A general centre manifold construction algorithm for the web, including isochrons of slow manifolds

A. J. Roberts*

May 28, 2017

Abstract

This code is the heart and muscle of a web service. The web service derives a centre manifold of any specified system of ordinary differential equations or delay differential equations, when the system has fast and centre modes. The centre modes may be slow, as in a pitchfork bifurcation, or oscillatory, as in a Hopf bifurcation, or some more complicated superposition. In the case when the fast modes all decay, the centre manifold supplies a faithful large time model of the dynamics. Further, this code now derives vectors defining the projection onto the centre manifold along the isochrons: this projection is needed for initial conditions, forcing, system modifications, and uncertainty quantification.

Contents

*School of Mathematical Sciences, University of Adelaide, South Australia 5005, Australia. http://www.maths.adelaide.edu.au/anthony.roberts/

1 Overall initialisation

In the following, assign `thecase:=myweb;` for the web service (or to read a system from file `cmsysb.red`), otherwise assign `thecase` to be any of the example dynamical systems in set `thecases`.

```
% see gcmafwFib.pdf for detailed explanation
% AJ Roberts, Nov 2013 -- May 2017
thecase:=myweb;
thecases:={onedde, anotherdde, twodde, dde2d, dde2d2ha,
dde2d2hb, simple2d, simple2ds, simple2dss, norm2dsimp, fourstater
```

```
another2d, another2ds, simple3d, simple3ds, geneigenvec,
bifurcate2d, simpleosc, perturbfreq, nonseparatedosc,
quasidelayosc, quasidelayoscmod, rosslerlike, doubleosc,
oscmeanflow, modulatedduffing, modulateoscillator,
StoleriuOne, StoleriuTwo, delayprolif, delayedprolif,
normalmodes, koopmanmodes, forcedvdp, lorenz86slow, lorenz86norm,
substablem }$
```

Define default parameters for the iteration: `maxiter_` is the maximum number of allowed iterations; `toosmall` is the order of errors in the analysis in terms of the parameter `small`. Specific problems may override these defaults.

```
maxiter_:=29$
factor small;
toosmall:=3$
```

For optional trace printing of test cases: comment out second line when not needed.

```
trace_:=0$
%trace_:=1; maxiter_:=9;
```

The `rationalize` switch makes code much faster with complex numbers. The switch `gcd` seems to wreck convergence, so leave it off.

```
on div; off allfac; on revpri;
on rationalize;
linelength 60$
```

Propose to use `e_` as basis vector for matrices and vectors. Declare it non-commutative so that multiplication does not commute.

```
operator e_;
noncom e_;
factor e_;
let { e_(~j,~k)*e_(~l,~m)=>0 when k neq l
    , e_(~j,~k)*e_(~l,~m)=>e_(j,m) when k=l
    , e_(~j,~k)^2=>0 when j neq k
```

```
    , e_(~j,j)^2=>e_(j,j) };
```

Also need a transpose operator: do complex conjugation explicitly when needed.

```
operator tpe_; linear tpe_;
let tpe_(e_(~i,~j),e_)=>e_(j,i);
```

Need to enter delayed factors in the ODEs, so use operators for the dependent variables in the dynamical system.

```
operator u;
```

Empty the output LaTeX file in case of error.

```
out "centreMan.tex";
write "This empty document indicates error.";
shut "centreMan.tex";
```

Automatically testing a set of examples does not yet work.

```
%foreach thecase in thecases do begin
```

2 Some example systems

Define the basic linear operator, centre manifold bases, and 'nonlinear' function. Note that Reduce's matrix transpose does not take complex conjugate. Then the web service inputs the system from a file, otherwise get the system from one of the examples that follow.

```
if thecase=myweb then begin
in "cmsysb.red"$
end;
```

2.1 Simple one variable delay differential equation

Model a delayed 'logistic' system in one variable with

$$\frac{du}{dt} = -(1+a)[1+u(t)]u(t-\pi/2),$$

for small parameter a. We code the parameter a as 'small', and observe it is consequently considered as 'small squared' because all nonlinear terms and already 'small' terms are multiplied by `small`. The marginal modes are $e^{\pm it}$ so nominate the frequencies ± 1. The eigenvectors are just $1 \cdot e^{\pm it}$. Because for delay differential equations the time dependence $e^{\pm i\omega t}$ is an integral part of the definition of the eigenvector; hence the coded eigenvectors can be the same, as here, because they are differentiated through the time dependence $e^{\pm i\omega t}$.

```
if thecase=onedde then begin
ff_:=tp mat((-(1+small*a)*(1+u1)*u1(pi/2)));
evalm_:=mat((i,-i));
ee_:=tp mat((1),(1));
zz_:=tp mat((1),(1));
toosmall:=3;
factor s,a,exp;
end;
```

The code works for orders higher than cubic, but is slow: takes about a minute per iteration.

The centre manifold

$$u_1 = e^{-2ti} s_2^2 \varepsilon \big(1/5i + 2/5\big) + e^{-ti} s_2 + e^{2ti} s_1^2 \varepsilon \big(- 1/5i + 2/5\big) + e^{ti} s_1$$

Centre manifold ODEs

$$\dot s_1 = s_2 s_1^2 \varepsilon^2 \big(-2/5i\pi - 12/5i - 6/5\pi + 4/5\big)/\big(\pi^2+4\big) + s_1 a \varepsilon^2 \big(4i + 2\pi\big)/\big(\pi^2+4\big)$$

$$\dot s_2 = s_2^2 s_1 \varepsilon^2 \big(2/5i\pi + 12/5i - 6/5\pi + 4/5\big)/\big(\pi^2+4\big) + s_2 a \varepsilon^2 \big(-4i + 2\pi\big)/\big(\pi^2+4\big)$$

Observe that the real parts of these ODEs indicate linear growth for positive parameter a, limited by nonlinear saturation. A classic Hopf bifurcation (although I have not recorded here evidence for the attractiveness).

2.2 Another one variable delay differential equation

Model a delayed 'logistic' system in one variable with

$$\frac{du}{dt} = -u(t) - (\sqrt{2} + a)u(t - 3\pi/4) + \mu u(t - 3\pi/4)^2 + \nu u(t - 3\pi/4)^3,$$

for small parameter a and nonlinearity parameters μ and ν. Numerical computation of the spectrum indicates that the system has a Hopf bifurcation as parameter a crosses zero.[1]

```
ac=-sqrt(2), tau=3*pi/4
ce=@(z) z+1-ac*exp(-tau*z)
lams=fsolve(ce,randn(100,2)*[2;2*i])
plot(real(lams),imag(lams),'o')
```

We code the parameter a as 'small', and observe it is consequently considered as 'small squared' because all nonlinear terms and already 'small' terms are multiplied by `small`. The marginal modes are $e^{\pm it}$ so nominate the frequencies ± 1. The eigenvectors are just $1 \cdot e^{\pm it}$. Because for delay differential equations the time dependence $e^{\pm i\omega t}$ is an integral part of the definition of the eigenvector; hence the coded eigenvectors can be the same, as here, because they are differentiated through the time dependence $e^{\pm i\omega t}$.

```
if thecase=anotherdde then begin
ff_:=tp mat((-u1-(sqrt(2)+small*a)*u1(3*pi/4)
    +mu*u1(3*pi/4)^2 +small*nu*u1(3*pi/4)^3));
evalm_:=mat((i,-i));
ee_:=tp mat((1),(1));
zz_:=tp mat((1),(1));
toosmall:=3;
```

[1] Replacing $-(\sqrt{2} + a)$ with $+(1 + a)$ leads to a pitchfork bifurcation with broken symmetry when $\mu \neq 0$.

```
factor s,a,mu,nu,cis;
end;
```

The modelling predicts a supercritical Hopf bifurcation as parameter a increases through zero, although if nonlinearity parameter ν is large enough negative, then the bifurcation will be subcritical.

2.3 Separated delay differential equations

Now consider the system

$$\dot{x} = -[1 + a - y(t)]x(t - \pi/2) \quad \text{and} \quad \dot{y} = -y + x^2.$$

Without the 'fast' variable y the x-ODE would be at marginal criticality when parameter $a = 0$. With the coupling, any oscillations in x should drive a positive y which then helps stabilise the oscillations. Let's see this in analysis.

Code the system as follows with small parameter a. Because the system is linearly separated, the eigenvectors are simple: the eigenvectors of the marginal modes are $(1, 0)e^{\pm it}$, as are the adjoint's eigenvectors.

```
if thecase=twodde then begin
ff_:=tp mat((
    -(1+small*a-u2)*u1(pi/2),
    -u2+u1^2
    ));
evalm_:=mat((i,-i));
ee_:=tp mat((1,0),(1,0));
zz_:=tp mat((1,0),(1,0));
toosmall:=3;
factor s,a,exp;
end;
```

The centre manifold

$$u_1 = e^{-ti}s_2 + e^{ti}s_1$$

$$u_2 = e^{-2ti}s_2^2\varepsilon\big(2/5i + 1/5\big) + e^{2ti}s_1^2\varepsilon\big(-2/5i + 1/5\big) + 2s_2s_1\varepsilon$$

Centre manifold ODEs

$\dot s_1 = s_2 s_1^2 \varepsilon^2 \big(-4/5i\pi - 36/5i - 18/5\pi + 8/5\big)/\big(\pi^2+4\big) + s_1 a \varepsilon^2 \big(4i + 2\pi\big)/\big(\pi^2+4\big)$

$\dot s_2 = s_2^2 s_1 \varepsilon^2 \big(4/5i\pi + 36/5i - 18/5\pi + 8/5\big)/\big(\pi^2+4\big) + s_2 a \varepsilon^2 \big(-4i + 2\pi\big)/\big(\pi^2+4\big)$

2.4 Linearly coupled 2D DDE

Here we explore a system where the centre modes involve both variables. Consider the system

$$\dot u_1 = u_2(t-\pi/2) - u_1^2 \quad \text{and} \quad \dot u_2 = u_1(t-\pi/2) + u_2^2.$$

We find the quadratic reaction does not stabilise oscillating growth.

Numerical solution of the characteristic equation indicate that there is one unstable mode, $\lambda = 0.4745$, two centre modes, $\lambda = \pm i$, and all the rest are stable modes with the gravest having eigenvalue $\lambda = -0.6846 \pm i2.8499$. The analysis gives the centre modes are nonlinearly unstable: $\dot a \approx (0.6758 \pm i1.8616)|a|^2 a$. The following Matlab/Octave code finds eigenvalues.

```
ce=@(z) z.^2-exp(-pi*z)
lams=fsolve(ce,randn(100,2)*[2;10*i])
plot(real(lams),imag(lams),'o')
```

Interestingly, the centre eigenvectors are $(1,-1)e^{\pm it}$ so that u_2 is in opposite phase to u_1. The adjoint's eigenvectors are the same.

```
if thecase=dde2d then begin
ff_:=tp mat((+u2(pi/2)-u1^2,+u1(pi/2)+u2^2));
evalm_:=mat((i,-i));
ee_:=tp mat((1,-1),(1,-1));
zz_:=tp mat((1,-1),(1,-1));
toosmall:=3; factor s,small;
end;
```

The centre manifold

$u_1 = s_2^2\varepsilon\big(-2/5\,e^{-2ti}i+1/5\,e^{-2ti}\big)-2s_2s_1\varepsilon+s_2\,e^{-ti}+s_1^2\varepsilon\big(2/5\,e^{2ti}i+1/5\,e^{2ti}\big)+$
$s_1\,e^{ti}$

$u_2 = s_2^2\varepsilon\big(2/5\,e^{-2ti}i-1/5\,e^{-2ti}\big)+2s_2s_1\varepsilon-s_2\,e^{-ti}+s_1^2\varepsilon\big(-2/5\,e^{2ti}i-1/5\,e^{2ti}\big)-$
$s_1\,e^{ti}$

Centre manifold ODEs

$\dot s_1 = s_2s_1^2\varepsilon^2\big(-36/5i\pi-16/5i-8/5\pi+72/5\big)/\big(\pi^2+4\big)$

$\dot s_2 = s_2^2s_1\varepsilon^2\big(36/5i\pi+16/5i-8/5\pi+72/5\big)/\big(\pi^2+4\big)$

This model predicts nonlinear growth of the centre modes, in addition to the growth of the unstable mode.

2.5 Double Hopf 2D DDE

Erneux (2009) [§7.2] explored an example of a laser subject to optoelectronic feedback. For certain parameter values it has a two frequency Hopf bifurcation.

Erneux (2009) [eq. (7.42)] transformed the laser system to the non-dimensional

$$(1+\eta)\frac{d^2\log[1+y]}{dt^2} = -\theta^2\left[y(t)+\eta y(t-\pi)\right],$$

for parameters η and θ. Erneux (2009) identified double Hopf bifurcations from the origin at parameters (η,θ) of $(3/5,2)$, $(7/25,4)$, $(-5/13,2)$ and $(-9/41,4)$, among others. Here we work with a system of first order, DDEs, so transform the DDE to

$$\begin{aligned}\dot x &= -\theta^2\left[y(t)+\eta y(t-\pi)\right]/(1+\eta),\\ \dot y &= [1+y(t)]x(t).\end{aligned}$$

The following Octave/Matlab code plots the spectrum for the equilibrium at the origin. The results indicate that in all four cases mentioned the centre manifold is attractive. The gravest eigenvalue being, respectively, $-0.69\pm i3.87$, $-0.38\pm i1.02$, -0.31 and $-0.41\pm i2.03$.

```
eta=3/5, theta=2
ce=@(z) (1+eta)*z.^2+theta^2*(1+eta*exp(-pi*z))
lams=fsolve(ce,randn(100,2)*[2;10*i])
plot(real(lams),imag(lams),'o')
```

Ensure you interpret 'left-eigenvectors' as the eigenvectors of the adjoint operator (the complex conjugate transpose of the operator).

2.5.1 Parameters $(\eta, \theta) = (3/5, 2)$

I invoke a slightly different perturbation of the parameter η to that of Erneux (2009). The eigenvectors are $(1, \mp i/\omega)e^{\pm i\omega t}$ for frequencies $\omega = 1, 2$, while the eigenvectors of the adjoint are $(1, \mp i\omega)e^{\pm i\omega t}$.

```
if thecase=dde2d2ha then begin
eta:=3/5;
theta:=2*(1+small*delta);
ff_:=tp mat((
    -theta^2*((1/(1+eta)-small*nu)*u2
            +(eta/(1+eta)+small*nu)*u2(pi)),
    +u1*(1+u2)
    ));
evalm_:=mat((i,2*i,-i,-2*i));
ee_:=tp mat((1,-i),(1,-i/2),(1,+i),(1,+i/2));
zz_:=tp mat((1,-i),(1,-2*i),(1,+i),(1,+2*i));
toosmall:=3;
factor s,delta,nu,cis;
end;
```

The centre manifold is rather complicated.

$u_1 = 1/6\, e^{-4ti} s_4^2 \varepsilon i + 3/16\, e^{-3ti} s_4 s_2 \varepsilon i + e^{-2ti} s_4 + e^{-2ti} s_2^2 \varepsilon\big(-9/2 i\pi^2 - 16i - 6\pi\big)/\big(9\pi^2 + 64\big) + e^{-ti} s_4 s_1 \varepsilon\big(9/4 i\pi^2 + 2i - 3/2\pi\big)/\big(9\pi^2 + 16\big) + e^{-ti} s_2 - 1/6\, e^{4ti} s_3^2 \varepsilon i - 3/16\, e^{3ti} s_3 s_1 \varepsilon i + e^{2ti} s_3 + e^{2ti} s_1^2 \varepsilon\big(9/2 i\pi^2 + 16i - 6\pi\big)/\big(9\pi^2 + 64\big) + e^{ti} s_3 s_2 \varepsilon\big(-9/4 i\pi^2 - 2i - 3/2\pi\big)/\big(9\pi^2 + 16\big) + e^{ti} s_1$

$u_2 = -1/6\,e^{-4ti}s_4^2\varepsilon - 9/16\,e^{-3ti}s_4s_2\varepsilon + 1/2\,e^{-2ti}s_4i + e^{-2ti}s_2^2\varepsilon\big(3i\pi - 9/4\pi^2 - 8\big)/\big(9\pi^2+64\big) + e^{-ti}s_4s_1\varepsilon\big(3/2i\pi+9/4\pi^2+2\big)/\big(9\pi^2+16\big) + e^{-ti}s_2i - 1/6\,e^{4ti}s_3^2\varepsilon - 9/16\,e^{3ti}s_3s_1\varepsilon - 1/2\,e^{2ti}s_3i + e^{2ti}s_1^2\varepsilon\big(-3i\pi-9/4\pi^2-8\big)/\big(9\pi^2+64\big) + e^{ti}s_3s_2\varepsilon\big(-3/2i\pi + 9/4\pi^2 + 2\big)/\big(9\pi^2 + 16\big) - e^{ti}s_1i$

Centre manifold ODEs describe complicated interactions, but mainly it is the coefficients that are complicated functions of π.

$\dot s_1 = s_4s_3s_1\varepsilon^2\big(-9963/4i\pi^6 - 38340i\pi^4 - 167424i\pi^2 - 147456i + 21141/16\pi^7 + 20007\pi^5 + 84096\pi^3 + 61440\pi\big)/\big(6561\pi^8 + 116640\pi^6 + 684288\pi^4 + 1474560\pi^2 + 1048576\big) + s_3s_2\varepsilon\big(-3i\pi - 4\big)/\big(9\pi^2 + 16\big) + s_2s_1^2\varepsilon^2\big(-2916i\pi^6 - 17280i\pi^4 - 3072i\pi^2 - 196608i - 8019/2\pi^7 - 44064\pi^5 - 93312\pi^3 + 122880\pi\big)/\big(6561\pi^8 + 116640\pi^6 + 684288\pi^4 + 1474560\pi^2 + 1048576\big) + s_1\delta\varepsilon^2\big(16i - 12\pi\big)/\big(9\pi^2 + 16\big) + s_1\nu\varepsilon^2\big(-64i + 48\pi\big)/\big(9\pi^2 + 16\big)$

$\dot s_2 = s_4s_3s_2\varepsilon^2\big(9963/4i\pi^6 + 38340i\pi^4 + 167424i\pi^2 + 147456i + 21141/16\pi^7 + 20007\pi^5 + 84096\pi^3 + 61440\pi\big)/\big(6561\pi^8 + 116640\pi^6 + 684288\pi^4 + 1474560\pi^2 + 1048576\big) + s_4s_1\varepsilon\big(3i\pi - 4\big)/\big(9\pi^2 + 16\big) + s_2^2s_1\varepsilon^2\big(2916i\pi^6 + 17280i\pi^4 + 3072i\pi^2 + 196608i - 8019/2\pi^7 - 44064\pi^5 - 93312\pi^3 + 122880\pi\big)/\big(6561\pi^8 + 116640\pi^6 + 684288\pi^4 + 1474560\pi^2 + 1048576\big) + s_2\delta\varepsilon^2\big(-16i - 12\pi\big)/\big(9\pi^2 + 16\big) + s_2\nu\varepsilon^2\big(64i + 48\pi\big)/\big(9\pi^2 + 16\big)$

$\dot s_3 = s_4s_3^2\varepsilon^2\big(-16/3i - 2\pi\big)/\big(9\pi^2 + 64\big) + s_3s_2s_1\varepsilon^2\big(-34992i\pi^6 - 252288i\pi^4 - 559104i\pi^2 - 393216i - 10206\pi^7 - 64800\pi^5 - 138240\pi^3 - 98304\pi\big)/\big(6561\pi^8 + 116640\pi^6 + 684288\pi^4 + 1474560\pi^2 + 1048576\big) + s_3\delta\varepsilon^2\big(128i + 48\pi\big)/\big(9\pi^2 + 64\big) + s_1^2\varepsilon\big(-24i\pi + 64\big)/\big(9\pi^2 + 64\big)$

$\dot s_4 = s_4^2s_3\varepsilon^2\big(16/3i - 2\pi\big)/\big(9\pi^2 + 64\big) + s_4s_2s_1\varepsilon^2\big(34992i\pi^6 + 252288i\pi^4 + 559104i\pi^2 + 393216i - 10206\pi^7 - 64800\pi^5 - 138240\pi^3 - 98304\pi\big)/\big(6561\pi^8 + 116640\pi^6 + 684288\pi^4 + 1474560\pi^2 + 1048576\big) + s_4\delta\varepsilon^2\big(-128i + 48\pi\big)/\big(9\pi^2 + 64\big) + s_2^2\varepsilon\big(24i\pi + 64\big)/\big(9\pi^2 + 64\big)$

2.5.2 Parameters $(\eta,\theta) = (7/25, 4)$

The eigenvectors are $(1, \mp i/\omega)e^{\pm i\omega t}$ for frequencies $\omega = 3, 4$, while the eigenvectors of the adjoint are $(1, \mp i\omega)e^{\pm i\omega t}$.

```
if thecase=dde2d2hb then begin
eta:=7/25;
theta:=4*(1+small*delta);
ff_:=tp mat((
    -theta^2*((1/(1+eta)-small*nu)*u2
           +(eta/(1+eta)+small*nu)*u2(pi)),
    +u1*(1+u2)
    ));
evalm_:=mat((3*i,-3*i,4*i,-4*i));
ee_:=tp mat((1,-i/3),(1,+i/3),(1,-i/4),(1,+i/4));
zz_:=tp mat((1,-3*i),(1,+3*i),(1,-4*i),(1,+4*i));
toosmall:=3;
factor s,delta,nu,cis;
end;
```

The centre manifold

$u_1 = 1/12\, e^{-8ti} s_4^2 \varepsilon i + 21/160\, e^{-7ti} s_4 s_2 \varepsilon i + 4/15\, e^{-6ti} s_2^2 \varepsilon i + e^{-4ti} s_4 + e^{-3ti} s_2 + 3/32\, e^{-ti} s_4 s_1 \varepsilon i - 1/12\, e^{8ti} s_3^2 \varepsilon i - 21/160\, e^{7ti} s_3 s_1 \varepsilon i - 4/15\, e^{6ti} s_1^2 \varepsilon i + e^{4ti} s_3 + e^{3ti} s_1 - 3/32\, e^{ti} s_3 s_2 \varepsilon i$

$u_2 = -1/24\, e^{-8ti} s_4^2 \varepsilon - 49/480\, e^{-7ti} s_4 s_2 \varepsilon - 1/10\, e^{-6ti} s_2^2 \varepsilon + 1/4\, e^{-4ti} s_4 i + 1/3\, e^{-3ti} s_2 i - 1/96\, e^{-ti} s_4 s_1 \varepsilon - 1/24\, e^{8ti} s_3^2 \varepsilon - 49/480\, e^{7ti} s_3 s_1 \varepsilon - 1/10\, e^{6ti} s_1^2 \varepsilon - 1/4\, e^{4ti} s_3 i - 1/3\, e^{3ti} s_1 i - 1/96\, e^{ti} s_3 s_2 \varepsilon$

Centre manifold ODEs

$\dot s_1 = s_4 s_3 s_1 \varepsilon^2 \big(- 243/20 i + 567/80 \pi\big)/\big(49\pi^2 + 144\big) + s_2 s_1^2 \varepsilon^2 \big(- 12/5 i + 7/5 \pi\big)/\big(49\pi^2 + 144\big) + s_1 \delta \varepsilon^2 \big(432 i - 252\pi\big)/\big(49\pi^2 + 144\big) + s_1 \nu \varepsilon^2 \big(- 768 i + 448\pi\big)/\big(49\pi^2 + 144\big)$

$\dot s_2 = s_4 s_3 s_2 \varepsilon^2 \big(243/20 i + 567/80 \pi\big)/\big(49\pi^2 + 144\big) + s_2^2 s_1 \varepsilon^2 \big(12/5 i + 7/5 \pi\big)/\big(49\pi^2 + 144\big) + s_2 \delta \varepsilon^2 \big(-432 i - 252\pi\big)/\big(49\pi^2 + 144\big) + s_2 \nu \varepsilon^2 \big(768 i + 448\pi\big)/\big(49\pi^2 + 144\big)$

$\dot s_3 = s_4 s_3^2 \varepsilon^2 \big(-32/3 i - 14/3 \pi\big)/\big(49\pi^2 + 256\big) + s_3 s_2 s_1 \varepsilon^2 \big(-256/5 i - 112/5 \pi\big)/\big(49\pi^2 + 256\big) + s_3 \delta \varepsilon^2 \big(1024 i + 448\pi\big)/\big(49\pi^2 + 256\big)$

$\dot s_4 = s_4^2 s_3 \varepsilon^2 \big(32/3i - 14/3\pi\big)/\big(49\pi^2 + 256\big) + s_4 s_2 s_1 \varepsilon^2 \big(256/5i - 112/5\pi\big)/\big(49\pi^2 + 256\big) + s_4 \delta \varepsilon^2 \big(- 1024i + 448\pi\big)/\big(49\pi^2 + 256\big)$

The interaction appears a lot simpler in this case. Presumably simpler because the frequencies are 'more irrational'.

2.6 Simple 2D ODE

Consider the system $\dot u_1 = -\varepsilon u_1^2 + u_2 - u_1$ and $\dot u_2 = \varepsilon u_2^2 - u_2 + u_1$

```
if thecase=simple2d then begin
ff_:=tp mat((-u1+u2-u1^2,u1-u2+u2^2));
evalm_:=mat((0));
ee_:=tp mat((1,1));
zz_:=tp mat((1,1));
toosmall:=5;
end;
```

The centre manifold $u_1 = 3/8\varepsilon^3 s_1^4 - 1/2\varepsilon s_1^2 + s_1$

$u_2 = -3/8\varepsilon^3 s_1^4 + 1/2\varepsilon s_1^2 + s_1$

Centre manifold ODEs $\dot s_1 = -3/4\varepsilon^4 s_1^5 + \varepsilon^2 s_1^3$

Normals to isochrons at the slow manifold Use these vectors: to project initial conditions onto the slow manifold; to project non-autonomous forcing onto the slow evolution; to predict the consequences of modifying the original system; in uncertainty quantification to quantify effects on the model of uncertainties in the original system.

$z_{11} = 3/2\varepsilon^4 s_1^4 + 3/4\varepsilon^3 s_1^3 - 1/2\varepsilon^2 s_1^2 - 1/2\varepsilon s_1 + 1/2$

$z_{12} = 3/2\varepsilon^4 s_1^4 - 3/4\varepsilon^3 s_1^3 - 1/2\varepsilon^2 s_1^2 + 1/2\varepsilon s_1 + 1/2$

2.6.1 The stable manifold

Appears to get sensible answers even for the stable manifold! Just invoke this case to characterise the linear stable subspace.

```
if thecase=simple2ds then begin
ff_:=tp mat((-u1+u2-u1^2,u1-u2+u2^2));
evalm_:=mat((-2));
ee_:=tp mat((1,-1));
zz_:=tp mat((1,-1));
toosmall:=5;
end;
```

The stable manifold where the double factor of i in the exponentials give decaying modes of $e^{-2t}, e^{-6t}, e^{-8t}$.

$$u_1 = 1/8\varepsilon^3 \, e^{8iti} s_1^4 + 1/4\varepsilon^2 \, e^{6iti} s_1^3 + 1/2\varepsilon \, e^{4iti} s_1^2 + \, e^{2iti} s_1$$

$$u_2 = -1/8\varepsilon^3 \, e^{8iti} s_1^4 - 1/4\varepsilon^2 \, e^{6iti} s_1^3 - 1/2\varepsilon \, e^{4iti} s_1^2 - \, e^{2iti} s_1$$

Stable manifold ODEs is the trivial $\dot s_1 = 0$

2.6.2 The slow-stable manifold

Appears to get sensible answers even for the slow-stable manifold!! Which in this system is a coordinate transform that nonlinearly separates the dynamics. Amazing.

```
if thecase=simple2dss then begin
ff_:=tp mat((-u1+u2-u1^2,u1-u2+u2^2));
evalm_:=mat((0,-2));
ee_:=tp mat((1,1),(1,-1));
zz_:=tp mat((1,1),(1,-1));
toosmall:=3;
end;
```

The invariant manifold These give the location of the invariant manifold in terms of parameters s_j.

$$u_1 = \varepsilon^3\big(1/8\,e^{-8t}s_2^4 + 1/2\,e^{-6t}s_2^3s_1 + 1/2\,e^{-4t}s_2^2s_1^2 - 1/2\,e^{-2t}s_2s_1^3 + 3/8s_1^4\big) + \varepsilon^2\big(1/4\,e^{-6t}s_2^3 + 3/4\,e^{-4t}s_2^2s_1\big) + \varepsilon\big(1/2\,e^{-4t}s_2^2 + e^{-2t}s_2s_1 - 1/2s_1^2\big) + e^{-2t}s_2 + s_1$$

$$u_2 = \varepsilon^3\big(-1/8\,e^{-8t}s_2^4 + 1/2\,e^{-6t}s_2^3s_1 - 1/2\,e^{-4t}s_2^2s_1^2 - 1/2\,e^{-2t}s_2s_1^3 - 3/8s_1^4\big) + \varepsilon^2\big(-1/4\,e^{-6t}s_2^3 + 3/4\,e^{-4t}s_2^2s_1\big) + \varepsilon\big(-1/2\,e^{-4t}s_2^2 + e^{-2t}s_2s_1 + 1/2s_1^2\big) - e^{-2t}s_2 + s_1$$

invariant manifold ODEs The system evolves on the invariant manifold such that the parameters evolve according to these ODEs.

$$\dot s_1 = -3/4\varepsilon^4s_1^5 + \varepsilon^2s_1^3$$

$$\dot s_2 = 1/4\varepsilon^4s_2s_1^4 - \varepsilon^2s_2s_1^2$$

2.7 Normal form for 2D system with exact slow manifold

```
if thecase=norm2dsimp then begin
ff_:=tp mat((-u1+u2^2-2*u1^2,-u1*u2));
evalm_:=mat((-1,0));
ee_:=tp mat((1,0),(0,1));
zz_:=tp mat((1,0),(0,1));
toosmall:=3;
end;
```

```
The evolution of the real/complex amplitudes
                        2     2               4     4
ds(1)/dt =  - 2*small *s(2) *s(1) - 4*small *s(2) *s(1)
                      2     3
ds(2)/dt =  - small *s(2)
Finished constructing Slow-stable manifold of ODE/DDE
3: sub(s(1)=>s(1)*exp(t),{df(uu_,e_(1,1)),df(uu_,e_(2,1))});
                        2        2            2     3
{s(1) + small*(2*s(1)  + s(2) ) + 4*small *s(1)
```

```
        3        4         2     2
 + small *(8*s(1)  - 4*s(2) *s(1) )
        4         5          2     3
 + small *(16*s(1)  - 16*s(2) *s(1) ),
                          3       2          2
s(2) + small*s(2)*s(1) + ---*small *s(2)*s(1)
                          2
        3   5           3         3
 + small *(---*s(2)*s(1)  - 2*s(2) *s(1))
            2
        4   35           4         3     2
 + small *(----*s(2)*s(1)  - 6*s(2) *s(1) )}
            8
```

2.8 Four state Markov chain

Variable ε characterise the perturbation.

$$\dot u_1 = -\varepsilon u_1 + u_2$$

$$\dot u_2 = \varepsilon\big(u_3 - u_2 + u_1\big) - u_2$$

$$\dot u_3 = \varepsilon\big(u_4 - u_3 + u_2\big) - u_3$$

$$\dot u_4 = -\varepsilon u_4 + u_3$$

The linear perturbation terms gets multiplied by `small` again, but I do not see how to avoid that without wrecking other desirable things: such as, it is useful to multiply some nonlinear terms by small to show they are of higher order than other nonlinear terms.

```
if thecase=fourstatemarkov then begin
factor epsilon;
ff_:=tp mat((u2,-u2,-u3,u3))
+small*tp mat((-u1,+u1-u2+u3,+u2-u3+u4,-u4));
evalm_:=mat((0,0));
ee_:=tp mat((0,0,0,1),(1,0,0,0));
```

```
zz_:=tp mat((0,0,1,1),(1,1,0,0));
toosmall:=7;
end;
```

The centre manifold $u_1 = \varepsilon^2\big(2s_2 - s_1\big) - \varepsilon s_2 + s_2$

$u_2 = \varepsilon^2\big(- 2s_2 + s_1\big) + \varepsilon s_2$

$u_3 = \varepsilon^2\big(s_2 - 2s_1\big) + \varepsilon s_1$

$u_4 = \varepsilon^2\big(- s_2 + 2s_1\big) - \varepsilon s_1 + s_1$

Centre manifold ODEs $\dot s_1 = \varepsilon^3\big(- 3s_2 + 3s_1\big) + \varepsilon^2\big(s_2 - s_1\big)$

$\dot s_2 = \varepsilon^3\big(3s_2 - 3s_1\big) + \varepsilon^2\big(- s_2 + s_1\big)$

Normals to isochrons at the slow manifold

$z_{11} = 6\varepsilon^6 - \varepsilon^4$

$z_{12} = 19\varepsilon^6 - 4\varepsilon^4 + \varepsilon^2$

$z_{13} = -19\varepsilon^6 + 4\varepsilon^4 - \varepsilon^2 + 1$

$z_{14} = -6\varepsilon^6 + \varepsilon^4 + 1$

$z_{21} = -6\varepsilon^6 + \varepsilon^4 + 1$

$z_{22} = -19\varepsilon^6 + 4\varepsilon^4 - \varepsilon^2 + 1$

$z_{23} = 19\varepsilon^6 - 4\varepsilon^4 + \varepsilon^2$

$z_{24} = 6\varepsilon^6 - \varepsilon^4$

2.9 Bifurcating 2D system

This example tests labelling a small parameter and having a cubic term labelled as smaller than a quadratic term.

$\dot u_1 = -\varepsilon^2 u_2 u_1^2 - u_2 - 1/2u_1$

$\dot u_2 = \varepsilon\big(-u_2^2 + u_2\epsilon\big) - 2u_2 - u_1$

```
if thecase=another2d then begin
ff_:=tp mat((
    -u1/2-u2-small*u1^2*u2,
    -u1-2*u2+small*epsilon*u2-u2^2
  ));
evalm_:=mat((0));
ee_:=tp mat((1,-1/2));
zz_:=tp mat((1,-1/2));
end;
```

The centre manifold $u_1 = \varepsilon\big(-1/25s_1^2 - 2/25s_1\epsilon\big) + s_1$

$u_2 = \varepsilon\big(-2/25s_1^2 - 4/25s_1\epsilon\big) - 1/2s_1$

Centre manifold ODEs $\dot s_1 = \varepsilon^2\big(54/125s_1^3 + 12/125s_1^2\epsilon + 8/125s_1\epsilon^2\big) + \varepsilon\big(1/10s_1^2 + 1/5s_1\epsilon\big)$

Normals to isochrons at the slow manifold

$z_{11} = \varepsilon^2\big(-352/3125s_1^2 - 8/125\epsilon\big) - 8/125\varepsilon s_1 + 4/5$

$z_{12} = \varepsilon^2\big(-544/3125s_1^2 - 16/125\epsilon\big) - 16/125\varepsilon s_1 - 2/5$

2.9.1 The stable manifold

Appears to also get the stable manifold.

```
if thecase=another2ds then begin
ff_:=tp mat((
    -u1/2-u2-small*u1^2*u2,
    -u1-2*u2+small*epsilon*u2-u2^2
  ));
```

```
evalm_:=mat((-5/2));
ee_:=tp mat((1,2));
zz_:=tp mat((1,2));
toosmall:=7;
end;
```

The stable manifold ignoring the as yet awful formatting of the exponential,

$$u_1=\varepsilon^2\big(838/1875\,e^{\big(15iti}/2\big)s_1^3+8/25\,e^{\big(5iti}/2\big)s_1\epsilon\big)+8/25\varepsilon\,e^{5iti}s_1^2+e^{\big(5iti}/2\big)s_1$$

$$u_2=\varepsilon^2\big(2116/1875\,e^{\big(15iti}/2\big)s_1^3-4/25\,e^{\big(5iti}/2\big)s_1\epsilon\big)+36/25\varepsilon\,e^{5iti}s_1^2+2\,e^{\big(5iti}/2\big)s_1$$

Stable manifold ODEs shows the change in rate due to parameter variation: $\dot s_1=4/5\varepsilon^2 s_1\epsilon$

2.9.2 The slow-stable manifold

Appears to also get the slow-stable manifold, namely a normal form coordinate transform of the 2D state space.

```
if thecase=another2dss then begin
ff_:=tp mat((
    -u1/2-u2-small*u1^2*u2,
    -u1-2*u2+small*epsilon*u2-u2^2
  ));
evalm_:=mat((0,-5/2));
ee_:=tp mat((1,-1/2),(1,2));
zz_:=tp mat((1,-1/2),(1,2));
toosmall:=5;
end;
```

2.10 Simple 3D system

This example is straightforward.

$$\dot u_1 = \varepsilon u_3 u_2 + 2u_3 + u_2 + 2u_1$$

$$\dot u_2 = -\varepsilon u_3 u_1 + u_3 - u_2 + u_1$$

$$\dot u_3 = -\varepsilon u_2 u_1 - 3u_3 - u_2 - 3u_1$$

```
if thecase=simple3d then begin
ff_:=tp mat((2*u1+u2+2*u3+u2*u3
  ,u1-u2+u3-u1*u3
  ,-3*u1-u2-3*u3-u1*u2));
evalm_:=mat((0));
ee_:=tp mat((1,0,-1));
zz_:=tp mat((4,1,3));
end;
```

The centre manifold $u_1 = -\varepsilon s_1^2 + s_1$

$$u_2 = \varepsilon s_1^2$$

$$u_3 = \varepsilon s_1^2 - s_1$$

Centre manifold ODEs $\dot s_1 = -9\varepsilon^2 s_1^3 + \varepsilon s_1^2$

Normals to isochrons at the slow manifold

$$z_{11} = 258\varepsilon^2 s_1^2 - 16\varepsilon s_1 + 4$$

$$z_{12} = 93\varepsilon^2 s_1^2 - 9\varepsilon s_1 + 1$$

$$z_{13} = 240\varepsilon^2 s_1^2 - 16\varepsilon s_1 + 3$$

2.10.1 Its 2D stable manifold with generalised eigenvectors

Despite the generalised eigenvectors, the following alternative appears to generate the stable manifold if you wish:

```
if thecase=simple3ds then begin
ff_:=tp mat((2*u1+u2+2*u3+u2*u3
  ,u1-u2+u3-u1*u3
  ,-3*u1-u2-3*u3-u1*u2));
evalm_:=mat((-1,-1));
ee_:=tp mat((1,-1,-1),(1,7/2,-5/2));
zz_:=tp mat((0,1,0),(1,0,1));
end;
```

The adjusted dynamical system Modified in order cater for the generalised eigenvector.

$$\dot u_1 = \varepsilon\big(u_3u_2 - u_3 - u_1\big) + 3u_3 + u_2 + 3u_1$$

$$\dot u_2 = \varepsilon\big(-u_3u_1 + u_3 + u_1\big) - u_2$$

$$\dot u_3 = \varepsilon\big(u_3 - u_2u_1 + u_1\big) - 4u_3 - u_2 - 4u_1$$

The stable manifold noting the double i factors give decaying modes.

$$u_1 = \varepsilon\big(-51/4\,e^{2iti}s_2^2 - 3\,e^{2iti}s_2s_1 + 3\,e^{2iti}s_1^2\big) + e^{iti}s_2 + e^{iti}s_1$$

$$u_2 = \varepsilon\big(-5/2\,e^{2iti}s_2^2 - 7/2\,e^{2iti}s_2s_1 - e^{2iti}s_1^2\big) + 7/2\,e^{iti}s_2 - e^{iti}s_1$$

$$u_3 = \varepsilon\big(25\,e^{2iti}s_2^2 + 13/2\,e^{2iti}s_2s_1 - 5\,e^{2iti}s_1^2\big) - 5/2\,e^{iti}s_2 - e^{iti}s_1$$

Stable manifold ODEs $\dot s_1 = 3/2\varepsilon s_2$ and $\dot s_2 = 0$

2.11 3D system with a generalised eigenvector

Took longer to converge, but converge it does. However, now I force the off-diagonal term to be small.

$$\dot{u}_1 = \varepsilon\big(u_3u_2 + u_3 + u_2 + u_1\big) + u_3 + u_1$$

$$\dot{u}_2 = -\varepsilon u_3u_1 + u_3 + u_1$$

$$\dot{u}_3 = \varepsilon\big(-u_3 - u_2u_1 - u_2 - u_1\big) - 2u_3 - 2u_1$$

```
if thecase=geneigenvec then begin
ff_:=tp mat((
      2*u1+u2+2*u3+u2*u3,
      u1+u3-u1*u3,
      -3*u1-u2-3*u3-u1*u2
      ));
evalm_:=mat((0,0));
ee_:=tp mat((1,0,-1),(0,1,0));
zz_:=tp mat((1,-1,0),(1,1,1));
toosmall:=3;
end;
```

The centre manifold $u_1 = 2\varepsilon s_2s_1 + s_1$

$$u_2 = 2\varepsilon s_2s_1 + s_2$$

$$u_3 = -4\varepsilon s_2s_1 - s_1$$

Centre manifold ODEs $\dot{s}_1 = \varepsilon^2\big(-10s_2^2s_1 - 6s_2s_1^2\big) + \varepsilon\big(-3s_2s_1 + s_2\big)$

$$\dot{s}_2 = \varepsilon^2\big(-6s_2^2s_1 + 2s_2s_1^2\big) + \varepsilon\big(-2s_2s_1 + s_1^2\big)$$

Normals to isochrons at the slow manifold

$$z_{11} = \varepsilon^2\big(50s_2^2 + 60s_2s_1 + 14s_1^2 + s_1\big) + \varepsilon\big(5s_2 + 3s_1\big) + 2$$

$$z_{12} = \varepsilon^2\big(10s_2s_1 + 6s_1^2\big)$$

$$z_{13} = \varepsilon^2\big(40s_2^2 + 54s_2s_1 + 14s_1^2 + s_1\big) + \varepsilon\big(5s_2 + 3s_1\big) + 1$$
$$z_{21} = \varepsilon^2\big(31s_2^2 + 8s_2s_1 - s_2 - 9s_1^2\big) + \varepsilon\big(3s_2 - s_1\big) + 1$$
$$z_{22} = \varepsilon^2\big(6s_2s_1 - 2s_1^2\big) + 1$$
$$z_{23} = \varepsilon^2\big(25s_2^2 + 10s_2s_1 - s_2 - 9s_1^2\big) + \varepsilon\big(3s_2 - s_1\big) + 1$$

2.12 Separated system

To see if small part in the slow variable ruins convergence. The answer is that it did—hence we include code to make anything non-oscillatory in the slow variables to be small. Also test a non-zero constant forcing.

$$\dot u_1 = \varepsilon\big(- u_2u_1 + u_1\alpha\big)$$
$$\dot u_2 = \varepsilon\big(\beta - 2u_2^2 + u_1^2\big) - u_2$$

```
if thecase=bifurcate2d then begin
ff_:=tp mat((
    alpha*u1-u1*u2,
    -u2+u1^2-2*u2^2+beta
    ));
evalm_:=mat((0));
ee_:=tp mat((1,0));
zz_:=tp mat((1,0));
toosmall:=4;
end;
```

The centre manifold $u_1 = s_1$

$$u_2 = \varepsilon(s_1^2 + \beta)$$

Centre manifold ODEs $\dot s_1 = -\varepsilon^2(s_1^3 - \beta s_1) + \varepsilon s_1\alpha$

Normals to isochrons at the slow manifold

$z_{11} = 2\varepsilon^2 s_1^2 + 1$

$z_{12} = -\varepsilon s_1$

2.13 Oscillatory centre manifold—separated form

Let's try complex eigenvectors. Adjoint eigenvectors zz_ must be the eigenvectors of the complex conjugate transpose matrix.

$\dot{u}_1 = u_2$

$\dot{u}_2 = -\varepsilon u_3 u_1 - u_1$

$\dot{u}_3 = 5\varepsilon u_1^2 - u_3$

```
if thecase=simpleosc then begin
ff_:=tp mat((u2,-u1-u1*u3,-u3+5*u1^2));
evalm_:=mat((i,-i));
ee_:=tp mat((1,+i,0),(1,-i,0));
%ee_:=tp mat((1+1/10,+i,0),(1+1/10,-i,0)); % causes fail, Jan 201
zz_:=tp mat((1,+i,0),(1,-i,0));
end;
```

The centre manifold $u_1 = e^{-ti} s_2 + e^{ti} s_1$

$u_2 = -e^{-ti} s_2 i + e^{ti} s_1 i$

$u_3 = \varepsilon\big(2\,e^{-2ti} s_2^2 i + e^{-2ti} s_2^2 - 2\,e^{2ti} s_1^2 i + e^{2ti} s_1^2 + 10 s_2 s_1\big)$

Centre manifold ODEs $\dot{s}_1 = \varepsilon^2\big(11/2 s_2 s_1^2 i + s_2 s_1^2\big)$

$\dot{s}_2 = \varepsilon^2\big(-11/2 s_2^2 s_1 i + s_2^2 s_1\big)$

2.14 Perturbed frequency oscillatory centre manifold—separated form

Putting real parameters into the linear operator works here also.

$$\dot u_1 = \varepsilon\big(u_2 b + u_1 a\big) + u_2$$

$$\dot u_2 = \varepsilon\big(u_2 d - u_1 c\big) - u_1$$

$$\dot u_3 = -u_3$$

```
if thecase=perturbfreq then begin
ff_:=tp mat((a*u1+(1+b)*u2,d*u2-(1+c)*u1,-u3));
evalm_:=mat((i,-i));
ee_:=tp mat((1,+i,0),(1,-i,0));
zz_:=tp mat((1,+i,0),(1,-i,0));
b:=c:=0; d:=a;
toosmall:=2;
end;
```

The centre manifold $u_1 = \varepsilon\big(1/4\, e^{-ti} s_2 a i + 1/4\, e^{-ti} s_2 b - 1/4\, e^{-ti} s_2 c - 1/4\, e^{-ti} s_2 d i - 1/4\, e^{ti} s_1 a i + 1/4\, e^{ti} s_1 b - 1/4\, e^{ti} s_1 c + 1/4\, e^{ti} s_1 d i\big) + e^{-ti} s_2 + e^{ti} s_1$

$$u_2 = \varepsilon\big(-1/4\, e^{-ti} s_2 a + 1/4\, e^{-ti} s_2 b i - 1/4\, e^{-ti} s_2 c i + 1/4\, e^{-ti} s_2 d - 1/4\, e^{ti} s_1 a - 1/4\, e^{ti} s_1 b i + 1/4\, e^{ti} s_1 c i + 1/4\, e^{ti} s_1 d\big) - e^{-ti} s_2 i + e^{ti} s_1 i$$

$$u_3 = 0$$

Centre manifold ODEs

$$\dot s_1 = \varepsilon^2\big(-1/8 s_1 a^2 i + 1/4 s_1 a d i - 1/8 s_1 b^2 i + 1/4 s_1 b c i - 1/8 s_1 c^2 i - 1/8 s_1 d^2 i\big) + \varepsilon\big(1/2 s_1 a + 1/2 s_1 b i + 1/2 s_1 c i + 1/2 s_1 d\big)$$

$$\dot s_2 = \varepsilon^2\big(1/8 s_2 a^2 i - 1/4 s_2 a d i + 1/8 s_2 b^2 i - 1/4 s_2 b c i + 1/8 s_2 c^2 i + 1/8 s_2 d^2 i\big) + \varepsilon\big(1/2 s_2 a - 1/2 s_2 b i - 1/2 s_2 c i + 1/2 s_2 d\big)$$

2.15 More general oscillatory centre manifold

Consider the frequency two dynamics of the following system in non-separated form.

$$\dot u_1 = \varepsilon\big(u_2u_1 + u_1\epsilon\big) - 2u_3 - 2u_2$$

$$\dot u_2 = -2u_3 - 3u_2 + u_1$$

$$\dot u_3 = 2u_3 + 3u_2 + u_1$$

```
if thecase=nonseparatedosc then begin
ff_:=tp mat((
    -2*u2-2*u3+epsilon*u1+u1*u2,
    u1-3*u2-2*u3,
    u1+3*u2+2*u3
    ));
evalm_:=mat((+2*i,-2*i));
ee_:=tp mat((1,1,-1-i),(1,1,-1+i));
zz_:=tp mat((1,-i,-i),(1,+i,+i));
end;
```

The centre manifold $u_1 = \varepsilon\big(1/3\,e^{-4ti}s_2^2i + 1/8\,e^{-2ti}s_2\epsilon i - 1/3\,e^{4ti}s_1^2i - 1/8\,e^{2ti}s_1\epsilon i\big) + e^{-2ti}s_2 + e^{2ti}s_1$

$u_2 = \varepsilon\big(5/51\,e^{-4ti}s_2^2i - 1/17\,e^{-4ti}s_2^2 - 11/40\,e^{-2ti}s_2\epsilon i - 1/5\,e^{-2ti}s_2\epsilon - 5/51\,e^{4ti}s_1^2i - 1/17\,e^{4ti}s_1^2 + 11/40\,e^{2ti}s_1\epsilon i - 1/5\,e^{2ti}s_1\epsilon - 2s_2s_1\big) + e^{-2ti}s_2 + e^{2ti}s_1$

$u_3 = \varepsilon\big(- 5/51\,e^{-4ti}s_2^2i - 11/102\,e^{-4ti}s_2^2 + 11/40\,e^{-2ti}s_2\epsilon i + 13/40\,e^{-2ti}s_2\epsilon + 5/51\,e^{4ti}s_1^2i - 11/102\,e^{4ti}s_1^2 - 11/40\,e^{2ti}s_1\epsilon i + 13/40\,e^{2ti}s_1\epsilon + 3s_2s_1\big) + e^{-2ti}s_2i - e^{-2ti}s_2 - e^{2ti}s_1i - e^{2ti}s_1$

Centre manifold ODEs $\dot s_1 = \varepsilon^2\big(-11/51s_2s_1^2i - 35/34s_2s_1^2 - 1/16s_1\epsilon^2i\big) + 1/2\varepsilon s_1\epsilon$

$\dot s_2 = \varepsilon^2\big(11/51s_2^2s_1i - 35/34s_2^2s_1 + 1/16s_2\epsilon^2i\big) + 1/2\varepsilon s_2\epsilon$

2.16 Quasi-delay differential equation

Shows Hopf bifurcation as parameter a crosses -4 to oscillations with base frequency two.

$\dot u_1 = \varepsilon^2\big(-u_3\alpha - u_1^3\big) - 2\varepsilon u_1^2 - 4u_3$

$\dot u_2 = -2u_2 + 2u_1$

$\dot u_3 = -2u_3 + 2u_2$

```
if thecase=quasidelayosc then begin
ff_:=tp mat((
    -4*u3-small*alpha*u3-2*u1^2-small*u1^3,
    2*u1-2*u2,
    2*u2-2*u3
    ));
evalm_:=mat((2*i,-2*i));
ee_:=tp mat((1,1/2-i/2,-i/2),(1,1/2+i/2,+i/2));
zz_:=tp mat((1,-i,-1-i),(1,+i,-1+i));
end;
```

The centre manifold $u_1 = \varepsilon\big(-7/12\, e^{-4ti} s_2^2 i + 1/12\, e^{-4ti} s_2^2 + 7/12\, e^{4ti} s_1^2 i + 1/12\, e^{4ti} s_1^2 - s_2 s_1\big) + e^{-2ti} s_2 + e^{2ti} s_1$

$u_2 = \varepsilon\big(-1/12\, e^{-4ti} s_2^2 i + 1/4\, e^{-4ti} s_2^2 + 1/12\, e^{4ti} s_1^2 i + 1/4\, e^{4ti} s_1^2 - s_2 s_1\big) + 1/2\, e^{-2ti} s_2 i + 1/2\, e^{-2ti} s_2 - 1/2\, e^{2ti} s_1 i + 1/2\, e^{2ti} s_1$

$u_3 = \varepsilon\big(1/12\, e^{-4ti} s_2^2 i + 1/12\, e^{-4ti} s_2^2 - 1/12\, e^{4ti} s_1^2 i + 1/12\, e^{4ti} s_1^2 - s_2 s_1\big) + 1/2\, e^{-2ti} s_2 i - 1/2\, e^{2ti} s_1 i$

Centre manifold ODEs $\dot s_1 = \varepsilon^2\big(-16/15 s_2 s_1^2 i - 1/5 s_2 s_1^2 + 1/5 s_1 \alpha i + 1/10 s_1 \alpha\big)$

$\dot s_2 = \varepsilon^2\big(16/15 s_2^2 s_1 i - 1/5 s_2^2 s_1 - 1/5 s_2 \alpha i + 1/10 s_2 \alpha\big)$

2.17 Detuned version of quasi-delayed

The following modified version of the previous shows that we can 'detune' the linear operator and my 'adjustment' of the linear operator seems to work. Here the 1/2 in $\mathcal{L}_{1,1}$ should be zero for these eigenvectors: my adjustment seems to fix it OK. But now, knowing the frequencies, my adjustment is different (and probably better).

$\dot u_1 = \varepsilon^2\big(-u_3\alpha - u_1^3\big) + \varepsilon\big(-1/5u_3 + 1/5u_2 - 2u_1^2 + 2/5u_1\big) - 19/5u_3 - 1/5u_2 + 1/10u_1$

$\dot u_2 = -2u_2 + 2u_1$

$\dot u_3 = -2u_3 + 2u_2$

```
if thecase=quasidelayoscmod then begin
ff_:=tp mat((
    u1/2-4*u3-small*alpha*u3-2*u1^2-small*u1^3,
    2*u1-2*u2,
    2*u2-2*u3
    ));
evalm_:=mat((2*i,-2*i));
ee_:=tp mat((1,1/2-i/2,-i/2),(1,1/2+i/2,+i/2));
zz_:=tp mat((1,-i,-1-i),(1,+i,-1+i));
toosmall:=3;
end;
```

The centre manifold

$u_1 = \varepsilon\big(-1840/3121\, e^{-4ti} s_2^2 i + 860/9363\, e^{-4ti} s_2^2 + 237/3842\, e^{-2ti} s_2 i + 87/1921\, e^{-2ti} s_2 +$ $1840/3121\, e^{4ti} s_1^2 i + 860/9363\, e^{4ti} s_1^2 - 237/3842\, e^{2ti} s_1 i + 87/1921\, e^{2ti} s_1 - 40/39 s_2 s_1\big) +$ $e^{-2ti} s_2 + e^{2ti} s_1$

$u_2 = \varepsilon\big(-760/9363\, e^{-4ti} s_2^2 i + 2380/9363\, e^{-4ti} s_2^2 + 21/7684\, e^{-2ti} s_2 i + 137/7684\, e^{-2ti} s_2 +$ $760/9363\, e^{4ti} s_1^2 i + 2380/9363\, e^{4ti} s_1^2 - 21/7684\, e^{2ti} s_1 i + 137/7684\, e^{2ti} s_1 - 40/39 s_2 s_1\big) +$ $1/2\, e^{-2ti} s_2 i + 1/2\, e^{-2ti} s_2 - 1/2\, e^{2ti} s_1 i + 1/2\, e^{2ti} s_1$

$u_3 = \varepsilon\big(800/9363\, e^{-4ti} s_2^2 i + 260/3121\, e^{-4ti} s_2^2 - 4/1921\, e^{-2ti} s_2 i + 353/7684\, e^{-2ti} s_2 - 800/9363\, e^{4ti} s_1^2 i + 260/3121\, e^{4ti} s_1^2 + 4/1921\, e^{2ti} s_1 i + 353/7684\, e^{2ti} s_1 - 40/39 s_2 s_1\big) + 1/2\, e^{-2ti} s_2 i - 1/2\, e^{2ti} s_1 i$

Centre manifold ODEs

$\dot s_1 = \varepsilon^2\big(-259684400/233822199 s_2 s_1^2 i - 1154340/5995441 s_2 s_1^2 + 390/1921 s_1 \alpha i + 200/1921 s_1 \alpha - 90446425/7088952961 s_1 i - 1300360/7088952961 s_1\big) + \varepsilon\big(-200/1921 s_1 i + 390/1921 s_1\big)$

$\dot s_2 = \varepsilon^2\big(259684400/233822199 s_2^2 s_1 i - 1154340/5995441 s_2^2 s_1 - 390/1921 s_2 \alpha i + 200/1921 s_2 \alpha + 90446425/7088952961 s_2 i - 1300360/7088952961 s_2\big) + \varepsilon\big(200/1921 s_2 i + 390/1921 s_2\big)$

Observe the terms linear in ε due to my fudging of the linear dynamics.

2.18 Rossler-like system

Has Hopf bifurcation as parameter crosses zero to oscillations of base frequency one.

$\dot u_1 = -u_3 - u_2$

$\dot u_2 = \varepsilon u_2 a + u_1$

$\dot u_3 = \varepsilon\big(u_3 u_1 - 1/5 u_2 u_1\big) - 5u_3$

```
if thecase=rosslerlike then begin
ff_:=tp mat((
    -u2-u3,
    u1+small*a*u2,
    -5*u3-u1*u2/5+u1*u3
    ));
evalm_:=mat((i,-i));
ee_:=tp mat((1,-i,0),(1,i,0));
zz_:=tp mat((-5+i,1+5*i,1),(-5-i,1-5*i,1));
end;
```

The centre manifold $u_1 = \varepsilon\big(-4/435\, e^{-2ti} s_2^2 i - 2/87\, e^{-2ti} s_2^2 - 1/4\, e^{-ti} s_2 a i + 4/435\, e^{2ti} s_1^2 i - 2/87\, e^{2ti} s_1^2 + 1/4\, e^{ti} s_1 a i\big) + e^{-ti} s_2 + e^{ti} s_1$

$u_2 = \varepsilon\big(-1/87\, e^{-2ti} s_2^2 i + 2/435\, e^{-2ti} s_2^2 - 1/4\, e^{-ti} s_2 a + 1/87\, e^{2ti} s_1^2 i + 2/435\, e^{2ti} s_1^2 - 1/4\, e^{ti} s_1 a\big) + e^{-ti} s_2 i - e^{ti} s_1 i$

$u_3 = \varepsilon\big(- 1/29\, e^{-2ti} s_2^2 i + 2/145\, e^{-2ti} s_2^2 + 1/29\, e^{2ti} s_1^2 i + 2/145\, e^{2ti} s_1^2\big)$

Centre manifold ODEs $\dot s_1 = \varepsilon^2\big(-92/28275 s_2 s_1^2 i - 4/1885 s_2 s_1^2 - 1/8 s_1 a^2 i\big) + 1/2 \varepsilon s_1 a$

$\dot s_2 = \varepsilon^2\big(92/28275 s_2^2 s_1 i - 4/1885 s_2^2 s_1 + 1/8 s_2 a^2 i\big) + 1/2 \varepsilon s_2 a$

2.19 Fudge a couple of these oscillations together

Use say different base frequencies of one and two. Put in a couple of coupling terms. It seems to work fine, although the computation time zooms up even for the basic third order errors.

$\dot u_1 = \varepsilon u_4^2 - u_3 - u_2$

$\dot u_2 = \varepsilon u_2 a + u_1$

$\dot u_3 = \varepsilon\big(u_3 u_1 - 1/5 u_2 u_1\big) - 5u_3$

$\dot u_4 = \varepsilon\big(u_6 u_5 + u_4 \epsilon\big) - 2u_6 - 2u_5$

$\dot u_5 = \varepsilon u_1^2 - 2u_6 - 3u_5 + u_4$

$\dot u_6 = 2u_6 + 3u_5 + u_4$

```
if thecase=doubleosc then begin
ff_:=tp mat((
  -u2-u3+u4^2,
  u1+a*u2,
  -5*u3-u1*u2/5+u1*u3,
  -2*u5-2*u6+small*epsilon*u4+u5*u6,
  u4-3*u5-2*u6+u1^2,
  u4+3*u5+2*u6
```

```
  ));
evalm_:=mat((i,-i,2*i,-2*i));
ee_:=tp mat((1,-i,0,0,0,0),(1,i,0,0,0,0)
  ,(0,0,0,1,1,-1-i),(0,0,0,1,1,-1+i));
zz_:=tp mat((-5+i,1+5*i,1,0,0,0),(-5-i,1-5*i,1,0,0,0)
  ,(0,0,0,1,-i,-i),(0,0,0,1,+i,+i));
end;
```

The centre manifold $u_1 = \varepsilon\big(4/15\, e^{-4ti} s_4^2 i - 4/435\, e^{-2ti} s_2^2 i - 2/87\, e^{-2ti} s_2^2 - 1/4\, e^{-ti} s_2 a i - 4/15\, e^{4ti} s_3^2 i + 4/435\, e^{2ti} s_1^2 i - 2/87\, e^{2ti} s_1^2 + 1/4\, e^{ti} s_1 a i\big) + e^{-ti} s_2 + e^{ti} s_1$

$u_2 = \varepsilon\big(-1/15\, e^{-4ti} s_4^2 - 1/87\, e^{-2ti} s_2^2 i + 2/435\, e^{-2ti} s_2^2 - 1/4\, e^{-ti} s_2 a - 1/15\, e^{4ti} s_3^2 + 1/87\, e^{2ti} s_1^2 i + 2/435\, e^{2ti} s_1^2 - 1/4\, e^{ti} s_1 a + 2 s_4 s_3\big) + e^{-ti} s_2 i - e^{ti} s_1 i$

$u_3 = \varepsilon\big(- 1/29\, e^{-2ti} s_2^2 i + 2/145\, e^{-2ti} s_2^2 + 1/29\, e^{2ti} s_1^2 i + 2/145\, e^{2ti} s_1^2\big)$

$u_4 = \varepsilon\big(-1/3\, e^{-4ti} s_4^2 i - 1/3\, e^{-4ti} s_4^2 + 1/8\, e^{-2ti} s_4 \epsilon i - 1/8\, e^{-2ti} s_2^2 + 1/3\, e^{4ti} s_3^2 i - 1/3\, e^{4ti} s_3^2 - 1/8\, e^{2ti} s_3 \epsilon i - 1/8\, e^{2ti} s_1^2 - s_2 s_1\big) + e^{-2ti} s_4 + e^{2ti} s_3$

$u_5 = \varepsilon\big(- 8/51\, e^{-4ti} s_4^2 i - 2/51\, e^{-4ti} s_4^2 - 11/40\, e^{-2ti} s_4 \epsilon i - 1/5\, e^{-2ti} s_4 \epsilon + 2/5\, e^{-2ti} s_2^2 i + 3/40\, e^{-2ti} s_2^2 + 8/51\, e^{4ti} s_3^2 i - 2/51\, e^{4ti} s_3^2 + 11/40\, e^{2ti} s_3 \epsilon i - 1/5\, e^{2ti} s_3 \epsilon - 2/5\, e^{2ti} s_1^2 i + 3/40\, e^{2ti} s_1^2 + 2 s_4 s_3 + s_2 s_1\big) + e^{-2ti} s_4 + e^{2ti} s_3$

$u_6 = \varepsilon\big(- 1/102\, e^{-4ti} s_4^2 i + 7/34\, e^{-4ti} s_4^2 + 11/40\, e^{-2ti} s_4 \epsilon i + 13/40\, e^{-2ti} s_4 \epsilon - 11/40\, e^{-2ti} s_2^2 i - 3/40\, e^{-2ti} s_2^2 + 1/102\, e^{4ti} s_3^2 i + 7/34\, e^{4ti} s_3^2 - 11/40\, e^{2ti} s_3 \epsilon i + 13/40\, e^{2ti} s_3 \epsilon + 11/40\, e^{2ti} s_1^2 i - 3/40\, e^{2ti} s_1^2 - 3 s_4 s_3 - s_2 s_1\big) + e^{-2ti} s_4 i - e^{-2ti} s_4 - e^{2ti} s_3 i - e^{2ti} s_3$

Centre manifold ODEs $\dot s_1 = \varepsilon^2\big(-1/130 s_4 s_3 s_1 i + 1/26 s_4 s_3 s_1 - 92/28275 s_2 s_1^2 i - 4/1885 s_2 s_1^2 - 1/8 s_1 a^2 i\big) + 1/2 \varepsilon s_1 a$

$\dot s_2 = \varepsilon^2\big(1/130 s_4 s_3 s_2 i + 1/26 s_4 s_3 s_2 + 92/28275 s_2^2 s_1 i - 4/1885 s_2^2 s_1 + 1/8 s_2 a^2 i\big) + 1/2 \varepsilon s_2 a$

$\dot s_3 = \varepsilon^2\big(- 223/204 s_4 s_3^2 i - 167/68 s_4 s_3^2 - 1/2 s_3 s_2 s_1 i - s_3 s_2 s_1 - 1/16 s_3 \epsilon^2 i - 1/4 s_1^2 a - 1/16 s_1^2 \epsilon\big) + \varepsilon\big(1/2 s_3 \epsilon + 1/2 s_1^2 i\big)$

$\dot s_4 = \varepsilon^2\big(223/204 s_4^2 s_3 i - 167/68 s_4^2 s_3 + 1/2 s_4 s_2 s_1 i - s_4 s_2 s_1 + 1/16 s_4 \epsilon^2 i - 1/4 s_2^2 a - 1/16 s_2^2 \epsilon\big) + \varepsilon\big(1/2 s_4 \epsilon - 1/2 s_2^2 i\big)$

2.20 Fudge an oscillatory mode

With frequency two, with a system with one slow mode. Couple them with something ad hoc.

$\dot u_1 = \varepsilon\big(u_4 u_1 + u_2 u_1\big) - 2u_3 - 2u_2$

$\dot u_2 = -2u_3 - 3u_2 + u_1$

$\dot u_3 = 2u_3 + 3u_2 + u_1$

$\dot u_4 = \varepsilon\big(- u_4^2 - u_2 u_1\big) + u_5 - u_4$

$\dot u_5 = \varepsilon u_5^2 - u_5 + u_4$

```
if thecase=oscmeanflow then begin
ff_:=tp mat((
    -2*u2-2*u3+u4*u1+u1*u2,
    u1-3*u2-2*u3,
    u1+3*u2+2*u3,
    -u4+u5-u4^2-u1*u2,
    +u4-u5+u5^2
    ));
evalm_:=mat((2*i,-2*i,0));
ee_:=tp mat((1,1,-1-i,0,0),(1,1,-1+i,0,0)
  ,(0,0,0,1,1));
zz_:=tp mat((1,-i,-i,0,0),(1,+i,+i,0,0)
  ,(0,0,0,1,1));
end;
```

The centre manifold $u_1 = \varepsilon\big(1/3\, e^{-4ti} s_2^2 i + 1/8\, e^{-2ti} s_3 s_2 i - 1/3\, e^{4ti} s_1^2 i - 1/8\, e^{2ti} s_3 s_1 i\big) + e^{-2ti} s_2 + e^{2ti} s_1$

$u_2 = \varepsilon\big(5/51\, e^{-4ti} s_2^2 i - 1/17\, e^{-4ti} s_2^2 - 11/40\, e^{-2ti} s_3 s_2 i - 1/5\, e^{-2ti} s_3 s_2 - 5/51\, e^{4ti} s_1^2 i - 1/17\, e^{4ti} s_1^2 + 11/40\, e^{2ti} s_3 s_1 i - 1/5\, e^{2ti} s_3 s_1 - 2 s_2 s_1\big) + e^{-2ti} s_2 + e^{2ti} s_1$

$u_3 = \varepsilon\big(-5/51\, e^{-4ti} s_2^2 i - 11/102\, e^{-4ti} s_2^2 + 11/40\, e^{-2ti} s_3 s_2 i + 13/40\, e^{-2ti} s_3 s_2 + 5/51\, e^{4ti} s_1^2 i - 11/102\, e^{4ti} s_1^2 - 11/40\, e^{2ti} s_3 s_1 i + 13/40\, e^{2ti} s_3 s_1 + 3 s_2 s_1\big) + e^{-2ti} s_2 i - e^{-2ti} s_2 - e^{2ti} s_1 i - e^{2ti} s_1$

$u_4 = \varepsilon\big(-9/40\, e^{-4ti} s_2^2 i - 1/20\, e^{-4ti} s_2^2 + 9/40\, e^{4ti} s_1^2 i - 1/20\, e^{4ti} s_1^2 - 1/2 s_3^2 - 1/2 s_2 s_1\big) + s_3$

$u_5 = \varepsilon\big(-1/40\, e^{-4ti} s_2^2 i + 1/20\, e^{-4ti} s_2^2 + 1/40\, e^{4ti} s_1^2 i + 1/20\, e^{4ti} s_1^2 + 1/2 s_3^2 + 1/2 s_2 s_1\big) + s_3$

Centre manifold ODEs $\dot s_1 = \varepsilon^2\big(-1/16 s_3^2 s_1 i - 1/4 s_3^2 s_1 - 421/4080 s_2 s_1^2 i - 887/680 s_2 s_1^2\big) + 1/2 \varepsilon s_3 s_1$

$\dot s_2 = \varepsilon^2\big(1/16 s_3^2 s_2 i - 1/4 s_3^2 s_2 + 421/4080 s_2^2 s_1 i - 887/680 s_2^2 s_1\big) + 1/2 \varepsilon s_3 s_2$

$\dot s_3 = \varepsilon^2\big(s_3^3 + 6/5 s_3 s_2 s_1\big) - \varepsilon s_2 s_1$

Used this system for a benchmark to compare several ways of handling matrices and vectors. This analysis using `e_` as basis for matrices and vectors takes about a second or two in the following five iterations.

```
lengthres := 10
Time: 20 ms
lengthres := 124
Time: 120 ms
lengthres := 289
Time: 420 ms
lengthres := 169
Time: 580 ms
lengthres := 1
Time: 420 ms
SUCCESS: converged to an expansion
```

2.21 Modulate Duffing oscillation

Tests that this code generates complex amplitude model for purely oscillating dynamics. Here model the frequency correction in the Duffing oscillator $\ddot u + u - u^3 = 0$. Code for $u_1 = u$ and $u_2 = \dot u$.

$\dot u_1 = u_2$

$\dot u_2 = \varepsilon u_1^3 - u_1$

```
if thecase=modulateduffing then begin
ff_:=tp mat((u2,-u1+u1^3-small*2*nu*u2));
evalm_:=mat((i,-i));
ee_:=tp mat((1,i),(1,-i));
zz_:=tp mat((1,i),(1,-i));
end;
```

Find the coordinate transform is $u_1 = \varepsilon\big(- 1/8\, e^{-3ti} s_2^3 + 3/4\, e^{-ti} s_2^2 s_1 - 1/8\, e^{3ti} s_1^3 + 3/4\, e^{ti} s_2 s_1^2\big) + e^{-ti} s_2 + e^{ti} s_1$ where the amplitudes evolve according to $\dot s_1 = -51/16 \varepsilon^2 s_2^2 s_1^3 i - 3/2 \varepsilon s_2 s_1^2 i$ and its complex conjugate. This correctly predicts the frequency shift in the Duffing oscillator.

2.22 Modulate another oscillation

Retest that this code generates complex amplitude model for purely oscillating dynamics. Here model the frequency correction in the oscillator $\ddot u + u + \dot u^3 = 0$. Code for $u_1 = u$ and $u_2 = \dot u$.

$\dot u_1 = u_2$

$\dot u_2 = -\varepsilon u_2^3 - u_1$

```
if thecase=modulateoscillator then begin
ff_:=tp mat((u2,-u1-u2^3));
evalm_:=mat((i,-i));
ee_:=tp mat((1,i),(1,-i));
zz_:=tp mat((1,i),(1,-i));
end;
```

The coordinate transform $u_1 = e^{-ti}s_2 + e^{ti}s_1 + \varepsilon\big(1/8\,e^{-3ti}s_2^3 i + 3/4\,e^{-ti}s_2^2 s_1 i - 1/8\,e^{3ti}s_1^3 i - 3/4\,e^{ti}s_2 s_1^2 i\big)$ looks fine; although note that here higher orders do differ to other work due to the orthogonality I build in. The evolution seems appropriate: $\dot s_1 = -3/2\varepsilon s_2 s_1^2 - 27/16\varepsilon^2 s_2^2 s_1^3 i$

2.23 An example from Iulian Stoleriu

Consider the case Stoleriu (2012) calls $(3\pi/4, k^2/2)$. Use Taylor expansions for trigonometric functions in the ODEs. Eigenvalues are ±1 and $\pm i$, so we find the centre manifold among stable and unstable modes. Sometimes we can have a parameter (here σ) in the linear operator, but may need to specify its real and imaginary parts.

```
if thecase=StoleriuOne then begin
let {repart(sigma)=>sigma,impart(sigma)=>0};
ff_:=tp mat((
    u2,
    sigma*u3+u1^2/2-small*u1^4/24,
    u4,
    u1/sigma+u3*u1+(u3+1/sigma)*(-small*u1^3/6)
    ));
evalm_:=mat((i,-i));
ee_:=tp mat((sigma,i*sigma,-1,-i),(sigma,-i*sigma,-1,+i));
zz_:=tp mat((+i,-1,-i*sigma,sigma),(-i,-1,+i*sigma,sigma));
end;
```

A centre manifold is $x = u_1 = \varepsilon\big(-1/5\,e^{-2ti}s_2^2\sigma^2 - 1/5\,e^{2ti}s_1^2\sigma^2 + 2s_2 s_1\sigma^2\big) + e^{-ti}s_2\sigma + e^{ti}s_1\sigma$ and $y = u_3 = \varepsilon\big(3/10\,e^{-2ti}s_2^2\sigma + 3/10\,e^{2ti}s_1^2\sigma - s_2 s_1\sigma\big) - e^{-ti}s_2 - e^{ti}s_1$. On this centre manifold the oscillations have a frequency shift, but no amplitude evolution (to this order nor the next): $\dot s_1 = -6/5\varepsilon^2 s_2 s_1^2 i\sigma^2$. Remember the system is unstable due to the unstable mode.

2.24 An second example from Iulian Stoleriu

Consider the case Stoleriu (2012) calls $(\pi/2, 0)$. Use Taylor expansions for trigonometric functions in the ODEs. Eigenvalues are $\pm i$, multiplicity two, so we find modulation equations for coupled oscillators.

The system is

- $\dot u_1 = u_2$
- $\dot u_2 = -1/120\varepsilon^2 u_1^5 + 1/6\varepsilon u_1^3 + u_3\sigma - u_1$
- $\dot u_3 = u_4$
- $\dot u_4 = -1/24\varepsilon^2 u_3 u_1^4 + 1/2\varepsilon u_3 u_1^2 - u_3$

```
if thecase=StoleriuTwo then begin
ff_:=tp mat((
    u2,
    -u1+u1^3/6-small*u1^5/120+sigma*u3,
    u4,
    -u3+u3*(u1^2/2-small*u1^4/24)
    ));
evalm_:=mat((i,-i,i,-i));
ee_:=tp mat((1,i,0,0),(1,-i,0,0),(0,0,1,i),(0,0,1,-i));
zz_:=tp mat((1,i,0,0),(1,-i,0,0),(0,0,1,i),(0,0,1,-i));
toosmall:=3;
end;
```

This used to take five iterates to construct the coordinate transform and modulation equations, but now less as the off-diagonal term is made small by the linear adjustment. The original variables are approximately

- $x = u_1 = 1/4\, e^{-ti} s_4\sigma + e^{-ti} s_2 + 1/4\, e^{ti} s_3\sigma + e^{ti} s_1$
- $y = u_3 = e^{-ti} s_4 + e^{ti} s_3$

The modulation equations are the following, and their complex conjugates:

- $\dot s_1 = \varepsilon\big(- 1/64 s_4 s_3^2 i\sigma^3 - 3/32 s_4 s_3 s_1 i\sigma^2 - 1/8 s_4 s_1^2 i\sigma - 5/64 s_3^2 s_2 i\sigma^2 - 1/4 s_3 s_2 s_1 i\sigma - 1/4 s_2 s_1^2 i\big) - 1/2 s_3 i\sigma$;
- $\dot s_3 = \varepsilon\big(-3/64 s_4 s_3^2 i\sigma^2 - 1/4 s_4 s_3 s_1 i\sigma - 1/4 s_4 s_1^2 i - 1/8 s_3^2 s_2 i\sigma - 1/2 s_3 s_2 s_1 i\big)$.

Since every term is multiplied by i one expects there to be just frequency shifts, but there are oscillator interaction terms as well. These should be equivalent to the averaging method, but more easily extended to higher order (just change parameter `toosmall`).

2.25 Periodic chronic myelogenous leukemia

Ion & Georgescu (2013) explored Hopf bifurcations in a delay differential equation modelling leukaemia:[2]

$$\dot x = -\frac{x(t)}{1+x(t)^n} - \delta x(t) + \frac{kx(t-r)}{1+x(t-r)^n}$$

For simplicity we fix upon parameters $n = 2$, $\delta \approx 1/8$, $k = 3/2$ and time delay $r = 64/3$; that is,

$$\dot x = -\frac{x(t)}{1+x(t)^2} - (\tfrac18 + \delta')x(t) + \frac{\frac32 x(t-r)}{1+x(t-r)^2}$$

Near these parameters the equilibrium $x = X = \sqrt3$ perhaps undergoes a Hopf bifurcation. 'Perhaps' because instead of a precise time delay, we model $x(t-r)$ via two intermediaries in the system, after defining $x(t) = X + u_1(t)$,

$$\begin{aligned}
\dot u_1 &= -\frac{(X+u_1)}{1+(X+u_1)^2} - (\tfrac18 + \delta')(X+u_1) + \frac{\frac32 (X+u_3)}{1+(X+u_3)^2}\,,\\
\dot u_2 &= \frac{3}{32}(u_1-u_2),\\
\dot u_3 &= \frac{3}{32}(u_2-u_3).
\end{aligned}$$

[2] Their parameter β_0 is absorbed in a time scaling.

This system does undergo a Hopf bifurcation as δ' decreases through zero. My code only analyses multinomial forms, so Taylor expand the rational function:

$$\begin{aligned}\frac{X+u}{1+(X+u)^2} &= \frac{X}{1+X^2} + \frac{1-X^2}{(1+X^2)^2}u + \frac{X(X^2-3)}{(1+X^2)^3}u^2 + \frac{-1+6X^2-X^4}{(1+X^2)^4}u^3 + \cdots \\ &= \frac{\sqrt{3}}{4} - \frac{1}{8}u + 0u^2 + \frac{1}{32}u^3 + \cdots \quad \text{at } X=\sqrt{3}\,.\end{aligned}$$

```
if thecase=delayprolif then begin
ff_:=tp mat((
     -3/16*u3-u1^3/32-small*delta*(sqrt(3)+u1)+3/64*u3^3,
     3/32*u1-3/32*u2,
     3/32*u2-3/32*u3
     ));
evalm_:=mat((3/32*i,-3/32*i));
ee_:=tp mat((1,1/2-i/2,-i/2),(1,1/2+i/2,+i/2));
zz_:=tp mat((1,-i,-1-i),(1,+i,-1+i));
toosmall:=2;
factor delta,s;
end;
```

The specified dynamical system

$$\dot u_1 = \varepsilon\big(-\sqrt{3}\delta + 3/64u_3^3 - 1/32u_1^3 - u_1\delta\big) - 3/16u_3$$

$$\dot u_2 = -3/32u_2 + 3/32u_1$$

$$\dot u_3 = -3/32u_3 + 3/32u_2$$

The centre manifold

$$u_1 = e^{-3t/32i}s_2 + e^{3t/32i}s_1$$

$$u_2 = 1/2\,e^{-3t/32i}s_2i + 1/2\,e^{-3t/32i}s_2 - 1/2\,e^{3t/32i}s_1i + 1/2\,e^{3t/32i}s_1$$

$$u_3 = 1/2\,e^{-3t/32i}s_2i - 1/2\,e^{3t/32i}s_1i$$

Centre manifold ODEs

$$\dot s_1 = \varepsilon\big(3/256 s_2 s_1^2 i - 21/512 s_2 s_1^2 + 1/5 s_1 \delta i - 2/5 s_1 \delta\big)$$

$$\dot s_2 = \varepsilon\big(- 3/256 s_2^2 s_1 i - 21/512 s_2^2 s_1 - 1/5 s_2 \delta i - 2/5 s_2 \delta\big)$$

These indicate that $\vec s = \vec 0$ is stable for $\delta' \geq 0$. For parameter $\delta' < 0$ there is a stable limit cycle of amplitude $|s_j| = 16\sqrt{\frac{-2\delta'}{105}}$.

2.25.1 Delayed version

Return to the original system linearised about $x = \sqrt{3}$, the following finds the spectrum and identifies a Hopf bifurcation of frequency 3/16.

```
% linearised about x=sqrt3, freq is 3/16
delta=1/8, k=1+4*delta, r=8/3*pi
ce=@(z) -z+1/8-delta-k/8*exp(-r*z)
lams=fsolve(ce,randn(100,2)*[1;3*i]/2)
plot(real(lams),imag(lams),'o')
```

The following works only by careful use of smallness.

```
if thecase=delayedprolif then begin
r3:=sqrt(3);
delta:=1/8; k:=1+4*delta; r:=8/3*pi;
ff_:=tp mat((
    -r3*(1/4-3/8/r3*u1+1/8*u1^2-3/32/r3*u1^3*small)
    -u1*(1/4-3/8/r3*u1+1/8*u1^2*small)
%    -(r3+u1)*(1/4-3/8/r3*u1+1/8*u1^2-3/32/r3*small^2*u1^3)
    -delta*(r3+u1)
    +k*r3*(1/4-3/8/r3*u1(r)+1/8*u1(r)^2-3/32/r3*u1(r)^3*small)
    +k*u1(r)*(1/4-3/8/r3*u1(r)+1/8*u1(r)^2*small)
%    +k*(r3+u1(r))*(1/4-3/8/r3*u1(r)+1/8*u1(r)^2-small^2*3/32/r3*
    ));
evalm_:=mat((3/16*i,-3/16*i));
ee_:=tp mat((1),(1));
zz_:=tp mat((1),(1));
```

```
toosmall:=4;
factor s;
end;
```

The specified dynamical system

$$\dot u_1 = \varepsilon^2\big(3/64D_{t,(8\pi)/3}(u_1)^3 - 1/32u_1^3\big) - 3/16D_{t,(8\pi)/3}(u_1)$$

The centre manifold

$$u_1 = s_2^3\varepsilon^2\big(-1/24\,e^{(-9ti/16)}i + 1/16\,e^{(-9ti/16)}\big) + s_2\,e^{(-3ti/16)} + s_1^3\varepsilon^2\big(1/24\,e^{(9ti/16)}i$$
$$1/16\,e^{(9ti/16)}\big) + s_1\,e^{(3ti/16)}$$

Centre manifold ODEs

$$\dot s_1 = s_2 s_1^2\varepsilon^2\big(3/16i\pi - 9/16i - 9/32\pi - 3/8\big)/\big(\pi^2+4\big)$$
$$\dot s_2 = s_2^2 s_1\varepsilon^2\big(-3/16i\pi + 9/16i - 9/32\pi - 3/8\big)/\big(\pi^2+4\big)$$

2.26 Nonlinear normal modes of Renson

Renson et al. (2012) explored finite element construction of the nonlinear normal modes of a pair of coupled oscillators. Defining two new variables one of their example systems is

$$\begin{aligned}
\dot x_1 &= x_3\,,\\
\dot x_2 &= x_4\,,\\
\dot x_3 &= -2x_1 + x_2 - \tfrac12 x_1^3 + \tfrac{3}{10}(-x_3 + x_4)\,,\\
\dot x_4 &= x_1 - 2x_2 + \tfrac{3}{10}(x_3 - 2x_4)\,.
\end{aligned}$$

In the following code, force the linear damping to be effectively small (which then makes it small squared); consequently scale the smallness of the cubic nonlinearity.

```
if thecase=normalmodes then begin
r3:=sqrt(3);
ff_:=tp mat((
    u3,
    u4,
    -2*u1+u2-small*u1^3/2+small*3/10*(-u3+u4),
    u1-2*u2+small*3/10*(u3-2*u4)
    ));
evalm_:=mat((i,-i,r3*i,-r3*i));
ee_:=tp mat((1,1,+i,+i),(1,1,-i,-i)
          ,(1,-1,i*r3,-i*r3),(1,-1,-i*r3,i*r3));
zz_:=tp mat((1,1,+i,+i),(1,1,-i,-i)
          ,(-i*r3,+i*r3,1,-1),(+i*r3,-i*r3,1,-1));
toosmall:=4;
end;
```

The square root frequencies do not cause any trouble (although may need to reformat the LaTeX of the cis operator). In the model, observe that $s_1 = s_2 = 0$ is invariant, as is $s_3 = s_4 = 0$. These are the nonlinear normal modes.

The centre manifold

$$u_1 = e^{-\sqrt{3}ti} s_4 + e^{-ti} s_2 + e^{\sqrt{3}ti} s_3 + e^{ti} s_1$$

$$u_2 = -e^{-\sqrt{3}ti} s_4 + e^{-ti} s_2 - e^{\sqrt{3}ti} s_3 + e^{ti} s_1$$

$$u_3 = -\sqrt{3}\, e^{-\sqrt{3}ti} s_4 i - e^{-ti} s_2 i + \sqrt{3}\, e^{\sqrt{3}ti} s_3 i + e^{ti} s_1 i$$

$$u_4 = \sqrt{3}\, e^{-\sqrt{3}ti} s_4 i - e^{-ti} s_2 i - \sqrt{3}\, e^{\sqrt{3}ti} s_3 i + e^{ti} s_1 i$$

Centre manifold ODEs

$$\dot s_1 = \varepsilon\big(3/4 s_4 s_3 s_1 i + 3/8 s_2 s_1^2 i - 3/40 s_1\big)$$

$$\dot s_2 = \varepsilon\big(-3/4 s_4 s_3 s_2 i - 3/8 s_2^2 s_1 i - 3/40 s_2\big)$$

$$\dot s_3 = \varepsilon\big(1/8\sqrt{3} s_4 s_3^2 i + 1/4\sqrt{3} s_3 s_2 s_1 i - 3/8 s_3\big)$$

$$\dot s_4 = \varepsilon\big(- 1/8\sqrt{3}s_4^2s_3i - 1/4\sqrt{3}s_4s_2s_1i - 3/8s_4\big)$$

2.27 Periodically forced van der Pol oscillator

Hinvi et al. (2013) used renormalisation group to explore periodically forced van der Pol oscillator

$$\ddot x + x - \epsilon(1 - ax^2 - b\dot x^2)\dot x = \epsilon c\sin\Omega t\,.$$

Introducing $u_1 = x$, rewrite as the system

$$\begin{aligned}
\dot u_1 &= u_2\,,\\
\dot u_2 &= -u_1 + \epsilon(1 - au_1^2 - bu_2^2)u_2 + \epsilon cu_3\,,\\
\dot u_3 &= \Omega u_4\,,\\
\dot u_4 &= -\Omega u_3\,.
\end{aligned}$$

This system has eigenvalues $\pm i$ and $\pm i\Omega$ so we seek the modulation equations of the oscillations.

Only the directly resonant case appears to be interesting, so set $\Omega = 1$, and then perturb it in the equations.

```
if thecase=forcedvdp then begin
om:=1;
ff_:=tp mat((
    +u2,
    -u1+small*(1-a*u1^2-b*u2^2)*u2+small*c*u3,
    +om*u4*(1+small*omega),
    -om*u3*(1+small*omega)
    ));
evalm_:=mat((i,-i,om*i,-om*i));
ee_:=tp mat((1,+i,0,0),(1,-i,0,0)
          ,(0,0,1,+i),(0,0,1,-i));
zz_:=tp mat((1,+i,0,0),(1,-i,0,0)
          ,(0,0,1,+i),(0,0,1,-i));
toosmall:=4;
end;
```

2.28 Slow manifold of Lorenz 1986 model

In this case we construct the slow sub-centre manifold, analogous to quasi-geostrophy, in order to disentangle the slow dynamics from fast oscillations, analogous to gravity waves. The normals to the isochrons determine 'balancing' onto the slow manifold.

```
if thecase=lorenz86slow then begin
factor b;
ff_:=tp mat((-u2*u3+b*u2*u5
    ,u1*u3-b*u1*u5
    ,-u1*u2
    ,-u5
    ,+u4+b*u1*u2));
evalm_:=mat((0,0,0));
ee_:=zz_:=tp mat((1,0,0,0,0),(0,1,0,0,0),(0,0,1,0,0));
toosmall:=4;
end;
```

The centre manifold These give the location of the centre manifold in terms of parameters s_j.

$u_1 = s_1$

$u_2 = s_2$

$u_3 = s_3$

$u_4 = -b\varepsilon s_2 s_1$

$u_5 = b\varepsilon^2\big(-s_3 s_2^2 + s_3 s_1^2\big)$

Centre manifold ODEs The system evolves on the centre manifold such that the parameters evolve according to these ODEs.

$\dot s_1 = b^2\varepsilon^3\big(-s_3 s_2^3 + s_3 s_2 s_1^2\big) - \varepsilon s_3 s_2$

$\dot{s}_2 = b^2\varepsilon^3\big(s_3 s_2^2 s_1 - s_3 s_1^3\big) + \varepsilon s_3 s_1$

$\dot{s}_3 = -\varepsilon s_2 s_1$

Normals to isochrons at the slow manifold The normal vector $\vec{z}_j := (z_{j1}, \ldots, z_{jn}\ )$

$z_{11} = b^2\varepsilon^2 s_2^2 + 1$

$z_{12} = b^2\varepsilon^2 s_2 s_1$

$z_{13} = 0$

$z_{14} = b^3\varepsilon^3\big(s_2^3 - s_2 s_1^2\big) + b\varepsilon^3\big(- s_2^3 + s_2 s_1^2\big) + b\varepsilon s_2$

$z_{15} = 0$

$z_{21} = -b^2\varepsilon^2 s_2 s_1$

$z_{22} = -b^2\varepsilon^2 s_1^2 + 1$

$z_{23} = 0$

$z_{24} = b^3\varepsilon^3\big(- s_2^2 s_1 + s_1^3\big) + b\varepsilon^3\big(s_2^2 s_1 - s_1^3\big) - b\varepsilon s_1$

$z_{25} = 0$

$z_{31} = 0$

$z_{32} = 0$

$z_{33} = 1$

$z_{34} = -4b\varepsilon^3 s_3 s_2 s_1$

$z_{35} = b\varepsilon^2\big(- s_2^2 + s_1^2\big)$

2.28.1 Normal form shows drift from the fast waves

Finds that any fast waves will generate a mean drift effect on the slow dynamics (in the $s_3 \approx u_3$ equation), an effect quadratic in amplitude of the fast waves.

```
if thecase=lorenz86norm then begin
factor b;
ff_:=tp mat((-u2*u3+b*u2*u5
     ,u1*u3-b*u1*u5
     ,-u1*u2
     ,-u5
     ,+u4+b*u1*u2));
evalm_:=mat((0,0,0,i,-i));
ee_:=zz_:=tp mat((1,0,0,0,0),(0,1,0,0,0),(0,0,1,0,0)
     ,(0,0,0,1,-i),(0,0,0,1,+i));
toosmall:=4;
end;
```

2.29 Check the dimensionality of specified system

Extract dimension information from the specification of the dynamical system: seek mD centre manifold of an nD system.

```
if thecase=myweb then begin
  out "cmsyso.txt"$
  ODE_function:=ff_;
  subspace_eigenvalues:=evalm_;
  subspace_eigenvectors:=ee_;
  adjoint_eigenvectors:=zz_;
end;
```

```
write "total no. of modes  ",
n:=part(length(ee_),1);
write "no. of manifold modes ",
m:=part(length(ee_),2);
```

```
if trace_ then
write dims:={length(evalm_),length(zz_),length(ee_),length(ff_)}
if {length(evalm_),length(zz_),length(ee_),length(ff_)}
  ={{1,m},{n,m},{n,m},{n,1}}
  then write "Input dimensions are OK"
  else <<write "INCONSISTENT INPUT DIMENSIONS, I QUIT";
      if trace_ then rederr("WELL ALMOST") else quit>>;
```

Need an $m \times m$ identity matrix for normalisation of the isochron projection.

```
eyem_:=for j:=1:m sum e_(j,j)$
```

3 Dissect the linear part

Define exponential `exp(u)` $= e^u$. Do not (yet) invoke the simplification of `exp(0)` as I want it to label modes of no oscillation/growth, zero frequency.

```
clear exp;
operator exp;
let { df(exp(~u),t) => df(u,t)*exp(u)
    , exp(~u)*exp(~v) => exp(u+v)
    , exp(~u)^~p => exp(p*u)
    };
```

Need function `conj_` to do parsimonious complex conjugation.

```
procedure conj_(a)$
    ((a where {i=>i__}) where {i__=>-i})$
```

Make an array of eigenvalues for simplicity (instead of a matrix).

```
array evl_(m);
for j:=1:m do evl_(j):=evalm_(1,j);
```

Decide the presumed nature of the invariant manifold from an "or" of the eigenvalues. To cater for complex eigenvalues, temporarily turn on complex. To cater for possibly variable eigenvalues, insert a pre-check that the eigenvalues are simple numbers: this pre-check potentially wrecks the naming of the invariant manifold because the code assumes the nature determined by the numerical eigenvalues only.

```
on complex;
slowM_:=centreM_:=stableM_:=unstabM_:=0$
for j:=1:m do begin
    slowM_:=if numberp(evl_(j)) and
      evl_(j)=0 then 1 else slowM_;
    centreM_:=if numberp(evl_(j)) and
      repart(evl_(j))=0 and evl_(j) neq 0 then 1 else centreM_;
    stableM_:=if numberp(evl_(j)) and
      repart(evl_(j))<0 then 1 else stableM_;
    unstabM_:=if numberp(evl_(j)) and
      repart(evl_(j))>0 then 1 else unstabM_;
end;
natureMan_:=part({"invariant","Slow","Centre","Centre"
    ,"Stable","Slow-stable","Centre-stable","Centre-stable"
    ,"Unstable","Slow-unstable","Centre-unstable","Centre-unstabl
    ,"Invariant","Invariant","Fast","Invariant"
    },1+slowM_+2*(centreM_+2*(stableM_+2*unstabM_)));
off complex;
```

3.1 Normalise the adjoint eigenvectors

When we include delay differential equations, then we need to account for the history of the eigenvector as well. Hence multiply each eigenvector by its oscillating factor, $e^{i\omega t}$, and then take the mean. This multiplication by its oscillating factor should not make any difference for non-delay equations by the natural orthogonality of left and right eigenvectors of different eigenvalues. Reduce implements `conj` via `repart` and `impart`, so let `repart` do the conjugation of the cis factors.

Note: the 'left eigenvectors' have to be the eigenvectors of the complex conjugate transpose, and for the complex conjugate frequency. This seems best: for example, when the linear operator is $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ then the adjoint and the right eigenvectors are the same.

For un/stable manifolds we have to cope with complex frequencies. Seems to need zz_ to have minus?? complex conjugated frequency so store in cexp_—which is the same as dexp_ for real frequencies?? Need to decide on the inner product, especially to cater for the case of DDEs??

```
operator cis;
matrix aa_(m,m),dexp_(m,m),cexp_(m,m);
for j:=1:m do dexp_(j,j):=exp(evl_(j)*t);
for j:=1:m do cexp_(j,j):=exp(-conj_(evl_(j))*t);
aa_:=(tp map(conj_(~b),ee_*dexp_)*zz_*cexp_ );
write "Normalising the left-eigenvectors:";
aa_:=(aa_ where {exp(0)=>1, exp(~a)=>0 when a neq 0});
if det(aa_)=0 then << write
    "ORTHOGONALITY ERROR IN EIGENVECTORS; I QUIT"; quit>>;
zz_:=zz_*aa_^(-1);
```

3.2 Operator to represent delays

Introduce an operator to represent delay factors more conveniently for analysis.

What do we do about cos(a) and sin(a) in the following?? for general eigenvalues??

```
operator d_; linear d_;
let { d_(~a^~p,t,~dt)=>d_(a,t,dt)^p
    , d_(~a*~b,t,~dt)=>d_(a,t,dt)*d_(b,t,dt)
    , d_(cis(~a),t,~dt)=>cis(a)
        *sub(t=-dt,cos(a)+i*sin(a))
    , df(d_(~a,t,~dt),~b)=>d_(df(a,b),t,dt)
    , d_(~a,t,0)=>a
    , d_(d_(~a,t,~dta),t,~dtb)=>d_(a,t,dta+dtb)
    };
```

Now rewrite the (delay) factors in terms of this operator. Need to say that the symbol u depends upon time; later we write things into u and this dependence would be forgotten.

Create synonyms for as many variables as necessary, uk:=u(k), so that it is easier for people to enter ODEs.

```
for k:=1:n do set(mkid(u,k),u(k));
```

```
depend u,t;
ff_:=(ff_ where {u(~k,~dt)=>d_(u(k),t,dt)})$
```

3.3 Linearise at the origin

Assume the equilibrium is at the origin. Find the linear operator at the equilibrium. Include small=0 as we notionally adjoin it in the list of variables. Do not need to here make small any non-zero forcing at the equilibrium as it gets multiplied by small later??

```
matrix ll_(n,n);
for j:=1:n do for k:=1:n do begin
  ll_(j,k):=df(ff_(j,1),u(k));
  ll_(j,k):=(ll_(j,k) where {small=>0,u(~l)=>0});
end;
write "Find the linear operator is";
ll_:=ll_;
```

We need a vector of unknowns for a little while: only used once.

```
matrix uvec(n,1);
for j:=1:n do uvec(j,1):=u(j);
```

3.4 Eigen-check

Variable aa_ appears here as the diagonal matrix of frequencies. Check that the frequencies and eigenvectors are specified correctly.

Again need to worry about delays??

```
write "Check ",natureMan_," subspace linearisation ";
for j:=1:m do for k:=1:m do aa_(j,k):=0;
for j:=1:m do aa_(j,j):=evl_(j);
write %temporary write
reslin:=(ll_*(ee_*dexp_)-(ee_*dexp_)*aa_
    where cis(~a)*d_(1,t,~dt)=>sub(t=-dt,cos(a)+i*sin(a))*cis(a)
ok_:=1$
for j:=1:n do for k:=1:m do
    ok_:=if reslin(j,k)=0 then ok_ else 0$
if ok_ then write "Linearisation is OK";
```

Try to find a correction of the linear operator that is 'close'. Multiply by the adjoint eigenvectors and then average over time: operator $\mathcal{L}_{\text{new}} := \mathcal{L} - \mathcal{L}_{\text{adj}}$ should now have zero residual. Lastly, correspondingly adjust the ODEs, since `lladj` does not involve delays we do not need delay operator transforms in the product.

Again delays??

```
if not ok_ then for iter:=1:2 do begin
write "WARNING: I NEED TO ADJUST LINEAR OPERATOR";
write
lladj:=reslin*tp map(conj_(~b),zz_*cexp_);
write
lladj:=(lladj where {exp(0)=>1, exp(~a)=>0 when a neq 0});
write
ll_:=ll_-lladj;
write
reslin:=(ll_*(ee_*dexp_)-(ee_*dexp_)*aa_
    where exp(~a)*d_(1,t,~dt)=>sub(t=-dt,cos(a)+i*sin(a))*cis(a)
%for j:=1:n do for k:=1:m do
%    if reslin(j,k) neq 0 then << write
%    "OOPS, INCONSISTENT EIGENVALUES, EIGENVECTORS AND OPERATOR.
%    EMAIL ME; I QUIT"; write reslin:=reslin; rederr "aaaaah";qu
ok_:=1$
for j:=1:n do for k:=1:m do
```

```
    ok_:=if reslin(j,k)=0 then ok_ else 0$
if ok_ then iter:=iter+1000;
end;
if not ok_ then << write
    "OOPS, INCONSISTENT EIGENVALUES, EIGENVECTORS AND OPERATOR.
    EMAIL ME; I QUIT"; rederr "aaaaah";quit >>;
```

3.5 Ameliorate the nonlinearity

Anything not in the linear operator gets multiplied by `small` to be treated as small in the analysis. The feature of the second alternative is that when a user invokes `small` then the power of smallness is not then changed; however, causes issues in the relative scaling of some terms, so restore to the original version.

This might need reconsidering ?? but the if always chooses the first simple alternative.

```
somerules:=for j:=1:n collect
  (d_(1,t,~dt)*u(j)=d_(u(j),t,dt))$
ff_:=(if 1 then small*ff_
    else ff_-(1-small)*sub(small=0,ff_)) +(1-small)
    *(ll_*uvec where somerules)$
```

Any constant term in the equations `ff_` has to be multiplied by `exp(0)`.

```
ff_:=ff_+(exp(0)-1)*(ff_ where {small=>0,u(~l)=>0})$
```

From the matrix versions of the equations, create algebraic form using the matrix basis.

```
rhsfn_:=for i:=1:n sum e_(i,1)*ff_(i,1)$
```

Also, create the algebraic form of the jacobian transpose using the matrix basis: take the conjugate later when used.

```
rhsjact_:=for i:=1:n sum for j:=1:n sum
    e_(j,i)*df(ff_(i,1),u(j))$
```

3.6 Store invariant manifold frequencies

Extract all the frequencies in the invariant manifold, and the set of all the corresponding modes in the invariant manifold variables. The slow modes have zero frequency. Remember the frequency set is not in the 'correct' order. Array `modes` stores the set of indices of all the modes of a given frequency.

```
array evals(m),modes(m);
neval:=0$ evalset:={}$
for j:=1:m do if not(evl_(j) member evalset) then begin
  neval:=neval+1;
  evals(neval):=evl_(j);
  evalset:=evl_(j).evalset;
  modes(neval):=for k:=j:m join
    if evl_(j)=evl_(k) then {k} else {};
end;
```

Set a flag for the case of a slow manifold when all frequencies are zero, as then we compute the isochron projection. The next challenge is to get this isochron code working for the case of non-slow invariant manifolds.

```
itisSlowMan_:=if evalset={0} then 1 else 0$
if trace_ then write itisSlowMan_:=itisSlowMan_;
```

Put in the non-singular general case as the zero entry of the arrays.

```
evals(0):=geneval$
modes(0):={}$
```

3.7 Precompute matrices for updates

Precompute matrices to solve for updates for each of the critical frequencies, and the general case `k` $= 0$. The matrix

$$\texttt{llzz} = \begin{bmatrix} -\mathcal{L} + \partial_t & \mathcal{E}_0 \\ \mathcal{Z}_0^\dagger & 0 \end{bmatrix}$$

and then put its inverse in place. Subsequently, extract the blocks for the generalised inverses and solvability condition into `linvs` and `ginvs`.

```
matrix llzz(n+m,n+m);
array linvs(neval),ginvs(neval);
array l1invs(neval),g1invs(neval),l2invs(neval),g2invs(neval);
operator sp_; linear sp_;
for k:=0:neval do begin
```

Code the operator $\mathcal{L}\hat{v}$ where the delay is to only act on the oscillation part.

Again, what do we do about cos() and sin() of delays??

```
  for ii:=1:n do for jj:=1:n do llzz(ii,jj):=(
      -sub(small=0,ll_(ii,jj))
      where d_(1,t,~dt)=>cos(freqs(k)*dt)-i*sin(freqs(k)*dt));
```

Code the operator $\partial\hat{v}/\partial t$ where it only acts on the oscillation part.

```
  for j:=1:n do llzz(j,j):=evals(k)+llzz(j,j);
```

Now code the part leading to the solvability condition which arises from allowing the (oscillation) amplitude to evolve. Use operator `sp_` to extract the delay parts that subtly affect the updates of the evolution.

Again cos() and sin() here??

```
  for j:=1:length(modes(k)) do
    for ii:=1:n do llzz(ii,n+j):=ee_(ii,part(modes(k),j))
     +(for jj:=1:n sum
       sp_(ll_(ii,jj)*ee_(jj,part(modes(k),j)),d_)
       where { sp_(1,d_)=>0
             , sp_(d_(1,t,~dt),d_)=>dt*(
               cos(freqs(k)*dt)-i*sin(freqs(k)*dt))
             });
```

Force the updates to be orthogonal to the left-eigenvectors in the complex conjugate transpose adjoint.,

```
  for i:=1:length(modes(k)) do
```

```
    for j:=1:n do llzz(n+i,j):=conj_(zz_(j,part(modes(k),i)));
```

Set the bottom-right corner of the matrix to zero.

```
  for i:=1:length(modes(k)) do
    for j:=1:m do llzz(n+i,n+j):=0;
```

Add some trivial rows and columns to make the matrix up to the same size for all frequencies.

```
  for i:=length(modes(k))+1:m do begin
    for j:=1:n+i-1 do llzz(n+i,j):=llzz(j,n+i):=0;
    llzz(n+i,n+i):=1;
  end;
```

Invert the matrix and unpack into arrays ready for use by the inversion operators.

```
if trace_ then write llzz:=llzz;
  llzz:=llzz^(-1);
if trace_ then write llzz:=llzz;
  linvs(k):=for i:=1:n sum for j:=1:n sum e_(i,j)*llzz(i,j);
  ginvs(k):=for i:=1:length(modes(k)) sum
    for j:=1:n sum e_(part(modes(k),i),j)*llzz(i+n,j);
```

Unpack the conjugate transpose for inverse operators used for the isochrons. A difference here is that the orthogonality condition is non-trivial (in the slow manifold we assumed amplitudes were exactly orthogonal to the left-eigenvectors), so we need to remember more parts of the inverse of the matrix. Will it need to be more subtle for the un/stable manifolds case??

```
%  realgeneval:={repart(geneval)=>geneval, impart(geneval)=>0}$
  l1invs(k):=for i:=1:n sum for j:=1:n sum
      e_(i,j)*conj_(llzz(j,i));
  l2invs(k):=for i:=1:n sum for j:=1:length(modes(k)) sum
      e_(i,part(modes(k),j))*conj_(llzz(j+n,i));
  g1invs(k):=for i:=1:length(modes(k)) sum for j:=1:n sum
      e_(part(modes(k),i),j)*(llzz(j,i+n)); %conj_??
  g2invs(k):=
```

```
    for i:=1:length(modes(k)) sum for j:=1:length(modes(k)) sum
      e_(part(modes(k),i),part(modes(k),j))*conj_(llzz(j+n,i+n))
end;
```

3.8 Define operators that invoke these inverses

Decompose residuals into parts, and operate on each. First for the invariant manifold. But making `e_` non-commutative means that it does not get factored out of these linear operators: must post-multiply by `e_` because the linear inverse is a premultiply.

```
operator linv; linear linv;
let linv(e_(~j,~k)*exp(~a),exp)=>linvproc(a/t)*e_(j,k);
procedure linvproc(a);
  if a member evalset
  then << k:=0;
    repeat k:=k+1 until a=evals(k);
    linvs(k)*exp(a*t) >>
  else sub(geneval=a,linvs(0))*exp(a*t)$
```

Second for the evolution on the invariant manifold.

```
operator ginv; linear ginv;
let ginv(e_(~j,~k)*exp(~a),exp)=>ginvproc(a/t)*e_(j,k);
procedure ginvproc(a);
  if a member evalset
  then << k:=0;
    repeat k:=k+1 until a=evals(k);
    ginvs(k) >>
  else sub(geneval=a,ginvs(0))$
```

Copy and adjust the above for the projection. But first define the generic procedure. Perhaps use conjugate/negative of the frequency when applying to the general case of oscillations—but it might already have been accounted for??

```
procedure invproc(a,invs);
```

```
  if a member evalset
  then << k:=0;
    repeat k:=k+1 until a=evals(k);
    invs(k)*exp(a*t) >>
  else sub(geneval=a,invs(0))*exp(a*t)$
```

Then define operators that we use to update the projection.

```
operator l1inv; linear l1inv;
operator l2inv; linear l2inv;
operator g1inv; linear g1inv;
operator g2inv; linear g2inv;
let { l1inv(e_(~j,~k)*exp(~a),exp)=>invproc(a/t,l1invs)*e_(j,k)
    , l2inv(e_(~j,~k)*exp(~a),exp)=>invproc(a/t,l2invs)*e_(j,k)
    , g1inv(e_(~j,~k)*exp(~a),exp)=>invproc(a/t,g1invs)*e_(j,k)
    , g2inv(e_(~j,~k)*exp(~a),exp)=>invproc(a/t,g2invs)*e_(j,k)
    };
```

This section writes to various files so the output to `cmsyso.txt` must be redone afterwards.

4 Initialise LaTeX output

This section writes to various files so the output to `cmsyso.txt` must be redone afterwards.

First define how various tokens get printed.

```
load_package rlfi;
%deflist('((!( !{!\!b!i!g!() (!) !\!b!i!g!)!}) (!P!I !\!p!i! )
%         (!p!i !\!p!i! ) (!E !e) (!I !i) (e !e) (i !i)),'name)$
deflist('((!( !\!l!P!a!r! ) (!) !\!r!P!a!r)
    (!P!I !\!p!i! ) (!p!i !\!p!i! ) (!E !e) (!I !i)
    (e !e) (i !i)),'name)$
```

Force all fractions (coded in Reduce as `quotient`) to use `\frac` command so we can change how it appears.

```
put('quotient,'laprifn,'prinfrac);
```

Override the procedure that prints annoying messages about multicharacter symbols. It ends the output of one expression. This is just a copy from `rlfi.red` with the appropriate if-statement deleted. While interfering, hardcode that the mathematics is in inline mode.

```
%write "Ignore immediately following messages";
symbolic procedure prinlaend;
<<terpri();
  prin2t "\)\par";
  if !*verbatim then
      <<prin2t "\begin{verbatim}";
        prin2t "REDUCE Input:">>;
  ncharspr!*:=0;
  if ofl!* then linelength(car linel!*)
    else laline!*:=cdr linel!*;
  nochar!*:=append(nochar!*,nochar1!*);
  nochar1!*:=nil >>$
  %
```

Similarly, hardcode at the beginning of expression output that the mathematics is in inline mode.

```
symbolic procedure prinlabegin;
% Initializes the output
<<if !*verbatim then
      <<terpri();
        prin2t "\end{verbatim}">>;
  linel!*:=linelength nil . laline!*;
  if ofl!* then linelength(laline!* + 2)
    else laline!*:=car linel!* - 2;
  prin2 "\(" >>$
```

Override the procedure that outputs the LaTeX preamble upon the command `on latex`. Presumably modified from that in `rlfi.red`. Use it to write a decent header that we use for one master file.

In the following, not clear that we should simply omit parentheses with the exp function. Could do something cleverer with `\lPar` and `\rPar` such as have a counter and cycle through the alternatives depending upon the counter.

```
symbolic procedure latexon;
<<!*!*a2sfn:='texaeval;
  !*raise:=nil;
  prin2t "\documentclass[11pt,a5paper]{article}";
  prin2t "\usepackage[a5paper,margin=13mm]{geometry}";
  prin2t "\usepackage{parskip,time} \raggedright";
  prin2t "\def\lPar{\mathchoice{\big(}{\big(}{(}{(}}";
  prin2t "\def\rPar{\mathchoice{\big)}{\big)}{)}{)}}";
  prin2t "\let\FRaC\frac";
  prin2t "\renewcommand{\frac}[2]{\mathchoice%";
  prin2t "    {\FRaC{#1}{#2}}{\FRaC{#1}{#2}}{#1/#2}{#1/#2}}";
  prin2t "\def\exp{\,e}";
  prin2t "\def\eps{\varepsilon}";
  prin2t "\title{Invariant manifold of your dynamical system}";
  prin2t "\author{A. J. Roberts, University of Adelaide\\";
  prin2t "\texttt{http://www.maths.adelaide.edu.au/anthony.robert
  prin2t "\date{\now, \today}";
  prin2t "\begin{document}";
  prin2t "\maketitle";
  prin2t "Throughout and generally: the lowest order, most";
  prin2t "important, terms are near the end of each expression."
  prin2t "\input{centreManSys}";
  if !*verbatim then
      <<prin2t "\begin{verbatim}";
        prin2t "REDUCE Input:">>;
  put('tex,'rtypefn,'(lambda(x) 'tex)) >>$
```

Set the default output to be inline mathematics.

```
mathstyle math;
```

Define the Greek alphabet with `small` as well.

```
defid small,name="\eps";%varepsilon;
defid alpha,name=alpha;
defid beta,name=beta;
defid gamma,name=gamma;
defid delta,name=delta;
defid epsilon,name=epsilon;
defid varepsilon,name=varepsilon;
defid zeta,name=zeta;
defid eta,name=eta;
defid theta,name=theta;
defid vartheta,name=vartheta;
defid iota,name=iota;
defid kappa,name=kappa;
defid lambda,name=lambda;
defid mu,name=mu;
defid nu,name=nu;
defid xi,name=xi;
defid pi,name=pi;
defid varpi,name=varpi;
defid rho,name=rho;
defid varrho,name=varrho;
defid sigma,name=sigma;
defid varsigma,name=varsigma;
defid tau,name=tau;
defid upsilon,name=upsilon;
defid phi,name=phi;
defid varphi,name=varphi;
defid chi,name=chi;
defid psi,name=psi;
defid omega,name=omega;
defid Gamma,name=Gamma;
defid Delta,name=Delta;
defid Theta,name=Theta;
defid Lambda,name=Lambda;
defid Xi,name=Xi;
```

```
defid Pi,name=Pi;
defid Sigma,name=Sigma;
defid Upsilon,name=Upsilon;
defid Phi,name=Phi;
defid Psi,name=Psi;
defid Omega,name=Omega;

defindex e_(down,down);
defid e_,name="e";
defindex d_(arg,down,down);
defid d_,name="D";
defindex u(down);
%defid u1,name="u\sb1";
%defid u2,name="u\sb2";
%defid u3,name="u\sb3";
%defid u4,name="u\sb4";
%defid u5,name="u\sb5";
%defid u6,name="u\sb6";
%defid u7,name="u\sb7";
%defid u8,name="u\sb8";
%defid u9,name="u\sb9";
defindex s(down);
defindex exp(up);
defid exp,name="e"; %does not work??
```

Can we write the system? Not in matrices apparently. So define a dummy array `tmp` that we use to get the correct symbol typeset.

```
array tmp(n),tmps(m),tmpz(m);
defindex tmp(down);
defindex tmps(down);
defindex tmpz(down);
defid tmp,name="\dot u";
defid tmps,name="\vec e";
defid tmpz,name="\vec z";
rhs_:=rhsfn_$
```

```
for k:=1:m do tmps(k):={for j:=1:n collect ee_(j,k),exp(evl_(k)*t
for k:=1:m do tmpz(k):={for j:=1:n collect zz_(j,k),exp(evl_(k)*t
```

We have to be shifty here because `rlfi` does not work inside a loop: so write the commands to a file, and then input the file. The output line length of each 'write' statement must be short enough as otherwise Reduce puts in a line break.

```
out "scratchfile.red";
write "write ""\)
\paragraph{The specified dynamical system}
\("";";
for j:=1:n do write "tmp(",j,"):=coeffn(rhs_,e_(",j,",1),1);";
write "write ""\)
\paragraph{",natureMan_,"
subspace basis vectors}","
\("";";
for j:=1:m do write "tmps(",j,"):=tmps(",j,");";
for j:=1:m do write "tmpz(",j,"):=tmpz(",j,");";
write "end;";
shut "scratchfile.red";
```

Now print the dynamical system to the LaTeX sub-file.

```
on latex$
out "centreManSys.tex"$
in "scratchfile.red"$
shut "centreManSys.tex"$
off latex$
```

Finish the input.

```
end;
```

```
in_tex "latexinit2.tex"$
```

5 Linear approximation to the invariant manifold

But first, and if for the web, open the output file and write out the possibly adjusted nonlinear right-hand side function. According to the manual, this will append to the earlier output to the file.

```
if thecase=myweb then out "cmsyso.txt"$
```

```
write "Analyse ODE/DDE system du/dt = ",ff_;
```

Parametrise the invariant manifold in terms of these amplitudes.

```
operator s; depend s,t;
let df(s(~j),t)=>coeffn(gg_,e_(j,1),1);
```

Invoke the following procedure to substitute whatever the current approximation is into (nonlinear) expressions. ??

```
procedure manifold_;
    for j:=1:n collect u(j)=coeffn(uu_,e_(j,1),1)$
```

The linear approximation to the invariant manifold must be the following corresponding to the frequencies down the diagonal (even if zero). The amplitudes s_j are slowly evolving as they are either slow modes, or the complex amplitudes of oscillating modes.

```
uu_:=for j:=1:m sum s(j)*exp(evl_(j)*t)
  *(for k:=1:n sum e_(k,1)*ee_(k,j))$
gg_:=0$
```

For some temporary trace printing??

```
procedure matify(a,m,n)$
  begin matrix z(m,n);
    for i:=1:m do for j:=1:n do z(i,j):=coeffn(a,e_(i,j),1);
    return (z where {exp(0)=>1,small=>s});
    end$
```

For the isochron may need to do something different with frequencies, but this should work as the inner product is complex conjugate transpose. The pp_ matrix is proposed to place the projection residuals in the range of the isochron.

```
zs_:=for j:=1:m sum exp(evl_(j)*t)
  *(for k:=1:n sum e_(k,j)*zz_(k,j))$
pp_:=0$
```

6 Iteratively construct the invariant manifold

But first establish the Taylor series in any delay factors of slow amplitudes.

```
let d_(s(~k),t,~dt)=>s(k)+(for n:=1:toosmall sum
        (-dt)^n*df(s(k),t,n)/factorial(n));
```

Truncate expansions to specified order of error, and start the iteration.

```
for j:=toosmall:toosmall do let small^j=>0;
write "Start iterative construction of ",natureMan_," manifold";
for iter:=1:maxiter_ do begin
if trace_ then write "
ITERATION = ",iter,"
-------------";
```

Compute residual vector (matrix) of the dynamical system Roberts (1997).

```
resde_:=-df(uu_,t)+sub(manifold_(),rhsfn_);
if trace_ then write "resde_=",matify(resde_,n,1);
```

Get the local directions of the coordinate system on the curving manifold: store transpose as $m \times n$ matrix.

```
est_:=tpe_(for j:=1:m sum df(uu_,s(j))*e_(1,j),e_);
est_:=conj_(est_);
if trace_ then write "est_=",matify(est_,m,n);
```

Compute residual matrix for the isochron projection Roberts (1989, 2000). But only when the evalset is for slow manifolds: the reason is that there is no sensible concept of isochron for un/stable modes when in the presence of centre modes. [3] For example, consider the normal form system $\dot{X} = 0$ and $\dot{Y} = G(Y)Y$: it has solutions $Y(t) = Y_0 e^{G(X_0)t}$ and so for general G there are no curves $Y(X)$ which have the same rate of decay to the slow manifold; that is, there are no curves that 'collapse together'.

```
if itisSlowMan_ then begin
    jacadj_:=conj_(sub(manifold_(),rhsjact_));
if trace_ then write "jacadj_=",matify(jacadj_,n,n);
    resd_:=df(zs_,t)+jacadj_*zs_+zs_*pp_;
if trace_ then write "resd_=",matify(resd_,n,m);
```

Compute residual of the normalisation of the projection.

```
    resz_:=est_*zs_-eyem_*exp(0);
if trace_ then write "resz_=",matify(resz_,m,m);
end else resd_:=resz_:=0; % for when not slow manifold
```

Write lengths of residuals as a trace print (remember that the expression 0 has length one).

```
write lengthRes:=map(length(~a),{resde_,resd_,resz_});
```

Solve for updates all the hard work is already encoded in the operators.

```
uu_:=uu_+linv(resde_,exp);
gg_:=gg_+ginv(resde_,exp);
if trace_ then write "gg_=",matify(gg_,m,1);
if trace_ then write "uu_=",matify(uu_,n,1);
```

Now update the isochron projection, with normalisation.

```
if itisSlowMan_ then begin
zs_:=zs_+l1inv(resd_,exp)-l2inv(resz_,exp);
pp_:=pp_-g1inv(resd_,exp)+youshouldnotseethis*g2inv(resz_,exp);
if trace_ then write "zs_=",matify(zs_,n,m);
```

[3] Although there is a sensible concept of 'isochron' if there are no centre modes—justified by the Hartman–Grossman theorem which asserts topological equivalence to the local linearisation.

```
if trace_ then write "pp_=",matify(pp_,m,m);
end;
```

Terminate the loop once residuals are zero.

```
showtime;
if {resde_,resd_,resz_}={0,0,0} then write iter:=iter+10000;
end;
```

Only proceed to print if terminated successfully.

```
if {resde_,resd_,resz_}={0,0,0}
  then write "SUCCESS: converged to an expansion"
  else <<write "FAILED TO CONVERGE; I QUIT";
    if thecase=myweb then <<shut "cmsyso.txt";
    quit >> >>;
```

```
%write "Temporarily halt here";end;
```

7 Output text version of results

Once construction is finished, simplify `exp(0)`.

```
let exp(0)=>1;
```

Invoking switch `complex` improves some of the output of the complex numbers, but wrecks other parts of the output. Best left off.

Write text results.

```
write "The ",natureMan_," manifold is (to one order lower)";
for j:=1:n do write "u",j," = ",
  coeffn(small*uu_,e_(j,1),1)/small;
write "The evolution of the real/complex amplitudes";
for j:=1:m do write "ds(",j,")/dt = ",
  coeffn(gg_,e_(j,1),1);
```

Optionally write the projection vectors.

```
if itisSlowMan_ then begin
  write "The normals to the isochrons at the slow manifold.
Use these vectors: to project initial conditions
onto the slow manifold; to project non-autonomous
forcing onto the slow evolution; to predict the
consequences of modifying the original system; in
uncertainty quantification to quantify effects on
the model of uncertainties in the original system.";
  for j:=1:m do write "z",j," = ",
    for i:=1:n collect coeffn(zs_,e_(i,j),1);
end;
```

Write text results numerically evaluated when expressions are long.

```
if length(gg_)>30 then begin
on rounded; print_precision 4;
write "Numerically, the ",natureMan_," manifold is (to one order
for j:=1:n do write "u",j," = ",
  coeffn(small*uu_,e_(j,1),1)/small;
write "Numerically, the evolution of the real/complex amplitudes'
for j:=1:m do write "ds(",j,")/dt = ",
  coeffn(gg_,e_(j,1),1);
if itisSlowMan_ then begin
  write "Numerically, normals to isochrons at slow manifold.";
  for j:=1:m do write "z",j," = ",
    for i:=1:n collect coeffn(zs_,e_(i,j),1);
end;
off rounded;
end;
```

```
if thecase=myweb then shut "cmsyso.txt"$
```

There is an as yet unresolved problem in the typesetting when the argument of `exp` (eigenvalue) is a rational number instead of integer ??: the numerator has an extra pair of parentheses which then makes the typesetting wrong; maybe we need a pre-LaTeX filter??

8 Output LaTeX version of results

Change the printing of temporary arrays.

```
array tmpzz(m,n);
defid tmp,name="u";
defid tmps,name="\dot s";
defid tmpz,name="\vec z";
defid tmpzz,name="z";
defindex tmpzz(down,down);
```

Gather complicated result

```
%for k:=1:m do tmpz(k):=for j:=1:n collect (1*coeffn(zs_,e_(j,k),
for k:=1:m do for j:=1:n do tmpzz(k,j):=(1*coeffn(zs_,e_(j,k),1))
```

Write to a file the commands needed to write the LaTeX expressions. Write the invariant manifold to one order lower than computed. The output line length of each 'write' statement must be short enough as otherwise Reduce puts in a line break—its counting is a bit mysterious!

```
out "scratchfile.red";
write "write ""\)
\paragraph{The ",natureMan_,"
manifold}";
write "These give the location of the invariant manifold in
terms of parameters~\(s\sb j\).
\("";";
for j:=1:n do write "tmp(",j,
  "):=coeffn(small*uu_,e_(",j,",1),1)/small;";
```

Write the commands to write the ODEs on the centre manifold.

```
write "write ""\)
\paragraph{",natureMan_,"
manifold ODEs}";
write "The system evolves on the invariant manifold such
that the parameters evolve according to these ODEs.
```

```
\("";";
for j:=1:m do write "tmps(",j,"):=1*coeffn(gg_,e_(",j,",1),1);";
```

Optionally write the commands to write the projection vectors on the slow manifold.

```
if itisSlowMan_ then begin
  write "write ""\)
\paragraph{Normals to isochrons at the slow manifold}
Use these vectors: to project initial conditions
onto the slow manifold; to project non-autonomous
forcing onto the slow evolution; to predict the
consequences of modifying the original system; in
uncertainty quantification to quantify effects on
the model of uncertainties in the original system.
The normal vector \(\vec z\sb j:=(z\sb{j1},\ldots,z\sb{jn})\)
\("";";
  for i:=1:m do for j:=1:n do
  write "tmpzz(",i,",",j,"):=tmpzz(",i,",",j,");";
end;
```

Finish the scratchfile.

```
write "end;";
shut "scratchfile.red";
```

Execute the file with the required commands, with output to the main centre manifold LaTeX file.

```
out "centreMan.tex"$
on latex$
in "scratchfile.red"$
off latex$
shut "centreMan.tex"$
```

```
end;
```

```
in_tex "latexout2.tex"$
```

9 Fin

That's all folks.

```
write "Finished constructing ",natureMan_," manifold of ODE/DDE"
```

```
if thecase=myweb then begin
quit;
end;
```

```
%end;%loop over cases--not working
end;
```

References

Erneux, T. (2009), *Applied Delay Differential equations*, Vol. 3 of *Surveys and Tutorials in the Applied Mathematical Sciences*, Springer.

Ion, A. V. & Georgescu, R. M. (2013), 'Bautin bifurcation in a delay differential equation modeling leukemia', *Nonlinear Analysis, Theory, Methods & Applications* **82**, 142–157.
http://arXiv.org/abs/1208.1707

Renson, L., Deliege, G. & Kerschen, G. (2012), Finite element computation of nonlinear normal modes of nonconservative systems, *in* 'Proceedings of the ISMA 2012 conference'.
http://hdl.handle.net/2268/129189

Roberts, A. J. (1989), 'Appropriate initial conditions for asymptotic descriptions of the long term evolution of dynamical systems', *J. Austral. Math. Soc. B* **31**, 48–75.

Roberts, A. J. (1997), 'Low-dimensional modelling of dynamics via computer algebra', *Computer Phys. Comm.* **100**, 215–230. doi:10.1016/S0010-4655(96)00162-2.

Roberts, A. J. (2000), 'Computer algebra derives correct initial conditions for low-dimensional dynamical models', *Computer Phys. Comm.* **126**(3), 187–206.

Stoleriu, I. (2012), Periodic orbits of a pair of coupled oscillators near resonance, Technical report, University of Iasi. http://www.math.uaic.ro/~ITN2012/files/talk/Stoleriu.pdf