On statistical-mechanical descriptions of decaying two-dimensional turbulence

Zhaohua Yin
Zhaohua Yin
Fluid Dynamics Laboratory, Cascade 2.13
Eindhoven University of Technology
P.O.Box 513
5600 MB Eindhoven
The Netherlands

Cover illustration:
The Unknown Lake, Beida, 2003

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Yin, Zhaohua

On statistical-mechanical descriptions of decaying two-dimensional turbulence / by Zhaohua Yin. -
ISBN 90-386-1735-6
NUR 910
Trefw.: turbulentie / statistische mechanica / parallele programmering
Subject headings: turbulence / statistical mechanics / parallel programming
On statistical-mechanical descriptions of decaying two-dimensional turbulence

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof.dr. R.A. van Santen, voor een commissie aangewezen door het College voor Promoties in het openbaar te verdedigen op maandag 15 september 2003 om 16.00 uur

door

ZHAOHUA YIN

geboren te Qingdao, China
Dit proefschrift is goedgekeurd door de promotoren:

prof.dr. D.C. Montgomery
en
prof.dr.ir. G.J.F. van Heijst

Copromotor:
dr. H.J.H. Clercx

This work is sponsored by the Stichting Nationale Computerfaciliteiten (National Computing Facilities Foundation, NCF) for the use of supercomputer facilities, with financial support from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (Netherlands Organization for Scientific Research, NWO).
看西游记的人总会问：
孙悟空既然有那么大的神通，
为什么不带上唐僧一个筋斗翻上西天，
而要跟着他一步一步的走，
经历九九八十一难？

答案很简单：
唐僧的路只能靠他自己一步一步的走，
否则他就不能成佛。

冯友兰
Contents

1 Introduction
   1.1 General introduction to two-dimensional turbulence .............. 7
   1.2 Basic concepts for 2D flows .................................................. 9
   1.3 Laboratory experiments on quasi-2D turbulence ..................... 14
   1.4 Direct numerical simulations of 2D turbulence ..................... 15
   1.5 Thesis outline ................................................................. 17

2 Statistical mechanics and 2D turbulence ................................. 19
   2.1 Background ................................................................. 19
   2.2 Statistical mechanics in 2D turbulence ............................... 21
   2.3 Further remarks and recent developments ............................. 29

3 Predicting 2D decaying turbulence ....................................... 31
   3.1 Introduction ................................................................. 31
   3.2 Solving the nonlinear Poisson equation ............................... 35
      3.2.1 Introduction ............................................................. 35
      3.2.2 The solution of the sinh-Poisson equation ....................... 39
      3.2.3 The solution of the three-level Poisson equation ............ 40
   3.3 The energy-entropy (E-S) plots ........................................ 42
      3.3.1 Solution strategy for the E-S plot in the “point” theory ...... 42
      3.3.2 Solution strategy for the E-S plot in the “patch” theory ...... 43
   3.4 Prediction of the late-time decay of 2D turbulence ................ 43
   3.5 Conclusion ................................................................. 46

4 Numerical simulation of turbulence ....................................... 47
   4.1 Direct numerical simulation (DNS) with pseudospectral methods .... 47
   4.2 Parallelization of the 2D Fourier pseudospectral code ............ 49
      4.2.1 General introduction to parallel computation ................. 49
      4.2.2 Background ............................................................. 50
      4.2.3 Parallelization by task distribution ............................. 51
      4.2.4 Comparison with the former parallel scheme ................. 53
   4.3 Conclusion ................................................................. 58
5 Dynamical solutions and comparisons \hspace{1cm} 61
  5.1 Maximum entropy states \hspace{1cm} 61
    5.1.1 The classical dipolar final state \hspace{1cm} 61
    5.1.2 Observation of the “bar” final state \hspace{1cm} 65
    5.1.3 Further investigations on the “bar” final state \hspace{1cm} 71
  5.2 Local maximum states \hspace{1cm} 76
    5.2.1 Evolution from 64-pole initial condition \hspace{1cm} 76
    5.2.2 Evolution starting from a sinh-Poisson quadrupole \hspace{1cm} 81
  5.3 Summary of a few oddities \hspace{1cm} 84
    5.3.1 A run with features of both the “point” and the “patch” formulation \hspace{1cm} 84
    5.3.2 “Double-valued” structure in the $\omega$-$\psi$ scatter plot \hspace{1cm} 84
  5.4 Conclusion \hspace{1cm} 86

6 Conclusions and recommendations \hspace{1cm} 91

A Direct solution method of the linearized Poisson equation \hspace{1cm} 93

B A “smart” solver for Poisson-like equations \hspace{1cm} 95
  B.1 The strategy of the “smart” solver \hspace{1cm} 95
  B.2 The instruction of reading the flowchart \hspace{1cm} 96
  B.3 Conclusion \hspace{1cm} 99
Chapter 1

Introduction

1.1 General introduction to two-dimensional turbulence

Turbulence is an important phenomenon that can be observed in many different occasions in our daily life, both directly by watching the flow evolution and indirectly by viewing the dispersive properties of turbulent flows. For example, the turbulent character of a plume from a chimney is visualized by the smoke, and when you drop your tea bag into hot water turbulent mixing is apparent. As a first and rather simplified observation one can conclude that a turbulent flow field varies significantly and irregularly in time and space.

In all actual cases, turbulence is a three-dimensional phenomenon. From everyday experience we know that turbulent flows increase the "disorder" in a fluid, resulting in a rather efficient mixing of material elements. From experiments with passive tracers in turbulent flows it is clearly shown that large vortical structures in such a flow, usually called eddies, are unstable and break up into smaller eddies. In turn, these smaller eddies are again unstable and break up into even smaller ones. This process continues down to the Kolmogorov length scale where advection and diffusion are in balance. The kinetic energy of the flow is eventually dissipated, i.e. transformed into heat, by molecular processes. Hence, there is a spectral energy flux from the large to the small (Kolmogorov) length scales, and this phenomenon is known as the "energy cascade" in three-dimensional (3D) turbulence. Phenomenologically, this cascade expresses itself in the development of a relatively disorderly flow from an initially rather ordered flow.

Although turbulence is a three-dimensional phenomenon it does not exclude the possibility of strong anisotropies. For example, severe geometrical confinement induces a strong anisotropy of the large-scale turbulent motion in the atmosphere and in the oceans. The characteristic horizontal velocity and length scales are substantially larger than their vertical counterparts and the turbulent flow can be treated like a quasi-two-dimensional (quasi-2D) or even like a two-dimensional (2D) phenomenon. The quasi-two-dimensionality of the turbulent motions in planetary flows are not only due to geometrical confinement of the flow, but also body forces related with the density stratification of the fluid (for example, the temperature stratification in the atmosphere
Chapter 1. Introduction

Figure 1.1: Jupiter’s Great Red Spot, the oval at the top of the picture, is a very large vortex. The picture shown here was taken by NASA’s Voyager 1 spacecraft on 25 February 1979. The Great Red Spot (GRS) is known to exist since the invention of the telescope (in 1610), hence it is a rather stable vortex, undergoing only minor variations in size and colour (between pale pink and brick red).

and the combined temperature and salt stratification in the oceans) and the planetary rotation, or with a combination of both, which have severe consequences for the flow dynamics. The formation and the dynamics of large-scale vortices in the oceans can be understood by taking these body forces into account. However, most of us and surely the general public are more familiar with a similar example of the existence of a large-scale vortex: the Great Red Spot of Jupiter (see Fig. 1.1). A good understanding of the origin and the dynamics of these large-scale vortices is of great importance, because of their role in the global transport of matter and heat.

The investigations reported in this thesis concern the decay properties of 2D turbulence. From a mathematical point of view, 2D turbulence can be treated like a simplified version of 3D turbulence. However, a striking difference between the dynamics of 2D and 3D turbulence exists. In contrast to its 3D counterpart, 2D turbulence is characterized by “self-organisation” of the flow\(^1\). The main difference between the dynamics of 3D and 2D turbulence lies in the presence and absence of “vortex stretching” in the former and the latter, respectively. Consider a vortex tube in a 3D (not necessarily turbulent) flow. If this vortex tube is stretched, matter is pulled towards the rotation axis of the vortex tube, and due to conservation of angular momentum it will rotate faster (this is comparable to the pirouette of a figure-skater, who can increase his rotation rate by bringing both arms against his body). In a 2D flow the mechanism of vortex stretching is absent, and for inviscid 2D flows the vorticity of a two-dimensional fluid element is thus conserved. Together with the conservation of total kinetic energy

\(^1\)Small vortices in a 2D flow tend to organize spontaneously in large-scale coherent structures.
1.2 Basic concepts for 2D flows

The Navier-Stokes equation and the equation for mass conservation for a 3D incompressible flow read

\[
\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v},
\]

\[
\nabla \cdot \mathbf{v} = 0,
\]

with \( \mathbf{v} = (u, v, w) \) the velocity field (with \( u, v \) and \( w \) the velocity components with respect to a Cartesian coordinate system \((x, y, z)\)), \( \rho \) the density, \( p \) the pressure and \( \nu \) the kinematic viscosity of the fluid. We assume here, and elsewhere in this thesis, that the fluid density is constant and equal to unity. The vorticity equation is obtained by taking the curl of equation (1.1) which yields

\[
\frac{\partial \omega}{\partial t} + (\mathbf{v} \cdot \nabla) \omega = (\mathbf{v} \cdot \nabla) \mathbf{v} + \nu \nabla^2 \omega.
\]

The first term on the right hand side of equation (1.2) represents vortex stretching and tilting, and is assumed to be negligible for quasi-2D flows if the vorticity field is reduced to

\[
\omega = (0, 0, \omega) = \nabla \times \mathbf{v},
\]

and some scaling arguments are used. For example, for rotating flows the Taylor-Proudman theorem is used to show that \( \partial v/\partial z = 0 \) provided the rotation axis is parallel to the \( z \)-axis of the coordinate system. In the two-dimensional case, the Navier-Stokes equation (1.1) is thus simplified into the scalar vorticity equation

\[
\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \nu \nabla^2 \omega.
\]

Mass conservation, i.e. \( \partial u/\partial x + \partial v/\partial y = 0 \) (note that from now on we use \( \mathbf{v} = (u, v) \)), is implicitly satisfied by introducing the stream function \( \psi \) according to

\[
u = \frac{\partial \psi}{\partial y},
\]

\[
v = -\frac{\partial \psi}{\partial x}.
\]

Summarizing, we have to solve the following set of partial differential equations to solve for the two-dimensional flow field

\[
\frac{\partial \omega}{\partial t} + \left( \frac{\partial \omega}{\partial x} \frac{\partial \psi}{\partial y} - \frac{\partial \omega}{\partial y} \frac{\partial \psi}{\partial x} \right) = \nu \nabla^2 \omega
\]

(1.6)
and
\[ \omega = -\nabla^2 \psi . \]  
(1.7)

A shorthand notation for the nonlinear contribution to Eq. (1.6) is the Jacobian
\[ J(\omega, \psi) = \frac{\partial \omega}{\partial x} \frac{\partial \psi}{\partial y} - \frac{\partial \omega}{\partial y} \frac{\partial \psi}{\partial x} . \]  
(1.8)

For inviscid flows Eq. (1.6) reduces to the Euler equation
\[ \frac{D\omega}{Dt} = \frac{\partial \omega}{\partial t} + J(\omega, \psi) = 0 , \]  
(1.9)

which states that the vorticity of a fluid element is conserved for inviscid flows (note
that \( D/Dt \) represents the material derivative).

We can see that if there exists a functional relation \( \omega = f(\psi) \), we will get
\[ J(\omega, \psi) = \frac{\partial f(\psi)}{\partial x} \frac{\partial \psi}{\partial y} - \frac{\partial f(\psi)}{\partial y} \frac{\partial \psi}{\partial x} = \left( \frac{\partial f}{\partial \psi} \frac{\partial \psi}{\partial x} \right) \frac{\partial \psi}{\partial y} - \left( \frac{\partial f}{\partial \psi} \frac{\partial \psi}{\partial y} \right) \frac{\partial \psi}{\partial x} = 0 . \]  
(1.10)

This means that the experimental or numerical observation of the functional relation-
ship \( \omega = f(\psi) \) indicates the presence of a stationary state of inviscid flow. Usually,
this observation is treated as an indication of the presence of a nearly steady state of
high Reynolds number flows, i.e., the case when \( \nu \to 0 \).

Although the \( (\omega, \psi) \) relation is an important tool in the characterization of the so-
called “final” state of decaying 2D turbulence, we should notice that \( \omega = f(\psi) \) is a
sufficient but not necessary condition of the near-equilibrium states of high Reynolds
number flows.

In Figs. 1.2, we see a good example of the functional relationship between \( \omega \) and \( \psi \)
- the famous Lamb dipole (for details see [1, 2]). It has the linear relation of \( \psi - \omega \) (for
\(-0.5 < \Psi < 0.5 \)). Another example is the sinh-Poisson relationship [3–5], of which we
will give a thorough discussion in the next few chapters.

In Figs. 1.3, we see a double-valued structure of the \( \omega - \psi \) plot and the associated
contour plot\(^2\). Apparently, the flow is not described by a functional relation between \( \omega \)
and \( \psi \) anymore, but nevertheless it manages to get to one specific kind of equilibrium.
In Section 5.3.2, we will give a more detailed discussion of this special case of \( \omega - \psi \)
relations.

Consider an unbounded two-dimensional domain \( \mathcal{D} \) (or a doubly-periodic square
domain \( \mathcal{D} \)). The kinetic energy of the flow (with unit density) is defined as
\[ E = \frac{1}{2} \int_{\mathcal{D}} |\mathbf{v}|^2 \, dx \, dy . \]  
(1.11)

\(^2\)Such relation, which is also called “multi-valued function” sometimes, is different from the ordinary
definition. In Eq. 1.10, the relation we use to derive \( J(\omega, \psi) = 0 \) is normal sense of function, which is
so-called one-to-one or many-to-one function (for one given value of \( x \), we can get only one value of \( y \)
through relation \( y = f(x) \) ).
1.2. Basic concepts for 2D flows

Figure 1.2: The Lamb dipole with a linear relationship between \( \omega \) and \( \psi \), can be considered as a stationary two-dimensional solution of the Euler equations. The figure on the right indicates the vorticity field, with one positive vortex and one negative vortex confined within the circle. Outside that circle, the vorticity is zero.

Figure 1.3: The double-valued structure, representing a non-functional \( \omega - \psi \) relation, still is a stationary solution of the Euler equation. The associated vorticity contour plot is shown in the right panel.
In a similar way we define the enstrophy and the palinstrophy of the flow,

$$\Omega = \frac{1}{2} \int_D \omega^2 \, dx \, dy$$

(1.12)

and

$$P = \frac{1}{2} \int_D |\nabla \omega|^2 \, dx \, dy.$$  

(1.13)

The time rate of change of the kinetic energy, $dE/dt$, can easily be related with the enstrophy of the flow by taking the inner product of the two-dimensional analogon of Eq. (1.1) with $v$, integrating over the flow domain and using the proper boundary conditions:

$$\frac{dE}{dt} = -2\nu \Omega,$$

(1.14)

reflecting the property that the kinetic energy of the flow always decreases as a result of viscous dissipation (note that $\Omega \geq 0$). A similar approach, but now multiplying the 2D vorticity equation (1.4) with $\omega$, integrating over the flow domain and using the proper boundary conditions for the vorticity and velocity yields an expression for the time rate of change of the enstrophy, $d\Omega/dt$, in terms of the palinstrophy,

$$\frac{d\Omega}{dt} = -2\nu P.$$  

(1.15)

Since the palinstrophy is positive, the enstrophy is also bounded by its initial value, i.e. $\Omega(t) \leq \Omega(t = 0)$ [6]. It should be noted that this conclusion is valid for 2D flows in unbounded or doubly-periodic domains\(^3\). From the observation $\Omega(t) \leq \Omega(t = 0)$ we can directly conclude that $dE/dt \to 0$ when $\nu \to 0$.

The kinetic energy, the enstrophy and the palinstrophy can also be computed in Fourier space, yielding relations more convenient for theoretical interpretation. Moreover, these relations are particularly suited to compute these global quantities from the Fourier components of the vorticity field as obtained in Direct Numerical Simulations (DNS) of the 2D vorticity equation. For the kinetic energy of the flow we can rewrite Eq. (1.11) as

$$E = \frac{1}{2} \sum_k |v(k)|^2 = \frac{1}{2} \sum_k \frac{|\omega(k)|^2}{k^2} = \int_0^\infty \tilde{E}(k) \, dk,$$

(1.16)

where we have used

$$\omega(r) = \sum_k \omega(k) e^{ikr},$$

(1.17)

with $r = (x, y)$, $k = (k_x, k_y)$, the wave number $k = |k|$, and $\tilde{E}(k) = 2\pi k E(k)$ the so-called omnidirectional energy. In the derivation of Eq. (1.16) we have also used the following properties for incompressible 2D flows: $\omega(k) = ik_x v(k) - ik_y u(k)$ and $k_x u(k) = -k_y v(k)$. In a similar way we obtain for the enstrophy and the palinstrophy

$$\Omega = \frac{1}{2} \sum_k |\omega(k)|^2 = \int_0^\infty k^2 \tilde{E}(k) \, dk$$

(1.18)

\(^3\)For bounded domains with no-slip or stress-free walls additional boundary contributions should be added to the right hand side of Eq. (1.15) (but not to Eq. (1.14)). In particular, the condition $\Omega(t) \leq \Omega(t = 0)$ for flows in domains with no-slip boundaries is invalidated.
1.2. Basic concepts for 2D flows

Figure 1.4: In one of our direct simulations of 2D turbulence (to be discussed later), the value of \( \frac{\Omega}{E} \) is initially locked near the value of two. After some time, it drops and reaches a value close to one finally.

and

\[ P = \frac{1}{2} \sum_{k} k^2 |\omega(k)|^2 = \int_{0}^{\infty} k^4 \tilde{E}(k) dk, \quad (1.19) \]

respectively.

Using Eqs. (1.16), (1.18) and (1.19), we can derive

\[ \frac{d}{dt} \left( \frac{\Omega}{E} \right) = -\frac{\nu}{E^2} \int \left( k^2 - (k')^2 \right)^2 \tilde{E}(k) \tilde{E}(k') dk dk' \leq 0, \quad (1.20) \]

which means that the value of \( \frac{\Omega}{E} \) will always become smaller except when all the energy is concentrated in a single wavenumber \( k \). The wave number dependent kinetic energy \( E(k) \) of the flow in 2D inviscid fluid dynamics (\( \nu = 0 \)) is redistributed in such a way, keeping the total kinetic energy and the total entrophy of the flow constant (see Eqs. (1.14) and (1.15) with \( \nu = 0 \)), that the kinetic energy is mainly transported to the lower wavenumbers (the so-called inverse energy cascade). As a direct result, \( \frac{\Omega}{E} \) will decay until it reaches the value one (the smallest wavenumber). However, it does not exclude the possibility that in some circumstance (for example, some local equilibrium states) \( \frac{\Omega}{E} \) will be locked in a higher wavenumber for a certain long period. This is illustrated in Fig. 1.4, where we have plotted the time evolution of \( \frac{\Omega}{E} \) for a particular numerical simulation (which will be discussed in more detail in section 5.2.2 of this thesis).

The inverse energy cascade leads to the most striking phenomenon in 2D turbulence – the emergence of coherent structures. No matter what kind of random initial condition is used in a DNS of freely evolving 2D turbulence (usually in a doubly-periodic domain), the merger of vortices is always the dominating feature, with the decreasing total number of vortices and increasing average size of existing vortices. When the Reynolds number of the flow is sufficiently high, the merging process continues until a
long-lived stable structure is formed, without a substantial decrease of the total kinetic energy of the flow. The structure, which is normally referred as quasi-stationary state, has a size comparable to the width of the computational domain.

1.3 Laboratory experiments on quasi-2D turbulence

Although the research reported in this thesis has a theoretical and numerical character it is still worthwhile to present a brief overview of recent experimental investigations on quasi-2D turbulence. Some of the experimental investigations approach the rather academic concept of 2D turbulence quite closely, putting present investigation in a more general framework.

Purely two-dimensional turbulence does not exist in laboratory experiments, but it is possible to set up experimental flow configurations where vortex stretching is suppressed (supporting conservation of vorticity of fluid elements in inviscid flows), and typical 2D flow phenomena such as self-organization and interactions of coherent vortices. Previous experiments to study (quasi-)2D turbulence in the laboratory have used a wide variety of forcing mechanisms to generate quasi-2D turbulence in order to follow its subsequent decay, or to maintain a quasi-2D turbulent flow during the run of an experiment. Moreover, different ways of making the flow quasi-2D have been employed. The first category of experiments to be discussed briefly are those where geometrical confinement is used. In this brief survey we will not make a distinction between forced and decaying turbulence experiments, although the work presented in this thesis concerns 2D decaying (Navier-Stokes) turbulence.

Shallow layers of electrolyte that were subjected to forcing by a magnetic field were pioneered by groups in France and in Russia [7, 8]. These studies focussed on the role of bottom friction on the evolution of simple coherent structures and on the decay of 2D turbulence (including comparison with some recent theoretical predictions of the decay process). Soap films provide another way of making a flow geometrically thin and quasi-2D. This kind of experiment was for the first time reported by Couder [9], and recently several forced and decaying 2D turbulence experiments were reported by Kellay and coworkers [10, 11], who investigated decaying turbulence by towing a comb through these soap films and showed evidence for the presence of a direct enstrophy cascade in decaying 2D turbulence by means of homodyne photon correlation spectroscopy and by optical fiber velocimetry, and by Rutgers [12]. Rutgers has claimed to have measured the simultaneous presence of the $k^{-5/3}$ and $k^{-3}$ spectrum in forced 2D turbulence using laser Doppler velocimetry. It should be emphasized, however, that the quasi-2D turbulent flow in these soap film experiments shows some lack of horizontal incompressibility of the flow as a consequence of film-thickness fluctuations supporting local vortex squeezing and stretching effects.

Flows can also be made quasi-2D using strong stratification (both in linearly stratified fluids as well as at the interface in two-layer fluids) as shown during the last decade by, for example, van Heijst and Flör [13], Voropayev et al. [14], Yap and van Atta [15], Fincham and coworkers [16] and Maassen et al. [17, 18]. Most of these experiments are carried out for grid or jet-forced stratified turbulence and show a complicated transition
1.4 Direct numerical simulations of 2D turbulence

from a 3D to a quasi-2D flow. One of the most remarkable observations is the emergence of large-scale coherent structures from initially small-scale turbulence, a clear feature of quasi-two-dimensionality of the freely evolving flow. Recently, quasi-2D decaying turbulence in stratified fluids in square and circular containers by Maassen et al. [17,18] have been compared with DNS of 2D decaying turbulence in such geometries by Li and Montgomery (circular containers) [19,20] and by Clercx et al. [21] (square containers) and Maassen [22,23] (rectangular containers). The agreement between the flow dynamics observed in the simulations and in the experiments were striking, and supporting the conjecture that experiments to investigate 2D turbulence are indeed feasible.

Experiments in rapidly rotating fluids, which by the Taylor-Proudman theorem will be essentially two-dimensional in character, have also been conducted. Unfortunately, not many experiments are reported on decaying or forced quasi-2D turbulence. Some well-known experiments on rotating turbulence are reported by Collin de Verdière [24] on quasi-geostrophic turbulence in a rotating homogeneous fluid and by Hopfinger et al. [25,26].

Although all these (turbulent) flows are far more complex than the pure 2D flows, laboratory experiments have revealed that the “simple” model of 2D turbulence is certainly useful to understand the dynamical behaviour of realistic quasi-2D (turbulent) flows.

1.4 Direct numerical simulations of 2D turbulence

Numerical modelling of turbulence is a rapidly growing field of research for several important reasons. The need for turbulence simulations can be understood by realizing its very importance for many industrial processes, several branches of (engineering) sciences and for fundamental turbulence research. Obviously, the development of fast computers and multiprocessor systems supported a rapid development of modelling, including Reynolds Averaged Navier Stokes (RANS) simulations and Large Eddy Simulations (LES), and Direct Numerical Simulations (DNS) of many fundamental turbulence problems. A few of these fundamental problems concern 3D homogeneous and isotropic turbulence, 3D homogeneous turbulence with mean shear, rotating turbulence, stratified turbulence and the idealized case of homogeneous and isotropic 2D turbulence. Pure 2D turbulence can be simulated by solving (the two-dimensional variant of) the set of partial differential equations (1.1) or, equivalently, the set represented by Eqs. (1.4) and (1.7) when proper initial and boundary conditions are specified. Any three-dimensional effects are then excluded.

Many different methods exist to discretize partial differential equations and in the field of fundamental turbulence research (assuming simple flow geometries) the application of spectral methods have become a very popular and successful tool for DNS. Although we have no intention to downplay the importance of many other methods in turbulence simulations we will restrict ourselves to a brief outline of spectral methods only. An extensive survey of the application of spectral methods in fluid dynamics and its mathematical foundations can be found in the book by Canuto et al. [27].
Spectral methods may be viewed as an extreme development of the class of discretization schemes for differential equations known generically as the methods of weighted residuals (MWR). The key elements of the MWR are the trial functions (also called the expansion or approximating functions) and the test functions (also known as weight functions). The trial functions are used as the basis functions for a truncated series expansion of the solution. The test functions are used to ensure that the differential equation is satisfied as closely as possible by the truncated series expansion. This is achieved by minimizing the residual, i.e., the error in the differential equation produced by using the truncated expansion instead of the exact solution, with respect to a suitable norm. An equivalent requirement is that the residual satisfies a suitable orthogonality condition with respect to each of the test functions.

The choice of trial functions is one of the features which distinguish spectral methods from finite-element and finite-difference methods. The trial functions for spectral methods are infinitely differentiable global functions. In the case of finite-element methods, the domain is divided into small elements, and a trial function is specified in each element. The trial functions are thus local in character, and well suited for handling complex geometries. The finite-difference trial functions are likewise local. The relatively low accuracy of these two discretization methods is an obstacle to the accurate representation of complex flows. In this thesis, 2D turbulence in doubly-periodic domains is investigated and the most logical choice of a set of expansion functions is the set of Fourier functions for both coordinate directions.

The success of the Fourier methods for turbulence computations was due to the efficiency of the fast Fourier transform (FFT) algorithm for the calculation of the nonlinear terms (the Jacobian $J(\omega,\psi)$, see Eq. (1.8)) through the "pseudospectral" technique: the differentiations are made in the spectral space and the products are calculated in the physical space. The connection between both spaces is made by the FFT algorithm.

As we mentioned above, DNS of turbulence has experienced great developments recently because of the increasing power of supercomputers (with multiprocessor capability). However, there are still a lot of limitations if we want to perform very high resolution simulations to improve our DNS results. Actually, we end up with the almost classical complaint by computational fluid dynamicists: no computer is fast enough. Therefore, we cannot rely completely on the chip and computer manufacturing companies for faster computers, but have to apply alternative strategies to improve our DNS and to reduce the requested CPU time.

To deal with this problem, the first choice is, of course, making the code faster by changing the numerical scheme, to apply a different modelling procedure (e.g., applying LES instead of DNS to simulate a turbulent flow thereby representing the small-scale turbulence by a subgrid scale model), or to parallelize the numerical algorithm. Simply changing the applied numerical scheme, perhaps will make the code faster, but it will hurt some other features of the simulation. For example, in the finite difference methods, it is possible to replace the implicit scheme by an explicit scheme, but this kind of change will inevitably hurt the stability of the scheme. In the field of DNS of 2D turbulence, several investigators have addressed this problem from an entirely different
1.5. Thesis outline

In this thesis, we use statistical-mechanical descriptions of ideal Euler fluids and direct numerical simulations of the Navier-Stokes equations to study the quasi-stationary final states of decaying 2D turbulence. Efforts to investigate 2D ideal fluids by methods from statistical mechanics started some 30 years ago, and it is proven to be quite useful (and successful) to explain several phenomena of decaying 2D turbulence, in particular the long-time quasi-stationary final states of the flow. This Ph.D research is aimed at testing the ability of two different statistical-mechanical approaches, one based on discretization of the initial flow field in terms of point vortices and the other in terms of mutually-exclusive patches of uniform vorticity, to predict the details of the final structure in decaying 2D turbulence using direct numerical simulation of high-Reynolds number flows. For the first time, a so-called one-dimensional “bar” structure as a quasi-stationary final state of decaying 2D turbulent flows is taken into consideration. This final state emerges under certain circumstances in favour of the classical result: the dipolar final state. We find that under different initial assumptions,
the two different statistical-mechanical approaches ("point" versus "patch" theory) give
different predictions. These predictions are tested with direct numerical simulations of
the two-dimensional Navier-Stokes equations.

Present study required many numerical simulations to make the necessary compar-
isons, and successful efforts have been made to parallelize the existing numerical code
using MPI (Message Passing Interface), which has also proved to be very successful.

In chapter 2, the two existing statistical mechanical methods, the so-called "point"
theory and the "patch" theory are introduced, both of which are based on the Euler
equation for inviscid fluid dynamics. I am indebted to Prof. David C. Montgomery, who
provided a compact theoretical explanation of these methods. They are now known and
available as the Spring Notes in our group. I believe it is an irreplaceable work, which
has not been published in the open literature in the present form, and it is essential to
completely understand the investigations reported in this thesis. With his permission,
I have put a slightly modified version of these Notes in chapter 2. Solution strategies
to obtain the energy-entropy plots for both the "point" and the "patch" theory are
provided in chapter 3, where also the entropy of the two theories are compared under
the condition of the same energy and vorticity flux. For the first time, the one-dimensional
"bar" solution is taken into consideration in a doubly-periodic domain. This most time-
consuming part of this research predicts that 2D decaying turbulence may have different
final states under different initial conditions: a "bar" or a "dipole". The validity of this
prediction is proved by the DNS results shown in chapter 5. In chapter 4, we describe
the details of the pseudospectral code that has been used for the numerical simulation
of 2D decaying Navier-Stokes turbulence. Parallelization of the existing numerical
code using MPI (Message Passing Interface) has been described. Our method can
easily be implemented and is very efficient. This part of the investigation turned out
to be indispensable; without extremely time-consuming parallel computations many of
the results presented in this thesis could not be produced. We have made a comparison
between the numerical results and the statistical predictions in chapter 5. We mainly
performed simulations to verify the theory results introduced in chapter 3. The "bar"
solution, which is predicted by the theory, was discovered repeatedly. The conclusions
are summarized in chapter 6. In the two appendices, we describe some technical details
of our research. In appendix A, we give a stable scheme to get the nontrivial solution
of the Poisson-like equations introduced in chapter 3, and appendix B shows how to
make the Poisson solver smarter.

The main results presented in chapter 3 and 5 are published in Physics of Fluids
(Alternative statistical-mechanical descriptions of decaying two-dimensional turbulence
in terms of “patches” and “points,” by Yin, Montgomery and Clercx, Phys. Fluids
15, 1937-1953 (2003)). A Modified version of chapter 4 has been accepted for publica-
tion in Computers and Fluids (An easily implemented task-based parallel scheme 2D
Fourier pseudo-spectral solver applied to 2D Navier-Stokes turbulence, by Yin, Clercx
and Montgomery).
Chapter 2

Statistical mechanics and 2D turbulence

2.1 Background

It has been known for several years that two-dimensional Navier-Stokes (2D NS) turbulence with periodic boundary conditions and at high Reynolds numbers (in excess of a very few thousand) will relax to long-lived, quasi-steady states whose topology is preserved and whose energy decays at a rate more or less inversely proportionally to the Reynolds number, computed with respect to a box dimension and an rms initial turbulent velocity. When this energy decay time is large compared to the initial eddy turnover time, the quasi-steady state can be reached at a time when the total energy decay is fractionally small, though the enstrophy decay can be fractionally large. It came as a surprise when, a decade or more ago, a series of such computations [3–5] revealed that the late-time quasi-steady state for an initially turbulent run at a Reynolds number above 14,000 showed a pointwise hyperbolic sinusoidal dependence between stream function and vorticity.

The prediction of such a dependence, in the context of a mean-field treatment of ideal line vortices (or guiding-center plasma rods, see Fig. 2.1(a)) had been given thirty years ago [31,32] and has since been extended and refined in a series of investigations by several groups [33–47]: in every case referring to ideal, non-viscous systems. The system is Hamiltonian with a finite phase space, and it is natural to apply Boltzmann statistics to its dynamics, as originally suggested by Onsager [48] (see also Lin [49]). The surprise came in the extent to which the ideal Euler mean-field predictions fit the Navier-Stokes results. At least one attempt was made to define an entropy for the case of finite viscosity [50] but generated new puzzles of its own.

In the late 1980s and early 1990s, an alternative formulation was given by Robert, Sommeria and Chavanis [51–54], by Miller and colleagues [55] and later explored by Brands, Maassen and Clercx [56]. The principal difference was that the vorticity field was discretized not in terms of delta-functions, but rather in terms of finite-area, mutually-exclusive “patches” of vorticity, to which Lynden-Bell statistics [57] could be applied (see Fig. 2.1(b)). The choice of parameters in the patch formulation is wider
than that of equal-strength point vortices, in that one must choose in advance the size of the patches, and the number of “levels,” or strengths, that the patches carry. There seems to be no deductive mechanism for making such choices, and so we stick with the simplest, taking equal-area patches and never more than three levels, including zero.

The derivations of the “point” and “patch” formulations can be given in terms of straightforward but tedious operations which are frequently treated in an introductory statistical mechanics course. The “point” formulation can be obtained as a limit of the “patch” formulation. In the latter, the vorticity is coarse-grained by replacing it by its average over finite spatial areas that are regarded as non-overlapping and tile the basic box. For convenience, they can be taken as of equal area. The box in the $x$, $y$ plane is ruled off into equal cells each containing one or more of the mutually-exclusive patches of vorticity, of different values or “levels.” One of these levels may be zero; i.e., each cell or some fraction of it may be empty of vorticity. The expression adopted for the probability of any configuration (which now amounts to a coarse-grained specification of the vorticity as a function of $x$ and $y$) is the same one as used in Lynden-Bell statistics (or Fermi-Dirac statistics, lacking the quantum complications [55,57]). In Table2.1, a summary of the four types of statistical-mechanical approach is shown.

The logarithm is maximized subject to the constraints of constant total (discretized) energy and constant total fluxes associated with each vorticity level. The method of Lagrange multipliers is used, and the numbers are considered large enough for Stirling’s approximation to apply. The result is a “most probable” vorticity distribution associated with that energy and set of vorticity fluxes. The “most probable” vorticity states so obtained often lead to only local, rather than absolute, entropy maxima,
Table 2.1: Four types of statistics.

<table>
<thead>
<tr>
<th>Indistinguishable particles</th>
<th>Distinguishable particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>No exclusion</td>
<td>Einstein-Bose statistics</td>
</tr>
<tr>
<td></td>
<td>Maxwell-Boltzmann statistics (&quot;point&quot; theory)</td>
</tr>
<tr>
<td>Exclusion</td>
<td>Fermi-Dirac statistics</td>
</tr>
<tr>
<td></td>
<td>Lynden-Bell statistics</td>
</tr>
<tr>
<td></td>
<td>(&quot;patch&quot; theory)</td>
</tr>
</tbody>
</table>

and further comparisons are required to determine which one is the distribution that actually maximizes the assumed entropy globally. A mean-field approximation is then assumed, in which the sizes of the patches and cells are regarded as negligible, and the patches become more and more numerous. The vorticity distribution is thereafter treated as continuous and differentiable. It is expressed as a function of the stream function, now itself continuous. A final step involves inserting the expression for the entropy-maximizing vorticity field into Poisson’s equation, which must then be solved self-consistently (and numerically) for the associated stream function. One parameter in the procedure remains arbitrary and has no physical basis for its choice: the ratio of the “patch” size to the cell size. It should be kept in mind that throughout, one has in mind a representation of an ideal, conservative, Euler equation system, with no viscous dissipation of any kind.

2.2 Statistical mechanics in 2D turbulence

In this section, we attempt to review the most elementary calculations of the “most probable” vorticity distribution, inside a periodic square domain, that is compatible with a fixed value of the total kinetic energy and fixed (equal) fluxes of positive and negative vorticity. We do this for the case of Lynden-Bell statistics, which can then be specialized to Boltzmann statistics as a special case. It should be emphasized that in both pictures, vorticity is regarded as a pointwise conserved quantity, so that viscosity is not involved. This review is largely based on a private communications with D.C. Montgomery.

Consider the problem of rearranging a continuous periodic vorticity distribution \( \omega(x, y; t = 0) \) to constitute a “more probable” arrangement of the vorticity elements. This can only be approached through some kind of discretization. We consider coarse graining the vorticity distribution in the following sense.

Cut the periodic box into many cells of equal volume; \( i \) will label the cells. The cells are of area \( \Delta \). The coarse-grained vorticity distribution will be defined by the numbers

\[
\omega_i = \frac{1}{\Delta} \int \int_{cell i} \omega(x, y) \, dx \, dy.
\]  

We now represent the vorticity \( \omega_i \) inside the \( i \)th cell in terms of smaller sub-units. These can clearly either have finite areas ("patches") or zero areas ("points"). We should be able to take the "patch" formulation and achieve the "point" formulation by suitable limiting processes, so we start with the patches. The ultimate question will
be which, if either, satisfactorily represents the long-time state of the Navier-Stokes
evolution of the $\omega(x, y; t = 0)$ at high Reynolds number. It is an academic and not-
very-interesting question as to which one better predicts the long-time Euler equation
evolution of $\omega(x, y; t = 0)$, since we can never compute that anyway.

Note that if $\omega(x, y; t = 0)$ is an analytic function and we look at the set of points $x,$
y at which $\omega$ takes on any value $\omega_0,$ say, that set is always a set of measure zero—i.e.,
zero area. So arriving at any finite number of “levels” will always implicitly assume
that some kind of “coarse-graining,” as in Eq. (2.1), has already been done.

We now cut each cell up into equal areas (they do not have to have any particular
shape), each one the size of a “patch.” We say $M_i$ is the maximum number of patches
allowed in the $i$th cell. Without apparent loss of generality, we can take all the patches
as of equal area, but we will carry along the index $i$ on $M_i$. There is no obvious
theoretical basis for choosing the size of a “patch,” $\Delta / M_i$, except that it is far smaller
than $\Delta$, so $M_i \gg 1$.

We assume there are different “strengths” of the patches, though this will later come
to seem perhaps unnecessary. Let the subscript $j$ represent the “species,” or strength of
vortices of type $j$, which will be called $K_j$. $K_j$ is the integral of the (uniform) vorticity
of the patch over its area. Let $j = 0, 1, 2, 3...q$, where $K_0 = 0$ (an “empty” site, with no
patch in it), and $q$ different strengths represented, which can be positive or negative,
but not zero.

Giving a set of occupation numbers $N_i^j$ will give a discretized microscopic representa-
tion of all the $\omega_i$:

$$\tilde{\omega}_i = \frac{1}{\Delta} \sum_j N_i^j K_j.$$  

Obviously, there is no unique way to do this. Note that

$$\sum_{j=1}^q N_i^j \leq M_i \text{ and } \sum_{j=0}^q N_i^j = M_i. \quad (2.2)$$

Also $\sum_i N_i^j = N^j$ and is constant for any rearrangement that does not create or destroy
patches. ($\sum_i$ will always run over all the cells.)

For the “probability” of any total set of occupation numbers $N_i^j$, we adopt the
expression

$$W = \text{Const.} \prod_i \prod_{j=1}^q \frac{N_i^j!}{(N_i^j!) (M_i - \sum_{l=1}^q N_i^l)!}. \quad (2.3)$$

Notice that the second of the constraints (2.2) has been built in and the $N_i^j$ may be
varied independently for $j = 1, 2, 3...q$, so long as we demand that in the variation
$N_i^j \rightarrow N_i^j + \delta N_i^j$

$$\sum_i N_i^j = \text{Const.} = N^j, j = 1, 2, 3...q,$$

or that

$$\sum_i \delta N_i^j = 0. \quad (2.4)$$
2.2. Statistical mechanics in 2D turbulence

The last term in the denominator of (2.3) comes from the empty sites or “holes” in which no finite-vorticity patch resides.

Assuming that all the $N_i^j \gg 1$ and $M_i \gg 1$, we have for the entropy, using Stirling’s approximation,

$$S = \ln W \simeq \text{Const.} - \sum_i \sum_{j=1}^q (N_i^j \ln N_i^j - N_i^j) -$$

$$\sum_i\{\left(M_i - \sum_{l=1}^q N_i^l\right) \ln \left(M_i - \sum_{l=1}^q N_i^l\right)$$

$$- \left(M_i - \sum_{l=1}^q N_i^l\right)\} + \text{(negligible terms).} \quad (2.5)$$

The discretely represented energy is

$$E \equiv \frac{1}{2} \sum_i \sum_{j,k} K_{jk} N_i^j \psi_{ik} N_k^i. \quad (2.6)$$

The $j = 0$ and $l = 0$ terms will give nothing, since $K_0 = 0$. Note that $\psi_{ik} = \psi_{ki}$ is just the interaction energy of two unit-strength vortices in cells $i$ and $k$, and that $\psi_{ik}$ is species-independent. We may also write

$$E = \sum_{i,j} \frac{1}{2} K_{ij} N_i^j \psi_i, \quad (2.7)$$

where

$$\psi_i \equiv \sum_{l,k} K_{lk} N_k^l \psi_{lk}, \quad (2.8)$$

$\psi_i$ is the discretized stream function.

When $N_i^j \to N_i^j + \delta N_i^j$, $\delta E = \sum_i \sum_j K_{ji} \psi_i \delta N_i^j. \quad (2.9)$

For $\delta S$, we have

$$\delta S = - \sum_i \sum_{j=1}^q \delta N_i^j \ln N_i^j + \sum_i \sum_{j=1}^q \ln \left(M_i - \sum_{l=1}^q N_i^l\right) \delta N_i^j. \quad (2.10)$$

To maximize $S$ under the constraints (2.4) and $\delta E = 0$, using Lagrange multipliers, we set

$$\delta S + \tilde{\beta} \delta E + \sum_{i,j=1}^q \tilde{\alpha}_j \delta N_i^j = 0,$$

where the Lagrange multipliers are $\tilde{\beta}$ and $\tilde{\alpha}_j$, and the $\delta N_i^j$ are now regarded as independent, so their coefficients may be set equal to 0 for all $i$ and $j = 1, 2, 3, ... q$. We get

$$- \ln N_i^j + \tilde{\alpha}_j + \tilde{\beta} K_{ij} \psi_i + \ln \left( M_i - \sum_{l=1}^q N_i^l \right) = 0, \quad (2.11)$$
we let $\tilde{\beta} = -\beta$, $\tilde{\alpha}_j = \alpha_j$ to conform with earlier publications (they are still undetermined, anyway).

Exponentiating results in

$$\frac{N_i^j}{M_i - \sum_{l=1}^{q} N_i^l} = e^{\alpha_j - \beta K_j \psi_i}, \quad j = 1, 2, 3 \ldots q,$$

(2.12)

but $M_i - \sum_{l=1}^{q} N_i^l = N_i^0$ is equal to the number of empty sites in cell $i$, always. So

$$N_i^j = N_i^0 e^{\alpha_j - \beta K_j \psi_i},$$

(2.13)

and

$$\sum_{j=0}^{q} N_i^j = M_i = N_i^0 \sum_{j=0}^{q} e^{\alpha_j - \beta K_j \psi_i},$$

(2.14)

and so

$$N_i^j = M_i \frac{1}{\sum_{l=0}^{q} e^{\alpha_l - \beta K_l \psi_i}}, \quad j = 1, 2, 3 \ldots q.$$ 

(2.15)

The same expression will work with $j = 0$ if we take $\alpha_0 = 0$. So

$$N_i^0 = M_i \frac{1}{\sum_{l=0}^{q} e^{\alpha_l - \beta K_l \psi_i}},$$

(2.16)

and Eq. (2.15) can be extended to the case $j = 0$, also.

The equation (2.15)-(2.16) give the most probable set of occupation numbers $N_i^j$ for fixed $E$ and $N^j$. If we take the $N_i^j$ from Eqs. (2.15) and (2.16), we have

$$N^j = \sum_i N_i^j, \quad j = 1, 2, 3 \ldots q,$$

(2.17)

and

$$E = \frac{1}{2} \sum_{i,k} \sum_{j,l} K_j N_i^j \psi_{ik} K_l N_k^l,$$

(2.18)

as $q + 1$ equations which determine the $q + 1$ unknowns $\alpha_j$ and $\beta$, $j = 1, 2, 3 \ldots q$, as functions of $E$ and the $N^j$.

If we want to associate a “partial vorticity” with each species, we may write

$$\omega_i^j = \frac{M_i K_j e^{\alpha_j - \beta K_j \psi_i}}{Z_i},$$

(2.19)

$$\omega_i = \sum_j \omega_i^j,$$

(2.20)

where

$$Z_i \equiv \sum_{l=0}^{q} e^{\alpha_l - \beta K_l \psi_i}, \quad (\alpha_0 = 0).$$

(2.21)
2.2. Statistical mechanics in 2D turbulence

Now the physical vorticity associated with the most probable state is

$$\omega(x_i, y_i) = \sum_j \int_\text{cell}_i \omega_i^j dx dy, \quad (2.22)$$

where \((x_i, y_i)\) lie inside the \(i\)th cell. In this formulation, the constraints are

$$\frac{1}{2} \sum_i \omega_i \psi_i = E, \quad (2.23)$$

$$\frac{\Delta}{K_j} \sum_i \omega_i^j = N_j^i. \quad (2.24)$$

We need a tractable expression for \(S\), in order to compare different entropies for different sets of \(N_j^i\) for the same \(E\) and \(N_j^1\). We may rewrite Eq. (2.3) as

$$W = \text{Const.} \prod_i \prod_{j=0}^{q} \frac{N_j^i!}{N_j^0!}, \quad (2.25)$$

so that

$$S = \text{Const.} - \sum_i \sum_{j=0}^{q} (N_j^i \ln N_j^i - N_j^0) + \text{(negligible terms)}, \quad (2.26)$$

or

$$S = \text{Const.} - \sum_i \sum_{j=1}^{q} N_j^i (\alpha_j - \beta K_j \psi_i + \ln N_j^0)$$
$$- \sum_i N_j^0 \ln N_j^0 + \sum_i N_j^0,$$
$$= \text{Const.} - \sum_j \alpha_j N_j^1 + 2\beta E$$
$$- \sum_i (\ln N_j^0)(\sum_{j=1}^{q} N_j^i + N_j^0) + \sum_i N_j^0. \quad (2.27)$$

Note that \(\sum_{j=1}^{q} N_j^i + N_j^0 = M_i\), and \(\sum_i N_j^0\) represents the total unoccupied sites, and both of these are just fixed constants. So up to unimportant additive constants, the entropy of the stationary states is

$$S_{eq.} = - \sum_j \alpha_j N_j^j + 2\beta E - \sum_i M_i \ln N_i^0. \quad (2.28)$$

The first two terms of Eq. (2.28) are familiar from the “point” theory. The last one is new, and is a consequence of the finite area of the patches. If the “patch” area shrinks to zero, the last term becomes an infinite additive constant, apparently, and the “point” entropy is recovered.

If we assume all the \(M_i\) are equal (there seems to be no obvious reason for not doing so), the last term simplifies:

$$- \sum_i M_i \ln N_i^0 = - \sum_i (M \ln M) + M \sum_i \ln Z_i.$$
The first term is constant, and the second term has the integral representation

\[ M \sum_{i} \ln Z_i = \frac{M}{\Delta} \int \int_{\text{whole box}} \ln \sum_{j=0}^{q} e^{\alpha_j - \beta K_j \psi(x,y)} \, dx \, dy. \tag{2.29} \]

Note that \( M/\Delta \) is still entirely up to us to choose.

If we only use three values of \( K \): namely, \( K_j = -1, 0, \) or, +1, then we have

\[ S = -\alpha_+ N_+ - \alpha_- N_- + 2\beta E + M \sum_{i} \ln Z_i, \tag{2.30} \]

or in the continuous representation,

\[ S = \frac{M}{\Delta} \int \int_{(\text{box})} \left( \ln[1 + e^{\alpha_+ - \beta \psi} + e^{\alpha_- + \beta \psi}] - \alpha_+ N_+ - \alpha_- N_- + 2\beta E \right) \, dx \, dy. \tag{2.31} \]

If we now demand that \( \psi \) obeys Poisson’s equation, passing to the limit of arbitrarily many patches of arbitrarily small area,

\[ \nabla^2 \psi = -\omega = -\frac{M}{\Delta} \left[ \frac{e^{\alpha_+ - \beta \psi} - e^{\alpha_- + \beta \psi}}{1 + e^{\alpha_+ - \beta \psi} + e^{\alpha_- + \beta \psi}} \right]. \tag{2.32} \]

This would be the analogue of the “sinh-Poisson” or “tanh-Poisson” equation to be discussed later on.

It is simple, computationally, to consider initial states for which there is, for every \( \omega_i \), another level with the value \( -\omega_i \). This will in fact guarantee \( \int \int_{(\text{box})} \omega \, dx \, dy = 0 \). It also makes it plausible to assume +, - symmetry, so that \( \alpha_+ = \alpha_- \). Then we would have

\[ \nabla^2 \psi = -\omega = \frac{M}{\Delta} \left[ 2 \sinh(\beta \psi) \right]. \tag{2.33} \]

Suppose we want to let the patch area shrink to zero, i.e. \( \Delta \to 0 \), now. This would mean \( M/\Delta \to \infty \). The only way the right hand side could stay finite and have a finite flux, \( \int \int |\omega| \, dx \, dy < \infty \), would be for \( e^{-\alpha} \to \infty \), and do it in such a way that \( (M/\Delta)/e^{-\alpha} \) stayed finite:

\[ \frac{M}{\Delta e^{-\alpha}} \to \frac{\lambda^2}{2} = \text{const.} \]

So

\[ \nabla^2 \psi = \lambda^2 \sinh(\beta \psi), \tag{2.34} \]

and the sinh-Poisson equation is recovered.

It appears that for the symmetric periodic case, we can also get by with no unoccupied sites, that is \( N_0^i = 0, \forall i \). Every \( \omega_i \) could be made up of just the right number of \( K = +1 \) and \( K = -1 \) patches to give all the desired \( \omega_i \), including zero. To treat this case, we can re-start the whole procedure, with only one species independent (\( K = +1 \), say), so that \( N_i^- = M_i - N_i^+ \), always.
2.2. Statistical mechanics in 2D turbulence

Then

\[ S = \ln W \cong - \sum_i (N_i^+ \ln N_i^+ - N_i^+) \]
\[ - \sum_i [(M_i - N_i^+) \ln (M_i - N_i^+) - (M_i - N_i^+)], \]

and the analogue of Eq. (2.12) is

\[ N_i^\pm = e^{\alpha - \beta \psi_i} (M_i - N_i^\pm), \]

or

\[ N_i^+ = \frac{M_i e^{\alpha - \beta \psi_i}}{1 + e^{\alpha - \beta \psi_i}} = \frac{M_i e^{\frac{1}{2}(\alpha - \beta \psi_i)}}{e^{\frac{1}{2}(\alpha - \beta \psi_i)} + e^{-\frac{1}{2}(\alpha - \beta \psi_i)}}, \]

with

\[ N_i^- = \frac{M_i e^{-\alpha + \beta \psi_i}}{1 + e^{-\alpha + \beta \psi_i}} = \frac{M_i e^{-\frac{1}{2}(\alpha - \beta \psi_i)}}{e^{-\frac{1}{2}(\alpha - \beta \psi_i)} + e^{\frac{1}{2}(\alpha - \beta \psi_i)}}. \]

This gives an overall vorticity of

\[ \omega_i = \frac{M_i \sinh(\frac{1}{2}(\alpha - \beta \psi_i))}{\Delta \cosh(\frac{1}{2}(\alpha - \beta \psi_i))}. \quad (2.35) \]

In this case, \( E \) is invariant to letting \( \psi_i \to \psi_i + \text{const.} \), so we can always choose the constant so that \( -\beta \times \text{Const.} + \alpha = 0 \), leaving

\[ \omega_i = -\frac{M_i}{\Delta} \tanh \frac{1}{2} \beta \psi_i. \]

Since \( \beta \) is as yet undetermined, we might as well let \( \beta \to 2\beta \), \( M_i / \Delta = \lambda^2 \), and we have

\[ \nabla^2 \psi = -\omega = \lambda^2 \tanh(\beta \psi). \quad (2.36) \]

Eq. (2.34) corresponds to the limiting case in which the “patches” have no area, and Eq. (2.36) to the limiting case in which they have all the area, and there is no empty space. Both apply to the symmetric case in which for every initial \( \omega_i \), there is somewhere another with \(-\omega_i \) as its value.

The most general, completely unrestricted, case has, as its analogue of sinh-Poisson,

\[ \nabla^2 \psi = -\omega = -\sum_{j=1}^a \frac{M_j}{\Delta} K_j e^{\alpha_j - \beta \psi K_j}. \]

(2.37)

Comparing Eqs. (2.37), (2.36), (2.34), and (2.33), it seems clear that every choice of the \( M/\Delta \) and the possible \( K_j \) will lead to a different equation and a different \( \psi \). There seem to be no compelling arguments as to why one might be superior to the other in predicting long-time Navier-Stokes behaviour, and no point in speculating which will do a better job of predicting long-term Euler equation evolution, since we can never know that. It is perhaps worth remarking that if we want to shrink
\( \Delta \) so much that there is no approximation involved in letting the \( \omega_i \) of Eq. (2.1) represent an analytic function \( \omega(x,y,0) \), then the “points” are the only possibility. Every “level” approximation of the type of Eq. (2.1), with \( i \) bounded from above is already a function significantly different from \( \omega(x,y,0) \). For example, it has an infinite palinstrophy \( P = (1/2) \int \int_{\text{whole box}} |\nabla \omega|^2 \, dx \, dy \), and thus an infinite initial enstrophy decay rate, because of the sharp discontinuities between the cells.

A possible strategy is to search for initial conditions where the presence of finite-area conserved “patches” would seem to have the most important consequences: Eq. (2.33) with positive \( \alpha \). For example, in Eq. (2.33), which presumably can be solved numerically, choose an “\( \alpha \)” and a “\( \beta \)” which will give a relatively flat scatter plot (like Fig. 2.2).

![Figure 2.2: The \( \omega-\psi \) plot for a possible initial condition. The plot is rather flat around \( \omega = \pm 1 \).](image)

Pick an \( M/\Delta \) that is large enough that \( \psi \) seems slowly varying on the scale \( \sim \sqrt{\Delta} \), but \( M \gg 1 \). This will give a function \( \omega(x,y) \) which is presumably well-represented by patches, if anything can be. Do not pick the maximum entropy state but something like a quadrupole or higher multipole which can be expected to break up and generate turbulence when used as an initial condition for a Navier-Stokes code at small kinematic viscosity \( \nu \).

There may have to be a small random perturbation on those initial conditions to get it to go turbulent as soon as possible. Notice that choosing \( \alpha, \beta \) and \( M/\Delta \) in advance provides computable expressions for energy and vorticity fluxes. The first question to ask is, what “final” state does the system relax to for large times – what does its scatter plot look like? The second question is, then, using those expressions for energy and vorticity flux in the sinh-Poisson code, what does the final maximum-entropy state look like – its scatter-plot in particular? Is there any sense in which the initially flat areas of vorticity flux are still there?
2.3 Further remarks and recent developments

We should caution the reader that the word “turbulence” in this thesis will be used in a somewhat looser sense than is customary. It will not always mean a state of the fluid in which the kinetic energy is widely shared by many Fourier modes. We have concentrated on initial conditions which seemed to us most relevant to producing effects peculiar to the “patch” description: namely, initial vorticity distributions in which a few flat levels of nearly constant vorticity could be readily identified. We expected, because of instabilities in the shear flows these represent, that turbulence of a typical broad-band modal structure would soon be generated, as it is in three dimensions. We found that instabilities could indeed be activated, but often, broad-band turbulence of the conventional kind was not the result. Instead, energy spectra dominated by only a few low-lying (in Fourier space) modes were nonlinearly dynamically converted to Fourier spectra dominated by only a few (but different) low-lying modes. We have in fact found it nearly impossible to generate genuinely broad-band 2D NS turbulence through instabilities, rather than through initial conditions or externally-imposed stirring. Perhaps surprisingly, the dynamical evolution involved sometimes, but not nearly always, led to late-time quasi-steady states that seemed to correspond to most-probable “patch” predictions: most interestingly, the one-dimensional “bar” state, as will be seen. Broad-band, initially-excited turbulence continued to lead to the classical dipolar late-time state, as it has in previous turbulence simulations.

Extension and elaboration of the statistical mechanics of the Euler equations has continued and has led to a variety of predictions which are not being tested here [58–65]. Consideration of these mathematically ambitious theories lies outside the scope of this thesis. Our interest here is focused on what happens as a consequence of Navier-Stokes, not ideal, dynamics, and involves a comparison of the predictions of (historically) the first two Euler entropy maximizations with a Navier-Stokes decay. As far as we are concerned, it is not yet understood why a Navier-Stokes decay, which involves true, well-resolved dissipation by viscosity should lead to anything predicted by any 2D Euler-equation statistical mechanics; no arguments to date that “coarse-graining” should in any accurate way mimic viscous dissipation seem persuasive to us. In any case, no coarse-graining is involved in the computation, which is well-resolved. There are not likely to be accurate continuum Euler-equation solutions over several hundred eddy turnover times to compare with in the near future. This is not to say that the predictions of the recently-presented theories mentioned above [58–65] might not eventually be shown to have an even better predictive capacity for viscous turbulent decays than these two. Sharpening up what predictions might be extracted from them that could be compared with a Navier-Stokes decay computation appears as a desirable activity, and actually making the comparisons appears as a demanding one.

Despite the general developments made during the last ten to fifteen years in this field, and the more detailed new approaches reported recently, a few critical comments should also be made. The study of relative simply models for the 2D Euler equation, such as the “point” or “patch” approach, is a prerequisite when applying such models to investigate the late-time states of ideal Euler flows or late-time 2D Navier-Stokes
turbulent decays. Any more complicated theory with more parameters, usually including the former theory as a special case, reduces the transparency and efficiency to apply the method to realistic high-Reynolds number flows. For example, the present study to distinguish between the “point” and one special case of the “patch” formulation for 2D Navier-Stokes turbulence required already two person-years to set up and to execute the investigation. This observation hints at the virtual impossibility to validate more complicated theoretical models. Moreover, one should keep in mind that the increased number of parameters in these new theoretical models, which obviously decreases the transparency, should be related with the data from experiments (if possible at all) and numerical simulations. In my view, further elaboration in this direction should be avoided.
Chapter 3

Predicting 2D decaying turbulence

3.1 Introduction

We address ourselves to the long-time dynamics in two dimensions (2D) of the Navier-Stokes equation in the usual vorticity representation,

\[ \frac{\partial \omega}{\partial t} + \mathbf{v} \cdot \nabla \omega = \nu \nabla^2 \omega, \]  

(3.1)

where the fluid velocity \( \mathbf{v} \) is to be written in terms of the stream function \( \psi \) as \( \mathbf{v} = \nabla \times (\hat{e} \psi) \), and \( \hat{e} \) is a unit vector in the \( z \) direction. The velocity \( \mathbf{v} \) has only \( x \) and \( y \) components, which depend only upon \( x \), \( y \), and the time \( t \). In the natural dimensionless units of the problem, the kinematic viscosity \( \nu \) may be interpreted as the reciprocal of the Reynolds number, which we will specify in more detail presently. The curl of \( \mathbf{v} \) is the vorticity \( \omega = (0, 0, \omega) \), which has only one component, also in the \( z \)-direction. The stream function \( \psi \) and the non-vanishing component of vorticity \( \omega \) are related by Poisson’s equation,

\[ \nabla^2 \psi = -\omega. \]  

(3.2)

We note that dropping the final viscous term in Eq. (3.1) leaves us with the 2D equations of an ideal Euler fluid. We note moreover that a time-independent solution of these Euler equations results any time \( \omega \) is a differentiable function of \( \psi \).

We will work in a periodic box with sides \( 2\pi \) in length, and will use the unit length 1 to define the Reynolds number; thus our basic unit of length is roughly \( 1/6 \) of a box dimension. The unit of velocity will typically be a root-mean-square value of the initial velocity, and we will attempt to make this equal to 1 whenever possible. Thus \( \nu \) in Eq. (3.1) may reasonably be identified with the reciprocal of an initial Reynolds number, \( \text{Re} \). We are interested in values of \( \text{Re} \) of at least several thousand, so that the final term in Eq. (3.1) is formally very small, except in regions of steep vorticity gradients. The “eddy turnover time,” which we shall use as a unit of \( t \), is thus about \( 1/6 \) of a box dimension divided by an initial rms velocity. It is well known that under such circumstances, the kinetic energy \( E \), defined by
Chapter 3. Predicting 2D decaying turbulence

\[ E = \frac{1}{2} \left( \frac{1}{(2\pi)^2} \right) \int \int v^2 \, dxdy \]  

(3.3)

decays proportionally to \( \nu \) or slower (see section 1.2). However, the decays of higher-order Euler-equation invariants constructed as integral moments of \( \omega \), such as the enstrophy,

\[ \Omega = \frac{1}{2} \left( \frac{1}{(2\pi)^2} \right) \int \int \omega^2 \, dxdy, \]  

(3.4)

appear to continue to decay at an \( O(1) \) rate; even a weak dependence of their decay rate upon \( \nu \) has not been convincing demonstrated. Note that both ideal invariants are referred to unit volume, in the dimensionless units.

When high Reynolds numbers exist, then the turbulent decay of energy is seen to be very slow, relative to any other identifiable ideal Euler invariant. When the energy decay time is large enough to be well-separated from the eddy turnover time, it is possible to observe in numerical computations [3–5] that a quasi-steady state is reached in a time over which the fractional decay of energy is small (a few hundred eddy turnover times). One might expect to find a “selectively decayed” state, in which the enstrophy to energy ratio is minimal \([66,67]\), and indeed, if one waits long enough, it can be analytically proved that such a state must be approached \([68]\). However, it was found \([5]\) that long before that time, the quasi-steady, slowly-decaying state that is reached has a rather sharp one-to-one pointwise correspondence between \( \omega \) and \( \psi \).

(As we have seen in Figs. 1.3, this is not always true.) The elucidation and testing of this correspondence is the principal purpose of this thesis.

Clearly, even though a slow viscous decay may be superimposed on such a state with \( \omega \) approximately a function of \( \psi \), the state is also closely approximated by a time-independent solution of the Euler equations. There is no a priori reason why this should be so. It is a fact we attempt to incorporate here into a coordinated set of direct numerical simulations of Eq. (3.1)-(3.2) and a combined analytical and numerical argument, based upon statistical mechanics, in pursuit of what the connection between \( \omega \) and \( \psi \) should be.

The statistical mechanics depends upon a discretization of vorticity in terms of delta-function line vortices (“points”) or in terms of mutually-exclusive, finite-area “patches.” Boltzmann statistics are applied to the former, to define an entropy, or logarithm of the probability of a state, and Lynden-Bell statistics are applied to the latter. The limits of zero-area, finite-vorticity patches are points. Thus the patch formulation can be viewed as containing the point version of the theory, and either is or is not a useful generalization of it. In both cases, what results is a “most probable” dependence of vorticity upon stream function. A mean-field limit (infinitely-many points or patches, of arbitrarily small strength) is then taken, to yield continuous and differentiable functions \( \omega(\psi) \). It can be shown (see chapter 2) that for points, the resulting function is:

\[ \nabla^2 \psi = -\omega = -e^{\alpha+\beta\psi} + e^{\alpha-\beta\psi}. \]  

(3.5)
3.1. Introduction

For patches, the resulting function is:

\[
\nabla^2 \psi = -\omega = - \sum_{j=1}^{q} \frac{M}{K_j} \frac{e^{\alpha_j \beta \psi K_j}}{\sum_{i=0}^{q} e^{\alpha_i \beta \psi K_i}}. 
\]

(3.6)

The symbols \( \alpha, \beta \) are Lagrange multipliers, which enter via a maximization of the appropriate entropy subject to the constraints of given energy and positive and negative fluxes of vorticity. They are determined in principle by demanding that the energy and vorticity fluxes calculated on the basis of Eqs. (3.5) or (3.6) match specified values. In Eq. (3.6), the \( K_j \) are the “levels” of vorticity corresponding to the different sized patches, and must be chosen somewhat arbitrarily. \( \Delta / M \) is a fixed size of a “patch,” and may be chosen arbitrarily, within wide limits.

Referring to the next section for a numerical method to solve Eqs. (3.5) and (3.6), we proceed in Chapter 5 to a description of the initial conditions used. Both Eqs. (3.5) and (3.6) have infinitely many solutions. The physical ones are interpreted to be those which maximize the appropriate entropy from which they were derived. The others represent local maxima, which, as we shall see, may in some cases represent attainable states in the computations.

The questions before us are:

- How does one determine the entropy-maximizing time-independent solutions to Eqs. (3.5) and (3.6)?

- How close may we come to those solutions in a dynamical computation that solves Eqs. (3.1) and (3.2) as a time-dependent initial-value problem with large but finite Reynolds number?

- Are there noticeable differences between the predictions of Eqs. (3.5) and (3.6) and are there reasons for preferring one to the other as predictors of late-time turbulent decays?

It is to be stressed that all three questions are answerable not in the abstract, but only as a result of somewhat demanding computations. There is no a priori reason why Navier-Stokes turbulent decays should be predicted at all by anything having to do with the Euler equations. The latter will never be soluble, for continuous initial conditions, over time intervals long enough to make ideal solutions of much interest, because it is in the nature of Euler codes, having no minimum physically-determined length scale, to overrun their own resolution in a very few eddy turnover times. If the predicted states had not shown some empirical relevance to Navier-Stokes solutions, there would be no justification for this activity. Experience with 2D turbulence computations shows that even though it is known that no singularity occurs for the continuum Euler equations within a finite time, the transfer of excitations to smaller spatial scales proceeds quite rapidly: typically an order of magnitude in wave number per eddy turnover time, or faster. Thus ideal phenomena that require hundreds of eddy turnover times to see (as these do) will not be feasible to compute with continuum Euler codes within the foreseeable future.
Chapter 3. Predicting 2D decaying turbulence

The dynamical code used here is of the now familiar Orszag-Patterson pseudospectral variety [69, 70]. The code is a parallelized MPI Fortran 90 version of an earlier Fortran 77 code provided by W.H. Matthaeus (private communication). It is fully de-aliased, using the shifted-grid method. It has been run, in the runs reported here, on the SGI Origin 3800 at SARA Supercomputing Centre in Amsterdam. Some of the simulations (resulting in the "bar" final state in Chapter 5.2.2) have also been recomputed with a different pseudospectral Fourier code (provided by Dr. A.H. Nielsen, Risø National Laboratories, Denmark). These computations yielded the same conclusions.

All runs resolve the Kolmogorov dissipation wave number based on enstrophy dissipation. Previously, initial conditions for turbulent decay runs have tended to use randomly loaded Fourier coefficients in the spectra of vorticity fields up to some upper wave number. These have typically been chosen to match some cascade wavenumber $k$-spectrum, such as the $-5/3$ direct enstrophy cascade spectrum predicted by Kraichnan [71]. The phases have been chosen from a random number generator. This has been done to achieve the most disordered (in some sense) initial field compatible with a particular power spectrum. The spectra contain no more useful information for this problem, it should be noted, than the phases of the Fourier coefficients; for this reason, we often give greater emphasis to spatial contour plots of vorticity and stream function, which incorporate the phase information as well as the amplitudes. There is relatively little emphasis on cascade-related considerations, such as power-law wavenumber spectra, since these require external stirring, or continued injection of excitations in $k$-space, and we say nothing about this situation here. Several of the runs to be reported later are therefore presented in the form of contour plots of vorticity and stream function only. The initialization procedure described above can only yield a vorticity distribution that is analytic in $x$ and $y$, since only sinusoidal functions are involved, even though the spatial dependence may be wildly fluctuating. Such a vorticity distribution can take on any particular value of $\omega$ only over a set of measure zero. For this reason, one might imagine it would tend to de-emphasize features that corresponded to the "levels" in the patch formulation, over which the vorticity is supposed to take on a constant value inside a compact area. Therefore, we have also stressed initial conditions that have large, flat areas of vorticity, separated by thin regions, with as steep spatial gradients between them as the code will resolve. Our original intention was that these would correspond to unstable laminar shear flows whose instabilities would subsequently generate turbulence. Somewhat to our surprise, we have found it easy to produce shear-flow instabilities but not ones that led to turbulence, in the sense of broad $k$-spectra. We have come to suspect that this is a feature inherent to two-dimensional flows, and perusal of the literature has uncovered only three-dimensional turbulence as a consequence of unstable shear flows, but not two-dimensional; but this must be left as a conjecture rather than a demonstrated fact.

In any case, in order to induce a large number of Fourier modes to participate in the subsequent dynamics, we have found it necessary, in the runs evolving from vorticity distributions with flat areas, to add significant amounts of random noise initially, in order to get a subsequent $k$-spectrum that could readily be called "turbulent." Even then, some of the evolution we find might have its status as turbulence disputed.
3.2 Solving the nonlinear Poisson equation

3.2.1 Introduction

In this section, we will present an outline how to get the solutions of those Poisson-like equations (Eqs. (3.5) and (3.6)). The goal is not only to find numerical solutions but to determine which among the solutions has the highest entropy for given energy and vorticity fluxes. Generally speaking, the entropies of states with more maxima and minima have been seen to be less than those with fewer, and we can be guided by this. But there are instances where solutions of two different topologies can be found whose entropies lie very close to each other, as found by Pointin and Lundgren [35], and we must concentrate on those.

![Graphs](image)

Figure 3.1: Typical $\omega - \psi$ relation from (a) sinh-Poisson, (b) 3-level Poisson equation and (c) tanh-Poisson equation (special case when $e^{-\alpha} \to 0$ in Eq. (3.11)).

We consider the point relation, Eq. (3.5), in the absence of any conditions that would suggest asymmetry between positives and negatives, so that the two Lagrange multipliers $\alpha_+ = \alpha_- = \alpha$. The result is the sinh-Poisson equation (a typical $\omega - \psi$ dependence of which is shown in Fig. 3.1(a)),

$$\nabla^2 \psi = 2e^{\alpha} \sinh(\beta \psi), \quad (3.7)$$

where $\beta$ is a negative [31] constant. Eq. (3.7) can be rewritten as

$$\nabla^2 \Psi = -\lambda^2 \sinh \Psi, \quad (3.8)$$

where we have defined

$$\Psi = |\beta| \psi \quad (3.9)$$
and

$$\chi^2 = 2 |\beta| e^\alpha. \quad (3.10)$$

We also consider the patch relation, Eq. (3.6), specialized to the three-level case of vorticity levels $-1$, $0$, and $+1$. Again assuming symmetry between positives and negatives (Fig. 3.1(b)), we have

$$\nabla^2 \psi = \frac{M}{\Delta} \cdot 1 \cdot \left[ \frac{2 \sinh(\beta \psi)}{e^{-\alpha} + 2 \cosh(\beta \psi)} \right] \quad (3.11)$$

where $\beta < 0$ and $\Delta/M$ is the (arbitrary) size of a patch. Eq. (3.11) can be simplified as

$$\nabla^2 \Psi = \frac{\chi^2 \sinh \Psi}{g + \cosh \Psi}, \quad (3.12)$$

by letting:

$$\Psi = |\beta| \psi, \quad (3.13)$$

$$g = \frac{1}{2} e^{-\alpha}, \quad (3.14)$$

and

$$\chi^2 = \frac{M |\beta|}{\Delta} \quad (3.15)$$

![Contours of constant stream function $\psi$ for the solutions of Eq. (3.5). Negative values are shown as broken lines, throughout.](image)

It is useful to introduce terms for the solutions of differing topology. For example, for a given energy and vorticity flux, the contours of stream function may look like Figs. 3.2. The structure shown in Fig. 3.2(a) will be called the “dipole” solution, and the one shown in Fig. 3.2(b) will be called the “quadrupole” solution, and so on through higher “multipoles” that correspond to successively higher (even) numbers.
of maxima and minima. We have also found one-dimensional solutions as illustrated in Fig. 3.2(c). These one-dimensional solutions will be called “bar” solutions, and there are also an infinite sequence of them, with basic periodicities $2\pi$, $\pi$, $\pi/2$, ... These states are obtained, as explained later in detail, by iterating a trial solution with similar topology until convergence is obtained.

![Figure 3.3: A monopolar solution (a) of the Poisson equation $\nabla^2 \Psi = -\lambda^2 \sinh \Psi$ (on the domain $[0, 2\pi] \times [0, 2\pi]$) with $\Psi = 0$ at the boundary, may be used to generate a quadrupolar one (b) and $N^2$-pole (c).](image)

It is possible to construct higher-order multipole solutions by putting monopole solutions side by side with alternating signs, and requiring doubly-periodic boundary conditions, as illustrated in Figs. 3.3.

Suppose we have a monopole solution of the equation $\nabla^2 \Psi = -\lambda^2 \sinh \Psi$ on the domain $[0, 2\pi] \times [0, 2\pi]$ (with $\Psi = 0$ at the domain boundary). This solution can be used to generate a quadrupolar one (see Fig. 3.3(b)) and a so-called $N^2$-pole as shown in Fig. 3.3(c) (we show a 3$^2$-pole here), and a “checkerboard” solution is obtained. The size of the monopole components should be reduced in order to fit the multipole solution on the $[0, 2\pi] \times [0, 2\pi]$ domain. Note that the $N^2$-pole is therefore not a solution of $\nabla^2 \Psi = -\lambda^2 \sinh \Psi$, but of the modified Poisson equation $\nabla^2 \Psi = -(\lambda/N)^2 \sinh \Psi$. The “checkerboard” solution is also used to construct the dipole and higher multipole solutions as indicated in Figs. 3.4. The dipole is then a solution of the Poisson equation $\nabla^2 \Psi = -(\lambda^2/2) \sinh \Psi$. Obviously, the process shown in Figs. 3.3 and 3.4 is reversible – it is also possible to construct the monopole or the quadrupole solutions from the dipole. As a consequence we can start to construct the sequence of solutions with the one we can compute most easily.

However, some loss of accuracy might be expected by using this procedure, but in our calculations we made sure that the accuracy of all solutions is of the order of $10^{-6}$ (based on monitoring an RMS error during the iteration as outlined in the following Section). In a similar way it is possible to obtain a hierarchy of “bar” solutions (see Figs. 3.5) in a domain with doubly-periodic boundary conditions starting with the
Figure 3.4: A periodic quadrupolar solution of \( \nabla^2 \Psi = -\lambda^2 \sinh \Psi \) (on the domain \([0, 2\pi] \times [0, 2\pi]\)) may be used to generate a dipolar one, depending upon the spatial sub-volume chosen, as shown here. The "dipole" solution which is cut from the "quadrupole" can be zoomed (by a factor of \( \sqrt{2} \) in \((x, y)\) space) to fit the \([0, 2\pi] \times [0, 2\pi]\) domain.

Figure 3.5: The serial solutions of the "bar" solutions in the doubly-periodic domain. Note that (c) is the lowest mode if we want the direction of the bar to be in the 45° direction to the horizontal line. Like the quadrupolar solutions, (c) will have a lower entropy line than the "bar" and the "dipole" solutions, as it will be seen later in this chapter.
3.2. Solving the nonlinear Poisson equation

The basic solution shown in Fig. 3.5(a). The procedure described in this section can also be applied to other Poisson-like equations introduced in this thesis.

3.2.2 The solution of the sinh-Poisson equation

![Graph showing the relationship between error and iteration steps.](image)

Figure 3.6: The plot of RMS error vs. iteration steps. Only the line in the middle will lead to the correct solutions.

The sinh-Poisson equation (3.8) has been solved by several researchers numerically and analytically (see Ref. [47] and references therein). Probably the most direct way to do it is to put the equation into spectral space and perform the iteration there:

\[
(\tilde{\psi}_{n+1})_k = \frac{\chi^2}{|k|^2} (\sinh \tilde{\psi}_n)_k.
\]  

(3.16)

Several numerical methods exist to solve Eq. (3.16), but we have chosen to use Fourier expansions to solve the sinh-Poisson equation. The primary reason is the high efficiency of the Fast Fourier Transform (FFT), which is much faster than other numerical methods when we want to obtain more accurate solutions by using higher resolutions. Moreover, by using Fourier expansion of the solution to Eq. (3.16) the doubly-periodic boundary conditions are satisfied automatically, thus we do not have to worry about the boundary conditions at all.

We define the root-mean-square error (RMS) by:

\[
e = \left( \sum_{i,j} f'^2 / \sum_{i,j} g'^2 \right)^{1/2}.
\]  

(3.17)
where
\[ f = \nabla^2 \Psi(i, j) + \lambda^2 \sinh(\Psi(i, j)) \]  
(3.18)
and
\[ g = \left| \nabla^2 \Psi(i, j) \right| + \left| \lambda^2 \sinh(\Psi(i, j)) \right|. \]  
(3.19)
The following procedure is used in the evaluation of the solution of the sinh-Poisson equation: we choose an initial condition based on an educated guess, subsequently we compute the solution of (3.16) and the RMS error (3.17) is tracked as the iteration continues. Suppose the solution of \( \Psi \) should converge to the final solution with RMS error \( \epsilon_0 \). Three scenarios for the evolution of \( \epsilon \) should be considered (see Fig. 3.6). Two of them can be discarded immediately: the one showing no convergence, i.e. \( \epsilon \) is growing without bound, and the trivial solution \( \Psi \to 0 \). The third one is the solution we are looking for (with convergence to it with RMS error \( \epsilon_0 \)). However, two aspects of this solution should be kept in mind:

- We have to choose the value of \( \lambda^2 \) such that the value of \( \epsilon_0 \) is sufficiently small to enable rapid convergence.
- When the RMS error \( \epsilon \) becomes small enough (say \( \epsilon \leq 10^{-6} \)), the iteration process should be terminated because further iteration will show a growing RMS error again.

We use numerical methods instead of analytical solutions to solve the sinh-Poisson equation because it is easier to control the amplitude of the solution, which is closely related to the energy of this solution. Numerically, we can rather easily control the energy of the solution, so the final plot of the entropy-energy diagram can be drawn on the basis of more or less equally spaced data (see Figs. 3.7 and 3.8).

We have shown three solutions of the sinh-Poisson equation in Fig. 3.2. In addition to the solutions exhibited, there are an infinite number more, characterized by more and more maxima and minima. They are obtained by starting with a trial function on the right hand side of Eq. (3.16) that has the desired number of final maxima and minima, as noted by several authors (see, for example, McDonald [33], Book et al. [34], Lundgren and Pointin [35]). Typically, the solutions with many maxima and minima are lower entropy solutions, for fixed energy and vorticity fluxes, and so we do not discuss them in any detail.

### 3.2.3 The solution of the three-level Poisson equation

It is well known that in particular smooth solutions of partial differential equations are described rather easily with spectral methods, but severe problems are expected to arise when a smooth solution is not guaranteed. In that case the spectral method loses its advantage over the classical lower order methods when trying to solve the three-level Poisson equation (3.12), an equation which might have some step function solutions. For these three-level Poisson equations no analytical solutions are available either. In the meanwhile, applying the numerical method used by McDonald [33],
3.2. Solving the nonlinear Poisson equation

which is essentially a finite difference method with its inherent disadvantages for the present problem, can yield some reasonable results.

The first step of McDonald’s procedure is to get the nontrivial solution for Eq. (3.12) with \( \Psi = 0 \) on a square boundary of \([0, 2\pi] \times [0, 2\pi]\). Starting from a trial solution \( W(x, y) \), we get the solution of \( v(x, y) \) from

\[
\nabla^2 v + v f'(W) = R(W, W),
\]
subject to \( v = 0 \) on the boundary. Here

\[
(A, B) = \int_0^\pi \int_0^\pi A(x, y)B(x, y)dxdy,
\]

\[
f(\Psi) = \frac{\lambda^2 \sinh \Psi}{g + \cosh \Psi},
\]

\[
R(x, y) = \frac{\nabla^2 W + f(W)}{(W, W)}.
\]

Then we can correct the trial solution:

\[
W \rightarrow W + \frac{v(x, y)(W, W)}{2(v, W) - (W, W)}. \tag{3.24}
\]

until we get a sufficiently accurate solution.

McDonald [33] gave a direct solution of the linearized equation (3.20). This method (see details in Appendix A) is conceptually simple, but the roundoff error propagation restricts its usage into the low resolution regime only. Even for a grid of \( 15 \times 15 \) the roundoff error was found to become as large as \( 10^{-5} \) despite the fact that we were performing the computations with double precision. So perhaps the most important change we made to McDonald’s methods is using the quadruple precision in the calculation (and this is not a luxury on modern computers). The accuracy problem in this method is overcome. As a result, our solutions can at least be obtained with an accuracy of \( 10^{-7} \) on a grid of \( 33 \times 33 \). For the iteration procedure the plot of the RMS error vs. iteration steps shown in Fig. 3.6 still applies to this case.

The original code that we have written, based on the McDonald approach, appeared to have several important drawbacks. The most severe one is that the computation needs to be interfered with from time to time, in order to change the value of the \( \lambda^2 \) to get a solution which is sufficiently accurate. Moreover, the computation needs to be halted artificially when sufficient accuracy has been achieved. A few of the other problems are the following: To obtain solutions with high accuracy we need a substantially higher resolution and, since we cannot use FFTs any more, longer computation times are needed per iteration step. The quadruple precision replaces the double precision in our calculation, which makes the code even more time-consuming. As it will be shown on the next section of this chapter, to make the entropy-energy plot of “patch” theory need to solve the Poisson-like equation much more frequently than in the case of the “point” theory. Obviously, there is some need to construct a “smart” code when we want to solve Eq. (3.12). Here we mean by “smart” that the code can
find the final solution after limited times of human interference only, and stop in time without interference. Appendix B elucidates the algorithm of the modified code. An important feature in the newly devised code consists of the computation of solutions which satisfy the boundary condition $\Psi = 0$ only. To get a doubly-periodic solutions, we have to rely on the procedures as already illustrated in Figs. 3.3, 3.4 and 3.5.

3.3 The energy-entropy (E-S) plots

3.3.1 Solution strategy for the E-S plot in the “point” theory

As a first step to compute the data set necessary for the calculation of the E-S plot in the “point” theory, we need to solve the sinh-Poisson equation (3.8) numerically. This can be carried out rather straightforwardly. Once the sinh-Poisson equation has been solved, for a given value of $\lambda$, we must keep in mind that in order to evaluate the entropy associated with the solution, we must revert to the parameters of Eq. (3.7), while guaranteeing that $\alpha$ and $\beta$ satisfy Eq. (3.10). Since for each fixed value of $\lambda^2$, we have infinite sets of possibilities for $\alpha$ and $\beta$, a recipe is needed for choosing them. Since our goal is to plot the entropy vs. energy for a fixed value of the vorticity flux,

$$\Omega_\psi = \frac{1}{L^2} \int \int e^{\alpha + \beta \psi} dx dy = \frac{1}{L^2} \int \int e^{-\beta \psi} dx dy$$

for the domain $[0, L] \times [0, L]$ (here we let $L = 2\pi$ and $\Omega_\psi = 1$, for convenience), we must also be assured that the solution we get for $\psi$ satisfies this condition.

Fortunately, these two conditions can be satisfied simultaneously. Combining Eqs. (3.9), (3.10) and the condition $\Omega_\psi = 1$ leads to the conclusion that $|\beta| = \Omega_\psi$:

$$\Omega_\psi = \frac{1}{L^2} \int \int \frac{\lambda^2}{2} e^{-\psi} dx dy$$

$$= \frac{1}{L^2} \int \int |\beta| e^{\alpha - |\psi|} dx dy = |\beta|.$$

(3.25)

Because Eq. (3.8) is the equation solved, the value of $\Omega_\psi$ is readily obtained. Thus we obtain $\beta$ and from Eq. (3.10) the numerical value for the remaining parameter, $\alpha$, can be identified. We have now all the parameters that correspond to fixed vorticity flux of either sign. From Eqs. (3.26) and (3.27),

$$E_\psi = \frac{1}{2L^2} \int \int \psi \omega_\psi dx dy = \frac{1}{2L^2 \beta^2} \int \int \Psi \omega_\psi dx dy$$

(3.26)

$$S_\psi = -2\alpha \Omega_\psi - 2 |\beta| E_\psi = -2\alpha - 2 |\beta| E_\psi,$$

(3.27)

we can draw plots of the entropy ($S_\psi$) vs. energy ($E_\psi$) for fixed unit flux of positive and negative vorticity.
3.3.2 Solution strategy for the E-S plot in the “patch” theory

Solving the three-level Poisson Eq. (3.12) is not as easy as solving the sinh-Poisson equation because there is an extra parameter, $g$, which implicitly reflects the arbitrary choice of the patch size. Another strategy is required:

1. We choose the size of the patch, so that

$$\frac{M}{\Delta} = \frac{\chi^2}{|\beta|} = \text{Const.}; \quad (3.28)$$

2. Then a trial value of $g$ is chosen before solving Eq. (3.12), again iterating about a trial solution of desired topology. Using Eq. (3.28), we have the value of $\beta$, and $\alpha$ from Eq. (3.14), $E_\psi$ and $\Omega_\psi$ are given by

$$E_\psi = \frac{1}{2L^2/\beta^2} \int \int \omega_\psi \Psi dx dy, \quad (3.29)$$

$$\Omega_\psi = \frac{M}{L^2 |\beta|} \int \int \frac{e^\Psi}{2g + 2\cosh \Psi} dx dy; \quad (3.30)$$

3. However, the parameters obtained in this way are not usable since the condition of unit positive and negative vorticity flux is not in general satisfied. We must return to the second step and change $g$ until the desired accuracy is obtained from Eq. (3.30) with $\Omega_\psi = 1$. (This would be an extremely time-consuming procedure if the “smart” code were not used; it is described in some detail in Appendix B.) Once this has been done, we are then in a position to evaluate the entropy from the algebraic expression:

$$S_\psi = -2\alpha\Omega_\psi - 2|\beta| E_\psi + \chi^2 \int \int \ln(1 + e^{\alpha-\Psi} + e^{\alpha+\Psi}) dx dy. \quad (3.31)$$

3.4 Prediction of the late-time decay of 2D turbulence

Fig. 3.7 shows a plot of the entropy vs. energy, for fixed unit positive and negative vorticity flux, for the quadrupole, bar, and dipole solutions computed for the “point” discretization. Observe the very small difference that makes the entropy of the dipole greater than that of the bar. It will turn out that it is possible to find either solution as a consequence of the development of certain initial conditions, as we shall see later.

We may plot the entropy of Eq. (3.31) in Fig. 3.4, which is obtained by the choice of $M/\Delta = 3.7814$, a relatively large “patch” size. It will be seen that for this large
Figure 3.7: Entropy vs. energy for the solutions of Eq. (3.5) for unit positive and negative vorticity flux, computed for the “point” discretization.

patch size, the entropy of the bar solution is greater than that of the dipole, a different conclusion than that of the point calculation. However, if we reduce the size of the patch, we find that the result approaches that of the point formulation. In Fig. 3.4, we see the result when \( M/\Delta = 25 \), and in Fig. 3.4 the result when \( M/\Delta = 100 \), where the maximum entropy state is again the dipole one. Considering the very small entropy differences between the dipole and the bar solutions shown in Figs. 3.7 and 3.8, it might not be thought surprising if the fluid had difficulty making up its mind which state to relax into. It may appear to the reader that the shift of the bar state to its status as “most probable,” solely as a consequence of enlarging the patch size seems artificial (Figs. 3.8), since there seems to be no unique physical determination for the patch size. It seems somewhat artificial to us also, but the fact that the dynamical code seeks out the bar solution for some quite dissimilar initial conditions, as will be seen in what follows, convinces us that the bar solution has some reality.

Before turning to the results of the dynamical computations, we offer a few observations on the relations of the patch versions of the theory to the point version, and to each other. It is clear from the derivation that in the limit that the patch sizes become smaller and smaller, at fixed and finite separation, the point version of the theory is recovered. Depending upon the number of levels chosen for the patch formulation, there are many versions of the most-probable-state patch equation; the three-level version of Eq. (3.11) is not the most general by any means. Each will have different solutions. There is, however, no apparent unique or practical prescription for how many levels, or what size patches, should be used to represent a particular initial analytic vorticity
Figure 3.8: Entropy vs. energy at unit vorticity fluxes for the solutions of Eq. (3.11): (a) with a relatively large patch size ($M/\Delta = 3.7814$); (b) with a somewhat smaller patch size ($M/\Delta = 25$); (c) with a still smaller patch size ($M/\Delta = 100$). Note that in (c) the dipole solution has become slightly more probable than the bar.
distribution with high accuracy. It is not clear that it can be done without requiring the patch size to shrink to zero, at which point it becomes indistinguishable from a point representation. In fairness, we should also say that there is another aspect to the point formulation that is also ambiguous: namely, there is no reason to choose the point vortices of the mean-field theory to be of equal strength. If some are of different strength, Eq. (3.7) will also change. Keeping these in mind, we will turn to the effort to see some of the solutions as consequences of direct numerical solution of the Navier-Stokes equation.

3.5 Conclusion

In the first part of this chapter, we have introduced numerical solution procedures for two kinds of Poisson equations. We found that for the sinh-Poisson equation, it is easier and more efficient to use spectral methods (by applying Fourier decomposition). However, for the three-level Poisson equation, it appeared to be easier to use finite difference methods. To get the nontrivial solution for the three-level Poisson equation, we have to change the initial condition wisely satisfying the condition $\Psi = 0$ (instead of periodic boundary conditions), and it is convenient to have a “smart” code to do so (appendix B). After finding the converged solution, the periodic solution is extracted from it.

In the last part of this chapter, we have introduced two different kinds of strategies to make the $E - S$ plot for the “point” and the “patch” theory, respectively. We have shown that the “point” theory and the “patch” theory yield two qualitatively different kinds of predictions. This gives us the motivation to perform numerical simulations of 2D decaying Navier-Stokes turbulence to test the applicability of these theoretical results.

The numerical simulations of 2D decaying Navier-Stokes turbulence concern very-long time simulations at high resolution. Therefore, before presenting and discussing the results of the numerical simulations, we will first focus on the issue of parallelization of the 2D Fourier pseudospectral code.
Chapter 4

Numerical simulation of turbulence

4.1 Direct numerical simulation (DNS) with pseudospectral methods

To test the applicability of the mean-field statistical predictions, it is necessary to obtain long-time numerical solutions for 2D NS turbulent flows. The emphasis has been on trying to identify initial conditions for which “point” and “patch” predictions differ as greatly as possible. The numerical solutions required put strong demands on computational capabilities. Unusually long computation times (for a simulation over several hundred eddy turnover times) are required on single-cpu workstations, and mandate the use of parallel computing. As an example, we may consider an influential computation made by Matthaeus et al. [3–5] over a decade ago on the Cray YMP: a run which required three years to complete. Though considerable increases in the speed of computers have been made in the meantime, this run still requires three weeks of continuous computation on a single-cpu workstation (Compaq Alphastation XP 1000). Similar runs needed to produce a “bar” final state require up to three months on such a single-cpu workstation.

An efficient parallel scheme is proposed for performing DNS of two-dimensional Navier-Stokes turbulence at high Reynolds numbers. As will be shown, the computation of runs producing the “bar” final state referred to in the preceding paragraph can be reduced to two weeks on an SGI Origin 3800 using 6 CPUs, by parallelization. That is what makes the present series of computations feasible.

The two-dimensional incompressible Navier-Stokes equations can be written as

\[
\frac{\partial \omega}{\partial t} + \mathbf{v} \cdot \nabla \omega = \nu \nabla^2 \omega ,
\]

\[
\nabla^2 \psi = -\omega ,
\]

where \( \mathbf{v} = (u, v) \) is the velocity field defined with respect to a Cartesian coordinate system \((x, y)\), \( \omega = \partial v / \partial x - \partial u / \partial y \) is the (scalar) vorticity field, \( \psi \) is the stream function \((u = \partial \psi / \partial y \text{ and } v = -\partial \psi / \partial x)\) and \( \nu \) represents the kinematic viscosity. We consider a square domain \( \mathcal{D} = [0, 2\pi] \times [0, 2\pi] \) with doubly-periodic boundary conditions.
\[ \frac{\partial \omega}{\partial t} + \mathbf{v} \cdot \nabla \omega = \nu \Delta \omega \]

Figure 4.1: A schematic sketch of the evaluation of J (with the de-aliasing procedure). Each solid arrow corresponds to an FFT.

We use pseudospectral methods to solve this equation by expanding \( u, v \) and \( \omega \) in a truncated Fourier series. The actual time integration is carried out by evaluating the Fourier expansion coefficients in course of time, and we only go back to physical space when we compute the nonlinear term \( J = \mathbf{v} \cdot \nabla \omega \) of the vorticity equation. We discretize the advection term \( J \) with a 2\textsuperscript{nd} order Adams-Bashforth scheme, and the dissipative term \( \nu \nabla^2 \omega \) according to the Crank-Nicolson scheme. The application of this semi-implicit scheme, generally referred to as ABCN, to eq. (4.1) in spectral space gives

\[ \hat{\omega}_k^{t+1} - \hat{\omega}_k^t = -\Delta t \left( \frac{3}{2} \hat{J}_k^t - \frac{1}{2} \hat{J}_k^{t-1} \right) - \nu \Delta t \frac{k^2}{2} \left( \hat{\omega}_k^{t+1} + \hat{\omega}_k^t \right) , \quad (4.3) \]

with \( \mathbf{k} = (k_x, k_y) \in [-N/2, ..., N/2 - 1] \times [-N/2, ..., N/2 - 1] \), and \( \hat{\omega}_k \) and \( \hat{J}_k \) are the Fourier coefficients of \( \omega \) and \( J \) at time \( t \), respectively.

The value of \( \hat{J}_k \) is obtained by the combination of FFTs and the so-called de-aliasing technique by phase shifts [27, 73]. Here we take the treatment of \( S = u \partial \omega / \partial x \) as an example. In the first step we compute the following quantities on the grid points in physical space using FFTs:

\[ u_j = \sum_k \hat{u}_k e^{ik \cdot x_j} , \quad U_j = \sum_k \hat{u}_k e^{ik \cdot (x_j + \Delta)} , \quad (4.4) \]

\[ \left( \frac{\partial \omega}{\partial x} \right)_j = \sum_k ik_x \hat{\omega}_k e^{ik \cdot x_j} , \quad \left( \frac{\partial \Omega}{\partial x} \right)_j = \sum_k ik_x \hat{\Omega}_k e^{ik \cdot (x_j + \Delta)} , \quad (4.5) \]

where \( j = (j_x, j_y) \in [0, 1, ..., N - 1] \times [0, 1, ..., N - 1] \), \( x_j = (2 \pi j_x / N, 2 \pi j_y / N) \) and \( \Delta = (\pi / N, \pi / N) \). Then we compute on each of the grid points \( j \) the quantities \( r_j = u_j (\partial \omega / \partial x)_j \) and \( R_j = U_j (\partial \Omega / \partial x)_j \). Subsequently, we use the FFT again to transform these quantities to spectral space:

\[ \hat{r}_k = \frac{1}{N^2} \sum_j r_j e^{-ik \cdot x_j} , \quad \hat{R}_k = \frac{1}{N^2} \sum_j R_j e^{-ik \cdot (x_j + \Delta)} . \quad (4.6) \]
The Fourier coefficients of $S = u\partial \omega / \partial x$ are obtained by $\hat{S}_k = (\hat{r}_k + \hat{\tilde{r}}_k)/2$. A complete overview of the evaluation of the nonlinear term with the de-aliasing procedure is schematically illustrated in Fig. 4.1.

### 4.2 Parallelization of the 2D Fourier pseudospectral code

#### 4.2.1 General introduction to parallel computation

If a single computer can solve a problem in ten seconds, can ten of these computers, connected by a network, solve the same problem in only one second? These and similar questions have intrigued computer scientists since the earliest days of computers. Due to the increasing computational demands in scientific research and engineering applications, scientists and engineers become increasingly relied on large-scale parallel computing to perform numerical simulations.

In order to coordinate tasks of multiple nodes (cpu’s) working on the same problem, some form of inter-processor communication is required to:

1. convey information and data between processors,
2. synchronize node activities.

The way processors communicate is dependent upon memory architecture, which, in turn, will affect how you write your parallel program. From the point of view of the programmers, the parallel computers can be divided into two categories: shared memory systems and distributed memory systems. The schematic sketches of such systems with four cpu’s is shown in Fig. 4.2 (shared memory) and Fig. 4.3 (distributed memory). For shared memory computers, multiple processors operate independently but share the same memory resources. Only one processor can access the shared memory location at a time. Synchronization is achieved by controlling tasks’ reading
from and writing to the shared memory. This kind of architecture is easier to use efficiently, and data sharing among tasks is fast (the same speed of memory access). However, because memory is bandwidth limited, increase of processors without increase of bandwidth can cause severe bottlenecks. In addition, the user is responsible for specifying the process of synchronization, e.g., locks of the data to avoid different processors writing into the same part the memory at the same time.

For distributed memory computers, multiple processors operate independently but each processor has its own private memory. Data is shared across a communications network using message passing. The user is responsible for synchronization using message passing. Unlike shared memory computers, memory is scalable to the number of processors. Increasing the number of processors, the size of memory and bandwidth increases as well. Each processor can rapidly access its own memory without interference. Nevertheless, it is difficult to map existing data structures to this memory organization. Moreover the user is responsible for sending and receiving data among processors.

Generally speaking, the hardware suitable for shared memory systems is much more expensive, but it is easier to obtain high parallel efficiency. This is in contrast with distributed memory systems: the hardware is cheaper, but it is less easy to get a high parallel efficiency. It is the tendency to combine these two primary memory architectures to build larger and larger supercomputers.

For shared memory systems, the typical programming language is OpenMP, while MPI (Message Passing Interface) is a standard for distributed memory systems. Because it is possible for shared memory computers to be treated like distributed memory systems (simply by dividing the big memory into smaller ones), MPI can also be used on shared memory computers (as the SGI Origin 3800 which has been used for the present investigation). This makes MPI the standard language of parallel computing.

In the following, we will use MPI to parallelize our code. In this particular problem, there are only a few places in the code where data is transferred between different processors. It is easier to program using MPI because we can control the flow of the data directly.

4.2.2 Background

Although it is a quite natural approach in the CFD community to parallelize three-dimensional Fourier pseudospectral codes for DNS of (homogeneous and isotropic) turbulence [74–76], one has been more reluctant to deal with the parallelization of its two-dimensional counterpart. This is due to the presence of some bottlenecks that particularly affect the improvement of the efficiency of parallelized 2D Fourier pseudospectral algorithms. Two of these bottlenecks (with the second one also affecting the efficiency of 3D simulations) are:

1. Unlike the case of three-dimensional flow computations, the 2D computation has an array that is too small to distribute to more cpu’s. The overhead involved in distributing the array can sometimes require more time than the time needed to compute the flow evolution with one cpu only (this is, of course, hardware
4.2. Parallelization of the 2D Fourier pseudospectral code

Table 4.1: The distribution of tasks. The second line indicates the total amount of communication per time step in the code.

<table>
<thead>
<tr>
<th></th>
<th>1 cpu</th>
<th>2 cpu's</th>
<th>4 cpu's</th>
<th>6 cpu's</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of FFTs per cpu</td>
<td>10</td>
<td>6</td>
<td>3</td>
<td>1 or 2</td>
</tr>
<tr>
<td>Message passing per time step</td>
<td>0</td>
<td>$2N^2$</td>
<td>$6N^2$</td>
<td>$12N^2$</td>
</tr>
</tbody>
</table>

dependent). For example, an array of $128^3$, a relatively low Reynolds number computation for 3D flows, is already big enough to achieve a good effect through parallelization. But in the 2D computations, where an array of $512^2$ is often adequate for the computations we want to perform, it is still not worthwhile to distribute it over several cpu’s (a further discussion about this issue is supplied in Chapter 4.2.4).

2. We are dealing with a time-dependent problem, so it is not possible to parallelize the code with a time-advanced loop. For our simulations, about $10^5$ – $10^6$ time steps are needed for one typical simulation if we want to see the appearance of the final states.

4.2.3 Parallelization by task distribution

Keeping these two remarks in mind, the only way to parallelize the code is to distribute the major, time-consuming part of the code within the time loop, and assign it to different cpu’s. In the meanwhile, we should make sure that the amount of message passing per time step is as small as possible.

Considering the time consumed in a single cpu simulation with a resolution of $512^2$, we found that 91% of the time is used for the (de-aliased) computation of the nonlinear contribution $J$ of the vorticity equation. Moreover, during the computation of $J$ we found that 95% of the time is consumed for the evaluation of the FFTs. For the resolution of $1024^2$, even a larger proportion of time is consumed on FFTs. Consequently, and not surprisingly, common sense advises to parallelize the FFTs [77, 78]. For the 3D Fourier pseudospectral codes this approach yields quite easily a high parallelization efficiency. However, for the 2D codes, one has to be very careful to minimize the total volume of the data communication, and exploit the computer’s ability to find the fastest communication methods [79, 80].

There is a complication usually overlooked in this particular problem: For each time step, ten FFTs have to be evaluated, no matter how you arrange your code, and there is inevitably a too large amount of message passing. However, the fact that most computation time is consumed on FFTs, does not imply that we are obliged to parallelize our code with parallelized FFTs. As is shown in Fig. 4.1: ten FFTs are required for each time step - the first six FFTs are needed by the later four FFTs. As a result, the method of parallelization becomes straightforward: rearrange those FFTs within the time loop, and distribute them to different cpu’s (Figs. 4.4). The structure of the code is modified such that each processor handles fewer FFTs. The different possibilities are summarized in Table 4.1. For the case with 2 cpu’s and 4 cpu’s, the
Figure 4.4: Parallel Task Distribution (PTD) scheme for the case of 2 (top left), 4 (top right) and 6 (bottom) cpu’s (the parts divided by the dashed lines belong to different cpu’s). The long $J_k$ box represents computations belonging to one cpu: all the data are gathered into one processor to get $\hat{J}_k$ and then $\hat{\omega}_k$ of the next time step, after which the new $\hat{\omega}_k$ is passed to all other cpu’s to start the next time-step computation.
values of \( \tilde{\omega}_k \) and \( \tilde{\Omega}_k \) are calculated twice to save the time of message passing and keep all the processors busy. For the case with 6 cpu's, there are two processors that only deal with one FFT each, so there is about 50\% killing time for these two processors. Additionally, there is more message passing time involved in the example with 6 cpu's.

For the 4 cpu's and 6 cpu's cases, some of the communication can happen simultaneously. For example, the value of \( \tilde{\omega}_k \) and \( \tilde{\Omega}_k \) can be transferred at the same time for the 6 cpu's case, since there is no overlapping between the sending processors and receiving processors. The total communication time is actually equal to \( 4N^2 \) for 4 cpu's case (\( 7N^2 \) for 6 cpu's case).

### 4.2.4 Comparison with the former parallel scheme

Here we report three runs with the resolution of \( 512^2 \), and with a time step \( \Delta t = 0.0005 \). Three different initial vorticity fields are used in order to illustrate: 1) the appearance of different final states, 2) the relevant time-scale beyond which the stationary final states emerge, and 3) the need to perform parallel computations to test the applicability of the mean-field statistical predictions.

Figs. 4.5 refers to a run that leads to the “bar” state which is predicted by the statistical-mechanical theory in the “patch” formulation for large enough patch size. The vorticity in the four quadrants of the box is alternately flat and positive (top-left, bottom-right) or negative (top-right, bottom-left), with some small-scale random noise added on top of the basic vorticity profile in order to stimulate a rapid evolution and symmetry breaking of the vorticity field. One aspect in this run that is worth noticing can be found at \( t \approx 50 \), when the profile of the “bar” has already begun to emerge, but it more looks like a travelling wave. It lasts so long (until \( t \approx 1000 \), which is equivalent to several months of computing on a Compaq Alphasstation XP 1000) that one might conclude already at \( t \approx 100 \) that either a dipolar state will finally emerge or a travelling wave pattern is the final state of the simulation. The travelling wave state is not predicted by any statistical-mechanical theory, and one might consider this observation as a failure of the theory. Apparently, a long-time simulation (until \( t \approx 1250 \)) is necessary to reveal the “bar” solution consistent with the prediction of the statistical-mechanical method based on the “patch” formulation for large enough patch size.

Figs. 4.6 shows a much more turbulent initial vorticity field and the dipole state into which it quickly evolves. Actually, this simulation is characteristic for many numerical simulations of 2D decaying Navier-Stokes turbulence. Figs. 4.7 shows four vortices that stay at a relatively large distance from each other (with initially some small-scale noise added on top of this vorticity field). Here, a similar situation as shown in Figs. 4.5 occurs: the four-vortex state stays for quite long time, before the final dipolar state emerges, and one could arrive at the false conclusion that the quadrupolar state finally prevails.

The simulation shown in Figs. 4.6, computed by our MPI Fortran 90 code, took about 30 hours using 6 cpu's on the SGI Origin 3800 (SARA, Amsterdam). A week of computation time was needed for the same run on a single-cpu workstation (Compaq
Figure 4.5: Contour plots of evolving vorticity in a simulation (with Re = 5000) which leads to a one-dimensional (“bar”) late-time state. Dashed contours represent positive vorticity and drawn contours represent negative vorticity. The evolution is perhaps less than fully-developed turbulence, but it turned out to be predicted by the statistical-mechanical theory based on the “patch” formulation, for large enough patches (see Figs. 3.8).
4.2. Parallelization of the 2D Fourier pseudospectral code

Figure 4.6: Contour plots of evolving vorticity in a run (with Re = 5000) leading to a more familiar dipolar late-time state, which is initially fully-developed, and which is a prediction of the “point” version of the statistical theory. Dashed contours represent positive vorticity and solid contours represent negative vorticity.
Figure 4.7: Contour plots of evolving vorticity in a run (with Re = 10000) which is also predicted by the “point” theory. Dashed contours represent positive vorticity and solid contours represent negative vorticity.
4.2. Parallelization of the 2D Fourier pseudospectral code

![Graph showing speedup vs. number of processors]

Figure 4.8: The speedup of our 2D Fourier pseudospectral code as function of the number of processors on the SGI Origin 3800.

Alphastation XP 1000). The simulation shown in Figs. 4.5 took about 2 weeks using 6 cpu’s which implies a kind of burdensome task for local workstations (where it would take approximately three months).

A limitation of the presently proposed method of parallelization is related with the limited number of cpu’s which can be used in a simulation. Without splitting the array, it is not possible to let the number of cpu’s be changed arbitrarily. So far we can only use 2, 4 and 6 cpu’s. With this requirement in mind, we can conclude that the code is efficiently parallelized. This is illustrated in Fig. 4.2.4, which shows the speedup on a SGI-Origin 3800. For the resolution of 512² (which is the most frequently used resolution in our simulations), acceptable speedup is observed. For the resolution of 1024², we obtain an even better speedup using our scheme. As comparison, the scheme based on employing parallel FFTs in this subject may not work better than our scheme. This is partly due to the improvement of the speed of the currently available single cpu’s which outruns the increase in communication speed between the processors. The efficiency of parallel FFTs will therefore decrease faster than the efficiency of the currently proposed method with increasing processor speed. In general it might be conjectured that the improvement of the speed of cpu’s encourages a further minimization of the amount of message passing in order to achieve a sufficiently large parallelization efficiency.

Table 4.2 shows the results of a comparison of the performance of the presently proposed parallelization scheme, which we will denote by PTD (Parallel Task Distribution) and the scheme solely based on parallelized FFTs on the SGI Origin 3800 for simulations with a resolution of 256² and 512². We have made a comparison using the most efficient communication method by the system [81–84]. In particular, the one-side communication method in MPI-2 is also taken into consideration to make sure that our communication is implemented in the most efficient way [83]. For the scheme based on parallel FFTs (PFFT), we consider the Transpose-Split FFT method, which has proved to be more efficient [78, 85]. (The fact that the Transpose-Split FFT needs less message passing makes this particular implementation faster than the other
Table 4.2: Comparison of the PTD-scheme and the PFFT-scheme for a resolution of $256^2$ (top) and $512^2$ (bottom). The time (in seconds) needed for the completion of one time step in the simulation.

<table>
<thead>
<tr>
<th></th>
<th>256$^2$</th>
<th>1 cpu</th>
<th>2 cpu’s</th>
<th>4 cpu’s</th>
<th>6 cpu’s</th>
<th>16 cpu’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTD</td>
<td>0.3</td>
<td>0.20</td>
<td>0.16</td>
<td>0.18</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>PFFT</td>
<td>0.3</td>
<td>0.26</td>
<td>0.21</td>
<td>-</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>512$^2$</td>
<td>1 cpu</td>
<td>2 cpu’s</td>
<td>4 cpu’s</td>
<td>6 cpu’s</td>
<td>16 cpu’s</td>
</tr>
<tr>
<td>PTD</td>
<td>1.6</td>
<td>1.04</td>
<td>0.65</td>
<td>0.58</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>PFFT</td>
<td>1.6</td>
<td>0.86</td>
<td>0.71</td>
<td>-</td>
<td>0.57</td>
<td></td>
</tr>
</tbody>
</table>

parallel FFTs$^1$.) Although the size of the array that this version of PFFT passes is smaller than that of PTD, many of these arrays have to be communicated between the different processors. Therefore, the Transpose-Split FFT version of PFFT still requires much more frequent message passing than the currently proposed PTD-scheme. We can conclude from Table 4.2 that our simple parallel scheme based on task distribution (PTD) offers a better performance than PFFT for the resolution $256^2$, and a reasonable speedup for the resolution of $512^2$ in comparison with PFFT. For higher resolutions, which is not essential for the present investigation, the PFFT will likely show a better performance [88]. In that case, it is recommended to apply a combination of both methods: PTD with parallel FFTs. That is, instead of using 2, 4 or 6 cpu’s, we might use $2n$, $4n$ and $6n$ cpu’s (with $n$ the number of cpu’s used per FFT).

To summarize, we propose the use of PTD scheme, which is very simple to implement, to parallelize a two-dimensional Fourier spectral code as the first step in any parallelization attempt. If this approach appears unsuccessful, any more complicated scheme will also not result in a speed-up of the computations. If the PTD approach is successful, a more complex parallelization scheme could be implemented. For example, a combination of PTD and PFFT schemes could be considered. It is anticipated that this combination will increase the communication complexity.

### 4.3 Conclusion

An outline has been presented of a two-dimensional Fourier pseudospectral code for Navier-Stokes simulations that has been structured to efficiently utilize parallel computers. The novel idea is based on task distribution for the computation of the Jacobian $J = \mathbf{v} \cdot \nabla \omega$ instead of the classical approach by applying parallelized FFTs (abbreviated as PFFT-scheme). A Parallel Task Distribution (PTD) scheme has therefore been implemented. The comparison of the PTD-scheme with the classical PFFT-method

$^1$The transpose-split 2D FFT requires, at each message-passing phase, each processor to send $p - 1$ messages of size $N^2/p^2$ ($p$ is the total number of the processors), whereas the parallelized 2D FFT would require each processor to send $\log_2 p$ messages of length $N^2/p$, at each exchange.
shows that our scheme is simple to implement, efficient and superior.

Some interesting simulations have been performed due to the availability of the efficient PTD-scheme which provided a better understanding of the late-time states of decaying 2D Navier-Stokes turbulence. Particularly the intensive study of the “bar” final state was not possible without the parallelized code based on the PTD-scheme. As will be seen in Chapter 5, several runs aimed at studying the emergence of the “bar” quasi-stationary final state have been conducted. Those simulations required five to ten times longer computation times than the by now classical simulations that lead to the dipolar quasi-stationary final state. It is noteworthy that it would be virtually impossible to conclude our investigation of the “bar” quasi-stationary final state in a 4-year research schedule with a serial code.

The three runs briefly discussed in this chapter to illustrate the need for the PTD-scheme, and which served as test cases for speedup studies, will be discussed in more detail in Chapter 5.
Chapter 4. Numerical simulation of turbulence
Chapter 5

Dynamical solutions and comparisons

In this chapter, we present the results of dynamical, pseudo-spectral-method solutions of the 2D NS equation, using a resolution of $512 \times 512$ Fourier modes. The time step in all simulations is fixed at 0.0005 and determined by the CFL condition. The initial energy, using the normalization of Eq. (3.3), is 0.5. There is no hyperviscosity or small-scale smoothing of any kind.

We will discuss 13 runs altogether. About 40 more simulations were performed, but the main conclusions can be illustrated by these 13 runs. Readers can consult Tables 5.1, 5.2 and Figs. 5.1 for a complete overview of 13 runs.

5.1 Maximum entropy states

5.1.1 The classical dipolar final state

The first run we report here is essentially a reproduction of the run of Matthaeus et al. [3–5], but with a lower Reynolds number ($1/\nu = 5000$), corresponding to an initial Taylor-scale Reynolds number

$$R_{\lambda} \approx \sqrt{\frac{10}{3}} \frac{E}{\nu \sqrt{\Omega}} \approx 558.$$ 

$R_{\lambda}$ has increased to 3143 by the end of the run.

This run served as a benchmark for our parallelized code in the beginning, and the McWilliams initial conditions [72] that were used in the computations by Matthaeus et al. can also be used as noise, with variable amplitude, to be added to later simulations to break unwanted symmetries and accelerate the dynamical development. The run also provides an opportunity to introduce the types of data displays that will be used throughout the rest of this Section. The Fourier modal energy spectrum $E(k) \equiv (1/2) |\psi(k)|^2$ is initialized according to

$$E(k) = \frac{C}{1 + (\frac{k}{k_0})^4}$$
<table>
<thead>
<tr>
<th>Section</th>
<th>Subsection</th>
<th>Initial conditions</th>
<th>Run Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Maximum states</td>
<td>5.1.1</td>
<td>Random noise</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>5.1.2</td>
<td>tanh-Poisson quadrupole</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>5.1.3</td>
<td>smaller patch placed asymmetrically(1)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>smaller patch placed asymmetrically(2)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>smallest patch placed asymmetrically(1)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>smallest patch placed asymmetrically(2)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>slanting bar</td>
<td>7</td>
</tr>
<tr>
<td>5.2 Local Maximum states</td>
<td>5.2.1</td>
<td>tanh-Poisson 64-pole</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tanh-Poisson 64-pole</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tanh-Poisson 8-bar</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>5.2.2</td>
<td>sinh-Poisson quadrupole</td>
<td>11</td>
</tr>
<tr>
<td>5.3 Oddities</td>
<td>5.3.1</td>
<td>tanh-Poisson 4-bar</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>5.3.2</td>
<td>tanh-Poisson 16-pole(4 patches removed)</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Run Number</th>
<th>With noise?</th>
<th>Figure Number</th>
<th>Final states</th>
<th>$1/\nu$</th>
<th>$Re_\lambda(t = 0)$</th>
<th>$Re_\lambda(t = \text{final})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>5.2-5.3</td>
<td>dipole</td>
<td>5000</td>
<td>558</td>
<td>3143</td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>5.4-5.9</td>
<td>bar</td>
<td>5000</td>
<td>2391</td>
<td>4920</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>5.10(a)</td>
<td>bar</td>
<td>8000</td>
<td>6870</td>
<td>8592</td>
</tr>
<tr>
<td>4</td>
<td>No</td>
<td>5.10(b)</td>
<td>bar</td>
<td>8000</td>
<td>6908</td>
<td>8616</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>5.11(a)</td>
<td>dipole</td>
<td>8000</td>
<td>6813</td>
<td>9955</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>5.11(b)</td>
<td>dipole</td>
<td>8000</td>
<td>7077</td>
<td>9929</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>5.12</td>
<td>bar</td>
<td>8000</td>
<td>4002</td>
<td>8035</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>5.14</td>
<td>8-bar</td>
<td>10000</td>
<td>2041</td>
<td>2006</td>
</tr>
<tr>
<td>9</td>
<td>Yes</td>
<td>5.15</td>
<td>dipole</td>
<td>10000</td>
<td>2036</td>
<td>11400</td>
</tr>
<tr>
<td>10</td>
<td>Yes</td>
<td>5.16</td>
<td>dipole</td>
<td>10000</td>
<td>3228</td>
<td>11500</td>
</tr>
<tr>
<td>11</td>
<td>Yes</td>
<td>5.17-5.21</td>
<td>dipole</td>
<td>10000</td>
<td>5930</td>
<td>11186</td>
</tr>
<tr>
<td>12</td>
<td>Yes</td>
<td>5.22</td>
<td>dipole</td>
<td>20000</td>
<td>12878</td>
<td>22426</td>
</tr>
<tr>
<td>13</td>
<td>No</td>
<td>5.23-5.25</td>
<td>quadrupole</td>
<td>12500</td>
<td>6374</td>
<td>10569</td>
</tr>
</tbody>
</table>
5.1. Maximum entropy states

Figure 5.1: Time evolutions of Taylor-scale Reynolds number ($R_\lambda \approx \sqrt{\frac{10}{3} \frac{E}{\nu \sqrt{\lambda}}}$) for all numerical simulations.
Figure 5.2: Equally spaced contours of constant vorticity at three different times (left column) and corresponding modal energies at the lower values of k (right column), during the evolution of the McWilliams/Matthaeus initial conditions. These have no flat patches of vorticity initially, even approximately.
Figure 5.3: The \( \omega - \psi \) scatter plot for the run shown in Fig. 5.2 (which is close to Fig. 3.1(a)).

for \( 1 \leq k \leq 120 \) (here, and for this purpose only, the wave number \( k \) is binned in integer values by a standard Fortran routine, and the partition of energies among modes with the same integer \( k \) is decided by a random number generator), and zero otherwise. The phases of the Fourier coefficients are chosen from a Gaussian random number generator. \( C \) is a constant to be adjusted to make the total initial energy equal to 0.5.

The results were quite similar in all respects to those of Mattheus et al. [3–5], so only a few figures (Figs. 5.2 and 5.3) are shown here. The contour plots of vorticity and stream function relax to the familiar dipole final states. The left column of Figs. 5.2 shows the vorticity contours at three separate times. The right column of Figs. 5.2 shows the modal energy spectra, for the lower part of \( k \)-space (the maximum \( k \) is 241) at those same times, exhibited as three-dimensional perspective plots. They are initially broad-band, but evolve as expected to be concentrated in the lowest values of \( k \). In Fig. 5.3, a scatter plot of \( \omega \) vs. \( \psi \) shows a good agreement with the hyperbolic sinusoidal dependence of vorticity on stream function, as predicted in the point formulation (Fig. 3.7).

5.1.2 Observation of the “bar” final state

A different, and interesting, evolution results (\( 1/\nu = 5000 \)) if we initialize using the quadrupole solution from Eq. (3.12). In this run, \( R_\Lambda \) increases from 2391 initially to 4920 at the end. This quadrupole solution is not predicted to be the maximum entropy state from Eq. (3.31), or any other equation, patch or point; and so should evolve when initialized with some random noise. The quadrupole solution was found
to have a symmetry which persists in time, and the noise, it is hoped, will permit that symmetry to be broken. It was found that the symmetry persisted until the noise was raised to quite substantial levels. Figs. 5.4 are perspective plots of the initial vorticity as a function of $x$ and $y$, with and without the noise added. The noise level in this case is about twice the quadrupole vorticity, and its randomness should break any symmetry that might be present.

Fig. 5.5 shows the evolution of the vorticity and stream function contours that result from the initial conditions shown in Fig. 5.4(b). The intermediate panel ($t = 15$) shows no obvious connection to the symmetry of Fig. 5.4(a) or to that of the final panels ($t = 1250$) of Figs. 5.5 (the first is odd about the centre lines of the basic square, but clearly, the $t = 15$ state has no such symmetry). The relaxation to the one-dimensional “bar” state at $t = 1250$ has been quite robust in several such runs. Fig. 5.6 is a $\omega - \psi$ scatter plot for the bar state. There has been considerable decay of vorticity, as evidenced by the disappearance in Fig. 5.6 of the high and low values visible in Fig. 5.4(b). By $t = 1250$, the energy ($E$) has decreased to 58.4% of the initial value.

When we consider the modal energy spectra in Fig. 5.8, for this evolution, we see that the initial and final states are dominated by four and two Fourier modes, respectively. In fact, the evolution after $t = 40$ has this character. The energy spectrum at $t = 15$ is somewhat more broad-band than either, but whether it should properly be called “turbulent” may be debated; it never achieves the fully broad-band character of the evolution shown in Fig. 5.2, for example. The final state achieved is consistent with the maximum-entropy prediction of the “patch” formulation for suitably large-sized
Figure 5.5: Contours of constant vorticity (left column) and constant stream function (right column) at three different times for the run originating from the initial vorticity distribution (with noise) shown in Fig. 5.4(b).
Figure 5.6: The $\omega - \psi$ scatter plot of the late-time state achieved in the run shown in Fig. 5.5 (which is close to Fig. 3.1(c)). The plot is also almost a straight line, which characterizes a selectively decayed state with only one $k$ in the spectrum.

Figure 5.7: Angle-averaged modal energy and enstrophy spectra at three different times for the run shown in Figs. 5.5 and 5.6. Note that these are not the usual omnidirectional spectra; since they do not contain the factor $2\pi k$, their maxima occur at the minimum $k$. 
5.1. Maximum entropy states

Figure 5.8: Low-k modal energy spectra (left column) and vorticity histogram (right column) at three different times for the run shown in Figs. 5.4(b) and 5.5. The vorticity histogram shows the number of times a particular value of $\omega$ is recorded as one cycles through the computational grid. Notice that there is substantial variation from one time to another. (An Euler equation solution should preserve this histogram in time.)
Figure 5.9: Time evolution, for the run shown in Figs. 5.4(b) through 5.8, of the four global quantities energy, enstrophy, $\sqrt{\Omega/E}$ or mean wave number, and ratio of the square root of palinstrophy to enstrophy. Note that the mean wave number (Taylor microscale) is never large, and so the strict definition of turbulence is not fulfilled at any time during the run. Nevertheless, the topology changes dramatically. This is effectively a selectively decayed state [66], and may not invite a probabilistic explanation.
5.1. Maximum entropy states

patches (Fig. 3.4), but not with the same formulation for smaller-sized ones (Fig. 3.4). Figs. 5.7 are binned plots of the angle-averaged modal energy spectrum \((E(k))\) and enstrophy \((\Omega(k))\) spectrum at the three different times. The right column of Figs. 5.8 shows a histogram of the vorticity at three different times in this evolution; it is in effect a frequency distribution for the appearance of various values of the vorticity over the plane, one per computational cell in the \(512^2\) array. It will be seen that this distribution changes enormously over the course of the run, which it cannot do, of course, in any ideal Euler picture. Fig. 5.9 shows the time history of several global quantities for this run: energy, enstrophy, mean wave number (square root of the ratio of enstrophy to energy), and square root of the ratio of palinstrophy \((P \equiv \Sigma_k (1/2) k^2 |\omega(k)|^2)\) to enstrophy, showing spikes that are characteristic of vortex merger events at high but finite Reynolds numbers.

Figs. 5.4(b) through 5.9, show, then, a reproducible evolution of an initial condition consisting of a patch quadrupole plus large amounts of random noise into a one-dimensional “bar” most-probable state. This state is more probable than the dipole, according to a “patch” analysis if the patch size is big enough, though it is not for a smaller but finite patch size.

A similar initial condition is used by Segre and Kida [86]. However, their computation made use of hyperviscosity and appears not to have run long enough for the proper late-time state to evolve.

5.1.3 Further investigations on the “bar” final state

In the previous section we have investigated the emergence of the “bar” final state by considering an antisymmetric basic flow (the quadrupole solution) with a considerable amount of noise added to it in order to break the symmetry of the basic flow. We decided to devote some more efforts to find other initial conditions leading to the “bar” quasi-stationary final state, and we will see shortly that the appearance of the “bar” final state is not that accidental as might erroneously be concluded from the previous set of simulations. The first idea here is to start from the same quadrupolar solution, but distort the initial condition slightly in the following way: we shrink the patch size with a small amount and reposition the patches slightly (see, for example, Figs. 5.10).

The symmetry of the basic flow has been broken already, and there is no need to add any noise to it. The simulations are carried out with a resolution of \(512 \times 512\) Fourier modes, and the Reynolds number is fixed at \(1/\nu = 8000\). As can be seen from Figs. 5.10 we have performed two simulations with two different distorted quadrupolar initial conditions, with the patch size reduced by a factor of \(7/8 \times 7/8 = 49/64\) compared to the patch size of the original quadrupole initial condition (see Fig. 5.4(a)), and both runs clearly reveal the emergence of the quasi-stationary “bar” final state. A similar set of simulations has been carried out, but now with the patch size even further reduced. In Figs. 5.11, we have shown the vorticity contour plots of runs with the patch size reduced by a factor of \(3/4 \times 3/4 = 9/16\), and it is clear that no “bar” final state is
Figure 5.10: Contours of constant vorticity for two runs with slightly different initial conditions in the left (a) and the right (b) column. In both runs, the initial patch size is reduced by a factor of $7/8 \times 7/8 = 49/64$, and the patches are displaced with respect to the quadrupole initial condition shown in Figs. 5.4.
5.1. Maximum entropy states

Figure 5.11: Contours of constant vorticity for two runs with slightly different initial conditions in the left (a) and the right (b) column. In both runs, the initial patch size is reduced by a factor of $3/4 \times 3/4 = 9/16$, which is a somewhat larger reduction than the initial conditions displayed in Figs. 5.10, and the patches are displaced with respect to the quadrupole initial condition shown in Figs. 5.4.
contours of $\omega$ at $t=0$

contours of $\omega$ at $t=30$

contours of $\omega$ at $t=1175$

contours of $\psi$ at $t=0$

contours of $\psi$ at $t=30$

contours of $\psi$ at $t=1175$

t=1175

Figure 5.12: Another initial condition leads to the “bar”.
5.1. Maximum entropy states

![Diagram](image)

Figure 5.13: Three pairs of solutions. (a) dipole and bar; (b) quadrupole and slanting bar; (c) octapole and 4-bar.

found in this case.

We may recall that in Figs. 3.8, the $E - S$ plot predicts that a large patch size leads to the “bar” quasi-stationary final state and that a small patch size leads to the dipolar final state. The four simulations shown in Figs. 5.10 and 5.11 provides a much more direct proof of our theoretical results on the statistical-mechanical final-state predictions.

One question that might be raised concerns the direction of the “bar” final state. As can be observed in Figs. 5.5 and 5.10, it can happen in the horizontal or vertical direction (and should occur with equal probability due to their symmetrical equivalence). However, the “bar” final state (with $2\pi$ periodicity perpendicular to the flow direction) has never been observed in any other direction due to lack of periodicity of such a solution. Note that a solution with periodicity less than $2\pi$ perpendicular to the flow direction enables a flow rotated with respect to the $x$ and $y$ direction (e.g., a direction of $45^\circ$ is needed for a bar solution with a $\sqrt{2}\pi$-periodicity).

We have conducted another set of simulations with initial conditions that was found to lead to the “bar” final state. The initial condition is based on the slanted bar solution already referred to in the previous paragraph ($\sqrt{2}\pi$-periodicity), where again a certain amount of noise is added to it. We let the flow evolve and as shown in Figs. 5.12 the “bar” final solution is obtained eventually. The Reynolds number in this simulation is fixed at $1/\nu = 8000$ (initially $R_\lambda = 4002$ and it increases to 8035 at the end of the simulation). Attention should be drawn to the $\omega$-$\psi$ scatter plot obtained at the end of the simulation. It is similar to the scatter plot obtained for the “bar” solution in the
preceding Section 5.1.2, but some subtle differences can be observed. When considered in more detail, the scatter plot is in fact a double-valued structure. Although the double-valued structure is not very clear, it nevertheless indicates a slight asymmetry in the band structure of $\omega$ and $\psi$. It is not clear how persistent the double-valued structure is. We will discuss these particular scatter plots in more detail in Section 5.3.2 where much more clear scatter plots with a double-valued structure are found.

The dipole and the bar solution obtained from the Poisson-like equations have a very close entropy (see Figs. 3.7 and 3.8). It is therefore inferred that the $45^\circ$ slanted bar solution and the bar solution with $\pi$-perodicity can be related, in a similar way as the bar-dipole combination, with multipole solutions of the Poisson-like equations. In Fig. 5.13, we see the three pairs of solutions from the Poisson-like equations. The first group of solutions (a) is already familiar to us and we will extrapolate conclusions obtained from the analysis of this pair (see Chapter 3). In the case of the second group of solutions (b) we anticipate that the slanting bar will have a larger entropy than the quadrupole (keeping $E$ fixed) provided that relatively large patch sizes are assumed. Reducing the patch size we might assume that the entropy of the slanting bar final state will become smaller than the entropy of the quadrupolar final state. The numerical proof of this statement (similar to our analysis outlined in Chapter 3) is time-consuming and looks trivial to us at present. However, we can get an indirect proof from our numerical experiments. We have conducted a few simulations which hints at the appearance of the slanted bar solution. In the very beginning of our quadrupole-to-bar simulation, we have observed the slanting bar solution in the case we did not put any noise on top of the quadrupole (and we did not shrink the patch size)\footnote{There are about 40 more simulations which we finished, but we don’t show there. This run is one of them.}. After the quadrupole breaks up, there is a very short time when the fluid pattern looks like the slanting bar. The fact that it appears in the quadrupole-to-bar simulation suggests that for large patch size the slanting bar must have a larger entropy than the quadrupole. The slanting bar intermediate state was never found when noise was added on top of the quadrupolar initial vorticity distribution.

Finally, the two solutions shown in Figs. 5.13(c) represent flows with a relatively small-scale structure. As a consequence, both the octupole and the four-bar solution always have a smaller entropy than the dipole or bar solution, although the entropy associated with the octupole and 4-bar solutions is expected to be rather close to each other. In particular, simulations with the octupole or 4-bar as initial conditions (with small amount of noise to break the symmetry), always resulted in a dipolar quasi-stationary final states.

5.2 Local maximum states

5.2.1 Evolution from 64-pole initial condition

The contrasting evolutions of the two initial states in the previous section naturally arouse curiosity about whether there is a limit in which one behaviour goes over into
5.2. Local maximum states

contours of $\omega$ at t=0

contours of $\omega$ at t=10

contours of $\omega$ at t=50

contours of $\omega$ at t=250

t=250

Figure 5.14: Time evolution of a 64-pole initial condition, triggered only by round-off error, into a bar state (the scatter plot of final state is close to Fig. 3.1(c)). No noise beyond round-off error has been added to the initial condition.
Figure 5.15: Time evolution of the vorticity contours, starting from the same initial condition as in Fig. 5.14, but with a healthy addition of random noise. (In this run, $R_\Delta$ increases from 2036 initially to 11400 at the end.) The last panel is the late-time $\omega - \psi$ scatter plot for the resulting dipole (which is close to Fig. 3.1(a)).
5.2. Local maximum states

Figure 5.16: Evolution of the one-dimensional “8-bar” initial condition, with random noise added initially. (In this run, $R_\lambda$ increases from 3228 initially to 11500 at the end.) The evolution is toward a dipole, now, as seen in the bottom panel (which is close to Fig. 3.1(a)).
the other. In this sub-section, we attempt an answer to this by considering initial conditions which originate in high-order multipole solutions of the patch formulation, not maximum entropy states in either formulation, but intuitively closer to the random initial conditions that led to the dipole solution before, in the first run reported.

There are four numerical solutions in this group, with $1/\nu = 10,000$ for all four runs. They all originate in the 64-pole solution of the patch formulation, using the 3-level equation, Eq. (3.12). The same conclusions that we reach can also be reached using 16-pole initial states, but we will not display those results here.

The motivation was to see if the high-order multipole initializations would lead to “bar” final states, the way the quadrupole does. Intuitively, they would seem to be closer to our picture of what true turbulence might look like. The bar states are no longer found, but there are “local maximum” entropy states that can be achieved in the limit of low noise in the initial conditions.

First consider what happens to the 64-pole initial conditions without any noise, as shown in Fig. 5.14. A straightforward laminar evolution occurs, with the end product ($t = 250$) being a one-dimensional bar state with a total of eight maxima and minima. This state is essentially dominated by one Fourier mode, as indicated by the essentially linear pointwise dependence of vorticity on stream function. In Figs. 5.15, we show the evolution of the same initial conditions with only a low level of noise: $1/2000$ of the amplitude of the 64-pole solution, not visible on the $t = 0$ contour plot. In this case, fully-developed turbulence does seem to result, with a dipole final state as the end result.

The third run in the group concerns the result of taking the final bar state as shown in Figs. 5.14, we raise the initial energy: $E(t = 0) = 0.5$, without putting any noise into it, and allow it to run. This appears to be a time-independent state, stable in the presence of round-off error for the duration of the run, but not a maximum-entropy state and not stable in the presence of noise of greater amplitude. This is clear from Figs. 5.16, which show the development of the unstable evolution induced by putting on the same level of random noise as in Figs. 5.15, with a fully-developed turbulence and a dipole final state as the result.

There are thus some subtleties revealed by these four runs. Bar final states, predicted by the patch theory with sufficiently large patch size, do result from the evolution of a patch quadrupole plus sufficient noise. However, higher order multipoles from the patch formulation seem to be unstable at low levels of random noise, and evolve into dipoles, consistently. Still less easy to fit into the picture is the laminar evolution of the “local maximum” 64-pole state without noise into the bar state shown in Figs. 5.14. Neither state is in this case an absolute maximum entropy one, though both are in some sense local maxima.

The initial conditions illustrated in the first panel of Figs. 5.14 are not those of an approximately steady state. What Figs. 5.14 to 5.16 apparently show is that such a checkerboard initial condition can evolve to a metastable quasi-stationary state of less than absolute maximum entropy which is nonetheless unstable if it is excited by greater random noise than pseudospectral round-off error provides. The high degree of symmetry apparent in the initial condition may prejudice the evolution, also. The
5.2. Local maximum states

differences in the evolutions in Figs. 5.14 and 5.15 support this speculation.

5.2.2 Evolution starting from a sinh-Poisson quadrupole

![Contours of ω at t=0](image1)

![Contours of ω at t=60](image2)

![Contours of ω at t=175](image3)

![Contours of ω at t=300](image4)

Figure 5.17: Evolution from initial conditions corresponding to a sinh-Poisson quadrupole, plus random noise. Note the differences in evolution from those shown in Figs. 5.4(b) - 5.9, whose initial conditions are superficially similar. The low-k part of the Fourier space contains most of the energy throughout, so the evolution again is not classically “turbulent.”

We have started several runs from initial conditions that originate with a sinh-Poisson quadrupole state, with 1/ν = 10,000. The first is the quadrupole state without any added random noise, and the others are the same state plus smaller or larger amounts of random noise.

The zero-random noise initial conditions seems to be stable and to remain in the same shape in the presence of only round-off error for the duration of the run, as
Chapter 5. *Dynamical solutions and comparisons*

![Energy Spectrum](image)

**Figure 5.18:** Evolution of 2d energy spectrum.

![Vorticity Histogram](image)

**Figure 5.19:** Vorticity histogram at three different times for the evolution shown in Figs. 5.17.
5.2. Local maximum states

Figure 5.20: Scatter plot of the pointwise dependence of $\omega$ upon $\psi$ at two different times, for the evolution shown in Figs. 5.17 and 5.19 (both of them are similar to Fig. 3.1(a)).

Figure 5.21: Evolution of the global quantities as functions of time for the run shown in Figs. 5.17 through 5.20. The merger of vortices around $t = 175$ (from quadrupole to dipole) leads to the “spike” in (d). This is a relatively short process (from $t \approx 170$ to $t \approx 190$). Palinstrophy, or mean square vorticity gradient, is proportional to the enstrophy dissipation rate, and it is known that large vortex merger is an abrupt event, resulting in a rather dramatic loss of total enstrophy [4].
reported previously [87]. Even after putting in enough random noise to manifestly break the symmetry (Figs. 5.17), the quadrupolar shape is maintained for quite a long time before breaking down into a dipole by \( t = 300 \). Figs. 5.19 are vorticity histograms computed at three different times, showing more concentration of \( \omega \) near zero values, characteristic of sinh-Poisson states. The \( \omega - \psi \) scatter plots shown in Figs. 5.20 also illustrate this feature. Evolution of the global quantities for this run is shown in Figs. 5.21, showing abrupt changes in enstrophy and palinstrophy when vortex merger occurs, but otherwise not displaying characteristics of turbulent behaviour. Note that energy is well conserved for this case. In this run, the quadrupole persists for a long time because the energy is concentrated in the four well-separated vortices, before breaking down into a dipole. This is noticeably different behaviour than when we started with the patch quadrupole.

5.3 Summary of a few oddities

In this section, we will present and briefly discuss a few results of simulations yielding quite strange quasi-stationary final states. Although many features of these particular runs are not understood yet, they illustrate the need for future research of decaying 2D Navier-Stokes turbulence.

5.3.1 A run with features of both the “point” and the “patch” formulation

A run \((1/\nu = 20,000)\) that leads to an “unclassifiable” final state is shown in Figs. 5.22. It begins with a 4-bar state, plus a considerable amount of random noise, and ends at a state that shows features of both dipoles and bars. A tentative interpretation of this evolution is that the nonlinearity is simply exhausted by the viscous decay before either evolution can be completed. Once the amplitudes fall below amounts or in Fourier configurations at which the activity is effectively nonlinear, the pattern in place is “frozen” in its topology, and can only slowly decay. The scatter plot has features of both the point and patch predictions.

5.3.2 “Double-valued” structure in the \( \omega-\psi \) scatter plot

Several other runs demonstrated quite puzzling odd features. For example, metastable states were found that would persist for a long time as a consequence of the flow evolution that, though disordered, might be thought to be less than totally turbulent. Consider the numerical simulation with a special type of initial condition: the 16-pole solution of the 3-level patch equation, where some of the (square) vorticity patches are removed, has been used. Most of these runs yielded dipolar final states, but one of the atypical ones reached a vaguely quadrupolar one at \( t = 350 \). The Reynolds
Figure 5.22: Evolution of a 4-bar initial condition, plus random noise, that ceases to evolve (perhaps because of loss of nonlinearity due to decay) before either a dipole or a bar final state is achieved.
number of this particular run was $1/\nu = 12,500$ and the initial and final Taylor-scale Reynolds numbers are approximately $R_\lambda(t = 0) = 6374$ and $R_\lambda(t = 350) = 10569$. An impression of the flow evolution of this run and the vorticity and stream function contour plots of the quasi-stationary final state are shown in Figs. 5.23. The associated $\omega$-$\psi$ scatter plot, depicted in Fig. 5.24, shows a bilinear form as if two locally “most probable” states had formed into a quasi-equilibrium that was slow to break up and decay toward anything globally “most probable”. Figs. 5.25 show the evolution of the global quantities $E$, $\Omega$, $\sqrt{\Omega/E}$ and $\sqrt{P/\Omega}$ for this run.

By comparing the $\omega$-$\psi$ scatter plot shown in Fig. 5.24 with the contour plots of the vorticity and stream function (Figs. 5.23(d) and (e)) we can conclude that the larger negative vortex, which is indicated by the solid arrow, corresponds to the longer negative branch of the $\omega$-$\psi$ scatter plot. The smaller negative vortex (see the dashed arrow) is then represented by the short (negative) branch of the $\omega$-$\psi$ scatter plot.

The vorticity field shown in Fig. 5.23(d) represents a quasi-stationary solution of the Euler equation $D\omega/Dt = 0$, although in this particular case no functional relation between $\omega$ and $\psi$ exists\textsuperscript{2}. This behaviour cannot be explained by the statistical-mechanical theory for Euler flows, even not if the most general formulation of the theory, viz. Eq. (2.37), has been used. At this point, we should admit that although the application of a statistical-mechanical approach to predict the quasi-stationary final states of inviscid flows appears to be very powerful to investigate freely evolving 2D turbulent flows, it still has some limitations which cannot easily be understood.

Several additional numerical runs were carried out using the metastable quasi-quadrupole state from Fig. 5.23(d) as the basis for initial conditions, and we added a significant amount of random noise. The initial Reynolds number of these runs was fixed at $1/\nu = 10,000$. The evolution then was typically that the system, after having lingered awhile, evolved into the dipolar configuration.

### 5.4 Conclusion

With a series of direct numerical simulations with a 2D Fourier pseudospectral code the quasi-stationary final states of several freely evolving 2D turbulent Navier-Stokes flows have been computed and compared with statistical-mechanical predictions based on the “point” and on the “patch” theory. Three different situations have been considered. The possible confirmation of predicted maximum entropy states, the emergence of (intermediate) local maximum entropy states, and finally we have briefly discussed several “unclassifiable” quasi-stationary final states.

The dynamical computations confirm the theoretical predictions made in Chapter 3 of this thesis. In particular, the observation of the “bar” quasi-stationary final state instead of the classical dipolar final state, and the subsequent investigation of the role of the patch size of vorticity on the emergence of the “bar” and “dipole” state (see the numerical experiments with the slightly reduced patch size and repositioning of the patch in order to induce symmetry breaking), provides in our view strong support for

\textsuperscript{2}See also the discussion in Section 1.2.
Figure 5.23: A second run, whose late-time state is outside the patch/point classifications, that seems to have reached some metastable late-time state. Here, a large area of zero vorticity was created in the initial conditions by removing, asymmetrically, four pieces of a 16-pole patch solution.
Figure 5.24: The $\omega - \psi$ scatter plot of the late-time state achieved in the run shown in Figs. 5.23. It suggests two independent co-existing sinh-Poisson states which have developed asymmetrically.

Figure 5.25: Time evolution of the global quantities for the run shown in Figs. 5.23. Note that the mean wave number never reaches above 2.5.
5.4. Conclusion

these statistical-mechanical approaches.

The numerical investigation revealed that, besides the “bar” and “dipole” final states, also other solutions of the Poisson-like equations might appear as intermediate or as quasi-stationary final states. This are the so-called local maximum entropy states that have a lower entropy than the “bar” or the “dipole” solution. Observation of these local maximum entropy states as a quasi-stationary final state strongly indicates the depletion of nonlinear interactions in the flow. Applying these quasi-stationary final states, added with sufficient random noise, as initial condition for new numerical simulations will quite often show a rapid relaxation towards the maximum entropy state.

We have also found strong indications that some limitations exist of the application of statistical-mechanical theories. We have illustrated this by emphasizing a few numerical runs of 2D turbulent flows where quasi-stationary final states are obtained which cannot easily be explained in terms of these theories. Further investigation in this direction seems worthwhile, at lower viscosity and for longer durations.

Nevertheless, we might conclude that the theoretical results, summarized in Figs. 3.7 and 3.8, combined with the numerical simulations of 2D Navier-Stokes turbulent flows proved to be very successful for predicting the quasi-stationary final states. In particular, these simulations showed the predictive capacity of statistical-mechanical theories of 2D Euler flows: without the prediction of the “bar” quasi-stationary final state it is highly doubtful that such final states could have been identified in 2D direct numerical simulations.
Chapter 5. Dynamical solutions and comparisons
Chapter 6

Conclusions and recommendations

We have set out to test the relevance of the predictions of Eqs. (3.5) and (3.6) and their respective vorticity discretizations to the numerically-determined, long-time states of a 2D NS fluid with Reynolds numbers of several thousand subject to doubly-periodic boundary conditions. Eq. (3.5) has been derived by modeling the vorticity distribution as a mean-field limit of equal delta-function point vortices, while Eq. (3.6) models the vorticity distribution as made up of flat mutually exclusive patches of an area whose choice is arbitrary. Both equations have an infinite number of solutions, characterized by different topologies, but for fixed vorticity fluxes and total energy, all but one are only local maxima and there seems to be always one uniquely defined maximum-entropy prediction. As seen in Figs. 3.7 through 3.8, it is sometimes a matter of painstaking analysis to determine which state this is, however. We did computational runs of two basic types: runs with broad-band, totally disordered initial conditions of the type previously investigated, and runs originating from vorticity distributions that consisted of large areas of nearly flat vorticity patches plus random noise to break possible unwanted symmetries. In the former case, there were no surprises: the previously found dipolar late-time states inevitably resulted. This was not the case for the second kind of initial conditions, however. They did not generate broad-band turbulence in which energy was shared widely among many Fourier modes, but did exhibit considerable nonlinear activity in which the energy remained primarily in the lower parts of Fourier space. It is perhaps a semantic quibble as to whether the evolution should be called fully turbulent. This second class of initial conditions, in any case, exhibited a more diverse range of possible behaviour than the broad-band initializations did, and often came close to a late-time state that could be identified as one of the solutions of Eq. (3.6). Most interesting of those was the one-dimensional “bar” state exemplified by Figs. 5.5, 5.10, 5.12 and 5.14. As Figs. 3.7 through 3.8 illustrate, it may or may not be considered to be the most-probable patch state, depending upon the choice of the patch size $\Delta/M$, which seems to be arbitrary. In any case, it is not the most probable point state predicted from Eq. (3.5). In the finite-size patch theory leading to Eq. (3.6), the arguments for it proceed from “coarse graining” the Euler equation behaviour and presuppose a minimum observational scale below which fine spatial structure cannot be resolved. In a well-resolved NS computation (presumably including
all of the ones reported here), there is no such scale, and observation of all scales participating is feasible. Thus that the patch formulation of Eq. (3.6), with a large enough value of $\Delta$, can predict a radically different topology than it does with a smaller $\Delta$, and then substantiate that prediction in a dissipative NS computation, is another of the accumulating puzzles which remain to be deciphered. In our opinion, there can be said to be an interesting 2D NS regime to be explored that has opened up in these investigations in which nonlinear evolution, perhaps not fully turbulent in the conventional sense, nevertheless leads from one state that can be identified to another laminar, late-time state that is quantitatively more probable than its initial conditions: an essentially thermodynamic behaviour. The conditions for assigning this probability remain incompletely defined, and seem to us worthy of further numerical investigation.

The general question of whether there is a sharp sense in which Euler equation dynamics can satisfactorily mimic turbulent Navier-Stokes decays must, in our opinion, remain open. It seems beyond question, however, that entropies defined in terms of ideal fluid models have some predictive power for high Reynolds number decays that go beyond coincidence. It remains to be determined what precisely the limitations, for this purpose, of the two entropies considered here may be.

The time-consuming numerical simulations of 2D decaying Navier-Stokes turbulence motivated us to parallelize the Fourier pseudospectral code. Due to the relative small arrays representing the vorticity field and the stream function in either physical or in spectral (Fourier) space, the main challenge in parallelizing a 2D Fourier pseudospectral scheme is how to reduce the total amount of messages passing which is needed to exchange information between the different processors. Parallelization of our code is based on the task distribution method, which is very easy to implement. Later investigations and comparisons with former parallelization techniques reported in the literature make us believe that in the field of 2D DNS, the task distribution method will be the first choice for constructing parallel schemes, and likely it will be the only efficient way.

With the efficiently parallelized Fourier pseudospectral Navier-Stokes solver, it is possible to carry out a complete investigation of the late-time predictions, obtained from statistical-mechanical theories, by DNS of decaying high-Reynolds number turbulence. Especially in Chapter 5, results of several simulations have been discussed which were based on predictions from these statistical-mechanical theories, and such an investigation was not feasible without an efficiently parallelized Navier-Stokes solver. In my view, the results of those simulations inversely give deeper thoughts on the application of statistical mechanics to high-Reynolds number Navier-Stokes turbulence, and surely will trigger further numerical investigations.
Appendix A

Direct solution method of the linearized three-level Poisson equation

In this appendix a brief outline is provided of a direct solution method for the linearized version of the three-level Poisson equation. The procedure is a compact version of the method introduced previously by McDonald [33].

The first step of McDonald’s procedure is to obtain the nontrivial solution for the three-level Poisson equation (3.12) with \( \Psi = 0 \) on a square boundary \([0,1] \times [0,1]\). Starting from a trial solution \( W(x, y) \), we get the solution of \( v(x, y) \) from

\[
\frac{\partial^2 v(x,y)}{\partial x^2} + \frac{\partial^2 v(x,y)}{\partial y^2} + q(x,y) v(x,y) = s(x,y)
\]

subject to \( v = 0 \) on the boundary. Comparing this expression with Eq. (3.20) enables the following identification: \( s(x,y) = (W,W)R(x,y) \) and \( q(x,y) = f'(W) \), with \( (W,W), f'(W) \) and \( R \) defined in Eqs. (3.21)-(3.23). This partial differential equation can be represented on a discrete rectangular grid, with \((N_X + 1) \times (N_Y + 1)\) grid points \((x_i,y_j) = (i/N_X, j/N_Y) (0 \leq i \leq N_X \text{ and } 0 \leq j \leq N_Y)\), by a five-point difference equation of the form:

\[
\frac{v(x_{i+1},y_j) - 2v(x_i,y_j) + v(x_{i-1},y_j)}{(\Delta x)^2} + \frac{v(x_i,y_{j+1}) - 2v(x_i,y_j) + v(x_i,y_{j-1})}{(\Delta y)^2} + q(x_i,y_j)v(x_i,y_j) = s(x_i,y_j),
\]

with \( \Delta x = 1/N_X \) and \( \Delta y = 1/N_Y \), for \( 1 \leq i \leq N_X - 1 \text{ and } 1 \leq j \leq N_Y - 1 \), and \( v(x_i,y_j) = 0 \) on the boundary. From now on we will abbreviate \( v(x_i,y_j) \) by \( v(i,j) \) when it indicates a matrix element. We can see that if we know the values of \( v \) on the columns \( j = 0 \) and \( j = 1 \) in the beginning of the solution process, we can obtain the solution of the whole field by a forward sweep: get the values of the column \( j = 2 \) with the columns \( j = 0 \) and \( j = 1 \), then get \( j = 3 \) with \( j = 1 \) and \( j = 2 \), and so on.

So the essential part of this method is to find the \( N_X - 1 \) unknown values on the column \( j = 1 \) (we already know that \( v(x_0,y_1) = 0 \) and \( v(x_N,y_1) = 0 \)). To find these,
we calculate a set of “basis functions” \( v_k \), linearly independent solutions to Eq. (A.2) to satisfy \( v_k = 0 \) on the three boundaries \( x_0, y_0 \) and \( x_{N_X} \).

What we do is first define \( v_0 \) from

\[
\begin{align*}
v_0(i, 0) &= 0, \\
v_0(i, 1) &= 0,
\end{align*}
\]

(A.3)

sweeping the solution across the grid using Eq. (A.2). Then we calculate the other \( v_k \), \( 1 \leq k \leq N_X - 1 \), from

\[
\begin{align*}
v_k(i, 0) &= 0, \\
v_k(i, 1) &= \delta_{ik},
\end{align*}
\]

(A.4)

here

\[
\delta_{ik} = \begin{cases} 
0 & \text{if } i \neq k, \\
1 & \text{if } i = k.
\end{cases}
\]

The solutions initiated by Eq. (A.4) are swept forward with \( s(i, j) = 0 \). The desired solution to Eq. (A.2) is the linear combination

\[
v(i, j) = v_0(i, j) + \sum_{k=1}^{N_X-1} c_k v_k(i, j),
\]

(A.5)

with \( v(i, N_Y) = 0, \forall i \). The coefficients \( c_k \) are determined by solving the following equations:

\[
\sum_{k=1}^{N_X-1} c_k v_k(i, N_Y) = -v_0(i, N_Y).
\]

(A.6)

Once the values of \( c_k \) are known, we can sweep Eq. (A.2) and get \( v(i, j) \) by

\[
\begin{align*}
v_k(i, 0) &= 0, \\
v_k(i, 1) &= c_i, \quad 1 \leq i \leq N_X - 1.
\end{align*}
\]

(A.7)

The propagation of roundoff errors during a forward sweep is increased very rapidly from column-to-column in the \( y \)-direction. If we use double precision arithmetic in our calculation, the initial roundoff error of \( 10^{-16} \) will go to \( 10^{-5} \) in a computation on a \( 15 \times 15 \) grid. This can cause severe problems on the accuracy. By using quadruple precision in the calculations the accuracy of this method can be increased considerably (a roundoff error of \( 10^{-7} \) is then found for a computation on a \( 33 \times 33 \) grid).
Appendix B

A “smart” solver for Poisson-like equations

The main structure of a “smart” solver of the Poisson-like equation:

$$\nabla^2 \Psi = -\frac{\lambda^2 \sinh \Psi}{g + \cosh \Psi}, \quad (B.1)$$

will be described in this appendix. The goal of developing this code is to let the computer find the final solution of Poisson-like equation itself after a limited number of manual interruptions.

B.1 The strategy of the “smart” solver

Before describing how to solve Eq. (B.1) automatically, it is quite necessary to know how to solve the equation manually. There are some features which will make the differences when we do it manually and automatically. For the automatical solver, there will be a problem if the code does not follow the middle line in Fig. 3.6. If the code follows the upper line, there will be an emergent stop because of the numerical overflow. If the code follows the lower line (in which case the trivial solution $\Psi = 0$ will be obtained), it has to be distinguished from the nontrivial solution. Although it is possible to handle these two problems manually (simply restart the solver), it will be rather difficult to handle this properly by an automatic code. Therefore, these differences have to be kept in mind when writing the automatic solver (Fig. B.1):

1. Start with a trial function, use the procedure in Chapter 3 (Eqs. 3.20 - 3.24), see if it leads to an accurate solution of Eq. (3.12).

2. If not, the value of the $\lambda^2$ has to be changed: to avoid the numerical shutdown, this has to be done in a very small step at a time (here we call it $\Delta$).

3. We start first with a relatively large $\Delta$, then change the value of $\lambda^2$ with $\Delta$. After some steps, it will not be possible to get more accurate solution any longer with the initial relatively large $\Delta$. A new value of $\Delta$ (in our code, the old $\Delta$ is simply
divided by 10 to make it smaller) should be used to modify the value of \( \lambda^2 \). This step should be repeated until an accurate solution is reached.

4. The “smart” solver should also be able to decide when the \( \Delta \) should be made smaller. We always let \( \lambda^2 + \Delta \) to update the value of \( \lambda^2 \) the first time we get a new \( \Delta \). Maybe this is not the most efficient way, but it is possible to make the logical structure simpler (see the dashed rectangle in Fig. B.1).

5. After the accurate resolution is reached, the code should be able to remember the value of \( \lambda^2 \), which is changed constantly, for later calculation of the E-S plot (see section 3.3).

In practice, the first few steps should be done manually to let the code get a chance to warm-up. This is due to the high possibility that there will be a numerical shutdown because we did not choose the proper initial value of \( \lambda^2 \) or the initial value of \( \Delta \) is too large. After the value of \( \lambda^2 \) has been changed for several times (in the meantime, the code keeps the history of those changes), the logic structure described in Fig. B.1 should be able to make its own judgment.

At the beginning of each loop (see the black boundary rectangle in Fig. B.1), one of the following decisions can be made by the code:

1. If the solution is accurate enough, the code should be stopped;

2. making the value of \( \lambda^2 \) larger by using the value of \( \lambda^2 + \Delta \) to update the value of \( \lambda^2 \): \( \lambda^2 \leftarrow \lambda^2 + \Delta \);

3. making the value of \( \lambda^2 \) smaller: \( \lambda^2 \leftarrow \lambda^2 - \Delta \);

4. making the value of \( \Delta \) smaller: \( \Delta \leftarrow \frac{1}{10}\Delta \).

As it will be seen, the judgement of changing the value of \( \lambda^2 \) can be made on the actions taken in the last three steps. The possible actions are thus:

1. \( \lambda^2 \leftarrow \lambda^2 + \Delta \);

2. \( \lambda^2 \leftarrow \lambda^2 - \Delta \);

3. \( \Delta \leftarrow \frac{1}{10}\Delta \).

### B.2 The instruction of reading the flowchart

The following description gives the instruction of how to read Fig. B.1:

1. The first step is to choose a trial function, \( \lambda^2 \) and \( \Delta \) [box(1)], then the final result of this trial function will be put into the Eq. B.1 to evaluate the accuracy [box(2)]. If the error is small enough (here we use \( 10^{-7} \)), then the code will output the results and the code stops [box(3)].
Figure B.1: The automatic procedure to get the final solution.
2. If the error is too large, then the value of \( \lambda^2 \) will be updated by \( \lambda^2 + \Delta \) [box(4)], after which a test will be carried out again to check whether or not the error becomes smaller than before the value of \( \lambda^2 \) was changed [box(5)].

3. If the error does become smaller, then the code will carry out a test to know whether the error is already small enough [box(6)]. If yes, then the results will be output and the code stopped [box(3)]. If not, then we should realize that the action we made to change the value of \( \lambda^2 \) is in the good direction, but not good enough. The code should follow this good direction [box(7)]. Here are two possibilities: we make error smaller by replacing the value of \( \lambda^2 \) with the value of \( \lambda^2 + \Delta \) or \( \lambda^2 - \Delta \). (It is possible that the procedure in the dashed rectangle in Fig. B.1 leads to \( \lambda^2 \leftarrow \lambda^2 - \Delta \) [box(13)], which will be discussed later.) Whatever we did last time to modify \( \lambda^2 \), we should let the same action continue until the error does not become smaller or an accurate solution is reached.

4. If the error does not become smaller, then the code will go into the dashed rectangle in Fig. B.1.

The first task of this part of code is to find out what leads to the worse result, namely, how the value of \( \lambda \) is modified, by plus or minus \( \Delta \) [box(8)]. It should be addressed that we always let \( \lambda^2 \leftarrow \lambda^2 + \Delta \) when a new value of \( \Delta \) is obtained. This is the reason why it leads to the different procedure whether the last step is \( \lambda^2 \leftarrow \lambda^2 - \Delta \) or \( \lambda^2 \leftarrow \lambda^2 + \Delta \) (which will be discussed in the end of this section):

(a) If it is \( \lambda^2 \leftarrow \lambda^2 - \Delta \), then the value of \( \lambda^2 \) will be replaced by \( \lambda^2 + \Delta \)- a value which is tested before [box(10)]. At this stage, it is not possible to get better results by modify the \( \lambda^2 \) with the same \( \Delta \), the value of \( \Delta \) has to be smaller [box(12)] and then the code jumps to step 2 to start another iteration [box(4)].

(b) If it is \( \lambda^2 \leftarrow \lambda^2 + \Delta \), then the value of \( \lambda^2 \) will be replaced by \( \lambda^2 - \Delta \)-a value which is tested before [box(9)]. For the new value of \( \lambda^2 \), we should recall whether the value of \( \lambda^2 - \Delta \) has been tested [box(11)]:

i. If yes, it is not possible to get better results by modifying the \( \lambda^2 \) with the same \( \Delta \), the value of \( \Delta \) has to be made smaller [box(12)] and then the code will jump to step 2 to start another iteration [box(4)].

ii. If not, then \( \lambda^2 \) should be updated by \( \lambda^2 - \Delta \) [box(13)], after which the accuracy of the results should be tested [box(5)].

There are three possibilities when a new \( \Delta \) is obtained:

1. This new \( \Delta \) is too large. \( \lambda^2 \leftarrow \lambda^2 + \Delta \) or \( \lambda^2 \leftarrow \lambda^2 - \Delta \) will always lead to worse results.

For this situation, the code will do:

(a) \( \lambda^2 \leftarrow \lambda^2 + \Delta \), and find the situation getting worse.
(b) $\lambda^2 \leftarrow \lambda^2 - \Delta$ - go back to the original situation.
(c) $\lambda^2 \leftarrow \lambda^2 - \Delta$, and find the situation getting worse, too.
(d) $\lambda^2 \leftarrow \lambda^2 + \Delta$ - go back to the original situation, and make up its mind to
make the value of $\Delta$ even smaller.

2. This new $\Delta$ can and only can be used $n$ ($n \geq 1$) times to make the $\lambda$ larger to
get better results.

For this situation, the code will do:

(a) $\lambda^2 \leftarrow \lambda^2 + \Delta$, and find the situation getting better\(^1\).
   Repeat this situation $n$ times.
(b) $\lambda^2 \leftarrow \lambda^2 + \Delta$, and find the situation getting worse.
(c) $\lambda^2 \leftarrow \lambda^2 - \Delta$, and make up its mind to make the value of $\Delta$ even smaller.

3. This new $\Delta$ can and only can be used $n$ ($n \geq 1$) times to make the $\lambda$ smaller to
get better results.

For this situation, the code will do:

(a) $\lambda^2 \leftarrow \lambda^2 + \Delta$, and find the situation getting worse.
(b) $\lambda^2 \leftarrow \lambda^2 - \Delta$ - go back to the original situation.
(c) $\lambda^2 \leftarrow \lambda^2 - \Delta$, and find the situation getting better.
   Repeat this situation $n$ times.
(d) $\lambda^2 \leftarrow \lambda^2 - \Delta$, and find the situation getting worse.
(e) $\lambda^2 \leftarrow \lambda^2 + \Delta$, and make up its mind to make the value of $\Delta$ even smaller.

It can be seen from above that because we always let $\lambda^2 \leftarrow \lambda^2 + \Delta$ for a new $\Delta$, it
is possible to make the logic structure suitable for programming.

**B.3 Conclusion**

A “smart” code is developed to solve the Poisson-like equation. The work concerned
with the zooming-in process in section 3.3.2 will be significantly reduced, because we
can start several tests simultaneously (with different values of $g$).

\(^1\)Whenever the situation gets better, a test should be carried out to know whether or not the result
is already accurate enough.
Appendix B. A “smart” solver for Poisson-like equations
Bibliography


BIBLIOGRAPHY


Summary

It is well known that two-dimensional decaying turbulent flows at high Reynolds numbers evolve to a state that can be accurately predicted by a maximum entropy analysis conserving both the kinetic energy of the flow and the vorticity flux. The most significant phenomenon observed is the formation of a few large vortices, the details of which depend upon the initial conditions and the boundary conditions. Defining the entropy of turbulent flows requires a discretization of the vorticity field, followed by a passage to a mean-field limit once the combinatorics is accomplished. The original discretization was in terms of line vortices (or “points” in two-dimensional flows) and corresponded to Boltzmann statistics. The discretization in terms of finite-area, mutually-exclusive vortices (denoted by “patches”) has been proposed recently, and corresponds to Lynden-Bell statistics. We have examined in detail a variety of high Reynolds number, two-dimensional flow evolutions with an eye to determining what differences exist between the two formulations and how important they are. The numerical method is based on a pseudospectral Fourier computation in a periodic domain. We concentrate on initial conditions that should lead to outcomes as divergent as possible for the “point” and the “patch” description.

We did computational runs of two basic types: runs with broad-band, totally disordered initial conditions of the type previously investigated, and runs originating from vorticity distributions that consisted of large areas of nearly flat vorticity patches plus random noise to break possible unwanted symmetries. In the former case, there were no surprises: the previously found dipolar late-time states inevitably resulted. This was not the case for the second kind of initial conditions, however. They did not generate broad-band turbulence in which energy was shared widely among many Fourier modes, but did exhibit considerable nonlinear activity in which the energy remained primarily in the lower parts of Fourier space. This second class of initial conditions, in any case, exhibited a more diverse range of possible behaviour than the broad-band initializations did, and often came close to a late-time state that could be identified as one of the solutions of the “patch” formulation. Most interesting of those was the one-dimensional “bar” state.

Long-time numerical integrations are required to validate the predictive capability of statistical-mechanical methods. This motivated us to use parallel computing. A task-based parallel approach has been adopted to pseudospectral simulations of the unsteady incompressible two-dimensional Navier-Stokes equations in a square domain (with periodic boundary conditions), rather than the usual space-based approach. The latter technique is actually similar to a domain decomposition method to evaluate
the FFTs. It is identified that several of the required FFTs, necessary to evaluate the nonlinear contribution to the Navier-Stokes equations, can be computed independently and therefore in parallel. The apparent advantage is that this approach reduces the number of relatively expensive communications between processors - something that is of paramount importance in the 2D (as opposed to the 3D) case because of the limited amount of work available to amortize communication overhead. While the speed-up potential is limited to just a few processors, it is nonetheless achievable, relatively easy to exploit, and useful for the application at hand. The basic timing results to illustrate the potential is provided, and we think this approach could be combined with conventional (transposed-based) approaches to multi-dimensional FFTs to obtain more parallelism than possible with the transposed-based approach alone.

With the fast parallelized Navier-Stokes solver, the statistical-mechanical predictions of the late-time states of decaying two-dimensional turbulent flows is tested in different ways. The prediction of the bar final states (the vorticity or stream function is independent of one of the coordinate directions) is tested here for the first time. Some of the earlier simulations provided good tests of the vortices predicted by the theory, but were not able to reach the bar final state. Some other earlier simulation concluded that propagating waves can be obtained as final states, in apparent contradiction with the statistical-mechanical theory. The present computations show that such propagating waves are only transitory states. Additionally, we also found and briefly discuss some novel features of the final states of 2D Navier-Stokes turbulence, which cannot be well explained by the statistical-mechanical theory for ideal fluids, but trigger the possibility of further investigations in this field.
Acknowledgments

I would like to give my thanks to Dave, this thesis would not be finished without his continuing help. His constant encouragement and his passion to science helped me to survive the first two years in the Netherlands when most of the time was devoted to deepening my insights in statistical mechanics of ideal fluids and to devise solution methods for the sinh-Poisson and tanh-Poisson equations. Although we exchanged emails quite intensively, we discussed a lot during his visits of the Fluid Dynamics Laboratory as a J.M. Burgers professor and during my visit to Dartmouth, USA, I felt it was still a too short time altogether for intensive scientific discussions. However, I enjoyed the time when I talked with him— which mostly happened during the lunchtime: we have history lunch, culture lunch, film lunch, politics lunch, seafood lunch and science lunch.

Thanks to Herman, who welcomed me in the airport when I touched the soil of a foreign country for the first time, and helped me to pass the first few lonely months. He also provided very good assistance to Dave to supervise my research. Although most of the scientific activities between us happened in our offices after we made appointments, sometimes we discussed our research together in a bar with a few beers or on the train.

I would also like to give my thanks to Gert Jan. He provided me the quietest corner in this world to do research with all the necessary scientific facilities. I felt strong support from his direction especially in the last two years of my research. He also gave me good support to let me have a chance to build up my scientific knowledge.

Thanks to Brigitte, Slava, Tanya, Anita, Robin, Adrian, Ralph, Dennis, Mathew, Maurice, David, Werner, Luis, Michel, Saskia, Marcel, Ad, Freek, Gert, etc.—the list can last until all the members in this lab are included. I would like to thank them to help me in almost all aspects of my life in Holland. Especially I thank Marjan who helped me with lots of social life problems, and Ruben who helped me with many scientific problems.

Thanks to the friends in the Chinese students union in TU/Eindhoven and Dartmouth, especially Mingwen Tian, Zhaohui Wang, Yiheng Zhang and Rui Feng.

For my research life, I would like to take this opportunity to thank:

Mr. A. Tal, Dr. X. Ke, Dr. S. A. Schwarz for their support to use the supercomputer;
Dr. R.M. Schoemaker, Dr. Jian Chen for introducing me in the world of parallel computing;
Dr. B. T. Kress for his help to make my first MPI code;
Dr. Haixiang Lin, Dr. A.H. Nielsen for useful discussion on parallel schemes;
Prof. W.H. Matthaeus for providing his pseudospectral code;
Dr. B.N. Kuvshinov for useful discussions on the Poisson-solver;
Thanks to all the teachers in the Burgers Center’s courses.
Final remark should be given to Prof. Jianping Wang – my former supervisor in China. The training he gave me helped me to survive research life abroad. I also benefit from him by some useful discussions during last four years.
Curriculum Vitae

Zhaohua Yin was born in Jiaozhou, Shandong (P.R. China) on 15 August 1973. From 1979 to 1991, he received pre-university education in Jiaozhou.

In September 1991, he went to Beijing to study mechanics at Peking University (Beida) and graduated in July 1996 with a thesis on the numerical simulation of non-linear water waves using ALE and finite-element methods, under the supervision of Prof. Jiang-hang Wu.

From 1996 to 1999, he continued to stay in Beida to carry out his CFD master research. His master supervisor is Prof. dr. Jian-Ping Wang, and the thesis concerned the numerical simulation of incompressible fluid using finite-spectral methods.

In 1999, he moved to the Netherlands where he was appointed trainee research assistant (AOI) to the Vortex Dynamics Group of Eindhoven University of Technology. This doctoral thesis is the result of the four-year period in Eindhoven, under the supervision of Prof. dr. D.C. Montgomery.
个人简历

尹兆华
(1973 年 8 月生于山东青岛)

北京大学计算机科学技术系软件专业 (1993 年 9 月-1996 年 7 月: 辅修) 
河南信阳陆军学院 (1991 年 9 月-1992 年 7 月) 
论文名称: 用有限元方法数值模拟非线性水波 
导 师: 吴江航教授 (北京大学)

硕士学位: 北京大学力学与工程科学系流体力学专业, 湍流国家重点实验室 
(1996 年 9 月-1999 年 7 月) 
研究方向: 计算流体力学(网格生成, 多重网格方法, 有限差分方法) 
论文名称: 用有限谱法数值模拟不可压缩流动 
导 师: 王建平教授 (北京大学, 日本四日市大学)

博士学位: 荷兰艾因德霍温大学应用物理系流体力学实验室 
(1999 年 10 月-2003 年 9 月) 
研究方向: 二维湍流统计力学, 湍流直接数值模拟, 并行计算 
论文名称: On the statistical-mechanical description of decaying 2D turbulence 
导 师: David. C. Montgomery 教授 (美国达特茅茨学院)

访问学者: 美国达特茅茨学院物理与天文学系 (2000 年 6-8 月; 2001 年 9-12 月) 
合作人: David. C. Montgomery 教授