

Computer algebra derives normal forms of general stochastic and non-autonomous differential equations

A. J. Roberts*

November 19, 2008; modified June 27, 2017

Abstract

Modelling stochastic systems has many important applications. Stochastic coordinate transforms to a normal form is a powerful way of disentangling emergent long term dynamics. Since the analysis involves classic calculus, then the approach also applies to a wide class of non-autonomous dynamical systems. Further, cater for deterministic autonomous systems by simply omitting the time dependence in the system. For generality, this approach now caters for unstable modes, and for differential equation systems with a rational right-hand side. Use this code via the website¹.

Contents

1	Introduction	3
2	Choose the SDEs	4
2.1	Simple pair of SDEs	5
2.2	Simple rational ODEs	7
2.3	Future noise in the transform	8

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

¹<http://www.maths.adelaide.edu.au/anthony.roberts/sdenf.php>

2.4	Other methodologies fail	9
2.5	Levy area contraction: off-diagonal example	10
2.6	Position-momentum: the Jordan form	11
2.7	Radek’s slow oscillation with fast noise	12
2.8	Simple hyperbolic system	13
2.9	Duan’s hyperbolic system for foliation	14
2.10	A four mode hyperbolic system	15
2.11	Monahan’s four examples	16
2.11.1	Example four: ‘three’ time scales	16
2.11.2	Example one: simple nonlinear	17
2.11.3	Example three: many fast modes	19
2.11.4	Example two: irregular slow manifold	21
2.11.5	Idealised Stommel-like model of meridional overturning circulation	23
2.12	Majda’s triad models	24
2.12.1	Multiplicative triad model	24
2.12.2	Additive triad model	26
2.13	Potzsche and Rasmussen deterministic non-autonomous ex- amples	28
2.13.1	Lorenz near the pitchfork bifurcation	28
2.13.2	Fluctuating kdV example	30
2.14	Local analysis of heat exchanger	32
2.14.1	Near the boundary	33
2.14.2	Heat exchanger with quadratic reaction	34
2.15	Michaelis–Menten–Henri deterministic model	36
3	General SDE preliminaries	37
3.1	Extract and scale slow equations	37
3.2	Extract and scale stable fast equations	38
3.3	Extract and scale unstable fast equations	40
4	Setup LaTeX output using rlf	42
5	Delayed write of text info	47
6	Represent the noise	47

1	<i>Introduction</i>	3
7	Solve homological equation with noise	49
8	Initialise approximate transform	51
9	Iterative updates	52
9.1	Fast stable modes	52
9.2	Fast unstable modes	53
9.3	Slow modes	54
10	Post-processing	54
10.1	Plain text output	55
10.2	L ^A T _E X output	56

1 Introduction

Construct stochastic normal form of a wide class of non-autonomous or stochastic differential equations (SDEs), based upon earlier research (Cox & Roberts 1991, Chao & Roberts 1996, Roberts 2008). Interpret all SDEs in the Stratonovich sense so the analysis applies to deterministic differential equations, both non-autonomous and autonomous. To construct normal forms of deterministic autonomous differential equations, simply omit specifying any noise. This article documents code designed for an interactive web site (Roberts 2009) that is available to all to use.

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

```

1 %% Execute in reduce with in_tex "cadnfgsde.tex"$
2 %% See cadnfgsde.pdf for detailed explanation.
3 thecase:=sdemmh;
4 thecases:={sdesimple, oderat, sdemulti, sdebif, sdelac,
5 sdemom, sderadek, sdehyper, sdeduan, sdehifour, sdeMona1,
6 sdeMona2, sdeMona3, sdeMona4, sdeMonaSS, sdeMajda3m,
7 sdeMajda3a, sdePRLorenz, sdehe, sdehebc, sdeheqr, sdemmh,
```

```
8 sdePRKdV }$
```

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. The code cannot handle any higher order in noise amplitude `sigma`.

```
9 maxiter_:=29$
10 factor small;
11 toosmall:=3$
12 let sigma^3=>0;
```

2 Choose the SDEs

For an SDE with m slow modes, n_y fast stable modes (quickly decaying), and/or n_z fast unstable modes (quickly growing), you must denote the slow modes by `x(1)` through to `x(m)`, the stable fast modes by `y(1)` through to `y(ny)`, and the unstable fast modes by `z(1)` through to `z(nz)`. Each Stratonovich white noise, derivative of a Stratonovich Wiener process, must be denoted by `w(.)` where the dot denotes almost any label you care to choose: simple numbers such as `w(1)` and/or `w(2)` are the usual choices; but other labels for the noise can be used. The SDEs must be linearly diagonalised.²

Load the reduce to L^AT_EX package so we can also generate a nicer version of the output via L^AT_EX. Load now so that a canny definition of the SDEs can invoke some of the options.

```
13 load_package rlfi;
```

First, define the operators to be used in the specification of the SDEs.

```
14 operator x;
15 operator y;
16 operator z;
17 operator w;
```

²Although a Jordan form is also acceptable, there are issues in the error control.

Cater for rational function SDEs by allowing time dependence in these variables at specification. Then users must multiply each SDE by a common denominator, and put on the right-hand side the nonlinear terms involving the time derivative: see examples in sections 2.2 and 2.11.2.

```
18 depend x,t;
19 depend y,t;
20 depend z,t;
```

Use the RHS part of an SDE to specify an SDE. Form the RHSSs into lists to specify the system of SDEs. Set trivial defaults for the SDEs in case I forget.

```
21 xrhs:=yrhs:=zrhs:={} $
```

In the case `webpage`, get the SDEs from the file `sdeb.red` which is where the web script writes a user specified system.

```
22 if thecase=webpage then in "sdeb.red" $
```

2.1 Simple pair of SDEs

A classically simple pair of fast/slow SDEs is

$$\dot{x} = -xy \quad \text{and} \quad \dot{y} = -y + x^2 - 2y^2 + \sigma w(t), \quad (1)$$

where lowercase $w(t)$ denotes the formal derivative dW/dt of a Stratonovich Wiener process $W(t, \omega)$. Parameter σ controls the strength of the noise. Use `x(1)` to denote variable x , `y(1)` to denote variable y , and `w(1)` to denote Stratonovich noise w . Alternatively, $w(t)$ could denote some non-autonomous deterministic forcing, control or other extrinsic input.

Specify the slow x SDE by allocating the one element list of its right-hand side to the variable `xrhs`.

```
23 if thecase=sdesimple then begin
24   xrhs:={-x(1)*y(1)};
```

Specify the nonlinear and noise terms of the y SDE as the one element list assigned to `yrhs`.

```
25 yrhs:={-y(1)+x(1)^2-2*y(1)^2+w(1)};
```

There are no unstable modes.

```
26 zrhs:={};
```

The code automatically multiplies the noise factors by a parameter **sigma** so there is no need include the parameter σ in the specification of the problem (unless you particularly want to), as it will be done for you. The code uses the parameter **small** to control truncation in nonlinearity; just ignore parameter **small** wherever it appears.

```
27 factor small,sigma;
28 end;
```

The stochastic coordinate transform

$$y_1 = Y_1 + X_1^2 + 2Y_1^2 + \sigma e^{-t} w_1 (1 + 4Y_1)$$

$$x_1 = X_1 + X_1 Y_1 + \sigma e^{-t} w_1 X_1$$

Result normal form SDEs

$$\dot{Y}_1 = -Y_1 - 2X_1^2 Y_1 - 4\sigma w_1 Y_1 + 8\sigma^2 w_1 e^{-t} w_1 Y_1$$

$$\dot{X}_1 = -X_1^3 - \sigma w_1 X_1 + 2\sigma^2 w_1 e^{-t} w_1 X_1$$

Easy stochastic bifurcation For example, modifying the x equation to $\dot{x} = ax - xy$ induces a stochastic bifurcation as parameter a crosses zero. Construct the stochastic normal form for this by simply changing to

```
29 xrhs:={small*a*x(1)-x(1)*y(1)};
```

The extra factor of **small** causes only low powers of parameter a to be retained, as appropriate for such a bifurcation. Some may like to consider it analogous to scaling a with ϵ^2 and scaling x with ϵ .

2.2 Simple rational ODEs

A simple system of fast/slow ODEs in rational functions is

$$\dot{x} = -\frac{xy}{1+z}, \quad \dot{y} = -\frac{y}{1+2y} + x^2, \quad \dot{z} = 2\frac{z}{1+3x}. \quad (2)$$

Use $\mathbf{x}(1)$ to denote variable x , $\mathbf{y}(1)$ to denote variable y , and $\mathbf{z}(1)$ to denote z . Multiply each ODE by the denominator for the ODE and shift the nonlinear d/dt terms to the right-hand side.

```
30 if thecase=oderat then begin
31  xrhs:={-x(1)*y(1)-z(1)*df(x(1),t)};
32  yrhs:={-y(1)+x(1)^2*(1+2*y(1))-2*y(1)*df(y(1),t)};
33  zrhs:={2*z(1)-3*x(1)*df(z(1),t)};
```

Truncate to one higher order because it is a simple system.

```
34 toosmall:=4;
35 end;
```

Time dependent coordinate transform

$$\begin{aligned} z_1 &= 6X_1Y_1Z_1\varepsilon^2 + Z_1 \\ y_1 &= 2X_1^4\varepsilon^2 - 4X_1^2Y_1^2\varepsilon^2 + X_1^2\varepsilon + 6Y_1^3\varepsilon^2 - 2Y_1^2\varepsilon + Y_1 \\ x_1 &= 2X_1^3Y_1\varepsilon^2 - 1/2X_1Y_1^2\varepsilon^2 + X_1Y_1Z_1\varepsilon^2 + X_1Y_1\varepsilon + X_1 \end{aligned}$$

Result normal form DEs For example, from this form we see the slow manifold is $Y_1 = Z_1 = 0$.

$$\begin{aligned} \dot{Z}_1 &= -54X_1^3Z_1\varepsilon^3 + 18X_1^2Z_1\varepsilon^2 - 6X_1Z_1\varepsilon + 2Z_1 \\ \dot{Y}_1 &= 8X_1^4Y_1\varepsilon^3 + 4X_1^2Y_1\varepsilon^2 + 2X_1^2Y_1\varepsilon - Y_1 \\ \dot{X}_1 &= -2X_1^5\varepsilon^3 - X_1^3\varepsilon^2 - 2X_1Y_1^2Z_1\varepsilon^3 \end{aligned}$$

2.3 Future noise in the transform

An interesting pair of fast/slow SDEs derived from stochastic advection/dispersion is

$$\dot{x} = -\sigma y w(t) \quad \text{and} \quad \dot{y} = -y + \sigma x w(t), \quad (3)$$

where lowercase $w(t)$ denotes the formal derivative dW/dt of a Stratonovich Wiener process $W(t, \omega)$. Parameter σ controls the strength of the noise. In stochastic advection/dispersion parameter σ represents the lateral wavenumber of the concentration profile.

Use `x(1)` to denote variable x , `y(1)` to denote variable y , and `w(1)` to denote Stratonovich noise w . Specify the slow x SDE by allocating the one element list of its right-hand side to the variable `xrhs`. Specify the nonlinear and noise terms of the y SDE as the one element list assigned to `yrhs`.

```
36 if thecase=sdemulti then begin
37   factor small,sigma;
38   xrhs:={-y(1)*w(1)};
39   yrhs:={-y(1)+x(1)*w(1)};
40   zrhs:={};
41 end;
```

The code automatically multiplies the noise factors by a parameter `sigma` so there is no need include the parameter σ in the specification of the problem (unless you particularly want to), as it will be done for you.

Via the coordinate transform

$$x \approx X + \sigma Y e^t \star w \quad \text{and} \quad y \approx Y + \sigma X e^{-t} \star w,$$

the resultant normal form is

$$\dot{X} \approx -\sigma^2 X w e^{-t} \star w \quad \text{and} \quad \dot{Y} \approx -Y + \sigma^2 Y w e^t \star w.$$

One of the interesting aspects of this example is the quickness with which we could go to higher order noise interactions, higher orders in σ . However, I do not compute such higher order terms in this code.

2.4 Other methodologies fail

Consider for small parameter ϵ

$$\text{slow mode } \dot{x} = \epsilon x + x^3 - (1 - \sigma w)xy, \quad (4)$$

$$\text{fast mode } \dot{y} = -y + x^2 + y^2 + \sigma y w. \quad (5)$$

Deterministically, there is a bifurcation to two equilibria for small $\epsilon > 0$. The noise w affects this bifurcation somehow.

Why is this tricky? Cross-sectional averaging is simply projection onto the slow space $y = 0$ which predicts instability of subcritical bifurcation $\dot{x} = \epsilon x + x^3$. Whereas adiabatic approximation, singular perturbation and multiple scales set $\dot{y} = 0$ whence $y \approx x^2$ and thus predict linear growth of $\dot{x} = \epsilon x$. Normal form transforms get the deterministic dynamics correctly. But what happens for stochastic dynamics?

Multiply a nonlinear term in the x SDE in order to get cancellation when the right-hand sides are multiplied by `small`. Multiply the bifurcation parameter by `small^2` in order to make small.

```
42 if thecase=sdebif then begin
43   xrhs:={small*eps*x(1)+small*x(1)^3-x(1)*y(1)*(1-small*w(1))};
```

Insert the noise in a rather special way so that its dominant effects cancel.

```
44   yrhs:={-y(1)+x(1)^2+y(1)^2+y(1)*w(1)};
```

There are no unstable modes.

```
45   zrhs:={};
```

Truncate to higher order in the amplitudes in order to discern the subtle bifurcation.

```
46   toosmall:=5;
47   factor small,sigma;
48 end;
```

The coordinate transform is very messy, but dominantly (dropping subscripts for simplicity)

$$\begin{aligned} x \approx & X + XY + 2X^3Y \\ & + \sigma \left[(-XY^2 + 3X^3Y)e^{+t\star} + (XY^2 + XY^3)e^{2t\star} \right. \\ & \left. + X^3e^{-t\star} - XY^3e^{3t\star} \right] w, \end{aligned} \quad (6)$$

$$\begin{aligned} y \approx & Y - Y^2 + Y^3 - Y^4 + X^2 - 7X^2Y^2 + X^4 \\ & + \sigma \left[(-Y + 2Y^3 - 3Y^4 - 10X^2Y^2 - 4X^2Y^2e^{+t\star})e^{+t\star} \right. \\ & \left. + (X^2 - 2X^2Y + 3X^2Y^2 + X^4 + 4X^4e^{-t\star})e^{-t\star} + 2X^2Y^2e^{2t\star} \right] w. \end{aligned} \quad (7)$$

In these coordinates the slow mode SDE becomes the normal form

$$\dot{X} \approx \epsilon X - X^5 - 2\sigma X^5 w - 3\sigma^2 X^5 w e^{-t\star} w, \quad (8)$$

$$\dot{Y} \approx (-1 + 4X^2 + 6X^4)Y + \sigma(1 + 2X^2 + 22X^4)Yw. \quad (9)$$

The ‘drift’ of the quadratic noise in \dot{X} should also be nonlinearly stabilising.

2.5 Levy area contraction: off-diagonal example

Pavliotis & Stuart (2008) assert the following system of five coupled SDEs are interesting.

$$dx_1 = \epsilon y_1 dt, \quad (10)$$

$$dx_2 = \epsilon y_2 dt, \quad (11)$$

$$dx_3 = \epsilon(x_1 y_2 - x_2 y_1) dt, \quad (12)$$

$$dy_1 = (-y_1 - \alpha y_2) dt + dW_1, \quad (13)$$

$$dy_2 = (+\alpha y_1 - y_2) dt + dW_2. \quad (14)$$

This stochastic system has two noise sources. I presume $W_i(t, \omega)$ are Stratonovich Wiener processes. Use $\mathbf{x}(\mathbf{i})$ to denote variable x_i , $\mathbf{y}(\mathbf{i})$ to denote variable y_i , and $\mathbf{w}(\mathbf{i})$ to denote noise dW_i/dt .

Let **eps** denote parameter ϵ . Thus specify the slow dynamics via this three component list allocated to **xrhs**.

```

49 if thecase=sdelac then begin
50 factor small,sigma,eps;
51 toosmall:=4;
52 xrhs:={eps*y(1),eps*y(2),eps*(x(1)*y(2)-x(2)*y(1))};

```

For the fast modes, specify the linear parts separately from the rest of the SDE. Here the linear part has an off-diagonal component parametrised by α . This code cannot exactly analyse such systems. Thus analyse with the α moderated terms when treated as a perturbation of the decay at rate one. Specify the dynamics of the y SDE as the two element list assigned to `yrhs`.

```

53 yrhs:={-y(1)-a*y(2)+w(1),-y(2)+a*y(1)+w(2)};

```

There are no unstable modes.

```

54 zrhs:={};
55 end;

```

The stochastic normal form is

$$\begin{aligned}
\dot{X}_1 &\approx \epsilon \sigma w_1 - \epsilon \sigma w_2 \alpha, \\
\dot{X}_2 &\approx \epsilon \sigma w_2 + \epsilon \sigma w_1 \alpha, \\
\dot{X}_3 &\approx \epsilon \sigma (-w_1 X_2 + w_2 X_1) + \epsilon \sigma (w_1 X_1 \alpha + w_2 X_2 \alpha) \\
&\quad + \epsilon^2 \sigma^2 (w_1 e^{-t} * w_2 - w_2 e^{-t} * w_1), \\
\dot{Y}_1 &\approx -Y_1 - Y_2 \alpha, \\
\dot{Y}_2 &\approx -Y_2 + Y_1 \alpha.
\end{aligned}$$

2.6 Position-momentum: the Jordan form

Suppose you want to analyse the semi-mechanical system of SDEs

$$\ddot{x} = -xy \quad \text{and} \quad \dot{y} = -2y + x^2 + \dot{x}^2 + \sigma w(t), \quad (15)$$

where $w(t)$ denotes the formal derivative dW/dt of a Stratonovich Wiener process $W(t, \omega)$. Parameter σ controls the strength of the noise. Introduce $x_1 = x$, $x_2 = \dot{x}$ and $y_1 = y$ to convert to the system of three coupled SDEs

$$\dot{x}_1 = x_2, \quad (16)$$

$$\dot{x}_2 = -x_1 y_1, \quad (17)$$

$$\dot{y}_1 = -2y_1 + x_1^2 + x_2^2 + \sigma w(t). \quad (18)$$

Specify the slow x SDEs by allocating the two element list of its right-hand side to the variable `xrhs`. Divide the x_1 term by `small` on the right-hand side of \dot{x}_2 in order to overcome the automatic multiplication of the right-hand side by `small`. Iteration still works for this Jordan form system: but I am not responsible for anyone who divides by `small` or `sigma`.

```
56 if thecase=sdemom then begin
57   xrhs:={x(2)/small,-x(1)*y(1)};
58   yrhs:={-2*y(1)+x(1)^2+x(2)^2+w(y)};
59   factor small,sigma;
```

Note: the code automatically multiply the noise factors by a parameter `sigma` so there is no need include the parameter σ in the specification of the problem.

There are no unstable modes.

```
60   zrhs:={};
61 end;
```

The resultant normal form is

$$\begin{aligned} \dot{X}_1 &\approx X_2 + \sigma\left(\frac{1}{4}wX_1 + \frac{1}{4}wX_2\right) \\ &\quad + \sigma^2\left(-\frac{3}{32}wX_1e^{-2t} \star w - \frac{3}{64}wX_2e^{-2t} \star w\right) \\ \dot{X}_2 &\approx \sigma\left(-\frac{1}{2}wX_1 - \frac{1}{4}wX_2\right) + \left(-\frac{1}{2}X_1^3 + \frac{1}{2}X_2X_1^2 - \frac{3}{4}X_2^2X_1\right) \\ &\quad + \sigma^2\left(\frac{1}{8}wX_1e^{-2t} \star w + \frac{3}{32}wX_2e^{-2t} \star w\right) \\ \dot{Y}_1 &\approx -2Y_1 + \left(-\frac{1}{2}X_1^2Y_1 + \frac{1}{2}X_2X_1Y_1 + \frac{1}{2}X_2^2Y_1\right) \end{aligned}$$

2.7 Radek's slow oscillation with fast noise

Consider Radek's system

$$\dot{x} = -\epsilon xz, \quad \dot{y} = +\epsilon yz \quad \text{and} \quad \dot{z} = -(z-1) + \sigma w(t).$$

Transform to our standard form by

$$x = x_1, \quad y = x_2 \quad \text{and} \quad z = 1 + y_1.$$

Then obtain the stochastic normal form with the following code.

```
62 if thecase=sderadek then begin
63 yrhs:={-y(1)+w(1)};
64 xrhs:={-eps*x(2)*(1+y(1)),eps*x(1)*(1+y(1))};
65 factor small,sigma,eps;
66 zrhs:={};
67 end;
```

The normal form equations are

$$\begin{aligned}\dot{X}_1 &\approx -\epsilon(1 + \sigma w)X_2, \\ \dot{X}_2 &\approx +\epsilon(1 + \sigma w)X_1, \\ \dot{Y}_1 &\approx -Y_1.\end{aligned}$$

The dynamics clearly oscillate in (X_1, X_2) with phase angle $\theta = \epsilon(t + \sigma W(t, \omega))$. This normal form arises from the coordinate transform

$$\begin{aligned}x_1 &\approx X_1 + \epsilon X_2 Y_1 + \epsilon \sigma X_2 e^{-t} \star w - \frac{1}{2} \epsilon^2 X_1 Y_1^2, \\ x_2 &\approx X_2 - \epsilon X_1 Y_1 - \epsilon \sigma X_1 e^{-t} \star w - \frac{1}{2} \epsilon^2 X_2 Y_1^2, \\ y_1 &= Y_1 + \sigma e^{-t} \star w.\end{aligned}$$

Different analysis has to be done for the case of fast oscillation in this system.

2.8 Simple hyperbolic system

$$\begin{aligned}\dot{y}_1 &= w_1 z_1 \sigma - y_1 \\ \dot{z}_1 &= w_1 y_1 \sigma + z_1\end{aligned}$$

```
68 if thecase=sdehyper then begin
69 yrhs:={-y(1)+z(1)*w(1)};
70 xrhs:={};
71 zrhs:={+z(1)+y(1)*w(1)};
72 end;
```

The stochastic coordinate transform $z_1 = -e^{2t} w_1 Y_1 \sigma + Z_1$
 $y_1 = e^{-2t} w_1 Z_1 \sigma + Y_1$

Result normal form SDEs $\dot{Z}_1 = e^{-2t} w_1 w_1 Z_1 \sigma^2 + Z_1$
 $\dot{Y}_1 = -e^{2t} w_1 w_1 Y_1 \sigma^2 - Y_1$

2.9 Duan's hyperbolic system for foliation

Used as an example by [Sun et al. \(2011\)](#).

$$\dot{y}_1 = w_1 y_1 \sigma - y_1$$

$$\dot{z}_1 = w_1 z_1 \sigma + y_1^2 \varepsilon + z_1$$

```
73 if thecase=sdeduan then begin
74   yrhs:={-y(1)+y(1)*w(1)};
75   xrhs:={};
76   zrhs:={+z(1)+y(1)^2+z(1)*w(1)};
77 end;
```

The stochastic coordinate transform

$$z_1 = -1/3 e^{3t} w_1 Y_1^2 \sigma - 1/3 Y_1^2 + Z_1$$

$$y_1 = Y_1$$

Result normal form SDEs

$$\dot{Z}_1 = w_1 Z_1 \sigma + Z_1$$

$$\dot{Y}_1 = w_1 Y_1 \sigma - Y_1$$

2.10 A four mode hyperbolic system

$$\dot{x}_1 = w_2 y_1 \sigma + w_1 z_1 \sigma + y_1 z_2 \varepsilon$$

$$\dot{y}_1 = w_1 z_1 \sigma - y_1$$

$$\dot{z}_1 = w_1 y_1 \sigma + z_1$$

$$\dot{z}_2 = w_2 y_1 \sigma + 2z_2$$

```

78 if thecase=sdehifour then begin
79   yrhs:={-y(1)+z(1)*w(1)};
80   xrhs:={y(1)*w(2)+z(1)*w(1)+y(1)*z(2)};
81   zrhs:={+z(1)+y(1)*w(1)
82     ,+2*z(2)+y(1)*w(2)};
83 end;
```

The stochastic coordinate transform

$$z_1 = -e^{2t} \star w_1 Y_1 \sigma + Z_1$$

$$z_2 = -e^{3t} \star w_2 Y_1 \sigma + Z_2$$

$$y_1 = e^{-2t} \star w_1 Z_1 \sigma + Y_1$$

$$x_1 = -e^{3t} \star w_2 Y_1^2 \sigma + e^{2t} \star w_2 Y_1^2 \sigma - e^t \star w_2 Y_1 \sigma + e^{-1t} \star w_1 Z_1 \sigma + e^{-2t} \star w_1 Z_2 Z_1 \sigma - e^{-3t} \star w_1 Z_2 Z_1 \sigma + X_1 + Y_1 Z_2$$

Result normal form SDEs

$$\dot{Z}_1 = e^{-2t} \star w_1 w_1 Z_1 \sigma^2 + Z_1$$

$$\dot{Z}_2 = 2Z_2$$

$$\dot{Y}_1 = -e^{2t} \star w_1 w_1 Y_1 \sigma^2 - Y_1$$

$$\dot{X}_1 = e^{3t} \star w_2 w_1 Y_1 Z_1 \sigma^2 - 2e^{-2t} \star w_1 w_2 Y_1 Z_1 \sigma^2$$

2.11 Monahan’s four examples

Monahan & Culina (2011) discuss stochastic averaging and give four examples in an appendix which I also analyse here. They also give a couple of interesting examples in the body of the article which I may also explore at some time. I contend that they really need my approach as “a large separation often does not exist in atmosphere or ocean dynamics” between the fast and slow time scales.

2.11.1 Example four: ‘three’ time scales

Monahan & Culina (2011) comment that this linear system has three time scales. But I do not see that: I only see varying strength interactions. And as a linear system, it is the simplest. They consider

$$\frac{dx}{dt} = -x + \frac{a}{\sqrt{\tau}}y \quad \text{and} \quad \frac{dy}{dt} = \frac{1}{\sqrt{\tau}}x - \frac{1}{\tau}y + \frac{b}{\sqrt{\tau}}\dot{W}.$$

Let $\tau = \epsilon^2$ and rescale time, $t = \tau t' = \epsilon^2 t'$ so that $d/dt = \frac{1}{\tau}d/dt'$ and $\dot{W} = \frac{1}{\sqrt{\tau}}dW/dt'$. Then, dropping dashes, the SDE system is

$$\frac{dx}{dt} = -\epsilon^2 x + a\epsilon y \quad \text{and} \quad \frac{dy}{dt} = \epsilon x - y + b\dot{W}.$$

Using the default inbuilt parametrisation of noise by **sigma** to represent parameter **b**, and using **small** in the x-SDE so that it counts the numbers of small ϵ , code these as the following.

```
84 if thecase=sdeMona4 then begin
85   xrhs:={eps*a*y(1)-eps^2*small*x(1)};
86   yrhs:={eps*x(1)-y(1)+w(1)};
87   zrhs:={ };
88   toosmall:=4;
89   factor small,sigma,eps,yy,y,w,ou;
90 end;
```


The stochastic coordinate transform

$$\begin{aligned} y_1 &= -e^{-1t} \star e^{-1t} \star w_1 \epsilon^2 \sigma a - e^{-1t} \star w_1 \epsilon^2 \sigma a + e^{-1t} \star w_1 \sigma + Y_1 + \epsilon X_1 \\ x_1 &= -e^{-1t} \star w_1 \epsilon \sigma a - Y_1 \epsilon a + X_1 \end{aligned}$$

Result normal form SDEs

$$\begin{aligned} \dot{Y}_1 &= -Y_1 \epsilon^2 a - Y_1 \\ \dot{X}_1 &= w_1 \epsilon^3 \sigma (-2a^2 + a) + w_1 \epsilon \sigma a + \epsilon^2 (X_1 a - X_1) \end{aligned}$$

Monahan & Culina (2011) derive the last two terms in the X-equation, but not the first as it is too small for their averaging analysis. They comment that $a > 1$ is some sort of difficulty; but I have no problem with $a > 1$ (until X growth invalidates the linearity), especially as the decay rate to the stochastic slow manifold, the Y-SDE, is $(1 + \epsilon a)$ which gets stronger with parameter a .

2.11.2 Example one: simple nonlinear

Monahan & Culina (2011) first consider the example

$$\frac{dx}{dt} = -x + \Sigma(x)y \quad \text{and} \quad \frac{dy}{dt} = -\frac{1}{\tau}y + \frac{1}{\sqrt{\tau}}\dot{W},$$

for general smooth functions $\Sigma(x)$. Rescale time, $t = \tau t'$ so that $d/dt = \frac{1}{\tau}d/dt'$ and $\dot{W} = \frac{1}{\sqrt{\tau}}dW/dt'$. Then, dropping dashes, the SDE is

$$\frac{dx}{dt} = -\tau x + \tau \Sigma(x)y \quad \text{and} \quad \frac{dy}{dt} = -y + \dot{W}.$$

Restricting to a rational function $\Sigma = (a_0 + a_1x + a_2x^2)/(1 + b_1x + b_2x^2)$, code these as the following (multiply through by the denominator).

```
91 if thecase=sdeMonal then begin
92 operator a; defindex a(down);
93 operator b; defindex b(down);
```

```

94 xrhs:={-tau*x(1)*(1+b(1)*x(1)+b(2)*x(1)^2)
95   -(b(1)*x(1)+b(2)*x(1)^2)*df(x(1),t)
96   +tau*y(1)*(a(0)+a(1)*x(1)+a(2)*x(1)^2) };
97 yrhs:={-y(1)+w(1)};
98 zrhs:={ };
99 toosmall:=3;
100 factor small,sigma,tau,yy,y,w,ou;
101 end;

```

The stochastic coordinate transform In the following expressions, recall that σ parametrises the noise; for comparison with the modelling of [Monahan & Culina \(2011\)](#), take $\sigma = 1$. Also recall that a rational represents the function Σ .

$$y_1 = e^{-1t} \star w_1 \sigma + Y_1$$

$$x_1 = e^{-1t} \star w_1 \sigma \varepsilon \tau (-a_2 X_1^2 - a_1 X_1 - a_0) + Y_1 \varepsilon \tau (-a_2 X_1^2 - a_1 X_1 - a_0) + X_1$$

Result normal form SDEs

$$\dot{Y}_1 = -Y_1$$

$$\begin{aligned} \dot{X}_1 = & w_1 \sigma \varepsilon^2 \tau^2 (a_2 b_2 X_1^4 - a_2 X_1^2 + 2a_1 b_2 X_1^3 + a_1 b_1 X_1^2 + 3a_0 b_2 X_1^2 + 2a_0 b_1 X_1 + \\ & a_0) + w_1 \sigma \varepsilon^2 \tau (-a_2 b_2 X_1^4 - a_2 b_1 X_1^3 - a_1 b_2 X_1^3 - a_1 b_1 X_1^2 - a_0 b_2 X_1^2 - a_0 b_1 X_1) + \\ & w_1 \sigma \varepsilon \tau (a_2 X_1^2 + a_1 X_1 + a_0) + \varepsilon^2 \tau (b_2^2 X_1^5 + 2b_2 b_1 X_1^4 + b_2 X_1^3 + b_1^2 X_1^3 + b_1 X_1^2) + \\ & \varepsilon \tau (-b_2 X_1^3 - b_1 X_1^2 - X_1) \end{aligned}$$

[Monahan & Culina \(2011\)](#) derive some of this X equation. The others here are higher order terms that become significant at finite parameter values. For example, the next correction to their analysis, $w_1 \tau^2 (-3a_4 X_1^4 - 2a_3 X_1^3 - a_2 X_1^2 + a_0)$, is probably derivable as $\tau^2 (\Sigma - x \Sigma') \dot{W}$ (when rescaled).

2.11.3 Example three: many fast modes

Monahan & Culina (2011) third considered the example

$$\frac{dx}{dt} = -x + \Sigma(x)\|\vec{y}\| \quad \text{and} \quad \frac{d\vec{y}}{dt} = -\frac{1}{\tau}\vec{y} + \sqrt{\frac{2}{\tau}}\dot{W},$$

for general smooth functions $\Sigma(x)$. As before, rescale time, $t = \tau t'$ so that $d/dt = \frac{1}{\tau}d/dt'$ and $\dot{W} = \frac{1}{\sqrt{\tau}}dW/dt'$. Here I also cheat: they have $\|\vec{y}\|$ in the slow equation; but $\|\vec{y}\|$ is not a smooth multinomial and so my generic program cannot apply; instead I replace $\|\vec{y}\|$ with $\|\vec{y}\|^2$ which has the same symmetry but is multinomial. Then, upon the rescaling of time, and dropping dashes, the SDE is

$$\frac{dx}{dt} = -\tau x + \tau \Sigma(x)\|\vec{y}\|^2 \quad \text{and} \quad \frac{d\vec{y}}{dt} = -\vec{y} + \sigma \dot{W}.$$

Restricting to the general quartic $\Sigma = a_0 + a_1x + \dots + a_4x^4$, code these as the following (the generic program automatically inserts the σ in the noise). Currently restrict to just a two component \vec{y} as I do not see any reason for any more and Monahan & Culina (2011) do not appear to specify.

```

102 if thecase=sdeMona3 then begin
103   operator a; defindex a(down);
104   xrhs:={-tau*x(1)+tau*(y(1)^2+y(2)^2
105   *(a(0)+a(1)*x(1)+a(2)*x(1)^2+a(3)*x(1)^3+a(4)*x(1)^4) };
106   yrhs:={-y(1)+w(1),-y(2)+w(2)};
107   zrhs:={ };
108   toosmall:=3;
109   factor small,sigma,tau,yy,y,w,ou;
110 end;
```

The stochastic coordinate transform In the following expressions, recall that σ parametrises the noise; for comparison with the modelling of Monahan & Culina (2011), you might take $\sigma = \sqrt{2}$ but as I changed the SDE somewhat an exact comparison is not possible. Also recall that a general quartic represents the function Σ .

$$y_1 = e^{-1t} \star w_1 \sigma + Y_1$$

$$y_2 = e^{-1t} \star w_2 \sigma + Y_2$$

$$\begin{aligned} x_1 = & e^t \star w_2 Y_2 \sigma \tau (-a_4 X_1^4 - a_3 X_1^3 - a_2 X_1^2 - a_1 X_1 - a_0) + e^{-1t} \star w_2 Y_2 \sigma \tau (- \\ & a_4 X_1^4 - a_3 X_1^3 - a_2 X_1^2 - a_1 X_1 - a_0) + e^t \star w_1 Y_1 \sigma \tau (-a_4 X_1^4 - a_3 X_1^3 - a_2 X_1^2 - \\ & a_1 X_1 - a_0) + e^{-1t} \star w_1 Y_1 \sigma \tau (-a_4 X_1^4 - a_3 X_1^3 - a_2 X_1^2 - a_1 X_1 - a_0) + Y_2^2 \tau (- \\ & 1/2 a_4 X_1^4 - 1/2 a_3 X_1^3 - 1/2 a_2 X_1^2 - 1/2 a_1 X_1 - 1/2 a_0) + Y_1^2 \tau (-1/2 a_4 X_1^4 - 1/2 a_3 X_1^3 - \\ & 1/2 a_2 X_1^2 - 1/2 a_1 X_1 - 1/2 a_0) + X_1 \end{aligned}$$

Result normal form SDEs

$$\dot{Y}_1 = -Y_1$$

$$\dot{Y}_2 = -Y_2$$

$$\begin{aligned} \dot{X}_1 = & e^{-1t} \star w_2 w_2 \sigma^2 \tau^2 (-3/2 a_4 X_1^4 - a_3 X_1^3 - 1/2 a_2 X_1^2 + 1/2 a_0) + e^{-1t} \star w_2 w_2 \sigma^2 \tau (a_4 X_1^4 + \\ & a_3 X_1^3 + a_2 X_1^2 + a_1 X_1 + a_0) + e^{-1t} \star w_1 w_1 \sigma^2 \tau^2 (-3/2 a_4 X_1^4 - a_3 X_1^3 - 1/2 a_2 X_1^2 + \\ & 1/2 a_0) + e^{-1t} \star w_1 w_1 \sigma^2 \tau (a_4 X_1^4 + a_3 X_1^3 + a_2 X_1^2 + a_1 X_1 + a_0) - \tau X_1 \end{aligned}$$

In this modelling for X , the three terms linear in τ are the leading order, and rewritten are

$$\dot{X} \approx -\tau X + \tau \Sigma(X) \sigma^2 (w_1 e^{-t} \star w_1 + w_2 e^{-t} \star w_2)$$

Such quadratic terms in the noise generate both fluctuations and mean drift ([Chao & Roberts 1996](#)): the mean drift effect from each is $\sigma^2/2$, so their mean sum is just σ^2 . Hence the mean part of this model reduces to the form of (A33) ([Monahan & Culina 2011](#)).

The fluctuations in $w_1 e^{-t} \star w_1$ are skewed, on finite times, and so should contribute to the skewness commented on by [Monahan & Culina \(2011\)](#). A generalisation of the Fokker–Planck analysis of [Chao & Roberts \(1996\)](#) suggests that such skewness decays algebraically in the scale separation τ : such algebraic decay in τ makes skewness much more noticeable at finite τ than other modelling approximations which decay exponentially in the scale separation.

2.11.4 Example two: irregular slow manifold

Monahan & Culina (2011) second consider the example

$$\frac{dx}{dt} = x - x^3 + \Sigma(x)y \quad \text{and} \quad \frac{dy}{dt} = -\frac{1}{x\tau}y + \frac{1}{\sqrt{\tau}}\dot{W},$$

for general smooth functions $\Sigma(x)$. Since the y -dynamics are at least exponentially unstable for negative x , we restrict attention to $x > 0$. Even for positive x the system is singular as $x \rightarrow 0$ so the slow manifold is irregular in some sense (although ‘singular’ in a good way in that the scale separation between fast and slow becomes infinite). Here I think we have to be more sophisticated in rescaling time: let’s choose the new fast time t' so that $dt = x\tau dt'$; that is, $t' = \int (x\tau)^{-1} dt$ which would not be explicitly known until after a solution $x(t')$ is found. I presume that the noise then transforms as $\dot{W} = \frac{1}{\sqrt{x\tau}} dW/dt'$ (needs checking). Then, dropping dashes, the SDE is

$$\frac{dx}{dt} = \tau \left[x^2 - x^4 + x\Sigma(x)y \right] \quad \text{and} \quad \frac{dy}{dt} = -y + \sqrt{x}\dot{W}.$$

Now the \sqrt{x} is a problem in my generic computer algebra which requires multinomial systems so transform to $x = \xi^2$ so that $2\xi d\xi = dx$. Then the SDE system takes on a multinomial form

$$\frac{d\xi}{dt} = \frac{1}{2}\tau \left[\xi^3 - \xi^7 + \xi\Sigma(\xi^2)y \right] \quad \text{and} \quad \frac{dy}{dt} = -y + \xi\dot{W}.$$

The Stratonovich and Ito versions of the above SDE are still the same (an unresolved question is whether the non-uniform time scaling introduces a difference). Restricting to the general linear $\Sigma = a_0 + a_1x$, code the SDE system as the following (remember $x(1) = \xi = \sqrt{x}$).

```

111 if thecase=sdeMona2 then begin
112 operator a; defindex a(down);
113 xrhs := {1/2*tau*(x(1)^3-x(1)^7+x(1)*(a(0)+a(1)*x(1)^2)*y(1))}$
114 yrhs := { -y(1)+x(1)*w(1) }$
115 zrhs := {}$
116 factor small,sigma,tau,yy,y,w,ou;
117 end;
```

The stochastic coordinate transform In the following expressions, recall that σ parametrises the noise; for comparison with the modelling of [Monahan & Culina \(2011\)](#), take $\sigma = 1$.

$$y_1 = e^{-1t} \star e^{-1t} \star w_1 \sigma \tau (1/2X_1^7 - 1/2X_1^3) + e^{-1t} \star w_1 \sigma X_1 + Y_1$$

$$x_1 = e^{-1t} \star w_1 \sigma \tau (-1/2a_1X_1^4 - 1/2a_0X_1^2) + Y_1 \tau (-1/2a_1X_1^3 - 1/2a_0X_1) + X_1$$

Result normal form SDEs

$$\begin{aligned} \dot{Y}_1 = & e^t \star w_1 w_1 Y_1 \sigma^2 \tau^2 (1/4a_1^2 X_1^6 + 1/2a_1 a_0 X_1^4 + 1/4a_0^2 X_1^2) + \\ & e^{-1t} \star w_1 w_1 Y_1 \sigma^2 \tau^2 (3/4a_1^2 X_1^6 + a_1 a_0 X_1^4 + 1/4a_0^2 X_1^2) + w_1 Y_1 \sigma \tau^2 (-a_1 X_1^9 - \\ & 3/2a_0 X_1^7 + 1/2a_0 X_1^3) + w_1 Y_1 \sigma \tau (-1/2a_1 X_1^3 - 1/2a_0 X_1) - Y_1 \end{aligned}$$

$$\begin{aligned} \dot{X}_1 = & e^{-1t} \star w_1 w_1 \sigma^2 \tau^2 (-1/4a_1^2 X_1^7 - 1/2a_1 a_0 X_1^5 - 1/4a_0^2 X_1^3) + w_1 \sigma \tau^2 (a_1 X_1^{10} + \\ & 3/2a_0 X_1^8 - 1/2a_0 X_1^4) + w_1 \sigma \tau (1/2a_1 X_1^4 + 1/2a_0 X_1^2) + \tau (-1/2X_1^7 + 1/2X_1^3) \end{aligned}$$

Using just the leading order terms, the ones linear in τ , and recalling $X_1 \approx \xi = \sqrt{x}$, the last SDE gives the model

$$\frac{dx}{dt'} \approx \tau \left[x^2 - x^4 + x^{3/2} \Sigma(x) \sigma \frac{dW}{dt'} \right].$$

But recall that $dt' = dt/(x\tau)$ (although one should be more careful as $X_1 \approx \sqrt{x}$, not exact equality) and similarly $dW/dt' = \sqrt{x\tau} dW$ so that this model becomes

$$\frac{dx}{dt} \approx x - x^3 + \sqrt{\tau x} \Sigma(x) \sigma \frac{dW}{dt}.$$

This agrees with the Stratonovich part of (A28) by [Monahan & Culina \(2011\)](#). But again, the above derivation has the systematic higher order corrections that are needed for finite scale separation τ (such corrections appear to be of the same order as the difference between the Ito and Stratonovich versions of this model).

2.11.5 Idealised Stommel-like model of meridional overturning circulation

Monahan & Culina (2011) also analyse the Idealised Stommel-like model

$$\begin{aligned}\frac{dx}{dt} &= \mu - |y - x|x + \sigma_A \dot{W}_1, \\ \frac{dy}{dt} &= +\frac{1}{\tau}(1 - y) - |y - x|y + \sqrt{\frac{2}{\tau}}\sigma_M \dot{W}_2.\end{aligned}$$

The mod-functions do not fit into my generic computer algebra so replace them with squares to at least preserve the symmetry. As before, introduce $\epsilon^2 = \tau$ and rescale time, $t = \tau t' = \epsilon^2 t'$ so that $d/dt = \frac{1}{\tau}d/dt'$ and $\dot{W}_j = \frac{1}{\sqrt{\tau}}dW_j/dt' = \frac{1}{\epsilon}dW_j/dt'$. Since for small τ , the fast variable y is strongly attracted to one, change the reference point for y by setting $y = 1 + y'(t)$. Then the SDEs becomes akin to

$$\begin{aligned}\frac{dx}{dt'} &= \epsilon^2 \left[\mu - (1 + y' - x)^2 x \right] + \epsilon \sigma_A \frac{dW_1}{dt'}, \\ \frac{dy'}{dt'} &= -y' - \epsilon^2 (1 + y' - x)^2 (1 + y') + \sqrt{2}\sigma_M \frac{dW_2}{dt'}.\end{aligned}$$

Let $\rho = \sigma_A/(\sqrt{2}\sigma_M)$, use the inbuilt $\sigma = \sqrt{2}\sigma_M$, and invoke `small` to correctly count the number of small ϵ s in the analysis. Code the above dynamics as the following.

```
118 if thecase=sdeMonaSS then begin
119   xrhs := {small*eps^2*(mu-(1+y(1)-x(1))^2*x(1))
120     +small*eps*rho*w(1)}$
121   yrhs := { -y(1)-small*eps^2*(1+y(1)-x(1))^2*(1+y(1))+w(2) }$
122   zrhs := {}$
123   factor small,sigma,eps,rho,yy,y,w,ou;
124   toosmall:=4;
125 end;
```

The stochastic coordinate transform

$$\begin{aligned}
y_1 &= e^{-1t} \star e^{-1t} \star w_2 \epsilon^2 \sigma (-X_1^2 + 4X_1 - 3) + 3/2 e^{1t} \star w_2 Y_1^2 \epsilon^2 \sigma + \\
&3/2 e^{-1t} \star w_2 Y_1^2 \epsilon^2 \sigma + e^{-1t} \star w_2 Y_1 \epsilon^2 \sigma (-4X_1 + 6) + e^{-1t} \star w_2 \sigma + 1/2 Y_1^3 \epsilon^2 + \\
&Y_1^2 \epsilon^2 (-2X_1 + 3) + Y_1 + \epsilon^2 (-X_1^2 + 2X_1 - 1) \\
x_1 &= e^{1t} \star w_2 Y_1 \epsilon^2 \sigma X_1 + e^{-1t} \star w_2 Y_1 \epsilon^2 \sigma X_1 + e^{-1t} \star w_2 \epsilon^2 \sigma (-2X_1^2 + 2X_1) + \\
&1/2 Y_1^2 \epsilon^2 X_1 + Y_1 \epsilon^2 (-2X_1^2 + 2X_1) + X_1
\end{aligned}$$

Result normal form SDEs

$$\begin{aligned}
\dot{Y}_1 &= -3e^{-1t} \star w_2 w_2 Y_1 \epsilon^2 \sigma^2 + 4e^{-1t} \star w_2 w_1 Y_1 \epsilon^3 \rho \sigma^2 + w_2 Y_1 \epsilon^2 \sigma (4X_1 - 6) + \\
&Y_1 \epsilon^2 (-X_1^2 + 4X_1 - 3) - Y_1 \\
\dot{X}_1 &= -e^{-1t} \star w_2 w_2 \epsilon^2 \sigma^2 X_1 + e^{-1t} \star w_2 w_1 \epsilon^3 \rho \sigma^2 (4X_1 - 2) + w_2 \epsilon^2 \sigma (2X_1^2 - 2X_1) + \\
&w_1 \epsilon \rho \sigma + \epsilon^2 (-X_1^3 + 2X_1^2 - X_1 + \mu)
\end{aligned}$$

Deterministically, this model has multiple equilibria for small μ , but only one equilibria for $\mu > 4/27$, at finite amplitude. The noise \dot{W}_1 causes transitions between such multiple equilibria, and the multiplicative noise \dot{W}_2 contributes as well. But the same order of smallness is the first term in the X_1 SDE above which is a quadratic noise that has a mean drift effect that should enhance the stability of the small x equilibrium.

2.12 Majda's triad models

[Majda et al. \(2002\)](#) investigated averaging in two 3D SDE systems. I also looked at these in 2003.³ Let's look at the stochastic normal form.

2.12.1 Multiplicative triad model

The multiplicative triad model of [Majda et al. \(2002\)](#) consists of three modes, v_1 , v_2 and v_3 . These evolve in time according to

$$\frac{dv_1}{dt} = b_1 v_2 v_3, \quad \frac{dv_2}{dt} = b_2 v_1 v_3, \quad \frac{dv_3}{dt} = -v_3 + b_3 v_1 v_2 + \sigma \dot{W}, \quad (19)$$

³Centre manifold analysis of stochastic multiplicative triad model, technical report.

where b_i and σ are some constants and the noise forces the third mode. Here I have already scaled the equations so that the rate of decay of the third mode is one. Thus on long time scales we expect the third mode to be essentially negligible and the system to be modelled by the relatively slow evolution of the first two modes.

```

126 if thecase=sdeMajda3m then begin
127 operator b; defindex b(down);
128 xrhs := {b(1)*x(2)*y(1),b(2)*x(1)*y(1)}$
129 yrhs := { -y(1)+b(3)*x(1)*x(2)+w(3) }$
130 zrhs := {}$
131 factor small,sigma,yy,y,w,ou;
132 toosmall:=5;
133 end;

```

The stochastic coordinate transform

$$y_1 = e^{-1t} \star e^{-1t} \star w_3 \sigma(-b_3 b_2 X_1^2 - b_3 b_1 X_2^2) - 4e^{-1t} \star w_3 Y_1 \sigma b_3 b_2 b_1 X_2 X_1 + e^{-1t} \star w_3 \sigma(-b_3 b_2 X_1^2 - b_3 b_1 X_2^2 + 1) - 2Y_1^2 b_3 b_2 b_1 X_2 X_1 + Y_1 - b_3^2 b_2 X_2 X_1^3 - b_3^2 b_1 X_2^3 X_1 + b_3 X_2 X_1$$

$$x_1 = e^{-1t} \star e^{-1t} \star w_3 \sigma(b_3 b_2 b_1 X_2 X_1^2 + b_3 b_1^2 X_2^3) - 1/2 e^{-1t} \star w_3 Y_1^2 \sigma b_2 b_1^2 X_2 + e^{-1t} \star w_3 Y_1 \sigma e^{-1t} \star w_3 \sigma(2b_3 b_2 b_1 X_2 X_1^2 + 2b_3 b_1^2 X_2^3 - b_1 X_2) - 1/6 Y_1^3 b_2 b_1^2 X_2 + 1/2 Y_1^2 b_2 b_1 X_1 + Y_1(b_3 b_2 b_1 X_2 X_1^2 + b_3 b_1^2 X_2^3 - b_1 X_2) + X_1$$

$$x_2 = e^{-1t} \star e^{-1t} \star w_3 \sigma(b_3 b_2^2 X_1^3 + b_3 b_2 b_1 X_2^2 X_1) - 1/2 e^{-1t} \star w_3 Y_1^2 \sigma b_2^2 b_1 X_1 + e^{-1t} \star w_3 Y_1 \sigma e^{-1t} \star w_3 \sigma(2b_3 b_2^2 X_1^3 + 2b_3 b_2 b_1 X_2^2 X_1 - b_2 X_1) - 1/6 Y_1^3 b_2^2 b_1 X_1 + 1/2 Y_1^2 b_2 b_1 X_2 + Y_1(b_3 b_2^2 X_1^3 + b_3 b_2 b_1 X_2^2 X_1 - b_2 X_1) + X_2$$

Result normal form SDEs

$$\dot{Y}_1 = e^{-1t} \star w_3 w_3 Y_1 \sigma^2(2b_3 b_2^2 b_1 X_1^2 + 2b_3 b_2 b_1^2 X_2^2) + 4w_3 Y_1 \sigma b_3 b_2 b_1 X_2 X_1 + Y_1(b_3^2 b_2^2 X_1^4 + 2b_3^2 b_2 b_1 X_2^2 X_1^2 + b_3^2 b_1^2 X_2^4 - b_3 b_2 X_1^2 - b_3 b_1 X_2^2 - 1)$$

$$\dot{X}_1 = -2e^{-1t} \star w_3 w_3 \sigma^2 b_3 b_2 b_1^2 X_2^2 X_1 + w_3 \sigma(-2b_3 b_2 b_1 X_2 X_1^2 - 2b_3 b_1^2 X_2^3 + b_1 X_2) - b_3^2 b_2 b_1 X_2^3 X_1 - b_3^2 b_1^2 X_2^4 X_1 + b_3 b_1 X_2^2 X_1$$

$$\dot{X}_2 = -2e^{-1t} \star w_3 w_3 \sigma^2 b_3 b_2^2 b_1 X_2 X_1^2 + w_3 \sigma (-2b_3 b_2^2 X_1^3 - 2b_3 b_2 b_1 X_2^2 X_1 + b_2 X_1) - b_3^2 b_2^2 X_2 X_1^4 - b_3^2 b_2 b_1 X_2^3 X_1^2 + b_3 b_2 X_2 X_1^2$$

Majda et al. (2002) predicts, their equation (52), the two leading order terms in the deterministic part and the linear noise part. I suspect their first term in each equation is an Ito version of my Stratonovich modelling. All the higher order terms here are missed by their averaging.

2.12.2 Additive triad model

The additive triad model of Majda et al. (2002) consists of three modes, v_1 , v_2 and v_3 , as before. However, these now evolve in time according to

$$\begin{aligned} \frac{dv_1}{dt} &= b_1 v_2 v_3, \\ \frac{dv_2}{dt} &= -v_2 + b_2 v_1 v_3 + \sigma_2 \dot{W}_2, \\ \frac{dv_3}{dt} &= -v_3 + b_3 v_1 v_2 + \sigma_3 \dot{W}_3, \end{aligned} \tag{20}$$

where b_i and σ_i are some constants, and there is independent stochastic forcing of the second and third modes. Here I have already scaled the equations so that the rate of decay of *both* the second and third mode is one.⁴ Thus on long time scales we expect the second and third modes to be essentially negligible and the system to be modelled by the relatively slow evolution of the first mode. This section constructs the stochastic normal form of its centre manifold model as the basis for a model over long time scales with new noise processes.

```
134 if thecase=sdeMajda3a then begin
135   operator b; defindex b(down);
136   xrhs := {b(1)*y(2)*y(1)}$
```

⁴In contrast, Majda et al. (2002) set the two modes to have different decay rates. I do not expect much difference in using the same decay rate, it is just more convenient that the memory convolutions are then identical for the two modes rather than being different. Having the decay rates the same is also closer to my expected application to spatial problems.

```

137 yrhs := { -y(1)+b(2)*x(1)*y(2)+b(21)*w(2)
138           , -y(2)+b(3)*x(1)*y(1)+b(31)*w(3) }$
139 zrhs := {}$
140 factor small,sigma,yy,y,xx,x;
141 toosmall:=3;
142 end;

```

The stochastic coordinate transform

$$\begin{aligned}
y_1 &= X_1 \sigma b_{31} b_2 e^{-1t} \star e^{-1t} \star w_3 + Y_1 + \sigma b_{21} e^{-1t} \star w_2 \\
y_2 &= X_1 \sigma b_{21} b_3 e^{-1t} \star e^{-1t} \star w_2 + Y_2 + \sigma b_{31} e^{-1t} \star w_3 \\
x_1 &= X_1 - 1/2 Y_2 Y_1 b_1 + Y_2 \sigma \left(-1/2 b_{21} b_1 e^t \star w_2 - 1/2 b_{21} b_1 e^{-1t} \star w_2 \right) + \\
&Y_1 \sigma \left(-1/2 b_{31} b_1 e^t \star w_3 - 1/2 b_{31} b_1 e^{-1t} \star w_3 \right)
\end{aligned}$$

Result normal form SDEs

$$\begin{aligned}
\dot{Y}_1 &= X_1 Y_2 b_2 + Y_2 \sigma^2 \left(-1/2 b_{31} b_{21} b_2 b_1 e^{-1t} \star e^{-1t} \star w_3 w_2 - \right. \\
&1/2 b_{31} b_{21} b_2 b_1 e^{-1t} \star w_3 w_2 - 1/2 b_{31} b_{21} b_2 b_1 e^{-1t} \star w_2 w_3 \left. \right) + Y_1 \sigma^2 \left(- \right. \\
&1/2 b_{31}^2 b_2 b_1 e^{-1t} \star e^{-1t} \star w_3 w_3 - 1/2 b_{31}^2 b_2 b_1 e^{-1t} \star w_3 w_3 \left. \right) - Y_1 \\
\dot{Y}_2 &= X_1 Y_1 b_3 + Y_2 \sigma^2 \left(-1/2 b_{21}^2 b_3 b_1 e^{-1t} \star e^{-1t} \star w_2 w_2 - \right. \\
&1/2 b_{21}^2 b_3 b_1 e^{-1t} \star w_2 w_2 \left. \right) - Y_2 + Y_1 \sigma^2 \left(-1/2 b_{31} b_{21} b_3 b_1 e^{-1t} \star e^{-1t} \star w_2 w_3 - \right. \\
&1/2 b_{31} b_{21} b_3 b_1 e^{-1t} \star w_3 w_2 - 1/2 b_{31} b_{21} b_3 b_1 e^{-1t} \star w_2 w_3 \left. \right) \\
\dot{X}_1 &= X_1 \sigma^2 \left(1/2 b_{31}^2 b_2 b_1 e^{-1t} \star e^{-1t} \star w_3 w_3 + 1/2 b_{31}^2 b_2 b_1 e^{-1t} \star w_3 w_3 + \right. \\
&1/2 b_{21}^2 b_3 b_1 e^{-1t} \star e^{-1t} \star w_2 w_2 + 1/2 b_{21}^2 b_3 b_1 e^{-1t} \star w_2 w_2 \left. \right) + \\
&\sigma^2 \left(1/2 b_{31} b_{21} b_1 e^{-1t} \star w_3 w_2 + 1/2 b_{31} b_{21} b_1 e^{-1t} \star w_2 w_3 \right)
\end{aligned}$$

The only terms in the model are the quadratic noise-noise interaction terms. [Majda et al. \(2002\)](#) recognise the last, σ^2 term, but not the first, $X_1 \sigma^2$ term. They represent it as a mean drift and independent noise (presumably the mean drift comes from the Ito representation of my Stratonovich noise).

2.13 Potzsche and Rasmussen deterministic non-autonomous examples

Potzsche & Rasmussen (2006) establish Taylor approximations of various integral manifolds of non-autonomous systems. They give two examples.

2.13.1 Lorenz near the pitchfork bifurcation

Example 5.1 of Potzsche & Rasmussen (2006) is

$$\begin{aligned}\dot{x}_1 &= \sigma_\epsilon(x_2 - x_1), \\ \dot{x}_2 &= \rho_\epsilon x_1 - x_2 - x_1 x_3, \\ \dot{x}_3 &= -\beta_\epsilon x_3 + x_1 x_2.\end{aligned}$$

where parameters are $\sigma_\epsilon = \sigma_0 + \epsilon\sigma(t)$, $\rho_\epsilon = 1 + \rho_0 + \epsilon\rho(t)$ and $\beta_\epsilon = \beta_0 + \epsilon\beta(t)$. When there is no parametric fluctuations, $\epsilon = 0$, there is a pitchfork bifurcation as ρ_0 crosses zero.

To analyse dynamics at this pitchfork bifurcation in the presence of fluctuations, Potzsche & Rasmussen (2006) take a linear transform of the system to variables \vec{y} and set $\rho_0 = 0$. In the following coding I use $\mathbf{x}(1) = y_3$, $\mathbf{y}(1) = y_1$ and $\mathbf{y}(2) = y_2$; there are no unstable modes. Also the fluctuations $\epsilon\rho(t)$ are represented in the input by $\mathbf{w}(\mathbf{rho})$ whereas in the output it is represented by $\sigma\mathbf{w}_\rho$, and similarly for the other fluctuating quantities. Note that the algorithm automatically multiplies time varying quantities by the ‘small’ parameter σ (distinct from the σ in the Lorenz system!) corresponding to their ϵ .

```
143 if thecase=sdePRLorenz then begin
144   sig0:=1; bet0:=1;
145   sig1:=sig0/(sig0+1);
146   xrhs := {sig0*sig1*y(1)*y(2)-sig1*x(1)*y(2)
147     +sig1*x(1)*w(rho)
148     +(w(sigma)-w(rho)/(sig0+1))*y(1)}$
149   yrhs := { -(sig0+1)*y(1)+sig1*y(1)*y(2)-x(1)*y(2)/(sig0+1)
150     +w(rho)/(sig0+1)*x(1)
```

```

151      -(w(sigma)+w(rho)/(sig0+1))*y(1)
152      , -bet0*y(2)-sig0*y(1)^2+(1-sig0)*x(1)*y(1)+x(1)^2
153      -w(beta)*y(2)
154      }$
155  zrhs := {}$
156  factor small,sigma,yy,y,xx,x;
157  toosmall:=3;
158  end;

```

For the general computer algebra I set σ_0 and β_0 to some definite values, here $\sigma_0 = \beta_0 = 1$.

The specified dynamical system

$$\begin{aligned}
 \dot{x}_1 &= -1/2x_1y_2\varepsilon + 1/2x_1\sigma w_\rho + 1/2y_2y_1\varepsilon + y_1\sigma(-1/2w_\rho + w_\sigma) \\
 \dot{y}_1 &= -1/2x_1y_2\varepsilon + 1/2x_1\sigma w_\rho + 1/2y_2y_1\varepsilon + y_1\sigma(-1/2w_\rho - w_\sigma) - 2y_1 \\
 \dot{y}_2 &= x_1^2\varepsilon - y_2\sigma w_\beta - y_2 - y_1^2\varepsilon
 \end{aligned}$$

The stochastic coordinate transform

$$\begin{aligned}
 y_1 &= X_1Y_2\sigma(-1/2e^{-1t}\star w_\beta + e^{-1t}\star w_\rho - 1/4e^{-2t}\star w_\rho + 1/2e^{-1t}\star w_\sigma) - \\
 &1/2X_1Y_2 + 1/2X_1\sigma e^{-2t}\star w_\rho + Y_2Y_1\sigma(1/2e^{1t}\star w_\beta - 1/4e^{2t}\star w_\rho + 1/3e^t\star w_\rho + \\
 &1/2e^{2t}\star w_\sigma - 1/2e^t\star w_\sigma) - 1/2Y_2Y_1 + Y_1 \\
 y_2 &= X_1^2\sigma(-e^{-1t}\star w_\beta - e^{-1t}\star w_\rho) + X_1^2 + X_1Y_1\sigma(e^{2t}\star w_\rho - 2/3e^t\star w_\rho + \\
 &1/3e^{-2t}\star w_\rho - 2e^{2t}\star w_\sigma + 2e^{1t}\star w_\sigma) + Y_2 + Y_1^2\sigma(1/3e^{3t}\star w_\beta - 1/3e^{3t}\star w_\rho - \\
 &2/3e^{3t}\star w_\sigma) + 1/3Y_1^2 \\
 x_1 &= X_1Y_2\sigma(-1/2e^t\star w_\beta - 1/3e^t\star w_\rho - 1/12e^{-2t}\star w_\rho + 1/2e^t\star w_\sigma) + 1/2X_1Y_2 + \\
 &X_1 + Y_2Y_1\sigma(1/6e^{3t}\star w_\beta - 1/3e^{3t}\star w_\rho + 1/4e^{2t}\star w_\rho + 7/6e^{3t}\star w_\sigma - 1/2e^{2t}\star w_\sigma) - \\
 &1/6Y_2Y_1 + Y_1\sigma(1/2e^{2t}\star w_\rho - e^{2t}\star w_\sigma)
 \end{aligned}$$

Result normal form SDEs

$$\begin{aligned} \dot{Y}_1 = & X_1^2 Y_1 \sigma^2 (1/2 e^{2t} \star e^{2t} \star w_\rho w_\rho + 1/4 e^{-2t} \star e^{-2t} \star w_\rho w_\rho - 1/2 e^{-2t} \star e^{-2t} \star w_\rho w_\sigma - \\ & e^{2t} \star e^{2t} \star w_\sigma w_\rho + 1/12 e^{2t} \star w_\beta w_\rho + 1/2 e^{-1t} \star w_\beta w_\beta + 1/3 e^{-1t} \star w_\beta w_\rho - 1/2 e^{-1t} \star w_\beta w_\sigma - \\ & 3/4 e^{2t} \star w_\rho w_\beta + 13/48 e^{2t} \star w_\rho w_\rho + 3/8 e^{2t} \star w_\rho w_\sigma + 1/3 e^t \star w_\rho w_\beta - 2/3 e^t \star w_\rho w_\rho - \\ & 1/3 e^t \star w_\rho w_\sigma + 1/2 e^{-1t} \star w_\rho w_\beta + 1/3 e^{-1t} \star w_\rho w_\rho - 1/2 e^{-1t} \star w_\rho w_\sigma - 1/6 e^{-2t} \star w_\rho w_\beta + \\ & 7/12 e^{-2t} \star w_\rho w_\rho - 7/12 e^{-2t} \star w_\rho w_\sigma + 3/2 e^{2t} \star w_\sigma w_\beta - 3/8 e^{2t} \star w_\sigma w_\rho - 3/4 e^{2t} \star w_\sigma w_\sigma - \\ & e^t \star w_\sigma w_\beta + 2e^t \star w_\sigma w_\rho + e^t \star w_\sigma w_\sigma) + X_1^2 Y_1 \sigma (-1/2 w_\beta - 3/4 w_\rho - 1/4 w_\sigma) + \\ & 1/2 X_1^2 Y_1 + X_1 Y_2^2 \sigma^2 (-1/8 e^{2t} \star e^{2t} \star w_\rho w_\rho + 1/4 e^{2t} \star e^{2t} \star w_\sigma w_\rho - 1/12 e^{2t} \star w_\beta w_\rho + \\ & 1/4 e^t \star w_\beta w_\beta - 1/2 e^{1t} \star w_\beta w_\rho - 1/4 e^t \star w_\beta w_\sigma + 1/4 e^{-1t} \star w_\beta w_\beta + 1/6 e^{-1t} \star w_\beta w_\rho - \\ & 1/4 e^{-1t} \star w_\beta w_\sigma + 1/96 e^{2t} \star w_\rho w_\rho + 1/6 e^t \star w_\rho w_\beta - 1/3 e^t \star w_\rho w_\rho - 1/6 e^t \star w_\rho w_\sigma - \\ & 1/2 e^{-1t} \star w_\rho w_\beta - 1/3 e^{-1t} \star w_\rho w_\rho + 1/2 e^{-1t} \star w_\rho w_\sigma - 1/12 e^{-2t} \star w_\rho w_\beta + 1/96 e^{-2t} \star w_\rho w_\sigma - \\ & 1/6 e^{-2t} \star w_\rho w_\sigma - 1/6 e^{2t} \star w_\sigma w_\rho - 1/4 e^t \star w_\sigma w_\beta + 1/2 e^t \star w_\sigma w_\rho + 1/4 e^t \star w_\sigma w_\sigma - \\ & 1/4 e^{-1t} \star w_\sigma w_\beta - 1/6 e^{-1t} \star w_\sigma w_\rho + 1/4 e^{-1t} \star w_\sigma w_\sigma) + 3/4 X_1 Y_2^2 \sigma w_\rho - 1/2 X_1 Y_2^2 + \\ & Y_1 \sigma^2 (1/4 e^{2t} \star w_\rho w_\rho - 1/2 e^{2t} \star w_\sigma w_\rho) + Y_1 \sigma (-1/2 w_\rho - w_\sigma) - 2Y_1 \end{aligned}$$

$$\begin{aligned} \dot{Y}_2 = & X_1^2 Y_2 \sigma^2 (e^t \star w_\beta w_\beta + e^t \star w_\beta w_\rho - 1/3 e^{-1t} \star w_\beta w_\rho + e^{-1t} \star w_\beta w_\sigma + 2/3 e^t \star w_\rho w_\beta + \\ & 2/3 e^t \star w_\rho w_\rho + 2/3 e^{-1t} \star w_\rho w_\rho - 2e^{-1t} \star w_\rho w_\sigma + 1/3 e^{-2t} \star w_\rho w_\beta - 13/24 e^{-2t} \star w_\rho w_\rho + \\ & 5/6 e^{-2t} \star w_\rho w_\sigma - e^t \star w_\sigma w_\beta - e^t \star w_\sigma w_\rho + 1/3 e^{-1t} \star w_\sigma w_\rho - e^{-1t} \star w_\sigma w_\sigma) + \\ & X_1^2 Y_2 \sigma (-w_\beta - 1/2 w_\rho + w_\sigma) + X_1^2 Y_2 - Y_2 \sigma w_\beta - Y_2 \end{aligned}$$

$$\begin{aligned} \dot{X}_1 = & X_1^3 \sigma^2 (-1/4 e^{-2t} \star e^{-2t} \star w_\rho w_\rho + 1/2 e^{-2t} \star e^{-2t} \star w_\rho w_\sigma - 1/2 e^{-1t} \star w_\beta w_\beta - \\ & 1/3 e^{-1t} \star w_\beta w_\rho + 1/2 e^{-1t} \star w_\beta w_\sigma + 1/4 e^{-2t} \star w_\beta w_\rho - 1/2 e^{-2t} \star w_\beta w_\sigma - 1/2 e^{-1t} \star w_\rho w_\beta - \\ & 1/3 e^{-1t} \star w_\rho w_\rho + 1/2 e^{-1t} \star w_\rho w_\sigma - 1/12 e^{-2t} \star w_\rho w_\beta - 13/48 e^{-2t} \star w_\rho w_\rho + \\ & 3/8 e^{-2t} \star w_\rho w_\sigma - 1/8 e^{-2t} \star w_\sigma w_\rho + 1/4 e^{-2t} \star w_\sigma w_\sigma) + X_1^3 \sigma (1/2 w_\beta + 3/4 w_\rho - \\ & 1/4 w_\sigma) - 1/2 X_1^3 + X_1 \sigma^2 (-1/4 e^{-2t} \star w_\rho w_\rho + 1/2 e^{-2t} \star w_\rho w_\sigma) + 1/2 X_1 \sigma w_\rho \end{aligned}$$

In their analysis [Potzsche & Rasmussen \(2006\)](#) explicitly report the last and third-to-last terms above, for these choices of σ_0 and β_0 , to deduce their model (5.3) which here is

$$\dot{X} \approx \frac{1}{2} \sigma w_\rho X - \frac{1}{2} X^3.$$

Nice agreement.

2.13.2 Fluctuating kdV example

[Potzsche & Rasmussen \(2006\)](#) [Example 5.4] seek travelling wave solutions, $u(x - ct)$ with wave speed c , of a modified KdV equation. This leads to the

following system

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = x_3, \quad \dot{x}_3 = c^2 x_2 - a(t) x_1^2 x_2.$$

My analysis is for wave speed $c^2 = 1$. A transform to diagonalise the linear part into slow variable x , stable y and unstable z is then that $x_1 = x + y + z$, $x_2 = z - y$ and $x_3 = z + y$.

```

159 if thecase=sdePRKdV then begin
160 afn:=w(a)*(x(1)+y(1)+z(1))^2*(z(1)-y(1));
161 xrhs := { afn }$
162 yrhs := { -y(1)-afn/2 }$
163 zrhs := { +z(1)-afn/2 }$
164 factor small,sigma,zz,z,yy,y,xx,x;
165 toosmall:=3;
166 end;
```

Using $w(a)$ to denote the variable coefficient $a(t)$, it is represented in this output by σw_a .

The specified dynamical system

$$\dot{x}_1 = -x_1^2 y_1 \sigma w_a + x_1^2 z_1 \sigma w_a - 2x_1 y_1^2 \sigma w_a + 2x_1 z_1^2 \sigma w_a - y_1^3 \sigma w_a - y_1^2 z_1 \sigma w_a + y_1 z_1^2 \sigma w_a + z_1^3 \sigma w_a$$

$$\dot{y}_1 = 1/2 x_1^2 y_1 \sigma w_a - 1/2 x_1^2 z_1 \sigma w_a + x_1 y_1^2 \sigma w_a - x_1 z_1^2 \sigma w_a + 1/2 y_1^3 \sigma w_a + 1/2 y_1^2 z_1 \sigma w_a - 1/2 y_1 z_1^2 \sigma w_a - y_1 - 1/2 z_1^3 \sigma w_a$$

$$\dot{z}_1 = 1/2 x_1^2 y_1 \sigma w_a - 1/2 x_1^2 z_1 \sigma w_a + x_1 y_1^2 \sigma w_a - x_1 z_1^2 \sigma w_a + 1/2 y_1^3 \sigma w_a + 1/2 y_1^2 z_1 \sigma w_a - 1/2 y_1 z_1^2 \sigma w_a - 1/2 z_1^3 \sigma w_a + z_1$$

Time dependent coordinate transform

$$z_1 = -1/2 X_1^2 Y_1 \sigma e^{2t} \star w_a - X_1 Y_1^2 \sigma e^{3t} \star w_a - X_1 Z_1^2 \sigma e^{-1t} \star w_a - 1/2 Y_1^3 \sigma e^{4t} \star w_a - 1/2 Y_1^2 Z_1 \sigma e^{2t} \star w_a - 1/2 Z_1^3 \sigma e^{-2t} \star w_a + Z_1$$

$$y_1 = -1/2 X_1^2 Z_1 \sigma e^{-2t} \star w_a - X_1 Y_1^2 \sigma e^t \star w_a - X_1 Z_1^2 \sigma e^{-3t} \star w_a - 1/2 Y_1^3 \sigma e^{2t} \star w_a - 1/2 Y_1 Z_1^2 \sigma e^{-2t} \star w_a + Y_1 - 1/2 Z_1^3 \sigma e^{-4t} \star w_a$$

$$x_1 = X_1^2 Y_1 \sigma e^t \star w_a + X_1^2 Z_1 \sigma e^{-1t} \star w_a + 2X_1 Y_1^2 \sigma e^{2t} \star w_a + 2X_1 Z_1^2 \sigma e^{-2t} \star w_a + X_1 + Y_1^3 \sigma e^{3t} \star w_a + Y_1^2 Z_1 \sigma e^t \star w_a + Y_1 Z_1^2 \sigma e^{-1t} \star w_a + Z_1^3 \sigma e^{-3t} \star w_a$$

Putting $Z_1 = 0$ into the coordinate transform gives the centre-stable manifold. Then the expression for z_1 in the above coordinate transform leads to the same convolutions as those of [Potzsche & Rasmussen \(2006\)](#) [pp.453–4]. Conversely, putting $Y_1 = 0$ gives the centre-unstable manifold and the expression for y_1 above leads to the same convolutions as those of [Potzsche & Rasmussen \(2006\)](#). Presumably the distortions of the other variables have a higher order influence on this nice agreement.

2.14 Local analysis of heat exchanger

[Roberts \(2013\)](#) provides novel theoretical support for the method of multiple scales in spatio-temporal systems, and then extends this important method. Perhaps the simplest example is the heat exchanger: the non-autonomous slow manifold analysis that is at the heart of the novel methodology is determined here. Expand advection-exchange in a heat exchanger in powers of $(x-X)^n/n!$. With Taylor Remainder Theorem closing the problem in terms of unknown functions which here are represented by the non-autonomous forcing w_i . Note that $y(j) = d_{j-1}$ and $x(j) = c_{j-1}$. Also $w(1) = d_{4X}\eta_x$ and $w(2) = c_{4X}\xi_x$ and evaluate at intensity $\sigma = 5$.

```

167 if thecase=sdehe then begin
168   xrhs:={y(2),y(3),y(4),y(5),w(1)};
169   yrhs:={-y(1)+x(2),-y(2)+x(3),-y(3)+x(4),-y(4)+x(5),-y(5)+w(2)};
170   zrhs:={ };
171   toosmall:=6;
172   factor small,sigma;
173 end;
```

Specified dynamical system The above writes the ODEs as the following.
 $\dot{x}_1 = \varepsilon y_2, \dot{x}_2 = \varepsilon y_3, \dot{x}_3 = \varepsilon y_4, \dot{x}_4 = \varepsilon y_5, \dot{x}_5 = \sigma w_1, \dot{y}_1 = \varepsilon x_2 - y_1, \dot{y}_2 = \varepsilon x_3 - y_2, \dot{y}_3 = \varepsilon x_4 - y_3, \dot{y}_4 = \varepsilon x_5 - y_4, \dot{y}_5 = \sigma w_2 - y_5.$

Time dependent coordinate transform $y_1 = \sigma \varepsilon^4 (e^{-1t} \star e^{-1t} \star e^{-1t} \star w_2 + 2e^{-1t} \star e^{-1t} \star w_2 + 3e^{-1t} \star w_2) - \varepsilon^3 X_4 + \varepsilon X_2 + Y_1$, $y_2 = \sigma \varepsilon^3 (e^{-1t} \star e^{-1t} \star w_1 + 2e^{-1t} \star w_1) - \varepsilon^3 X_5 + \varepsilon X_3 + Y_2$, $y_3 = \sigma \varepsilon^2 (-e^{-1t} \star e^{-1t} \star w_2 - e^{-1t} \star w_2) + \varepsilon X_4 + Y_3$, $y_4 = -\sigma \varepsilon e^{-1t} \star w_1 + \varepsilon X_5 + Y_4$, $y_5 = \sigma e^{-1t} \star w_2 + Y_5$.

And the slow variables $x_1 = \sigma \varepsilon^4 (-e^{-1t} \star e^{-1t} \star w_1 - 3e^{-1t} \star w_1) + \varepsilon^3 Y_4 - \varepsilon Y_2 + X_1$, $x_2 = \sigma \varepsilon^3 (e^{-1t} \star e^{-1t} \star w_2 + 2e^{-1t} \star w_2) + \varepsilon^3 Y_5 - \varepsilon Y_3 + X_2$, $x_3 = \sigma \varepsilon^2 e^{-1t} \star w_1 - \varepsilon Y_4 + X_3$, $x_4 = -\sigma \varepsilon e^{-1t} \star w_2 - \varepsilon Y_5 + X_4$, $x_5 = X_5$.

Result normal form DEs $\dot{Y}_1 = \varepsilon^4 Y_5 - \varepsilon^2 Y_3 - Y_1$, $\dot{Y}_2 = -\varepsilon^2 Y_4 - Y_2$, $\dot{Y}_3 = -\varepsilon^2 Y_5 - Y_3$, $\dot{Y}_4 = -Y_4$, $\dot{Y}_5 = -Y_5$.

$\dot{X}_1 = 3\sigma \varepsilon^4 w_1 - \varepsilon^4 X_5 + \varepsilon^2 X_3$, $\dot{X}_2 = -2\sigma \varepsilon^3 w_2 + \varepsilon^2 X_4$, $\dot{X}_3 = -\sigma \varepsilon^2 w_1 + \varepsilon^2 X_5$, $\dot{X}_4 = \sigma \varepsilon w_2$, $\dot{X}_5 = \sigma w_1$.

2.14.1 Near the boundary

This is for the case of boundary conditions $c + p d = c d_0(t)$ at $x = 0$ for some parameter p . Computer algebra finds boundary conditions on the fields that reduce the dynamics near the boundary to the following with $x(1) = c_1$, $x(2) = c_3$, $y(1) = d_0$, $y(2) = d_2$ and $w(1) = d_3 \chi \eta_x$ with $\sigma = 4$. Curiously, there is no dependence upon parameter p in these dynamics.

```

174 if thecase=sdehebc then begin
175   xrhs:={y(2),w(1)};
176   yrhs:={-y(1)+x(1),-y(2)+x(2)};
177   zrhs:={ };
178   toosmall:=6;
179   factor small,sigma;
180 end;
```

Again, I believe the following results are exact.

Specified dynamical system $\dot{x}_1 = \varepsilon y_2$, $\dot{x}_2 = \sigma w_1$, $\dot{y}_1 = \varepsilon x_1 - y_1$, $\dot{y}_2 = \varepsilon x_2 - y_2$.

Time dependent coordinate transform $y_1 = \sigma \varepsilon^3 (e^{-1t} \star e^{-1t} \star w_1 + 2e^{-1t} \star w_1) - \varepsilon^3 X_2 + \varepsilon X_1 + Y_1$, $y_2 = -\sigma \varepsilon e^{-1t} \star w_1 + \varepsilon X_2 + Y_2$, $x_1 = \sigma \varepsilon^2 e^{-1t} \star w_1 - \varepsilon Y_2 + X_1$, $x_2 = X_2$.

Result normal form DEs $\dot{Y}_1 = -\varepsilon^2 Y_2 - Y_1$, $\dot{Y}_2 = -Y_2$, $\dot{X}_1 = -\sigma \varepsilon^2 w_1 + \varepsilon^2 X_2$, $\dot{X}_2 = \sigma w_1$.

2.14.2 Heat exchanger with quadratic reaction

Expand advection-reaction-exchange in a heat exchanger in powers of $(x - X)^n/n!$. The reaction is some quadratic that should generate Burgers' equation model. With Taylor Remainder Theorem closing the problem in terms of unknown functions which here are represented by the non-autonomous forcing w_i . Note that $y(j) = d_{j-1}$ and $x(j) = c_{j-1}$. Also $w(1) = 3d_{2x}$ and $w(2) = 3c_{2x}$ and evaluate at intensity $\sigma = 1$.

```

181 if thecase=sdeheqr then begin
182   xrhs:={y(2)-x(1)*y(1)
183         ,y(3)-x(1)*y(2)-x(2)*y(1)
184         ,small*w(1)-x(1)*y(3)-2*x(2)*y(2)-x(3)*y(1)
185         };
186   yrhs:={-y(1)+x(2)-(x(1)^2+y(1)^2)/2
187         ,-y(2)+x(3)-x(1)*x(2)-y(1)*y(2)
188         ,-y(3)+small*w(2)-x(2)^2-x(1)*x(3)-y(2)^2-y(1)*y(3)
189         };
190   zrhs:={ };
191   toosmall:=4;
192   factor small,sigma;
193 end;
```

Alternatively, we could divide the off-diagonal linear terms by `small` (and remove the multiplication of forcing `w`), and the algorithm still converges, albeit in more iterations. The resulting asymptotic expressions then do not assume that x derivatives are successively smaller.

The following uses the default scaling which corresponds to successively smaller ϵ -derivatives provided I also multiply the forcing by **small**.

Specified dynamical system $\dot{x}_1 = \epsilon(-x_1 y_1 + y_2)$, $\dot{x}_2 = \epsilon(-x_2 y_1 - x_1 y_2 + y_3)$, $\dot{x}_3 = \sigma \epsilon w_1 + \epsilon(-x_3 y_1 - 2x_2 y_2 - x_1 y_3)$, $\dot{y}_1 = \epsilon(x_2 - 1/2x_1^2 - 1/2y_1^2) - y_1$, $\dot{y}_2 = \epsilon(x_3 - x_2 x_1 - y_2 y_1) - y_2$, $\dot{y}_3 = \sigma \epsilon w_2 + \epsilon(-x_3 x_1 - x_2^2 - y_3 y_1 - y_2^2) - y_3$

Time dependent coordinate transform

$$y_1 = 1/4\epsilon^2 Y_1^3 + \epsilon(X_2 - 1/2X_1^2 + 1/2Y_1^2) + Y_1$$

$$y_2 = -\sigma\epsilon^2 e^{-1t} \star w_1 + 3/4\epsilon^2 Y_2 Y_1^2 + \epsilon(X_3 - X_2 X_1 + Y_2 Y_1) + Y_2$$

$$y_3 = \sigma\epsilon^2(e^{-1t} \star w_2 Y_1 + e^{-1t} \star w_1 X_1) + \sigma\epsilon e^{-1t} \star w_2 + \epsilon^2(3/4Y_3 Y_1^2 + 3/2Y_2^2 Y_1) + \epsilon(-X_3 X_1 - X_2^2 + Y_3 Y_1 + Y_2^2) + Y_3$$

$$x_1 = \epsilon^2(3/4X_1 Y_1^2 - Y_2 Y_1) + \epsilon(X_1 Y_1 - Y_2) + X_1$$

$$x_2 = -\sigma\epsilon^2 e^{-1t} \star w_2 + \epsilon^2(3/4X_2 Y_1^2 + 3/2X_1 Y_2 Y_1 - Y_3 Y_1 - Y_2^2) + \epsilon(X_2 Y_1 + X_1 Y_2 - Y_3) + X_2$$

$$x_3 = \sigma\epsilon^2(e^{-1t} \star w_2 X_1 + e^{1t} \star w_1 Y_1) + \epsilon^2(3/4X_3 Y_1^2 + 3X_2 Y_2 Y_1 + 3/2X_1 Y_3 Y_1 + 3/2X_1 Y_2^2 - 3/2Y_3 Y_2) + \epsilon(X_3 Y_1 + 2X_2 Y_2 + X_1 Y_3) + X_3$$

Result normal form DEs

$$\dot{Y}_1 = \epsilon^2(-1/2X_1^2 Y_1 + 2X_1 Y_2 - Y_3) - Y_1$$

$$\dot{Y}_2 = 2\sigma\epsilon^3 w_1 Y_1 + \epsilon^2(-X_2 X_1 Y_1 + 2X_2 Y_2 - 1/2X_1^2 Y_2 + 2X_1 Y_3) - Y_2$$

$$\dot{Y}_3 = \sigma\epsilon^3(-2w_1 X_1 Y_1 + 2w_1 Y_2) - \sigma\epsilon^2 w_2 Y_1 + \epsilon^2(-X_3 X_1 Y_1 - X_3 Y_2 - X_2^2 Y_1 - 2X_2 X_1 Y_2 + X_2 Y_3 - 1/2X_1^2 Y_3) - Y_3$$

$$\dot{X}_1 = -\sigma\epsilon^3 w_1 + \epsilon^2(X_3 - 2X_2 X_1 + 1/2X_1^3)$$

$$\dot{X}_2 = 2\sigma\epsilon^3 w_1 X_1 + \sigma\epsilon^2 w_2 + \epsilon^2(-2X_3 X_1 - 2X_2^2 + 3/2X_2 X_1^2)$$

$$\dot{X}_3 = \sigma\epsilon^3(2w_1 X_2 - w_1 X_1^2) - \sigma\epsilon^2 w_2 X_1 + \sigma\epsilon w_1 + \epsilon^2(-3X_3 X_2 + 3/2X_3 X_1^2 + 3X_2^2 X_1)$$

Hmmm, looks like this generates the slowly varying model that

$$\frac{\partial C}{\partial t} \approx \frac{\partial^2 C}{\partial x^2} - 2C \frac{\partial C}{\partial x} + \frac{1}{2} C^3.$$

Interestingly there is an extra factor of two in the nonlinear advection, and a net cubic reaction.

2.15 Michaelis–Menten–Henri deterministic model

$$\begin{aligned}\dot{x} &= \epsilon[-x + (x + \kappa - \lambda)y], \\ \dot{y} &= x - (x + \kappa)y.\end{aligned}$$

A manifold of equilibria occur at $y = x/(x + \kappa)$ and $\epsilon = 0$ (also if $\epsilon \neq 0$ and $\lambda = 0$ but we do not consider this case). Let's explore dynamics based at arbitrary point on this equilibrium manifold: substitute $x(t) = x_0 + x_1(t)$ and $y(t) = x_0/(x_0 + \kappa) + y_1(t)$, and derive

$$\begin{aligned}\frac{1}{x_0 + \kappa} \dot{x}_1 &= x'_1 = \frac{\epsilon}{x_0 + \kappa} [-x_0 - x_1 + (x_0 + x_1 + \kappa - \lambda)(x_0/(x_0 + \kappa) + y_1)] \\ \frac{1}{x_0 + \kappa} \dot{y}_1 &= y'_1 = -y_1 + \frac{1}{x_0 + \kappa} \left[-x_1 y_1 + x_0 + x_1 - \frac{(x_0 + x_1 + \kappa)x_0}{x_0 + \kappa} \right]\end{aligned}$$

We need the above form in order to get the decay rate of y_1 to be a simple number—but it does mean that we have to be careful interpreting the results. The reason is that the time derivative in the analysis is stretched by the factor $x_0 + \kappa$: that is, $(x_0 + \kappa)dt = d\tau$ where τ is the time of the analysis. But have not yet put in stochastic effects.

```

194 if thecase=sdemmh then begin
195 % define rho=1/(x0+kappa)
196 let rho*x0=>1-rho*kappa;
197 lam:=lambda;
198 xrhs:={eps*rho*(-x0-x(1)+(x0+x(1)+kappa-lam)*(x0*rho+y(1)))};
199 yrhs:={-y(1)+rho*(-x(1)*y(1)+x0+x(1)-(x0+x(1)+kappa)*x0*rho)};
200 zrhs:={ };
```

```

201 toosmall:=4;
202 factor eps,sigma;
203 end;

```

In the deterministic results, $Y_1 \rightarrow 0$ to form the slow manifold. Then the x -evolution is modelled by simply putting $X_1 = 0$ in the right-hand side of \dot{X}_1 . This gets rid of a lot of terms. Justify??

3 General SDE preliminaries

Deterministic, autonomous, normal forms are constructed simply by omitting any noise term $w()$ in the differential equations.

The right-hand sides must be multinomial in variables x_i , y_i , z_i and w_i , but off-hand I do not know an easy way to check for this.

Improve appearance Improve appearance of printed output.

```

204 on div; off allfac; on revpri;
205 linelength 70$

```

If for the web, then send text output to file

```

206 if thecase=webpage then out "sdeo.txt"$

```

3.1 Extract and scale slow equations

The number of slow equations is the number of terms in the list in `xrhs`.

```

207 write "no. of slow modes ",m:=length(xrhs);

```

Multiply all the right-hand sides by `small` so we can control the truncation of the asymptotic construction through discarding high powers of `small`. Users could use `small` in their equations for appropriate effects.

```
208 xrhs:=for i:=1:m collect small*part(xrhs,i)$
```

Adjust the noise terms. Remove the `small` multiplication of noise terms, and instead multiply by `sigma` to empower me to independently control the truncation in noise amplitude.

```
209 xrhs:=(xrhs where w(~j)=>sigma*w(j,1)/small)
210           where w(~j,1)=>w(j))$
```

Section 5 writes the resulting differential equations for information.

3.2 Extract and scale stable fast equations

The number of stable fast equations is the number of terms in the list in `yrhs`.

```
211 write "no. of stable fast modes ",ny:=length(yrhs);
```

Extract decay rates Extract the linear decay rates of the fast equations into an array. For each expression in the provided set of right-hand sides:

```
212 array rate(ny);
213 for i:=1:ny do begin
```

For the i th right-hand side get the linear dependence upon `y(i)`, then set other dynamic variables to zero to get just the coefficient.

```
214   rate(i):=coeffn(part(yrhs,i),y(i),1);
215   rate(i):=(rate(i) where {x(~j)=>0,y(~j)=>0,z(~j)=>0,w(~j)=>0})
```

However, the coefficient may depend upon parameters, so if it is not simply a number, but is a sum, then trawl through the sum looking for a simple number to use as the decay rate.

```
216   if not numberp(rate(i)) then
217     if part(rate(i),0)=plus then begin
218       rr:=0;
219       for j:=1:arglength(rate(i)) do
220         if numberp(part(rate(i),j))
```

```

221         then rr:=part(rate(i),j);
222         rate(i):=rr;
223     end;

```

Change sign to make **rate** into positive decay rates, rather than negative growth rates.

```

224     rate(i):=-rate(i);

```

If all the above has not ended up with a simple number, then exit with an error message.

```

225     if numberp(rate(i))and rate(i)>0 then
226     else begin
227         write "***** Error *****
228         Linear coeffs of y-decay must be negative numbers";
229         if thecase=wbepage then <<
230             shut "sdeo.txt"; quit >>;
231     end;

```

End the loop over all right-hand sides.

```

232 end;

```

Flag later warning if the linear part not diagonal.

```

233 offdiag:=0$
234 for i:=1:ny do for j:=1:ny do if i neq j then begin
235     jac:=coeffn(part(yrhs,i),y(j),1);
236     if (jac where {x(~k)=>0,y(~k)=>0,z(~k)=>0,w(~k)=>0}) neq 0
237     then offdiag:=1$
238 end;

```

Multiply all the ‘nonlinear’ terms right-hand sides by **small** so we control the truncation of the asymptotic construction through discarding high powers of **small**. Leave the identified linear decay terms intact. Users could use **small** in their equations for interesting effects.

```

239 yrhs:=for i:=1:ny collect
240     small*part(yrhs,i)+(1-small)*(-rate(i)*y(i))$

```

Remove the `small` multiplication of noise terms, and instead multiply by `sigma` to empower me to independently control the truncation in noise amplitude.

```
241 yrhs:=(yrhs where w(~j)=>sigma*w(j,1)/small)
242   where w(~j,1)=>w(j))$
```

Section 5 writes the resulting differential equations for information.

3.3 Extract and scale unstable fast equations

The number of unstable fast equations is the number of terms in the list in `zrhs`.

```
243 write "no. of unstable fast modes ",nz:=length(zrhs);
```

Extract decay rates Extract the linear decay rates of the fast equations into an array. For each expression in the provided set of right-hand sides:

```
244 array ratf(nz);
245 for i:=1:nz do begin
```

For the i th right-hand side get the linear dependence upon `z(i)`, then set other dynamic variables to zero to get just the coefficient.

```
246   ratf(i):=coeffn(part(zrhs,i),z(i),1);
247   ratf(i):=(ratf(i) where {x(~j)=>0,y(~j)=>0,z(~j)=>0,w(~j)=>0})
```

However, the coefficient may depend upon parameters, so if it is not simply a number, but is a sum, then trawl through the sum looking for a simple number to use as the decay rate.

```
248   if not numberp(ratf(i)) then
249     if part(ratf(i),0)=plus then begin
250       rr:=0;
251       for j:=1:arglength(ratf(i)) do
252         if numberp(part(ratf(i),j))
253           then rr:=part(ratf(i),j);
```



```

254     ratf(i):=rr;
255 end;

```

If all the above has not ended up with a simple number, then exit with an error message.

```

256 if numberp(ratf(i))and ratf(i)>0 then
257 else begin
258     write "***** Error *****
259     Linear coeffs of z-growth must be positive numbers";
260     if thecase=webpage then <<
261         shut "sdeo.txt"; quit >>;
262 end;

```

End the loop over all right-hand sides.

```

263 end;

```

Flag warning if the linear part not diagonal.

```

264 for i:=1:nz do for j:=1:nz do if i neq j then begin
265     jac:=coeffn(part(zrhs,i),z(j),1);
266     if (jac where {x(~k)=>0,y(~k)=>0,z(~k)=>0,w(~k)=>0}) neq 0
267     then offdiag:=1$
268 end;

```

Multiply all the ‘nonlinear’ terms right-hand sides by `small` so we control the truncation of the asymptotic construction through discarding high powers of `small`. Leave the identified linear decay terms intact. Users could use `small` in their equations for interesting effects.

```

269 zrhs:=for i:=1:nz collect
270     small*part(zrhs,i)+(1-small)*(+ratf(i)*z(i))$

```

Remove the `small` multiplication of noise terms, and instead multiply by `sigma` to empower me to independently control the truncation in noise amplitude.

```

271 zrhs:=((zrhs where w(~j)=>sigma*w(j,1)/small)
272     where w(~j,1)=>w(j))$

```

Section 5 writes the resulting differential equations for information.

Turn off output to file while writing L^AT_EX.

```
273 if thecase=webpage then shut "sdeo.txt"$
```

4 Setup LaTeX output using rlfi

Now setup the rlfi package to write a L^AT_EX version of the output. It is all a bit tricky and underhand, so hope it works. We override some stuff from `rlfi.red`.⁵

Override some rlfi things First, change `name` to get Big delimiters, not left-right delimiters, so L^AT_EX can break lines.

```
274 deflist('(( ( !\b!i!g!() (!) !\b!i!g!)) (!P!I !\p!i! )
275           (!p!i !\p!i! ) (!E !e) (!I !i) (e !e) (i !i)), 'name)$
```

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.

```
276 symbolic procedure prinlaend;
277 <<terpri();
278   prin2 "\end{";
279   prin2 mstyle!*;
280   prin2t "}\par";
281   if !*verbatim then
282     <<prin2t "\begin{verbatim}";
283     prin2t "REDUCE Input:">>;
284   ncharspr!*:=0;
285   if ofl!* then linelength(car linel!*)
286     else laline!*:=cdr linel!*;
287   nochar!*:=append(nochar!*,nochar1!*);
```

⁵Find it in `reduce-algebra/trunk/packages/misc/rlfi.red`

```
288 nochar1!*:=nil >>$
```

Override the procedure that outputs the \LaTeX preamble upon the command `on latex`.

```
289 symbolic procedure latexon;
290 <<!*!*a2sfn:='texaeval;
291   !*raise:=nil;
292   prin2t "\documentclass[11pt,a5paper]{article}";
293   prin2t "\usepackage[a5paper,margin=13mm]{geometry}";
294   prin2t "\usepackage{parskip,time} \raggedright";
295   prin2t "\def\ou\big(#1,#2,#3\big){e^{\if#3\else#3\fi t}\star}";
296   prin2t "\title{Normal form of your dynamical system}";
297   prin2t "\author{A. J. Roberts, University of Adelaide\\}";
298   prin2t "\texttt{http://www.maths.adelaide.edu.au/anthony.roberts}";
299   prin2t "\date{\now, \today}";
300   prin2t "\begin{document}";
301   prin2t "\maketitle";
302   prin2t "\input{sdeo1}";
303   if !*verbatim then
304     <<prin2t "\begin{verbatim}";
305     prin2t "REDUCE Input:">>;
306     put('tex,'rttypefn,'(lambda(x) 'tex)) >>$
```

Use inline math environment so that long lines, the norm, get line breaks. The command `\raggedright` in the \LaTeX preamble appears the best option for the line breaking, but `\sloppy` would also work reasonably.

```
307 mathstyle math;
```

Define names for \LaTeX formatting Define some names I use, so that *rlfi* translates them to Greek characters in the \LaTeX .

```
308 %defid sig,name=sigma;
309 defid eps,name=epsilon;
310 defid small,name=varepsilon;
```

Should not need these translation definitions but somehow we do in order for users to get the Greek alphabet to appear. I am puzzled??

```
311 defid alpha,name=alpha;
312 defid beta,name=beta;
313 defid gamma,name=gamma;
314 defid delta,name=delta;
315 defid epsilon,name=epsilon;
316 defid varepsilon,name=varepsilon;
317 defid zeta,name=zeta;
318 defid eta,name=eta;
319 defid theta,name=theta;
320 defid vartheta,name=vartheta;
321 defid iota,name=iota;
322 defid kappa,name=kappa;
323 defid lambda,name=lambda;
324 defid mu,name=mu;
325 defid nu,name=nu;
326 defid xi,name=xi;
327 defid pi,name=pi;
328 defid varpi,name=varpi;
329 defid rho,name=rho;
330 defid varrho,name=varrho;
331 defid sigma,name=sigma;
332 defid varsigma,name=varsigma;
333 defid tau,name=tau;
334 defid upsilon,name=upsilon;
335 defid phi,name=phi;
336 defid varphi,name=varphi;
337 defid chi,name=chi;
338 defid psi,name=psi;
339 defid omega,name=omega;
340 defid Gamma,name=Gamma;
341 defid Delta,name=Delta;
342 defid Theta,name=Theta;
```

```
343 defid Lambda,name=Lambda;
344 defid Xi,name=Xi;
345 defid Pi,name=Pi;
346 defid Sigma,name=Sigma;
347 defid Upsilon,name=Upsilon;
348 defid Phi,name=Phi;
349 defid Psi,name=Psi;
350 defid Omega,name=Omega;
```

For the variables names I use, as operators, define how they appear in the \LaTeX , and also define that their arguments appear as subscripts.

```
351 defindex w(down);
352 defindex x(down);
353 defindex y(down);
354 defindex z(down);
355 defid xx,name="X";
356 defid yy,name="Y";
357 defid zz,name="Z";
358 defindex xx(down);
359 defindex yy(down);
360 defindex zz(down);
361 defindex hh(down);
362 defindex gg(down);
363 defindex ff(down);
```

First use these for the specified dynamical system, later use them for the normal form equations.

```
364 defid hh,name="\dot z";
365 defid gg,name="\dot y";
366 defid ff,name="\dot x";
```

The Ornstein–Uhlenbeck operator is to translate into a \LaTeX command, see the preamble, that typesets the convolution in a reasonable manner. The definition of the \LaTeX command is a bit dodgy as convolutions of convolutions are not printed in the correct order; however, convolutions

commute so it does not matter.

```
367 defid ou,name="\ou";
368 defindex ou(arg,arg,arg);
```

Write the \LaTeX dynamical system Because of the way `rfi` works, to get good quality output to the \LaTeX document, I need to write the algebraic expressions to a file, then read them back in again. While being read back in, I send the output to the \LaTeX file. In this convoluted way I avoid extraneous output lines polluting the \LaTeX .

Temporarily use these arrays for the right-hand sides of the dynamical system.

```
369 array ff(m),gg(ny),hh(nz);
```

Write expressions to the file `sdeo.red` for later reading. Prepend the expressions with an instruction to write a heading, and surround the heading with anti-math mode to cancel the math environment that `rfi` puts in.

```
370 out "sdeo.red"$
371 write "write ""\end{math}"
372 \paragraph{Specified dynamical system}
373 \begin{math}""$";
374 for i:=1:m do write "ff(",i,"):=1*part(xrhs,",i,");";
375 for i:=1:ny do write "gg(",i,"):=1*part(yrhs,",i,");";
376 for i:=1:nz do write "hh(",i,"):=1*part(zrhs,",i,");";
377 write "end;";
378 shut "sdeo.red";
```

Then switch on \LaTeX output before writing to file as this \LaTeX file is to be input from the main \LaTeX file and hence does not need a header. The header here gets sent to the ‘terminal’ instead. Then write to `sdeo1.tex` the expressions we stored in `sdeo.red` as nice \LaTeX .

```
379 on latex$
380 out "sdeo1.tex"$
381 in "sdeo.red"$
382 shut "sdeo1.tex"$
```

```
383 off latex$
```

5 Delayed write of text info

Because it is messy to interleave L^AT_EX and plain output, I delay writing anything much in plain text until here. Here start writing to the text output file `sdeo.txt`; finish writing to file upon success, or otherwise, of the iteration.

```
384 if thecase=webpage then out "sdeo.txt"$
```

Write the delayed warning message about off-diagonal terms.

```
385 if offdiag then write "  
386 ***** Warning *****  
387 Off diagonal linear terms in y- or z- equations assumed  
388 small.  Answers are rubbish if not asymptotically  
389 appropriate. "$
```

Write the plain text versions of the dynamical system.

```
390 write "no. of slow modes ",m:=length(xrhs);  
391 for i:=1:m do write "dx(",i,")/dt = ",1*part(xrhs,i);  
392 write "no. of stable fast modes ",ny:=length(yrhs);  
393 for i:=1:ny do write "dy(",i,")/dt = ",1*part(yrhs,i);  
394 write "no. of unstable fast modes ",nz:=length(zrhs);  
395 for i:=1:nz do write "dz(",i,")/dt = ",1*part(zrhs,i);
```

6 Represent the noise

The white noises \mathbf{w} depend upon time. But we find it useful to discriminate upon the notionally fast time fluctuations of the noise processes, and the notionally ordinary time variations of the dynamic variables x_i , y_i and z_i . Thus introduce a notionally fast time variable `tt`, which depends upon the ordinary time `t`. Equivalently, view `tt`, a sort of ‘partial `t`’, as representing variations in time independent of those in the variables x_i , y_i and z_i .

```

396 depend w,tt;
397 depend tt,t,tttyz;

```

In the construction, convolutions of the noise arise, both backwards over history and forwards to anticipate the noise. For any non-zero parameter μ , define the Ornstein–Uhlenbeck convolution

$$e^{\mu t} \star \phi = \begin{cases} \int_{-\infty}^t \exp[\mu(t - \tau)] \phi(\tau) d\tau, & \mu < 0, \\ \int_t^{+\infty} \exp[\mu(t - \tau)] \phi(\tau) d\tau, & \mu > 0, \end{cases} \quad (21)$$

so that the convolution is always with a bounded exponential. Five useful properties of this convolution are

$$e^{\mu t} \star 1 = \frac{1}{|\mu|}, \quad (22)$$

$$\frac{d}{dt} e^{\mu t} \star \phi = -\operatorname{sgn} \mu \phi + \mu e^{\mu t} \star \phi, \quad (23)$$

$$E[e^{\mu t} \star \phi] = e^{\mu t} \star E[\phi], \quad (24)$$

$$E[(e^{\mu t} \star \phi)^2] = \frac{1}{2|\mu|}, \quad (25)$$

$$e^{\mu t} \star e^{\nu t} \star = \begin{cases} \frac{1}{|\mu - \nu|} [e^{\mu t} \star + e^{\nu t} \star], & \mu \nu < 0, \\ \frac{-\operatorname{sgn} \mu}{\mu - \nu} [e^{\mu t} \star - e^{\nu t} \star], & \mu \nu > 0 \text{ \& } \mu \neq \nu. \end{cases} \quad (26)$$

Also remember that although with $\mu < 0$ the convolution $e^{\mu t} \star$ integrates over the past, with $\mu > 0$ the convolution $e^{\mu t} \star$ integrates into the future over a time scale of order $1/\mu$.

The operator `ou(f,tt,mu)` represents the convolution $e^{\mu t} \star f$ as defined by (21): called `ou` because it is an Ornstein–Uhlenbeck process. The operator `ou` is ‘linear’ over fast time `tt` as the convolution only arises from solving PDEs in the operator $\partial_t - \mu$. Code its derivative (23) and its action upon deterministic terms (22):

```

398 operator ou; linear ou;
399 let { df(ou(~f,tt,~mu),t)=>-sign(mu)*f+mu*ou(f,tt,mu)
400      , ou(1,tt,~mu)=>1/abs(mu)

```


Also code the transform (26) that successive convolutions at different rates may be transformed into several single convolutions.

```

401      , ou(ou(~r,tt,~nu),tt,~mu) =>
402      (ou(r,tt,mu)+ou(r,tt,nu))/abs(mu-nu) when (mu*nu<0)
403      , ou(ou(~r,tt,~nu),tt,~mu) =>
404      -sign(mu)*(ou(r,tt,mu)-ou(r,tt,nu))/(mu-nu)
405      when (mu*nu>0)and(mu neq nu)
406  };

```

The above properties are *critical*: they must be correct for the results to be correct.

Second, identify the resonant parts, some of which must go into the evolution $\mathbf{gg}(\mathbf{i})$, and some into the transform. It depends upon the exponent of \mathbf{yz} compared to the decay rate of this mode, here \mathbf{r} .

```

407 operator reso; linear reso;
408 let { reso(~a,yz,~r)=>1 when df(a,yz)*yz=r*a
409      , reso(~a,yz,~r)=>0 when df(a,yz)*yz neq r*a
410      };

```

Lastly, the remaining terms get convolved at the appropriate rate to solve their respective homological equation by the operator \mathbf{zres} .

```

411 depend yz,tt,yz;
412 operator zres; linear zres;
413 let zres(~a,tt,yz,~r)=>ou(sign(df(a,yz)*yz/a-r)
414      *sub(yz=1,a),tt,df(a,yz)*yz/a-r);

```

7 Solve homological equation with noise

When solving homological equations of the form $F + \xi_t = \text{Res}$ (the resonant case $\mu = 0$), we separate the terms in the right-hand side Res into those that are integrable in fast time, and hence modify the coordinate transform by changing ξ , and those that are not, and hence must remain in the evolution by changing F . the operator \mathbf{zint} extracts those parts of a term that we know

are integrable; the operator **znon** extracts those parts which are not. Note: with more research, more types of terms may be found to be integrable; hence what is extracted by **zint** and what is left by **zint** may change with more research. These transforms are not critical: changing the transforms may change intermediate computations, but as long as the iteration converges, the computer algebra results will be algebraically correct.

```
415 operator zint; linear zint;
416 operator znon; linear znon;
```

First, avoid obvious secularity.

```
417 let { zint(w(~i),tt)=>0, znon(w(~i),tt)=>w(i)
418 , zint(1,tt)=>0, znon(1,tt)=>1
419 , zint(w(~i)*~r,tt)=>0, znon(w(~i)*~r,tt)=>w(i)*r
```

Second, by (23) a convolution may be split into an integrable part, and a part in its argument which in turn may be integrable or not.

```
420 , zint(ou(~r,tt,~mu),tt)=>ou(r,tt,mu)/mu+zint(r,tt)/abs(mu)
421 , znon(ou(~r,tt,~mu),tt)=>znon(r,tt)/abs(mu)
```

Third, squares of convolutions may be integrated by parts to an integrable term and a part that may have integrable or non-integrable parts.

```
422 , zint(ou(~r,tt,~mu)^2,tt)=>ou(~r,tt,~mu)^2/(2*mu)
423                               +zint(r*ou(r,tt,mu),tt)/abs(mu)
424 , znon(ou(~r,tt,~mu)^2,tt)=>znon(r*ou(r,tt,mu),tt)/abs(mu)
```

Fourth, different products of convolutions may be similarly separated using integration by parts.

```
425 , zint(ou(~r,tt,~mu)*ou(~s,tt,~nu),tt)
426   =>ou(r,tt,mu)*ou(s,tt,nu)/(mu+nu)
427   +zint(sign(mu)*r*ou(s,tt,nu)+sign(nu)*s*ou(r,tt,mu),tt)
428   /(mu+nu) when mu+nu neq 0
429 , znon(ou(~r,tt,~mu)*ou(~s,tt,~nu),tt)=>
430   +znon(sign(mu)*r*ou(s,tt,nu)+sign(nu)*s*ou(r,tt,mu),tt)
431   /(mu+nu) when mu+nu neq 0
```

However, a zero divisor arises when $\mu + \nu = 0$ in the above. Here code rules to cater for such terms by increasing the depth of convolutions over past history.

```

432 , zint(ou(~r,tt,~mu)*ou(~s,tt,~nu),tt)=>
433   ou(ou(r,tt,-nu),tt,-nu)*ou(s,tt,nu)
434   +zint(ou(ou(r,tt,-nu),tt,-nu)*s,tt) when (mu+nu=0)and(nu>0)
435 , znon(ou(~r,tt,~mu)*ou(~s,tt,~nu),tt)=>
436   znon(ou(ou(r,tt,-nu),tt,-nu)*s,tt) when (mu+nu=0)and(nu>0)

```

The above handles quadratic products of convolutions. Presumably, if we seek cubic noise effects then we may need cubic products of convolutions. However, I do not proceed so far and hence terminate the separation rules.

```

437 };

```

8 Initialise approximate transform

Truncate asymptotic approximation of the coordinate transform depending upon the parameter `toosmall`, up to a maximum of six. Use the ‘instant evaluation’ property of a loop index to define the truncation so that Reduce omits small terms on the fly.

```

438 for j:=toosmall:toosmall do let small^j=>0;

```

Variables `x`, `y` and `z` were operators in the specification of the equations. We now want them to store the approximation to the coordinate transform, so clear and reallocate as storage for the normal form expressions.

```

439 clear x,y,z;
440 array x(m),y(ny),z(nz);

```

Express the normal form in terms of new evolving variables X_i, Y_i and Z_i , denoted by operators `xx(i)`, `yy(i)` and `zz(i)`, which are nonlinear modifications to x_i, y_i and z_i . The expressions for the normal form SDEs are stored in `ff`, `gg` and `hh`.

```

441 operator xx; operator yy; operator zz;

```

```

442 depend xx,t; depend yy,t; depend zz,t;
443 let { df(xx(~i),t)=>ff(i)
444      , df(yy(~i),t)=>gg(i)
445      , df(zz(~i),t)=>hh(i) };

```

The first linear approximation is then $x_i \approx X_i$, $y_i \approx Y_i$ and $z_i = Z_i$, such that $\dot{X}_i \approx 0$, in $\mathbf{ff}(i)$, $\dot{Y}_i \approx -r_i Y_i$, in $\mathbf{gg}(i)$, and $\dot{Z}_i \approx +r_i Z_i$, in $\mathbf{hh}(i)$.

```

446 for i:=1:m do begin x(i):=xx(i); ff(i):=0; end;
447 for i:=1:ny do begin y(i):=yy(i); gg(i):=-rate(i)*yy(i); end;
448 for i:=1:nz do begin z(i):=zz(i); hh(i):=+ratf(i)*zz(i); end;

```

Update the Y_i evolution $\mathbf{gg}(i)$ and the y_i transform. The residual is of the form of a sum of terms $\prod_j Y_j^{q_j} Z_k^{r_k} \in \text{Res}$. So updates involve dividing by, or convolving with, $\beta_i - \sum_j \beta_j q_j + \sum_k \gamma_k r_k$. First, form the substitutions needed to introduce \mathbf{yz} to count the number of variables Y_i and Z_i in any given term, weighted according to their rate coefficient in the homological equation.

```

449 y4y:=for i:=1:ny collect yy(i)=yy(i)*yz^rate(i)$
450 z4z:=for i:=1:nz collect zz(i)=zz(i)/yz^ratf(i)$
451 y4y:=append(y4y,z4z)$

```

9 Iterative updates

We iterate to a solution of the governing SDEs to residuals of some order of error. For the moment, iterate for a maximum of nineteen iterations and to the pre-specified errors.

```

452 for it:=1:maxiter_ do begin
453   ok:=1;

```

9.1 Fast stable modes

Compute the residual of each of the y_i SDEs, updating **ok** to track whether all SDEs are satisfied.

```

454   for i:=1:ny do begin
455       res:=-df(y(i),t)+part(yrhs,i);
456       ok:=if res=0 then ok else 0;

```

Trace print the length of the residuals to check how the iteration is progressing.

```

457       write lengthresy:=length(res);

```

Within the loop: first insert the weighted count of Y and Z variables; then split the residual into two parts of possibly resonant, **res0** and the rest, **res1**; then allocate to the evolution or the transform.

```

458     res:=sub(y4y,res);
459     res0:=reso(res,yz,+rate(i));
460     res1:=res-res0*yz^rate(i);
461     gg(i):=gg(i)+znnon(res0,tt);
462     y(i):=y(i) +zint(res0,tt) -zres(res1,ttyz,rate(i));
463   end;

```

9.2 Fast unstable modes

Compute the residual of each of the z_i SDEs, updating **ok** to track whether all SDEs are satisfied.

```

464   for i:=1:nz do begin
465       res:=-df(z(i),t)+part(zrhs,i);
466       ok:=if res=0 then ok else 0;

```

Trace print the length of the residuals to check how the iteration is progressing.

```

467       write lengthresz:=length(res);

```

Update the Z_i evolution **hh(i)** and the z_i transform. Within the loop: first insert the weighted count of Y and Z variables; then split the residual into two parts of possibly resonant, **res0**, and the rest, **res1**; then allocate to the evolution or the transform.

```

468     res:=sub(y4y,res);
469     res0:=reso(res,yz,-ratf(i));

```

```

470     res1:=res-res0/yz^ratf(i);
471     hh(i):=hh(i)+znon(res0,tt);
472     z(i):=z(i) +zint(res0,tt) -zres(res1,ttyz,-ratf(i));
473 end;

```

9.3 Slow modes

Compute the residual of each of the x SDEs, updating `ok` to track whether all SDEs are satisfied.

```

474 for i:=1:m do begin
475     res:=-df(x(i),t) +part(xrhs,i);
476     ok:=if res=0 then ok else 0;

```

Trace print the length of this residual.

```

477     write lengthresx:=length(res);

```

Update the X_i evolution `ff(i)` and the x_i transform. Use the same process as for the fast variables; the difference is that here the mode rate is zero.

```

478     res:=sub(y4y,res);
479     res0:=reso(res,yz,0);
480     res1:=res-res0;
481     ff(i):=ff(i)+znon(res0,tt);
482     x(i):=x(i) +zint(res0,tt) -zres(res1,ttyz,0);
483 end;

484 showtime;
485 if ok then write "Number of iterations ",
486     it:=1000000+it;
487 end;

```

10 Post-processing

Terminate if the iteration has not converged.

```

488 if ok=0 then begin
489     write "*****Error *****
490     Failed to converge in maximum allowed iterations";
491     if thecase=webpage then <<
492         shut "sdeo.txt"; quit >>;
493 end;

```

If converged, then print results.

```

494 write "***** Success *****";

```

10.1 Plain text output

Print the resultant coordinate transform: but only print to one lower power in `small` and `sigma` in order to keep output relatively small.

```

495 write "The stochastic/non-autonomous coordinate transform";
496 for i:=1:nz do begin z(i):=sigma*small*z(i);
497     write z(i):=z(i)/small/sigma; end;
498 for i:=1:ny do begin y(i):=sigma*small*y(i);
499     write y(i):=y(i)/small/sigma; end;
500 for i:=1:m do begin x(i):=sigma*small*x(i);
501     write x(i):=x(i)/small/sigma; end;

```

Lastly print the normal form SDEs: first the fast, second the slow.

```

502 write "The normal form S/ODEs";
503 for i:=1:nz do write "dzz(",i,")/dt = ",hh(i);
504 for i:=1:ny do write "dyy(",i,")/dt = ",gg(i);
505 for i:=1:m do write "dxx(",i,")/dt = ",ff(i);

```

Close the output file and no longer quit but move on to L^AT_EX output.

```

506 if thecase=webpage then shut "sdeo.txt";

```

10.2 L^AT_EX output

As before, we have to write expressions to file for later reading so they get printed without extraneous dross in the L^AT_EX source. First open up the temporary file `sdeo.red` again.

```
507 out "sdeo.red";
```

Write the stochastic coordinate transform to file, with a heading, and with an anti-math environment to cancel the auto-math of `rlfi`. For some reason we have to keep these writes short as otherwise it generates a spurious fatal blank line in the L^AT_EX.

```
508 write "write ""\end{math}"
509 \paragraph{Time dependent coordinate transform}
510 \begin{math}"";";
511 for i:=1:nz do write "z(",i,"):=z(",i,");";
512 for i:=1:ny do write "y(",i,"):=y(",i,");";
513 for i:=1:m do write "x(",i,"):=x(",i,");";
```

Write the resultant stochastic normal form to file, with a heading, and with an anti-math environment to cancel the auto-math of `rlfi`.

```
514 write "write ""\end{math}"
515 \paragraph{Result normal form DEs}
516 \begin{math}"";";
517 for i:=1:nz do write "hh(",i,"):=hh(",i,");";
518 for i:=1:ny do write "gg(",i,"):=gg(",i,");";
519 for i:=1:m do write "ff(",i,"):=ff(",i,");";
520 write "end;";
```

Shut the temporary output file.

```
521 shut "sdeo.red";
```

Get expressions from file and write the main L^AT_EX file. But first redefine how these names get printed, namely as the normal form time derivatives.

```
522 defid hh,name="\dot Z";
```



```

523 defid gg,name="\dot Y";
524 defid ff,name="\dot X";

```

Finally write to the main L^AT_EX file so switch on latex after starting to write to the file. Then write expressions in **sdeo.red** to **sdeo.tex** as nice L^AT_EX. Switch off latex, to get the end of the document, and finish writing.

```

525 out "sdeo.tex"$
526 on latex$
527 in "sdeo.red"$
528 off latex$
529 shut "sdeo.tex"$

```

Everything done, so say so and quit.

```

530 write "***** Finished *****";
531 if thecase=webpage then quit;
532 end;

```

References

- Chao, X. & Roberts, A. J. (1996), ‘On the low-dimensional modelling of Stratonovich stochastic differential equations’, *Physica A* **225**, 62–80. doi:[10.1016/0378-4371\(95\)00387-8](https://doi.org/10.1016/0378-4371(95)00387-8).
- Cox, S. M. & Roberts, A. J. (1991), ‘Centre manifolds of forced dynamical systems’, *J. Austral. Math. Soc. B* **32**, 401–436.
- Majda, A., Timofeyev, I. & Vanden-Eijnden, E. (2002), ‘A priori tests of a stochastic mode reduction strategy’, *Physica D* **170**, 206–252.
- Monahan, A. H. & Culina, J. (2011), ‘Stochastic averaging of idealized climate models’, *Journal of Climate* **24**(12), 3068–3088.
- Pavliotis, G. A. & Stuart, A. M. (2008), *Multiscale methods: averaging and homogenization*, Vol. 53 of *Texts in Applied Mathematics*, Springer.

- Potzsche, C. & Rasmussen, M. (2006), ‘Taylor approximation of integral manifolds’, *Journal of Dynamics and Differential Equations* **18**, 427–460. <http://dx.doi.org/10.1007/s10884-006-9011-8>
- Roberts, A. J. (2008), ‘Normal form transforms separate slow and fast modes in stochastic dynamical systems’, *Physica A* **387**, 12–38. doi:[10.1016/j.physa.2007.08.023](https://doi.org/10.1016/j.physa.2007.08.023).
- Roberts, A. J. (2009), Normal form of stochastic or deterministic multiscale differential equations, Technical report, <http://www.maths.adelaide.edu.au/anthony.roberts/sdenf.php>. Last revised Jun 2013.
- Roberts, A. J. (2013), Macroscale, slowly varying, models emerge from the microscale dynamics in long thin domains, Technical report, [<http://arxiv.org/abs/1310.1541>].
- Sun, X., Kan, X. & Duan, J. (2011), Approximation of invariant foliations for stochastic dynamical systems, Technical report, Illinois Institute of Technology.