Symmetry-breaking perturbations on the global attractor of the Kuramoto–Sivashinsky equation

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Mathematics

by

Beau James Smith

Eric John Olson/Thesis Advisor

May 2017
We recommend that the thesis prepared under our supervision by

**BEAU JAMES SMITH**

Entitled

Symmetry-breaking Perturbations on the Global Attractor of the Kuramoto-Sivashinsky Equation

be accepted in partial fulfillment of the requirements for the degree of

**MASTER OF SCIENCE**

Eric J. Olson, Ph. D., Advisor

Aleksey S. Telyakovskiy, Ph. D., Committee Member

Paul J. Hurtado, Ph. D., Committee Member

David M. Leitner, Ph. D., Graduate School Representative

David W. Zeh, Ph. D., Dean, Graduate School

May, 2017
Abstract

Mathematics

Master of Science

by Beau James Smith

We study symmetry-breaking of solutions on the global attractor of the Kuramoto–Sivashinsky equation. In our theory we prove that trajectories which result from small perturbations of a point on the global attractor stay close to the global attractor. In our numerics we exhibit a choice of parameters for the Kuramoto–Sivashinsky equation such that every $2\pi$-periodic initial condition (which isn’t zero or periodic on some smaller domain) converges to a traveling wave solution and such that every $4\pi$-periodic initial condition converges to a distinctly different fixed point. Our main result is to compute a non-recurrent trajectory on the attractor, connecting the traveling wave to the fixed point, given as the limit of smaller and smaller symmetry-breaking perturbations.
To my mom, Cindy Fox, for her love and encouragement.
I wish to thank Drs. Aleksey Telyakovskiy, Paul Hurtado and David Leitner for serving on my defense committee. I appreciate the time they took to read my work and to ask me questions that both improved the project and gave me direction for future study.

Also, I wish to thank Dr. Judith Fredrickson at Truckee Meadows Community College, under whom I learned C++ programming in 2010 and 2011. Her computer programming classes prepared me for the coding that this thesis (and several other math classes) needed. Her accessible teaching style and her skillfully crafted assignments prepared me to write, modify and understand the code that this work demanded.

Finally, I wish to thank my thesis advisor, Dr. Eric Olson. I appreciate his excellent explanations of the concepts, his helping me become a better programmer and mathematician, and his commitment to my academic success. Though it sounds cliché, it must be said: This project, not to mention my success in the graduate program, would not have been possible without him.
# Contents

Abstract i

Acknowledgements iii

List of Figures vi

1 Introduction 1

2 Theoretical Results 6

3 Numeric Scheme 14
   3.1 Rescaling 18

4 Computational Results 20
   4.1 A Fixed Point and a Traveling Wave 20
   4.2 Breaking the Symmetry 25

5 Conclusion 30
   5.0 What We Have Shown 30

6 Future Work 34

A The Global Attractor 37

B Source Code 41
   B.1 makeic.m 41
   B.2 convstudy.m 42
   B.3 makeconvstudy.m 47
   B.4 norm1.m 48
   B.5 norm2.m 49
   B.6 rk4graph.m 51
   B.7 makeimg.m 53
   B.8 period.m 54
   B.9 periodcall.m 56
List of Figures

1.1 Non-recurrent trajectory on the global attractor connecting a $2\pi$-periodic traveling wave to a $4\pi$-periodic fixed point. ........................................... 4

3.1 A convergence study of the Kassam–Trefethen fourth-order method in comparison to the Euler and Cox–Matthews fourth-order methods when $\mu = 0.1$, $\nu = 0.027$, $T = 1.0$ and $h = 1/N$. ................................. 17

4.1 A $4\pi$-periodic fixed point created when $\mu = 0.1$ and $\nu = 0.027$. ........... 20

4.2 Evolution of $|u(t)|$ starting at 10 different randomly chosen $4\pi$-periodic initial conditions. Each trajectory converges to a fixed point with norm approximately equal to 1.1804. ..................................................... 22

4.3 A $2\pi$-periodic traveling wave created when $\mu = 0.1$ and $\nu = 0.027$. ........ 23

4.4 Evolution of $|u(t)|$ starting at 10 different randomly chosen $2\pi$-periodic initial conditions. Each trajectory converges to a traveling wave with norm approximately equal to 0.7794 and velocity $\pm 0.0724$. ................................. 23

4.5 The evolution of the $|u(t)|$ with initial condition given by the $2\pi$-periodic traveling wave plus a $4\pi$-periodic perturbation that is not $2\pi$-periodic. ... 25

4.6 Graph of $T_\delta$ versus $\delta$ where $T_\delta$ is the time needed for the $2\pi$-symmetry to be broken by a $\delta$ sized perturbation. ..................................................... 26

4.7 Different values of $\delta$ lead to similar graphs translated in $x$. The graph on the left represents the points for $\delta = 10^{-2}$ and the right for $\delta = 10^{-12}$. 28

4.8 Evolution of the periodicity measure $p(u)$ over time. Trajectories for different values of $\delta$ have been vertically offset for clarity. Smaller values of $\delta$ shift the graph to the right with $u(t)$ tracing a similar trajectory through phase space. ..................................................... 29
Chapter 1

Introduction

The Kuramoto-Sivashinsky equation was derived in 1974 by George Homsy [6] and Alexander Nepomnyashchii [11], modeling equations for and studying liquid film flowing down an inclined plane; in 1976 by Yoshiki Kuramoto and Toshio Tsuzuki [8], studying persistent wave propagation through reaction-diffusion media; and in 1977 by Gregory Sivashinsky [15], studying instability in laminar flames; A Painlevé test and the presence of chaotic solutions indicate that no explicit general analytic solutions exist for this equation [3]; however, in this paper we will approximate solutions numerically using a method derived by Cox and Matthews [2] and refined by Kassam and Trefethen [7].

The Kuramoto-Sivashinsky equation is given by

$$u_t + uu_x + \mu u_{xx} + \nu u_{xxxx} = 0 \quad (1.1)$$
with initial condition \( u(0, x) = u_0(x) \) for \( x \in \mathbb{R} \). Here \( u \) represents, for example, the flame-front velocity, \( \mu \) is a dimensional constant that represents the heat released by the combustion reaction, and \( \nu \) represents the heat required to preheat the incoming reactants (see the discussion in the appendix to Chapter 11 in Griffiths and Schiesser [5]). The well-posed nature of (1.1) has been known since Tadmor affirmed it in 1986 [16]. From this point on, this paper will use the abbreviation KSE to refer to (1.1).

The KSE’s solutions are uniformly bounded in time with respect to the \( L^2 \) norm, which we shall denote as

\[
|w|^2 = \int_0^L w^2 \, dx. \tag{1.2}
\]

If \( u_0 \) is periodic such that \( u_0(x + L) = u_0(x) \) then \( u(t, x + L) = u(t, x) \) for all time \( t > 0 \). We therefore impose \( L \)-periodic boundary conditions on (1.1) and consider the phase space of all \( L \)-periodic solutions. Further note that if \( u_0 \) has zero average, then \( \int_0^L u(t, x) \, dx = 0 \) for all time \( t > 0 \). In the remainder of this work we consider only solutions to (1.1) with zero spatial average.

It is was first shown by Nicolaenko, Scheurer and Temam [12] (see also the discussion in Appendix A of this thesis) that the KSE with \( L \)-periodic boundary conditions has a unique global attractor, which we call \( \mathcal{A}_L \). The KSE with \( 2L \)-periodic boundary conditions also has a unique global attractor, which we call \( \mathcal{A}_{2L} \).

Consider the KSE with \( 2L \)-periodic boundary conditions. Since an \( L \)-periodic solution to the KSE is also \( 2L \)-periodic, then

\[ \mathcal{A}_L \subseteq \mathcal{A}_{2L}. \]
It could happen that $A_L = A_{2L}$, for example, when the attractor in both cases reduces to a single point at the origin. Our numerics show that there are choices of parameters in which $A_L \neq A_{2L}$. In particular, we exhibit parameters for which there exists a fixed point that is $2L$-periodic but not $L$-periodic.

Given a $2L$-periodic function $u$, define the $L$-periodicity measure of $u$ to be

$$p(u) = \left( \int_0^L |u(x) - u(x + L)|^2 dx \right)^{1/2}.$$  \hspace{1cm} (1.3)

Note that $u \in A_L$ implies $p(u) = 0$; however, $p(u) > 0$ for any solution which is $2L$-periodic but not $L$-periodic. This thesis focuses on breaking the symmetry of an $L$-periodic function on the global attractor by means of an $2L$-periodic perturbation (that is not $L$-periodic) small enough that the solution essentially remains on the global attractor. Theorem 2.2, proved in Chapter 2, shows that if the perturbation is small, then the resulting trajectory remains close to the attractor for all future times. For the choice of parameters $L = 2\pi$, $\mu = 0.1$ and $\nu = 0.027$, the numerics in Chapter 4 show that vanishingly small perturbations break the symmetry, which never returns as $t \to \infty$.

Once the symmetry is broken, the solution eventually converges to a fixed point with $p(u) \approx 1.6193$. Moreover, the trajectories in phase space taken by the symmetry-breaking perturbations converge to a unique set of points as the size of the perturbation vanishes.

Figure 1.1 illustrates points on the global attractor connecting a $2\pi$-periodic traveling wave solution to a $4\pi$-periodic fixed point for the choice of parameters $\mu = 0.1$ and $\nu = 0.027$. Each horizontal raster of the image corresponds to a point in the phase space.
Figure 1.1: Non-recurrent trajectory on the global attractor connecting a $2\pi$-periodic traveling wave to a $4\pi$-periodic fixed point.

Our numerics obtain the same picture, though perhaps shifted in time, no matter which perturbation we choose. Note that the traveling wave solution appearing at the bottom of the figure for $t \leq 250$ has 4 relative extrema. On the other hand, the fixed point appearing for $t \geq 650$ has 6 relative extrema which are unrelated to the traveling wave.

Generally, to find points on the global attractor of a partial differential equation, one is limited to evolving an arbitrary initial condition forward in time until the solution is close to the attractor. We emphasize, for the choice of parameters considered here, that the only points on the global attractor of the Kuramoto-Sivashinsky equation with
4π-periodic boundary conditions that could be found by evolving an arbitrary initial condition forward in time are the fixed point (and its various spatial translations) and the zero solution. Our computations show that the attractor is, in fact, much more complicated and includes a 2π-periodic traveling wave along with the non-recurrent symmetry-breaking trajectory depicted in Figure 1.1 that connects the traveling wave to a fixed point. Although there are set-based algorithms for approximating the global attractor for low-dimensional ordinary differential equations (see, for example, [4]), such techniques are impractical for higher dimensional systems and, in particular, for partial differential equations.
Chapter 2

Theoretical Results

The notion of an attractor was first proposed by Liapunov [10] in 1892. In 1955, Coddington and Levinson [1] defined the “attractor” as a single invariant point. The global attractor—the set that attracts all bounded sets in the norm-induced topology of the energy space—was first constructed by Olga Ladyzhenskaya [9] in 1987. We know that the Kuramoto–Sivashinky equation is well-posed and defines a solution operator $S(t)$ such that

$$u(t) = S(t)u_0 \quad \text{is the solution where} \quad u(0, x) = u_0(x).$$

Moreover, the KSE has a global attractor (see, for example, [12, 17]). For completeness of this thesis we outline the proof in Appendix A following the sequence of exercises suggested in Robinson [13]. That said, following Robinson, we define the global attractor in this manner:
Definition 2.1. The global attractor $A$ is the maximal compact invariant set

$$S(t)A = A \quad \text{for all} \quad t \geq 0$$

and the minimal set that attracts all bounded sets

$$\text{dist}(S(t)X, A) \to 0 \quad \text{as} \quad t \to \infty$$

for any bounded set $X \subseteq H$.

Throughout this paper, we will discuss the global attractor of the KSE with respect to the $L^2$ norm. Thus, for two sets $A$ and $B$ of $L$-periodic functions in the phase space,

$$\text{dist}(A, B) = \sup_{u \in A} \inf_{v \in B} |u - v|,$$

where

$$|u - v| = \left( \int_0^L (u(x) - v(x))^2 \, dx \right)^{1/2}.$$

With that, we introduce our main theoretical result.

Theorem 2.2. Let $A = A_L$ be the global attractor of the KSE (1.1) with $L$-periodic boundary conditions. Given $\epsilon > 0$, there is $\delta > 0$ such that, for any point $v_0 \in L^2([0, L]; \mathbb{R})$ with zero average,

$$\text{dist}\left(\{v_0\}, A\right) < \delta \quad \text{implies} \quad \text{dist}(\{S(t)v_0\}, A) < \epsilon$$
for all $t \geq 0$.

Before we prove this, we need an estimate on the continuity by which solutions depend on their initial data. Our result is a modification of the standard result that compares the evolution of two solutions $u$ and $v$ with different initial conditions $u_0$ and $v_0$. This modification assumes that $u_0$ (and consequently $u$) lies on the global attractor; we use this fact to obtain a slightly stronger result which can be used to show that $v$ is close to the global attractor.

**Lemma 2.3.** Let $u(t)$ and $v(t)$ be solutions to the KSE. If $u(t) \in \mathcal{A}$, then there is a constant $\beta > 0$ depending only on $L$, $\mu$ and $\nu$ such that

$$|u(t) - v(t)|^2 \leq |u_0 - v_0|^2 e^{\beta t} \text{ for all } t > 0.$$

**Proof.** Note that we integrate with respect to $x$ except where otherwise noted. Let $w = u - v$ where $u$ and $v$ are both solutions to the KSE. Then

$$w_t = -\nu u_{xxxx} - \mu u_{xx} - uu_x + \nu v_{xxxx} + \mu v_{xx} + vv_x + (uw_x - uv_x)$$

$$= -\nu (u - v)_{xxxx} - \mu (u - v)_{xx} - u(u_x - v_x) - v_x(u - v) \quad (2.1)$$

$$= -\nu w_{xxxx} - \mu w_{xx} - uw_x - vw_x.$$

Therefore,

$$w_t w = \frac{1}{2} \frac{\partial}{\partial t} w^2 = -\nu w_{xxxx}w - \mu w_{xx}w - uw_xw - vw_xw^2. \quad (2.2)$$
Integrating both sides with respect to $x$ from 0 to $L$, we get

\[
\frac{1}{2} \int_0^L \frac{\partial}{\partial t} w^2 \, dt = \frac{1}{2} \int_0^L w^2 \, dt \\
= - \int_0^L (\nu w_{xxxx} + \mu w_{xx} + uw_x + v_x w^2).
\]

(2.3)

Integrating by parts yields

\[
\int_0^L w_{xx}w = - \int_0^L w_x^2 \quad \text{and} \quad \int_0^L w_{xxxx}w = \int_0^L w_{xx}^2
\]

(2.4)

since the boundary terms

\[
w_x w \bigg|_0^L = 0, \quad w_{xxx} w \bigg|_0^L = 0 \quad \text{and} \quad w_{xx} w_x \bigg|_0^L = 0
\]

(2.5)

by $w$’s periodicity. Thus,

\[
\frac{1}{2} \int_0^L \frac{d}{dt} w^2 = -\nu \int_0^L w_{xx}^2 + \mu \int_0^L w_x^2 - \int_0^L uw_x w - \int_0^L v_x w^2.
\]

(2.6)

Now we consider the bound on $w$. Using (2.6) with (1.2), we get

\[
\frac{1}{2} \int_0^L |w|^2 + \nu |w_{xx}|^2 = \mu |w_x|^2 - \int_0^L uw_x w - \int_0^L v_x w^2.
\]

(2.7)

Observe that

\[
\int_0^L u_x w^2 = u w^2 \bigg|_0^L - 2 \int_0^L uw_x w = -2 \int_0^L uw_x w,
\]

(2.8)
since $uw^2\bigg|_0^L = 0$ by periodicity. Therefore

$$\frac{1}{2} \frac{d}{dt} |w|^2 + \nu |w_{xx}|^2 = \mu |w_x|^2 + \int_0^L uw_x w + \int_0^L w_x w^2. \tag{2.9}$$

Since

$$\int_0^L w_x w^2 = \frac{1}{3} w^3 \bigg|_0^L = 0,$$

then

$$\frac{1}{2} \frac{d}{dt} |w|^2 + \nu |w_{xx}|^2 = \mu |w_x|^2 + \int_0^L uw_x w. \tag{2.10}$$

Next, estimate $|u(t, x)|$ pointwise. Since $u$ has zero average by assumption, then for any time $t$ there is a point $x_*$ such that $u(t, x_*) = 0$. Consequently,

$$u(t, x) = u(t, x) - u(t, x_*) = \int_{x_*}^x u_x(t, s) \, ds$$

Therefore, the Cauchy-Schwarz inequality followed by (A.2) implies, since $u$ is on the global attractor, that

$$|u(t, x)| \leq \sqrt{\int_0^L 1 \sqrt{\int_0^L \big|u_x(t, s)\big|^2 \, ds}} = \sqrt{L} |u_x| \leq R_1 \sqrt{L}.$$

We have replaced the interval of integration by $[0, L]$, as $L$ is bigger than $|x_* - x|$. We now use this estimate (along with Young’s inequality, $2ab \leq a^2 + b^2$) to provide a bound
on the right-hand side of (2.10):

\[ \int_0^L uw_x w \leq R_1 \sqrt{L} |w_x||w| \leq \mu |w_x|^2 + \frac{R_1^2}{4 \mu} |w|^2. \]

Thus, equation (2.10) becomes

\[ \frac{1}{2} \frac{d}{dt} |w|^2 + \nu |w_{xx}|^2 \leq 2 \mu |w_x|^2 + \frac{R_1^2 L}{4 \mu} |w|^2. \]  \( (2.11) \)

Since

\[ |w_x|^2 = \int_0^L w_x^2 = - \int_0^L w_{xx} w \leq \sqrt{\int_0^L w_{xx}^2} \sqrt{\int_0^L w^2} = |w_{xx}| |w|, \]

then

\[ 2 \mu |w_x|^2 \leq 2 \mu |w_{xx}| |w| \leq \nu |w_{xx}|^2 + \frac{\mu^2}{\nu} |w|^2. \]

Plugging this estimate into (2.11) yields

\[ \frac{d}{dt} |w|^2 \leq \beta |w|^2 \quad \text{where} \quad \beta = \frac{2 \mu^2}{\nu} + \frac{R_1^2 L}{2 \mu}. \]

Multiply by the integrating factor \( e^{-\beta t} \) to obtain

\[ \frac{d}{dt} \left( |w|^2 e^{-\beta t} \right) \leq 0. \]

One last integration in \( t \) yields \( |w|^2 e^{-\beta t} - |w_0|^2 \leq 0 \), and so \( |w|^2 \leq |w_0|^2 e^{\beta t} \). \( \square \)

While it is well known that KSE is well-posed (i.e., it has unique solutions which depend
continuously on the initial data), the upshot of Lemma 2.3 is that the continuous dependence on initial data is controlled by an explicit constant \( \beta \) which depends only on \( L, \mu \) and \( \nu \). The existence of this constant depends on the fact that one of the solutions lies on the global attractor. We can now prove that if a second solution starts sufficiently close to the attractor, then it will remain close to the attractor for all future times.

**Proof of Theorem 2.2.** Let \( r > 0 \) and define

\[
X = \bigcup_{u_0 \in \mathcal{A}} B_r(u_0) \quad \text{where} \quad B_r(u_0) = \{ u \in H : |u - u_0| < r \}
\]

is the ball of radius \( r \) centered at \( u_0 \) in \( H \). Since \( \mathcal{A} \) is bounded, then \( X \) is bounded. Moreover, \( X \) is open and \( \mathcal{A} \subseteq X \). By definition of the attractor, for all \( \epsilon > 0 \) there is a time \( T > 0 \) such that

\[
\text{dist}(S(t)X, \mathcal{A}) < \epsilon \quad \text{for all} \quad t > T. \tag{2.12}
\]

Choose \( \delta > 0 \) sufficiently small such that \( \delta < r \) and \( \delta^2e^{\beta T} < \epsilon^2 \). Note that since \( \delta < r \) then \( \text{dist}(\{v_0\}, \mathcal{A}) < \delta \) implies \( v_0 \in X \). Moreover, if \( \text{dist}(\{v_0\}, \mathcal{A}) < \delta \) then there is \( u_0 \in \mathcal{A} \) such that \( |u_0 - v_0| < \delta \). It follows that \( u(t) = S(t)u_0 \) and \( v(t) = S(t)v_0 \) satisfy

\[
|u(t) - v(t)|^2 \leq |u_0 - v_0|^2e^{\beta t} \leq \delta^2e^{\beta T} < \epsilon^2 \quad \text{for} \quad t \in [0, T].
\]

Consequently,

\[
\text{dist}(\{S(t)v_0\}, \mathcal{A}) < \epsilon \quad \text{for all} \quad t \in [0, T].
\]
On the other hand, since \( v_0 \in X \), we already know from (2.12) that

\[
\text{dist}(\{S(t)v_0\}, \mathcal{A}) < \epsilon \quad \text{for all} \quad t > T.
\]
Chapter 3

Numeric Scheme

Before turning to our computational results, we first describe the numeric schemes used for our calculations. We used three different numeric schemes, not only to approximate solutions or to study their evolution over time, but also to confirm the correctness of our results. One scheme was a first-order method used for preliminary calculations and to verify the correctness of a higher order method used for our final results. For our final results we used a refinement of the Cox–Matthews method (see also Zavalani [19] for particular details relating to the KSE) described by Kassam and Trefethen in [7]. This method is a regularization of the fourth-order method derived and proposed by Cox and Matthews [2] to avoid numerical instability. We remark that the computational work for this thesis also provides an independent verification that the original Cox–Matthews scheme is unsuitable for approximating solutions to the KSE.
All computations were performed using a Fourier series representation of the solutions. In particular, we use fast Fourier transforms to approximate

\[ u(t, x) \approx \sum_{k=-M/2+1}^{M/2} \hat{u}_k(t) e^{ikx} \]  

(3.1)

and the nonlinear term as

\[ u(t, x)u_x(t, x) \approx \sum_{k=-M/2+1}^{M/2} \hat{B}_k(u(t, \cdot)) e^{ikx}, \]

where \( \hat{B} \) is defined as the Fourier coefficients of the function \( x \mapsto u(x)u_x(x) \).

Our first-order method is a split time-stepping method where the linear terms are integrated exactly in time and the nonlinear term is integrated according to an explicit Euler time step. Specifically, we set \( t_n = t_0 + nh \) where \( h > 0 \) and compute

\[
\hat{u}_k^{n+1} = \begin{cases} 
(\hat{u}_k^n - h\hat{B}_k) \exp \left( h(\mu|k|^2 - \nu|k|^4) \right) & \text{for } k \in [-M/2, M/2] \\
0 & \text{otherwise.}
\end{cases}
\]

MATLAB code to implement this scheme appears in Appendix B.2. Note that the code forces the solution \( u(t, x) \) to be real valued by taking the real part of every inverse fast Fourier transform performed throughout the code. This turns out to be necessary: Without it, rounding error leads to a non-zero imaginary part that exponentially grows and eventually destroys the numeric approximation. Note that we also enforce the fact that our solution \( u(t, x) \) has zero average by setting \( \hat{u}_0 = 0 \) in line 93 at every step.

The fourth-order method employed in our computations is the Runge–Kutta method.
derived by Cox and Matthews [2] and refined by Kassam and Trefethen [7]. The Cox–Matthews fourth-order method is

\[\hat{u}_{n+1}^k = \hat{u}_k^e ch + \{F(\hat{u}_k^n, t_n)[-4 - ch + e^{ch}(4 - 3hc + h^2c^2)] \\
+ 2(F(a_n, t_n + h/2) + F(b_n, t_n + h/2))[2 + ch + e^{ch}(hc - 2)] \\
+ F(c_n, t_n + h)[-4 - 3ch - h^2c^2 + e^{ch}(4 - ch)]\}/h^2c^3.\]

where

\[a_n = \hat{u}_k^n e^{ch/2} + (e^{ch/2} - 1) F(\hat{u}_k^n, t_n)/c,\]

\[b_n = \hat{u}_k^n e^{ch/2} + (e^{ch/2} - 1) F(a_n, t_n + h/2)/c,\]

\[c_n = a_n e^{ch/2} + (e^{ch/2} - 1) (2F(b_n, t_n + h/2) - F(\hat{u}_k^n, t_n))/c.\]

In the case of the KSE, the constant \(c = \mu|k|^2 - \nu|k|^4\) depends on \(k\), and \(F(u, t) = -B_k(u)\). MATLAB code to implement the original Cox–Matthews’s method for comparison purposes appears in lines 112 through 169 in Appendix B.2. A modification of Cox and Matthews’s fourth-order method was described by Kassam and Trefethen [7]. They take the mean of the terms \([-4 - ch + e^{ch}(4 - 3hc + h^2c^2)], [2 + ch + e^{ch}(hc - 2)], [-4 - 3ch - h^2c^2 + e^{ch}(4 - ch)]\) and \((e^{ch/2} - 1)/ch\) to stabilize the scheme. The MATLAB code to implement this method, appearing in Kassam and Trefethen’s original paper, was adapted for our final computations. Our adaptation may be found in lines 13 through 53 in Appendix B.2 as well as throughout the other MATLAB subroutines presented in
Appendix B.

**Figure 3.1:** A convergence study of the Kassam–Trefethen fourth-order method in comparison to the Euler and Cox–Matthews fourth-order methods when $\mu = 0.1$, $\nu = 0.027$, $T = 1.0$ and $h = 1/N$.

Figure 3.1 (see also Table 3.1) compares the rate of convergence of the Euler, Cox–Matthews and Kassam–Trefethen methods when computing a traveling wave solution in a $2\pi$-periodic domain for the parameters $\mu = 0.1$ and $\nu = 0.027$. As noted already, our tests verified that the Cox–Matthews method suffers from numeric instability when $h$ is small. In particular, the numerical instability with this method is so severe that its theoretical fourth-order convergence is barely evident. The Kassam–Trefethen method, on the other hand, has the same accuracy for large values of $h$ as the Cox–Matthews method and maintains the fourth-order convergence to achieve error levels below $10^{-9}$. We therefore use the Kassam–Trefethen method for our main computational results.
Table 3.1: Numeric results from the convergence study of the Kassam–Trefethen, the Euler and Cox–Matthews fourth-order methods.

<table>
<thead>
<tr>
<th>N</th>
<th>Euler</th>
<th>Cox–Matthews</th>
<th>Kassam–Trefethen</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6.052032830805079e-03</td>
<td>1.767170255843081e-06</td>
<td>1.767394808627359e-06</td>
</tr>
<tr>
<td>8</td>
<td>3.356388752860855e-03</td>
<td>2.197711209184392e-07</td>
<td>2.192326684470450e-07</td>
</tr>
<tr>
<td>16</td>
<td>1.781864755458604e-03</td>
<td>6.946876075309452e-08</td>
<td>2.218337364779630e-08</td>
</tr>
<tr>
<td>32</td>
<td>9.209066149152879e-04</td>
<td>3.856716575863000e-08</td>
<td>1.87956127924961e-09</td>
</tr>
<tr>
<td>64</td>
<td>4.6860634905980e-04</td>
<td>3.359029583048308e-07</td>
<td>1.388467936676712e-10</td>
</tr>
<tr>
<td>128</td>
<td>2.364360800147974e-04</td>
<td>1.281574053173745e-05</td>
<td>9.39348225890561e-12</td>
</tr>
<tr>
<td>256</td>
<td>1.187637454180111e-04</td>
<td>1.571688948464732e-04</td>
<td>6.074290464922053e-13</td>
</tr>
<tr>
<td>512</td>
<td>5.951989902635945e-05</td>
<td>3.328809924717493e-04</td>
<td>4.498271303429166e-14</td>
</tr>
<tr>
<td>1024</td>
<td>2.979465782546606e-05</td>
<td>1.848106051796373e-02</td>
<td>9.361490795707066e-14</td>
</tr>
<tr>
<td>2048</td>
<td>1.490603128131137e-05</td>
<td>6.488464483592826e-02</td>
<td>3.546320442890997e-14</td>
</tr>
</tbody>
</table>

3.1 Rescaling

The numerical methods discussed in the previous section each approximate $2\pi$-periodic solutions where $u$ is given by (3.1). In order to compute solutions on a $4\pi$-periodic domain, we rescaled $\mu$, $\nu$ and $L$ using a factor $\lambda$ as follows.

Given an $L$-periodic solution $u(x + L, t) = u(x, t)$ for all $x \in \mathbb{R}$ we first rescale $x$ by the scaling factor $\lambda$. Taking $\xi = \lambda x$ allows us to view $u$ as an $L\lambda$-periodic function of $\xi$. Then $d\xi/dx = \lambda$ and

$$u_x = \frac{\partial u}{\partial x} = \frac{\partial u}{\partial \xi} \frac{d\xi}{dx} = \lambda u_\xi.$$

Similarly, $u_{xx} = \lambda^2 u_{\xi\xi}$ and $u_{xxxx} = \lambda^4 u_{\xi\xi\xi\xi}$. Then the KSE given by (1.1) becomes

$$u_t + \lambda uu_\xi + \mu \lambda^2 u_{\xi\xi} + \nu \lambda^4 u_{\xi\xi\xi\xi} = 0.$$
Next, we rescale $t$. Let $\tau = \eta t$. We get $u_t = \eta u_\tau$, yielding

$$\eta u_\tau + \lambda uu_\xi + \mu \lambda^2 u_{\xi\xi} + \nu \lambda^4 u_{\xi\xi\xi\xi} = 0.$$  \hspace{1cm} (3.2)

By setting $\eta = \lambda$, we can divide both sides by $\lambda$ to obtain

$$u_\tau + uu_\xi + \mu \lambda u_{\xi\xi} + \nu \lambda^3 u_{\xi\xi\xi\xi} = 0.$$

Thus, we transform the equation (1.1) with $L$-periodic boundary conditions using the parameters

$$\xi = \lambda x, \quad \tau = \lambda t, \quad \tilde{\mu} = \mu \lambda, \quad \tilde{\nu} = \nu \lambda^3$$

to obtain

$$u_\tau + uu_\xi + \tilde{\mu} u_{\xi\xi} + \tilde{\nu} u_{\xi\xi\xi\xi} = 0,$$

which has the $L\lambda$-periodic boundary conditions that we desire.
Chapter 4

Computational Results

4.1 A Fixed Point and a Traveling Wave

Figure 4.1: A $4\pi$-periodic fixed point created when $\mu = 0.1$ and $\nu = 0.027$.

For our first set of calculations we used the following parameters: $L = 4\pi$, $\mu = 0.1$ and $\nu = 0.027$. These computations were performed using the Kassam–Trefethen numerical
scheme with fast Fourier transforms of size $M = 1024$ and a time step of size $h = 1/100$. Note that in order to compute a $4\pi$-periodic solution using our numeric codes we rescaled the equation by setting $\tilde{\mu} = 0.05$ and $\tilde{\nu} = 0.003375$ as in (3.2). It was observed when starting with a number of different initial conditions that the solution converged to a fixed point consisting of 3 relative maxima and 3 relative minima. Figure 4.1 depicts this $4\pi$-periodic fixed point solution. Other initial conditions converged to the same curve translated in space.

Figure 4.2 illustrates the time evolution of $|u(t)|$ starting at 10 different randomly chosen $4\pi$-periodic initial conditions. These initial conditions were sampled from the probability distribution

$$u_0(x) = \sum_{k=-M/2+1}^{M/2} \delta Z_k e^{-\gamma|k|} e^{ik\xi} \quad \text{with} \quad \xi = x/2,$$

where $\delta \approx 287.2777$, $\gamma = 0.2$,

$$Z_k = \begin{cases} X_k \exp(2\pi iY_k) & \text{for } k > 0, \\ 0 & \text{for } k = 0, \\ X_{-k} \exp(-2\pi iY_{-k}) & \text{for } k < 0, \end{cases}$$

and $X_k$ and $Y_k$ are uniformly distributed independent random variables on the interval $[0, 1]$. Note that $Z_{-k} = \overline{Z_k}$ ensures the initial condition is real-valued and $Z_0 = 0$ ensures the mean is zero. Here $\delta$ and $\gamma$ were chosen so that

$$E[|u_0|] \approx 1 \quad \text{and} \quad e^{-\gamma N/2} \approx 10^{-23}.$$
Thus, the expected norm of the initial condition is similar in magnitude to the norm of the subsequent evolution of the solution and the amplitudes of the highest Fourier modes are essentially zero in comparison to the lowest ones. Note that the exponential decay of the Fourier modes given by the parameter $\gamma$ ensures theoretically that the initial condition is smooth in space as $M \to \infty$. Each of the trajectories whose norms were given in Figure 4.2 converge to a fixed point with $|u(t)| \approx 1.1804$ that is a translation of the curve depicted in Figure 4.1. While these computations were performed using the Kassam–Trefethen numerical scheme, the Euler and Cox–Matthews schemes result in what our simulations show to be translations of the same fixed point.

**Figure 4.2**: Evolution of $|u(t)|$ starting at 10 different randomly chosen $4\pi$-periodic initial conditions. Each trajectory converges to a fixed point with norm approximately equal to 1.1804.

Theoretically, if any of the initial conditions considered above are translated by some amount, then the resulting fixed point would be translated by the same amount. Therefore, we let $\mathcal{L}$ be the subset of phase space which contains the curve in Figure 4.1 and its translates, and claim that the global attractor $\mathcal{A}_{4\pi}$ contains $\mathcal{L}$. Because the zero solution
is also a fixed point, we know that $0 \in A_{4\pi}$. Thus, $\{0\} \cup \mathcal{L} \subset A_{4\pi}$. Now, since the global attractor is a connected set, this means there must be additional points in the attractor that connect 0 to $\mathcal{L}$. In fact, the attractor is much more complicated than the fixed points that have been exhibited so far. The main goal of this thesis is to use symmetry-breaking perturbations to compute some of the additional points in $A_{4\pi}$.

**Figure 4.3:** A $2\pi$-periodic traveling wave created when $\mu = 0.1$ and $\nu = 0.027$.

**Figure 4.4:** Evolution of $|u(t)|$ starting at 10 different randomly chosen $2\pi$-periodic initial conditions. Each trajectory converges to a traveling wave with norm approximately equal to 0.7794 and velocity $\pm 0.0724$. 
Since the fixed point in Figure 4.1 is not $2\pi$-periodic, we consider specially chosen initial conditions in the $4\pi$-periodic domain which are, in fact, $2\pi$-periodic. Since the KSE (1.1) preserves periodicity, no matter how far forward in time any $2\pi$-periodic initial condition is evolved, it will never converge to the fixed point in Figure 4.1. It was observed when starting with a number of randomly chosen $2\pi$-periodic initial conditions that the solution converged to a traveling wave consisting of 2 relative maxima and 2 relative minima. Sometimes the traveling wave moved from right to left and other times the solution settled into a mirror image which moved from left to right. The speed of the traveling wave was approximately 0.0724. Figure 4.3 depicts the version of the traveling wave solution that moved from right to left.

Let $\mathcal{M}$ be the subset of phase space which contains the function depicted in Figure 4.3, its translates, and the mirror images of its translates. We conclude that global attractor $\mathcal{A}_{2\pi}$ contains $\mathcal{M}$. Moreover, since $\mathcal{A}_{2\pi} \subseteq \mathcal{A}_{4\pi}$ then

$$\{0\} \cup \mathcal{L} \cup \mathcal{M} \subseteq \mathcal{A}_{4\pi}.$$

It is worth remarking that one could also consider a $\pi$-periodic initial condition. Although the $\pi$-periodicity is preserved as the initial condition is evolved forward in time, the resulting solution converges to zero as $t \to \infty$. Since the zero solution is $\pi$-periodic (as well as periodic with respect to any other period) there is no contradiction here. However, no new points on the $\mathcal{A}_{4\pi}$ periodic attractor are found in this case.

One more case remains to be considered: the case of the $4\pi/3$-periodic initial condition.
However, initial conditions which are $4\pi/3$-periodic converge to the same fixed point described in Figure 4.1, and no new points on the global attractor have been found.

The next section uses symmetry-breaking to compute additional points on the global attractor.

4.2 Breaking the Symmetry

Figure 4.5: The evolution of the $|u(t)|$ with initial condition given by the $2\pi$-periodic traveling wave plus a $4\pi$-periodic perturbation that is not $2\pi$-periodic.

Figure 4.5 illustrates the time evolution of $|u(t)|$ with the initial condition given by the $2\pi$-periodic traveling wave $u_0$ in Figure 4.3 perturbed by

$$u_\delta(x) = (2\pi)^{-1/2} \delta \cos(x/2),$$
where $\delta = 10^{-8}$. Thus, $u(0, x) = u_0(x) + u_\delta(x)$. Note that $u_\delta$ is $4\pi$-periodic but not $2\pi$-periodic, and that $|u_\delta| = \delta$. For $t < 270$, the norm $|u(t)|$ stays around 0.7794 until the perturbation becomes noticeable. Afterward, $|u(t)|$ fluctuates and hits its lowest level before rising and converging to approximately 1.1804 for $t > 700$.

Theorem 2.2 implies that if $\delta$ is small enough, then $u(t)$ will stay close to $A_{4\pi}$ for all $t > 0$. Even though $u(0, x)$ is within $10^{-8}$ of the $2\pi$-periodic traveling wave and $u(700, x)$ is similarly close to the $4\pi$-periodic fixed point, further evidence is needed before concluding that $u(t, x)$ is close to $A_{4\pi}$ for every $t \in [0, T]$. To verify that $\delta$ is small enough, we try different sizes of $\delta$.

**Figure 4.6:** Graph of $T_\delta$ versus $\delta$ where $T_\delta$ is the time needed for the $2\pi$-symmetry to be broken by a $\delta$ sized perturbation.

When $\delta$ is smaller it takes longer for the effects of the perturbation to make a noticeable difference on the symmetry of the resulting solution. To measure the time it takes to
break the symmetry we employ our $L$-periodicity measure (1.3) with $L = 2\pi$. Define

$$T_\delta = \sup \left\{ T : p(u(t)) \leq 0.1 \text{ for all } t \in [0, T] \right\}$$

where $u$ is the solution with initial condition $u(0, x) = u_0(x) + u_\delta(x)$ and 0.1 is the value of the periodicity measure when the perturbation of the solution in the $L$ domain becomes noticeable. Note that 0.01, 1.5 or any other value that uniquely determines the relative position of the oscillating pattern in Figure 4.8 for different values of $\delta$ would work equally well. Figure 4.6 plots the values of $T_\delta$ versus $\log_{10}(\delta)$ for $\delta = 10^{-n}$ for $n \in \{2, 4, \ldots, 16\}$. The points lie in a straight line given by the least squares fit

$$T_\delta \approx -43.043 \log_{10}(\delta) - 58.486.$$ 

The goodness of fit is striking: Changing $\delta$ changes the symmetry-breaking time by an easily predictable amount over a range of 14 orders of magnitude.

Even more striking is the time evolution of $p(u(t))$ versus $t$ for the different values of $\delta$. Figure 4.8 shows that $p(u(t))$ follows the same pattern of fluctuations for every value of $\delta$ tested. It appears that smaller values of $\delta$ merely delay when the symmetry is broken, but the pattern with which the symmetry is broken is essentially the same in all cases. This suggests that, given any perturbation of any size, the solution will travel a similar path in phase space. Representative trajectories appear in Figure 4.7. Note that each trajectory has been shifted in time to take into account the different values $T_\delta$. Moreover, after adjusting for $T_\delta$ and the translation in $x$, each graph is the same as given in Figure
Figure 4.7: Different values of δ lead to similar graphs translated in x. The graph on the left represents the points for δ = 10^{-2} and the right for δ = 10^{-12}.

1.1. Thus, the numerics suggest there exists $x_\delta$ such that

$$(x, t) \to S(t + (T_{10^{-8}} - T_\delta))(u_0 + u_\delta)(x - x_\delta)$$

is indistinguishable from Figure 1.1 for every $\delta > 0$. We conclude that these points (and their translates in x) are on the global attractor $A_{4\pi}$. 
Figure 4.8: Evolution of the periodicity measure $p(u)$ over time. Trajectories for different values of $\delta$ have been vertically offset for clarity. Smaller values of $\delta$ shift the graph to the right with $u(t)$ tracing a similar trajectory through phase space.
Chapter 5

Conclusion

5.0 What We Have Shown

The purpose of this thesis was to use symmetry-breaking perturbations on solutions of the KSE to find points on the global attractor. Our experiments used the fact that solutions on the global attractor remain on the attractor, and that very small perturbations—with \( \delta \) on the order of \( 10^{-19} \)—can break the symmetry. We also saw that the solutions in both \( L \) - and \( 2L \)-periodic domains exhibit various kinds of behavior, including traveling waves in a \( 2\pi \) domain chaos in a \( 4\pi \)-domain, and fixed points in a \( 4\pi \) domain. Numerically, we find a traveling-wave solution which is \( 2\pi \)-periodic and a fixed point which is \( 4\pi \)-periodic. Both these points are clearly contained in the global attractor \( A_{4\pi} \). Note that the traveling wave moved with a velocity of 0.0724 and therefore represents a solution which is periodic in time with period \( 2\pi/0.0724 \approx 86.784 \) such that \( u(t + 86.74, x) = u(t, x) \) for all \( t \) and \( x \).
Our theoretical results tell us that if an initial condition is $\delta$-close to the attractor, then it stays $\epsilon$-close. Our computations tell us solutions whose initial conditions are $\delta$ close to the $2\pi$-periodic traveling wave converge over time to the $4\pi$-periodic fixed-point. For definiteness, we take our initial conditions equal to the $2\pi$-periodic traveling wave perturbed by $\delta(2\pi)^{-1/2}\cos(x/2)$. For sufficiently small values of $\delta$, the points

$$V_\delta = \left\{ S(t)(u_\delta + u_0) \mid t \geq 0 \right\}$$

through phase space that these solutions pass through appear numerically identical when translated in $x$ by $x_\delta$. Other types of perturbations yielded similar results but have not been described here. We conclude that the trajectory given by Figure 1.1 represents points on the global attractor $A_{4\pi}$.

Theoretically, if all graphs are similar except for a shift along the $x$-axis, one could find a subsequence $\delta_n \to 0$ such that $x_{\delta_n}$ converges. This subsequence exists because of the compactness of $[0,2L]$. In this case, then $V_{\delta_n}$ would converge in the Hausdorff metric and the limit set would consist of points on the global attractor. This set is noteworthy because it is non-recurrent—i.e., for any $u_0 \in V_\delta$ there is $\epsilon > 0$ and $T > 0$ such that $|u_0 - S(t)u_0| > \epsilon$ for $t \geq T$. Since the traveling wave and the fixed point are demonstrably on the attractor, our theory coupled with a limit of smaller and smaller symmetry-breaking perturbations provides convincing evidence that the points given in Figure 1.1 do, in fact, lie on the global attractor.
Numerically, our results are approximations good to within the double-precision arithmetic used for our computations and the fourth-order Kassam–Trefethen scheme. As a result, numerical rounding implies that

\[(2\pi)^{-1/2}\delta \cos(x/2) + u_0(x) = u_0(x)\]  \hspace{1cm} (5.1)

whenever \(\delta\) is 15 orders of magnitude smaller than \(u_0(x)\). Therefore, there are numerical constraints when computing the limit \(\delta \to 0\). On the other hand, in the Fourier representation the same perturbation (up to a phase shift) may be written as

\[\frac{\delta}{\sqrt{8\pi}} + \hat{u}_1 \quad \text{and} \quad \frac{\delta}{\sqrt{8\pi}} + \hat{u}_{-1}\]

where \(\hat{u}_k\) are the Fourier modes of \(u_0\). Since \(u_0\) is \(2\pi\) periodic, it follows that \(\hat{u}_k = 0\) for all \(k\) odd. In particular, no rounding issues appear when the Fourier modes are perturbed by any representable non-zero \(\delta\). On the other hand, the non-linear term \(uu_x\) is always computed in the physical space by means of fast Fourier transforms. Therefore, even if it is possible to represent the perturbation of \(u_0\) in Fourier space accurately, the resulting non-linear terms used in our numerics are still subject to the rounding issue already mentioned in (5.1). Double-precision arithmetic calculations show that the \(\delta\) perturbation plays a role in the linear terms of the computation until \(\delta\) becomes as small as approximately \(1.023211357913 \times 10^{-19}\). In particular, the log-linear relationship between \(\delta\) and \(T_\delta\) depicted in Figure 4.6 is not likely to hold for value of delta less than \(10^{-19}\) unless higher precision floating-point arithmetic is used on the computer.
We end by noting that our simulations provide an independent verification of the statement by Kassam and Trefethen [7] that the fourth-order numeric scheme developed by Cox and Matthews [2] suffers numerical instability when used to compute solutions to the KSE. In particular, we have confirmed the accuracy of Kassam’s and Trefethen’s refinement of the Cox–Matthews method and used this refinement to perform our experiments with different initial conditions, types of perturbations, and values of $\mu$ and $\nu$. 
Chapter 6

Future Work

Thus far, we have obtained numerical solutions that converge to fixed points, solutions that converge to 0 as $t \to \infty$, and solutions that converge to traveling waves. Our most peculiar and interesting case, with $\mu = 0.1$ and $\nu = 0.027$, started as a traveling wave in an $L$-periodic region; then, when we doubled the period, it transformed into a chaotic solution which eventually settled into a fixed point with 6 extrema.

Our study focused on calculations performed with the parameters $\mu = 0.1$, $\nu = 0.027$ and $L = 2\pi$. During our work we discovered another choice of parameters, $\mu = 1$ and $\nu = 0.05$, which led to a fixed point with 8 extrema in an $2\pi$-periodic domain and another fixed point, but with 14 extrema, in a $4\pi$-periodic domain. It is worth confirming whether taking $\delta \to 0$ would give a similar set of points on the global attractor in this case. Alternatively, further experiments may yield a solution in which the number of extrema
doubles as the period doubles. It would be interesting to find solutions in which the number of extrema double or triple when the period is doubled or tripled, respectively.

We are interested in studying smaller values of $\delta$ using higher-precision floating-point arithmetic. In particular, could advanced rigorous numeric techniques be used to justify our claim that Figure 1.1 consists of points on the global attractor? Although we were able to reproduce Figure 1.1 using the split Euler method with very small time steps, we want to further study how the spatial resolution affects these results.

Also, all our simulations and experiments were conducted using the $L^2$-norm. Since all numerics are by necessity finite-dimensional, then all norms are equivalent. However, from a theoretical point of view, we are interested in proving versions of Theorem 2.2 where distances are measured in higher Sobolev spaces and when using $L^p$-norms with $p > 2$. The symmetry-breaking perturbation by $\cos(x/2)$ is smooth and therefore contained in every Sobolev and $L^p$ space. Does it follow that the resulting trajectory stays close to the global attractor with respect to these higher norms when $\delta$ is small enough. On the other hand, we have computational tried perturbations that are rough (for example, white noise in space) and found that Figure 1.1 is still reproduced as $\delta \to 0$.

Furthermore, we are interested in discovering the effects of using perturbations to break the symmetry of solutions of the KSE in larger domains—i.e., $3L$- and $4L$-periodic domains. Yet another extension of our research could take us into considering the Kuramoto–Sivashinsky equation in two or three dimensions, or other equations, such as the Navier–Stokes equations. Could this symmetry-breaking technique be used to find
points on the global attractor for other equations? What happens to solutions of such PDEs if we break the symmetry of those solutions?

Another area of interest is the real-world application of our findings. Do these results have any implications for the physical interpretations of the KSE with respect to, for example, the movement of flame fronts?
Appendix A

The Global Attractor

Before we build up to our main theoretical result (2.2), we need to discuss some select properties of the KSE’s global attractor. The main result we need is the bound on $|u_x|$ for solutions on the global attractor used in the proof of Lemma 2.3. This result follows from standard bounds used to show the existence of the global attractor. Namely, there exists $R$ and $M$ depending only on $\mu$, $\nu$ and $L$ such that

$$|u(t)| \leq R \quad \text{and} \quad \int_t^{t+1} |u_{xx}(s)|^2 ds \leq M \quad \text{for all} \quad t \geq t_0(|u_0|). \quad \text{(A.1)}$$

A proof of this result in the case of odd solutions to the KSE appears in Robinson [14], Temam [17], and references therein. These results extend to solutions which are not odd. We now prove
Theorem A.1. There exists $R_1$ depending only on $\mu$, $\nu$ and $L$ such that

$$|u_x| \leq R_1 \quad \text{for every} \quad u \in \mathcal{A}.$$ (A.2)

Proof. Take the inner product of

$$\frac{du}{dt} + uu_x + \mu u_{xx} + \nu u_{xxxx} = 0$$

with $-u_{xx}$ to obtain

$$\frac{1}{2} \frac{d}{dt} |u_x|^2 + \nu |u_{xxx}|^2 = \mu |u_{xx}|^2 + \int_0^L uu_x u_{xx}$$

Since $u$ has zero average there is $x_0 \in [0, L]$ such that $u(x_0) = 0$. Therefore, by the fundamental theorem of calculus

$$u(\xi) = u(x_0) + \int_{x_0}^\xi u_x = \int_{x_0}^\xi u_x.$$ Consequently,

$$|u(\xi)| = \left| \int_{x_0}^\xi u_x \right| \leq \int_0^L |u_x| \leq \sqrt{L} |u_x|$$

and therefore $|u|_{L^\infty} \leq \sqrt{L} |u_x|$. Using this bound we estimate

$$\int_0^L uu_x u_{xx} \leq |u|_{L^\infty} \int_0^L |u_x u_{xx}| \leq |u|_{L^\infty} |u_x| |u_{xx}| \leq \sqrt{L} |u_x|^2 |u_{xx}|.$$
Now
\[ \frac{1}{2} \frac{d}{dt} |u_x|^2 + \nu |u_{xxx}|^2 \leq \mu |u_{xx}|^2 + \sqrt{L} |u_{xx}||u_x|^2. \]

We neglect the term in \(|u_{xxx}|^2\) to obtain
\[ \frac{d}{dt} |u_x|^2 \leq \mu |u_{xx}|^2 + \sqrt{L} |u_{xx}||u_x|^2. \]

We may estimate as
\[ \frac{d}{dt} (\psi |u_x|^2) \leq \psi \mu |u_{xx}|^2 \quad \text{where} \quad \psi(t) = \exp \left( -\sqrt{L} \int_{t_*}^{t} |u_{xx}| \right) \quad (A.3) \]
and \(t_* \in [t-1, t]\) has been chosen so that \(|u_x(t_*)|^2 \leq R\sqrt{M}\). Such a \(t_*\) exists because by equation (A.1) the average value of \(|u_{xx}|^2\) satisfies
\[ \int_{t-1}^{t} |u_{xx}|^2 \leq M \]
so there is at least one point in the interval \([t-1, t]\) such that \(|u_{xx}|^2\) is less than its average value. Consequently, integrating by parts and applying the Cauchy–Schwartz inequality yields
\[ |u_x(t_*)|^2 = -\int_0^L u(t_*, x)u_{xx}(t_*, x) \, dx \leq |u(t_*)| |u_{xx}(t_*)| \leq R\sqrt{M}. \]

Now we integrate (A.3) over the interval \([t_*, t]\) and obtain
\[ \psi(t)|u_x(t)|^2 - \psi(t_*)|u_x(t_*)|^2 = \psi(t)|u_x(t)|^2 - |u_x(t_*)|^2 \leq \mu \int_{t_*}^{t} \psi(s)|u_{xx}(s)|^2 \, ds \]
since $\psi(t_*) = 1$. If we divide out $\psi(t)$, we get

$$|u_x(t)|^2 \leq |u_x(t_*)|^2 \frac{1}{\psi(t)} + \frac{1}{\psi(t)} \int_{t_*}^{t} \psi(s)|u_{xx}(s)|^2 ds.$$ 

Since $\psi$ is a decreasing function and $t_* + 1 > t$ then using equation (A.1)

$$\int_{t_*}^{t} |u_{xx}| \leq \int_{t_*}^{t_*+1} |u_{xx}| \leq \sqrt{\int_{t_*}^{t_*+1} |u_{xx}|^2} \leq \sqrt{M}$$

implies

$$\frac{1}{\psi(t)} \leq \frac{1}{\psi(t_* + 1)} \leq e^{\sqrt{LM}}.$$

Furthermore, we chose $t_*$ so that $|u_x(t_*)|^2 \leq R\sqrt{M}$. Therefore

$$|u_x(t)|^2 \leq |u_x(t_*)|^2 e^{\sqrt{LM}} + \mu e^{\sqrt{LM}} \int_{t_*}^{t_*+1} |u_{xx}(s)|^2 ds \leq R\sqrt{M} e^{\sqrt{LM}} + \mu Me^{\sqrt{LM}}.$$ 

It follows that

$$|u_x|^2 \leq R_1^2 \quad \text{where} \quad R_1^2 = (R\sqrt{M} + \mu M)e^{\sqrt{LM}} \quad \text{provided} \quad t \geq t_0(|u_0|).$$

In the case that $u_0 \in \mathcal{A}$ there is no need to take $t \geq t_0$ in inequality (A.2) because $u_0$ already reflects the long-time evolution of a solution to the KSE. We therefore have the bound

$$|u_x| \leq R_1 \quad \text{for every} \quad u \in \mathcal{A}.$$ 

Finally, we emphasize that $R_1$ depends only on $L$, $\mu$ and $\nu$. 

\[\square\]
Appendix B

Source Code

The following programs make use of Kassam and Trefethen’s fourth-order Runge-Kutta method, presented in [7] and modified for our research.

B.1 makeic.m

This program creates the traveling wave solution in the $2\pi$-length domain. The initial condition $u(x,0) = 0$ is perturbed by $(2\pi)^{-1/2}\delta \cos x/2$. The program then uses the Kassam–Trefethen integrator to generate the traveling wave. Finally, it saves the Fourier modes into a MATLAB data file and the physical solution into a separate MATLAB data file (which are loaded and used by our other scripts).

```matlab
1 % Spatial grid and initial condition:
2 lambda=1/2;
3 delta=1e-4;
4 N = 512;
5 x = 2*pi*(1:N)'/N;
6 % Generate random 3-by-1 vector.
7 u=delta*sin(x);
8 v = fft(u);
9 PMOD = 5;
```

% Method and integrator of Kassam and Trefethen's method (2005)
mu = .1; nu = .027;
h = 1/50; % time step
k = [0:N/2-1 0 -N/2+1:-1]'; % wave numbers
L = mu*k.^2 - nu*k.^4; % Fourier multipliers
E = exp(h*L); E2 = exp(h*L/2);
M = 16; % no. of points for complex means
r = exp(1i*pi*((1:M)-.5)/M); % roots of unity
LR = h*L(:,ones(M,1)) + r(ones(N,1),:);
Q = h*real(mean( (exp(LR/2)-1)./LR ,2));
f1 = h*real(mean( (-4-LR+exp(LR).*(4-3*LR+LR.^2))./LR.^3 ,2));
f2 = h*real(mean( (2+LR+exp(LR).*(-2+LR))./LR.^3 ,2));
f3 = h*real(mean( (-4-3*LR-LR.^2+exp(LR).*(4-LR))./LR.^3 ,2));
uu = u; tt = 0;
tmax = 2048; nmax = 1.5*round(tmax/h); nplt = floor((tmax/128)/h)/4;
g = -0.5i*k;
for n = 1:nmax
t = n*h;
Nv = g.*fft(real(ifft(v)).^2);
a = E2.*v + Q.*Nv;
Na = g.*fft(real(ifft(a)).^2);
b = E2.*v + Q.*Na;
Nb = g.*fft(real(ifft(b)).^2);
c = E2.*a + Q.*(2*Nb-Nv);
Nc = g.*fft(real(ifft(c)).^2);
v = E.*v + Nv.*f1 + 2*(Na+Nb).*f2 + Nc.*f3;
if mod(n,nplt)==0
    u = real(ifft(v));
end
end

% Save the Fourier modes (v)
% and spatial solution (u).
save('icv.mat','v');
save('icu.mat','u');

B.2 convstudy.m

This program generates the convergence study graph. It loads the physical solution as an initial condition to be used in each routine. The program then studies the convergence of
the Kassam–Trefethen method; after that it studies the convergence of the Euler method. Finally, it calls the script that plots the results.

```matlab
1 clear all
2
3 % K-T Method
4 vv = [] ; NN4 = [];
5 er= []; erEuler= []; erCM= [];
6 hr= []; hr1= [];
7 T=1.0;
8 nvals= [2:12];
9
10 load('icu.mat','u')
11 uRep= u;
12
13 for N4 = 2.^nvals
14
15 % Spatial grid and initial condition
16 N = 512;
17 x = (1:N)'/N;
18 u = uRep;
19 v = fft(u);
20
21 % Method and integrator of Kassam and Trefethen’s method (2005)
22 mu = 0.1; nu = 0.027;
23 h = T/N4; % time step
24 k = [0:N/2-1 0 -N/2+1:-1]'; % wave numbers
25 L = mu*k.'^2 - nu*k.'^4; % Fourier multipliers
26 E = exp(h*L); E2 = exp(h*L/2);
27 M = 16; % no. of points for complex means
28 r = exp(1i*pi*((1:M)-.5)/M); % roots of unity
29 LR = h*L(:,:,ones(M,1)) + r(ones(N,1),:);
30 Q = h*real(mean((exp(LR/2)-1)./LR,2));
31 f1 = h*real(mean(( -4-LR+exp(LR).*(-2+LR))./(LR.^3 ,2)));
32 f2 = h*real(mean(( 2+LR+exp(LR).*(-2+LR))./LR.^3 ,2));
33 f3 = h*real(mean(( -4-3*LR-LR.*2+exp(LR).*(-4-LR))./LR.^3 ,2));
34 g = -0.5i*k;
35 for n = 1:N4
36     t = n*h;
37     Nv = g.*fft(real(ifft(v)).'2);
38     a = E2.*v + Q.*Nv;
39     Na = g.*fft(real(ifft(a)).'2);
40     b = E2.*v + Q.*Na;
41     Nb = g.*fft(real(ifft(b)).'2);
```
c = E2.*a + Q.*(2*Nb-Nv);
Nc = g.*fft(real(ifft(c)).^2);
v = E.*v + Nv.*f1 + 2*(Na+Nb).*f2 + Nc.*f3;
end

u = real(ifft(v));
NN4 = [NN4,N4];
vv = [vv,u];
c0 = size(vv);
c0 = c0(2);
end

fileID=fopen('kt.dat','w');
fprintf(fileID,'%6s %16s
','N','En=|un-u2n|');
for n = 1:(co-1)
hr(n) = 1.0/NN4(n);
er(n) = norm(vv(:,n)-vv(:,n+1));
fprintf(fileID,'%6d %0.15e
',2^(n+1), er(n));
end
fclose(fileID);

% Euler Method
M=512;
NN1 = []; vv1 = [];
for N4=2."nvals
x=(0:M-1)/M;
u1=uRep';
avg=sum(u1)/length(u1);
u1=u1-avg; %Make it zero-average.
uld=uld;
M=length(u1);
B=floor(M/3)+1;
mid=floor(M/2)+1;
ufft=fft(u1);
ufft(1)=0.0;
ufft(mid-B:mid+B)=0;
h=T/N4;
kvec=[0:M/2-1 0 -M/2+1:-1];
k2vec=kvec.*kvec;
ikec=1i*[0:M/2-1 0 -M/2+1:-1];
efact=exp(h*(mu-nu*k2vec).*k2vec);
for n=1:N4
uxfft=ikvec.*ufft;
ux=real(ifft(uxfft));
b=real(u1.*ux);
bfft=fft(b);
bfft(mid-B:mid+B)=0;
ufft=(ufft-h*bfft).*efact;
ufft(1)=0.0;
u1=real(ifft(ufft));
old=u1;
end
NN1 = [NN1,N4];
v1 = [v1,old'];
c1 = size(v1);
c1 = c1(2);
end

fileID=fopen('euler.dat','w');
fprintf(fileID,'%16s
','En=|un-u2n|');
for n = 1:(c1-1)
    hr1(n) = 1.0/NN1(n);
erEuler(n) = norm(vv1(:,n)-vv1(:,n+1));
    fprintf(fileID,'%0.15e
', erEuler(n));
end
fclose(fileID);

% Cox-Matthews RK4 Method
M=512;
NN2 = [];
v2 = [];
mu=0.1; nu=0.027;

for N4=2.^nvals
    x=(0:M-1)/M;
u=uRep';
    avg=sum(u)/length(u);
u=u-avg; %Make it zero-average.
M=length(u);
B=floor(M/3)+1;
mid=floor(M/2)+1;
ufft=fft(u);
ufft(1)=0.0;
ufft(mid-B:mid+B)=0;
h=T/N4;
% Create indexing vector.
kvec=[0:M/2,-M/2+1:-1];
k2vec=kvec.*kvec;
ikivec=1i*kvec;
L=(mu-nu*k2vec).*k2vec;
L=L(2:M);
efact=exp(h*L);

for n=1:N4
    Fn=-ikivec.*fft(u.*u)/2;
    afft(2:M)=ufft(2:M).*efact+(efact-1)./L.*Fn(2:M);
    afft(1)=0;
    a=real(ifft(afft));
    Fa=-ikivec.*fft(a.*a)/2;

    bfft(2:M)=ufft(2:M).*efact+(efact-1).*Fa(2:M);
    bfft(1)=0;
    b=real(ifft(bfft));
    Fb=-ikivec.*fft(b.*b)/2;

    cfft(2:M)=afft(2:M).*efact+(2*Fb(2:M)-Fn(2:M));
    cfft(1)=0;
    c=real(ifft(cfft));
    Fc=-ikivec.*fft(c.*c)/2;

    ufft(2:M)=ufft(2:M).*efact+... % (4-3*h*L+3*h^2*L^2-2+4*h*L-3)./(h^2*L^3);
    ufft(1)=0;
    u=real(ifft(ufft));
end

NN2 = [NN2,N4];
vv2 = [vv2,u'];
co2 = size(vv2);
co2 = co2(2);

fileID=fopen('cm4.dat','w');
fprintf(fileID,'%16s
','En=|un-u2n|');
for n = 1:(co2-1)
    hr(n) = 1.0/NN2(n);
    erCM(n) = norm(vv2(:,n)-vv2(:,n+1));
    fprintf(fileID,'%0.15e
', erCM(n));
end
end
fclose(fileID);

makeconvstudy

B.3 makeconvstudy.m

This program generates the convergence study graph and data table from B.2.

% Normalizing factor for plotting.
s2pin=sqrt(2*pi/N);

% Plot Euler convergence.
loglog(hr1,s2pin*erEuler, '-^k')
hold on

% Plot Cox-Matthews convergence.
s2pin=sqrt(2*pi/M);
loglog(0.5.^nvals(1:length(nvals)-1),s2pin*erCM, '-ok')
hold on

% Plot K-T convergence.
loglog(hr,s2pin*er, '-sk')
hold on

% Plot h^1 and h^4.
loglog(hr1,hr1, '-.k');
hold on
loglog(hr, hr.^4, '-.k');
hold off

xlabel('h');
ylabel('|u_h(T)-u_{h/2}(T)|');
legend('Euler','C-M','K-T','h^1','h^4','Location','southeast')
print('convstudy','-dpng','-r600')

% Create data table.
fileID=fopen('convdata.txt','w');
fprintf(fileID, '%16s %16s %16s %16s
', 'N','Euler', 'Cox-Matthews','K-T');
fprintf(fileID, '%4d %0.15e %0.15e %0.15e
', NN1(1:10); s2pin*erEuler; s2pin*erCM; s2pin*er)
fclose(fileID);
This program graphs the evolution of the norm $|u|$ in the $2\pi$ domain. It creates a $2\pi$-periodic initial condition of the form

$$u(x, 0) = \delta \left( \sum_{n=1}^{3} P_n \sin (nx + R_n) \right),$$

where $\delta < 1$ is the damping of the trigonometric polynomial.

```matlab
% Spatial grid and initial condition:
clear data
data = [];
lambda=1/2;
N = 512;
g = 0.2;
delta=N*sqrt(1-exp(-2*g))*sqrt(3)/sqrt(4*pi)*2;

vp=delta*rand(N/2,1).*exp((-1:-1:-N/2)*g').*exp(1i*2*pi*rand(N/2,1));
vp(N/2+1)=0;
vp(N/2+2:N)=flip(conj(vp(2:N/2)));

v=vp;
v(1)=0.0; % Make it zero-average.
x=(2*pi*(1:N)'/N)/lambda;
u=real(ifft(v));

% Method and integrator of Kassam and Trefethen's method (2005)
mu = .1; nu = .027;
h = 1/100; % time step
k = [0:N/2-1 0 -N/2+1:-1]'; % wave numbers
L = mu*k.'*2 - nu*k.'*4; % Fourier multipliers
E = exp(h*L); E2 = exp(h*L/2);
M = 16; % no. of points for complex means
r = exp(1i*pi*((1:M)-.5)/M); % roots of unity
LR = h*real( (exp(LR/2)-1)./LR ,2));
f1 = h*real( (-4-LR+exp(LR).*(-4-3+LR.*LR.^2))./LR.^3 ,2));
f2 = h*real( (2+LR+exp(LR).*(-2+LR))./LR.^2 ,2));
f3 = h*real( (-4-3*LR-LR.*2+exp(LR).*(-4-LR))./LR.^2 ,2));

uu = u; tt = 0;
tmax = 500; nmax = 1.5*round(tmax/h); nplt = 50;
```
% Define the damping parameter
\[ \delta \] is the damping of the trigonometric polynomial.

\[ g = -0.5i*k; \]
for \( n = 1:nmax \)
\[ t = n*h; \]
\[ Nv = g.*fft(real(ifft(v)).^2); \]
\[ a = E2.*v + Q.*Nv; \]
\[ Na = g.*fft(real(ifft(a)).^2); \]
\[ b = E2.*v + Q.*Na; \]
\[ Nb = g.*fft(real(ifft(b)).^2); \]
\[ c = E2.*a + Q.*(2*Nb-Nv); \]
\[ Nc = g.*fft(real(ifft(c)).^2); \]
\[ v = E.*v + Nv.*f1 + 2*(Na+Nb).*f2 + Nc.*f3; \]
if \( \text{mod}(n,nplt)==0 \)
\[ u = \text{real}(\text{ifft}(v)); \]
\[ uu = [uu,u]; \]
\[ tt = [tt,t]; \]
\[ C=n/nplt; \]
\[ \text{data}(C,1)=n*h/lambda; \]
\[ \text{data}(C,2)=\sqrt{4\pi}\text{norm}(v)/N; \]
end

% Plot the energy pattern.
hold on
plot(data((1:500),1),data((1:500),2),'k')
xlabel('t')
ylabel('|u|')

B.5 norm2.m

This program graphs the evolution of the norm \( |u| \) in the \( 4\pi \) domain. It loads the Fourier modes created by makeic.m and perturbs the physical solution by a \( 4\pi \)-periodic perturbation of the form
\[ \delta(P_1 \sin(x/2 + P_2)), \]
where \( \delta < 1 \) is the damping of the trigonometric polynomial.
N = 512;
tmax = 3000;
mu = .1; nu = .027;
g = 0.2;
delta = N*2*sqrt(1-exp(-2*g))*sqrt(3)/sqrt(4*pi);

% Transform by lambda.
clear alpha;
alpha(N/lambda,1)=0;
alpha((0:N-1)/lambda+1,1)=v/lambda;
N=N/lambda;
v=alpha;

% Perturbation of v.
vp=delta*rand(N/2,1).*exp((-1:-1:-N/2)*g)'.*exp(1i*2*pi*rand(N/2,1));
vp(N/2+1)=0;
vp(N/2+2:N)=flip(conj(vp(2:N/2)));

v=vp;
v(1)=0.0; % Make it zero-average.
x=(2*pi*(1:N)'/N)/lambda;
u=real(ifft(v));

mu=mu*lambda;
nu=nu*lambda^3;

% Method and integrator of Kassam and Trefethen's method (2005)
h = 1/100; % time step
k = [0:N/2-1 0 -N/2+1:-1]'; % wave numbers
L = mu*k.^2 - nu*k.^4; % Fourier multipliers
E = exp(h*L); E2 = exp(h*L/2);
M = 16; % no. of points for complex means
r = exp(1i*pi*((1:M)-.5)/M); % roots of unity
LR = h*L(:,ones(M,1)) + r(ones(N,1),:);
Q = h*real(mean( (exp(LR/2)-1)/LR ,2));
f1 = h*real(mean( (-4-LR+exp(LR)).*(4-3*LR+LR.^2)/LR.^3 ,2));
f2 = h*real(mean( (2+LR+exp(LR)).*(-2+LR)/LR.^3 ,2));
f3 = h*real(mean( (-4-3*LR-LR.^2+exp(LR)).*(4-LR)/LR.^3 ,2));

uu = u; tt = 0;
nmax = round(tmax*lambda/h); %nplt = floor((tmax/128)/h)/4;
nplt = 80;
g = -0.5i*k;
for n = 1:nmax
    t = n*h;
Nv = g.*fft(real(ifft(v)).^2);

a = E2.*v + Q.*Nv;

Na = g.*fft(real(ifft(a)).^2);

b = E2.*v + Q.*Na;

Nb = g.*fft(real(ifft(b)).^2);

c = E2.*a + Q.*(2*Nb-Nv);

Nc = g.*fft(real(ifft(c)).^2);

c = E2.*a + Q.*(2*Nb-Nv);

v = E.*v + Nv.*f1 + 2*(Na+Nb).*f2 + Nc.*f3;

v(1)=0.0;

if mod(n,nplt)==0
    u = real(ifft(v));
    %plot(x,u)
    %drawnow
    C=n/nplt;
    data(C,1)=n*h/lambda;
    data(C,2)=sqrt(4*pi)*norm(v)/N;
end

end

% Plot the energy pattern.
hold on
plot(data(:,1),data(:,2),'k')
xlabel('t')
ylabel('|u|')

B.6 rk4graph.m

This program takes the initial condition created by B.1 and rescales the wave into the 4π-periodic domain. It uses the Kassam–Trefethen integrator and animates the solution as it evolves in time.

lambda=1/2;
delta=1e-8;
N = 512;
tmax = 900;
mu = .1; nu = .027;

load('icv.mat','v');
x=(2*pi*(1:N)'/N);
u = real(ifft(v));

% Transform by lambda.
clear alpha;
alpha(N/lambda,1)=0;
alpha((0:N-1)/lambda+1,1)=v/lambda;
N=N/lambda;

% Perturb the largest Fourier modes.
delta1=N*delta/sqrt(8*pi);
alpha(2)=delta1; % Try putting delta1 at 4.
alpha(N)=delta1; % Try putting delta1 at N-2.
v=alpha;
x=(2*pi*(1:N)'/N)/lambda;
u=real(ifft(v));

mu=mu*lambda;
nu=nu*lambda^3;
plot(x,u,'-k')

% Method and integrator of Kassam and Trefethen's method (2005)
h = 1/100; % time step
k = [0:N/2-1 0 -N/2+1:-1]'; % wave numbers
L = mu*k.^2 - nu*k.^4; % Fourier multipliers
E = exp(h*L); E2 = exp(h*L/2);
M = 16; % no. of points for complex means
r = exp(1i*pi*((1:M)-.5)/M); % roots of unity
LR = h*L(:,ones(M,1)) + r(ones(N,1),:);
Q = h*real(mean( (exp(LR/2)-1)./LR ,2));
f1 = h*real(mean( (-4-LR+exp(LR).*(4-3*LR+LR.^2))/LR.^3 ,2));
f2 = h*real(mean( (2+LR+exp(LR).*(-2+LR))./LR.^3 ,2));
f3 = h*real(mean( (-4-3*LR-LR.^2+exp(LR).*(4-LR))./LR.^3 ,2));
uu = u; tt = 0;
nmax = round(tmax*lambda/h);
nplt = 20;
g = -0.5i*k;
for n = 1:nmax
t = n*h;
Nv = g.*fft(real(ifft(v)).^2);
a = E2.*v + Q.*Nv;
Na = g.*fft(real(ifft(a)).^2);
b = E2.*v + Q.*Na;
Nb = g.*fft(real(ifft(b)).^2);
c = E2.*a + Q.*(2*Nb-Nv);
Nc = g.*fft(real(ifft(c)).^2);
v = E.*v + Nv.*f1 + 2*(Na+Nb).*f2 + Nc.*f3;
if mod(n,nplt)==0
u = real(ifft(v));
uu = [uu,u]; tt = [tt,t];

C=n/nplt;
upic(C,:) = u;
data(C,1)=n*h/lambda;
data(C,2)=sqrt(4*pi)*norm(v)/N;
data(C,3)=sqrt(4*pi)*norm(u);
data(C,4)=sqrt(4*pi)*norm(v.*k.*k)/N;
data(C,5)=sqrt(4*pi/N)*norm(u(1:N/2)-u(N/2+1:N));
data(C,6)=sqrt(8*pi)*norm(v(2:2:N))/N;

if data(C,6)<=0.1
    TIME=C;
end
end

TIME(1:1024)=TIME;
makeimg2

\section*{B.7 makeimg.m}

This program generates a two-dimensional image of the solution evolving in a \(4\pi\)-periodic domain. It is written to be used after B.6 is executed. In this program, the two-dimensional surface is rendered as contour lines, with solid lines representing nonnegative values and dotted lines representing negative values. Finally, it saves the image as a PNG file with 600-point resolution.

\begin{verbatim}
figure
H=1024/4;
tp=1000;

% Set tick marks on axes.
a=min(min(upic));
b=max(max(upic));
image(256*((upic-a)/(b-a)).^1.2)
colormap(hsv(256))
M1=max(max(upic));
M2=-min(min(upic));
M=max(M1,M2);
U1=contour(upic(500:1750,:),(-1:.1:-0.1)*M,'k:');
hold on
U2=contour(upic(500:1750,:),(-0:.1:1)*M,'k-');
\end{verbatim}
% Spatial grid and initial condition:
lambda=1/2;
N = 512;
tmax = 1500;
mu = .1; nu = .027;

load('icv.mat','v');
x=(2*pi*(1:N)/N);
u = real(ifft(v));

% Transform by lambda.
clear alpha;
alpha(N/lambda,1)=0;
alpha((0:N-1)/lambda+1,1)=v/lambda;
N=N/lambda;

delta1=N*delta/sqrt(8*pi);
alpha(2)=delta1; % Try putting delta1 at 4.
alpha(N)=delta1; % Try putting delta1 at N-2.
v=alpha;

This function takes a value of $\delta$ as the damping for the perturbation, calculates the solution and stores the periodicity measure $p(u)$ every 40 iterations. It returns the time and periodicity measure to B.9, which will graph the time versus $p(u)$.
\[ x = \frac{(2\pi(1:N)'/N)}{\lambda}; \]
\[ u = \text{real}(\text{ifft}(v)); \]
\[ \text{mu} = \text{mu} \times \lambda; \]
\[ \text{nu} = \text{nu} \times \lambda^3; \]

% Method and integrator of Kassam and Trefethen's method (2005)
\[ h = 1/100; \quad \% \text{time step} \]
\[ k = [0:N/2-1 0 -N/2+1:-1]'; \quad \% \text{wave numbers} \]
\[ L = \text{mu} \times k.^2 - \text{nu} \times k.^4; \quad \% \text{Fourier multipliers} \]
\[ E = \exp(h \times L); E2 = \exp(h \times L/2); \]
\[ M = 16; \quad \% \text{no. of points for complex means} \]
\[ r = \exp(i \times pi \times ((1:M)-.5)/M); \quad \% \text{roots of unity} \]
\[ LR = h \times L(:,1:M,1) + r(ones(N,1),:); \]
\[ Q = h \times \text{real}(\text{mean}(\exp(LR/2)-1)./LR,2)); \]
\[ f1 = h \times \text{real}(\text{mean}(\text{(-4-LR+exp(LR).*((4-3*LR+LR.^2))/LR.^3 ,2))); \]
\[ f2 = h \times \text{real}(\text{mean}(\text{2+LR+exp(LR).*((-2+LR))/LR.^3 ,2))); \]
\[ f3 = h \times \text{real}(\text{mean}(\text{(-4-3*LR-LR.^2+exp(LR).*((4-LR))/LR.^3 ,2))); \]

uu = u; tt = 0;
nmax = round(tmax*lambda/h);
nplt = 40;
g = -0.5i*k;
for n = 1:nmax
t = n*h;
Nv = g.*fft(\text{real}(\text{ifft}(v)).^2);
a = E2.*v + Q.*Nv;
Na = g.*fft(\text{real}(\text{ifft}(a)).^2);
b = E2.*v + Q.*Na;
Nb = g.*fft(\text{real}(\text{ifft}(b)).^2);
c = E2.*a + Q.*(2*Nb-Nv);
Nc = g.*fft(\text{real}(\text{ifft}(c)).^2);
v = E.*v + Nv.^f1 + 2*(Na+Nb).*f2 + Nc.*f3;
if mod(n,nplt)==0
    u = \text{real}(\text{ifft}(v));
    uu = [uu,u]; tt = [tt,t];
    C=n/nplt;
    upic(C,:)=u;
data(C,1)=n*h/lambda;
data(C,2)=sqrt(4*pi)*norm(v)/N;
data(C,3)=sqrt(4*pi)*norm(u);
data(C,4)=sqrt(4*pi)*norm(v.*k.*k)/N;
data(C,5)=sqrt(4*pi)*norm(u(1:N/2)-u(N/2+1:N));
data(C,6)=sqrt(8*pi)*norm(v(2:2:N))/N;
This program uses a loop to call B.8 and gives various values of $\delta$. It retrieves the period measure for each value of $\delta$ and plots time versus the periodicity measure (1.3). Finally, it saves the stacked periodicity measure graphs as a PNG file with 600-point resolution.

```matlab
clear data1 dataNew
Delta=[1e-17 1e-14 1e-11 1e-8 1e-5 1e-2];
T = zeros(size(Delta));
data1=[];
dataNew=[];

for K=1:length(Delta)
    [data1, dataNew(:,K)]=period(Delta(K));
end

for K=length(Delta):-1:1
    hold on
    plot(data1, dataNew(:,K)+K,'k','LineWidth',0.6*K)
end
legend('
\delta=1e-2', '
\delta=1e-5', '
\delta=1e-8', ...
    '
\delta=1e-11', '
\delta=1e-14', '
\delta=1e-17')
set(gca,'YTickLabel',[]);
xlabel('t');
print('periodgraphs2','-dpng','-r600')
hold off
```

This function takes a value of $\delta$ as the damping for the perturbation, and it calculates the solution until such time as $p(u) \geq 0.1$. It sends this time $T$ to the program B.11.

```matlab
B.10 tdelta.m

This function takes a value of $\delta$ as the damping for the perturbation, and it calculates the solution until such time as $p(u) \geq 0.1$. It sends this time $T$ to the program B.11.
```
function T=tdelta(delta)

% Grid, final time, transformation parameter
lambda=1/2;
N = 512;
tmax = 100000000;

load('icv.mat','v');
x=(2*pi*(1:N)'/N);
u = real(ifft(v));
plot(x,u)

% Transform by lambda.
clear alpha;
alpha(N/lambda,1)=0;
alpha((0:N-1)/lambda+1,1)=v/lambda;
N=N/lambda;

delta1=N*delta/sqrt(8*pi);
alpha(2)=delta1; % Try putting delta1 at 4.
alpha(N)=delta1; % Try putting delta1 at N-2.
v=alpha;

x=(2*pi*(1:N)'/N)/lambda;
u=real(ifft(v));
plot(x,u)

mu = .1; nu = .027;
mu=mu*lambda;
u=nu*lambda^3;

% Method and integrator of Kassam and Trefethen's method (2005)

h = 1/100; % time step
k = [0:N/2-1 0 -N/2+1:-1]'; % wave numbers
L = mu*k.^2 - nu*k.^4; % Fourier multipliers
E = exp(h*L); E2 = exp(h*L/2);
M = 16; % no. of points for complex means
r = exp(1i*pi*((1:M)-.5)/M); % roots of unity
LR = h*L(:,ones(M,1)) + r(ones(N,1),:);
Q = h*real(mean( (exp(LR)-1)./LR ,2));
f1 = h*real(mean( (-4-LR+exp(LR).*((4-3*LR+LR.^2))/LR.^3 ,2));
f2 = h*real(mean( (2+LR+exp(LR).*(-2+LR))/LR.^3 ,2));
f3 = h*real(mean( (-4-3*LR-LR.^2+exp(LR).*((4-LR))/LR.^3 ,2));

uu = u; tt = 0;
nplt = 40;
g = -0.5i*k;
check=0;
n=1;
while check==0
    t = n*h;
    Nv = g.*fft(real(ifft(v)).^2);
    a = E2.*v + Q.*Nv;
    Na = g.*fft(real(ifft(a)).^2);
    b = E2.*v + Q.*Na;
    Nb = g.*fft(real(ifft(b)).^2);
    c = E2.*a + Q.*(2*Nb-Nv);
    Nc = g.*fft(real(ifft(c)).^2);
    v = E.*v + Nv.*f1 + 2*(Na+Nb).*f2 + Nc.*f3;
    if mod(n,nplt)==0
        u = real(ifft(v));
        uu = [uu,u]; tt = [tt,t];
        C=n/nplt;
        upic(C,:)=u;
        data(C,1)=n*h/lambda;
        data(C,2)=sqrt(4*pi)*norm(v)/N;
        data(C,3)=sqrt(4*pi/N)*norm(u);
        data(C,4)=sqrt(4*pi)*norm(v.*k.*k)/N;
        data(C,5)=sqrt(4*pi/N)*norm(u(1:N/2)-u(N/2+1:N));
        data(C,6)=sqrt(8*pi)*norm(v(2:2:N))/N;
        if data(C,6)>=0.1
            check=1;
            T=data(C,1);
        end
    end
    n=n+1;
end
hold off

B.11 tdeltacall.m

This program uses a loop to call B.10 and, for decreasing values of $\delta$, find the maximum time when $p(u) \leq 0.1$. After that it plots the $\delta$ values versus their respective times. Finally, it performs a least-squares analysis on these data points and saves the line and the data points as a PNG file with 600-point resolution.

Delta=10.^-(2:2:16);
T = zeros(size(Delta));

for K=1:length(Delta)
    T(K)=tdelta(Delta(K));
end
plot(log(Delta),T,'-k')
print('t_delta','-dpng','-r600')

A=[Delta' T'];

fileID=fopen('delta1new.txt','w');
fprintf(fileID,'%6.2e %15.8e
',A');
fclose(fileID);

% Do least-squares fitting.
B=[log10(Delta') ones(length(Delta),1)];
mb=B\T';
plot(log10(Delta),mb(1)*log10(Delta)+mb(2),'-k')
hold on
plot(log10(Delta),T,'ok')
hold off
xlabel('log_{10} \delta')
ylabel('T_\delta')
print('leastsq','-dpng','-r600');
Bibliography


