscopic map. For $x \in X$ we define the strobed orbit

$$\Sigma(x) = \{S^i(x) \in X : i = 0,1,2,\ldots\} = \{x, S(x), S^2(x), \ldots\} .$$

If a point $x_p \in X$ exists such that $\Sigma(x_p)$ is a finite set of T points and such that $S^T(x_p) = x_p$, then S has a periodic solution of period T.

If a point $x_T \in X$ exists such that $\Sigma(x_T)$ is an infinite set of points on a closed curve γ (one-dimensional manifold) in X and such that the iterates $S^i(x_T)$ wind round and round on γ, then S has a torus solution. We shall define a real number T' such that $S^{T'}(x_T) = x_T$. T' is the winding time of the torus, and the winding number is $w = 2\pi/T'$

We shall show that a torus solution may be treated as a fixed point of a Poincaré map; then the linearized stability of the torus is equivalent to the linearized stability of the fixed point, which is determined from the eigenvalues of the derivative of the Poincaré map in the fixed point. We also show that bifurcation phenomena and continuation in a control parameter can be performed for torus solutions. This therefore generalizes the well-known Poincaré map of periodic solutions. We shall study stroboscopic maps derived from an underlying dynamical system. The dynamical system may be described by an iterated map or by an ordinary differential equation (autonomous, periodically forced, or bi-periodically forced).

The origin of 'torus solution' stems from dynamical systems described by a system of three autonomous ordinary differential equations with dissipation. The flow of such a vector field will lie on the surface of a torus, and the strobed orbit Σ is a cross-section of the torus. Such a torus solution is also called a T^2-solution, since basically there are two periods in the solution: one period is the time it takes to return to the cross-section when starting in it; the other period is the time it takes to jump

(like a kangaroo) once round the strobed orbit. Other synonyms for a torus solution are: doubly periodic solution, almost periodic solution, quasi-periodic solution with two periods, T^2-solution, bi-periodic solution, and amplitude modulated signals; see [1, 2, 10, 11, 12, 14, 20]. We shall be concerned with T^2-solutions and only indicate the straightforward extensions to T^n-solutions.

A torus γ is said to be invariant, whenever the points $S^i(x)$, $i = 0,1,2,\ldots$ remain on γ for any initial $x \in \gamma$. This implies that the closed strobed orbit cannot cross itself in double points. That situation is analogous to what holds for a periodic orbit in state space of an autonomous system of ordinary differential equations. We note that torus solutions of bi-periodically forced ordinary differential equations need not be invariant.

A stable torus is nice, since if we start with an x close to the torus and iterate the map S, then we will be attracted to the torus. Therefore x may be only a rough approximation, and since we remain near the torus, time series analysis (using Fast Fourier Transform) has been applied [7, 16], and also the power spectra have been obtained. Bifurcation of tori has been determined from the Liapunov exponent [7, 16, 23]. Our method does not rely on the torus being stable, since we formulate a fixed point problem, where we find the fixed point first and afterwards determine its stability. In Examples 1 and 2 we have computed non-stable tori.

The study of torus solutions has been performed analytically by use of double Fourier expansions [2, 11] or by use of averaging techniques [2, 12, 20]. The Fourier expansion has the problem of small denominators, but will still converge if the KAM-condition is satisfied. The averaging techniques can only be performed when T' is irrational, and the approximate solution obtained may only be valid in a small parameter interval. A Fourier expansion is a global approximation of the torus. We shall perform only a local approximation, and it turns out that we do not have the small denomi-

nator problem, but the KAM-condition must still be satisfied. We shall explain how the KAM-condition enters our formulation. We shall also demonstrate a method to follow a torus in dependence of a parameter. To start up such a method averaging can be applied to get a close approximation to the torus. This we demonstrate in Example 2.

Fourier series have been used in numerical experiments (implemented as Galerkin methods) [5, 6]. Also graphical tools have been used [3] to study tori.

For periodic solutions the linearized stability is usually determined from the Floquet multipliers. To obtain these we have to solve the linearized variational equation [10, 14], which is a system of n^2 ordinary differential equations. It is an elaborate task to solve these equations even for small values of n. Furthermore, the Floquet theory cannot be generalized to cover torus solutions. We shall instead study the behaviour of the Poincaré map near the fixed point, which amounts to determining the eigenvalues of the derivative of the Poincaré map.

Bifurcations of tori from periodic solutions have been described in [2, 7, 10, 13, 14]. They approximate the centre manifold to some order. The formulas are very complicated, since the computation of derivatives involves solution of the linearized variational equation. We shall apply Newton's method to find the torus when we are near it, and therefore we are satisfied with a linear approximation to the centre manifold.

Torus solutions have attracted much interest, since Landau suggested an infinite sequence of higher and higher order tori bifurcating from each other as a route to chaos. [11, 20]. Newhouse, Rouelle, and Takens stopped at tori of third order [11]; however, higher order tori may exist [23, 24]. Torus solutions may also arise when a Hopf-bifurcation and a stationary bifurcation interact [17, 21]. Usually, torus solutions appear, when oscillators are coupled [12, 22].

The method we are going to describe can possibly be extended to cover conservative (Hamiltonian) systems, which have been treated extensively in [18].

First, to introduce the methods we shall study dynamical systems described by iterated maps. The basic thing is to define the stroboscopic map S. Then we define a linear subspace M and an affine subspace M_0 through some point x_0. M and M_0 depend on whether we consider periodic or torus solutions. Thereafter we define the residual map $Q : M_0 \to M$, that has the sought solution as a zero. Finally we define the Poincaré map $P : M_0 \to M_0$, that has the solution as a fixed point. We illustrate the theory with an example due to Kaneko [16]. Afterwards we study dynamical systems described by ordinary differential equations.

In the following we let $\langle \cdot,\cdot \rangle$ denote the usual scalar product in R^n, and $|\cdot|$ the induced norm; thus

$$\langle x,y \rangle = \sum_{i=1}^{n} x_i y_i \quad \text{and} \quad |x|^2 = \langle x,x \rangle ,$$

where

$$x = (x_1,\ldots,x_n) \in R^n , \quad y = (y_1,\ldots,y_n) \in R^n .$$

Periodic solutions

We shall study the nonlinear iterated map

$$f : X \to R^n \quad , \quad X \subseteq R^n$$

$$x_{k+1} = f(x_k) \ .$$

The stroboscopic map S on M = X is defined by

$$S : M \to M \quad , \quad S(x) = f(x) \ ,$$

and the iterates $S^i : M \to M$ of S are defined inductively by

$$S^i = S \circ S^{i-1} \quad , \quad i = 2,3,\ldots$$

$$S^1 = S = f \quad , \quad S^0 = I = \text{identity map}$$

A periodic orbit γ of S is a set characterized by a point $x_P \in M$ and an integer T, the period, such that

$$\gamma = \{x_P, S(x_P), \ldots, S^{T-1}(x_P)\} \quad ; \quad S^T(x_P) = x_P \quad \text{for} \quad x_P \in M \ .$$

Let $\{e_i\}$ be an orthonormal basis of M. Since $M \subseteq R^n$ we can use the canonical basis of R^n. M has its origin in 0. For $x_0 \in X$ let M_0 denote M when its origin has been translated to x_0. For the study of periodic solutions we may use $x_0 = 0$, $M_0 = M$. For all x in a neighbourhood $U \subseteq M_0$ of x_P we can define the residual vector $q \in M$

$$q = S^T(x) - x \quad , \quad x \in U \ .$$

The residual map Q is then the residual vector q expressed in the basis $\{e_i\}$,

$$Q : U \to M \quad , \quad Q(x) = (\langle q, e_1 \rangle, \ldots, \langle q, e_n \rangle) \quad , \quad U \subseteq M_0$$

and the Poincaré map P is

$$P : U \to M_0 \quad , \quad P(x) = (Q + I)(x) \quad , \quad U \subseteq M_0$$

where I is the identity. Wee see that x_P is on the periodic orbit γ when x_P is a zero of Q or equivalently x_P is a fixed point of P.

Let x be a point close to the zero of Q. To find the zero we can use Newton's method, i.e.

$$x_{new} = x_{old} + h \quad , \quad x_{new} \in U \quad , \quad x_{old} \in U \quad , \quad h \in M \ ,$$

where h is the solution of

$$DQ(x_{old})\, h = - Q(x_{old}) \ .$$

The derivative DQ can be approximated with a matrix of forward differences. When the period T is low, say T = 1, then DQ can be computed analytically. However, we have used numerical approximations. An optimal step-length for the differences has been found by use of the algorithm in [9].

The linearized stability of γ is determined from the eigenvalues of $DP(x_P)$. We have $DP(x_P) = DQ(x_P) + I(x_P)$, which is a square matrix of the same dimension as M. If all eigenvalues are inside the unit circle in the complex plane, then γ is stable; if one or more eigenvalues are outside, then γ is nonstable; see [10].

Torus solutions

A torus solution of S is characterized by a point x_T such that the iterates $S^i(x_T)$ loop round and round in a closed orbit γ. Thus $\Sigma(x_T) \subseteq \gamma$.

Let n_T be (an approximation to) the tangent of $\Sigma(x_T)$ in x_T with length 1, and let $\{n_T, e_1, \ldots, e_{n-1}\}$ be an orthonormal basis of $X \subseteq R^n$. Then we define M to be the linear subspace

$$M = \text{span}\{e_1, \ldots, e_{n-1}\}$$

orthogonal to n_T. M translated into x_0 is the affine subspace M_0, and we have

$$M_0 = \{x \in X : \ \langle x - x_0, n_T \rangle = 0\} .$$

We choose x_0 such that $x_T \in M_0$ and call M_0 a cross-section of the torus.

Let k and ℓ be integers such that the iterates $S^k(x_T)$ and $S^\ell(x_T)$ in X are on opposite sides of M_0, but near to x_T. Then the sequence of points

$$S^i(x_T) , \quad i = \ell, \ell+k, \ell+2k, \ldots, \ell+mk , \quad 1 \le m ,$$

will follow γ in one direction and be nearly equidistantly spaced. These points we interpolate. The interpolating curve crosses M_0 in x_T (within interpolating accuracy). See Figure 1.

First we discuss how we determine k, ℓ, and $\hat{m}$. Let us define

$$d_i = |S^i(x) - x| \qquad i = 1,2,\ldots , \quad x \in M_0$$
$$p_i = \langle S^i(x) - x, n_T \rangle \qquad i = 1,2,\ldots , \quad x \in M_0 .$$

d_i is the distance of the point $S^i(x)$ from the initial point x, and p_i is the projection of that distance on the normal n_T of M_0. Fix a number i_{max} to be the maximal number of iterates allowed and give m a value. Then we can define k by

$$|d_k| = \min\{|d_i| : \qquad i = 1,2,\ldots,i_{max}/m\}.$$

Thus $S^k(x)$ is the iterate closest to x.

We can define ℓ by

$$|d_\ell - \frac{m}{2} d_k| = \min\{|d_i - \frac{m}{2} d_k| : \ p_i \cdot p_k < 0. \ \ \& \ \ i = 1,2,\ldots,i_{max}/m\} .$$

Thus $S^\ell(x)$ is on the opposite side of $S^k(x)$. To ensure that the sequence of points will be distributed nearly symmetrically around x we redefine m to

$$m = -2 \cdot p_\ell / p_k .$$

Then $i_{max} = \ell + m \cdot k$.

The set of iterates we obtain are parameterized by p_i. For the parameter-value zero the interpolating curve hits M_0 in the point s.

Let $U \subseteq M_0$ be a neighbourhood of x_T. For any $x \in U$ and given (k,ℓ,m) we can define the residual vector $q \in M$

$$q = s - x \quad , \quad x \in U .$$

The residual map Q is then the residual vector q expressed in the basis $\{e_i\}$,

$$Q : U \to M \quad , \quad Q(x) = (\langle q, e_1 \rangle, \ldots, \langle q, e_{n-1} \rangle) \quad , \quad U \subseteq M_0 ,$$

and the Poincaré map P is

$$P : U \to M_0 \quad , \quad P(x) = (Q + I)(x) \qquad , \ U \subseteq M_0 .$$

Now we have reduced the torus problem to a fixed-point problem of P or equivalently to a zero problem of Q. If $x_T \in M_0$ is on the torus, then the linearized stability of x_T is determined from the eigenvalues of $DP(x_T)$. Since Q and therefore P are known numerically we must compute DP by difference approximations.

A periodic solution has a period T. The analogy for a torus γ is the winding time T', which we define now. Start in $x_T \in \gamma$. Then the iterates $S^1(x_T)$, $S^2(X_T)$,... will jump about on γ, see Figure 2.

Let $0 < i_1$ be the smallest number such that x_T is between $S^{i_1-1}(x_T)$ and $S^{i_1}(x_T)$. Then we have $i_1 - 1 < T' \leq i_1$. Let $i_1 < i_2$ be the smallest number such that x_T is between $S^{i_2-1}(x_T)$ and $S^{i_2}(x_T)$. Then we have $i_2 - 1 < 2 \cdot T' \leq i_2$. And so on. Then

$$T' = \lim_{n\to\infty} \frac{i_n}{n} \quad .$$

T' is a real number, and we have $S^{T'}(x_T) = x_T$.

For a periodic solution one defines the frequency $\nu = 1/T$ and the circular frequency $\omega = 2\pi/T$. The analogies for a torus are $\nu' = 1/T'$, which under special conditions stated below is the rotation number, and $\omega' = 2\pi/T'$, which we shall call the winding number and denote by w.

Thus

periodic solution: $\nu = 1/T$, $\omega = 2\pi/T$;

torus solution : $\nu' = 1/T'$, $w = \omega' = 2\pi/T'$.

When w is known we can use

$$d_i = |p_i|$$

$$p_i = i \cdot w \pmod{]-\pi,\pi]}$$

to determine k, ℓ, and m.

When the strobed orbit $\Sigma(x_T)$ is two-dimensional, i.e. $X \subseteq R^2$, ν' is called the rotation number of Σ, and it can be interpreted geometrically: Let x_0 be any point inside the strobed orbit. We can define the angle of rotations $a_i > 0$ between the two vectors $S^{i-1}(x_T) - x_0$ and $S^i(x_T) - x_0$ pointing from x_0 to two successive points on the strobed orbit. The rotation number is then

$$\nu' = \frac{1}{2\pi} \lim_{j\to\infty} \frac{1}{j} \sum_{i=1}^{j} a_i \quad .$$

The same interpretation is valid, when Σ essentially is lying in a two-dimensional hypersurface of X, $X \subseteq R^n$, $n > 2$, as is always the case when a torus bifurcates from a periodic solution.

When DP of a torus solution γ has -1 as one distinct eigenvalue the torus undergoes a torus doubling bifurcation. This means that the closed curve γ is split lengthwise into one new curve twice as long. Therefore the winding time T' doubles, the winding number w halves, but the rotation number (if defined) may remain constant.

Continuation of solutions

First we shall describe continuation of periodic solutions and then continuation of torus solutions. We take the map $f: X \to R^n$, $X \subseteq R^n$ and extend it with one real control parameter c, thus

$$f(y) \equiv f(x;c) : Y \to R^n$$

$$Y = X \times C \quad , \quad X \subseteq R^n \quad , \quad C \subseteq R \ .$$

X is the state space and C is the parameter space. Then the residual map for the periodic solution is

$$Q: M_0 \times C \to M \quad , \ M_0 = X \quad , \quad M \subseteq R^n \ .$$

Q defines implicitly a one-dimensional curve of zeros in $M_0 \times C$ by the equation

$$Q(y) = Q(x;c) = 0 \ .$$

This curve we shall call the path.

Let x_P be on a periodic orbit of f for $c = c_P$. We define $y_P = (x_P;c_P)$. So $Q(y_P) = Q(x_P;c_P) = 0$. A continuation method or a path following technique then traces out the path through y_P. Let $t_P \in R^{n+1}$ be a unit vector approximating the tangent of the path in y_P and let $\{t_P, f_1, \ldots, f_n\}$ be an orthonormal basis of $Y \subseteq R^{n+1}$. A scheme for a path following technique or a continuation method is this (cf. Figure 3):

(i) Take a step of length $h_P > 0$ from $y_P \in Y$ in the direction of t_P;

h_P could be determined by

a) $|Q(y_P + h_P t_P)| \leq \varepsilon_P$ for $0 < \varepsilon_P << 1$

b) $h_P \leq h_{max}$.

We then reach $z_P = y_P + h_P t_P$.

(ii) Let N be the hypersurface in Y through z_P and spanned by $\{f_1, \ldots, f_n\}$. Thus $N = R^n$ and $N = z_P + \text{span}\{f_1, \ldots, f_n\}$, so $\dim(M) = \dim(N)$.

(iii) Find the zero of the restriction of Q(y) to N,

$$Q|_N : N \to M \quad , \quad Q(x;c)|_N = 0 \ ,$$

by, for instance Newton's method. The zero is a point on the path.

For torus solutions we shall use $\{n_T, e_1, \ldots, e_{n-1}\}$ as an orthonormal basis of X, and $\{n_T, t_P, f_1, \ldots, f_{n-1}\}$ as an orthonormal basis of Y, such that

$$M = \text{span}\{e_1, \ldots, e_{n-1}\} \quad \text{and} \quad N = z_P + \text{span}\{f_1, \ldots, f_{n-1}\} \ .$$

For torus solutions the vector n_T in general depends on the parameter c. We must therefore update the bases now and then. As criterion for updating we have used

$$|< (\text{old } n_T) \ , \ (\text{new } n_T) >| \leq 0.95 \ .$$

In order to obtain the bases of X we first computed a set of linearly independent vectors orthogonal to n_T using the method in [4]. Then we used the modified Gram Schmidt process [19] to make it orthonormal.

The basis of Y with known t_P (and n_T) was obtained in a similar way.

To obtain the basis of Y with unknown t_P (but n_T known, and therefore also $e_1, \ldots, e_{n-1}$ known) we note that the i'th row of DQ, the derivative of Q, is the gradient ∇Q_i of the i'th coordinate of Q. Thus

$$\nabla Q_i = \left(\frac{\partial Q_i}{\partial e_1} \ , \ \ldots \ , \ \frac{\partial Q_i}{\partial e_{n-1}} \ , \ \frac{\partial Q_i}{\partial c} \right) \quad , \quad i = 1, \ldots, n-1 \ .$$

Let

$$f_i = \frac{\partial Q_i}{\partial e_1} \begin{bmatrix} e_1 \\ 0 \end{bmatrix} + \ldots + \frac{\partial Q_i}{\partial e_{n-1}} \begin{bmatrix} e_{n-1} \\ 0 \end{bmatrix} + \frac{\partial Q_i}{\partial c} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad , \quad i = 1, \ldots, n-1$$

be the gradient in Y. All gradients are seen to be orthogonal to n_T. Furthermore, the gradients point in the directions of maximal change of Q_i. But on the path Q_i is constant with value zero, so t_P must be orthogonal to f_i. If we assume that DQ always has maximal rank, then $\{f_i\}$ is a set of

linearly independent vectors. Then, whenever $\{n_T, f_1, \ldots, f_{n-1}, t_p\}$ is a set of linearly independent vectors the modified Gram Schmidt process will furnish t_p as the tangent of the path as well as an orthonormal basis of N. This procedure has also been used for paths of periodic solutions.

In general, t_p will depend on the parameter c. We must therefore update the basis of Y from time to time. As a criterion for updating we have used

$$|\langle(\text{old } t_p)\ ,\ (\text{new } t_p)\rangle| \leq 0.95\ .$$

For a general presentation of path-following techniques we refer to [8].

Bifurcation of a torus from a periodic solution

The periodic solution γ may for a parameter value c* have one pair of complex conjugated eigenvalues λ and $\overline{\lambda}$ of DP on the unit circle in the complex plane. We assume that all other eigenvalues are off the unit circle, and $\lambda, \overline{\lambda}$ to be distinct. We require λ to move away from the unit circle when the parameter c is changed from c*, i.e. the strict crossing condition

$$\frac{d}{dc}\,|\lambda|(c^*) \neq 0$$

is satisfied. If we furthermore assume weak resonance, i.e. $\lambda^i \neq 1$ for $i = 1,2,3,4$, then a torus bifurcates from the periodic solution at $c = c^*$. c* is the critical parameter value. In the next section we describe a method to locate c*. Here we shall examine the stability of the bifurcating torus.

Let $x_p^* \in M_0$ be on the periodic solution for $c = c^*$. Then $S^T(x_p^*) = x_p^*$ for some T, and the conditions of bifurcation above are satisfied. It is known that a two-dimensional centre manifold $W^c(x_p^*) \subseteq M_0$ through x_p^* exists, whose tangent space $E^c(x_p^*)$ is spanned by the two vectors e_1 and e_2, where

$$e_1 = \mathrm{Re}(v)\quad ,\quad e_2 = \mathrm{Im}(v)\quad ,\quad DP(x_p^*)\cdot v = \lambda\, v\ .$$

Furthermore, the strobed orbit Σ is to a first approximation an ellipse with its centre in x_p^* having e_1 and e_2 as conjugated diameters. Or, in other words, Σ will be a circle in the e_1-e_2-coordinate system. This has been proved by Iooss [13], see also [10, 20].

Near the bifurcation point, the stroboscopic map S^T is well described by its restriction S^c to $E^c(x_p^*)$. When $E^c(x_p^*)$ is considered as a complex plane with e_1 as the real axis, e_2 as the imaginary axis, and ζ as the complex variable, then

$$S^c = S^T|_{E^c(x_p^*)}\quad ,\quad S^c : E^c(x_p^*) \to E^c(x_p^*)\ ,$$

$$S^c(\zeta) = x_p^* + \zeta[\lambda + \alpha\,\zeta\,\overline{\zeta}] + O(|\zeta|^3)\quad ,\quad \alpha \in R\ .$$

If $\alpha > 0$ the strobed orbit is non-stable since the iterates of S^c are repelled from x_p^* and the non-stable torus exists for $c \leq c^*$ (subcritical bifurcation). If $\alpha < 0$ the iterates of S^c are attracted to x_p^*, so the torus is stable and exists for $c^* \leq c$ (supercritical bifurcation). If $\alpha = 0$ higher order terms must be included to determine stability.

To determine stability of the torus we do as follows. Let x be a point in $E^c(x_p^*)$ close to x_p^*, say $x = x_p^* + he_1$ for $h > 0$. Then we compute the rotated vector

$$r = S^T(x) - x_p^* .$$

If $e_3, \ldots, e_n$ are linearly independent and orthogonal to e_1 and e_2 we have

$$r = \beta_1 e_1 + \beta_2 e_2 + \beta_3 e_3 + \ldots + \beta_n e_n \quad , \quad \beta_i \in R .$$

The projection of r to $E^c(x_p^*)$ is then $\beta_1 e_1 + \beta_2 e_2$. We obtain β_1 and β_2 by taking scalar product with e_1 and e_2:

$$\beta_1 \langle e_1, e_1 \rangle + \beta_2 \langle e_2, e_1 \rangle = \langle r, e_1 \rangle$$

$$\beta_1 \langle e_1, e_2 \rangle + \beta_2 \langle e_2, e_2 \rangle = \langle r, e_2 \rangle$$

This linear system is regular, since e_1 and e_2 are linearly independent. The 2×2 matrix is independent of h. We define

$$\sigma = \sigma(h) = \sqrt{\beta_1^2 + \beta_2^2} / h$$

and have that $\sigma \to 1$ for $h \to 0+$. If $\sigma(h) < 1$, then the torus is stable; if $\sigma(h) > 1$ the torus is non-stable; if $\sigma(h) = 1$ we do not know the stability.

When we have found the bifurcation point c^* of the periodic solution x^*, then $(x^*; c^*)$ is an end-point on the path of torus solutions. We may start up the continuation method in the following way. Since the strobed orbit to a first approximation is two-dimensional we can define the winding number. It is known [10, 13, 20] that

$$w = \cos^{-1}(\mathrm{Re}\ \lambda)$$

and we can then determine suitable values of k, ℓ, and m. The strobed orbit is elliptical; we determine its major axis, which we use as t_p, and its minor axis, which we use as n_T. Then we can trace the torus solutions.

How to find a bifurcation point

The eigenvalues of the square matrix DP are complex numbers. On the complex plane we introduce the relation $\prec$:

$$a \prec b \quad \text{whenever} \quad |a| \leq |b| \quad , \quad a,b \text{ complex numbers.}$$

This relation is reflexive and transitive. Let $0 \leq n^s$ be the number of eigenvalues of DP with absolute value less than or equal to one, and let $0 \leq n^u$ be the number of eigenvalues outside the unit circle. Then $n^s + n^u = \dim M$ and we can arrange the eigenvalues such that

$$\lambda_1 \prec \ldots \prec \lambda_{n^s} \prec \lambda_{n^s+1} \prec \ldots \prec \lambda_{n^s+n^u} .$$

We define two real functions g^s and g^u by

$$g^s = \begin{cases} |\lambda_{n^s}|^2 - 1 & \text{for} \quad 0 < n^s \\ -1 & \text{for} \quad 0 = n^s \end{cases}$$

$$g^u = \begin{cases} |\lambda_{n^s+1}|^2 - 1 & \text{for} \quad 0 < n^u \\ +\infty & \text{for} \quad 0 = n^u \end{cases}$$

A bifurcation point is bracketed when n^s has different values in two points y_1 and y_2 on the path. Let $n^s = n_1$ in y_1, and $n^s = n_2$ in y_2. The bifurcation point is found as a zero of the function g:

$$g = \begin{cases} g^s & \text{for either} \quad (n^s = n_1 \;\&\; n_2 < n_1) \\ & \quad \text{or} \quad (n^s = n_2 \;\&\; n_1 < n_2) \\ g^u & \text{for either} \quad (n^s = n_2 \;\&\; n_2 < n_1) \\ & \quad \text{or} \quad (n^s = n_1 \;\&\; n_1 < n_2) \end{cases}$$

To find the zero we have used a modified regula falsi technique. First we compute the ratio $0 < r < 1$:

$$r = (0 - g(y_1))/(g(y_2) - g(y_1)) .$$

Then we define r' by

$$r' = \begin{cases} \alpha & \text{for} \quad r \leq \alpha \\ r & \text{for} \quad \alpha \leq r \leq 1-\alpha \\ 1-\alpha & \text{for} \quad 1-\alpha \leq r \end{cases}$$

We used $\alpha = 0.1$. (If $\alpha = 0.5$ we get bisection.) We define the affine subspace N to contain the point

$$(x;c) = y_1 + r'(y_2 - y_1) ,$$

and to be spanned by $\{f_1,\ldots\}$; then we use Newton's method on $Q|_N$ and compute $y_r = (x_r;c_r)$ on the path. We compute all eigenvalues of $DP(x_r)$ and evaluate $g(y_r)$. Depending on the sign of $g(y_r)$ we retain either $g(y_1)$ or $g(y_2)$.

The rationale for r' is that whenever the values of g indicate the root to be near the intervalends (the nearness is measured by α), then r' narrows the interval of the root better than r does.

We stop the iteration when $|y_2 - y_1| \leq \varepsilon_B$, $0 < \varepsilon_B \ll 1$.

Numerical considerations and practical details

Interpolation of the iterates was done with cubic splines. We use p_i as independent variable and for each coordinate of the iterates $S^i(x) \in X$ we perform an interpolation. We used the routines ICSCCU and ICSEVU from the IMSL-library.

The choice of n_T is not critical, since n_T is a normal of the cross-section that only has to be transversal to the strobed orbit. Since the number k can be found without knowing n_T we can always construct n_T as a unit vector parallel to $S^k(x) - x$. Here x must be close to the torus, but this requirement must always be satisfied in order to apply Newton's method.

We take x_T as a zero of Q, when $|Q(x_T)| \leq \varepsilon_Z$. We have the requirement $0 < \varepsilon_A < \varepsilon_Z \ll 1$, where ε_A is an upper bound on the accuracy of Q. To compute ε_A we have used the algorithm in [9].

The derivative DQ was computed numerically. We used a forward difference approximation, and the step-length to be used was found by means of the algorithm in [9].

The two algorithms in [9] could have been applied to the function g. However, we used $\varepsilon_B = 100 \cdot \varepsilon_Z$.

The eigenvalues of DP were computed by the IMSL-routine EIGRF, and the eigenvector v corresponding to the critical eigenvalue was found by an inverse power method.

The winding number w will in general be a function of c. Therefore, also (k,ℓ,m) will depend on c. Now, let c be fixed and let x_T be on the torus whose cross-section is M_0, and with winding number w. Then either two integers i and j exist such that

$$i \cdot w - j = 0,$$

or for all integers i and j we have

$$i \cdot w - j \neq 0 .$$

In the first case w is rational, and the torus is degenerate; in fact it is a periodic solution with period T = i. However, if i and j are large numbers it may be more convenient to treat the periodic solution as a bi-periodic solution.

In the second case w is irrational, but it may be impossible to perform the interpolation. To illustrate this let us consider the specific example $w = 2/17 + u$, $0 < u \ll 1$ with a u chosen such that w is irrational. Then we could use k = 17 and ℓ = 8, since $p_{17} = 17u$ is close to zero, and $p_8 = -1/17 + 8u$ is of opposite sign of p_{17}. The sequence $S^8(x_T)$, $S^{25}(x_T)$, $S^{42}(x_T)$, ... will be monotonous and it will proceed very slowly along the strobed orbit; in fact $S^{\ell}(x_T)$ and $S^{\ell+ik}(x_T)$ will be on opposite sides of the affine manifold M_0 when $p_{\ell+ik} > 0$, i.e. when

$$\left(-\frac{1}{17} + 8u\right) + i \cdot (17u) > 0 .$$

For a very small u, i will be very large. In general terms, we run into trouble when small integers i and j exist such that

$$|i \cdot w - j| \leq \varepsilon_T \quad \text{for} \quad 0 < \varepsilon_T \ll 1 .$$

This is the KAM-condition [10, 11, 14]. We do not get problems with small divisors, but we may be unable to afford that amount of computer time to obtain the required number of iterates.

Example 1

We have studied the map due to Kaneko [16]:

$$f(x;c): R^4 \times R \to R^4$$

$$f(x;c) = \begin{cases} ax_1 + (1-a)(1-cx_2^2) \\ x_3 \\ ax_3 + (1-a)(1-cx_4^2) \\ x_1 \end{cases}$$

where $a = 0.3$.

This map is dissipative, since the divergence of f equals 2a, which is different from ±1.

For $c = 1$ we have $x_P = (0,1,1,0)$ on a periodic solution of period $T = 1$. x_P is stable, since the eigenvalues of DP are $\lambda_1 = \lambda_2 = 0$, $\lambda_3 = \lambda_4 = a = 0.3$, all inside the unit circle in the complex plane. We have used the path-following method starting in $y_P = (x_P;c_P) = (0,1,1,0;1)$ and found bifurcation to a torus for $c = c^* = 1.31548781499$. The winding number was $w = 0.1060554$, and concerning the stability of the bifurcating torus we have the table below. It is seen, that $\sigma<1$,.so the torus is stable.

$h/10^{-7}$	σ
0.4	0.999999 991738
0.8	0.999999 96837
1.6	0.999999 92036
3.2	0.999999 8226
6.4	0.999999 6266

Starting in the bifurcation point c* with (k,ℓ,m) = (19,9,6) we have made a path following of the torus. The solution diagram is shown in Figure 4.

The torus becomes non-stable for c = 1.4706542. If, instead, we use (k,ℓ,m) = (19,47,1) the torus becomes non-stable for c = 1.4700663. The bifurcation point is therefore c* = 1.470 ± 0.001 in agreement with [16].

For c = 1 we also have the stable 1-periodic solution $x_P = (1,0,0,1)$. If we follow this solution it bifurcates supercritically to a torus for $c = c^* = 1.31548781499$.

Since our method does not rely on the torus being stable we have followed it beyond the bifurcation point. For c = 1.49 we have found

$$x_T = (-0.08333168259892598,\ 0.7731500446263004,$$
$$0.7331195145935536,\ -0.2887454426717730)$$

using (k,ℓ,m) = (19,28,2). The tangent of the strobed ortit was

$$n_T = (0.6753,\ -0.1548,\ 0.1606,\ 0.7030) .$$

The strobed orbit obtained starting in the x_T is shown in Figures 5a - 5d.

For the parameter value c = 1.4, x = (-0.12, 0.83, 0.82, -0.25) is close to the torus, and $n_T = (0.6765, -0.1514, 0.1413, 0.7067)$ is a unit vector near to the tangent of torus. Let $\{n_T, e_1, e_2, e_3\}$ be an orthonormal basis of R^4; we have used

$$e_1 = (0.2184,\ 0.9758,\ 0.0\ ,\ 0.0\ )$$

$$e_2 = (0.0\ ,\ 0.0\ ,\ -0.9806,\ 0.1961)$$

$$e_3 = (0.7033,\ -0.1574,\ -0.1359,\ -0.6797)$$

The projection of Q to e_j evaluated on the line $x + he_i$ is a univariate function q_{ij} and

$$q_{ij}: R \to R \quad , \quad q_{ij}(h) = \langle Q(x + he_i), e_j \rangle \qquad i,j = 1,2,3$$

For each pair (i,j) we have used the [9]-algorithms to determine both the accuracy ε_A of the value of q_{ij} and the optimal step-length h in numeri-

cal differentiation. In the construction of Q we have used $(k,\ell,m) = (19,9,6)$. Our results are shown in Tables 1a and 1b. Guided by these results we have chosen $\epsilon_Z = 10^{-8}$ and $h = 10^{-5}$ for all computations.

Note also that the non-stable periodic solution $x = (x_0,x_0,x_0,x_0)$ with $x_0 = (\sqrt{5}-1)/2$ and period $T = 1$ for $c = 1$ bifurcates subcritically to a torus for $c = 1.22448975$ with winding number $w \doteq 0.2260363$; the centre manifold is spanned by

$$e_1 = (\ 0.15\ ,\ 1.0,\ \ 0.15\ ,\ 1.0)$$

$$e_2 = (-0.98869,\ 0.0,\ -0.98869,\ 0.0)$$

around

$$x = (x^*,x^*,x^*,x^*)\ ,\quad x^* = 0.58333393916241\ .$$

To start the continuation method from the bifurcation point we found

$$n_T = (\ 0.5,\ 0.5,\ \ 0.5,\ 0.5)$$

$$t_P = (-0.5,\ 0.5,\ -0.5,\ 0.5,\ 0.0)\ .$$

A point on the non-stable torus $(k,\ell,m) = (9,44,2)$ was found as

$$x = (0.5812319821932173,\ 0.5854346965900306,$$
$$0.5812319221932173,\ 0.5854346965900306)$$

$$c = 1.2244892303\ .$$

One of the eigenvalues of DP was

$$\lambda_1 = 0.03126$$

inside the unit circle, and two eigenvalues were

$$\lambda_2 = 5.265 \quad \text{and} \quad \lambda_3 = 4732.0$$

outside the unit circle.

Ordinary differential equations

We assume that dissipation takes place. In order to use the algorithm described above we must define the stroboscopic map S and the manifold M. The differential equations define the flow φ. A solution starting in x_0 at time $t = 0$ is denoted by $\varphi(t;x_0)$. So $\varphi(0;x_0) = x_0$.

We shall study three classes of nonlinear ordinary differential equations.

Autonomous systems

The system is given by

$$\frac{dx}{dt} = \dot{x} = f(x) \quad , \quad x \in X\ ,\ X \subseteq R^n\ .$$

Let $\varphi(t;x_P)$ denote a periodic orbit γ of period T, $T > 0$. Then $\varphi(T;x_P) = \varphi(0;x_P) = x_P$. We introduce a hypersurface H in X that contains x_P and is transversal to γ. Let it have the unit normal n_P. So

$$H = \{x \in X : \ \langle x - x_0, n_P\rangle = 0\} \quad , \quad x_P \in H\ .$$

As x_0 we could choose x_P, and n_P could be the tangent of the orbit γ in x_P. If U is a neighbourhood of x_P in H, then we can define the stroboscopic map

$$S : U \to H \quad , \quad U \subseteq H$$

by the formula

$$S(x) = \varphi(T_R;x) \quad , \quad x \in U \quad , \quad T_R \text{ is close to } T\ .$$

T_R is the time it takes to return to H. When $x = x_P$, then $T_R = T$. For a general $x \in U$ we must determine T_R. We have used the algorithm in the box below.

Given $x \in U$, let T_1 be an approximation to T_R. Obtain $z_1 = \varphi(T_1;x)$. If $z_1 \in H$, then we are done and take $T_R = T_1$, $S(x) = z_1$.

Else $z_1 \notin H$. Now assume that we will solve the differential equations with Euler's method, starting in z_1 and only use one time-step to reach H. This means that we want $z_H \in H$, where

$$z_H = z_1 + \delta T_R \cdot f(z_1) .$$

δT_R is unknown, but subtract x_0 on both sides and make scalar product with n_P, we get one equation for δT_R:

$$0 = \langle z_1 - x_0, n_P \rangle + \delta T_R \cdot \langle f(z_1) , n_P \rangle .$$

Then define $T_2 = T_1 + \delta T_R$ and obtain $z_2 = \varphi(T_2;x)$. We continue this iteration process until we have z_i sufficiently near to H, that is

$$|\langle z_i - x_0 , n_P \rangle| \leq \varepsilon_R \quad , \quad 0 < \varepsilon_R << 1 .$$

Then we take $T_R = T_i$, $S(x) = z_i$.

We have used $\varepsilon_R = \tau/100$, where τ is the maximal local truncation error allowed in the integration process.

The strobed orbit Σ is in H. n_T is an approximation to the tangent of Σ, and we use $\langle n_T, n_P \rangle = 0$. When $\{n_T, n_P, e_1, \ldots, e_{n-2}\}$ is an orthonormal basis of X, we have the linear subspace M defined by

$$M = \text{span}\{e_1, \ldots, e_{n-2}\} ,$$

and the affine subspace through x_0 is defined by

$$M_0 = \{x \in X : \langle x - x_0, n_P \rangle = 0 \quad \& \quad \langle x - x_0, n_T \rangle = 0\} .$$

The formulas for d_i and p_i carry over.

The hypersurface H must be kept transversal to the flow. When x_T is on the torus, we let $n_f = f(x_T)/|f(x_T)|$ be a unit vector parallel to the

velocity vector in x_T. The surface H is updated when

$$|\langle n_P, n_f \rangle| \leq 0.95 ,$$

and we use n_f as the new n_P.

To apply the continuation method we have $\{n_T, n_P, e_1, \ldots, e_{n-2}\}$ as an orthonormal basis of X and $\{t_P, n_T, n_P, f_1, \ldots, f_{n-2}\}$ as an orthonormal basis of Y such that N has the basis $\{f_1, \ldots, f_{n-2}\}$. In order to use the bifurcation formulas we find e_1 and e_2 expressed in the basis of H. We transform them into vectors in X and then perform the bifurcation analysis.

Periodically forced systems

The period is $T > 0$ and the system is given by

$$\frac{dx}{dt} = \dot{x} = f(t,x) \quad , \quad f(t+T,x) = f(t,x)$$

$$0 \leq t \quad , \quad x \in X \quad , \quad X \subseteq R^n$$

The stroboscopic map S is then

$$S(x) = \varphi(T;x)$$

and its iterations are

$$S^i(x) = \varphi(iT;x) \qquad i = 1,2,\ldots$$

If we consider subharmonics jT of the system, we define

$$S(x) = \varphi(jT;x) .$$

Let x_T be on the torus and n_T be an approximation to the tangent of the strobed orbit. If $\{n_T, e_1, \ldots, e_{n-1}\}$ is an orthonormal basis in X, we have

$$M = \text{span}\{e_1, \ldots, e_{n-1}\}$$

$$M_0 = \{x \in X : \langle x - x_0, n_T \rangle = 0\}$$

and the formula for d_i and p_i carry over.

Biperiodically forced systems

The two periods are $0<T$ and $0<T'$, and the system is given by

$$\frac{dx}{dt} = \dot{x} = f(t,t,x)$$

$$f(t + T,t,x) = f(t,t + T',x) = f(t,t,x)$$

$$0 \leq t \quad , \quad x \in X \quad , \quad X \subseteq R^n \ .$$

The stroboscopic map is defined by

$$S^i(x) = \varphi(iT;x) \ , \quad i = 1,2,\ldots$$

For computational purposes we let $T<T'$. The formulas for d_i and p_i are given by

$$d_i = |p_i|$$

$$p_i = \frac{iT}{T'} \quad \left(\text{mod} \ \left]-\frac{1}{2} \ , \ \frac{1}{2}\right]\right) .$$

We have $M = R^n$, and we can use the canonical basis of R^n.

We can formally define a winding number $w = 2\pi T/T'$, but the strobed orbit may cross itself, so w has no geometrical interpretation in general. Three examples of biperiodically forced systems are given in [15].

Generalizations

The ordinary differential equations mentioned can all be extended by one control parameter; autonomous equations, for instance, will be on the form:

$$\frac{dx}{dt} = \dot{x} = f(x;c) \ , \quad x \in X \ , \quad c \in C \ , \quad X \subseteq R^n \ , \quad C \subseteq R \ .$$

We have made a transformation such that a torus solution turns into a fixed point of a Poincaré map. For autonomous systems the manifold M_0 is a codimension two surface in the state space X. For T-periodically forced systems and iterated maps, M_0 is of codimension one in the state space X, but is of codimension two in the space

$$X_1 = \{(\theta,x) : \theta = t \ (\text{mod } T) \ , \ x \in X\} \ .$$

For biperiodically forced systems with periods T and T' the manifold $M_0 = X$, but it is of codimension two in the space

$$X_2 = \{(\theta \ ,\theta',x) : \theta \ = t \ (\text{mod } T \ ) \ , \ \theta' = t \ (\text{mod } T') \ , \ x \in X\} \ .$$

A biperiodic solution is a closed curve of the stroboscopic map. This is easily generalized such that a triperiodic solution is a torus of the stroboscopic map and thus has two winding numbers.

Example 2

We shall study two coupled van der Pol oscillators originally treated by Hale [12]. Hale uses the method of averaging. Similar equations have been studied pictorially in [3]. The equations are

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -(2\pi)^2 x_1 + \varepsilon(1 - x_1^2 - a x_3^2 - b x_1^2 x_3^2) x_2$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = -(\sqrt{2}\pi)^2 x_3 + \varepsilon(1 - \alpha x_1^2 - x_3^2) x_4$$

For $\varepsilon = 0$ the $x_1 - x_2$-oscillator has period $T = 1$, and the $x_3 - x_4$-oscillator has period $T' = \sqrt{2}$. Based on this we have chosen $k = 7$, $\ell = 20$, $m = 6$.

It is known that for $b = 0$ and $0 < \varepsilon << 1$, $x = (x_{10}, 0, x_{30}, 0)$ is close to the torus; here

$$x_{10} = 2\sqrt{\frac{1-2a}{1-4a\alpha}} \quad , \quad x_{30} = 2\sqrt{\frac{1-2\alpha}{1-4a\alpha}} \, .$$

We use the stroboscopic map with return times close to T and therefore take $n_P = (0,1,0,0)$. To obtain x_{30} we must have $n_T = (0,0,0,1)$. Furthermore, if $a = 0.25$, $\alpha = 0.25$, then the torus is known to be stable.

First, using these facts we let $b = 0$ and perform a continuation in the parameter ε. We have obtained the results in Table 2.

Next, we fix $\varepsilon = 0.2$ and perform a continuation in b. We have obtained the results in Table 3.

If $a = 0.8$, $\alpha = 0.8$, then the torus is non-stable. For $b = 0$ we have made a continuation in ε. The results are given in Table 4.

In order to solve the ordinary differential equations we have used a 5-6 order Runge-Kutta-Verner method with the bound on the local truncation error $\tau = 10^{-8}$. For the stroboscopic map we have used $\varepsilon_R = 10^{-11}$. The [9]-algorithm has guided us to use $\varepsilon_Z = 10^{-8}$ and $h = 10^{-4}$.

Example 3

We consider a truncated Navier-Stokes equation due to Franceschini [7]. The equations $\dot{x} = f(x;c)$, where c is the Reynolds number, are:

$$\dot{x}_1 = -2 x_1 + 4\sqrt{5}\, x_2 x_3 + 4\sqrt{5}\, x_4 x_5$$

$$\dot{x}_2 = -9 x_2 + 3\sqrt{5}\, x_1 x_3 + 3\sqrt{5}\, x_6 x_7$$

$$\dot{x}_3 = -5 x_3 - 7\sqrt{5}\, x_1 x_2 + 9 x_1 x_7 + c$$

$$\dot{x}_4 = -5 x_4 - \sqrt{5}\, x_1 x_5$$

$$\dot{x}_5 = -x_5 - 3\sqrt{5}\, x_1 x_4$$

$$\dot{x}_6 = -8 x_6 - 4\sqrt{5}\, x_2 x_7$$

$$\dot{x}_7 = -5 x_7 + \sqrt{5}\, x_2 x_6 - 9 x_1 x_3$$

The system is dissipative, since $\mathrm{div} f = -35 < 0$. The torus we shall consider bifurcates from a periodic solution for $c = 226.98368$. At the top of the table below, x_P is a point on the periodic solution with period $T = 0.197353508$. The Poincaré map - defined on the hypersurface with normal n_P - has the derivative DP with two eigenvalues $\lambda, \bar{\lambda} = 0.73915 \pm i\, 0.67341$ on the unit circle. The centre manifold is spanned by e_1 and e_2, and the winding number of the strobed orbit is $w = 0.738984$. At the bottom of the table below it is seen that the torus bifurcates supercritically.

x_P	n_P	e_1	e_2
0.0813 13454	0.5538	0.1428	0.4752
-0.6221 98859	-0.1085	0.0299	-0.1828
-0.3252 52367	0.8137	0.0798	-0.3449
2.4439 4189	-0.0484	0.1779	0.0069
6.9864 5216	-0.0298	0.2451	0.0681
1.5043 6947	-0.1138	0.9221	0.0
-3.5353 1613	0.0568	-0.2205	0.0005

$h/10^{-3}$	σ
0.025	0.9999 8025
0.05	0.9999 7416
0.1	0.9999 6758
0.2	0.9999 5719
0.4	0.9999 3781
0.8	0.9999 997
1.6	0.9999 239
3.2	0.9999 724
6.4	0.9999 689

Let $c = 269$. Then x_T is a point on the torus, and in that point $\text{span}\{n_P, n_T\}$ is the tangent plane of the torus. With $(k,\ell,m) = (8,2,6)$ and T_R approximated by 0.1777862 we obtain

x_T	n_P	n_T
0.0004 04915	0.5789	-0.2736
-0.7348 99796	-0.1502	0.1322
-1.4233 5101	0.7885	0.3839
2.7786 9613	-0.0397	0.1630
7.7813 8729	-0.0222	0.1806
2.1048 7116	-0.1268	0.7949
-4.1932 8819	0.0500	-0.2630

Franceschini used $H = \{x \in R^7 : x_1 = 0\}$, so that $n_P = (1,0,\ldots,0)$, but for us that vector is not sufficiently parallel to the flow. We found the eigenvalues

$$\lambda_1 = 0.3582 \quad , \quad \lambda_2 = -0.0160 \quad , \quad \lambda_3, \lambda_4, \lambda_5 \simeq 0 .$$

A plot of the torus is shown in [7].

When we start in x_T and use n_P and n_T as above but take $c = 270$, the winding number has changed. This can be seen from Table 5. Table 5 shows that two rounds on the strobed orbit are performed in approximately 17 iterations. Because of this it may be unfeasible to compute $P(x_T)$. It is the transient that is manifested in the results at the top of the table, since x_T is not on the torus when $c = 270$.

Conclusions

We have shown how a quasi-periodic solution with two periods can be considered as a fixed point of a Poincaré map. This is a straightforward generalization of the usual treatment of periodic solutions.

When the fixed point problem is reformulated as a zero point problem standard methods to find a zero and to trace a path of zeros in dependence of a parameter can be used. Stability of fixed points of dynamical systems are well known. Our approach is algorithmic in spirit, and all results have been verified on a computer. We used an IBM 4341 under VM/CMS-control. Our code is in double precision standard Fortran. The code makes up an easy-to-use package PATH. With the help of this package we have studied an iterated map due to Kaneko [16] and a 7-mode truncated Navier-Stokes equation due to Franceschini [7]. In these two cases we have confirmed and extended their results. We also studied a system of two coupled van der Pol oscillators due to Hale [12] and confirmed his analytical results.

Acknowledgements

The modified Gram Schmidt process was mentioned to me by Dr. P.Chr. Hansen, Numerical Institute at our University. The paper of Kaneko [16] was handed to me by my colleague Dr. P.L. Christiansen. Thanks to Professor G. Iooss for discussions. Sincere thanks to Dr. Hans True of our Laboratory for reading through the whole manuscript and for his constructive criticism. Lastly thanks to all users of the program package PATH. Their criticism has been essential to make the code more reliable and has indirectly influenced the final presentation.

References

[1] V.I. Arnold: Ordinary differential equations, MIT Press (1980).

[2] V.I. Arnold: Geometrical methods in the theory of ordinary differential equations, Springer-Verlag (1983).

[3] R. Baxter, H. Eiserike, and A. Stokes: A pictorial study of an invariant torus in phase space of four dimensions. In L. Weiss (ed.): Ordinary differential equations, 1971-NRL-MRC Conference, Academic Press (1972).

[4] A. Ben-Israel and T.N.E. Greville: Generalized inverses: theory and applications, Krieger Publishing Company (1980).

[5] R. Bouc, M. Defilippi, and G. Iooss: On a problem of forced nonlinear oscillations. Numerical example of bifurcation into an invariant torus, Nonlin. Anal. Th. Met. & App., Vol. 2, No. 2, (1978) 211.

[6] L.O. Chua and A. Ushida: Algorithms for computing almost periodic steady-state response of nonlinear systems to multiple input frequencies, IEEE Trans. Circ. and Sys., Vol. Cas-28, No. 10 (1981) 953.

[7] V. Franceschini: Bifurcation of tori and phase locking in a dissipative system of differential equations, Physica 6D (1983) 285.

[8] C.B. Garcia and W.I. Zangwill: Pathways to solutions, fixed points, and equilibria, Prentice-Hall Inc. (1981).

[9] P.E. Gill, W. Murray, and M.H. Wright: Practical optimization, Academic Press (1981).

[10] J. Guckenheimer and P. Holmes: Nonlinear oscillations, dynamical systems, and bifurcations of vectorfields, Springer-Verlag (1983).

[11] H. Haken: Advanced synergetics, Springer-Verlag (1983).

[12] J.K. Hale: Oscillations in non-linear systems, McGraw-Hill (1963).

[13] G. Iooss: Bifurcation of maps and applications, North-Holland Publishing Company (1979).

[14] G. Iooss and D.D. Joseph: Elementary stability and bifurcation theory, Springer-Verlag (1980).

[15] C. Kaas-Petersen: Computation of quasi-periodic solutions of forced dissipative systems, J. Comput. Phys. (To appear.)

[16] K. Kaneko: Doubling of torus, Prog. Theor. Phys., Vol. 69, No. 6 (1983) 1806.

[17] W.F. Langford: Periodic and steady-state mode interactions lead to tori, SIAM J. App. Math., Vol. 37 (1979) 22.

[18] A.J. Lichtenberg and M.A. Lieberman: Regular and stochastic motion, Springer-Verlag (1983).

[19] B. Noble: Methods for computing the Moore-Penrose generalized inverse, and related matters, in Nashed (ed.): Generalized inverses and applications, Academic Press (1976).

[20] R.D. Richtmyer: Principles of advanced mathematical physics, Vol. II, Springer-Verlag (1981).

[21] J. Scheurle and J. Marsden: Bifurcation to quasi-periodic tori in the interaction of steady-state and Hopf bifurcations, SIAM J. Math. Anal., Vol. 15, No. 6 (1984) 1055.

[22] P.H. Steen and S.H. Davis: Transitions in weakly-coupled nonlinear oscillators, in J. Chadra and A.C. Scott (eds.): Coupled nonlinear oscillators, North-Holland Publishing Company (1983).

[23] R.K. Tavakol and A.S. Tworkowski: On the occurrence of quasi-periodic motion on three tori, Phys. Lett., Vol. 100A, No. 2 (1984) 65.

[24] R.K. Tavakol and A.S. Tworkowski: An example of quasi-periodic motion on T^4, Phys. Lett., Vol. 100A, No. 6 (1984) 273.

Table 1a. Upper bound on accuracy ϵ_A for $q_{i,j}$. The entries are $\epsilon_A/10^{-11}$.

j \ i	1	2	3
1	0.2708	0.4133	0.0809
2	0.2253	0.2071	0.1249
3	0.3952	2.4274	0.1365

Table 1b. Optimal steplength h in numerical differentiation for $q_{i,j}$. The entries are $h/10^{-4}$.

j \ i	1	2	3
1	0.8846	0.1890	0.3218
2	0.5862	0.8016	0.2164
3	0.7554	1.0058	0.1975

Table 2. a = 0.25, α = 0.25, b = 0 in Hale's equations.

	Initial conditions at t = 0		Eigenvalues of DP		Approximate return time
ε	x_1	x_3	λ_1	λ_2	T_R
0.0	1.632993162	1.632993162			1.0
0.01	1.632916700	1.633167571	0.6717	0.8758	1.000066051
0.02	1.632956580	1.633098034	0.4512	0.7670	1.000132023
0.05	1.632997055	1.633077861	0.1367	0.5151	1.000329509
0.1	1.633065860	1.633139317	0.0187	0.2649	1.000657007
0.2	1.633305868	1.633413482	0.00034	0.0693	1.001304092
0.5	1.634917089	1.635260312	$\simeq$ 0	0.0010	1.003136674
1.0	1.640460765	1.641858222	$\simeq$ 0	$\simeq$ 0	1.005388957

Table 3. a = 0.25, α = 0.25, ε = 0.2 in Hale's equations.

	Initial conditions at t = 0		Eigenvalues of DP		Approximate return time
b	x_1	x_3	λ_1	λ_2	T_R
0.0	1.633305868	1.633413482	0.00034	0.0693	1.001304092
0.01	1.618853192	1.640587089	0.00037	0.0700	1.001395837
0.02	1.604539179	1.647598364	0.00035	0.0706	1.001485959
0.05	1.562514607	1.667656385	0.00032	0.0719	1.001746264
0.1	1.495915068	1.697890872	0.00027	0.0726	1.002145585

Table 4. a = 0.8, α = 0.8, b = 0 in Hale's equations.

	Initial conditions at t = 0		Eigenvalues of DP		Approximate return time
ε	x_1	x_3	λ_1	λ_2	T_R
0.0	1.240347346	1.240347346			
0.01	1.240498330	1.240268184	1.096	0.6717	1.000121931
0.02	1.240422287	1.240322004	1.202	0.4511	1.000243803
0.05	1.240369433	1.240394951	1.584	0.1363	1.000608562
0.1	1.240326221	1.240557682	2.523	0.01814	1.001211177

Table 5. The results were obtained for c = 270; the initial point x_T, and the normals n_P & n_T corresponded to c = 269. p_i is the projection of the iterate on n_T, i.e. $p_i = \langle S^i(x_T) - x_T , n_T \rangle$.

i	p_i	i	p_i	i	p_i	i	p_i
0	0.0	1	-0.58430802	8	0.33156419	9	-0.27344641
17	-0.01975493	18	-0.59339426	25	0.30948621	26	-0.27681810
34	0.00148399	35	-0.59512208	42	0.29719716	43	-0.27631498
51	0.01629333	52	-0.59370128	59	0.29035136	60	-0.27391402
68	0.02755556	69	-0.59080579	76	0.28683236	77	-0.27043460
85	0.03661387	86	-0.58711210	93	0.28545456	94	-0.26628727
102	0.04423794	103	-0.58295173	110	0.28548083	111	-0.26170744
119	0.05090906	120	-0.57850387	127	0.28643205	128	-0.25684102
136	0.05693943	137	-0.57387113	144	0.28799143	145	-0.25178241
153	0.06253458	154	-0.56911434	161	0.28994824	162	-0.24659417
170	0.06783056	171	-0.56427041	178	0.29216241	179	-0.24131819
187	0.07291769	188	-0.55936210	195	0.29454057	196	-0.23598288
204	0.07785582	205	-0.55440367	212	0.29702103	213	-0.23060735
221	0.08268448	222	-0.54940414	229	0.29956282	230	-0.22520463
238	0.08742978	239	-0.54436933	246	0.30213883	247	-0.21978355
255	0.09210899	256	-0.53930303	263	0.30473117	262	-0.21435004
272	0.09673349	273	-0.53420775	280	0.30732807	281	-0.20890818
289	0.10131055	290	-0.52908530	297	0.30992184	298	-0.20346071

Figure captions

Figure 1. Construction of residual map Q of torus solution. We have taken $X = R^2$, and therefore M and M_0 become one-dimensional and are spanned by the unit vector e_1.

Figure 2. Sketch of a torus γ in R^3. The iterates $S^7(x_T)$ and $S^8(x_T)$ enclose x_T, so the winding time T' could have a value about 7.8752419...

Figure 3. Sketch of continuation of a torus of an iterated map for $X \subseteq R^2$, $C \subseteq R$, $Y = X \times C$. M_0 has n_T as normal and $\{e_1\}$ as basis. The vector t_P is in $M_0 \times C$. N has n_T & t_P as normals and $\{f_1\}$ as basis. The curve 'path' is the path of torus solutions.

Figure 4. Solution diagram of Kaneko's system. For the torus solutions the maximal and the minimal values of x_1 on the strobed orbits are shown. Full line means stable solution; dashed line means non-stable solution.

Figure 5. Projections of the torus for c = 1.49 with initial point x_T. In a and b the orbit is made of 100 points. In c and d the orbit is made of 2000 points. For this value of c the torus is non-stable. Due to numerical noise and the non-stability the iterates do not remain close to the torus. It is seen that a stable double torus attracts the iterates.

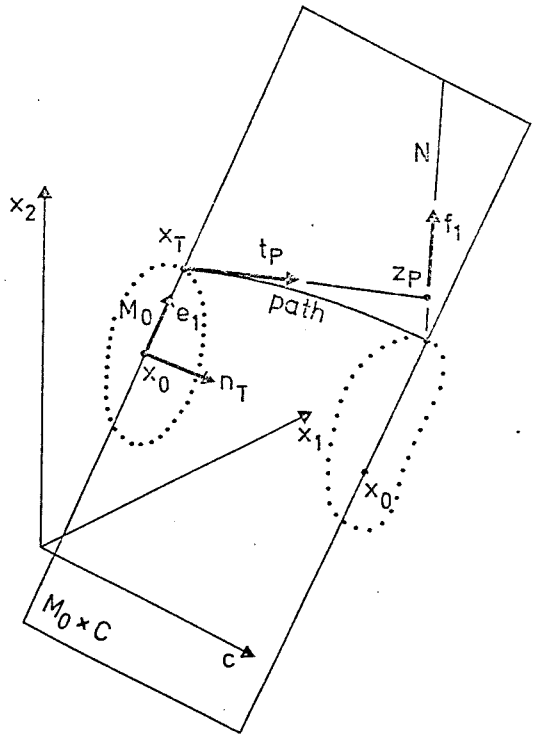


Figure 1

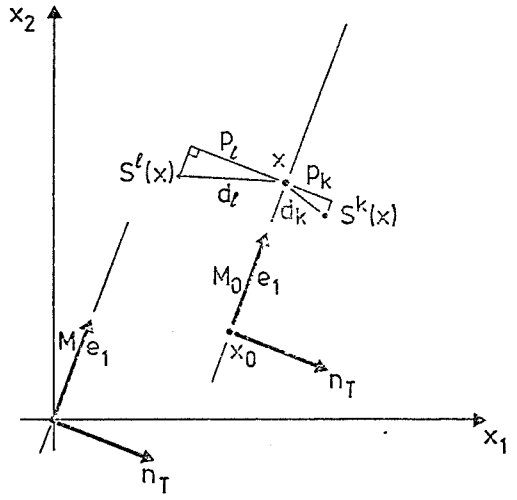


Figure 2

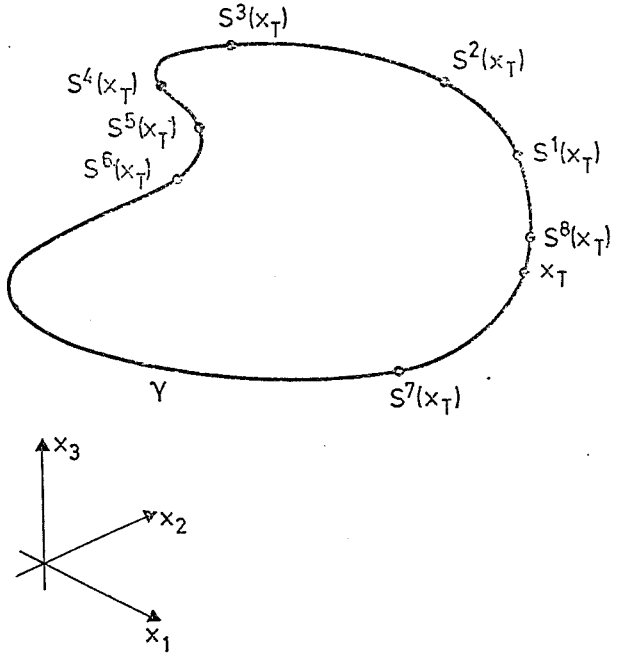


Figure 3

Figure

Chr. Kaas-Petersen

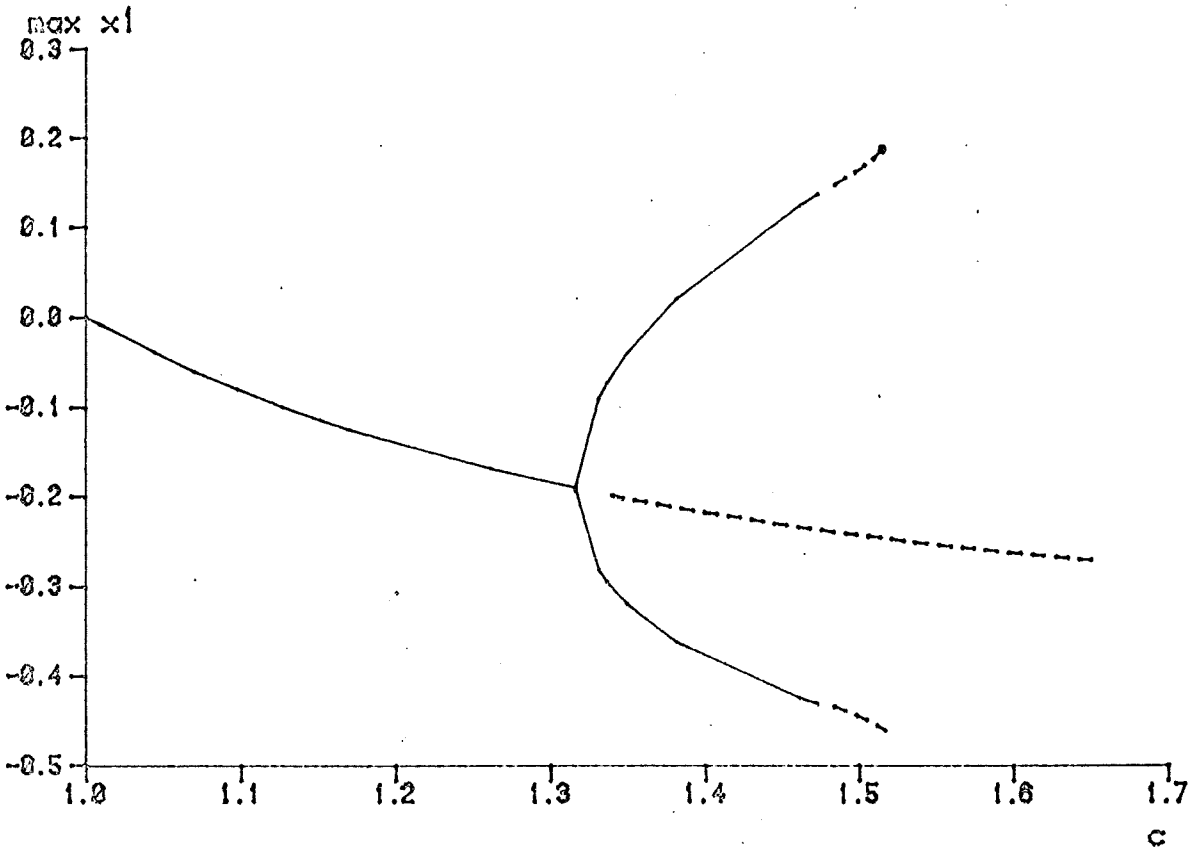


Fig. 5a

Chr. Kaas-Petersen

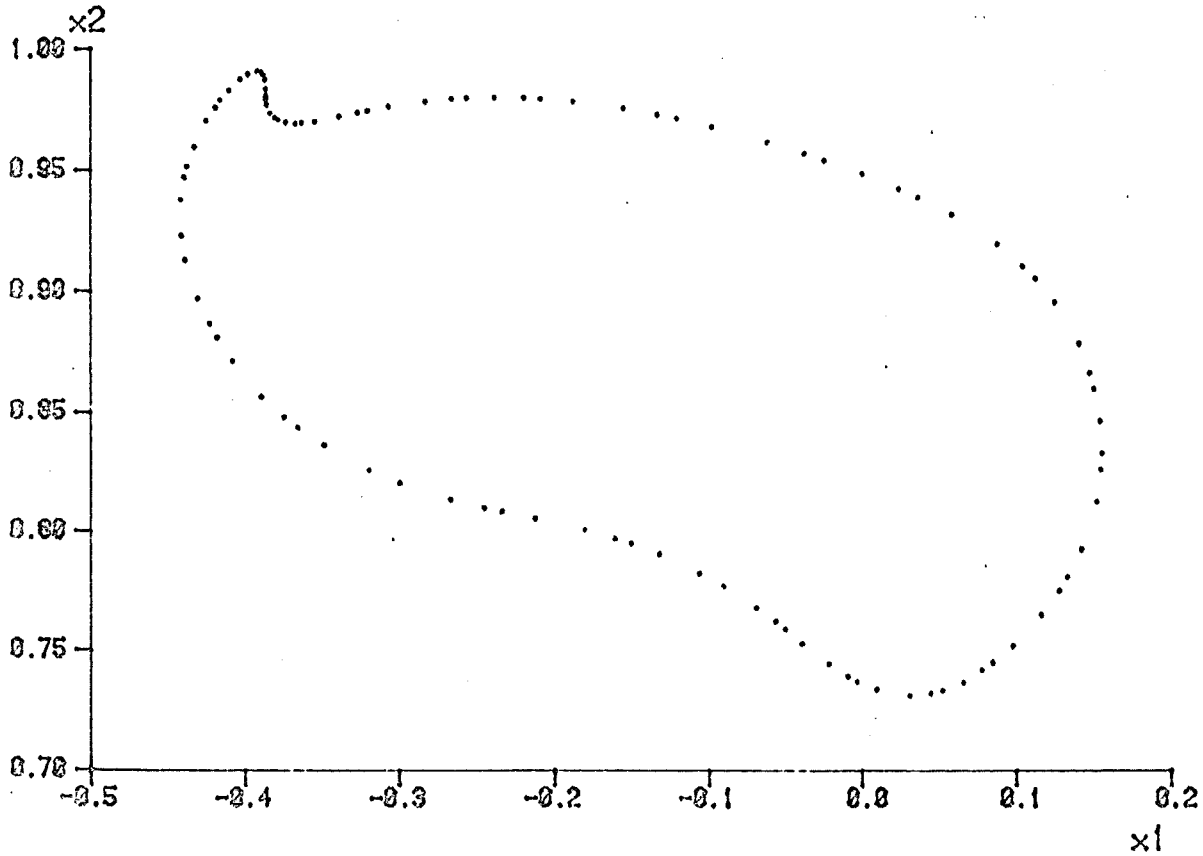


Fig 5b

Chr. Kaas-Petersen

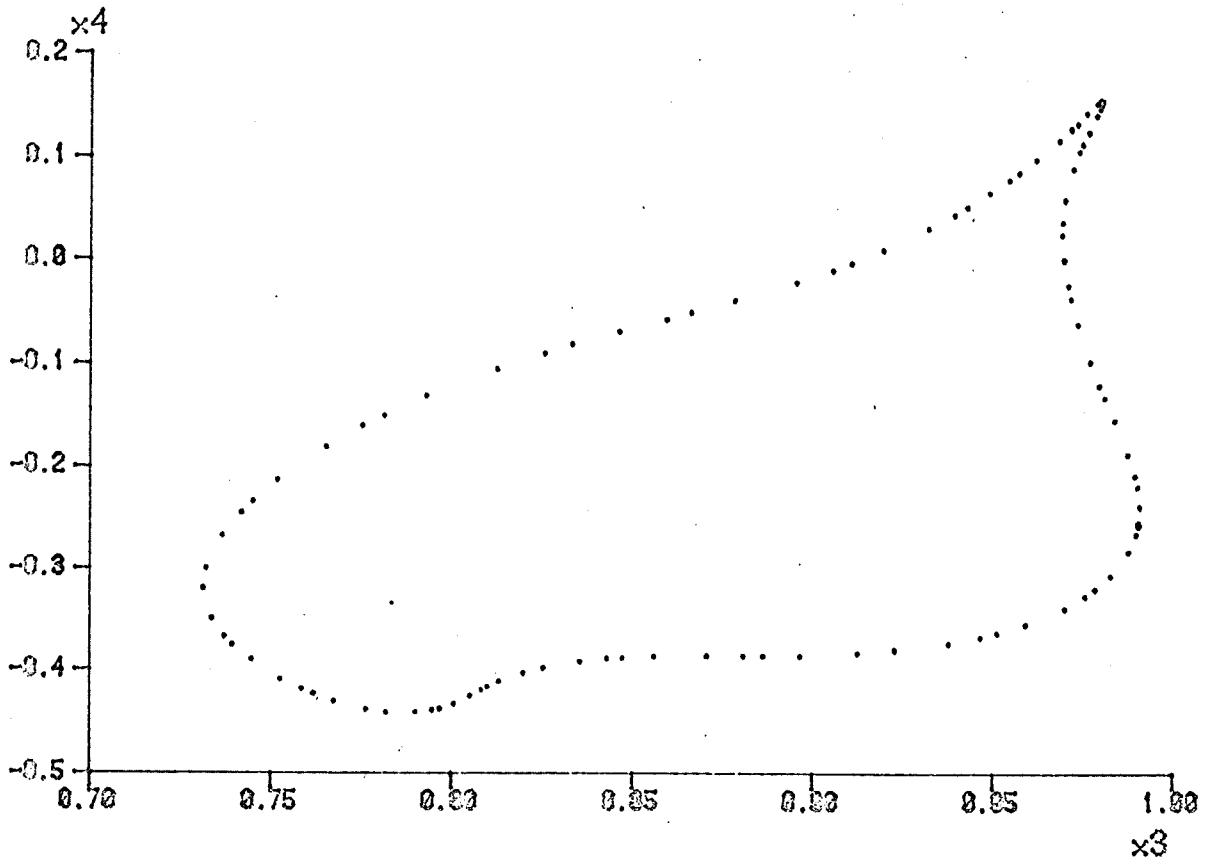


Figure 5

Chr. Kaas-Petersen

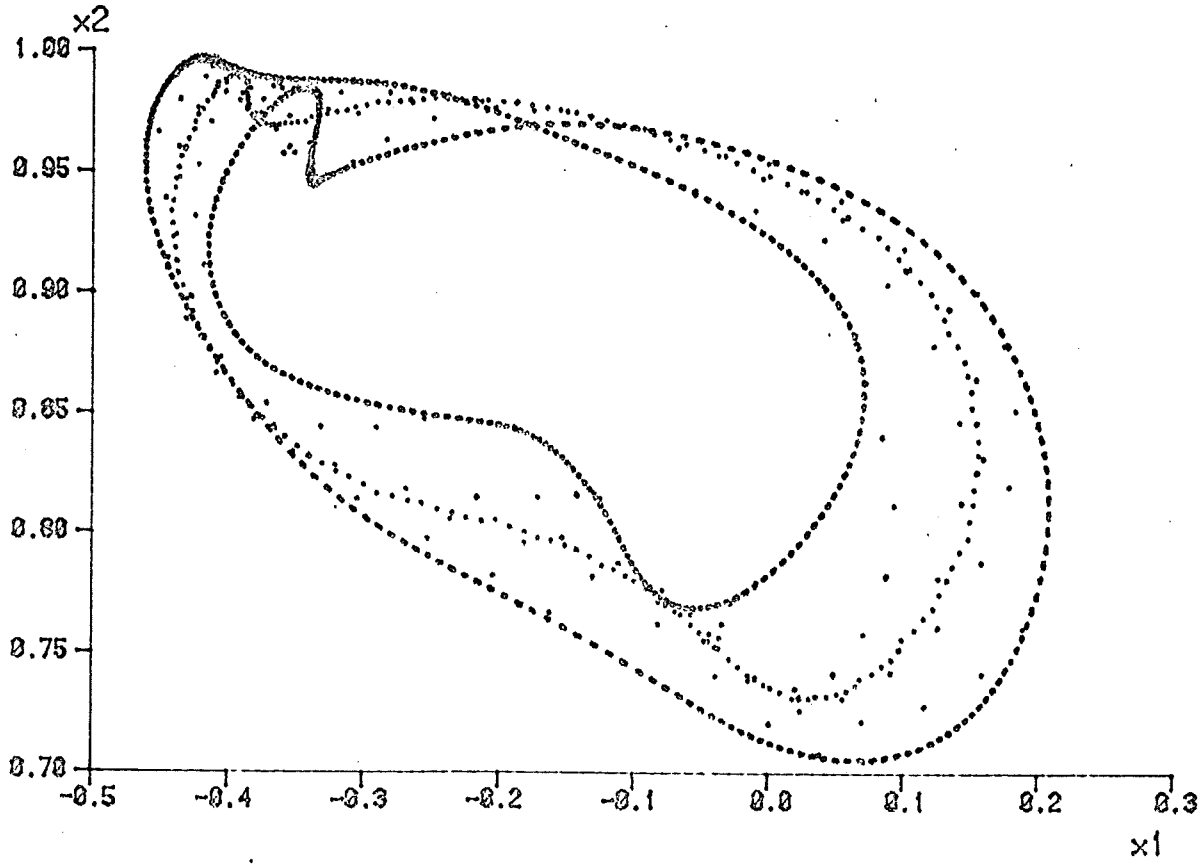
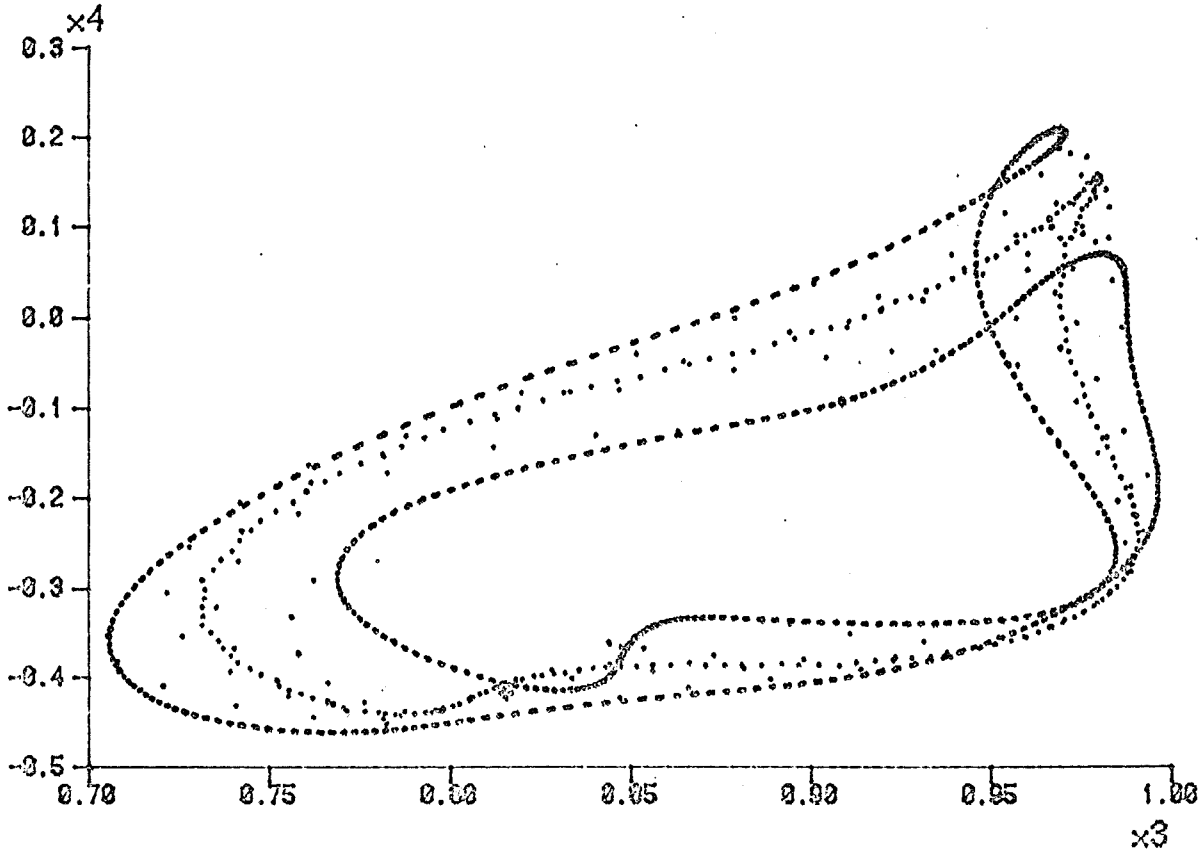


Fig. 5d

Chr. Kaas-Petersen



Continuation methods as the link between perturbation analysis and asymptotic analysis

by

Chr. Kaas-Petersen

Laboratory of Applied Mathematical Physics,
The Technical University of Denmark,
DK-2800 Lyngby, Denmark

24

This manuscript consists of 7 pages (text and figure captions) and 5 pages of drawings.

Running title: Continuation of solutions

Submitted for publication to SIAM Review, Classroom notes.
Version no. 1 : February 4, 1985 .

Abstract

We shall describe a method to trace an implicitly defined curve. First we consider a quadratic map with one control parameter. With this simple example in mind we present a continuation method, which enables us to follow an implicitly defined curve. Then we apply the method to follow periodic solutions of the Rayleigh oscillator.

Introduction

We shall solve equations of the form

$$Q(x;c) = 0 \qquad (*)$$

where $Q : R^1 \times R \to R^1$ is a non-linear smooth map.

Perturbation theory may be applied to obtain an approximate solution $x^p(c)$ of (*) valid for $0<c<<1$. Asymptotic theory may yield an approximate solution $x^a(c)$ of (*) valid for $1<<c$. The gap on the parameter axis can be filled either if we know the exact solution $x^e(c)$ of (*) or if we can compute coordinate-sets $(x_p;c_p)$, that within computing accuracy are zeros of Q. We shall formulate a simple algorithm for the second alternative.

In example 1 below the map Q is explicitly known. In example 2 the map Q is numerically known, since we must solve a system of non-linear autonomous ordinary differential equations.

Example 1

We shall study (cf. [3]) the zeros of the non-linear quadratic map

$$Q:R^1 \times R \to R^1 , \quad Q(x;c) = x^2 + cx - 1 .$$

In the halfspace $0 \leq x$ we have the exact solution (i.e. closed form expression)

$$x^e = (-c + \sqrt{4 + c^2}) / 2 .$$

This solution, valid for all c, is drawn in Figure 1. Perturbation analysis for $0<c<<1$ gives

$$x^p = 1 - \frac{1}{2}c + \frac{1}{8}c^2 + 0\cdot c^3 + O(c^4) .$$

In Figure 1 are shown $x_1^p = 1 - \frac{1}{2}c$ and $x_3^p = 1 - \frac{1}{2}c + \frac{1}{8}c^2$; only valid for small c. Asymptotic analysis for $1<<c$ gives

$$x^a = c^{-1} + 0\cdot c^{-2} - 1\cdot c^{-3} + O(c^{-4}) .$$

In Figure 1 we have shown $x_1^a = c^{-1}$ and $x_3^a = c^{-1} - c^{-3}$; only valid for large positive c.

If we instead had to study the quartic map

$$Q:R^1\times R\to R^1 , \quad Q(x;c) = x^4 + cx - 1 ,$$

we may still perform a perturbation analysis and an asymptotic analysis but the exact solution is not available. In order to glue the two approximate solutions together we can use a continuation method (see [1]), which we describe now.

Continuation method

We shall consider a non-linear smooth map $Q:R^1\times R\to R^1$,

$$Q = Q(y) = Q(x;c) , \quad y = (x;c) .$$

We shall refer to x as the variable and to c as the (control) parameter. The equation

$$Q(x;c) = 0$$

defines implicitly a curve. That curve we denote the 'path'.

Let y_p be a point on the path, i.e. $Q(y_p) = 0$. The gradient of Q, $\nabla Q = (\partial Q/\partial x , \partial Q/\partial c)$, points in the direction of most rapid change of Q. But Q is constantly zero on the path, so the tangent of the path t_p in the point y_p is orthogonal to $\nabla Q(y_p)$. Let e_p be a unitvector parallel to the gradient and let t_p be a unitvector; cf. Figure 2. Take a step of length h_p along the tangent of the path and reach

$z_p = y_p + h_p t_p$. Now consider the map Q restricted to the line through z_p with e_p as direction vector. The restricted map is given by

$$\hat{Q}: R^1\to R^1 , \quad \hat{Q}(u) = Q(z_p + u\cdot e_p) .$$

The zero of $\hat{Q}$ is a new point on the path.

We cannot obtain $Q = 0$ on a computer. We must use $|Q| \le \varepsilon_Z$. The choice of ε_Z depends on how accurately we can compute values of Q. In example 1 above we could take $\varepsilon_Z = 100\cdot\varepsilon_M$ (where ε_M is the largest number on the computer, for which $1 + \varepsilon_M = 1$).

In order to find the zero of $\hat{Q}(u)$ we can use Newton's method, i.e.

$$u_{new} = u_{old} + \Delta u , \quad \text{where } D\hat{Q}(u_{old})\cdot\Delta u = -\hat{Q}(u_{old}) .$$

The derivative $D\hat{Q}$ can be obtained with a forward difference approximation, that is

$$D\hat{Q}(u) = [\hat{Q}(u+s) - \hat{Q}(u)] / s = [Q(z_p+(u+s)\cdot e_p) - Q(z_p+u\cdot e_p)] / s ,$$

and in general the steplength $s = \sqrt{\varepsilon_Z}$ will work. Advanced techniques to choose ε_Z and s are given in [2]. The iteration continues until $|\hat{Q}| \le \varepsilon_Z$.

We reached z_p with a step of length h_p. h_p can be determined from the requirements

(i) $0 < h_p \le h_{max}$ (ii) $|Q(y_p+h_p t_p)| \le \varepsilon_p$, $0 < \varepsilon_p << 1$.

The value of ε_p must reflect how close we must remain to the path in order to have guaranteed convergence of Newton's method.

If our initial point y_0 is not on the path (this can happen if we use a perturbation result to generate y_0) then we first find a zero of $\hat{Q}$, where e_p is a unitvector parallel to the gradient of Q in y_0 (or just simply $e_p = (1,0)$).

The algorithm is summarized in the box below.

Algorithm

1) Initialize the quantities ε_Z , ε_P , and h_{max} , and let $y_0 = (x_0;c_0)$ be given.

2) If $|Q(y_0)| \le \varepsilon_Z$ then take $y_P = y_0$ and go to 3).
If $\varepsilon_Z < |Q(y_0)| \le \varepsilon_P$ then take $z_P = y_0$, and let e_P be a unit vector parallel to $\nabla Q(y_0)$. Go to 4).
If $\varepsilon_P < |Q(y_0)|$ then a better initial guess y_0 is required.

3) Compute $\nabla Q(y_P)$, e_P and t_P . Take a step of length h_P along the tangent of the path, and let $z_P = y_P + h_P t_P$.

4) Apply Newton's method to $\hat{Q}(u) = Q(z_P + u \cdot e_P)$ and return to the path.

5) Redefine y_P to the zero just found. If more points are wanted, then go to 3).

Example 2

We shall study periodic solutions of the Rayleigh oscillator

$$\ddot{u} + c(\dot{u}^3 - \dot{u}) + u = 0 , \quad (\dot{}) \equiv d(\,)/dt .$$

Let T denote the period, and u has maximal value when $\dot{u}=0$. For $0 < c \ll 1$ perturbation analysis (see [4] p. 130) yields

$$T \simeq 2\pi , \quad u_{max} \simeq 2/\sqrt{3} .$$

For $1 \ll c$ the periodic solution becomes a relaxation oscillation. Analysis of that oscillation (see [4] p. 142) yields

$$T \simeq (3 - 2 \cdot \ell n 2)c , \quad u_{max} \simeq \frac{2}{3\sqrt{3}} c .$$

The exact solution is not known. We shall follow the periodic solution starting at a small value of c .

Let $x = u$, $y = \dot{u}$, then Rayleigh's equation can be written

$$\dot{x} = y$$
$$\dot{y} = -c \cdot (y^3 - y) - x \qquad (**)$$

For any $0 < x_0$, $0 = y$, and $0 < c$ arbitrary but fixed, we let $(\varphi(t;x_0) , \psi(t;x_0))$ be the solution of (**) with initial condition $(x_0,0)$ at time $t = 0$. Let $0 < T_R$ be the smallest value of t for which

$$0 < \varphi(T_R;x_0) \quad \& \quad 0 = \psi(T_R;x_0) .$$

T_R is called the return time; cf. Figure 3. We define the residual $q = \varphi(T_R;x_0) - x_0$. Then we have defined a function Q , that to any input values of the variable x_0 and the parameter c , gives the output value q . Thus $Q(x_0;c) = q$, and the the equation

$$Q(x_0;c) = 0$$

defines implicitly a path of periodic solutions. Q is called the residual map.

Note that the period is determined implicitly. Since T_R depends on x_0 and c , one solves the two differential equations numerically in time until $0 < \varphi$ and $0 = \psi$. When x_0 is the initial condition for the periodic solution, then T_R is the period T . Furthermore x_0 is the maximal value of u in (**).

Computationally we stop integration when $|\psi| \le \varepsilon_R$. When the local error bound of the differential equation solver is τ we use $\varepsilon_R = \tau/100$. We used the solver [5] with $\tau = 10^{-8}$ and it turned out that $\varepsilon_Z = 10 \cdot \tau$ was suitable.

We started the continuation method for $c = 0.1$ where $x_0 = 2/\sqrt{3} = 1.1547...$ is close to the periodic solution; actually we found $x_0 = 1.15572266$ on the orbit, and the period $T = 6.287111222$ close to 2π . We followed the path to $c = 10.0$. The results are shown in Figure 4.

For large values of c the period is approximated by ([4] p. 146)

$$T = 1.6137\, c + 7.0143\, c^{-1/3} - 0.3333\, c^{-1} \ell n\, c - 1.3246\, c^{-1} + O(c^{-4/3}) .$$

This four-term expression of the period is also shown in Figure 4.

Conclusions

We have studied the zeros of two non-linear maps $Q: R^n \times R \to R^n$, $n=1$. In the first example the map Q was explicitly known and the path of zeros of Q could be obtained in explicit form. In the second example the map was numerically known and the path of zeros was obtained numerically. The continuation method described here can easyly be generalized to maps with $n > 1$.

Acknowledgement

Many thanks to my friend and colleague Dr. Morten Brøns, Mathematical Institute, for his constructive criticism of several drafts.

References

[1] C.B. Garcia and W.I. Zangwill: Pathways to solutions, fixed points, and equilibria, Prentice-Hall, 1981.

[2] P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright: Computing forward-difference intervals for numerical optimization, SIAM J.Sci. Stat.Comput., Vol.4, No.2, 1983, pp. 310-321.

[3] A.H. Nayfeh: Introduction to perturbation techniques, Wiley, 1981.

[4] A.H. Nayfeh and D.T. Mook: Nonlinear oscillations, Wiley, 1979.

[5] L. Petzold: Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations, SIAM J.Sci.Stat. Comput., Vol.4, No.1, 1983, pp. 136-148.

Figure captions

Figure 1. Path of zeros of the quadratic map. Note, that the abscissa has $0 \le c \le 2$ to the left and $\frac{1}{2} \le \frac{1}{c} < 0$ to the right.

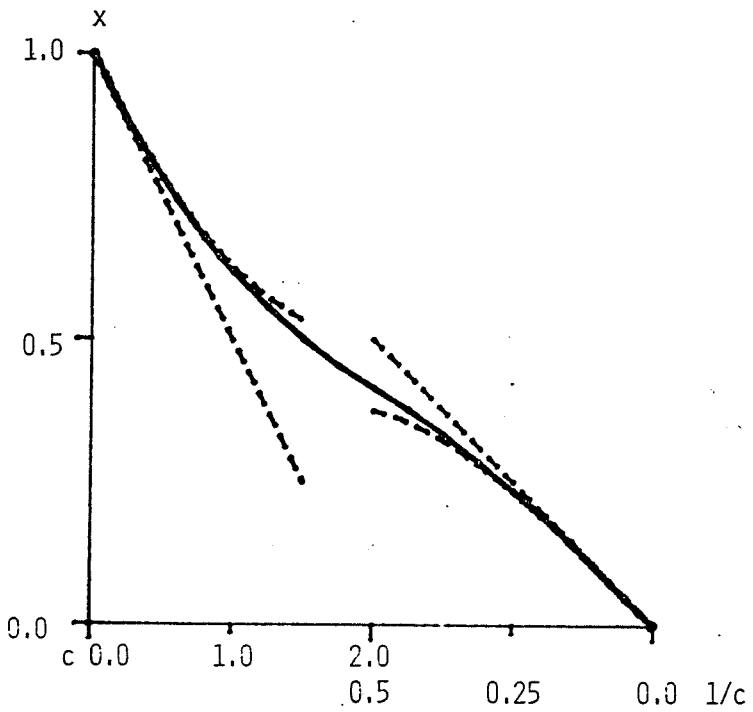
Figure 2. The path is defined by $Q(x;c) = 0$. y_p is a known point on the path, and the gradient of Q in y_p is orthogonal to the path. t_p is tangent to the path.

Figure 3. Phase plane (x,y) with one solution curve shown. At time $t = 0$ the orbit starts in $(x_0,0)$ and time increases along the orbit; at time $t = T_R$ the orbit has returned to the x-axis. The parameter c has fixed value.

Figure 4. Initial value x_0 and period T of the periodic orbits of Rayleigh's equation. The approximate analytical results are shown with dashed lines. In Figure b both the one term approximation (dashed line) and the four term approximation (dashdotted line) of T for $1 \ll c$ are shown.

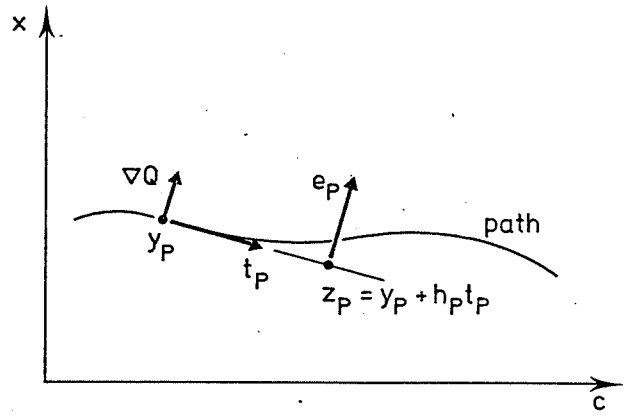
Chr. Kaas-Petersen

Figure 1

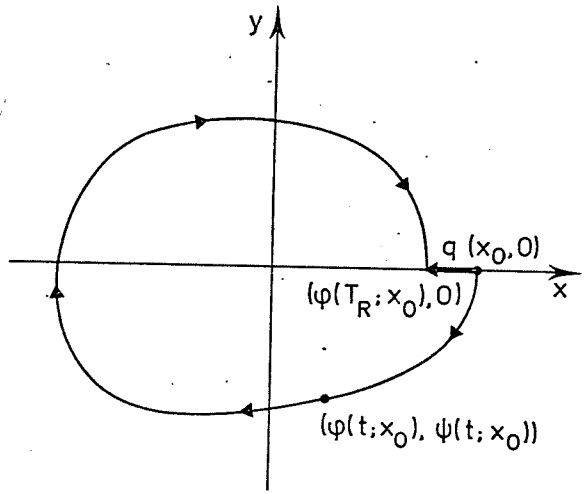


Chr. Kaas-Petersen

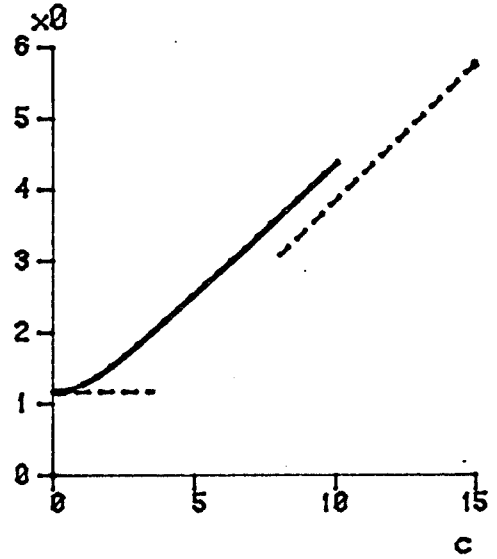
Figure 2



Chr. Kaas-Petersen
Figure 3

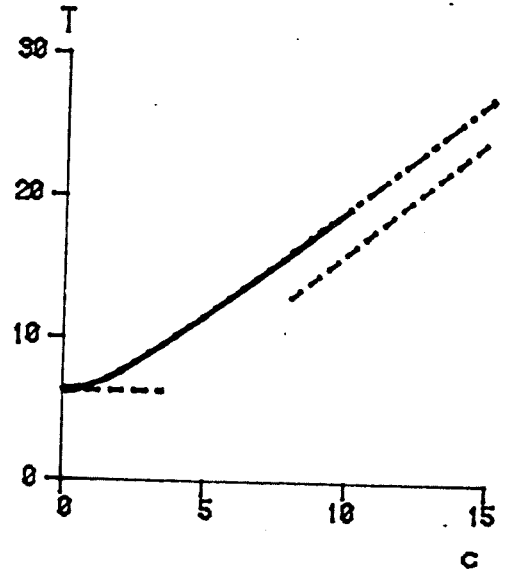


Chr. Kaas-Petersen
Figure 4a



Chr. Kaas-Petersen

Figure 4b



May 1985

PATH - User's Guide

Chr. Kaas-Petersen

25

Laboratory of Applied Mathematical Physics,
Building 303,
The Technical University of Denmark,
DK-2800 Lyngby,
Denmark.

Contents

Words of thanks

The package PATH has been developed during the last three years. It all began with computational methods for the Hopf bifurcation described in Hassard, Kazarinoff, and Wan (1981). Their program BIFOR2 was the first step. BIFOR2 performs local bifurcation analysis of autonomous ordinary differential equations. The first step was to follow the bifurcating periodic solution. The next step was to follow the stationary solutions, so that I could find the points where Hopf bifurcation took place. Since periodically forced ODEs and difference equations may have periodic solutions, they were also included. Periodic solutions may bifurcate to a torus. Then I developed a theory of torus solutions (bi-periodic solutions) which made it possible to compute them. In the present version, there may still be room for small modifications in the part where torus solutions of autonomous ODEs are considered. Computation of chaotic behaviour is also supported.

The development of PATH was speeded up when, firstly, I got a terminal in my office; secondly, when we got a full screen editing facility. I have worked in the IBM VM/CMS environment, and I have had a IBM 3278.3 terminal.

Many persons have influenced the work. I thank M.Sc. Chresten Meulengracht, LAMF, for carefully reading the whole manuscript and gave many valuable suggestions for this Guide. Among the users of PATH, I thank Dr. Morten Brøns, M.Sc. Morten Fordsmann, Dr. Steen Rasmussen, and Dr. Hans True.

1.1 Is PATH for you?

PATH is a tool. It can be used to examine dynamical systems. Such a system may change with time t. A given state x (a vector of state variables) of the system may change with time. The way in which the state and the system change is qualitatively given by the type of the dynamical equations. PATH can treat two types:

$$x(t+1) = f(t,x)$$

$$dx/dt = f(t,x) ,$$

i.e. difference and differential equations respectively. The dynamical system is quantitatively known, when the function expression $f(t,x)$ is known. The dynamical system is said to be non-linear when f is non-linear in the variable x.

The dynamical equations determine how the state x depends on time t, thus $x = x(t)$. PATH can find the initial state at $t=0$ of x - i.e. $x(0)$ - for stationary, periodic, and bi-periodic solutions provided you can give an approximation to $x(0)$. How good the approximation shall be depends on the problem examined.

When the dynamical system depends on a control parameter c (thus we have $f(t,x;c)$) then the initial condition $x(0)$ of a stationary, a periodic, or a bi-periodic solution will depend on c, i.e. $x(0) = x(0)(c)$. The curve $x(0)(c)$ is denoted the path. Such a path of initial conditions can be computed with PATH.

The robustness of a solution to small perturbations in the initial state $x(0)$ is determined by a stability analysis. Whenever a solution has been determined, PATH performs a stability analysis.

The robustness of a solution to a small perturbation in the parameter c is determined by a sensitivity analysis. In order to follow the path as close as possible, PATH performs a sensitivity analysis each time it takes a step along the path.

Paths of initial conditions x(0)(c) of solutions may cross each other or branch off from each other. This takes place in bifurcation points. When bifurcation points are found, PATH performs a bifurcation analysis.

Dynamical systems may have chaotic solutions. Included in the package is CHAOS. CHAOS is a tool. It can be used to compute the maximal Liapunov expontnts of a chaotic solution.

1.2 The systems you can examine; the solutions you can find .

In order to use PATH, you must be able to formulate your system in one of the following ways:

SYS1) 0 = f(x;c)

SYS2) dx/dt = f(x;c) t >= 0

SYS3a) dx/dt = f(t,x;c) t >= 0

SYS3b) x(t+1) = f(t,x;c) t = 0,1,2,3,...

where f(u+T,x;c) = f(u,x;c)

for all u , and T>0

SYS4a) dx/dt = f(t,t,x;c) t >= 0

SYS4b) x(t+1) = f(t,t,x;c) t = 0,1,2,3,...

where f(u+T, u' ,x;c) = f(u,u',x;c)

f(u ,u'+T',x;c) = f(u,u',x;c)

for all u , u' , and T'>T>0.

Furthermore x = (x(1),...,x(n)) belongs to R**n , c belongs to R , f is smooth, and the periods T and T' are independent of c . For SYS3a/4a T and T' can be any real numbers; for SYS3b/4b T must be a positive integer and T' can be a real number.

We shall denote the systems in this way:

SYS1 :	Map	
SYS2 :	Autonomous	)
SYS3a :	Periodically forced	) system of first order ordinary differential equations
SYS4a :	Bi-periodically forced	)
SYS3b :	(Periodically) forced	)
SYS4b :	Bi-periodically forced	) system of first order difference equations

These systems are said to be non-linear, whenever the function

f is non-linear in the x - variable.

You can find these types of solutions:

SOL1 : Zeros of SYS1

SOL2 : Stationary solutions of SYS2

SOL3 : Periodic solutions (of any period) of SYS2,SYS3a/b

SOL4 : Bi-periodic solutions (of any two periods) of SYS2, SYS3a/b, SYS4a/b

SOL5 : Chaotic solutions of SYS2, SYS3a/b, SYS4a/b

It is not the purpose here to describe the theory of linear stability, path followers or dynamical systems. For a short presentation of path following methods we refer you to Kaas-Petersen (1985a), and for the theory of periodic and bi-periodic solutions you should consult Kaas-Petersen (1985b,c,d) where more references on the subject can be found.

2.1 What you must do

PATH does not work by itself. You must put it to work. In order to make PATH applicable in as many situations as possible you must formulate your system in a standard way, namely as SYS1 - SYS4b . You can consult the examples of chapter 3 to see how that can be done in some specific cases.

The program is written in fortran, and therefore you must also program in fortran. In the examples you can see, what you need to do.

The program is available to all users under VM/CMS running on NEUCC, The Technical University of Denmark, and the program is on the A-disk of AMFCKP.

The purpose of this chapter is twofold: Firstly to describe in more detail what PATH can do. Secondly to pose and to answer the questions in the sequence they show up to you. Thus the chapters 2.x for 1 <= x <= 16 should be consulted any time you want to study a new system.

In the file REF PROGRAM you can find a main program and a subroutine FCN, which you can use to shape the program for your own problem.

2.2 So here is how to get started

Your program must have a CALL to the subroutine PATH:

```
      CALL PATH(FCN,N,X,TPER,TQUA,TP,ITP,NP,INP,NT,INT,EVCRIT,
     *          JSYS,JORB,JOB,IPRINT,IER,W,IW)
      IF (IER.NE.0) STOP
```

All the parameters in the list must be declared at the top of the main program (this example corresponds to N=4):

```
      IMPLICIT LOGICAL*4 (A-Z)
      EXTERNAL FCN
C
      INTEGER IER,N
      REAL*8  X(4),  C,TPER,TQUA
      COMMON /PARAM/ C
C
      INTEGER INP,INT,ITP,JSYS,JORB,JOB,IPRINT,IW
      REAL*8  NP(5),NT(5),TP(5),W(17)
      COMPLEX*16 EVCRIT
```

The variables must be defined before the call to PATH. There is one more way to communicate with PATH, namely through COMMON-areas. There are 4 named COMMON-areas (besides PARAM above):

```
      INTEGER         IODE,LRW
      REAL*8                  TOL,RW(4)
      COMMON /EQSOL/ IODE,LRW,TOL,RW
C
      INTEGER                    MTH,ITNWTN
      REAL*8          DELTA,EPSZ
      COMMON /DIFFC/ DELTA,EPSZ,MTH,ITNWTN
C
      INTEGER                     ITRET
      REAL*8          EPSR,DTMAXR
      COMMON /PMAPC/ EPSR,DTMAXR,ITRET
C
      INTEGER         ITPATH,NSTEP
      REAL*8          HMIN,HMAX,EPSH,SEND,FACINC,FACDEC,EPSB
      COMMON /PATHC/ HMIN,HMAX,EPSH,SEND,FACINC,FACDEC,EPSB,
     *               ITPATH,NSTEP
```

These variables, if needed, must also be defined before call to PATH. It is the purpose here to walk through the variables in a natural way; that is, assume you have a problem, how shall you code in order to solve it.

2.3 The FCN subroutine

The (dynamical) system is described by a map. The actual function expressions must be programmed in a subroutine. The subroutine can be of any name. Here we choose the name FCN. The subroutine must be of this form

```
      SUBROUTINE FCN (N,T,X,F)
```

N is an integer, the number of variables in X and the number of function expressions in F, and T is the time, X is the vector of state variables and F is the vector of the function values. The parameter C is passed in the named COMMON area PARAM. Note, that T will not appear in the function-expressions of SYS1 and SYS2. In the example below we show how other constants may be defined in the subroutine itself. In this example N=4 .

```
      SUBROUTINE FCN (N,T,X,F)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER N
      REAL*8  T,X(4),F(4)
C
      REAL*8          C
      COMMON /PARAM/ C
C
      REAL*8 B1,B1M1,B2,B2M1
      DATA   B1,B1M1,B2,B2M1/ 0.277237444454756D 00,
     *                        -0.722762555545244D 00,
     *                         0.795505997645123D 00,
     *                        -0.204494002354877D 00/
C
      F(1) = X(2)
      F(2) = C*X(2)*(  1.0D0-B1*X(1)+B1M1*X(2)  )
      F(3) = X(4)
      F(4) = C*X(4)*(  1.0D0-B2*X(3)+B2M1*X(4)  )
C
      RETURN
      END
```

2.4 The main program

The declarations given in Sec. 2.2 must be in the top of the program. When you have written the FCN-subroutine you know the value of N. The array X must have at least N elements and NP, NT, and TP must have at least N+1 elements.

In the main program you must assign the value of N to the variable N. (You know the value of N, and the program must also know it.)

When FCN has been written, JSYS can be defined as follows:

```
JSYS = -2  for SYS4b
JSYS = -1  for SYS3b and SYS1
JSYS =  0  for SYS2
JSYS = +1  for SYS3a
JSYS = +2  for SYS4a
```

You have until now defined the actual system you will consider.

Now you shall tell PATH what type of solution you are looking for. It is done by the indicator JORB, as follows:

```
JORB =  0  for SOL1 and SOL2
JORB =  1  for SOL3
JORB =  2  for SOL4
```

In any case you always have to define the vector X and the parameter C. For periodic solutions the period TPER must be defined. For bi-periodically forced systems the other period TQUA must be defined.

Chaotic solutions (SOL5) of the systems can be studied with the help of the routine CHAOS. CHAOS is described in App. C.

There are two work-spaces: W and RW.

The work-space W must be declared in a statement, say

```
REAL*8 W(58)
```

The work-space W have length IW, (in the case above IW = 58)

where IW must satisfy

```
IW > 3*N*N + 13*N + 4                  if JORB < 2
IW > 3*N*N + 15*N + 4 + NQUA*(N+5)     if JORB = 2
```

The work-space RW must be declared in a statement, say

```
REAL*8 RW(109)
```

The work-space RW have length LRW, (in the case above LRW = 109), where LRW must satisfy

for an IM : LRW > N

for an ODE: see Sec. 2.5

2.5 If you shall solve ODEs, then read this

There are 4 possible ODE-solution-methods.

a) The IMSL-routine DVERK. DVERK uses 5th-6th order Runge-Kutta method with variable steplength. Then IODE = 1 , RW must have at least 9*N+24 elements, and TOL is the local truncation error.

b) The routine LSODA from Lawrence Livermore National Laboratory. It automatically switches between stiff and nonstiff solvers depending on the actual problem. Then IODE = 2 , RW must have at least MAX(17,N+10)*N+44 elements and TOL is the bound on the absolute error.

c) Euler's one-step method. Then IODE = 3 and TOL is the step-length to be used. There is no error-control.

d) Method by Jesper Halding Jensen. Similar to DVERK but of 3rd-4th order Runge-Kutta method with variable step-length. Then IODE = 4 and RW must have at least 7*N+24 elements.

The number of elements by which the work-space RW is declared must be assigned to the INTEGER variable LRW. PATH will always stop, if LRW is too small and give you a message of how many elements are needed.

2.6 If JSYS=0, then read this

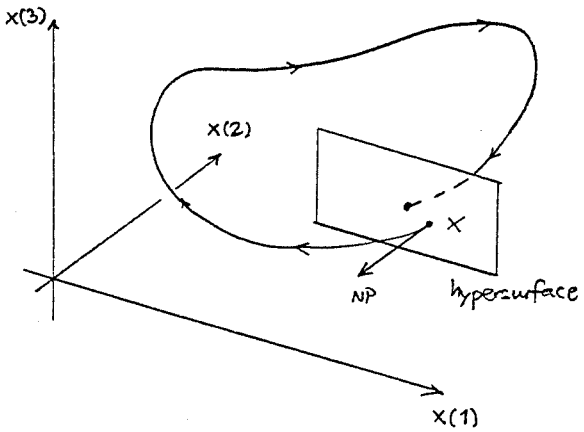
For stationary solutions, there is nothing to say. So go to the next section. For periodic or bi-periodic solutions (JORB=1 or JORB=2), the hypersurface in state space must be defined. The vector NP is the normal of the hypersurface. If the indicator INP = 0 then you have not specified NP, and then PATH itself will make one. If INP = 1 then you have defined NP; this means that you have defined the N first elements of NP. The last element is defined by PATH. NP can be defined with any euclidean length; PATH will always make it a unit vector, and let it point in the same direction as the flow in the initial point.

The named COMMON-area PMAPC contains parameters for hitting the hypersurface. EPSR measures how close you must be to the hypersurface in order to say, that you numerically are in the hypersurface. The default value is TOL/100 . To reach the hypersurface, at most ITRET iterations will be performed, and the maximal time-increment between two iterates is DTMAXR.

Note, that

EPSR < TOL <= EPSZ

See the Figure on the next page.



Orbit of autonomous ODEs dx/dt = f(x;c) . The initial point X at time T=0 is in the hypersurface of the X(1)-X(2)-X(3)-state space. The hypersurface has NP as normal. TPER must be an approximation to the return time. The orbit has returned when the endpoint is closer to the hypersurface then EPSR.

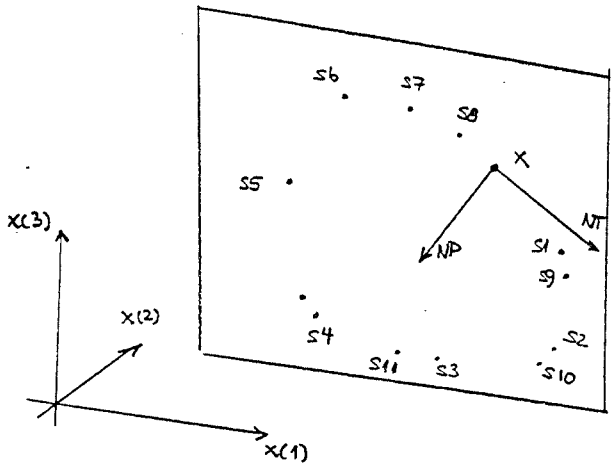
A good approximation to the return time TPER must be known before call to PATH. If a point close to the periodic solution is known then the return time can be found. The routine FTRET serves this purpose. The parameterlist is

```
CALL FTRET (FCN,N,T,X,TEND,NP,INP,X0,IX0,
            DTMAXR,ITRET,EPSR,IER,W,IW)
```

A hypersurface must be defined: its normal is NP and it contains the point X0. If NP is specified INP = 1, else FTRET normalizes the velocity-vector in the initial point. If X0 is specified IX0 = 1, else FTRET takes X0 = X . Any time the orbit starting in (T;X) hits the hypersurface the state vector X is printed. Integration continues until TEND, which should be taken greater than the expected return time. The hitting with the hypersurface is done with a modified regula falsi technique, however at most ITRET iterations are performed, and the routine stops when X is closer to the hypersurface than EPSR. DTMAXR is the time interval between two checks of crossing. IW must be at least N.

The named COMMON area EQSOL must be defined in the program calling FTRET.

For bi-periodic solutions (JORB = 2) the vector NT is the normal of the torus hypersurface. If the indicator INT = 0 then you have not specified NT, and then PATH will make one. If INT = 1 then you have defined the N first elements of NT. PATH normalizes NT to a unit vector. NT must point in the direction the strobes wind around. If that direction is not known, the call XWIND.



Strobed orbit of autonomous ODEs dx/dt = f(x;c) in the hypersurface with normal NP. NT is an approximation to the tangent of the strobed orbit. The orbit of dx/dt = f(x;c) starting in X at T=0 is not shown. S1, S2, S3,... are the strobe numbers. A subset of NQUA strobes is retained between which interpolation is performed.

2.7 If JORB=2, then read this

This section tells you how to cope with bi-periodic behaviour. For any x one computes a number of strobes - a certain number (namely NQUA) of these strobes are interpolated with cubic splines. When JSYS is different from zero you must know which strobes to be used. When JSYS = 0 then you just have to give the value of NQUA.

If the winding number is known you can use YWIND:

SUBROUTINE YWIND (WIND,IQUA,IER)

WIND is the winding number divided by 2*PI. Thus 0 < WIND < 1 . IQUA is the maximal number of strobes considered. Only the strobes crossing the return surface are printed. For JSYS=+2 or JSYS=-2 then WIND = TPER/TQUA . When a periodic orbit bifurcates to a bi-periodic orbit, then BIFUR prints WIND out, and also NT and TP.

When the winding number is not known you must use XWIND:

SUBROUTINE XWIND
(JSYS,FCN,N,X,TRET,NP,NT,IQUA,DMAX,IER,W,IW)

NP and NT must be specified unit-vectors orthogonal to each other. The vector NT is explained below in this section. W is of length IW, where IW >= 2*N . X must be an initial condition on the torus and TRET is the return time. Only the strobes closer to the return surface than DMAX are printed. XWIND stops when IQUA strobes have been determined.

Before call to PATH or DOFQ you must specify the number of strobes (=NQUA) between which interpolation must be performed. NQUA appears in its own named COMMON area QUASI:

```
INTEGER        NQUA
COMMON /QUASI/ NQUA
```

When JSYS is different from zero then the NQUA first elements of the work-vector W must be given the times for each of the NQUA strobes. When JSYS = 0 only NQUA must be specified. For JSYS = 1 or JSYS =-1 the winding number can change with the parameter c along the path. Therefore it may be necessary with new stop-times now and then.

When a bi-periodic solution bifurcates from a periodic solution the winding number is known, and so YWIND should be used to get the stop-times for the torus.

If you have JSYS = 2 or JSYS =-2, then TQUA is known and must be defined.

When JSYS is -1, 0, or +1 one needs the torus return hypersurface. The N first elements of the vector NT is the normal of the torus hypersurface. If you have specified NT, then the indicator INT = 1, otherwise INT = 0 .

2.8 Parameters for the numerical differentiation

You must tell, how the routine must work in order to reach the solution you want. First of all, how accurate can the solution be computed? If you solve ordinary differential equations you will normally not be able to obtain more accuracy than TOL. For iterated maps you may reach machine precision, but for bi-periodic solutions it will be the interpolation which determines the accuracy. EPSZ must be defined to be the required accuracy. The routine DOFQ (App. D) can supply you with a proper value, say EPSZ = 10*EPSA , where EPSA is found by DOFQ.

If you want to reach the solution in a brute force manner (iterate until convergence) take MTH = 0 ; however this is only allowed if JORB=1 or JORB=2. If you want Newton's method take MTH = 1 or if modified Newton's Method let MTH > 1 . If say MTH=4 the derivative of the map is only computed in each fourth iteration. The maximal number of iterations allowed is ITNWTN , and the derivative is approximated with a forward difference computed with the step-length DELTA . DELTA can be found with the routine DOFQ (App. D), where it is denoted HOPT .

2.9 How much of the solution do you want?

JOB indicates the amount of computations to be performed, and IPRINT the amount of output (to logical unit 6).

The print indicator is as follows:

IPRINT=1 only print when PATH stops

IPRINT=2 print after each step performed

IPRINT=3 print results for each Newton iteration.

The value of JOB is found in the table below:

You have	You want	You must
(X,C) is a bifurcation point	A bifurcation analysis	JOB=1 and EVCRIT must be an appriximation to the critical eigenvalue
	Find only one point on the bifurcating path	JOB=2. TP must be specified (and also NP for autonomous systems and NT for torus solutions)
	Find one or more points on the bifurcating path	JOB=3. TP must be specified (and also NP for autonomous systems and NT for torus solutions)
(X,C) is not a bifurcation point, but on the path	Find one or more points on the path	JOB=4
(X,C) is not a bifurcation point, and not on the path	Find one point on the path	JOB=5. NP, NT, and TP can be specified
	Find one or more points on the path	JOB=6. NP, NT, and TP can be specified

2.10 If you will follow the path, you must read this

The path is a curve in the state-parameter space. The tangent of the path is stored in the vector TP , where the N first elements are in the state space and the N+1 element is in the parameter space. If you specify TP then the indicator ITP must be 1 . (TP will be normalized to a unit vector.) If you do not specify a complete vector, you define ITP = 0 . Anyway, you must indicate a direction in which to follow the path. If the parameter C is to be increased then TP = (0,...,0,+1) , and if C is to be decreased, then TP = (0,...,0,-1) .

TP is always normalized to a unit vector. HMIN is the minimal step-length to be used along the path, HMAX is the maximal step-length allowed. When we follow the path, a step is taken, such that the norm of the residual map is less than EPSH. This ensures that we do not drift too far away from the path. If the bound EPSH is satisfied in the first try a longer step will be tried. FACINC is the factor, by which the actual step-length is scaled up. If EPSH is not satisfied in the first try a smaller step is tried. FACDEC is a factor, that down-scales the step-length. The default values are FACINC = 2 and FACDEC = 0.5 . At most ITPATH up or down-scalings are performed before return to the path.

At most NSTEP complete steps are performed, and at most a distance SEND along the path is traversed. If a bifurcation point is met it is located on the path within a distance of EPSB, and PATH stops.

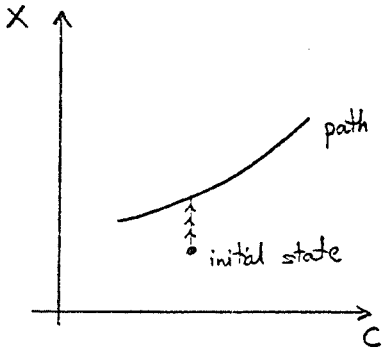
Note, that

EPSZ <= EPSB <= HMIN < HMAX

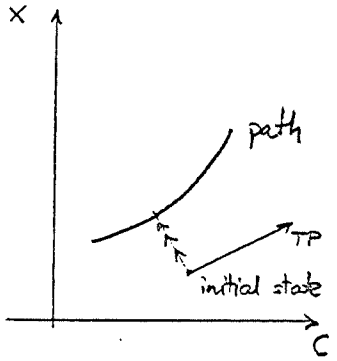
HMIN <= SEND

EPSB <= EPSH

If you want to follow the path accurate, EPSH is the essential parameter to modify. A small value of EPSH will usually require a small value of HMIN.



When JOB=5 and ITP=0
Newton's method brings you to the path keeping the parameter C fixed



When JOB=5 and ITP=1
Newton's method brings you to the path in a direction orthogonal to the vector TP.

2.11 Advices for the first try

In the first run with a new set of equations, you must not expect too much. So to save somputer-time make sure, that the program will stop. Therefore: JOB=5 , ITNWTN=2 , MTH=1 . To be sure, that something has been done, get some output; thus let IPRINT=3 .

The amount of work-space - both RW and W - is checked. Therefore you can always put LRW=1 and IW=1 in the first run. Then PATH will stop because of too little work-space, but it stops with a message of how many locations the two arrays must have.

IER is an error indicator. If IER=0 then PATH has performed succesfully. If IER0 then a trace back is printed, so that you can see, where the error occured.

2.12 When a bifurcation point says Hello to you ...

PATH follows that path of solutions you specify. You just have to start either on the path or close to it. When you are on the path you can always look around you in the state space to see, if another path exists close to you. This is the essence of the stability analysis, i.e. the location of the eigenvalues in the complex plane.

Bifurcation takes place in a point on the path, where the stability properties of the solution change. What bifurcates is another path. The two paths may correspond to the same type of solutions - this is the case when it is a real eigenvalue that changes the stability of the solution; the two paths may correspond to different types of solutions - this is the case when it is a complex conjugated pair of eigenvalues that changes the stability of the solution.

PATH stops when a bifurcation point has been located. The bifurcation-stop-condition is as follows: when PATH has found that two points on the path (on each side of the bifurcation point) are closer to each other than EPSB. Then the routine BIFUR makes a bifurcation analysis.

BIFUR prints relevant data to follow the path starting in the bifurcation point. Then you must have JOB=2 or JOB=3.

Some of the bifurcations have got special names:

Hopf-bifurcation: Bifurcation of a periodic solution from a stationary solution in an autonomous ODE.

Period-doubling-bifurcation: Bifurcation of a 2T-periodic solution from a T-periodic solution.

Torus-bifurcation: Bifurcation of a bi-periodic solution

from a periodic solution.

Saddle-node-bifurcation: When the path of solutions bends around when projected on the parameter axis.

2.13 Output from PATH

The values assigned to the variables in the named COMMON areas are written at the top. If you have specified improper values, then PATH will modify these. It is the modified values which are printed. At the left the name of the COMMON area is written . Below the variable name, the value is printed.

When IPRINT=1 output is made for the last computed point on the path.

When IPRINT=2 output is made for each computed point on the path.

When IPRINT=3 output is made for each iteration step in Newton's method.

A horizontal bar

==========...

is written whenever a point on the path has been determined. The values printed are these:

C	the parameter value	) (corresponding to the solution
TPER	the period	) (of type JORB for a system
TQUA	the second period	) (of type JSYS.
X	the state vector	) (

LAMDA eigenvalues determining the stability; first the real part is written followed by the imaginary part.

ITNWTN the number of Newton iterations used.

There are some data from the path following process:

COSNP, COSNT, COSTP are the cosines of the angles between the NP-, NT-, and TP-vectors and the optimal one. These cosines should be as close to 1 as possible. Whenever one of these becomes less than 0.95, new NP-, NT-, and TP-vectors are computed. These new vectors are outputted.

H the steplength taken along the path. HMIN < H < HMAX .

ITPATH the number of times the steplength H has been increased (if positive value of ITPATH) or decreased (if negative value of ITPATH).

S the accumulated arclength; the arclength traversed until now.

When IPRINT=3 then the line

NEWTON ==========...

means that below follows the results of that Newton iteration. IAUG=0 when the parameter C is kept fixed in the iterations. Else IAUG=1. In fact IAUG=ITP. MTH is the Newton-method used. NIT is the iteration number, X is the state vector at time zero, and Q is the residual. Newton's method tries to make Q equal to the zero vector.

**** PATH **********...

means that we have found the first point on the path, such that we are able to follow the path, which is performed in the routine PATHX.

BIFUR ==========...

signals, that a bifurcation analysis has been performed.

For periodic solutions, the maximal and the minimal values of each coordinate on the periodic orbit are printed. The

differences between the maximal and the minimal values are the amplitudes. Only NPOINT points on the periodic orbit are considered. There is the time-interval DT between each point.

2.14 How to visualize the solutions

When a periodic, bi-periodic or chaotic solution has been determined, you may be interested in a picture of that solution. In order to plot say X(1)(T), then a set of points (Ti,X(1)(Ti)) , i=0,1,...,NPOINT must be obtained. For that purpose the routine GRXOFT (= Graph X of T) can be used. The format of the output must be written in a special subroutine OXOFT (Sec. 3 Ex.7), which along with FCN is passed to GRXOFT as a parameter. OXOFT and FCN must appear in an EXTERNAL statement. The parameterlists are

```
GRXOFT(FCN,N,T,X,TEND,NPOINT,OXOFT,IDEV,IER)
OXOFT (T,X,N,IDEV)
```

IDEV must hold the number of the output file. When IDEV=27 output is stored in the file named FILE FT27F001 . IDEV must satisfy 10 < IDEV < 100 . The solution is starting at time T in the state X and ends at time TEND. The time-interval between two points is (TEND-T)/NPOINT. GRXOFT uses the two named COMMON areas PARAM and EQSOL.

For a bi-periodic solution you may be interested in a plot of the strobed orbit. For that purpose the routine GRSTRO (=Graph strobed orbit) can be used. The normal NP of the hypersurface must be specified (when JSYS=0), and the initial point T,X must be in that hypersurface. The output is formatted by the routine OSTRO. The parameterlists are

```
GRSTRO(JSYS,FCN,N,T,X,TPER,TQUA,NPOINT,
       NP,INP,OSTRO,IDEV,IER,W,IW)
OSTRO (TMOD,X,N,IDEV)
```

TQUA is only used when JSYS=+2 or JSYS=-2. When JSYS=0, then the

named COMMON area PMAPC is used.

2.15 Examples of how to cope with exotic dynamical systems

In the examples in Sec. 3 you can see how to formulate the equations in a proper form. Here in this section we shall give some examples, which at first sight seems not to fit into PATH.

Example A.

If the forcing term is say cos(c*t) and we want to study this system in dependence of the parameter c, then we must make a substitution. The reason is that as c changes, so will the period TPER, but TPER must be kept fixed when PATH is set to work with a periodically forced system. The substitution TAU = t*c will bring the equation in the right form.

Example B.

A general system of second order iterated maps on the form

$$A(t,u,\dot{u})*(\ddot{u}) + B(t,u,\dot{u}) = 0$$

where $\dot{u} = u(t+1)$ and $\ddot{u} = u(t+2)$

can with the substitution x=u, y=$\dot{u}$ be written in this form

$$\dot{x} = y$$

$$\dot{y} = -AINV(t,x,y)*B(t,x,y)$$

where AINV is the inverse of the matrix A. Remember, that the subroutine FCN knows t,x, and y at entry, and shall only - in one way or another - compute the values of $\dot{x}$ and $\dot{y}$. FCN should therefore set up the vector with right hand sides, afterwards set up the A matrix, and then solve the system. The solution of the linear system is then $\dot{y}$.

2.16 Here you may learn how to add subroutines to PATH

You may need some extensions to the program. For instance if you have found a periodic solution you may need its amplitude, or some other characteristic. In the routine PATHX is made an indication of where to put your own routines. That place will be passed each time a solution has been found. However, you must not destroy any of the values generated by PATH - you should copy the relevant variables to your own defined variables before processing your routine. Here you can also make extra output, say generate data for graphical purposes. PATH does not automatically save data to be used in graphical visualizations, neither of the orbits nor of the paths.

You may need special solvers of the ordinary differential equations. Such a routine must be called from the routine ODEX.

3. Examples

In this section we illustrate how to use PATH.

1) Purpose : Periodic solution of iterated map
   Reference: Lauwerier (1983)

2) Purpose : Bi-periodic solution of iterated map
   Reference: Lauwerier (1983)

3) Purpose : Periodic solution of autonomous system of ordinary differential equations
   Reference: Lichtenberg and Lieberman (1983) p. 386-389

4) Purpose : Periodic solution of periodically forced system of ordinary differential equations
   Reference: Ueda (1980)

5) Purpose : Chaotic attractor of Lorenz equation
   Reference: Shimada and Nagashima (1979)

6) Purpose : Computation of derivative

7) Purpose : Visualization of periodic solution
   Reference: Lichtenberg and Lieberman (1983) p. 386-389

8) Purpose : Bi-periodic solution of bi-periodically forced system
   Reference: Chua and Ushida (1981)

9) Purpose : Perform a bifurcation analysis

Reference: Hassard, Kazarinoo, and Wan (1981) p. 156-159

The programs in the examples are all available on the A-disk of AMFCKP. Example no. 1 is in the file named X1PATH FORTVS with the corresponding output in the file named X1PATH LISTING. Example no. 2 is in the file X2PATH ... and so on. In this version of the User's Guide all FORTVS and LISTING files are included.

Example no. 1

Lauwerier has studied pariodic, quasi-periodic and chaotic behaviour of the logistic delay map given by

$$\ddot{y} = a*\dot{y}*(1-(1-b)*\dot{y}-b*y)$$

where $\dot{y} = y(t+1)$ and $\ddot{y} = y(t+2)$

In this map $y=\dot{y}=1-1/a$ is a periodic solution with period 1. The solution is seen to be independent of b. The solution is stable for $a<A$ and non-stable for $A<a$, where $A=1+1/b$. We shall study one of these periodic solutions.

We use the substitution $x(1)=y$, $x(2)=\dot{y}$ and get

$$\dot{x}(1) = x(2)$$

$$\dot{x}(2) = a*x(2)*(1-(1-b)*x(2)-b*x(1))$$

The difference equations have two control parameters. We fix b to the value 0.79550 59976 45123 , and denote a by c. Then

$$\dot{x}(1) = x(2)$$

$$\dot{x}(2) = c*x(2)*(1-(1-b)*x(2)-b*x(1))$$

The right hand sides are written in the subroutine FCN. The right hand side is periodic with period 1, so TPER=1.0 and JSYS=-1. There are two expressions so N=2.

We start at c=1.2 where x(1)=x(2)=0.2 is close to the 1-periodic solution. We will determine that periodic solution, so JORB=1.

We are close to the path of periodic solutions. We just want to find the exact solution on the path, so we take JOB=5, MTH=2, and ITNWTN=2.

Since N=2 the work-space W must have at least 42 locations and RW must have 2 locations. With this knowledge, the program below has been written.

The construction with FIRST in the named COMMON area QQF makes it possible to have all data for the subroutine FCN defined in FCN.

```
      IMPLICIT LOGICAL*4 (A-Z)
      EXTERNAL FCN
C
      INTEGER IER,N
      REAL*8  X(2),  C,TPER,TQUA
      COMMON /PARAM/ C
C
      INTEGER INP,INT,ITP,JSYS,JORB,JOB,IPRINT,IW
      REAL*8  NP(3),NT(3),TP(3),W(42)
      COMPLEX*16 EVCRIT
C
      INTEGER        IODE,LRW
      REAL*8                 TOL,RW(2)
      COMMON /EQSOL/ IODE,LRW,TOL,RW
C
      INTEGER                    MTH,ITNWTN
      REAL*8         DELTA,EPSZ
      COMMON /DIFFC/ DELTA,EPSZ,MTH,ITNWTN
C
      INTEGER      FIRST
      COMMON /QQF/ FIRST
C
      FIRST =0
      JSYS  =-1
      N     = 2
C
      WRITE(6,5)
    5 FORMAT(' PROBLEM DESCRIPTION '/
     *       '  AN ITERATED MAPS OF LAUWERIER')
C
C  ---------------  PARAMETERS FOR SOLUTION OF ODE AND WORKSPACE
      LRW   = 2
      IW    = 42
C  ---------------  PARAMETERS FOR DIFFERENTIATION
      MTH   = 2
      ITNWTN= 6
C
      JORB = 1
      X(1) = 0.15D0
      X(2) = 0.15D0
      C    = 1.2D0
      TPER = 1.0D0
C
      ITP  = 0
C
C  ---------------  HERE WE MAY FOLLOW THE PATH
C
      JOB   = 5
      IPRINT= 3
C
      CALL PATH(FCN,N,X,TPER,TQUA,TP,ITP,NP,INP,NT,INT,EVCRIT,
     *          JSYS,JORB,JOB,IPRINT,IER,W,IW)
      IF (IER.NE.0) STOP
C
      STOP
      END
C
C  ---------------  HERE FOLLOW THE FUNCTION ROUTINE
C
C
      SUBROUTINE FCN (N,T,X,F)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER N
      REAL*8  T,X(2),F(2)
C
      REAL*8         C
      COMMON /PARAM/ C
C
      REAL*8        B,BM1
      COMMON /BVAR/ B,BM1
C
      INTEGER      FIRST
      COMMON /QQF/ FIRST
C
      IF (FIRST.EQ.0) GO TO 10
C
    5 CONTINUE
C
      F(1) = X(2)
      F(2) = C*X(2)*(  1.0D0-B*X(1)+BM1*X(2)  )
C
      RETURN
C
C
C
   10 CONTINUE
      FIRST = 1
C  ***************  HERE ALL PARAMETERS AND CONSTANTS OF THE
C   FUNCTION ARE DEFINED
C
      B    = 0.795505997645123D 00
      BM1  = B-1.0D0
C
C  ***************  END OF PARAMETER AND CONSTANT DEFINITION
C                   HERE WE CAN OUTPUT THE DEFINED PARAMETERS
      WRITE(6,15) B
   15 FORMAT(//' THE VALUES OF THE PARAMETERS IN THIS RUN ARE:'//
     *        '      B =',D24.16)
C
      GO TO 5
C
      END
```

Below is the output file

```
PROBLEM DESCRIPTION
 AN ITERATED MAPS OF LAUWERIER
```

```
================================================================================

    N   JSYS  JORB   JOB   ITP   INP   INT
    2    -1     1     5     0     0     0


IFFC :        DELTA                EPSZ                 MTH      ITNWTN
         0.1000000000D-06     0.1000000000D-13          2          6


ATHC :         HMIN                HMAX                 EPSH                   S
         0.1000000000D-03     0.1000000000D+00     0.3162277660D-03     0.100

            FACINC               FACDEC               EPSB
         0.2000000000D+01     0.5000000000D+00     0.1000000000D-11

           ITPATH    NSTEP
             3         0


HE VALUES OF THE PARAMETERS IN THIS RUN ARE:

   B =  0.7955059976451230D+00


EWTON ==========================================================

      JSYS       JORB       NIT       IAUG       MTH
       -1          1          0          0          2

     CONTROL PARAMETER                  PERIOD
  C  =     0.1200000000D+01       TPER =     0.1000000000D+01

     STATE VECTOR                       RESIDUAL VECTOR
( 1) =     0.1500000000D+00   :   Q( 1) =     0.0000000000D+00
( 2) =     0.1500000000D+00   :   Q( 2) =     0.3000000000D-02
ERIVATIVE COMPUTED


EWTON ==========================================================

      JSYS       JORB       NIT       IAUG       MTH
       -1          1          1          0          2

     CONTROL PARAMETER                  PERIOD
  C  =     0.1200000000D+01       TPER =     0.1000000000D+01

     STATE VECTOR                       RESIDUAL VECTOR
( 1) =     0.1687499971D+00   :   Q( 1) =     0.0000000000D+00
( 2) =     0.1687499971D+00   :   Q( 2) =    -0.4218744119D-03


EWTON ==========================================================

      JSYS       JORB       NIT       IAUG       MTH
       -1          1          2          0          2

     CONTROL PARAMETER                  PERIOD
  C  =     0.1200000000D+01       TPER =     0.1000000000D+01
```



```
     STATE VECTOR                       RESIDUAL VECTOR
X( 1) =     0.1661132825D+00   :   Q( 1) =     0.0000000000D+00
X( 2) =     0.1661132825D+00   :   Q( 2) =     0.1103093604D-03
DERIVATIVE COMPUTED


NEWTON ==========================================================

      JSYS       JORB       NIT       IAUG       MTH
       -1          1          3          0          2

     CONTROL PARAMETER                  PERIOD
  C  =     0.1200000000D+01       TPER =     0.1000000000D+01

     STATE VECTOR                       RESIDUAL VECTOR
X( 1) =     0.1666685163D+00   :   Q( 1) =     0.0000000000D+00
X( 2) =     0.1666685163D+00   :   Q( 2) =    -0.3699278420D-06


NEWTON ==========================================================

      JSYS       JORB       NIT       IAUG       MTH
       -1          1          4          0          2

     CONTROL PARAMETER                  PERIOD
  C  =     0.1200000000D+01       TPER =     0.1000000000D+01

     STATE VECTOR                       RESIDUAL VECTOR
X( 1) =     0.1666666543D+00   :   Q( 1) =     0.0000000000D+00
X( 2) =     0.1666666543D+00   :   Q( 2) =     0.2477027680D-08
DERIVATIVE COMPUTED


NEWTON ==========================================================

      JSYS       JORB       NIT       IAUG       MTH
       -1          1          5          0          2

     CONTROL PARAMETER                  PERIOD
  C  =     0.1200000000D+01       TPER =     0.1000000000D+01

     STATE VECTOR                       RESIDUAL VECTOR
X( 1) =     0.1666666667D+00   :   Q( 1) =     0.0000000000D+00
X( 2) =     0.1666666667D+00   :   Q( 2) =     0.1249000903D-15


**** PATHX ********************************************************************

================================================================================

      JSYS       JORB       JOB
       -1          1          4


     ISTEP     ITPATH     ITNWTN       MTH
       0          0          0          2

CONTROL PARAMETER    C   =  0.1200000000000000D+01
```

```
ERIOD                TPER =  0.1000000000000000D+01

     STATE VECTOR                                EIGENVALUES OF DERIVATIVE
( 1) =  0.1666666666666660D+00  :  LAMDA( 1) =  0.2133D+00  0.0000D+00
( 2) =  0.1666666666666660D+00  :  LAMDA( 2) =  0.7458D+00  0.0000D+00

  COSNP       COSNT       COSTP       H           S
 0.1000D+01  0.1000D+01  0.1000D+01  0.4000D-03  0.0000D+00

------------------------------------------------------------------------

  MESSAGE FROM ROUTINE PATH :

  MAXIMAL NUMBER OF STEPS    0 (=NSTEP)  USED

------------------------------------------------------------------------


========================================================================

      JSYS       JORB       JOB
       -1          1          4

     ISTEP     ITPATH     ITNWTN      MTH
        0          0          0        2

ONTROL PARAMETER     C    =  0.1200000000000000D+01
ERIOD                TPER =  0.1000000000000000D+01

     STATE VECTOR                                EIGENVALUES OF DERIVATIVE
( 1) =  0.1666666666666660D+00  :  LAMDA( 1) =  0.2133D+00  0.0000D+00
( 2) =  0.1666666666666660D+00  :  LAMDA( 2) =  0.7458D+00  0.0000D+00

  COSNP       COSNT       COSTP       H           S
 0.1000D+01  0.1000D+01  0.1000D+01  0.4000D-03  0.0000D+00
```


Example no. 2

We study a bi-periodic solution of the Lauwerier-map from Ex. 1. With b as before we take c=2.26 and initial conditions x1 = x2 = 0.5, and NT = (-1,+1) is a fairly good approximation to the tangent of the strobed orbit. PATH will normalize NT to a unit vector. NT points in the direction the strobes wander.

We do not know the winding time. With the above chosen values we call the routine XWIND to obtain the interpolation. It turns out, that we can use strobes no. 10, 26, 32, 48. The corresponding stop-times are 10.0, 26.0, 32.0, and 48.0 and so NQUA=4. We take EPSZ=10**(-8).

The subroutine FCN is as before, so we just list the main program.

```
      IMPLICIT LOGICAL*4 (A-Z)
      EXTERNAL FCN
C
      INTEGER IER,N
      REAL*8  X(2),  C,TPER,TQUA
      COMMON /PARAM/ C
C
      INTEGER INP,INT,ITP,JSYS,JORB,JOB,IPRINT,IW
      REAL*8  NP(3),NT(3),TP(3),W(74)
      COMPLEX*16 EVCRIT
C
      INTEGER         IODE,LRW
      REAL*8                    TOL,RW(2)
      COMMON /EQSOL/ IODE,LRW,TOL,RW
C
      INTEGER                      MTH,ITNWTN
      REAL*8          DELTA,EPSZ
      COMMON /DIFFC/ DELTA,EPSZ,MTH,ITNWTN
C
      INTEGER         NQUA
      COMMON /QUASI/ NQUA
C
      INTEGER       FIRST
      COMMON /QQF/ FIRST
C
      FIRST = 0
      JSYS  =-1
      N     = 2
```

```
C
      WRITE(6,5)
    5 FORMAT(' PROBLEM DESCRIPTION '/
     *       '  AN ITERATED MAPS OF LAUWERIER')
C
C  ----------------  PARAMETERS FOR SOLUTION OF ODE AND WORKSPACE
      LRW    = 2
      IW     = 74
C  ----------------  PARAMETERS FOR DIFFERENTIATION
      EPSZ   = 1.0D-08
      MTH    = 1
      ITNWTN= 6
C
      JORB = 2
      X(1) = 0.50D0
      X(2) = 0.50D0
      C    = 2.26D0
      TPER = 1.0D0
C
      NQUA = 4
      W(1) = 10.0D0*TPER
      W(2) = 26.0D0*TPER
      W(3) = 32.0D0*TPER
      W(4) = 48.0D0*TPER
C
      INT  = 1
      NT(1)=-1.0D0
      NT(2)= 1.0D0
C
C  ----------------  HERE WE MAY FOLLOW THE PATH
C
      JOB    = 5
      IPRINT= 3
C
      CALL PATH(FCN,N,X,TPER,TQUA,TP,ITP,NP,INP,NT,INT,EVCRIT,
     *          JSYS,JORB,JOB,IPRINT,IER,W,IW)
      IF (IER.NE.0) STOP
C
      STOP
      END
C
C  ----------------  HERE FOLLOW THE FUNCTION ROUTINE
C
C
      SUBROUTINE FCN (N,T,X,F)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER N
      REAL*8  T,X(2),F(2)
C
      REAL*8         C
      COMMON /PARAM/ C
C
      REAL*8        B,BM1
      COMMON /BVAR/ B,BM1
C
      INTEGER      FIRST
      COMMON /QQF/ FIRST
C
      IF (FIRST.EQ.0) GO TO 10
```



```
C
    5 CONTINUE
C
      F(1) = X(2)
      F(2) = C*X(2)*(  1.0D0-B*X(1)+BM1*X(2)  )
C
      RETURN
C
C
C
   10 CONTINUE
      FIRST = 1
C  ****************  HERE ALL PARAMETERS AND CONSTANTS OF THE
C   FUNCTION ARE DEFINED
C
      B    = 0.795505997645123D 00
      BM1  = B-1.0D0
C
C  ****************  END OF PARAMETER AND CONSTANT DEFINITION
C                    HERE WE CAN OUTPUT THE DEFINED PARAMETERS
      WRITE(6,15) B
   15 FORMAT(//' THE VALUES OF THE PARAMETERS IN THIS RUN ARE:'//
     *            '     B =',D24.16)
C
      GO TO 5
C
      END
```

Below is the output file

```
PROBLEM DESCRIPTION
 AN ITERATED MAPS OF LAUWERIER

================================================================================

     N   JFREQ  JORB  JOB   ITP   INP   INT
     2    -1     2     5     0     0     1

DIFFC :        DELTA                EPSZ                 MTH      ITNWTN
          0.1000000000D-03    0.1000000000D-07          1         6

PATHC :        HMIN                 HMAX                 EPSH                 SEN
          0.1000000000D-03    0.1000000000D+00    0.1000000000D-03    0.10000

               FACINC               FACDEC               EPSB
          0.2000000000D+01    0.5000000000D+00    0.1000000000D-05

            ITPATH    NSTEP
               3         0

QUASI :       NQUA       IQUA
                 4         48
```

```
THE VALUES OF THE PARAMETERS IN THIS RUN ARE:

    B =  0.7955059976451230D+00

NEWTON ================================================

       JFREQ       JORB        NIT        IAUG        MTH
        -1          2           0          0           1

     CONTROL PARAMETER                  PERIOD
  C  =     0.2260000000D+01       TPER =     0.1000000000D+01


    TSTOP            PROJ ON NT       X( 1)
 0.48000000D+02 -0.21131273D-01  0.49170252D+00
 0.32000000D+02 -0.14516034D-01  0.49303895D+00
 0.26000000D+02  0.26525697D-01  0.52798598D+00
 0.10000000D+02  0.33670218D-01  0.53752888D+00

     STATE VECTOR                       RESIDUAL VECTOR
X( 1) =     0.5000000000D+00   :  Q( 1) =     0.6833546548D-03
X( 2) =     0.5000000000D+00   :  Q( 2) =     0.6833546548D-03

NEWTON ================================================

       JFREQ       JORB        NIT        IAUG        MTH
        -1          2           1          0           1

     CONTROL PARAMETER                  PERIOD
  C  =     0.2260000000D+01       TPER =     0.1000000000D+01


    TSTOP            PROJ ON NT       X( 1)
 0.48000000D+02 -0.20520353D-01  0.49788255D+00
 0.32000000D+02 -0.14095334D-01  0.49928300D+00
 0.26000000D+02  0.23087805D-01  0.53012328D+00
 0.10000000D+02  0.29791691D-01  0.53934649D+00

     STATE VECTOR                       RESIDUAL VECTOR
X( 1) =     0.5065452814D+00   :  Q( 1) =     0.7421620460D-04
X( 2) =     0.5065452814D+00   :  Q( 2) =     0.7421620460D-04

NEWTON ================================================

       JFREQ       JORB        NIT        IAUG        MTH
        -1          2           2          0           1

     CONTROL PARAMETER                  PERIOD
  C  =     0.2260000000D+01       TPER =     0.1000000000D+01


    TSTOP            PROJ ON NT       X( 1)
 0.48000000D+02 -0.20393444D-01  0.49878206D+00
 0.32000000D+02 -0.14006773D-01  0.50018498D+00
 0.26000000D+02  0.22619875D-01  0.53045644D+00
 0.10000000D+02  0.29252794D-01  0.53961428D+00
```



```
     STATE VECTOR                       RESIDUAL VECTOR
X( 1) =     0.5074590331D+00   :  Q( 1) =     0.1405619300D-05
X( 2) =     0.5074590331D+00   :  Q( 2) =     0.1405619300D-05

NEWTON ================================================

       JFREQ       JORB        NIT        IAUG        MTH
        -1          2           3          0           1

     CONTROL PARAMETER                  PERIOD
  C  =     0.2260000000D+01       TPER =     0.1000000000D+01


    TSTOP            PROJ ON NT       X( 1)
 0.48000000D+02 -0.20390835D-01  0.49879991D+00
 0.32000000D+02 -0.14004951D-01  0.50020287D+00
 0.26000000D+02  0.22610662D-01  0.53046310D+00
 0.10000000D+02  0.29242156D-01  0.53961961D+00

     STATE VECTOR                       RESIDUAL VECTOR
X( 1) =     0.5074770808D+00   :  Q( 1) =    -0.1739825492D-08
X( 2) =     0.5074770808D+00   :  Q( 2) =    -0.1739825492D-08

**** PATHX ***********************************************************

    NORMAL OF TORUS
NT( 1) =  0.7071D+00
NT( 2) = -0.7071D+00

================================================================================

       JFREQ       JORB        JOB
        -1          2           4


      ISTEP     ITPATH      ITNWTN       MTH
         0          0           0         1


CONTROL PARAMETER     C    =  0.2260000000000000D+01
PERIOD                TPER =  0.1000000000000000D+01

     STATE VECTOR                                EIGENVALUES OF DERIVATIVE
X( 1) =  0.5074770807805656D+00   :  LAMDA( 1) =  0.9222D+00  0.0000D+00
X( 2) =  0.5074770807805656D+00


   COSNP       COSNT       COSTP       H           S
 0.1000D+01 -0.9907D+00  0.1000D+01  0.4000D-03  0.0000D+00


+------------------------------------------------------------------------------
!
!  MESSAGE FROM ROUTINE PATHX:
!
```

```
 MAXIMAL NUMBER OF STEPS    0 (=NSTEP)  USED
-------------------------------------------------------------------------------


===============================================================================

      JFREQ      JORB       JOB
        -1         2          4

     ISTEP     ITPATH     ITNWTN      MTH
       0          0          0         1

ONTROL PARAMETER     C    =  0.2260000000000000D+01
ERIOD                TPER =  0.1000000000000000D+01

     STATE VECTOR                                EIGENVALUES OF DERIVATIVE
( 1) =  0.5074770807805656D+00   :  LAMDA( 1) =  0.9222D+00  0.0000D+00
( 2) =  0.5074770807805656D+00

  COSNP       COSNT       COSTP       H           S
 0.1000D+01 -0.9907D+00  0.1000D+01  0.4000D-03  0.0000D+00

   NORMAL OF TORUS
T( 1) =  0.7071D+00
T( 2) = -0.7071D+00
```


Example no. 3

Purpose: Find periodic solution of autonomous system of ordinary differential equations.

We shall study periodic behaviour, so JORB=1 , of Rössler's equation:

$$dx/dt = - y - z$$

$$dy/dt = x + 0.2*y$$

$$dz/dt = 0.2 - c*z + x*z$$

We are given the intial conditions x=-0.5, y=3.0, z=2.25 for the parameter-value c=3, and the period is approximately 11.0.

First we define X(1):=x, X(2):=y, X(3):=z, C:=c . Since the system is autonomous the named COMMON area PMAPC must be included. We do not know the return plane so we set INP=0 . We will solve the ODEs with the IMSL-routine DVERK, so IODE=1 and therefore LRW=51 . We choose the tolerance TOL=10**(-8) and take EPSZ=10**(-7) . The work-space W is given 70 locations, thus IW=70 . The main program and the subroutine FCN is listed below.

```
      IMPLICIT LOGICAL*4 (A-Z)
      EXTERNAL FCN
C
      INTEGER IER,N
      REAL*8  X(3),  C,TPER,TQUA
      COMMON /PARAM/ C
C
      INTEGER INP,INT,ITP,JSYS,JORB,JOB,IPRINT,IW
      REAL*8  NP(4),NT(4),TP(4),W(70)
      COMPLEX*16 EVCRIT
C
      INTEGER        IODE,LRW
      REAL*8                 TOL,RW(51)
      COMMON /EQSOL/ IODE,LRW,TOL,RW
C
      INTEGER                    MTH,ITNWTN
```

```
      REAL*8          DELTA,EPSZ
      COMMON /DIFFC/ DELTA,EPSZ,MTH,ITNWTN
C
      INTEGER                      ITRET
      REAL*8          EPSR,DTMAXR
      COMMON /PMAPC/ EPSR,DTMAXR,ITRET
C
      JSYS  = 0
      N     = 3
C
      WRITE(6,5)
    5 FORMAT(' PROBLEM DESCRIPTION '/
     *       '  ROSSLERS EQUATION')
C
C  ----------------  PARAMETERS FOR SOLUTION OF ODE AND WORKSPACE
      IODE  = 1
      LRW   = 51
      TOL   = 1.0D-8
      IW    = 70
C  ----------------  PARAMETERS FOR HITTING RETURN SECTION
      DTMAXR=0.1D0
      ITRET =25
C  ----------------  PARAMETERS FOR DIFFERENTIATION
      EPSZ  = 1.0D-7
      MTH   = 1
      ITNWTN= 5
C
      JORB = 1
      X(1) =-0.50D0
      X(2) = 3.00D0
      X(3) = 2.25D0
      C    = 3.00D0
      TPER =11.0D0
C
      INP  = 0
C
C  ----------------  HERE WE MAY FOLLOW THE PATH
C
      JOB   = 5
      IPRINT= 3
C
      CALL PATH(FCN,N,X,TPER,TQUA,TP,ITP,NP,INP,NT,INT,EVCRIT,
     *          JSYS,JORB,JOB,IPRINT,IER,W,IW)
      IF (IER.NE.0) STOP
C
      STOP
      END
C
C  ----------------  HERE FOLLOW THE FUNCTION ROUTINE
C
C
      SUBROUTINE FCN (N,T,X,F)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER N
      REAL*8  T,X(3),F(3)
C
      REAL*8          C
      COMMON /PARAM/ C
C
```



```
      F(1) =-(X(2)+X(3))
      F(2) = X(1)+0.2D0*X(2)
      F(3) = 0.2D0+X(3)*(X(1)-C)
C
      RETURN
      END
```

Below is the output file

```
PROBLEM DESCRIPTION
 ROSSLER'S EQUATION

================================================================================

     N   JFREQ  JORB  JOB   ITP   INP   INT
     3     0     1     5     0     0     0

EQSOL :       IODE        LRW         TOL
                1          51    0.1000000000D-07

DIFFC :         DELTA                EPSZ                 MTH      ITNWTN
           0.3162277660D-03    0.1000000000D-06           1          5

PMAPC :         EPSR                 DTMAXR              ITRET
           0.1000000000D-08    0.1000000000D+00          25

PATHC :         HMIN                 HMAX                EPSH                 SEN
           0.1000000000D-02    0.1000000000D+00    0.1000000000D-02     0.10000

              FACINC               FACDEC              EPSB
           0.2000000000D+01    0.5000000000D+00    0.1000000000D-04

            ITPATH     NSTEP
              3          0

NEWTON ==================================================

        JFREQ      JORB       NIT       IAUG       MTH
          0          1          0          0          1

      CONTROL PARAMETER                PERIOD
   C  =    0.3000000000D+01      TPER =    0.1157584445D+02

      STATE VECTOR                     RESIDUAL VECTOR
X( 1) =   -0.5000000000D+00  :  Q( 1) =   -0.1188045586D+00
X( 2) =    0.3000000000D+01  :  Q( 2) =    0.5028365809D-01
X( 3) =    0.2250000000D+01  :  Q( 3) =    0.8192212360D-01

NEWTON ==================================================

        JFREQ      JORB       NIT       IAUG       MTH
```

```
        0         1         1         0         1

     CONTROL PARAMETER                PERIOD
  C  =     0.3000000000D+01      TPER =     0.1152737439D+02

     STATE VECTOR                     RESIDUAL VECTOR
X( 1) =    -0.5865330813D+00  :  Q( 1) =     0.1218299594D-01
X( 2) =     0.3085139494D+01  :  Q( 2) =     0.1236465259D-01
X( 3) =     0.2310301319D+01  :  Q( 3) =    -0.8172542099D-02


NEWTON ================================================

       JFREQ      JORB       NIT      IAUG       MTH
         0          1          2         0          1

     CONTROL PARAMETER                PERIOD
  C  =     0.3000000000D+01      TPER =     0.1153083480D+02

     STATE VECTOR                     RESIDUAL VECTOR
X( 1) =    -0.5611857070D+00  :  Q( 1) =     0.7642745998D-03
X( 2) =     0.3111255780D+01  :  Q( 2) =     0.7228550340D-03
X( 3) =     0.2293303002D+01  :  Q( 3) =    -0.5133753936D-03


NEWTON ================================================

       JFREQ      JORB       NIT      IAUG       MTH
         0          1          3         0          1

     CONTROL PARAMETER                PERIOD
  C  =     0.3000000000D+01      TPER =     0.1153107328D+02

     STATE VECTOR                     RESIDUAL VECTOR
X( 1) =    -0.5594568119D+00  :  Q( 1) =     0.2786191303D-05
X( 2) =     0.3112975660D+01  :  Q( 2) =     0.2600516172D-05
X( 3) =     0.2292142779D+01  :  Q( 3) =    -0.1871980779D-05


NEWTON ================================================

       JFREQ      JORB       NIT      IAUG       MTH
         0          1          4         0          1

     CONTROL PARAMETER                PERIOD
  C  =     0.3000000000D+01      TPER =     0.1153107416D+02

     STATE VECTOR                     RESIDUAL VECTOR
X( 1) =    -0.5594505070D+00  :  Q( 1) =    -0.2354905923D-08
X( 2) =     0.3112981895D+01  :  Q( 2) =    -0.2114118214D-08
X( 3) =     0.2292138547D+01  :  Q( 3) =     0.1591506038D-08


**** PATHX ***************************************************************


  NORMAL OF PERIODIC
WP( 1) = -0.5646D+00
WP( 2) =  0.1075D-01
WP( 3) = -0.8253D+00
```



```
================================================================================

       JFREQ      JORB       JOB
         0          1          4


       ISTEP     ITPATH     ITNWTN      MTH
         0          0          0          1

CONTROL PARAMETER     C    = 0.3000000000000000D+01
PERIOD                TPER = 0.1153107415832089D+02


     STATE VECTOR                                  EIGENVALUES OF DERIVATIVE
X( 1) = -0.5594505069791862D+00  :  LAMDA( 1) =  0.8705D-05  0.0000D+00
X( 2) =  0.3112981894827303D+01  :  LAMDA( 2) =  0.5866D+00  0.0000D+00
X( 3) =  0.2292138547377649D+01


   COSNP       COSNT       COSTP       H           S
 0.1000D+01  0.1000D+01  0.1000D+01  0.4000D-02  0.0000D+00


    TPER =  0.1153D+02      NPOINT =    45              DT =  0.2562D+00

XMIN( 1) = -0.4701D+01    XMAX( 1) =  0.5963D+01    AMP( 1) =  0.1066D+02
XMIN( 2) = -0.5599D+01    XMAX( 2) =  0.4018D+01    AMP( 2) =  0.9616D+01
XMIN( 3) =  0.2613D-01    XMAX( 3) =  0.5720D+01    AMP( 3) =  0.5694D+01


+------------------------------------------------------------------------------
!
!  MESSAGE FROM ROUTINE PATHX:
!
!  MAXIMAL NUMBER OF STEPS   0 (=NSTEP)  USED
!
+------------------------------------------------------------------------------


================================================================================

       JFREQ      JORB       JOB
         0          1          4


       ISTEP     ITPATH     ITNWTN      MTH
         0          0          0          1

CONTROL PARAMETER     C    = 0.3000000000000000D+01
PERIOD                TPER = 0.1153107415832089D+02


     STATE VECTOR                                  EIGENVALUES OF DERIVATIVE
X( 1) = -0.5594505069791862D+00  :  LAMDA( 1) =  0.8705D-05  0.0000D+00
X( 2) =  0.3112981894827303D+01  :  LAMDA( 2) =  0.5866D+00 -0.0000D+00
X( 3) =  0.2292138547377649D+01
```

```
 COSNP       COSNT       COSTP       H           S
0.1000D+01  0.1000D+01  0.1000D+01  0.4000D-02  0.0000D+00

 NORMAL OF PERIODIC
P( 1) = -0.5646D+00
P( 2) =  0.1075D-01
P( 3) = -0.8253D+00
```


Example no. 4

We shall study a special version of Duffing's equation

d2x/dt2 + 1/5 * dx/dt + x**3 = A*cos(t)

treated by Ueda. This equation is a periodically forced system with forcing period 2*PI. When A=0 then x(t)=0 is a solution. We will follow this solution in dependence of the parameter A. Starting in this point we want to take 3 steps along the path of solutions. We do not know the tangent of the path so ITP=0. But we want to follow the path for increasing parametervalues. Therefore we let TP be a unitvector in the parameter direction. A program to perform this task is listed below.

```
      IMPLICIT LOGICAL*4 (A-Z)
      EXTERNAL FCN
C
      INTEGER IER,N
      REAL*8  X(2),  C,TPER,TQUA
      COMMON /PARAM/ C
C
      INTEGER INP,INT,ITP,JSYS,JORB,JOB,IPRINT,IW
      REAL*8  NP(3),NT(3),TP(3),W(74)
      COMPLEX*16 EVCRIT
C
      INTEGER         IODE,LRW
      REAL*8                  TOL,RW(51)
      COMMON /EQSOL/ IODE,LRW,TOL,RW
C
      INTEGER                    MTH,ITNWTN
      REAL*8          DELTA,EPSZ
      COMMON /DIFFC/ DELTA,EPSZ,MTH,ITNWTN
C
      INTEGER         ITPATH,NSTEP
      REAL*8          HMIN,HMAX,EPSH,SEND,FACINC,FACDEC,EPSB
      COMMON /PATHC/ HMIN,HMAX,EPSH,SEND,FACINC,FACDEC,EPSB,
     *                ITPATH,NSTEP
C
      JSYS  = 1
      N     = 2
C
      WRITE(6,5)
    5 FORMAT(' PROBLEM DESCRIPTION '/
     *       ' DUFFINGS EQUATION')
C
C  ----------------  PARAMETERS FOR SOLUTION OF ODE AND WORKSPACE
```

```
      IODE   = 1
      LRW    = 51
      TOL    = 1.0D-08
      IW     = 74
C  ---------------  PARAMETERS FOR DIFFERENTIATION
      EPSZ   = 1.0D-07
      MTH    = 1
      ITNWTN= 6
C
      JORB = 1
      X(1) = 0.00D0
      X(2) = 0.00D0
      C    = 0.00D0
      TPER = 6.283185307179586D+00
C
      ITP  = 0
      TP(1)= 0.0D0
      TP(2)= 0.0D0
      TP(3)= 1.0D0
C ----------------  PARAMETERS FOR PATH FOLLOWING
      ITPATH= 4
      NSTEP = 3
      HMIN   = 0.01D0
      HMAX   = 0.2D0
      EPSH   = 1.0D-3
      FACINC= 3.0D0
      FACDEC= 0.3D0
      EPSB   = 1.0D-6
C
C  ----------------  HERE WE MAY FOLLOW THE PATH
C
      JOB    = 4
      IPRINT= 2
C
      CALL PATH(FCN,N,X,TPER,TQUA,TP,ITP,NP,INP,NT,INT,EVCRIT,
     *          JSYS,JORB,JOB,IPRINT,IER,W,IW)
      IF (IER.NE.0) STOP
C
      STOP
      END
C
C  ----------------  HERE FOLLOW THE FUNCTION ROUTINE
C
C
      SUBROUTINE FCN (N,T,X,F)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER N
      REAL*8  T,X(2),F(2)
C
      REAL*8         C
      COMMON /PARAM/ C
C
      F(1) = X(2)
      F(2) =-0.2D0*X(2)-X(1)*X(1)*X(1)+C*DCOS(T)
C
      RETURN
      END
```


Below is the output file

```
PROBLEM DESCRIPTION
 ROSSLER'S EQUATION

================================================================================

     N    JFREQ   JORB   JOB    ITP    INP    INT
     3      0      1      5      0      0      0

EQSOL :        IODE        LRW         TOL
                 1          51     0.1000000000D-07

DIFFC :         DELTA                EPSZ                 MTH      ITNWTN
           0.3162277660D-03     0.1000000000D-06          1          5

PMAPC :          EPSR                DTMAXR              ITRET
           0.1000000000D-08     0.1000000000D+00          25

PATHC :          HMIN                HMAX                EPSH                    S
           0.1000000000D-02     0.1000000000D+00     0.1000000000D-02     0.100

              FACINC              FACDEC              EPSB
           0.2000000000D+01     0.5000000000D+00     0.1000000000D-04

            ITPATH     NSTEP
              3          0

NEWTON ==========================================================

        JFREQ       JORB        NIT        IAUG        MTH
          0          1           0          0          1

      CONTROL PARAMETER                PERIOD
   C  =     0.3000000000D+01      TPER =     0.1157584445D+02

      STATE VECTOR                     RESIDUAL VECTOR
X( 1) =    -0.5000000000D+00   :  Q( 1) =    -0.1188045586D+00
X( 2) =     0.3000000000D+01   :  Q( 2) =     0.5028355809D-01
X( 3) =     0.2250000000D+01   :  Q( 3) =     0.8192212360D-01

NEWTON ==========================================================

        JFREQ       JORB        NIT        IAUG        MTH
          0          1           1          0          1

      CONTROL PARAMETER                PERIOD
   C  =     0.3000000000D+01      TPER =     0.1152737439D+02

      STATE VECTOR                     RESIDUAL VECTOR
X( 1) =    -0.5865330813D+00   :  Q( 1) =     0.1218299594D-01
X( 2) =     0.3065139494D+01   :  Q( 2) =     0.1236465259D-01
```

```
( 3) =    0.2310301319D+01  :  Q( 3) =   -0.8172542099D-02

EWTON ================================================

     JFREQ      JORB       NIT       IAUG       MTH
       0         1          2          0         1

    CONTROL PARAMETER                  PERIOD
  C  =    0.3000000000D+01       TPER =    0.1153083480D+02

    STATE VECTOR                       RESIDUAL VECTOR
( 1) =  -0.5611857070D+00  :  Q( 1) =    0.7642745998D-03
( 2) =    0.3111255780D+01  :  Q( 2) =    0.7228550340D-03
( 3) =    0.2293303002D+01  :  Q( 3) =   -0.5133753936D-03

EWTON ================================================

     JFREQ      JORB       NIT       IAUG       MTH
       0         1          3          0         1

    CONTROL PARAMETER                  PERIOD
  C  =    0.3000000000D+01       TPER =    0.1153107328D+02

    STATE VECTOR                       RESIDUAL VECTOR
( 1) =   -0.5594568119D+00  :  Q( 1) =    0.2786191303D-05
( 2) =    0.3112975660D+01  :  Q( 2) =    0.2600516172D-05
( 3) =    0.2292142779D+01  :  Q( 3) =   -0.1871980779D-05

EWTON ================================================

     JFREQ      JORB       NIT       IAUG       MTH
       0         1          4          0         1

    CONTROL PARAMETER                  PERIOD
  C  =    0.3000000000D+01       TPER =    0.1153107416D+02

    STATE VECTOR                       RESIDUAL VECTOR
( 1) =   -0.5594505070D+00  :  Q( 1) =   -0.2354905923D-08
( 2) =    0.3112981895D+01  :  Q( 2) =   -0.2114118214D-08
( 3) =    0.2292138547D+01  :  Q( 3) =    0.1591506038D-08

*** PATHX ***************************************************************

 NORMAL OF PERIODIC
P( 1) = -0.5646D+00
P( 2) =  0.1075D-01
P( 3) = -0.8253D+00

==========================================================================

     JFREQ      JORB       JOB
       0         1          4
```



```
     ISTEP     ITPATH     ITNWTN      MTH
       0          0          0         1

CONTROL PARAMETER     C    =  0.3000000000000000D+01
PERIOD                TPER =  0.1153107415832089D+02

    STATE VECTOR                              EIGENVALUES OF DERIVATIVE
X( 1) = -0.5594505069791862D+00  :  LAMDA( 1) =  0.8705D-05  0.0000D+00
X( 2) =  0.3112981894827303D+01  :  LAMDA( 2) =  0.5866D+00  0.0000D+00
X( 3) =  0.2292138547377649D+01

   COSNP       COSNT       COSTP       H           S
 0.1000D+01  0.1000D+01  0.1000D+01  0.4000D-02  0.0000D+00

    TPER =  0.1153D+02      NPOINT =    45              DT =  0.2562D+00

XMIN( 1) = -0.4701D+01    XMAX( 1) =  0.5963D+01    AMP( 1) =  0.1066D+02
XMIN( 2) = -0.5599D+01    XMAX( 2) =  0.4018D+01    AMP( 2) =  0.9616D+01
XMIN( 3) =  0.2613D-01    XMAX( 3) =  0.5720D+01    AMP( 3) =  0.5694D+01

+------------------------------------------------------------------------
!
!  MESSAGE FROM ROUTINE PATHX:
!
!  MAXIMAL NUMBER OF STEPS   0 (=NSTEP)  USED
!
+------------------------------------------------------------------------

==========================================================================

     JFREQ      JORB       JOB
       0         1          4

     ISTEP     ITPATH     ITNWTN      MTH
       0          0          0         1

CONTROL PARAMETER     C    =  0.3000000000000000D+01
PERIOD                TPER =  0.1153107415832089D+02

    STATE VECTOR                              EIGENVALUES OF DERIVATIVE
X( 1) = -0.5594505069791862D+00  :  LAMDA( 1) =  0.8705D-05  0.0000D+00
X( 2) =  0.3112981894827303D+01  :  LAMDA( 2) =  0.5866D+00  0.0000D+00
X( 3) =  0.2292138547377649D+01

   COSNP       COSNT       COSTP       H           S
 0.1000D+01  0.1000D+01  0.1000D+01  0.4000D-02  0.0000D+00

  NORMAL OF PERIODIC
NP( 1) = -0.5645D+00
```

```
( 2) =  0.1075D-01
( 3) = -0.8253D+00
```


Example no. 5

In this example we shall study chaotic behaviour in the Lorenz equations. This system of three autonomous non-linear ordinary differential equations of first order is:

$$dx/dt = -\sigma x + \sigma y$$

$$dy/dt = (\gamma - z)x - y$$

$$dz/dt = xy - bz$$

The system has for b=4, sigma=16, and gamma=40 a chaotic attractor, and x=10, y=0, z=30 is an initial point close to that attractor. We want the three largest Liapunov exponents (i.e. all of them) so M=3. We take the renormalization time to 0.5 with stop at every 2 renormalizations. We choose to stop 8 times. For simplicity we take three standard unit vectors to span the volume element. The program is listed below.

```
      IMPLICIT LOGICAL*4 (A-Z)
      EXTERNAL FCN
C
      INTEGER ICH,IER,IW,JSYS,K,M,N,NSTEP
      REAL*8  E(3,3),LIAX(3),T,TAU,X(3),W(50)
C
      INTEGER        IODE,LRW
      REAL*8                  TOL,RW(51)
      COMMON /EQSOL/ IODE,LRW,TOL,RW
C
      INTEGER      FIRST
      COMMON /QQF/ FIRST
C
      FIRST = 0
      JSYS  = 0
      N     = 3
C
      WRITE(6,5)
    5 FORMAT(' PROBLEM DESCRIPTION '/
     *       ' CHAOS IN LORENZ EQUATIONS')
C
C  ----------------  PARAMETERS FOR SOLUTION OF ODE AND WORKSPACE
      IODE  = 1
      LRW   = 51
      TOL   = 1.0D-08
      IW    = 50
```

```
C
      T    =  0.0D0
      X(1) = 10.0D0
      X(2) =  0.0D0
      X(3) = 30.0D0
C
      TAU    = 0.5D0
      NSTEP  = 2
      M      = 3
      E(1,1) = 1.0D0
      E(2,1) = 0.0D0
      E(3,1) = 0.0D0
      E(1,2) = 0.0D0
      E(2,2) = 1.0D0
      E(3,2) = 0.0D0
      E(1,3) = 0.0D0
      E(2,3) = 0.0D0
      E(3,3) = 1.0D0
C
      ICH  = 0
C
      DO 10  K=1,8
         CALL CHAOS(JSYS,FCN,N,T,X,TAU,NSTEP,
     *              E,M,ICH,LIAX,IER,W,IW)
         IF (IER.NE.0) STOP
C
         WRITE(6,15) ICH,LIAX(1),LIAX(2),LIAX(3)
   15    FORMAT(' ICH ',I5,'    EXP NO.1 ',D12.4,
     *          '    EXP NO.2 ',D12.4,'    EXP NO.3 ',D12.4)
C
   10 CONTINUE
C
      STOP
      END
C
C  ---------------  HERE FOLLOW THE FUNCTION ROUTINE
C
C
      SUBROUTINE FCN (N,T,X,F)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER N
      REAL*8  T,X(3),F(3)
C
      REAL*8         B,GAMMA,SIGMA
      COMMON /BVAR/ B,GAMMA,SIGMA
C
      INTEGER      FIRST
      COMMON /QQF/ FIRST
C
      IF (FIRST.EQ.0) GO TO 10
C
    5 CONTINUE
C
      F(1) =     -SIGMA*X(1)  +SIGMA*X(2)
      F(2) = (GAMMA-X(3))*X(1)-   X(2)
      F(3) =   X(1)*X(2)                -B*X(3)
C
      RETURN
C
```



```
C
C
   10 CONTINUE
         FIRST = 1
C  ****************  HERE ALL PARAMETERS AND CONSTANTS OF THE
C   FUNCTION ARE DEFINED
C
         B     =  4.0D0
         SIGMA= 16.0D0
         GAMMA= 40.0D0
C
C  ****************  END OF PARAMETER AND CONSTANT DEFINITION
         GO TO 5
C
      END
```

Below is the output file

```
PROBLEM DESCRIPTION
CHAOS IN LORENZ EQUATIONS
ICH      2    EXP NO.1    0.1711D+01    EXP NO.2  -0.3631D+01    EXP NO.3  -0.19
ICH      4    EXP NO.1    0.1475D+01    EXP NO.2  -0.1695D+01    EXP NO.3  -0.21
ICH      6    EXP NO.1    0.1735D+01    EXP NO.2  -0.1223D+01    EXP NO.3  -0.21
ICH      8    EXP NO.1    0.1279D+01    EXP NO.2  -0.7358D+00    EXP NO.3  -0.21
ICH     10    EXP NO.1    0.1405D+01    EXP NO.2  -0.6305D+00    EXP NO.3  -0.20
ICH     12    EXP NO.1    0.1417D+01    EXP NO.2  -0.5604D+00    EXP NO.3  -0.21
ICH     14    EXP NO.1    0.1360D+01    EXP NO.2  -0.4002D+00    EXP NO.3  -0.21
ICH     16    EXP NO.1    0.1374D+01    EXP NO.2  -0.3362D+00    EXP NO.3  -0.21
```

Example no. 6

Purpose: Computation of a partial derivative of a non-linear map.

We shall use the routine DOFQ.

The two function are

$$(2*(x)**2 - (x*c)**2 - (y)**2 ,$$

$$x*y*c - (y)**3 + c - 1) .$$

In the point (x,y;c) = (1,1;1) we want the partial derivative of the second coordinate function in the c-direction.

We define X(1):=x, X(2):=y, C:=c. Then N=2 and JSYS=-1, JORB=0. We shall make increments in the C-direction and project in the second coordinate direction. The program calling DOFQ is listed below.

```
      IMPLICIT LOGICAL*4 (A-Z)
      EXTERNAL FCN
C
      INTEGER IER,N
      REAL*8  X(2),  C,TPER,TQUA
      COMMON /PARAM/ C
C
      INTEGER INP,INT,ITP,JSYS,JORB,JOB,IPRINT,IW
      REAL*8  NP(3),NT(3),TP(3),W(70)
      REAL*8  PROJ(2),BAS(3)
C
      INTEGER        IODE,LRW
      REAL*8                  TOL,RW(2)
      COMMON /EQSOL/ IODE,LRW,TOL,RW
C
      INTEGER                        NEPSA
      REAL*8         EPSA,HA,HSTART
      COMMON /DERIV/ EPSA,HA,HSTART,NEPSA
C
      JSYS  =-1
      N     = 2
C
      WRITE(6,5)
    5 FORMAT(' PROBLEM DESCRIPTION '/
     *       '  DERIVATIVE OF A NON-LINEAR MAP')
C
C  ----------------  PARAMETERS FOR SOLUTION OF ODE AND WORKSPACE
      LRW   = 2
      IW    = 70
C
      JORB = 0
      X(1) = 1.0D0
      X(2) = 1.0D0
      C    = 1.0D0
C
      HA   = 1.0D-12
      NEPSA = 20
C
      BAS(1)  = 0.0D0
      BAS(2)  = 0.0D0
      BAS(3)  = 1.0D0
      PROJ(1) = 0.0D0
      PROJ(2) = 1.0D0
C
      IPRINT = 2
C
      CALL DOFQ(JSYS,JORB,FCN,N,X,TPER,TQUA,NP,INP,NT,INT,BAS,PROJ,
     *          IPRINT,IER,W,IW)
      IF (IER.NE.0) STOP
C
      STOP
      END
C
C  ----------------  HERE FOLLOW THE FUNCTION ROUTINE
C
C
      SUBROUTINE FCN (N,T,X,F)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER N
      REAL*8  T,X(2),F(2)
C
      REAL*8         C
      COMMON /PARAM/ C
C
      F(1) = 2.0D0*X(1)*X(1)-X(1)*X(1)*C*C-X(2)*X(2)
      F(2) = X(1)*X(2)*C-X(2)*X(2)*X(2)+C-1.0D0
C
      RETURN
      END
```

Below is the output file

```
PROBLEM DESCRIPTION
 DERIVATIVE OF A NON-LINEAR MAP

================================================================================

     N   JSYS  JORB    JOB    ITP    INP    INT
     2    -1     0      1      0      0      0

DIFFC :        DELTA                 EPSZ                  MTH      ITNWTN
          0.1000000000D-06     0.1000000000D-13           1         3

PATHC :        HMIN                  HMAX                  EPSH                  SEN
```

```
        0.1000000000D-09    0.1000000000D-07    0.3162277660D-03    0.10000

           FACINC              FACDEC              EPSB
        0.2000000000D+01    0.5000000000D+00    0.1000000000D-11

          ITPATH    NSTEP
            3         0


ERIV :       EPSA                HA                 HSTART              NEPSA
        0.0000000000D+00    0.1000000000D-11    0.1000000000D+09         20


      DIRECTION OF INCREMENT              DIRECTION OF PROJECTION
AS( 1) =    0.0000000000D+00   :  PROJ( 1) =    0.0000000000D+00
AS( 2) =    0.0000000000D+00   :  PROJ( 2) =    0.1000000000D+01
AS( 3) =    0.1000000000D+01

================================================================================

      JSYS       JORB       JOB
       -1          0         1


     ISTEP     ITPATH     ITNWTN      MTH
       0          0          0         0


ONTROL PARAMETER    C    =  0.1000000000000000D+01


     STATE VECTOR
( 1) =  0.1000000000000000D+01
( 2) =  0.1000000000000000D+01


  COSNP       COSNT       COSTP       H           S
 0.1000D+01  0.1000D+01  0.1000D+01  0.0000D+00  0.0000D+00


CCOFF ================================

           EPSA   =    0.1213465448D-15


DF    ======================================================

       =    0.0000000000D+00  :  H       =    0.5318870155D-06
HIF    =    0.1999999999D+01  :  EBPHIF   =    0.2281434615D-09
HIB    =    0.2000000000D+01  :  EBPHIB   =    0.2281434614D-09
HIC    =    0.2000000000D+01
HI2    =   -0.1079205492D-02  :  EBPHI2   =    0.1569807284D+01


DF    ======================================================

       =    0.0000000000D+00  :  H       =    0.5318870155D-05
HIF    =    0.2000000000D+01  :  EBPHIF   =    0.2281434614D-10
HIB    =    0.2000000000D+01  :  EBPHIB   =    0.2281434614D-10
```



```
PHIC   =    0.2000000000D+01
PHI2   =   -0.2943287706D-05  :  EBPHI2   =    0.5829293374D+01


CDF   ======================================================

F      =    0.0000000000D+00  :  H       =    0.5318870155D-04
PHIF   =    0.2000000000D+01  :  EBPHIF   =    0.2281434614D-11
PHIB   =    0.2000000000D+01  :  EBPHIB   =    0.2281434614D-11
PHIC   =    0.2000000000D+01
PHI2   =   -0.4905479510D-07  :  EBPHI2   =    0.3497576025D+01


CDF   ======================================================

F      =    0.0000000000D+00  :  H       =    0.5318870155D-03
PHIF   =    0.2000000000D+01  :  EBPHIF   =    0.2281434614D-12
PHIB   =    0.2000000000D+01  :  EBPHIB   =    0.2281434614D-12
PHIC   =    0.2000000000D+01
PHI2   =   -0.3924383608D-09  :  EBPHI2   =    0.4371970031D+01


CDF   ======================================================

F      =    0.0000000000D+00  :  H       =    0.5318870155D-02
PHIF   =    0.2000000000D+01  :  EBPHIF   =    0.2281434614D-13
PHIB   =    0.2000000000D+01  :  EBPHIB   =    0.2281434614D-13
PHIC   =    0.2000000000D+01
PHI2   =   -0.8829863119D-11  :  EBPHI2   =    0.1943097791D+01


CDF   ======================================================

F      =    0.0000000000D+00  :  H       =    0.5318870155D-01
PHIF   =    0.2000000000D+01  :  EBPHIF   =    0.2281434614D-14
PHIB   =    0.2000000000D+01  :  EBPHIB   =    0.2281434614D-14
PHIC   =    0.2000000000D+01
PHI2   =   -0.2943287706D-13  :  EBPHI2   =    0.5829293374D+01


CDF   ======================================================

F      =    0.0000000000D+00  :  H       =    0.5318870155D+00
PHIF   =    0.2000000000D+01  :  EBPHIF   =    0.2281434614D-15
PHIB   =    0.2000000000D+01  :  EBPHIB   =    0.2281434614D-15
PHIC   =    0.2000000000D+01
PHI2   =    0.0000000000D+00  :  EBPHI2   =    0.7230000000D+01


OPTFDI ========================================

F APPEARS TO BE ODD OR NEARLY LINEAR

          IT     =                    7
          EPSA   =    0.1213465448D-15
          HS     =    0.5318870155D-06      HOPT   =    0.5318870155D-06
          F      =    0.0000000000D+00
          D1     =    0.1999999999D+01      ERRD   =    0.4562869228D-09
          D2     =    0.0000000000D+00
```

```
FQX   **** ERROR RETURN FROM ROUTINE CALL
FQ    **** ERROR RETURN FROM ROUTINE CALL
```


Example no. 7

We want to plot Y(t) of the periodic solution of Rösler's equation found in Ex. no. 3. We found the initial values to be

X = -0.55945 05069 791862

Y = 3.11298 18948 27303

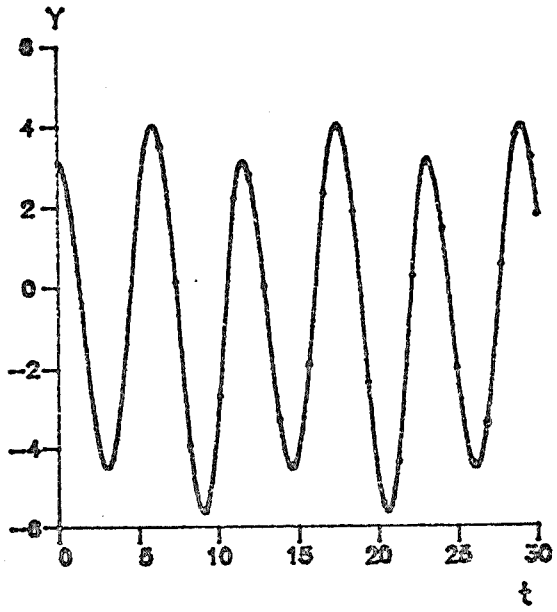
Z = 2.29213 85473 77649

and the period TPER = 11.53107 41583 2089. Since we only want the Y component we only have to output T and X(2). We send it to the file no. 79 . We integrate from T=0 to T=30 with a total of 1000 points (so the time-spacing between the points is 30/1000 = 0.03). The program is listed below.

```
      IMPLICIT LOGICAL*4 (A-Z)
      EXTERNAL FCN,OXOFT
C
      INTEGER IER,N
      REAL*8  X(3),  C
      COMMON /PARAM/ C
C
      INTEGER        IODE,LRW
      REAL*8                  TOL,RW(51)
      COMMON /EQSOL/ IODE,LRW,TOL,RW
C
      INTEGER IDEV,NPOINT
      REAL*8  TEND
C
      N     = 3
C
      WRITE(6,5)
    5 FORMAT(' PROBLEM DESCRIPTION '/
     *       '  ROSSLERS EQUATION')
C
C  ----------------  PARAMETERS FOR SOLUTION OF ODE AND WORKSPACE
      IODE  = 1
      LRW   = 51
      TOL   = 1.0D-8
C
      X(1) =-0.5594505069791862D+00
      X(2) = 3.112981894827303D+00
      X(3) = 2.292138547377649D+00
      C    = 3.00D0
C
      TEND  = 30.0D0
      NPOINT=1000
```

```
      IDEV  = 79
C
      CALL GRXOFT(FCN,N,X,TEND,NPOINT,OXOFT,IDEV,IER)
      IF (IER.NE.0) STOP
C
      STOP
      END
C
C
C
      SUBROUTINE OXOFT (T,X,N,IDEV)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER  N,IDEV,II
      REAL*8   T,X(1)
C
      WRITE(IDEV,10) T,X(2)
   10 FORMAT(2D16.6)
C
      RETURN
      END
C
C  ----------------  HERE FOLLOW THE FUNCTION ROUTINE
C
C
      SUBROUTINE FCN (N,T,X,F)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER N
      REAL*8  T,X(3),F(3)
C
      REAL*8         C
      COMMON /PARAM/ C
C
      F(1) =-(X(2)+X(3))
      F(2) = X(1)+0.2D0*X(2)
      F(3) = 0.2D0+X(3)*(X(1)-C)
C
      RETURN
      END
```

A plot of the solution can be seen on the next page.



Plot of the Y-component of the periodic solution.

Example no. 8

We shall consider a bi-periodically forced system

$$dx/dt = y$$

$$dy/dt = -k*y-x-x**3+c*\cos(t)+1.5*\cos(0.1147*t)$$

So N=2 and JSYS=2, where the two periods are TPER=2*PI and TQUA=2*PI/0.1147 with TPER < TQUA. Since the two periods are known we first call YWIND with WIND = TPER/TQUA . Based on the results obtained we choose the stop-times 17*TPER, 26*TPER, 35*TPER, 44*TPER, and 53*TPER, so that NQUA=5. These stop-times are stored in the 5 first locations of the work-space W before call to PATH. The stop-times are independent of the constants in the equations. We take c=0.3 and k=0.05. The program listing of the main program is given below.

```
      IMPLICIT LOGICAL*4 (A-Z)
      EXTERNAL FCN
C
      INTEGER IER,N
      REAL*8  X(2),  C,TPER,TQUA
      COMMON /PARAM/ C
C
      INTEGER INP,INT,ITP,JSYS,JORB,JOB,IPRINT,IW
      REAL*8  NP(3),NT(3),TP(3),W(300)
      COMPLEX*16 EVCRIT
C
      INTEGER        IODE,LRW
      REAL*8                  TOL,RWORK(51)
      COMMON /EQSOL/ IODE,LRW,TOL,RWORK
C
      INTEGER                    MTH,ITNWTN
      REAL*8         DELTA,EPSZ
      COMMON /DIFFC/ DELTA,EPSZ,MTH,ITNWTN
C
      INTEGER        NQUA
      COMMON /QUASI/ NQUA
C
      INTEGER IQUA
      REAL*8  DMAX,ROTNR
C
      WRITE(6,5)
    5 FORMAT(' PROBLEM DESCRIPTION '/
     *       ' BI-PERIODICALLY FORCED SYSTEM')
```



```
C
      JSYS  = 2
      N     = 2
C  ---------------  PARAMETERS FOR SOLUTION OF ODE AND WORKSPACE
      IODE  = 1
      LRW   = 51
      TOL   = 1.0D-08
      IW    = 300
C  ---------------  PARAMETERS FOR DIFFERENTIATION
      MTH   = 0
      ITNWTN= 2
C     DELTA =
      EPSZ  = 1.0D-4
C
      JORB = 2
      X(1) = 1.1D0
      X(2) = 0.3D0
      C    = 0.3D0
      TPER = 6.283185307179586D 00
      TQUA = TPER/0.1147D0
C
C
C     IQUA  = 100
C     DMAX  = 0.1D0
C     ROTNR =TPER/TQUA
C     CALL YWIND(TPER,ROTNR,IQUA,DMAX)
C     IF (2.GT.1) STOP
C  ---------------  PARAMETER  FOR INTERPOLATION OF STROBED ORBIT
      NQUA  = 5
      W(1)  = 17.0D0*TPER
      W(2)  = 26.0D0*TPER
      W(3)  = 35.0D0*TPER
      W(4)  = 44.0D0*TPER
      W(5)  = 53.0D0*TPER
C
C  ---------------  HERE WE MAY FOLLOW THE PATH
C
      JOB   = 5
      IPRINT= 3
C
      CALL PATH(FCN,N,X,TPER,TQUA,TP,ITP,NP,INP,NT,INT,EVCRIT,
     *          JSYS,JORB,JOB,IPRINT,IER,W,IW)
      IF (IER.NE.0) STOP
C
      STOP
      END
C
C  ---------------  HERE FOLLOW THE FUNCTION ROUTINE
C
C
      SUBROUTINE FCN (N,T,X,F)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER N
      REAL*8 T,X(2),F(2)
C
      REAL*8         C
      COMMON /PARAM/ C
C
      F(1)=X(2)
```

```
      F(2)=-0.05D0*X(2)-X(1)-X(1)**3
     *     +C*DCOS(T)+1.5D0*DCOS(0.1147D0*T)
C
      RETURN
      END
```

Below is the output file

```
ROBLEM DESCRIPTION
-PERIODICALLY FORCED SYSTEM

================================================================================

    N   JFREQ   JORB   JOB    ITP    INP    INT
    2     2      2     5      0      0      0

SOL :       IODE       LRW        TOL
              1         51    0.1000000000D-07

FFC :        DELTA                 EPSZ                  MTH      ITNWTN
         0.1000000000D-01     0.1000000000D-03          0          2

THC :         HMIN                 HMAX                 EPSH                  S
         0.1000000000D+01     0.1000000000D+02     0.1000000000D+01     0.100

            FACINC               FACDEC               EPSB
         0.2000000000D+01     0.5000000000D+00     0.1000000000D-01

           ITPATH     NSTEP
             3          0

ASI :       NQUA
              5

EWTON ==================================================

       JFREQ      JORB       NIT       IAUG       MTH
         2          2          0         0          0

     CONTROL PARAMETER                PERIOD
  C  =     0.3000000000D+00       TPER =     0.6283185307D+01

   TSTOP            PROJ ON NT       X( 1)
 0.10681415D+03 -0.50100000D-01   0.65484087D+00
 0.16336282D+03 -0.17800000D-01   0.10442454D+01
 0.21991149D+03  0.14500000D-01   0.12626995D+01
 0.27646015D+03  0.46800000D-01   0.10087104D+01
 0.33300882D+03  0.79100000D-01   0.71278743D+00

     STATE VECTOR                      RESIDUAL VECTOR
( 1) =     0.1100000000D+01   :   Q( 1) =     0.1167628805D+00
```



```
X( 2) =     0.3000000000D+00   :   Q( 2) =    -0.6006411199D-01

NEWTON ==================================================

       JFREQ      JORB       NIT       IAUG       MTH
         2          2          1         0          0

     CONTROL PARAMETER                PERIOD
  C  =     0.3000000000D+00       TPER =     0.6283185307D+01

   TSTOP            PROJ ON NT       X( 1)
 0.10681415D+03 -0.50100000D-01   0.66118079D+00
 0.16336282D+03 -0.17800000D-01   0.10423013D+01
 0.21991149D+03  0.14500000D-01   0.12622795D+01
 0.27646015D+03  0.46800000D-01   0.10087361D+01
 0.33300882D+03  0.79100000D-01   0.71281950D+00

     STATE VECTOR                      RESIDUAL VECTOR
X( 1) =     0.1216762880D+01   :   Q( 1) =    -0.1436963723D-02
X( 2) =     0.2399358880D+00   :   Q( 2) =    -0.2068209980D-03

NEWTON ==================================================

       JFREQ      JORB       NIT       IAUG       MTH
         2          2          2         0          0

     CONTROL PARAMETER                PERIOD
  C  =     0.3000000000D+00       TPER =     0.6283185307D+01

   TSTOP            PROJ ON NT       X( 1)
 0.10681415D+03 -0.50100000D-01   0.66111004D+00
 0.16336282D+03 -0.17800000D-01   0.10423353D+01
 0.21991149D+03  0.14500000D-01   0.12622852D+01
 0.27646015D+03  0.46800000D-01   0.10087351D+01
 0.33300882D+03  0.79100000D-01   0.71281906D+00

     STATE VECTOR                      RESIDUAL VECTOR
X( 1) =     0.1215325917D+01   :   Q( 1) =     0.2230426842D-04
X( 2) =     0.2397290670D+00   :   Q( 2) =     0.1029336390D-04

**** PATHX ***************************************************************

================================================================================

       JFREQ      JORB       JOB
         2          2          4

      ISTEP     ITPATH     ITNWTN       MTH
         0          0          0          0

CONTROL PARAMETER    C    =  0.3000000000000000D+00
PERIOD               TPER =  0.6283185307179586D+01
```

```
SECOND PERIOD        TPER2=  0.5477929648805219D+02

     STATE VECTOR                                EIGENVALUES OF DERIVATIVE
X( 1) =  0.1215325916743297D+01   :  LAMDA( 1) = -0.5378D-02  0.0000D+00
X( 2) =  0.2397290670089286D+00   :  LAMDA( 2) = -0.1341D-01  0.0000D+00

   COSNP        COSNT        COSTP        H            S
 0.1000D+01   0.1000D+01   0.1000D+01   0.4000D+01   0.0000D+00

+---------------------------------------------------------------------------
!
!  MESSAGE FROM ROUTINE PATHX:
!
!  MAXIMAL NUMBER OF STEPS    0 (=NSTEP)  USED
!
+---------------------------------------------------------------------------

===========================================================================

       JFREQ      JORB       JOB
         2          2          4

      ISTEP     ITPATH     ITNWTN      MTH
         0          0          0          0

CONTROL PARAMETER    C    =  0.3000000000000000D+00
PERIOD               TPER =  0.6283185307179586D+01
SECOND PERIOD        TPER2=  0.5477929648805219D+02

     STATE VECTOR                                EIGENVALUES OF DERIVATIVE
X( 1) =  0.1215325916743297D+01   :  LAMDA( 1) = -0.5378D-02  0.0000D+00
X( 2) =  0.2397290670089286D+00   :  LAMDA( 2) = -0.1341D-01  0.0000D+00

   COSNP        COSNT        COSTP        H            S
 0.1000D+01   0.1000D+01   0.1000D+01   0.4000D+01   0.0000D+00
```


Example no. 9

Purpose: We shall consider the Hopf bifurcation in the Lorenz system treated in example no. 5. The values of SIGMA and B are fixed to SIGMA=16 and B=4. The stationary solution

X = Y = square root (B*(GAMMA-1)) , Z = GAMMA-1

meets a Hopf bifurcation point for

GAMMA = SIGMA*(SIGMA+B+3)/(SIGMA-B-1)

and the critical eigenvalue is

0 +i * square root (B*SIGMA*(2*SIGMA+2)/(SIGMA-B-1))

We only want to perform a bifurcation analysis so JOB=1. The program is listed below.

```
      IMPLICIT LOGICAL*4 (A-Z)
      EXTERNAL FCN
C
      INTEGER IER,N
      REAL*8  X(3),  GAMMA,TPER,TQUA
      COMMON /PARAM/ GAMMA
C
      INTEGER INP,INT,ITP,JSYS,JORB,JOB,IPRINT,IW
      REAL*8  NP(4),NT(4),TP(4),W(300)
      COMPLEX*16 EVCRIT
C
      INTEGER         IODE,LRW
      REAL*8                    TOL,RW(4)
      COMMON /EQSOL/ IODE,LRW,TOL,RW
C
      INTEGER                      MTH,ITNWTN
      REAL*8          DELTA,EPSZ
      COMMON /DIFFC/ DELTA,EPSZ,MTH,ITNWTN
C
      INTEGER      FIRST
      COMMON /QQF/ FIRST
C
      FIRST =0
C
      WRITE(6,5)
    5 FORMAT(' PROBLEM DESCRIPTION '/
     *       ' HOPF BIFURCATION OF THE LORENZ EQUATIONS')
C
      JSYS  = 0
      N     = 3
C  ----------------  PARAMETERS FOR SOLUTION OF ODE AND WORKSPACE
      LRW   = 4
```

```
      IW    = 300
C  ---------------  PARAMETERS FOR DIFFERENTIATION
      EPSZ  = 1.0D-14
C
      JORB = 0
      GAMMA= 16.0D0*23.0D0/11.0D0
      X(1) = DSQRT(4.0D0*(GAMMA-1.0D0))
      X(2) = X(1)
      X(3) = GAMMA-1.0D0
C
C  ---------------  HERE WE MAY FOLLOW THE PATH
C
      EVCRIT=( 0.0D0, 1.0D0)*8.0D0*DSQRT(34.0D0/11.0D0)
C
      JOB   = 1
      IPRINT= 2
C
      CALL PATH(FCN,N,X,TPER,TQUA,TP,ITP,NP,INP,NT,INT,EVCRIT,
     *          JSYS,JORB,JOB,IPRINT,IER,W,IW)
      IF (IER.NE.0) STOP
C
      STOP
      END
C
C
C  ---------------  HERE FOLLOW THE FUNCTION ROUTINE
C
C
      SUBROUTINE FCN (N,T,X,F)
C
      IMPLICIT LOGICAL*4 (A-Z)
      INTEGER N
      REAL*8  T,X(3),F(3)
C
      REAL*8         GAMMA
      COMMON /PARAM/ GAMMA
C
      REAL*8        B,SIGMA
      COMMON /BVAR/ B,SIGMA
C
      INTEGER      FIRST
      COMMON /QQF/ FIRST
C
      IF (FIRST.EQ.0) GO TO 10
C
    5 CONTINUE
C
      F(1) =     -SIGMA*X(1)  +SIGMA*X(2)
      F(2) = (GAMMA-X(3))*X(1)-   X(2)
      F(3) =   X(1)*X(2)                  -B*X(3)
C
      RETURN
C
C
C
   10 CONTINUE
         FIRST = 1
C  ***************  HERE ALL PARAMETERS AND CONSTANTS OF THE
C   FUNCTION ARE DEFINED
C
```



```
         B    =  4.0D0
         SIGMA= 16.0D0
C
C  ***************  END OF PARAMETER AND CONSTANT DEFINITION
         GO TO 5
C
      END
```

Below is the output file.

```
PROBLEM DESCRIPTION
HOPF BIFURCATION OF THE LORENZ EQUATIONS

================================================================================

     N   JFREQ  JORB  JOB   ITP   INP   INT
     3     0     0     1     0     0     0

DIFFC :        DELTA                EPSZ                 MTH      ITNWTN
          0.1000000000D-06    0.1000000000D-13           1         3

PATHC :         HMIN                HMAX                EPSH                 SEN
          0.1000000000D-03    0.1000000000D+00    0.1000000000D-09    0.10000

           FACINC              FACDEC              EPSB
          0.2000000000D+01    0.5000000000D+00    0.1000000000D-11

          ITPATH    NSTEP
            3         0

================================================================================

       JFREQ      JORB       JOB
         0          0          1

      ISTEP     ITPATH     ITNWTN      MTH
        0          0          0          1

CONTROL PARAMETER     C     =  0.3345454545454545D+02

     STATE VECTOR
X( 1) =  0.1139377820646785D+02
X( 2) =  0.1139377820646785D+02
X( 3) =  0.3245454545454545D+02

   COSNP       COSNT       COSTP       H           S
 0.1000D+01  0.1000D+01  0.1000D+01  0.0000D+00  0.0000D+00

BIFUR  ==========================================================
```

```
HOPF BIFURCATION: TPER =     0.4467316942D+00

CRITICAL PARAMETERVALUE      0.3345454545D+02
CRITICAL EIGENVALUE          0.1391192286D-07    0.1406478501D+02

    TANGENT OF PATH          NORMAL OF PERIODIC
TP( 1) =   0.3266D+00   :  NP( 1) =   0.4694D+00
TP( 2) =   0.8462D+00   :  NP( 2) =   0.2414D+00
TP( 3) =  -0.4210D+00   :  NP( 3) =   0.8494D+00
TP( 4) =   0.0000D+00
```


4. Beyond the facade of PATH

If you want to study the programs more closely, for instance to make additions or to change the code, you should be aware of some of the leading ideas in the programming of the routines.
The purpose of this section is therefore not to tell you how you should program. I want to tell how I have programmed. Only the consciously known rules are stated, of course; and these I have tried to work through 100 percent.

On the lowest level:

```
IMPLICIT LOGICAL*4 (A-Z)
```

ensures at compile-time, that all variables in assignment statements have correct type. To use the standard fortran conventions of implicit type declarations are dangerous, I believe. Array indices follow this system:

```
I     is row-index of a matrix
J     is column-index of a matrix
K     is index of a vector
II    is index in vectors or matrices in WRITE-statements
NM1   =  N-1
NP1   =  N+1
NP2   =  N+2
```

PATH defines some vectors and matrices itself. We shall here mention some of the arrays:

BR : basis of range, N * N matrix

BD : basis of domain, (N+1)*(N+1) matrix

Q : residual map

DQ : derivative of Q

LAMDA : complex vector of eigenvalues of DP = DQ + I

XOLD : the most recently found point on the path

XNEW : approximation to new point on the path

XQUA : NQUA*(N+2) matrix. Contains the iterates to be interpolated for bi-periodic solutions.

All these variables do not appear in the parameterlist of PATH. Storage of these are all taken from the work-vector W. The routine PATH is only a preprocessor to organize and check data; the actual computations are handed to the routine PATHX.

Whereever possible I have tried to use already existing and documented software. I mention

solving systems of ordinary differential equations

(DVERK,LSODA)

computing eigenvalues (EIGRF)

spline interpolation (ICSCCU,ICSEVU)

basic linear algebraic operations (DDOT,DAXPY,DSCAL,...)

solving linear systems of equations

(DGEFA,DGESL,ZGEFA,ZGESL,...)

We have used the IMSL Library Routines.

I have received much inspiration from a number of books on programming. Among the best I mention Kernighan and Plauger (1978) and Myers (1979).

REFERENCES

L.O. Chua and A. Ushida (1981) Algorithms for computing almost periodic steady-state response of nonlinear systems to multiple input frequencies, IEEE Trans. Circ. Sys., Vol.Cas-28, No.10, pp. 953-971.

P.E. Gill, W. Murray, M.A. Saunders, and M. Wright (1983) Computing forward-difference intervals for numerical optimization, SIAM J. Sci. Stat. Comput., Vol.4, No.2, pp. 310-321.

B.D. Hassard, N.D. Kazarinoff, and Y-H. Wan (1981) Theory and applications of Hopf bifurcation, London Mathematical Society Lecture Note Series no. 41, Cambridge University Press, Cambridge.

IMSL Library Edition 9 (1982), IMSL, Inc., Houston, Texas.

Chr. Kaas-Petersen (1985a) Computation of quasi-periodic solutions of forced dissipative systems, to appear in J. Comput. Phys.

Chr. Kaas-Petersen (1985b) Computation, continuation, and bifurcation of torus solutions for dissipative maps and ordinary differential equations, submitted to Physica D.

Chr. Kaas-Petersen (1985c) Continuation methods as the link between perturbation analysis and asymptotic analysis, submitted to SIAM Review.

Chr. Kaas-Petersen (1985d) Computation of quasi-periodic solutions of forced dissipative systems II, submitted to J. Comput. Phys.

B.W. Kernighan and P.J. Plauger (1978) The elements of programming style, McGraw-Hill.

H.A. Lauwerier (1983) Local bifurcation of a logistic delay map, Stichting Mathematisch Centrum, Amsterdam.

A.J. Lichtenberg and M.A. Lieberman (1983) Regular and stochastic motion, Springer-Verlag.

G.J. Myers (1979) The art of software testing, Wiley.

I. Shimada and T. Nagashima (1979) A numerical approach to ergodic problems of dissipative dynamical systems, Prog. Theor. Phys., Vol.61, No.6, pp. 1605-1616.

Y. Ueda (1980) Steady motions exhibited by Duffing's equation: a picture book of regular and chaotic motions, in P. Holmes (ed.) New approaches to nonlinear problems in dynamics, SIAM, Philadelphia.

A . Short description of variables

We describe the variables known to the general user.

Variable name
1) type
2) length
3) appear in
4) meaning
5) allowed range
6) default value
7) comments
8) reference

BAS
1) REAL*8
2) N+1
3) parameterlist of DOFQ
4) direction-vector for partial derivative of Q
5) any non-zero vector
6) (1,0,...,0)
7) any specified vector is scaled to a unit-vector, the last element is in the parameter direction
8) App. D

C
1) REAL*8
2) 1
3) COMMON block /PARAM/
4) external control parameter of system examined
5) no restrictions
6) unknown
7) must be specified by the user
8) Sec. 1.2, 2.2, 2.3

DELTA
1) REAL*8
2) 1
3) COMMON block /DIFFC/
4) steplength in numerical differentiation of residual map
5) positive
6) square root of EPSZ
7) an optimal steplength can be found using DOFQ
8) Sec. 2.8, App. D

DMAX
1) REAL*8
2) 1
3) parameterlist of XWIND
4) strobes closer to initial point than DMAX are printed out
5) positive
6) 100.0D0
7)
8) Sec. 2.7

DTMAXR
1) REAL*8
2) 1
3) COMMON block /PMAPC/ parameterlist of FTRET
4) in order to hit the return hypersurface for autonomous ordinary differential equations no longer time-steps than DTMAXR are allowed
5) positive
6) TPER/100
7)
8) Sec. 2.6

E
1) REAL*8
2) N*M
3) parameterlist of CHAOS
4) the columns of E are edges in a volume element in state space
5) any linearly independent set of N-vectors
6) must be specified by the user
7) any set will be orthonormalized
8) App. C

EPSA
1) REAL*8
2) 1
3) COMMON block /DERIV/
4) upper bound on computational accuracy of residual map
5) positive
6) if not specified, it will be computed
7) will always be less than 1
8) App. D

EPSB
1) REAL*8
2) 1
3) COMMON block /PATHC/
4) a bifurcation point on the path is said to be located when two points on the path closer than EPSB (one point on each side of the bifurcation point) have been determined
5) greater than or equal to EPSZ
6) 100*EPSZ
7)
8) Sec. 2.10, 2.12

EPSH
1) REAL*8
2) 1
3) COMMON block /PATHC/
4) the norm of the residual map must be less than EPSH
to ensure that Newton's method can converge
5) greater than 100*EPSZ
6) quartic root of EPSZ
7) should only be small near saddle node bifurcations
8) Sec. 2.10

EPSR
1) REAL*8
2) 1
3) COMMON block /PMAPC/
4) the orbit is said to hit the return-surface
when the orbit is closer than EPSR to the surface
5) positive but less than TOL
6) TOL/100
7) only used for autonomous ordinary differential equations
8) Sec. 2.6

EPSZ
1) REAL*8
2) 1
3) COMMON block /DIFFC/
4) a zero of the residual map is said to be found when
the norm of the residual map is less than EPSZ
5) positive
6) depends on the system and the behaviour examined
7) greater than EPSA when EPSA is known
8) Sec. 2.8, App. D

EVCRIT
1) COMPLEX*16
2) 1
3) parameterlist of PATH
4) critical eigenvalue
5) either close to the imaginary axis in the complex plane
or close to the unit circle in the complex plane
6) none
7) only used when JOB=1
8) Sec. 2.9

F
1) REAL*8
2) N
3) parameterlist of FCN
4) values of function-expressions in system examined
5)
6)
7) F is the vector of values for a given T,X, and C
(or other parameters/constants in the system)
8) Sec. 1.2, 2.3

FACDEC
1) REAL*8
2) 1
3) COMMON block /PATHC/
4) factor by which a steplength along the path is
decreased if the EPSH-condition is not satisfied
5) between 0 and 1
6) 0.5
7)
8) Sec. 2.10

FACINC
1) REAL*8
2) 1
3) COMMON block /PATHC/
4) factor by which a steplength along the path is
increased if the EPSH-condition is satisfied
5) between 1 and 10
6) 2.0
7)
8) Sec. 2.10

FCN
1) routine name
2)
3) parameterlist of CHAOS,DOFQ,FTRET,GRSTRO,GRXOFT,
PATH,XWIND
4) the routine computes the right hand side of
the equations governing the dynamical system
5)
6) must be specified by the user
7) EXTERNAL FCN must appear in the main program
8) Sec. 2.3

HA
1) REAL*8
2) 1
3) COMMON block /DERIV/
4) steplength used to find EPSA
5) positive and small
6) 1.0D-8
7)
8) App. D

HMAX
1) REAL*8
2) 1
3) COMMON block /PATHC/
4) maximal steplength along the path
5) greater than or equal to 10*HMIN
6) 100*HMIN
7)
8) Sec. 2.10

HMIN
1) REAL*8
2) 1
3) COMMON block /PATHC/
4) minimal steplength along the path
5) greater than or equal to 10*EPSB
6) 100*EPSB
7) If PATH wants to try a step smaller than HMIN
then PATH stops with an error condition
8) Sec. 2.10

HSTART
1) REAL*8
2) 1
3) COMMON block /DERIV/
4) steplength to start with in order to obtain
an optimal steplength for numerical differentiation
5) positive and small
6) found by DOFQ
7)
8) App. D

ICH
1) INTEGER*4
2) 1
3) parameterlist of CHAOS
4) indicator for computation of chaotic solution
5) zero or positive
6) unknown
7) this indicator must be initialized to zero at the first
call to CHAOS, at return from CHAOS it is the
number of renormalizations performed until now
8) App. C

IDEV
1) INTEGER*4
2) 1
3) parameterlist of GRSTRO,GRXOFT
4) device number for output
5) 10-99
6) none
7) if no FILEDEF is issued, the output is stored
in FILE FTXXF001, where XX is the value of IDEV
8) Sec. 2.14

IER
1) INTEGER*4
2) 1
3) parameterlist of CHAOS,DOFQ,FTRET,GRSTRO,GRXOFT,
PATH,XWIND,YWIND
4) error indicator
5) zero or positive
6) at return from a routine IER is zero if the routine
has worked without errors. Else IER is non-zero
7) the value of IER should always be checked after
a call to a routine
8)

INP
1) INTEGER*4
2) 1
3) parameterlist of DOFQ,FTRET,GRSTRO,PATH
4) indicator for NP
5) 0 or 1
6) 0
7) INP and NP are only used for periodic or bi-periodic
orbits for autonomous ordinary differential equations
NP is specified when INP=1
8) Sec. 2.6

INT
1) INTEGER*4
2) 1
3) parameterlist of DOFQ,PATH
4) indicator of NT
5) 0 or 1
6) 0
7) INT and NT are only used for bi-periodic orbits for
autonomous and periodically forced ordinary differential
equations or difference equations
8) Sec. 2.6

IODE
1) INTEGER*4
2) 1
3) COMMON block /EQSOL/
4) ordinary differential equation indicator
5) positive
6) must be specified by the user
7) at present the values 1,2,3,4 are allowed
8) Sec. 2.5

IPRINT
1) INTEGER*4
2) 1
3) parameterlist of PATH
4) print indicator
5) 1,2,3
6) must be specified by the user
7) normally 2 will be a good value
8) Sec. 2.9, 2.13

IQUA
1) INTEGER*4
2) 1
3) parameterlist of XWIND,YWIND
4) strobes number 1,2,3,...,IQUA are computed
5) positive
6) must be specified by the user
7)
8) Sec. 2.7

ITNWTN
1) INTEGER*4
2) 1
3) COMMON block /DIFFC/
4) maximal number of iterations allowed
in Newton's method (MTH >= 1) or
in brute force method (MTH = 1)
5) positive
6) 3
7)
8) Sec. 2.8

ITP
1) INTEGER*4
2) 1
3) parameterlist of PATH
4) indicator for TP
5) 0 or 1
6) 0
7) when ITP=1 then the TP-vector is used to define the
residual map for which a zero is found
8) Sec. 2.10

ITPATH
1) INTEGER*4
2) 1
3) COMMON block /PATHC/
4) maximal number of iterations allowed
to in- or decrease steplength along the path
5) positive
6) 3
7)
8) Sec. 2.10

ITRET
1) INTEGER*4
2) 1
3) parameterlist of FTRET
COMMON block /PMAPC/
4) maximal number of iterations allowed
to locate crossing of return-plane
5) greater than or equal to 3
6) 20
7)
8) Sec. 2.6

IW
1) INTEGER*4
2) 1
3) parameterlist of CHAOS,DOFQ,FTRET,GRSTRO,PATH
4) length of workvector W
5) any positive value
6) 1
7) PATH stops if IW is too small to satisfy its needs
8) Sec. 2.4

IX0
1) INTEGER*4
2) 1
3) parameterlist of FTRET
4) X0 indicator
5) no restrictions
6) unknown
7) when IX0=1 then X0 is specified and used by FTRET
8) Sec. 2.6

JOB
1) INTEGER*4
2) 1
3) parameterlist of PATH
4) indicator for job to be performed
5) 1,2,3,4,5,6
6) must be specified by the user
7)
8) Sec. 2.9

JORB
1) INTEGER*4
2) 1
3) parameterlist of DOFQ,PATH
4) indicator for behaviour of solution
5) 0,1,2
6) must be specified by the user
7)
8) Sec. 2.4

JSYS
1) INTEGER*4
2) 1
3) parameterlist of CHAOS,DOFQ,GRSTRO,PATH,XWIND
4) number of periods in the dynamical system
5) -2,-1,0,+1,+2
6) must be specified by the user
7)
8) Sec. 2.4

LIAX
1) REAL*8
2) M
3) parameterlist of CHAOS
4) Liapunov exponent(s)
5)
6)
7) are computed before return from CHAOS
8) App. C

LRW

1) INTEGER*4
2) 1
3) COMMON block /EQSOL/
4) length of workvector RW
5) any positive value
6) 1
7) PATH stops if LRW is too small to satisfy its needs
8) Sec. 2.4, 2.5

M

1) INTEGER*4
2) 1
3) parameterlist of CHAOS
4) number of Liapunov exponents
5) 1,2,...,N
6) must be specified by the user
7)
8) App. C

MTH

1) INTEGER*4
2) 1
3) COMMON block /DIFFC/
4) indicator of Newton-method to be used
5) 0,1,2,...
6) 1
7) MTH=0 means brute force iteration
   (no derivatives computed)
8) Sec. 2.8

N

1) INTEGER*4
2) 1
3) parameterlist of CHAOS,FCN,FTRET,GRSTRO,GRXOFT,
   OSTRO,OXOFT,PATH,XWIND
4) number of function-expression in system examined
5) any positive number
6) unknown
7) must be specified by the user
   N is denoted n in Sec. 1.2
8) Sec. 2.3

NEPSA

1) INTEGER*4
2) 1
3) COMMON block /DERIV/
4) number of steps to be used in computations of EPSA
5) 5,6,7,8,...
6) 12
7)
8) App. D

NP

1) REAL*8
2) N+1
3) parameterlist of DOFQ,FTRET,GRSTRO,PATH,XWIND
4) normal of return-hypersurface or equivalently
   an approximation to the tangent of the orbit
   in the state space
5)
6) vector parallel to flow in the initial point
7) only the N first elements can be specified
   any vector is scaled to a unitvector
   vector used for autonomous ordinary differential eq.
8) Sec. 2.4, 2.6

NPOINT

1) INTEGER*4
2) 1
3) parameterlist of GRSTRO,GRXOFT
4) number of extra points on the solution (the initial
   point is always outputted)
5) positive
6) must be specified by the user
7) the format of the output must be written in the
   routines OSTRO,OXOFT
8) Sec. 2.14

NQUA

1) INTEGER*4
2) 1
3) COMMON block /QUASI/
4) number of points between which interpolation is done
5) 2,3,4,...
6) must be specified by the user
7) only used if JORB=2
8) Sec. 2.7

NSTEP

1) INTEGER*4
2) 1
3) COMMON block /PATHC/
   parameterlist of CHAOS
4) maximal number of steps to be found on the path
5) zero or positive
6) 0 for JOB=1,2,5
   10 for JOB=3,4,6
7) PATH may stop before NSTEP steps has been taken
8) Sec. 2.10, App. C

NT

1) REAL*8
2) N+1
3) parameterlist of DOFQ,PATH,XWIND
4) normal of hypersurface transversal to strobed orbit
   or equivalently an approximation to the strobed orbit
5)
6) vector parallel to strobed orbit in the initial point
7) only the N first elements can be specified
   any vector is scaled to a unitvector
   vector used for bi-periodic solutions
8) Sec. 2.4, 2.6, 2.7

OSTRO
1) routine name
2)
3) parameterlist of GRSTRO
4) the routine writes out computed results
5)
6) must be specified by the user
7) EXTERNAL OSTRO must appear in the main program
8) Sec. 2.14

OXOFT
1) routine name
2)
3) parameterlist of GRXOFT
4) the routine writes out computed results
5)
6) must be specified by the user
7) EXTERNAL OXOFT must appear in the main program
8) Sec. 2.14

PROJ
1) REAL*8
2) N
3) parameterlist of DOFQ
4) direction-vector for projection of
   partial derivative of Q
5) any non-zero vector
6) (1,0,...,0)
7) any specified vector is scaled to a unit-vector
8) App. D

RW
1) REAL*8
2) LRW
3) COMMON block /EQSOL/
4) real workspace to iterate map or to solve
   ordinary differential equations
5) no restrictions
6) unknown
7) values of this vector is nonsense to the user
8) Sec. 2.4, 2.5

SEND
1) REAL*8
2) 1
3) COMMON block /PATHC/
4) maximal allowed arclength along the path
5) greater than or equal to HMIN
6) 10*HMAX
7)
8) Sec. 2.10

T
1) REAL*8
2) 1
3) parameterlist of CHAOS,FCN,FTRET,GRSTRO,GRXOFT,OXOFT
4) time
5) no restrictions
6) unknown
7)
8) Sec. 2.3, 2.6

TAU
1) REAL*8
2) 1
3) parameterlist of CHAOS
4) renormalization time
5) positive
6) must be specified by the user
7)
8) App. C

TEND
1) REAL*8
2) 1
3) parameterlist of FTRET,GRXOFT
4) end value of time
5) greater than initial value
6) must be specified by the user
7)
8) Sec. 2.6, 2.14

TMOD
1) REAL*8
2) 1
3) parameterlist of OSTRO
4) (strobe-time T divided by TQUA) modulo 1
5) 0 to 1
6)
7) used when JORB=2 and (JSYS=-2 or JSYS=+2)
8) Sec. 2.14

TOL
1) REAL*8
2) 1
3) COMMON block /EQSOL/
4) upper bound on local truncation error in numerical
   solution of ordinary differential equations
5) positive
6) 1.0D-6
7)
8) Sec. 2.5, 2.8

TP

1) REAL*8
2) N+1
3) parameterlist of PATH
4) tangent of path or equivalently a unit normal vector
   of the domain of the residual map
5) any non-zero vector
6) TP will be computed if needed and not specified
7) any specified vector is scaled to a unit vector
   the (N+1)st element is in the parameter direction
8) Sec. 2.4, 2.10

TPER

1) REAL*8
2) 1
3) parameterlist of DOFQ,GRSTRO,PATH
4) period
5) positive
6) must be specified by the user
7) TPER is denoted T in Sec. 1.2
8) Sec. 2.4

TRET

1) REAL*8
2) 1
3) parameterlist of XWIND
4) time to return to hypersurface
5) positive
6) must be specified by the user
7)
8) Sec. 2.7

TQUA

1) REAL*8
2) 1
3) parameterlist of DOFQ,GRSTRO,PATH
4) second period in quasi-periodically forced system
5) positive and greater than TPER
6) must be specified by the user
7) TQUA is denoted T' in Sec. 1.2
8) Sec. 2.4

X

1) REAL*8
2) N
3) parameterlist of CHAOS,FCN,FTRET,GRSTRO,GRXOFT,
   OSTRO,OXOFT,PATH,XWIND
4) state variables of system examined
5) no restrictions
6) unknown
7) must be specified by the user
8) Sec. 1.2, 2.3, 2.4

X0

1) REAL*8
2) N
3) parameterlist of FTRET
4) origin of return-hypersurface
5) no restrictions
6) X
7) when X0 is specified by the user, then IX0=1
8) Sec. 2.6

W

1) REAL*8
2) IW
3) parameterlist of CHAOS,DOFQ,FTRET,GRSTRO,
   PATH,XWIND
4) real workspace
5) no restrictions
6) unknown
7) for JORB=2 or JORB=-2 the NQUA first elements
   are the times at
   which the solution is wanted; in all other cases
   the values of this vector is nonsense to the user
8) Sec. 2.4

WIND

1) REAL*8
2) 1
3) parameterlist of YWIND
4) winding number
5) 0 to 1
6) must be specified by the user
7) for JSYS=2 or JSYS=-2 then WIND=TPER/TQUA, else
   WIND is only known near bifurcation points
8) Sec. 2.7

B . Parameterlists of routines & lists of named COMMON-areas

Parameterlists of routines

```
CHAOS   (JSYS,FCN,N,T,X,TAU,NSTEP,E,M,ICH,LIAX,IER,W,IW)

DOFQ    (JSYS,JORB,FCN,N,X,TPER,TQUA,NP,INP,NT,INT,
         BAS,PROJ,IPRINT,IER,W,IW)

FCN     (N,T,X,F)

FTRET   (FCN,N,T,X,TEND,NP,INP,X0,IX0,DTMAXR,ITZERO,EPSR,IER,W,IW)

GRSTRO  (JSYS,FCN,N,T,X,TPER,TQUA,NPOINT,NP,INP,OSTRO,IDEV,IER,W,IW)

GRXOFT  (FCN,N,T,X,TEND,NPOINT,OXOFT,IDEV,IER)

OSTRO   (TMOD,X,N,IDEV)

OXOFT   (T,X,N,IDEV)

PATH    (FCN,N,X,TPER,TQUA,TP,ITP,NP,INP,NT,INT,EVCRIT,
         JSYS,JORB,JOB,IPRINT,IER,W,IW)

XWIND   (JSYS,FCN,N,X,TRET,NP,NT,IQUA,DMAX,IER,W,IW)

YWIND   (WIND,IQUA,IER)
```


List of named COMMON-areas

```
      REAL*8          C
      COMMON /PARAM/ C

      INTEGER         IODE,LRW
      REAL*8                   TOL,RW(4)
      COMMON /EQSOL/ IODE,LRW,TOL,RW

      INTEGER                      MTH,ITNWTN
      REAL*8          DELTA,EPSZ
      COMMON /DIFFC/ DELTA,EPSZ,MTH,ITNWTN

      INTEGER                       ITRET
      REAL*8          EPSR,DTMAXR
      COMMON /PMAPC/ EPSR,DTMAXR,ITRET

      INTEGER         ITPATH,NSTEP
      REAL*8          HMIN,HMAX,EPSH,SEND,FACINC,FACDEC,EPSB
      COMMON /PATHC/ HMIN,HMAX,EPSH,SEND,FACINC,FACDEC,EPSB,
     *                ITPATH,NSTEP

      INTEGER         NQUA
      COMMON /QUASI/ NQUA

      INTEGER                         NEPSA
      REAL*8          EPSA,HA,HSTART
      COMMON /DERIV/ EPSA,HA,HSTART,NEPSA
```

C . Chaotic behaviour

Chaotic behaviour is observed when the maximal Liapunov exponent is positive. For definition and computation of Liapunov exponents we refer to Lichtenberg and Lieberman (1983) pp. 262-268 & pp. 277-285.

CHAOS and PATH do not interact.

The parameter list of CHAOS is:

```
CHAOS(JSYS,FCN,N,T,X,TAU,NSTEP,E,M,ICH,LIAX,IER,W,IW)
```

In the first call ICH must be zero. ICH must not be changed by the user. ICH holds the total number of renormalizations performed. T and X must be initial conditions for the system in FCN. The system is characterized by JSYS. TAU is the renormalization time. M is the number of Liapunov exponents to be computed. The edges of a small volume-element in the state space must be given in the N*M-matrix E. The routine performs NSTEP renormalizations before computing the Liapunov exponents, which are stored in the M-vector LIAX at return time. The workspace W has length IW, and IW >= 2*N+2*M+NSTEP*M. When IER=0 at return time the routine has worked succesfully.

The named COMMON-area EQSOL must be included in the program calling CHAOS.

TAU should be an average time for one oscillation.

The routine CHAOS organizes the computations, which are performed by CHAOSX, LIASTP, VSRTAD, VSRPOS, and VSRSUM.

See example no. 5.

D . Numerical differentiation

For a real function f of a real variable the derivative of f in the point x is defined by

derivative of f in x = f'(x) =

limes (f(x+h)-f(x))/h for h approaching 0 .

We cannot perform the limit on a computer but we can come close to it. The routine DOFQ performs differentiation of the residual maps or of any other map using the method described in Gill, Murray, Saunders, and Wright (1983). The purpose of DOFQ is to supply you with proper values of the two variables DELTA and EPSZ in the named COMMON area DIFFC; not to supply you with the actual value of the derivative.

The call is of the form:

```
DOFQ (JSYS,JORB,FCN,N,X,TPER,TQUA,NP,INP,NT,INT,
      BAS,PROJ,IPRINT,IER,W,IW)
```

The vector BAS has length N+1 and is the direction in which to make increments (the last element of BAS is the parameter direction). PROJ is a N-vector and gives the direction in which to project the map. BAS and PROJ are made orthonormal to NP and NT (if these are needed to define the residual map). The array W has IW elements, and IW >= 5*(N+1)+NEPSA. For bi-periodic solutions IW >= 5*(N+1)+NEPSA+NQUA*(N+2), and the first NQUA locations of W must contain the stop-times for the stroboscopic map, as described in Sec. 2.7.

DOFQ is connected with PATH. They share the named COMMON areas: PARAM, EQSOL, PMAPC, and QUASI. Furthermore DOFQ has the named COMMON area DERIV, in this way:

```
INTEGER                          NEPSA
REAL*8          EPSA,HA,HSTART
COMMON /DERIV/ EPSA,HA,HSTART,NESPA
```

DOFQ gives you the value EPSA and an optimal steplength HOPT (which is always written). If you specify an EPSA > 0 that value of EPSA is used in the algorithm. If EPSA <= 0 then DOFQ will compute it using NEPSA function-values spaced the distance HA apart. To compute the optimal steplength, the first steplength tried is HSTART (when specified). You can then take EPSZ = 100*EPSA , and DELTA = HOPT.

The routine DOFQ organizes the computations, which are performed by NPUNIT, NTUNIT, COMPVE, DOFQX, RESMAP, ACCOFF, CDF, and OPTFDI.

We describe now the output from DOFQ when IPRINT=2. Below the horizontal line

```
ACCOFF ================================
```

the value (either computed or specified) of EPSA is printed. Then under the horizontal line

```
CDF    ================================...
```

the results of one iteration to find the optimal steplength is printed.

F — function value in the point where the derivative is wanted

H — the step-length used

PHIF — derivative of F computed with a forward difference approximation

EBPHIF — error in PHIF (or at least an upper bound on the error)

PHIB — derivative of F computed with a backward difference approximation

EBPHIB — error in PHIB (or at least an upper bound on the error)

PHIC — derivative of F computed with a central difference approximation

PHI2 — second derivative of F

EBPHI2 — error in PHI2 (or at least an upper bound on the error).

Under the horizontal line

```
OPTFDI ================================...
```

the final results are given.

IT — the number of iteration performed

HS — the smallest step-length tried for which EBPHIF and EBPHIB are acceptable

HOPT — the optimal steplength

F — function value

D1 — first derivative with optimal step-length used

ERRD — error in D1

D2 — second derivative with optimal step-length used.

E . Extensions/modifications of PATH - this is future

Here we list some extensions, that deeply will influence the code. The design can more or less be kept.

1) The derivatives af the map are computed numerically. It would be nice to have an option for an analytic jacobian. Note, that PATH computes a derivative, and also LSODA computes a derivative. Thus the user should be allowed to program a subroutine JAC(N,T,X,DF) and an indicator IJAC should tell PATH whether or not to use the JAC-subroutine.

2) For bi-periodic solutions, the winding number may change. An option for updating the strobe-numbers to be interpolated could be introduced.

3) Only one parameter in the problem can be used. It would be smart, to define an arbitrary curve to be followed when more parameters are present.

4) The derivatives are full matrices, and all eigenvalues are computed. But we only need the eigenvalues closest to criticallity. This could give some savings.

5) The strategy for taking a step along the path is very simple. More advanced methods are known, and could be applied.

F . Short description of all routines in PATH

ACCOFF	Accuracy of function f. CKP routine.
ADJGS	For a given set of linearly independent vectors a full set a basis vectors is found. The are made orthonormal. CKP routine.
AMPPER	Maximal, minimal, and amplitude of periodic solution. CKP routine.
AUTBIP	Computes strobes of bi-periodic solution of an autonomous ODE. CKP routine.
AUTPER	Computes periodic solution of an autonomous ODE. CKP routine.
BIFBIP	Computes strobes of bi-periodic solution of a bi-periodically forced system (either ODEs or IMs). CKP routine.
BIFUR	Bifurcation analysis. CKP routine.
CDF	Computes derivative of function f. CKP routine.
CHAOS	Sets up work spaces and basis vectors needed for the Liapunov exponents. CKP routine.
CHAOSX	Called by CHAOS to organize computations of the maximal Liapunov exponents. CKP routine.
CINVP	Performs inverse power method on a complex matrix. CKP routine.
COMPBD	Computes orthonormal basis vectors of domain of residual map. CKP routine.
COMPBR	Computes orthonormal basis vectors of range of residual map. CKP routine.
COMPVE	Computes orthonormal vectors. CKP routine. Called by DOFQ.
CRITIC	Regula falsi to locate the bifurcation point. CKP routine
DAXPY	Real constant times a vector plus a vector. IMSL routine.
DCOPY	Copies the real x-vector into the real y-vector. IMSL routine.
DDOT	Dot product of two real vectors. IMSL routine.
DEULER	Differential equation solver - Euler's method. CKP routine.
DNRM2	Euclidean norm of the real x-vector. IMSL routine.
DGEFA	Factors a real matrix by gaussian elimination. LINPACK routine.
DGESL	Solves the real system A*X = B using the factors computed by DGEFA. LINPACK routine.
DOFQ	Sets up work spaces, find unitvectors, in order to evaluate derivative. CKP routine.
DOFQX	Derivative of residual map Q. CKP routine.
DPETZ	Differential equation solver - call the LSODA package from the Lawrence Livermore National Laboratory.
DSCAL	Constant times a real vector. IMSL routine.
DTRANS	Transposes a matrix. CKP routine.
DVERK	Differential equation solver - Runge-Kutta-Verner fifth and sixth order method. IMSL routine.
D34RK	Differential eqaution solver - Runge-Kutta third and fourth order method. CKP routine.
EIGRF	Eigenvalues and (optionally) eigenvectors of a real general matrix in full storage mode. IMSL routine.

ELLIPS	Minor and major axis of ellipse where two conjugated diameters are known. CKP routine.
FCN	Equations of dynamical system. USER routine.
FELSOR	Sorts the elements of a real matrix by value in a specified column. CKP routine.
FTRET	Finds the return time of an autonomous ODE. CKP routine.
GRSTRO	Computes points on the strobed orbit. CKP routine.
GRXOFT	Computes points on the orbit (t,x(t)). CKP routine.
HITMAP	Finds the point where the orbit hits. CKP routine.
ICSCCU	Cubic spline interpolation (easy-to-use version). IMSL routine.
ICSEVU	Evaluation of a cubic spline. IMSL routine.
IDAMAX	Index of element of real vector having maximum absolute value. IMSL routine.
IDAMIN	Index of element of real vector having minimal value. CKP routine.
ILIZE	Initializes parameters used in the path following method. CKP routine.
IZAMAX	Index of element of complex vector having maximum absolute value. CKP routine.
JACOBI	Jacobian matrix of vector field or residual map. CKP routine.
LIASTP	One renormalization step in the computation of some of the Liapunov exponents. CKP routine.
MAPX	Iterates the difference equations. CKP routine.
MODGS	Modified Gram Schmidt orthonormalization process. CKP routine.
MRFX	Modified regula falsi technique to locate a zero of a univariate function. CKP routine.
NEWBAS	Computes new basis when TP, NP, or NT have been updated. CKP routine.
NEWTON	Newton's method to find a zero. CKP routine.
NPUNIT	Normal of Poincare hypersurface is (optionally) found and made a unit vector. CKP routine.
NTUNIT	Normal of torus is (optionally) found and made a unit vector. CKP routine.
ODEX	Calls the specified ODE-solver. CKP routine.
OPTFDI	Optimal forward difference interval to approximate the derivative numerically. CKP routine.
OSTRO	Outputs points on the strobed orbit. CKP routine.
OXOFT	Outputs points on the orbit (t,x(t)). CKP routine.
PATH	Sets up workspaces. If not on the path, we return to the path. If in a bifurcation point, we take the first step away from that point. CKP routine.
PATHX	Organizes computations of the path following method. (Update path direction TP, return plane NP, and tangent of strobed orbit NT.) CKP routine.
PEFBIP	Bi-periodic solution of periodically forced system (either ODEs or IMs). CKP routine.
PEFPER	Periodic solution of periodically forced system (either ODEs or IMs). CKP routine.
PRSTA	Prints state and parameters of system. CKP routine.
PRVEC	Prints vectors TP and/or NP and/or NT. CKP routine.
RESMAP	Residual map. CKP routine.
RINVP	Inverse power method of real matrix. CKP routine.
SORTEV	Sorts eigenvalues found in the equilibrium point. CKP routine.
STABIL	Derivative of zero point map or fixed point map. Optionally the eigenvalues are found. CKP routine.
STEP	Steps along the path (TP-direction). CKP routine.

TANOFP	Tangent of path. CKP routine
T2STOP	Parametrization of interpolating cubic spline for bi-periodically forced systems. CKP routine.
VSRPOS	Index of element having smallest positive value in a sorted array. CKP routine.
VSRSUM	Sums a double precision array sorted by absolute value. CKP routine.
VSRTAD	Sorts double precision arrays by algebraic value. ISML routine.
XWIND	Computes return times when the winding number is not known. CKP routine.
YWIND	Computes return times when the winding number is known. CKP routine.
ZAXPY	Complex constant times a complex vector plus a complex vector. LINPACK routine.
ZDOTZ	Forms the dot product of two vectors, conjugating the first vector. LINPACK routine.
ZGEFA	Factors a complex matrix by gaussian elimination. LINPACK routine.
ZGESL	Solves the complex system A*X = B using the factors computed by ZGEFA. LINPACK routine.
ZGESLB	Back-substitution on the complex system A*X = B using the factors computed by ZGEFA. LINPACK routine.
ZSCAL	Complex constant times a complex vector. LINPACK routine.

32

A BIFURCATION ANALYSIS OF NONLINEAR OSCILLATIONS IN RAILWAY VEHICLES

Hans True and Christian Kaas-Petersen

Laboratory of Applied Mathematical Physics, The Technical University of Denmark, Bldg. 303, DK-2800 Lyngby, Denmark

SUMMARY

Bifurcation analysis is applied to Cooperrider's two models [1] of the motion of a bogie on rails. Both models consist of a system of autonomous nonlinear ordinary differential equations. The Hopf bifurcation analysis is performed numerically using a routine developed by Hassard et al. [2]. It turns out, that the bifurcation is subcritical. The maximum amplitude and frequency of the lateral oscillation corresponding to the new branch (i.e. the periodic solution) is then calculated numerically as a function of the speed of the car using a new routine.

1. THE MATHEMATICAL METHOD

One of the main difficulties that arise in the solution of nonlinear problems is, that the problems very often have several solutions for a given set of external data. By external data we shall mean those values of the parameters that can be controlled. In the case of differential equations they may be initial or boundary conditions. The "other"solutions may "break symmetries" or have an unexpected character like time-periodic solutions in problems, where the time variable does not occur explicitly at all.

When we attempt to solve a nonlinear problem the desirable goal is to determine all the unknown functions of the independent variables for all values of their domain of definition and for all parameter values in question. We are often able to do just that in linear problems and such a solution is called a global solution. Unfortunately most nonlinear problems cannot be solved globally, but many nonlinear problems have a particularly simple solution for a wide range of external parameter values - often so simple, that it can be guessed. We may then ask the question: "Do other solutions of this problem exist that in some way can be described as being "close" to the already known solution"? If the answer is yes, then we can often determine them. In contrast to "global solution" the problem of finding a solution that in some sense is close to an already known solution is called a "local solution". The bifurcation problems fall into this category.

Bifurcation or branching is a phenomenon, which occurs in nonlinear problems, when the number of solutions depends on one or more parameters, which characterize the problem. Examples of such parameters are the Reynold's number in fluid mechanics and the speed of the vehicle in vehicle dynamics.

If two or more solutions of a given nonlinear problem exist for a given value of a parameter λ, and the solutions have the property, that by changing the parameter a value of the parameter, λ_o say, can be found such that the solutions coalesce for $\lambda = \lambda_o$, and the solutions exist in every neighborhood of λ_o, then we call λ_o a bifurcation point or branch point of the problem. When we do a bifurcation analysis, a solution of the problem must be known. This solution is called the fundamental or basic solution, and we are primarily concerned about other solutions that bifurcate away from this basic solution.

The fundamental theorem that applies to bifurcation of time periodic solutions from a steady solution in ordinary differential equations is the Hopf Bifurcation Theorem.

We consider an autonomous system of differential equations of the form

$$\frac{dX}{dt} = F(X,\nu) \qquad \text{(F and X are vectors).}$$

Let the system

$$F(X,\nu) = 0$$

have the solution $X = X_o(\nu)$ and define $x = X - X_o(\nu)$ and substitute the new vector variable x into the system.

The system will then be

$$\frac{dx}{dt} = f(x,\nu) \qquad \text{with } f(0,\nu) = 0.$$

The theorem then states, that under certain conditions of a mathematical nature bifurcation of time periodic solutions occur at $\nu = \nu_c$ if and only if the linearized operator (the Jacobian) $f_x(0,\nu_c)$ has a pair of complex conjugate eigenvalues and all other eigenvalues have strictly negative real parts. The proof can be found in the annotated translation of Hopf's original paper [3].

The time periodic solution is well defined up to certain "translations". As an example we may mention, that the phase of a time periodic bifurcation will be unknown.

It is interesting to notice, that the wellknown linear stability theory in this case links up with bifurcation theory. In case of a simple eigenvalue the loss of stability of a fundamental solution is connected with the existence of a branch bifurcating at the parameter value, where the fundamental solution loses its stability. The bifurcating solution, however, needs not be stable. In the present case it is in fact unstable. The theory of Hopf bifurcation is treated in detail in [2]. There it is shown, that the bifurcating periodic solution $x(t,\nu)$, with period $T(\nu)$ has the expressions

$$x(t,\nu) = \sqrt{\frac{\nu-\nu_c}{\mu_2}}\,\mathrm{Re}\left(e^{2\pi i t/T(\nu)} v_1\right) + O(\nu-\nu_c) \tag{1}$$

$$T(\nu) = \frac{2\pi}{\omega_o}\left(1 + \tau_2 \frac{\nu-\nu_c}{\mu_2} + O(\nu-\nu_c)^2\right) \tag{2}$$

$$\beta(\nu) = \beta_2 \frac{\nu-\nu_c}{\mu_2} + O(\nu-\nu_c)^2 \tag{3}$$

valid for $\mu_2 \neq 0$. $\beta(\nu)$ is the Floquet exponent that determines the stability of the periodic solution. If $\beta(\nu) < 0$ the solution is stable, and if $\beta(\nu) > 0$ the solution is unstable. The calculations of the values of ν_c, μ_2, v_1, ω_o, τ_2, and β_2 is performed by the routine BIFOR2 developed by Hassard and listed in [2].

Now taking a value of ν near ν_c such that $(\nu - \nu_c)/\mu_2$ is positive, we have an approximation to the periodic solution. These values can be used as initial values in a computation of the periodic solution of the full nonlinear equations. We use a Poincaré mapping to find the periodic solution. For a definition of a Poincaré map see HASSARD [2]. The important property of the map is, that it maps a plane of one dimension less than the number of dependent variables into itself. The Poincaré mapping, P, thus maps a point on a periodic solution into itself, which is the same as finding a fixed point of the Poincaré mapping. If we then use this Poincaré mapping minus the identity, I, we are searching for a zero of the mapping P-I, which can be found rather effectively using Newton's method. The Jacobian of P-I is calculated numerically.

2. THE MODELS

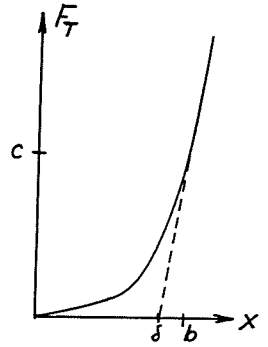
We have chosen the two models by Cooperrider [1]. We shall only briefly describe some important characteristics here; for details we refer the reader to the original article. The two models are called the simple model and the complex model. They were chosen, because they contain most of the features occurring in the modelling of the motion of railroad bogies, but they are on the other hand simple enough to avoid spending much computer time on refinements that are not appropriate for the initial tests of the applicability of bifurcation theory.

The bogie runs on a perfect, level, and straight track. All parts of the bogie is treated as stiff, and friction is only included in the wheel-rail-forces, see below. Movements are assumed small. Flange forces were originally described as rather stiff linear springs with a dead band, see Fig. 1, where δ is half the clearance between flange and railhead. This function has a abrupt change in the derivative at $x = \delta$ and this is not desirable when numerical integration is used. The numerical method (we use Gear's package for solving ordinary differential equations, which use methods of variable order and variable steplength) has to locate the point very accurately. This is avoided by using

$$F_T(x) = \begin{cases} \alpha \cdot \exp\left(-\frac{\alpha\, 1}{(x-x_f)}\right) + \beta x + \kappa & \text{for } 0 \le x < b \\ k_o(x-\delta) & \text{for } b \le x \end{cases} \qquad (4)$$

$$F_T(x) = -F_T(-x) \quad \text{for } x < 0$$

Here α, β, κ, and x_f is determined such that continuity and differentiability is ensured at $x = 0$ and at the mathcing point $x = b$. The value b was chosen such that $F_T(b) = c$, where c was given a suitable value; we used $c = 10^2$. The values used are as follows.



α = 0.1474128791127777E-03
β = 0.1016261260305364E+01
κ = 0.1793756792976414E+01
δ = 0.91E-02
b = 0.9106849315068492E-02
x_f = 0.9138788366764013E-02

Fig. 1 The approximation of the flange force.

The wheel was assumed conical, so no gravitational stiffness was included.

Parameters in description of wheel-rail geometry has the values

a = 0.7163 m λ = 0.05

δ = 0.0091 m r_o = 0.4572 m

where a is half the track gauge, λ is the wheel conicity, 2δ the rail-flange clearance and r_o the centered wheel rolling radius.

As creep forces we used an approximation similar to that proposed by Vermeulen and Johnson, where creep-force saturation is taken into account (but spin-creep-forces have been excluded). Thus

$$F_x = \frac{\xi_x}{\phi}\frac{F_R}{\xi_R} \qquad F_y = \frac{\xi_y}{\psi_1}\frac{F_R}{\xi_R} \qquad (5)$$

$$\frac{F_R}{\mu N} = \begin{cases} u - \frac{1}{3}u^2 + \frac{1}{27}u^3 & \text{for } u < 3 \\ 1 & \text{for } u \ge 3 \end{cases}, \quad u = \frac{G\pi ab}{\mu N}\xi_R \qquad (6)$$

$$\xi_R = \sqrt{\frac{\xi_x^2}{\phi^2} + \frac{\xi_y^2}{\psi_1^2}}$$

The mass N_{tot} is distributed equally on the 4 wheels. With N_{tot}=22248 kg together with the mass of the bogie, which is 4963 kg, the vertical force between wheel and rail is N=6803 kg. The coefficient of friction was set to μ=0.15 and wheel and rail are made of steel. The Hertz contact theory yields the values

ϕ = 0.54219 ψ_1 = 0.60252

$G\pi ab = 6.563 \cdot 10^6$ N μN = 10000 N.

Note however, that due to the square root, the functions F_x and F_y are not differentiable at x=0. Therefore instead of Cooperrider's functions eq. (6) we use

$$\frac{F_R}{\mu N} = \tanh u \quad , \quad \text{for } u \ge 0 \ , \quad u = \frac{G\pi ab}{\mu N}\xi_R \qquad (7)$$

This approximation was only used in BIFOR2. When the characteristics of the bifurcation were determined we use the expressions eqs. (5)-(6) in all further calculations.

The simple bogie model is shown on Fig. 2.

The governing equations are listed below.

$$m\ddot{q}_1 + 2D_2\dot{q}_1 + 2k_4q_1 + 2F_x(\xi_{x1},\xi_{y1}) + 2F_x(\xi_{x2},\xi_{y2}) + F_T(q_1+haq_2) + F_T(q_1-haq_2) = 0$$

$$I\ddot{q}_2 + k_6q_2 + 2ha\{F_x(\xi_{x1},\xi_{y1}) - F_x(\xi_{x2},\xi_{y2})\} + 2a\{F_y(\xi_{x1},\xi_{y1}) + F_y(\xi_{x2},\xi_{y2})\} + ha\{F_T(q_1+haq_2) + F_T(q_1-haq_2)\} = 0$$

These eq.s give the results below. However, a factor 2 ought to be in the position indicated

where

$$\xi_{x1} = \frac{\dot{q}_1}{V} + ha\frac{\dot{q}_2}{V} - q_2 \qquad \xi_{y1} = \frac{a}{V}\dot{q}_2 + \frac{\lambda}{r_o}(q_1+haq_2)$$

$$\xi_{x2} = \frac{\dot{q}_1}{V} - ha\frac{\dot{q}_2}{V} - q_2 \qquad \xi_{y2} = \frac{a}{V}\dot{q}_2 + \frac{\lambda}{r_o}(q_1-haq_2)$$

which by the substitution

$$x_1 = q_1 , \quad x_2 = q_2 , \quad x_3 = \dot{q}_1 , \quad x_4 = \dot{q}_2$$

give a system of 4 first order ordinary differential equations.
The parameter-values used are:

$h = 1.5$ $\qquad k_o = 14.60 \cdot 10^6$ N/m

$m = 4963$ kg $\qquad k_4 = 0.1823 \cdot 10^6$ N/m

$I = 8135$ kg-m^2 $\qquad k_6 = 2.710 \cdot 10^6$ N/m

$D_2 = 29200$ N-s/m

which give

$$V_c = 99.109 \text{ m/s} \quad \mu_2 = -81378.1 \text{ m/s} \quad \beta_2 = 6550.1$$

Taking V = 99.0 m/s, values of x and T near the numerical solution can be found using the formulas listed in the methods section. The Poincaré mapping then finds a point on the periodic solution using 7 iterations. We use 10^{-8} as an upper bound on the local error in the integration routine. The value 7 is rather high. This is because we are near the bifurcation point where $\beta(\nu)$ is near zero, and the solutions are therefore only weakly unstable. We get the values

T = 0.4226636396E+00

x(1) = 0.1161566560E-02

x(2) = 0.1657575932E-04

x(3) = 0.9505939847E-04

x(4) = 0.2904129383E-02

Using these values as initial values for a new choice of V we can trace the bifurcating periodic solution.
The complex bogie model is shown on Fig. 3.
The governing equations are listed below.

$$m_w\ddot{q}_1 + 2k_1q_1 - 2k_1q_5 - 2k_1haq_6 - 2k_1h_1q_7 + 2F_x(\xi_{x1},\xi_{y1}) + F_T(q_1) = 0$$

$$I_{w_2}\ddot{q}_2 + 2k_2d_1^2q_2 - 2k_2d_1^2q_6 + 2a \cdot F_y(\xi_{x1},\xi_{y1}) = 0$$

$$m_w\ddot{q}_3 + 2k_1q_3 - 2k_1q_5 + 2k_1haq_6 - 2k_1h_1q_7 + 2F_x(\xi_{x2},\xi_{y2}) + F_T(q_3) = 0$$

$$I_{w_2}\ddot{q}_4 + 2k_2d_1^2q_4 - 2k_2d_1^2q_6 + 2a \ F_y(\xi_{x2},\xi_{y2}) = 0$$

$$m_F\ddot{q}_5 + 2D_2\dot{q}_5 + (4k_1+2k_4)q_5 - 2k_1q_1 - 2k_1q_3 - 2D_2h_2\dot{q}_7 + (4k_1h_1 - 2k_4h_2)q_7 = 0$$

$$I_{F_3}\ddot{q}_6 + (4k_1h^2a^2 + 4k_2d_1^2 + k_6)q_6 - 2hak_1q_1 + 2hak_1q_3 - 2k_2d_1^2q_2 - 2k_2d_1^2q_4 = 0$$

$$I_{F_2}\ddot{q}_7 + (2D_1d_2^2 + 2D_2h_2^2)\dot{q}_7 + (4k_1h_1^2 + 4k_3d_1^2 + 2k_4h_2^2 + 2k_5d_2^2)q_7$$

$$-2k_1h_1q_1 - 2k_1h_1q_3 - 2D_2h_2\dot{q}_5 + (4k_1h_1 - 2k_4h_2)q_5 = 0$$

where

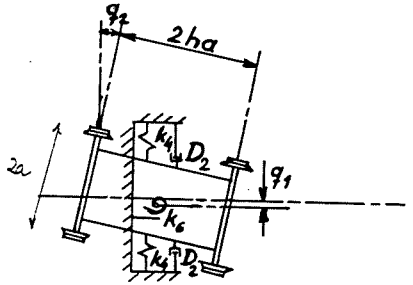
$$\xi_{x1} = \frac{\dot{q}_1}{V} - q_2 \qquad \xi_{y1} = \frac{a}{V}\dot{q}_2 + \frac{\lambda}{r_o}q_1$$

$$\xi_{x2} = \frac{\dot{q}_3}{V} - q_4 \qquad \xi_{y2} = \frac{a}{V}\dot{q}_4 + \frac{\lambda}{r_o}q_3$$

which by the substitution

$x_1 = q_1$, $\quad x_2 = \dot{q}_1$, $\quad x_3 = q_2$, $\quad x_4 = \dot{q}_2$, ...

give a system of 14 first order ordinary differential equations.



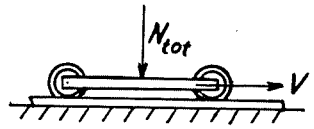
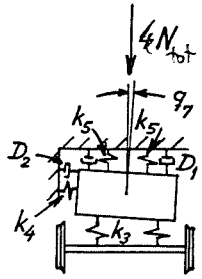


Fig. 2. Simple truck model.



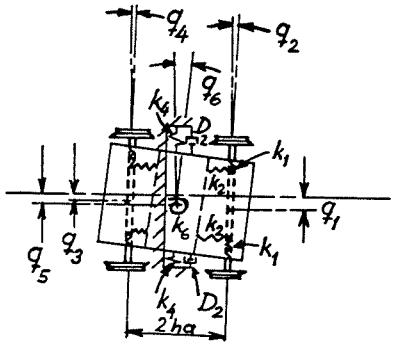


Fig. 3 Complex truck model

The parameter-values used are:

$h = 1.5$

$h_1 = 0.0762$ m

$h_2 = 0.6584$ m

$d_1 = 0.5852$ m

$d_2 = 1.079$ m

$m_w = 1022$ kg

$m_F = 2918$ kg

$I_{w_2} = 678$ kg-m^2

$I_{F_2} = 6780$ kg-m^2

$I_{F_3} = 6780$ kg-m^2

$D_1 = 20000$ N-s/m

$D_2 = 29200$ N-s/m

$k_o = 14.60 \cdot 10^6$ N/m

$k_1 = 1.823 \cdot 10^6$ N/m

$k_2 = 3.646 \cdot 10^6$ N/m

$k_3 = 3.646 \cdot 10^6$ N/m

$k_4 = 0.1823 \cdot 10^6$ N/m

$k_5 = 0.3333 \cdot 10^6$ N/m

$k_6 = 2.710 \cdot 10^6$ N/m

which give

$V_c = 66.269$ m/s $\quad \mu_2 = -16265.9$ m/s $\quad \beta_2 = 3994.9$.

3. RESULTS

The results of our calculations are shown on the Figs. 4-6. They are all typical bifurcation diagrams, where the maximum amplitude of the lateral oscillation and its frequency are shown as functions of the speed of the vehicle. The other kinds of oscillation will show a similar behavior as a result of the nonlinear coupling between the various modes of motion of the bogie. We show the results for Cooperrider's simple bogie model with flangeless wheels and with normal wheels and for the complex bogie model with normal wheels.

All the diagrams show subcritical bifurcation. Until the car reaches a certain critical speed V_t the zero-solution is the only solution, and it is asymptotically stable. It means, that any disturbance of the travelling bogie will decay exponentially. When the car moves with a speed larger than the critical speed V_t but slower than the speed V_c corresponding to the eigenvalue of the linearized problem, then there exist three solutions. One of them - the zero-solution - is still asymptotically stable, another - a time periodic oscillation with the larger amplitude - is also asymptotically stable, and the third solution - a time periodic oscillation with the smaller amplitude - is unstable. When the car moves with a speed larger than V_c then there exist only two solutions. They are the zero-solution, which is now unstable and the time periodic oscillation with large amplitude, which is still asymptotically stable.

The effect on the bogie will be the following. When the car moves with a speed less than V_t all disturbances will decay exponentially. When the car increases its speed above V_t it will continue to run straight forward unless the disturbances are larger than a certain value. If this is the case, the bogie will oscillate laterally with

an amplitude and frequency, which are uniquely determined by the speed of the car, and which follow the upper branch in the bifurcation diagram. If the disturbances are under the certain value mentioned before, the bogie will continue to run steadily with increasing speed of the car. The closer the speed comes to V_c the smaller will the magnitude of the disturbances be that is necessary "to push the motion up on the upper stable branch". Above the speed V_c the bogie will always exhibit lateral oscillations.

When the speed of the car decreases, the lateral oscillations will remain with amplitudes and frequencies following the upper stable branch, until the car reaches the speed V_t. Then the oscillations will stop abruptly. A hysteresis effect will occur within the velocity range $V_t < V < V_c$.

It is seen that the introduction of a wheel flange raises the critical velocity V_t. The resulting stabilization of the growth of the amplitude of the oscillation is only, what one would expect. However it is interesting to notice, that the frequency only changes little from the case of a wheel without flange to the case of a wheel with flange.

"The certain value" of the disturbance, mentioned earlier, is impossible to determine. It depends first of all on the modes of all the other degrees of freedom that are coupled to the lateral motion. Further investigation might however yield a sufficient condition for asymptotic stability of the zero solution, when $V_t < V < V_c$ and depending on the speed V. This domain in the diagram is called the domain of attraction of the zero solution. It becomes smaller when V approaches V_c. On the other hand it is certain, that the disturbance will start an oscillation at the speed V_1, $V_t < V_1 < V_c$, if the disturbance has a size larger than the ordinate of the point on the unstable branch corresponding to the speed V_1. The effect is illustrated by the case of an unstable limit cycle of a problem with one degree of freedom. We suppose, that zero within the limit cycle is a stable limit point, and that a stable limit cycle lies far out in the phase plane. In this case all disturbances smaller than the smallest distance s from zero to the unstable limit cycle will decay to zero. All disturbances larger than the largest distance ℓ from zero to the limit cycle will grow, until they reach the stable limit cycle. If one keeps in mind the weird shapes limit cycles can take, it is clear that a large circular domain with inner radius s and outer radius ℓ may exist, within which it will depend on the phase of the disturbance, whether it decays to zero or ends on the stable limit cycle.

In the bifurcation analysis the periodic motion is only determined modulo a phase, and therefore we have no way of determining whether a disturbance of a size between s and ℓ will grow or decay.

The calculations were executed on an IBM 4341.

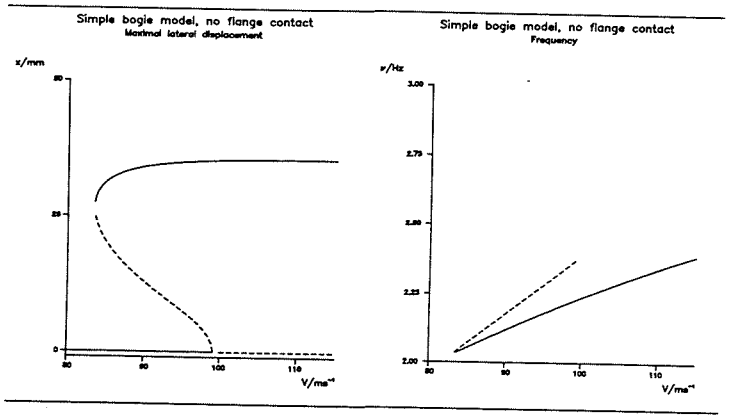


Figure 4.

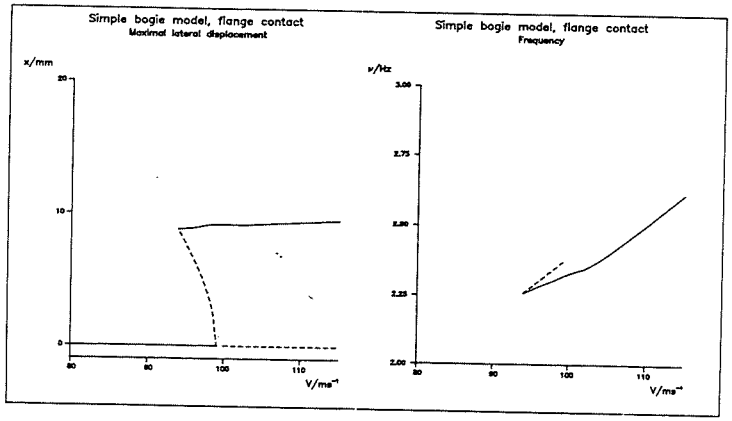


Figure 5.

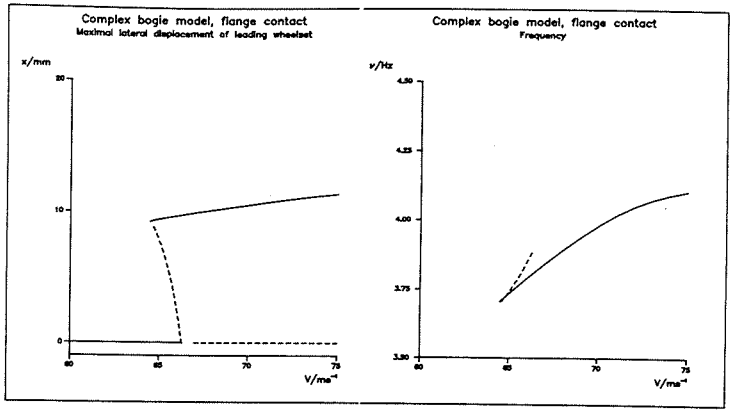


Figure 6.

ACKNOWLEDGEMENT

This work was supported by the Danish Council for Scientific and Industrial Research under Grant No. 95-110316.

REFERENCES

1. Cooperrider, N.K., The hunting behavior of conventional railway trucks, J. Eng. for Industry, 1971, pp. 1-10.
2. Hassard, B.D., Kazarinoff, N.D., Wan, Y-H., Theory and applications of Hopf bifurcation, Cambridge University Press, Cambridge, 1981.
3. Hopf, E., Abzweigung einer periodischen Lösung von einer stationären Lösung eines Differentialsystems, Ber. Verh. Sachs. Akad. Wiss. Leipzig Math.-Nat. 94 (1942), 3-22. Translation to English by L. Howard and N. Kopell in Marsden, J.E. and McCracken, M., The Hopf Bifurcation and Its Applications, Appl. Math. Sciences, Vol. 19, Springer-Verlag, N.Y. (1976) s. 163-205.

33

QUASI-PERIODIC OSCILLATIONS IN RAILWAY VEHICLES

Hans True and Christian Kaas-Petersen
Laboratory of Applied Mathematical Physics
The Technical University of Denmark
Bldg. 303, DK-2800 LYNGBY, Denmark

1. SUMMARY

We consider a mathematical model of a railway car moving on a straight, horizontal track. The dynamics of the model is described by 7 degrees of freedom leading to 14 ODE's of first order. Due to non-linear wheel-rail forces, the ODE's are non-linear. We determine the periodic solution bifurcating from the smallest bifurcation point of the trivial solution and follow it numerically as a function of the velocity. In this way we are able to determine the point where bifurcation to a quasi-periodic solution takes place. In our example the new quasi-periodic solution is non-stable near the bifurcation point.

2. THE MODEL

We consider a mathematical model of a railway car supported by two bogies. The car runs on a perfect, level and straight track. The carbody and all parts of the bogies are treated as stiff, and friction is only included in the non-linear wheel-rail forces. Relative movements in a coordinate system moving along a straight line with the velocity of the car are assumed small. Flanges on the wheels are absent in our model. We want to demonstrate that the horizontal oscillations - hunting - is not caused by the flanges. In the periodic case the flanges limit the amplitudes of oscillations, but the true cause of hunting is strongly connected with the non-linearity of the forces between the wheels and the rails.

The model of the bogies is taken from Cooperrider [1]. We have chosen his simple model with 2 degrees of freedom in order to simplify the calculations. The model still exhibits most of the features a bogie model should possess.

The wheels are assumed conical, so no gravitational stiffness is included.

Parameters in the description of wheel-rail geometry have the values:

$$a = 0.7163 \text{ m} \quad ; \quad \lambda = 0.05 \quad \text{og} \quad r_o = 0.4572 \text{ m}$$

where a is half the track gauge, λ the wheel conicity and r_o the centered wheel rolling radius.

We use the approximation to the creep force that was proposed by Vermeulen and Johnson [2], where creep-force saturation is taken into account (but spin-creep-forces have been excluded). Thus

$$F_x = \frac{\xi_x}{\phi}\frac{F_R}{\xi_R} \quad , \quad F_y = \frac{\xi_y}{\psi_1}\frac{F_R}{\xi_R} \quad ;$$

$$\frac{F_R}{\mu N} = \begin{cases} u - \frac{1}{3}u^2 + \frac{1}{27}u^3 & \text{for } u < 3 \\ 1 & \text{for } u \geq 3 \end{cases} \quad , \quad u = \frac{G\pi ab}{\mu N}\xi_R \; ;$$

$$\xi_R = \sqrt{\frac{\xi_x^2}{\phi^2} + \frac{\xi_y^2}{\psi_1^2}} \; .$$

where index x is lateral to the rail and index y is along the rail. The coefficient of friction is set to $\mu = 0.15$, ~~assuming that~~ and the wheels and rails are made of steel. The Hertz contact theory yields the values

$$\phi = 0.54219 \quad \psi_1 = 0.60252$$

$$G\pi ab = 6.563 \cdot 10^6 \text{N} \quad \mu N = 10\,000 \text{ N} \; .$$

a and b are the lengths of the major and minor axes of the contact ellipse and G the shear modulus.

We assume, that the total mass N_{tot} = 44496 kg, and the mass of the bogies, which is 4963 kg, are distributed equally on the eight wheels. Then the vertical force between wheel and rail is N = 6803 kg.

The model of the car and one bogie together with the nomenclature used are shown on Fig. 1.

The governing equations are listed below.

Introducing $x_1 = q_1$, $x_2 = \dot{q}_1$, $x_3 = q_2$, $\cdots$ we obtain 14 ODE's of first order. Since x_1, x_2, $\cdots$ describe the state of the system, we call R^{14} the state space.

$$
\begin{aligned}
&m_b \ddot{q}_1 + A_1 + 2(F_x^1 + F_x^2) = 0\\
&I_{by}\ddot{q}_2 + A_3 + 2b(F_x^1 - F_x^2) + 2a(F_y^1 + F_y^2) = 0\\
&m_b \ddot{q}_3 + A_2 + 2(F_x^3 + F_x^4) = 0\\
&I_{by}\ddot{q}_4 + A_4 + 2b(F_x^3 - F_x^4) + 2a(F_y^3 + F_y^4) = 0\\
&m_c \ddot{q}_5 - A_1 - A_2 = 0\\
&I_{cy}\ddot{q}_6 - d_6A_1 + d_6A_2 - A_3 - A_4 = 0\\
&I_{cr}\ddot{q}_7 - h_3A_1 - h_3A_2 + A_5 = 0
\end{aligned}
$$

where:

$$A_1 = 2k_4(q_1 - q_5 - d_6q_6 - h_3q_7) + 2D_2(\dot{q}_1 - \dot{q}_5 - d_6\dot{q}_6 - h_3\dot{q}_7)$$

$$A_2 = 2k_4(q_3 - q_5 + d_6q_6 - h_3q_7) + 2D_2(\dot{q}_3 - \dot{q}_5 + d_6\dot{q}_6 - h_3\dot{q}_7)$$

$$A_3 = k_6(q_2 - q_6)$$

$$A_4 = k_6(q_4 - q_6)$$

$$A_5 = 4d_2^2(k_5q_7 + D_1\dot{q}_7)$$

F_x^1, F_x^2, F_y^1, F_y^2, are creep forces for the front bogie

F_x^3, F_x^4, F_y^3, F_y^4, are creep forces for the rear bogie

We use the values $I_{cy} = 86 \cdot 10^4 \text{kgm}^2$ and $I_{cr} = 74 \cdot 10^3 \text{kgm}^2$. The other values are given in [1]. $d_6 = 7.62$ m, $h_3 = 0.61$ m, $d_2 = 0.68$m

3. RESULTS

We solve the problem numerically using a program for solution of systems of nonlinear ordinary differential equations. The program is started at the trivial solution at a certain velocity, and it "follows" the solution for growing parameter values, calculating for each step the eigenvalues of the linearized problem until a Hopf bifurcation takes place.

Starting now at the bifurcation point we "follow" the periodic solution

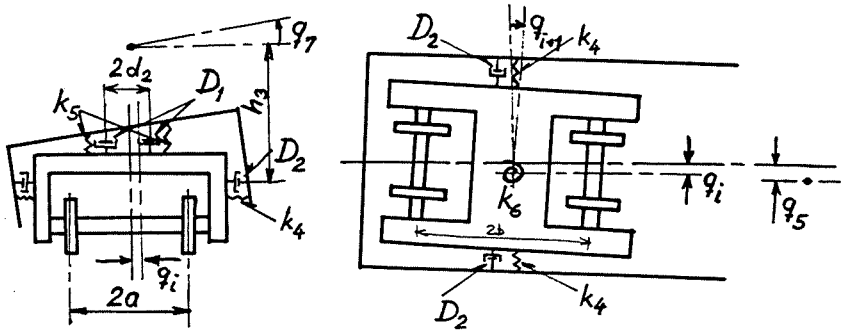
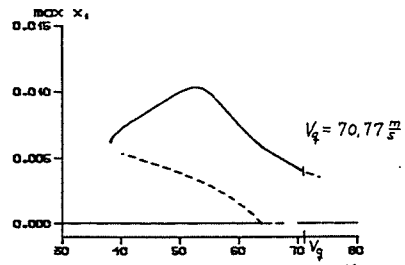


Figure 1. The car model, Front bogie: i = 1, Rear bogie: i = 3.

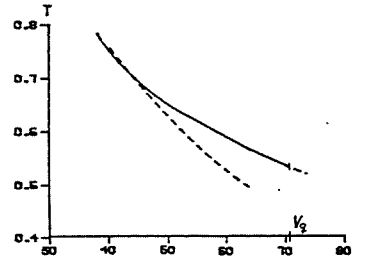
using the residual map Q := P − I, where P is the Poincaré map, defined on a cross section of the state space near the periodic orbit. The stability is determined by the eigenvalues of the Jacobian of P.

The program follows the path of periodic orbits around saddle-node bifurcations and determine the bifurcation into quasi-periodic solutions. The linearization is made by forward difference approximations.

The results for the car is shown on the diagrams below. The bifurcation to a non-stable quasi-periodic motion takes place at a value of the speed V_q very close to the second eigenvalue of the steady problem.



The maximum amplitude (m) as function of the speed (m/s)



The period (s) as function of the speed (m/s)

REFERENCES

1 Cooperrider, N.K., The hunting behavior of conventional railway trucks, J. Eng. for Industry, 1971, pp. 1-10.

2. Vermeulen, P.J. and Johnson, K.L., Contact of Nonspherical Elastic Bodies Transmitting Tangential Forces. Journal of Applied Mechanics, Trans ASME Vol. 31, June 1964, pp. 339-340.