Singularity Formation in the High-Dimensional Euler Equations and Sampling of High-Dimensional Distributions by Deep Generative Networks

Thesis by
Shumao Zhang

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2023
Defended August 23, 2022

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my deepest gratitude to my advisor, Prof. Thomas Yizhao Hou, for his unintermittent guidance and patience during my Ph.D. study at Caltech. The encouragement and inspiration I received from him helped me through the tough time in the past years, especially when I got stuck or lost. I feel extremely and increasingly fortunate to work closely with him, not only because of the interesting research projects, but also because of his attitude towards career and life. His unceasing enthusiasm and diligence on important and impactful problems deeply touches me and will keep influencing me for many years to come. When setback came, he taught me to spare no effort to attack the problem, but remain peaceful for the outcome. I would not forget the lessons I learned from him.

I want to thank Prof. Daniel I. Meiron, Prof. Kaushik Bhattacharya, and Prof. Houman Owhadi for serving on my thesis committee and their valuable suggestions. In addition, I would also like to thank Prof. Andrew Stuart for inspiring discussions and the kind help. Your valuable comments benefit my research a lot.

I would like to give special thanks to Pengchuan Zhang, who helped me set up my research when I started my PhD study, and guided me to the intersections of deep learning and scientific computing. I am very grateful for his generous sharing of his knowledge and insight. I would also like to thank a lot of people in the CMS department for inspiring discussions and casual chats, including Ka Chun Lam, De Huang, Ameera Abdelaziz, Dimitry Burov, Jiajie Chen, Victor Dorobantu, Spencer Gordon, Andrew Taylor, Yifan Chen, Ziyun Zhang, and Zhenzhen Li. I want to thank the warmhearted CMS staffs Maria Lopez, Jolene Brink, Sydney Garstang and Diana Bohler for their great help over the years, and especially the preparation of my thesis defense.

Spending my graduate life at Caltech is a unique and joyful experience. I am very delighted and grateful to get accompanied by many friends at campus and all around the world, including Kun Miao, Xinhong Chen, Zhiyang Jin, Yishen Zhang, Enrui Zhang, Xingfan Huang, Peiwei Chen, Yu-Wei Wu, Willam Kuo, Allen Shang, Yuelin Shi, Xin Su, Menghan Liu, Yukai Liu, Botao Hu, Zhihao Cui, Chen Li, Sijia Dong, Shibo Shu, Chang Liu, Yumeng Ma, Sijie Chen, Lingfei Yi and many many more. Your generous help and support are very precious to me. And the good time we spent together is definitely part of the best memory of my graduate life.

It is rather challenging to survive the strenuous Ph.D. journey in the time of the COVID-19 pandemic. However, I am extremely lucky to meet and fall in love with my dear girlfriend Xirui Wang during that special time. Despite the long distance between Los Angeles and Boston, despite the difference in fields we work on, we fight alongside in work and life. Your courage and wisdom support me through the dark times, and your company brings endless happiness to my life.

Finally yet most importantly, I would like to dedicate this thesis to my beloved parents, Yi Zhang and Yan Tong. You raise me up with the best environment I can imagine, and act as my role model in many many ways. Words cannot carry my gratitude and love for you. You are the best parents!

ABSTRACT

High dimensionality brings both opportunities and challenges to the study of applied mathematics. This thesis consists of two parts. The first part explores the singularity formation of the axisymmetric incompressible Euler equations with no swirl in $\mathbb{R}^n$, which is closely related to the Millennium Prize Problem on the global singularity of the Navier-Stokes equations. In this part, the high dimensionality contributes to the singularity formation in finite time by enhancing the strength of the vortex stretching term. The second part focuses on sampling from a high-dimensional distribution using deep generative networks, which has wide applications in the Bayesian inverse problem and the image synthesis task. The high dimensionality in this part becomes a significant challenge to the numerical algorithms, known as the curse of dimensionality.

In the first part of this thesis, we consider the singularity formation in two scenarios. In the first scenario, for the axisymmetric Euler equations with no swirl, we consider the case when the initial condition for the angular vorticity is C^α Hölder continuous. We provide convincing numerical examples where the solutions develop potential self-similar blow-up in finite time when the Hölder exponent $\alpha < \alpha^*$, and this upper bound α^* can asymptotically approach $1 - \frac{2}{n}$. This result supports a conjecture from Drivas and Elgindi [27], and generalizes it to the high-dimensional case. This potential blow-up is insensitive to the perturbation of initial data. Based on assumptions summarized from numerical experiments, we study a limiting case of the Euler equations, and obtain $\alpha^* = 1 - \frac{2}{n}$ which agrees with the numerical result. For the general case, we propose a relatively simple one-dimensional model and numerically verify its approximation to the Euler equations. This one-dimensional model might suggest a possible way to show this finite-time blow-up scenario analytically. Compared to the first proved blow-up result of the 3D axisymmetric Euler equations with no swirl and Hölder continuous initial data by Elgindi in [30], our potential blow-up scenario has completely different scaling behavior and regularity of the initial condition. In the second scenario, we consider using smooth initial data, but modify the Euler equations by adding a factor ε as the coefficient of the convection terms to weaken the convection effect. The new model is called the weak convection model. We provide convincing numerical examples of the weak convection model where the solutions develop potential self-similar blow-up in finite time when the convection strength $\varepsilon < \varepsilon^*$, and this upper bound ε^* should be close to $1 - \frac{2}{n}$. This result is closely related to the infinite-dimensional case of an open question [27] stated by Drivas and Elgindi. Our numerical observations also inspire us to approximate the weak convection model with a one-dimensional model. We give a rigorous proof that the one-dimensional model will develop finite-time blow-up if $\varepsilon < 1 - \frac{2}{n}$, and study the approximation quality of the one-dimensional model to the weak convection model numerically, which could be beneficial to a rigorous proof of the potential finite-time blow-up.

In the second part of the thesis, we propose the Multiscale Invertible Generative Network (MsIGN) to sample from high-dimensional distributions by exploring the low-dimensional structure in the target distribution. The MsIGN models a transport map from a known reference distribution to the target distribution, and thus is very efficient in generating uncorrelated samples compared to MCMC-type methods. The MsIGN captures multiple modes in the target distribution by generating new samples hierarchically from a coarse scale to a fine scale with the help of a novel prior conditioning layer. The hierarchical structure of the MsIGN also allows training in a coarse-to-fine scale manner. The Jeffreys divergence is used as the objective function in training to avoid mode collapse. Importance sampling based on the prior conditioning layer is leveraged to estimate the Jeffreys divergence, which is intractable in previous deep generative networks. Numerically, when applied to two Bayesian inverse problems, the MsIGN clearly captures multiple modes in the high-dimensional posterior and approximates the posterior accurately, demonstrating its superior performance compared with previous methods. We also provide an ablation study to show the necessity of our proposed network architecture and training algorithm for the good numerical performance. Moreover, we also apply the MsIGN to the image synthesis task, where it achieves superior performance in terms of bits-per-dimension value over other flow-based generative models and yields very good interpretability of its neurons in intermediate layers.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Shumao Zhang, Pengchuan Zhang, and Thomas Y Hou. “Multiscale Invertible Generative Networks for High-Dimensional Bayesian Inference”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12632–12641.
S.Z. participated in the conception of the project, designed and implemented the algorithm, prepared the data, conducted the numerical simulation, and participated in the writing of the manuscript.

[2] Thomas Y Hou et al. “Solving Bayesian inverse problems from the perspective of deep generative networks”. In: *Computational Mechanics* 64.2 (2019), pp. 395–408.
S.Z. participated in the conception of the project, designed and implemented the algorithm, prepared the data, conducted the numerical simulation, and participated in the writing of the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Number *Page*

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Overview

High dimensionality brings both opportunities and challenges to the study of applied mathematics. This thesis consists of two parts. The first part explores the singularity formation of the axisymmetric incompressible Euler equations with no swirl in $\mathbb{R}^n$, which is closely related to the Millennium Prize Problem on the global singularity of the Navier-Stokes equations. In this part, the high dimensionality contributes to the singularity formation in finite time by enhancing the strength of the vortex stretching term. The second part focuses on sampling from a high-dimensional distribution using deep generative networks, which has wide applications in the Bayesian inverse problem and the image synthesis task. The high dimensionality in this part becomes a significant challenge to the numerical algorithms, known as the curse of dimensionality.

In the first part of this thesis, we consider the singularity formation in two scenarios: the axisymmetric Euler equations with no swirl and with Hölder continuous initial data, and a weak convection model of the axisymmetric Euler equations with no swirl and with smooth initial data. In both scenarios, we provide convincing numerical evidence of the potential finite-time blow-up in $\mathbb{R}^n$ that has not been studied before. The potential finite-time blow-up is computationally robust with respect to the perturbation of initial data, implying that the potential blow-up mechanism should be quite generic and insensitive to the initial data. We propose simplified models to understand the mechanism of the potential blow-up. Our numerical results also support several conjectures on the finite-time blow-up of the Euler equations as proposed in a recent survey paper [27].

In the second part of this thesis, we propose the Multiscale Invertible Generative Network to sample from a high-dimensional distribution. The Multiscale Invertible Generative Network generates samples by transporting a simple reference distribution to the target distribution. As a deep generative network, the Multiscale Invertible Generative Network can control its capacity and computational cost by the number of network parameters, thus making it quite scalable to the high-dimensional problems. By exploring the low-dimensional structure in the high-dimensional distribution, we achieve superior performance over other approaches on the tested examples in the Bayesian inverse problem and the image synthesis task, especially in distribution approximation and multiple mode capturing.

1.2 Singularity Formation in the Euler Equations

Intentionally redacted.

1.3 Sampling of High-Dimensional Distributions

In this part, we introduce the Multiscale Invertible Generative Network, which is abbreviated as the MsIGN, to sample from high-dimensional distributions. Sampling from a distribution provides convenient ways to access the information carried by the distribution, for example, mean, variance, and the expected value of any function of the random variable. When the dimension of the distribution is high, calculating an integral of the distribution becomes computationally infeasible, but using the Monte Carlo method with samples of the distribution is still efficient. However, sampling from a high-dimensional distribution is very challenging. The curse of dimensionality significantly slows down algorithms that work well for low-dimensional problems and spoils the quality of the samples.

The MsIGN is a deep generative network that maps samples from a simple reference distribution to the target distribution. It makes use of the multiscale structure that widely appears in many high-dimensional distributions in applications to design its network architecture. As a deep generative network, the MsIGN can control its capacity and computational cost by the number of network parameters, thus making it quite scalable to the high-dimensional problems. We use the MsIGN to solve two high-dimensional distribution sampling problems: the Bayesian inverse problem, whose applications widely appear in fluid dynamics, geophysics and medical imaging, and the image synthesis task, which is one of the core problems in machine learning.

The Bayesian Inverse Problem

The inference of a parameter of interest of a complicated system from limited and noisy observation is a far-reaching problem that has a wide range of applications, including various scenarios in geophysics, fluid dynamics, and materials science. A popular setting is when the noise is an additive Gaussian to the observation:

$$y = \mathcal{F}(x) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \Gamma), \tag{1.1}$$

where $x \in X$ is the parameter of interest, and we assume that $(X, \|\cdot\|_X)$ is a Banach space. Here $y \in \mathbb{R}^{d_y}$ is the finite-dimensional observation, $\varepsilon \in \mathbb{R}^{d_y}$ is the centered Gaussian observational noise, and its covariance Γ is a $d_y \times d_y$ positive definite matrix. $\mathcal{F}$ is referred to as the forward map that describes some underlying dynamics of the system. We define the data-misfit functional from (1.1) as

$$\Phi(x; y) = -\frac{1}{2}\|y - \mathcal{F}(x)\|_\Gamma^2, \tag{1.2}$$

where we introduce the notation $\|z\|_\Gamma^2 := z^T\Gamma^{-1}z$, for $z \in \mathbb{R}^{d_y}$.

The Bayesian approach provides a powerful framework to the "inversion" from the observation y to the parameter x that organically blends the prior knowledge with the observation matching. More specifically, the Bayesian inverse problem casts a posterior distribution ν^y on the parameter x by

$$\frac{\mathrm{d}\nu^y}{\mathrm{d}\mu}(x) = \frac{1}{Z(y)}\mathcal{L}(x; y), \tag{1.3}$$

with

$$\mathcal{L}(x; y) = \exp\left(-\Phi(x; y)\right) \tag{1.4}$$

where μ is the Borel prior probability measure on X, $\mathcal{L}(x; y)$ is the likelihood, and the normalizing constant $Z(y)$ is given by

$$Z(y) = \int \mathcal{L}(x; y)\mathrm{d}\mu(x). \tag{1.5}$$

The posterior (1.3) gives full characterization of all possible solutions to the inverse inference of x based on y in (1.1), and this framework is very convenient in modeling and quantification of uncertainty in the inference problem. We refer the reader to [88, 22] for more theoretical discussion about the Bayesian framework presented here.

When X is an infinite-dimensional Banach space, the practical treatment of the posterior ν^y requires discretization to a finite-dimensional space. This is typically the case when the parameter x is a function or a field. Following Section 4.1 of [42], we assume X admits an unconditional normalized Schauder basis, and project x to a finite number of them. Under proper assumptions on the prior μ in [42], the projected posterior is consistent with the original posterior defined in (1.3) in the sense of the Hellinger distance. More examples of the consistency of the projected posterior to the original posterior can be found in [18, 88, 21, 49, 89]. Therefore, we will let $X = \mathbb{R}^d$ from now on, based on the practical and simplicity consideration.

The posterior distribution ν^y in (1.3) can be also characterized by its density q^y, which is the Radon-Nikodym derivative $\mathrm{d}\nu^y/\mathrm{d}x$ to the Lebesgue measure $\mathrm{d}x$ on $\mathbb{R}^d$:

$$q^y(x) = \frac{1}{Z(y)}\rho(x)\mathcal{L}(x;y), \tag{1.6}$$

where ρ is the density function of the prior μ. We remark that the normalizing constant $Z(y)$ defined in (1.5) is often computationally intractable, due to the high dimensionality of x.

In the following, since the observation y in (1.1) only helps in defining the posterior distribution, but does not play an active role in our purposed method and analysis, we will write ν^y as ν in (1.3), and q^y as q in (1.6) to simplify the notation.

We target at generating samples from the posterior distribution defined in (1.6) given its unnormalized density function, which is a long-standing challenge especially when the dimension of x is high. Since samples help build the estimate of quantities like $\mathbb{E}_{x\sim\nu}[f(x)]$ for any measurable function f on $\mathbb{R}^d$ by the Monte Carlo method, the sample generation is of great importance in the Bayesian framework.

The Curse of Dimensionality

While the posterior (1.3) in Bayesian inverse problems is very informative, its samples are needed for building statistical quantification, like mean and variance, of the inverse inference of x based on y. However, when $x \in \mathbb{R}^d$ is high-dimensional, sampling the posterior ν becomes a long-standing challenge. For example, an arbitrary posterior can have its importance regions, also known as “modes”, anywhere in the high-dimensional space, and as a consequence there will be an exponential growth of computational cost with respect to the problem dimension, for example, see [58, 39].

To deal with the curse of dimensionality, various Markov Chain Monte Carlo (MCMC) algorithms [4, 5, 95, 75, 94, 19, 39, 16, 9, 26, 20] have been proposed to improve the convergence rate by designing favorable proposals. For example, the Langevin diffusion is leveraged to design a better proposal distribution with advantages like higher acceptance rate in [4, 5, 94, 20]. The Hamiltonian dynamics, due to its energy preserving property, is also utilized to improve acceptance rate and lower sample correlation in [75, 16]. By considering proposal distributions well-defined on the function space, [19, 20] designed MCMC samplers to be independent of the discretization of the function. In [32, 95], the tempering method is used to accelerate mixing for multimodal distributions. In [39, 26], the multi-level MCMC samples a telescopic expansion of the discretization error using multiple correlated MCMC chains at different levels. However, when it comes to high-dimensional problems, MCMC-type methods still face challenges in computational cost, algorithm tuning, sample correlation and mode collapse. For example, the Langevin diffusion tends to move the MCMC chain toward the high density region, which would easily lead to mode collapse in the high-dimensional case. For the MCMC samplers independent of the discretization, detecting modes away from the current state could also be difficult when the dimension is high, and thus might also suffer from the mode collapse. The multi-level MCMC usually needs a lot of uncorrelated samples from the coarse scale to run the MCMC chain in the fine scale, which would be time consuming for high-dimensional problems. And for the tempering method, the parameter tuning could be sensitive in order to control the computational cost and avoid mode collapse, especially when the dimension is high.

Concurrently, the sampling problem is framed into a deterministic optimization by variational inference, and numerous methods are based on different formulations of the optimization, including the Stein variational gradient descent (SVGD) [69] and its related methods [64, 10, 14, 13], and the transport map approach [17, 74, 29, 78, 85, 48, 7, 57]. Despite the better robustness in algorithm tuning and reduced sample correlation, these methods can still have a scalability issue or suffer from mode collapse in high-dimensional cases. We will give a more detailed description on the transport map approach in Section 5.1. We remark that [78, 85, 98, 14, 7, 13] invoked the low-dimensional structure in the likelihood, and showed good potential in overcoming the high-dimension challenge. We will give a detailed discussion on the comparison of the low-dimensional structure in our MsIGN and other literature in Section 5.2.

The Image Synthesis Task

The image synthesis task looks for new, unseen samples x from a target distribution q characterized by a data set of ground-truth example samples $\{x_i\}_{i=1}^N$, where x is an image stored as a matrix or tensor. The density function of the target distribution q is in general unknown. The dimension of the target distribution is determined by the resolution of the image x (for example, 64×64), and the number of color channels (for example, 3 for the RGB format of images). As a consequence, the problem dimension can easily go beyond 10^3 or 10^4.

The image synthesis task is one of the core problems in machine learning. As an example of unsupervised learning (because there are no labels in the data), the image synthesis task is an important tool to learn the realistic world model from a large amount of data and can be extended to other similar tasks like image inpainting, denoising, and colorization. Solutions to the image synthesis task can potentially lead to more robust and data-efficient ways to simulate interactions with the real world.

The image synthesis task gives a good show case on the model capacity of the MsIGN, or in other words, the richness of the parametric family of transport maps modeled by the MsIGN. Due to different types of objects appearing in the images, the distribution of images is naturally multi-modal, and therefore, the result on the image synthesis task can show the capacity of our method for very complicated and multimodal target distribution. Besides, by benchmarking with other recent flow-based generative models, we can also demonstrate the parameter efficiency of our MsIGN design, and show the interpretability of internal neurons of our MsIGN.

There has been an enormous amount of studies on the image synthesis task, especially in the recent decade. Most studies follow the approach of generative adversarial networks (GANs) [34] or likelihood-based methods. Among the likelihood-based methods, autoregressive models [40, 36, 92, 91] generate new images pixel by pixel by sampling from the conditional distribution on the existing pixels. However, due to the sequential sampling strategy, it becomes troublesome for the high-dimensional problems like high-resolution images. The variational autoencoders [52, 55, 51] directly capture the distribution of the whole image by optimizing a lower bound on the log-likelihood of the data. The indirect optimization on the lower bound of the objective makes the training of variational autoencoders relatively challenging. The diffusion models [83, 37, 84] employ a stochastic differential equation to diffuse the image distribution to random noise. For sample generation, they solve the reverse-time diffusion process to move random noises to images, which could be very time-consuming for high-dimensional problems. Another category of the likelihood-based methods is the flow-based generative models, like the NICE [23], the Real NVP [24], the Glow [54], and the MsIGN, which look for a bijective transport map between a simple reference distribution, which is also called the latent space, and the target distribution. Compared to the generative adversarial networks and variational autoencoders, the flow-based generative models allow density evaluation and are very efficient in latent-variable inference. As a bijective map, the representation of an image in the latent space can be simply obtained by the inverse of the map. Since the log determinant of the Jacobian of the map is also accessible for flow-based generative models, the density evaluation of images is also possible. Furthermore, the efficient latent-variable inference of the flow-based generative models favors downstream tasks on the latent space, like image manipulation and conditional image synthesis.

Summary of Our Results

We propose the Multiscale Invertible Generative Network (MsIGN) to sample from high-dimensional distributions by exploring the low-dimensional structure in the target distribution. The MsIGN models a transport map from a known reference distribution to the target distribution, and thus is very efficient in generating uncorrelated samples compared to MCMC-type methods. The MsIGN captures multiple modes in the target distribution by generating new samples hierarchically from a coarse scale to a fine scale with the help of a novel prior conditioning layer. The hierarchical structure of the MsIGN also allows training in a coarse-to-fine scale manner. The Jeffreys divergence is used as the objective function in training to avoid mode collapse. Importance sampling based on the prior conditioning layer is leveraged to estimate the Jeffreys divergence, which is intractable in previous deep generative networks. In our numerical experiments applied to two Bayesian inverse problems, our results show that the MsIGN clearly captures multiple modes in the high-dimensional posterior and approximates the posterior accurately, demonstrating its superior performance compared with previous methods. We also provide the ablation study to show the necessity of our proposed network architecture and training algorithm to the good numerical performance. Moreover, we apply the MsIGN to the image synthesis task, where it achieves superior performance in terms of bits-per-dimension value over other flow-based generative models and yields very good interpretability of its neurons in intermediate layers.

1.4 Roadmap of the Thesis

In Part I, we discuss the singularity formation of the Euler equations in Chapter 2, 3, and 4.

We first consider the 3D axisymmetric Euler equations with no swirl in Chapter 2, where we present detailed numerical evidence of the potential finite-time self-similar blow-up in Section 2.2, 2.3, and 2.4. Then we study two factors that influence the behavior of the potential blow-up: the Hölder exponent α in Section 2.5, and the stretching factor δ in z-axis in Section 2.6. Since recently Elgindi proved the first blow-up result in the 3D axisymmetric Euler equations with no swirl and with Hölder continuous initial data in [30], we make a comprehensive comparison between our scenario and his scenario in Section 2.7. We also study the robustness of the potential blow-up to the initial data in Section 2.8.

In Chapter 3, we extend our blow-up scenario in Chapter 2 to the high-dimensional case. To start with, we discuss the formulation of the n-D axisymmetric Euler equations with no swirl in Section 3.1, and present detailed numerical evidence of the potential finite-time self-similar blow-up in Section 3.2. Then in Section 3.3 we study the potential blow-up in different settings of the Hölder exponent α, the stretching factor δ, and the dimension n and summarize our results. In Section 3.4, a potential mechanism is proposed for the limiting case of $\delta \to 0$, and together with observations from numerical experiments, we derive the asymptotic behavior of the scaling factor c_l and the upper bound α^* for α that could develop singularity. Both of these results match our numerical results very well. In Section 3.5, we propose a relatively simple one-dimensional model that approximates the original equations pretty well, which could potentially benefit the analytical study of our scenario.

In Chapter 4, we propose the weak convection model. In Section 4.1 we discuss our motivation of the model, compare the model with previous models and study some properties of our model. In Section 4.2, we present detailed numerical evidence of the potential finite-time self-similar blow-up in the weak convection model. We summarize the influence of the convection strength ε and the dimension n on the potential blow-up in Section 4.3. And in Section 4.4, we propose a one-dimensional model, and study its approximation to the weak convection model numerically. We also give a rigorous proof of the finite-time blow-up in the one-dimensional model in Section 4.4.

In Part II, we discuss the sampling of high-dimensional distributions in Chapter 5.

In Section 5.1, we review several important concepts and recent studies in high-dimensional distribution sampling using deep generative networks. The motivation of the MsIGN is discussed in Section 5.2. In Section 5.3 and 5.4, we introduce the network architecture and training strategy of the MsIGN in order to solve the Bayesian inverse problem. Numerical results of the MsIGN on two Bayesian inverse problems are shown in Section 5.5, and the ablation study that verifies the necessity of our proposals is presented in Section 5.6. Then, we move on to the image synthesis task and discuss the network architecture and training strategy of the MsIGN in Section 5.7, and present our numerical results in Section 5.8. In Section

5.9, we provide discussion on interesting topics for future study.

Part I

Singularity Formation in the High-Dimensional Euler Equations

Chapter 2

SELF-SIMILAR FINITE-TIME SINGULARITY FORMATION FOR HÖLDER CONTINUOUS SOLUTIONS TO THE INCOMPRESSIBLE EULER EQUATIONS ON $\mathbb{R}^3$

2.1 Problem Settings and Initial Data

Hölder Continuous Initial Data

In this chapter, we study the 3D axisymmetric Euler equations with no swirl and with Hölder continuous initial angular vorticity. The initial data for ω^θ is C^α Hölder continuous, and is of the form $\omega^\theta \sim r^\alpha$ near $r = 0$, where α is the Hölder exponent. Such Hölder continuity of the angular vorticity implies that the velocity field u is $C^{1,\alpha}$ continuous. To remove the formal singularity in (**??**) near $r = 0$ and improve regularity of the vorticity field in favor of numerical computation, we introduce the new variables

$$\omega_1(r,z) = \frac{1}{r^\alpha}\omega^\theta(r,z), \quad \psi_1(r,z) = \frac{1}{r}\psi^\theta(r,z). \tag{2.1}$$

In terms of the new variables (ω_1, ψ_1), the 3D axisymmetric Euler equations with no swirl have the following equivalent form

$$\omega_{1,t} + u^r\omega_{1,r} + u^z\omega_{1,z} = -(1-\alpha)\psi_{1,z}\omega_1, \tag{2.2a}$$

$$-\psi_{1,rr} - \psi_{1,zz} - \frac{3}{r}\psi_{1,r} = \omega_1 r^{\alpha-1}, \tag{2.2b}$$

$$u^r = -r\psi_{1,z}, \quad u^z = 2\psi_1 + r\psi_{1,r}. \tag{2.2c}$$

Self-Similar Solution

For nonlinear PDEs, people are particularly interested in studying self-similar blow-up solutions. A self-similar solution is when the local profile of the solution remains nearly unchanged in time after rescaling the spatial and the temporal variables of the physical solution. For example, for (2.2), the self-similar profile is the ansatz

$$\begin{aligned} \omega_1(x,t) &\approx \frac{1}{(T-t)^{c_\omega}}\Omega\left(\frac{x-x_0}{(T-t)^{c_l}}\right), \\ \psi_1(x,t) &\approx \frac{1}{(T-t)^{c_\psi}}\Psi\left(\frac{x-x_0}{(T-t)^{c_l}}\right), \end{aligned} \tag{2.3}$$

for some parameters c_ω, c_ψ, c_l, x_0 and T. Here T is considered as the blow-up time, and x_0 is the location of the self-similar blow-up. The parameters c_ω, c_ψ, c_l are called scaling factors.

The 3D Euler equations (**??**) enjoy the following scaling invariant property: if (u, p) is a solution to (**??**), then $(u_{\lambda,\tau}, p_{\lambda,\tau})$ is also a solution, where

$$u_{\lambda,\tau}(x,t) = \frac{\lambda}{\tau} u\left(\frac{x}{\lambda}, \frac{t}{\tau}\right), \quad p_{\lambda,\tau}(x,t) = \frac{\lambda^2}{\tau^2} p\left(\frac{x}{\lambda}, \frac{t}{\tau}\right), \tag{2.4}$$

and $\lambda > 0$, $\tau > 0$ are two constant scaling factors. In terms of the equivalent form (2.2) of 3D Euler equations, the scaling invariant property is equivalent to: if (ω_1, ψ_1) is a solution of (2.2), then

$$\left\{\frac{1}{\lambda^{\alpha}\tau}\omega_1\left(\frac{x}{\lambda}, \frac{t}{\tau}\right), \frac{\lambda}{\tau}\psi_1\left(\frac{x}{\lambda}, \frac{t}{\tau}\right)\right\} \tag{2.5}$$

is also a solution.

If we assume the existence of the self-similar solution (2.3), then the new solutions in (2.5) should also admit the same ansatz, regardless of the values of λ and μ. As a result, we must have

$$c_{\omega} = 1 + \alpha c_l, \quad c_{\psi} = 1 - c_l. \tag{2.6}$$

Therefore, the self-similar profile (2.5) of (2.2) only has one degree of freedom, for example c_l, in scaling factors. In fact, c_l cannot be determined by straightforward dimensional analysis.

As a consequence of the ansatz (2.3) and the scaling relation (2.6), we have

$$\|\omega^{\theta}(x,t)\|_{L^{\infty}} \sim \frac{1}{T-t}, \quad \|\psi_{1,z}(x,t)\|_{L^{\infty}} \sim \frac{1}{T-t}, \tag{2.7}$$

which should always hold true regardless of the value of c_l.

Boundary Condition and Symmetry

We consider the axisymmetric Euler equations with no swirl (2.2) in a cylinder region

$$\mathcal{D}_{\text{cyl}} = \{(r,z) : 0 \le r \le 1\},$$

We impose a periodic boundary condition in z with period 1

$$\omega_1(r,z) = \omega_1(r,z+1), \quad \psi_1(r,z) = \psi_1(r,z+1). \tag{2.8}$$

In addition, we enforce that (ω_1, ψ_1) are odd in z at $z = 0$,

$$\omega_1(r,z) = -\omega_1(r,-z), \quad \psi_1(r,z) = -\psi_1(r,-z). \tag{2.9}$$

And this symmetry will be preserved dynamically by the 3D Euler equations.

At $r = 0$, it is easy to see that $u^r(0, z) = 0$, so there is no need for the boundary condition for ω_1 at $r = 0$. Since $\psi^\theta = r\psi_1$ will at least be C^2-continuous, according to [66, 65], ψ^θ must be an odd function of r. Therefore, we impose the following pole condition for ψ_1

$$\psi_{1,r}(0, z) = 0. \tag{2.10}$$

Since we assume a solid "wall" at the boundary at $r = 1$, we impose the no-flow boundary condition

$$\psi_1(1, z) = 0. \tag{2.11}$$

This implies that $u^r(1, z) = 0$. So there is no need for the boundary condition for ω_1 at $r = 1$ as well.

Due to the periodicity and odd symmetry along the z direction, the equations (**??**) only need to be solved on the half-periodic cylinder

$$\mathcal{D} = \{(r, z) : 0 \leq r \leq 1, 0 \leq z \leq 1/2\}.$$

It is important to notice that the above boundary conditions of $\mathcal{D}$ allow no transportation across its boundaries. Indeed, we have

$$u^r = 0 \quad \text{on} \quad r = 0, \text{ and } r = 1,$$

and

$$u^z = 0 \quad \text{on} \quad z = 0, \text{ and } z = 1/2.$$

Initial Data

Inspired by the potential blow-up scenario in [43], we propose the following initial data for ω_1 in $\mathcal{D}$,

$$\omega_1^\circ = \frac{-12000\left(1 - r^2\right)^{18}\sin(2\pi z)}{1 + 12.5\cos^2(\pi z)}. \tag{2.12}$$

Later we will see in Section 2.8 that the self-similar singularity formation has some robustness to the choice of initial data. We solve the Poisson equation (**??**) to get the initial value ψ_1° of ψ_1.

The 3D profile and pseudocolor plot of $(\omega_1^\circ, \psi_1^\circ)$ can be found in Figure 2.1. Since most parts of ω_1° and ψ_1° are negative, we negate them for better visual effect when generating figures.

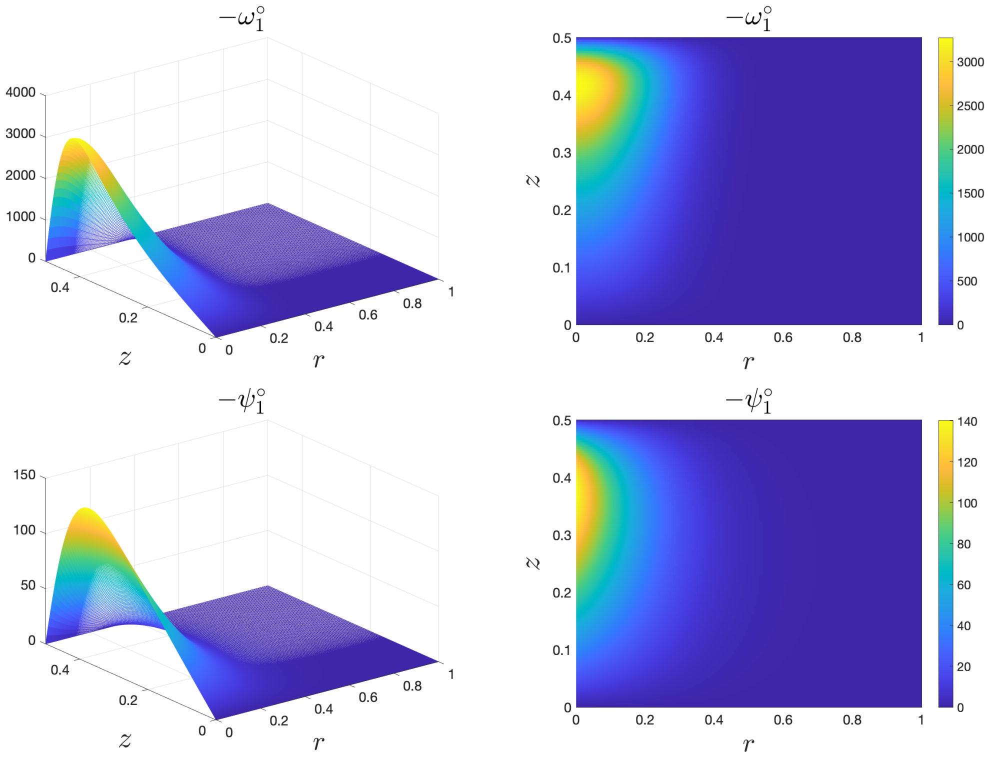


Figure 2.1: 3D profiles and pseudocolor plots of the initial value $-\omega_1^\circ$ and $-\psi_1^\circ$.

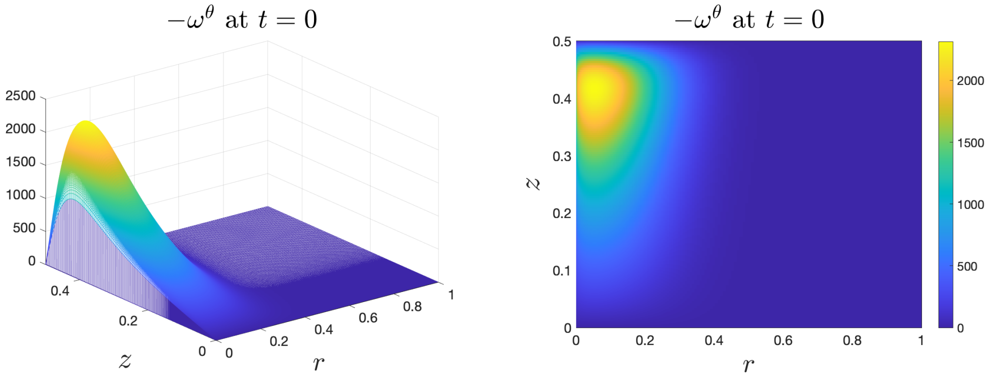


Figure 2.2: Angular vorticity ω^θ at $t = 0$.

In Figure 2.2, we show the 3D profile and pseudocolor plot of the angular vorticity ω^θ at $t = 0$. We can see that there is a sharp drop to zero of $-\omega^\theta$ near $r = 0$, which is due to the Hölder continuous of ω^θ at $r = 0$.

We plot the initial velocity field u^r and u^z in Figure 2.3. We can see that u^r is primarily positive near $z = 0$ and negative near $z = 1/2$ when r is small, and u^z is mainly negative when r is small. Such a pattern suggests a hyperbolic flow near

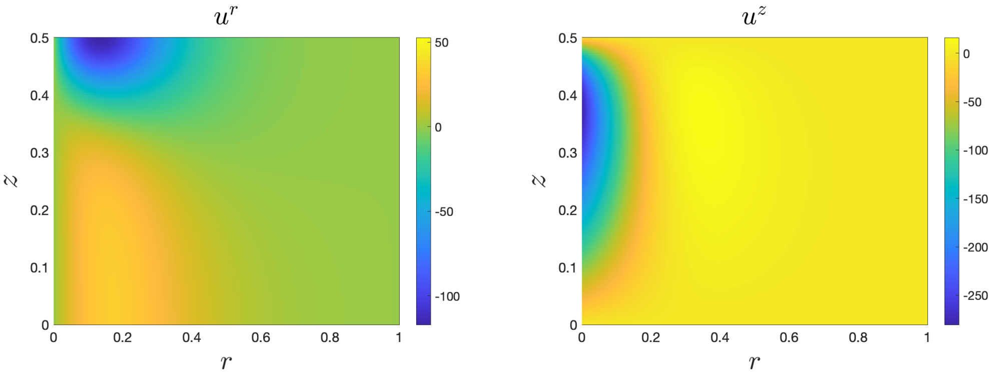


Figure 2.3: Initial velocity fields u^r and u^z.

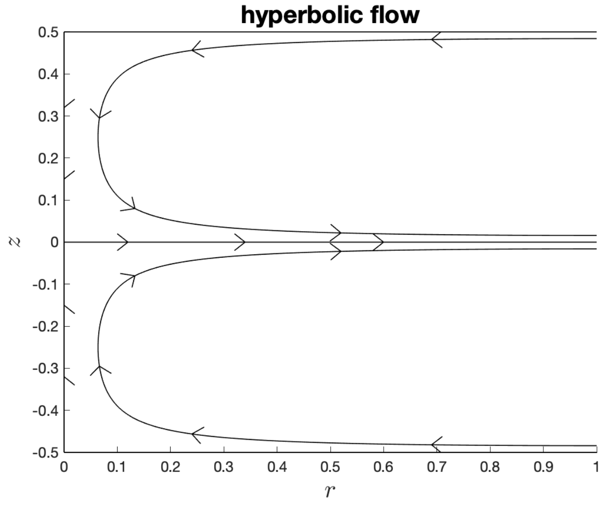


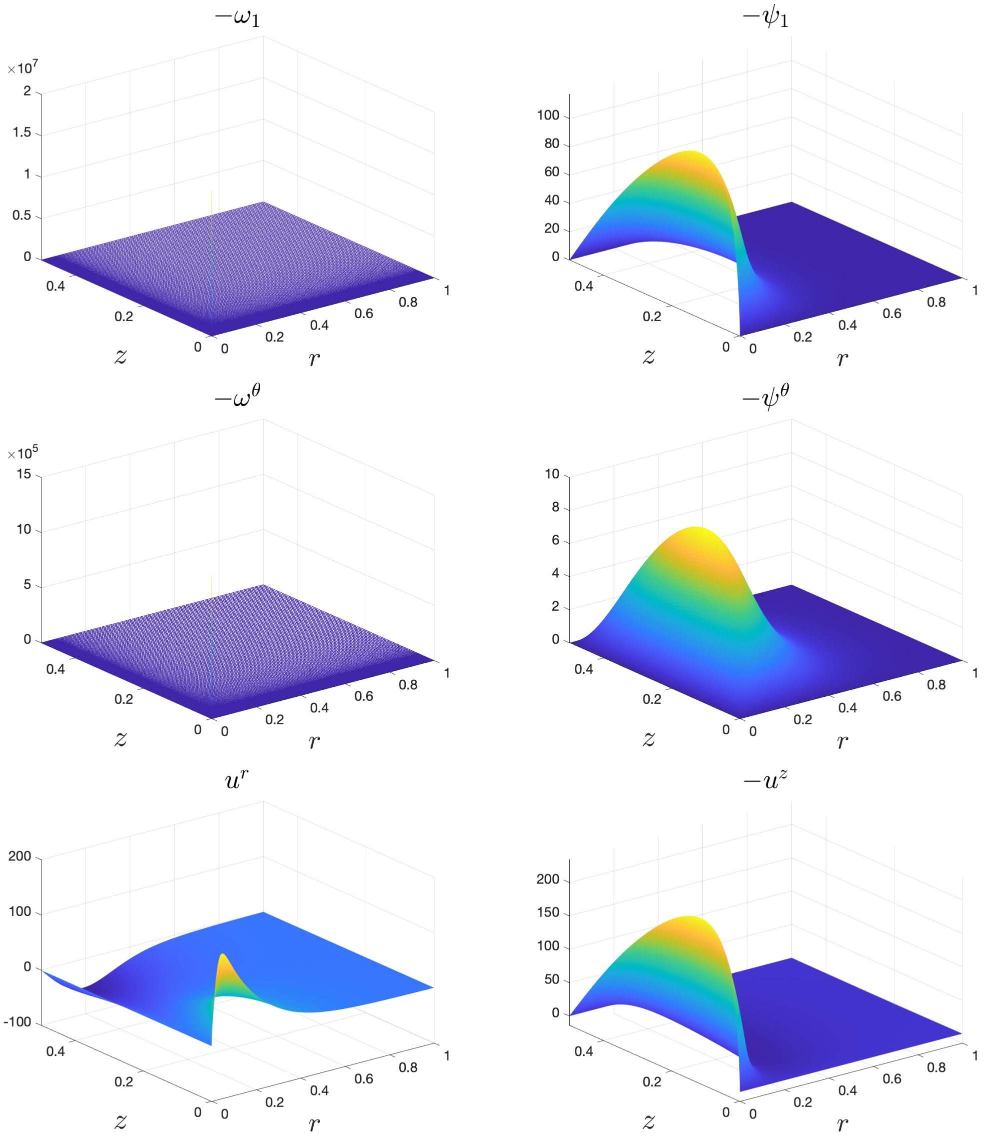
Figure 2.4: A heuristic diagram of the hyperbolic flow.

$(r, z) = (0, 0)$ as depicted in the heuristic diagram Figure 2.4, which will extend periodically in z.

2.2 Numerical Evidence for the Potential Blow-Up

From this section to Section 2.4, we will stick to the case with Hölder exponent $\alpha = 0.1$. We will present the results with different values of Hölder exponent α in Section 2.5, Section 2.6, and afterwards.

On 1024×1024 spatial resolution, we use the adaptive mesh method to solve (2.2) with Hölder exponent $\alpha = 0.1$, until the time when the local resolution of adaptive mesh gets close to the machine precision. The adaptive mesh method is described in details in Appendix A. The stopping time is reported as $t = 1.6524635 \times 10^{-3}$, after more than 6.5×10^4 iterations of computation. We plot the 3D profiles of ω_1, ψ_1,



$-\omega_1$ $-\psi_1$ $-\omega^\theta$ $-\psi^\theta$ u^r $-u^z$

Figure 2.5: Profiles of $-\omega_1$, $-\psi_1$, $-\omega^\theta$, $-\psi^\theta$, u^r and $-u^z$ at $t = 1.6524635 \times 10^{-3}$ on the whole domain $\mathcal{D}$.

ω^θ, ψ^θ, u^r, and u^z at this time in Figure 2.5. We can see that ω_1 is very concentrated near the origin, and so is ω^θ. Therefore, we zoom-in around the origin and plot the local near field profiles of ω_1, ψ_1, ω^θ, ψ^θ, u^r, and u^z in Figure 2.6.

From Figure 2.6, we observe that the "peak" of $-\omega_1$ locates at the z-axis where $r = 0$, and is being pushed toward the origin as implied by the velocity field u^r, u^z. Let $(R_1(t), Z_1(t))$ be the point to achieve maximum magnitude of $-\omega_1$ at time t, we have $R_1(t) = 0$. And at $(R_1(t), Z_1(t))$, the radial velocity u^r is zero, and the axial velocity u^z is negative, so $Z_1(t)$ should decrease in time.

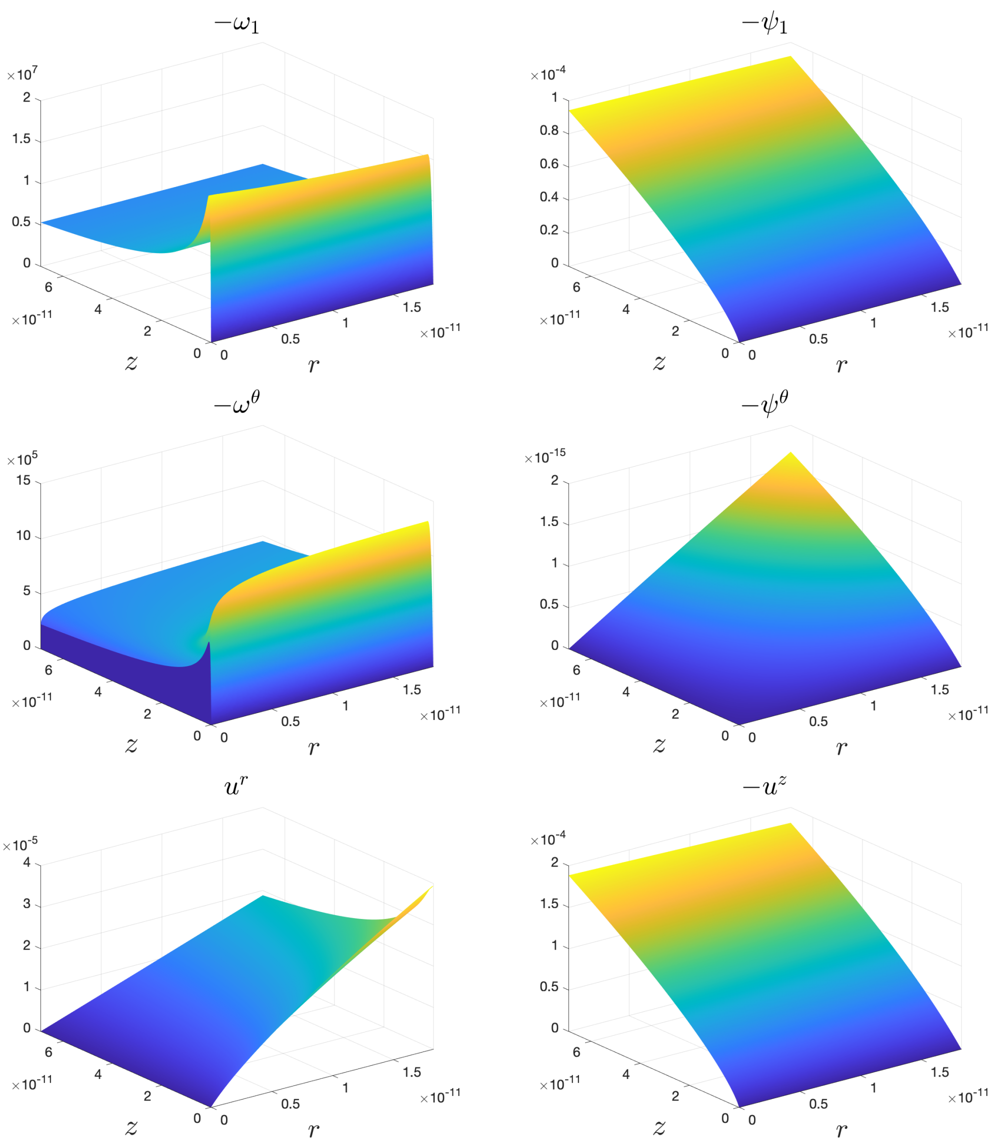


Figure 2.6: Zoomed-in profiles of $-\omega_1$, $-\psi_1$, $-\omega^{\theta}$, $-\psi^{\theta}$, u^r, and $-u^z$ near the origin $(0,0)$ at $t = 1.6524635 \times 10^{-3}$.

In Figure 2.7, we plot the local velocity field near the maximum of $-\omega^{\theta}$ and $-\omega_1$, respectively. We use the pseudocolor plots of $-\omega^{\theta}$ and $-\omega_1$ as the background, respectively for the figure in left and right, and mark the maximum of $-\omega^{\theta}$ or $-\omega_1$ with the red dot. The velocity field demonstrates a clear hyperbolic structure as depicted by Figure 2.4. And the velocity field clearly pushes the maximum $(R_1(t), Z_1(t))$ of $-\omega_1$ toward the origin.

In Figure 2.8, we show the local streamlines near the maximum of $-\omega^{\theta}$ in $\mathbb{R}^3$. The maximum of $-\omega^{\theta}$ locates in the red ring around the z-axis. In the left figure,

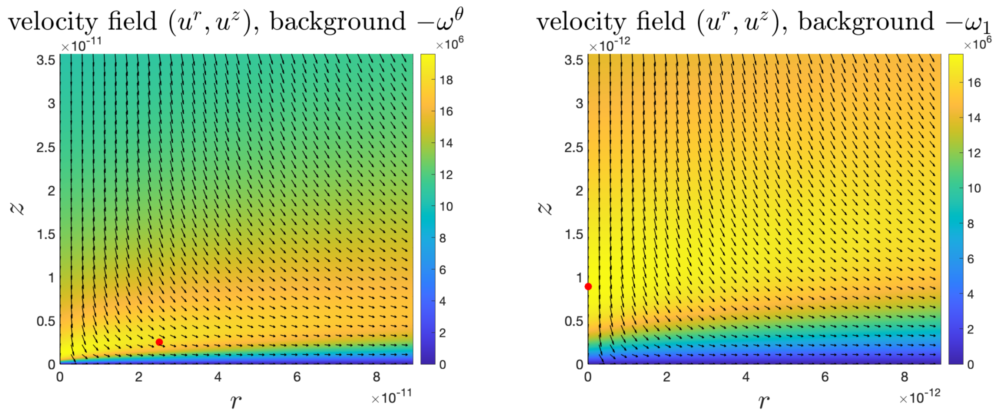


Figure 2.7: The local velocity field near the maximum of $-\omega^\theta$ and $-\omega_1$. The pseudocolor plot of $-\omega^\theta$ or $-\omega_1$ is the background, and the red dot is its maximum.

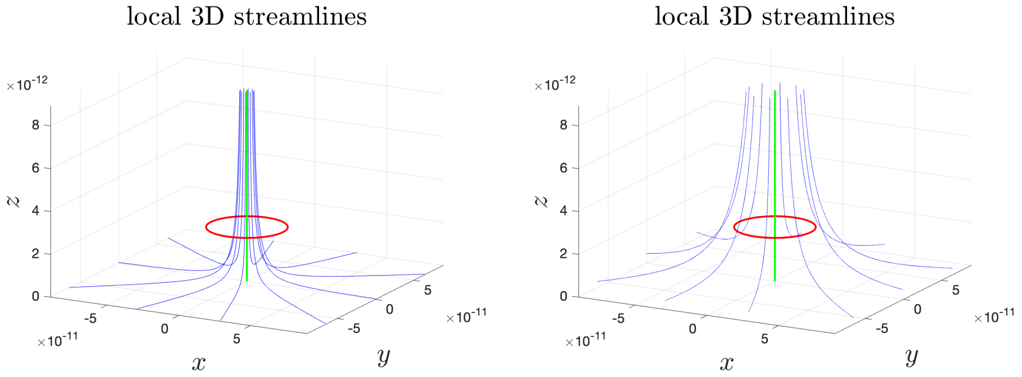


Figure 2.8: The local streamlines near the origin. The green pole is the z-axis, and the red ring is where $-\omega^\theta$ achieves its maximum.

we plot a set of streamlines that travel through the maximum ring from top to bottom. And in the right figure, we plot a set of streamlines that travel around the maximum ring from top to bottom. From Figure 2.8, we notice that the streamlines are axisymmetric, and do not form swirl around the z-axis.

In Figure 2.9, we record curves of important quantities of the system. The magnitude of ω_1 has grown significantly, especially near the end of the computation. At the final time of the computation, $\|\omega_1\|_{L^\infty}$ has increased by a factor of around 5400, and $\|\omega\|_{L^\infty}$ has increased by a factor of more than 560. We also observe that the double logarithm curve of the maximum vorticity magnitude, $\log\log\|\omega\|_{L^\infty}$, maintains a super-linear growth, and the time integral $\int_0^t \|\omega(s)\|_{L^\infty}\mathrm{d}s$ has rapid growth with strong growth inertia close to the stopping time. This provides strong evidence for a potential finite-time blow-up of the 3D Euler equations by the Beale-Kato-Majda

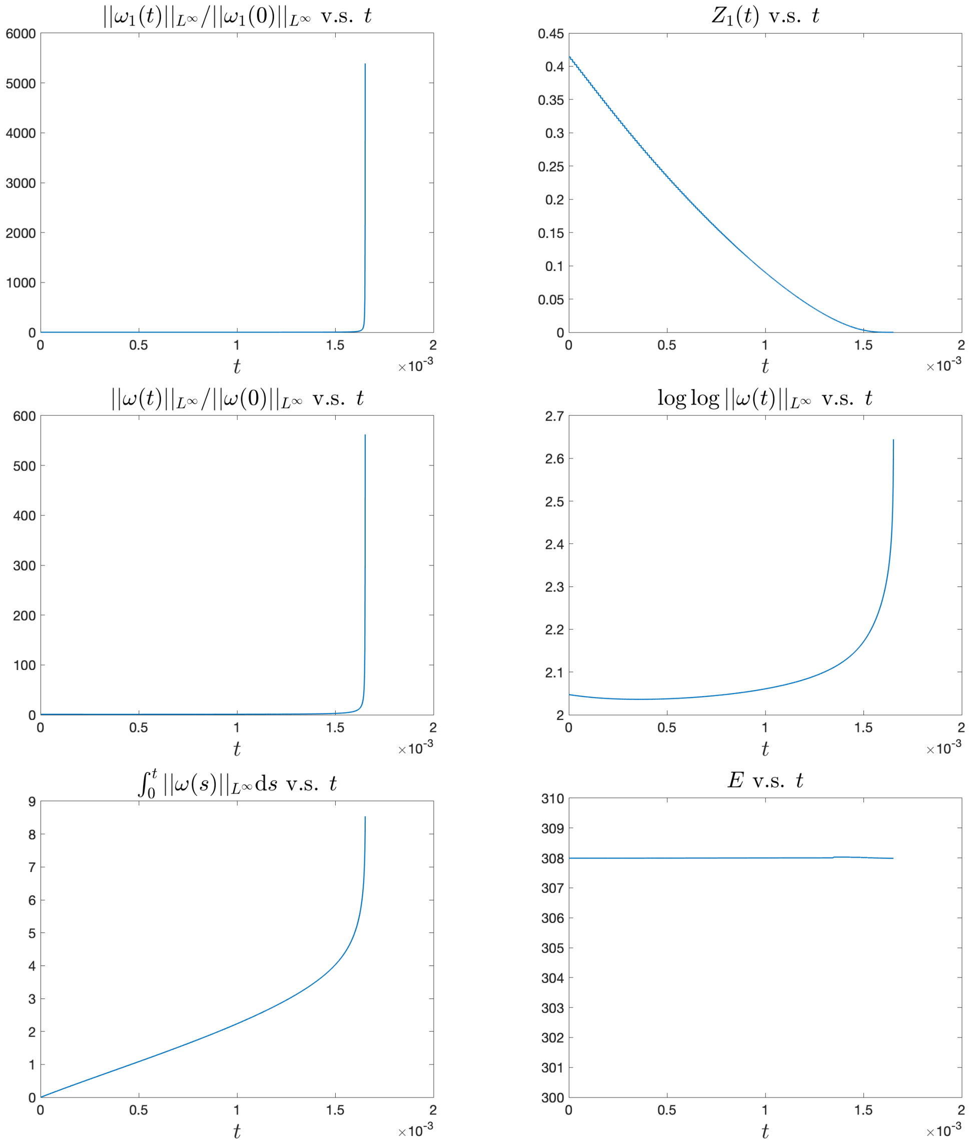


Figure 2.9: Curves of $\|\omega_1\|_{L^\infty}$, Z_1, $\|\omega\|_{L^\infty}$, $\log\log\|\omega\|_{L^\infty}$, $\int_0^t \|\omega(s)\|_{L^\infty}\mathrm{d}s$ and E as functions of time t.

blow-up criterion (**??**).

The kinetic energy E, which is defined as

$$E = \frac{1}{2}\int_{\mathcal{D}} |u|^2\,\mathrm{d}x = \pi\int_0^1\int_0^{1/2}\left(|u^r|^2 + |u^z|^2\right) r\mathrm{d}r\mathrm{d}z,$$

for our axisymmetric case with no swirl, is a conservative quantity of the 3D Euler equations. Despite the discretization error, numerical dissipation, round-up error, and other numerical errors in our numerical method, the kinetic energy E should still be bounded from below and above. In Figure 2.9, we can see that there is little change of the kinetic energy E as a function of time t. In fact, the major reason for the change of E in our computation is due to the update of adaptive mesh, where we need to interpolate ω_1 and ψ_1 from an old mesh to a new mesh. Since the new adaptive mesh will be more focusing on the near field around the origin, the far field velocity field might lose some accuracy, leading to a change in the kinetic energy E. However, such an update of adaptive mesh happens occasionally (35 times out of 65000 iterations), and each such change in the kinetic energy E is negligible. By the end of the computation, the change in the kinetic energy E is at most 1.4×10^{-4} of the magnitude of E.

From Figure 2.9, we can also see that $Z_1(t)$ monotonically decreases to zero with t. The curve of $Z_1(t)$ seems to be convex, especially in time windows close to the stopping time. We refer to Section 2.3 for more study of the behavior of $Z_1(t)$.

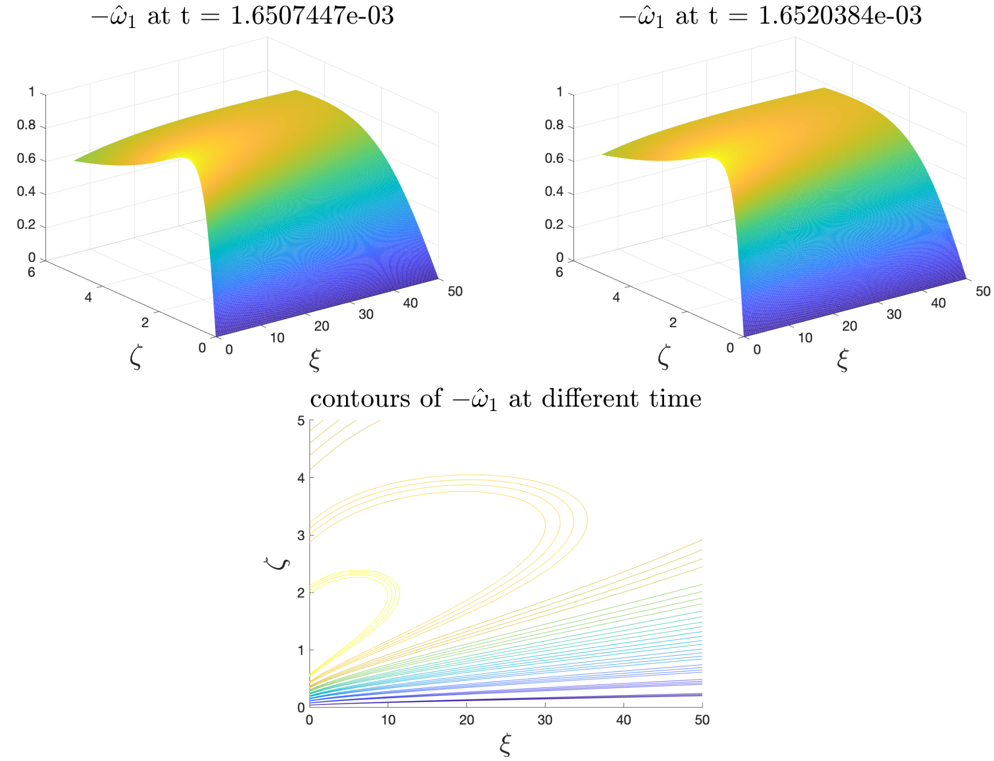


Figure 2.10: Top row: Local profiles of $-\hat{\omega}_1$ at $t = \{1.6507447, 1.6520384\} \times 10^{-3}$. Bottom row: Local contours of $-\hat{\omega}_1$ at $t = \{1.6507447, 1.6512953, 1.6517173, 1.6520384\} \times 10^{-3}$.

To check the self-similar property of the solution, we visualize the local profile of the scaled ω_1 near the origin. We define

$$\hat{\omega}_1(\xi, \zeta, t) = \omega_1\left(Z_1(t)\xi, Z_1(t)\zeta, t\right) / \|\omega_1(t)\|_{L^\infty},$$

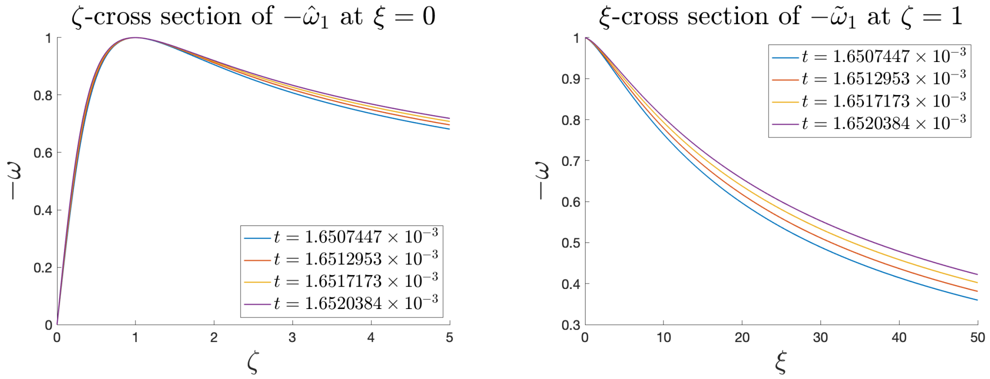


Figure 2.11: Cross sections of $-\hat{\omega}_1$ at different times.

as the scaled version of ω_1. The above definition pins the magnitude of $|\hat{\omega}_1|$ to 1, and pins the maximum location of $|\hat{\omega}_1|$ to $(\xi, \zeta) = (0, 1)$. We plot the profiles of $-\hat{\omega}_1$ near the origin at different time instants in the top row of Figure 2.10, and plot the contours of $-\hat{\omega}_1(\xi, \zeta)$ at different time in the bottom row. The profile of $-\hat{\omega}_1$ seems to change slowly in the late time, indicating a potential self-similar structure of the blow-up profile near the origin. In other words, $x_0 = 0$ in the self-similar ansatz (2.3). In Figure 2.11, we plot the cross sections of $-\hat{\omega}_1$ at $\xi = 0$ and $\zeta = 1$. The cross section at $\xi = 0$ shows a good potential for a self-similar blow-up, while the cross section at $\zeta = 1$ shows that the blow-up profile has not converged to a self-similar profile yet. This is reasonable because although we are very close to the potential blow-up time, the strong collapsing along the z-direction and the effect of round-off errors prevent us from continuing the computation. We refer to Section 2.4 where we use the dynamic rescaling method and indeed observe numerically the convergence to the potential self-similar profile.

Resolution Study

We perform resolution study on the numerical solutions of (2.2) to verify the validity of our numerical results. We first simulate the equations on spatial resolutions of $256k \times 256k$ with $k = 1, 2, \ldots, 6$. The highest resolution we used is 1536×1536. Next, for the numerical solution at resolution $256k \times 256k$, we compute its sup-norm relative error in several chosen quantities at selected time instants using the numerical solution at resolution $256(k+1) \times 256(k+1)$ as the reference, for $k = 1, 2, \ldots, 5$. Finally, we use the relative error obtained above to estimate the convergence order of the numerical method.

We consider two types of quantities. The first type is the function of the solutions.

Here we consider the magnitude of ω_1, $\|\omega_1\|_{L^\infty}$, the maximum norm of vorticity, $\|\omega\|_{L^\infty}$, and the kinetic energy, E. We remark that $\|\omega_1\|_{L^\infty}$ and $\|\omega\|_{L^\infty}$ only depend on the local field near the origin, and E should be considered as a global quantity. The second type is the vector fields of ω_1, ψ_1, u^r, and u^z that are actively participating in the simulated system (2.2).

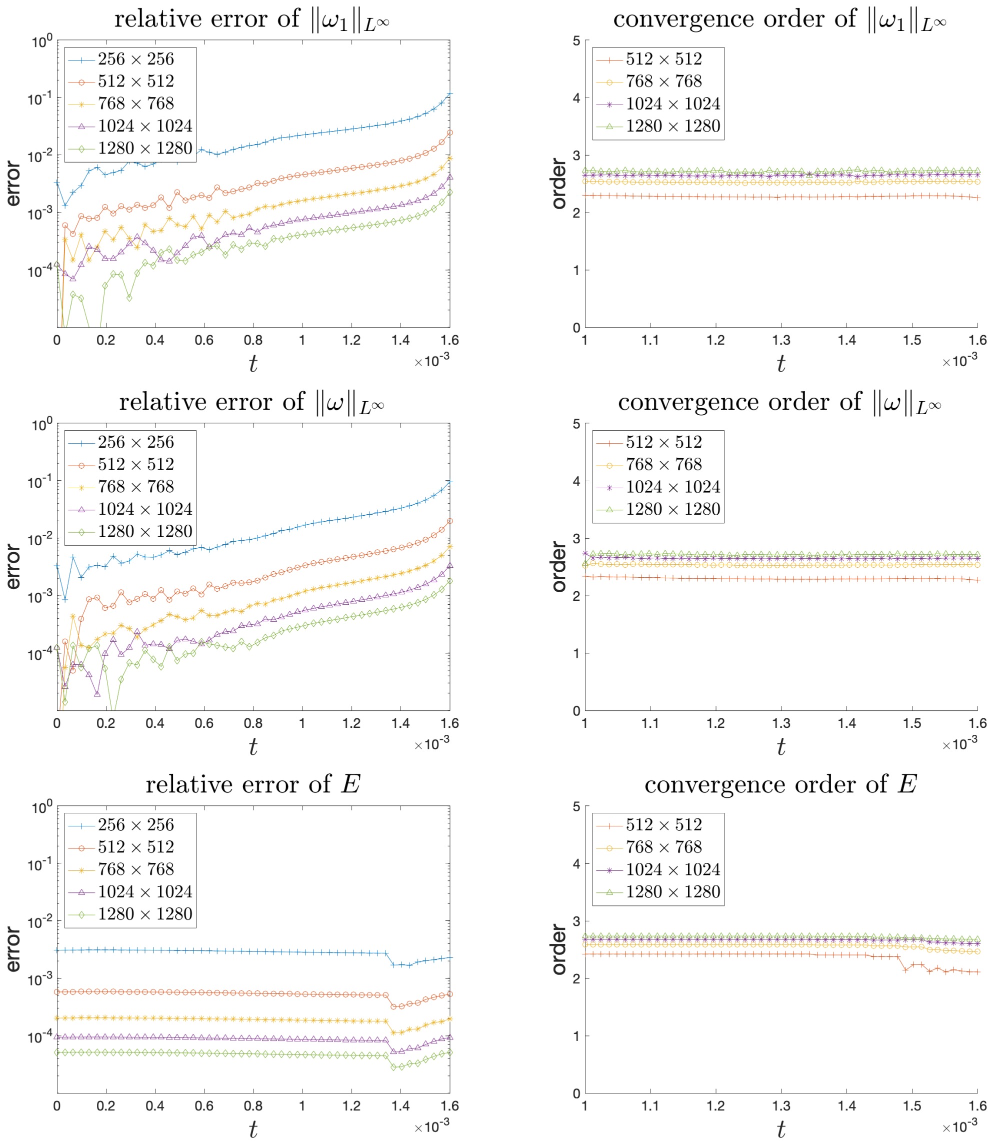


Figure 2.12: Relative errors and convergence orders of $\|\omega_1\|_{L^\infty}$, $\|\omega\|_{L^\infty}$, and E in sup-norm.

For each quantity, we use q_k to represent the estimate we get at resolution $256k \times 256k$. Then the sup-norm relative error e_k is defined as

$$e_k = \|q_k - q_{k+1}\|_{L^\infty} / \|q_{k+1}\|_{L^\infty}.$$

If q_k is a vector field, we first interpolate it to the reference resolution $256(k+1) \times 256(k+1)$, and then compute the relative error as above. The convergence order of the error β_k at this resolution can be estimated via

$$\beta_k = \log\left(\frac{e_{k-1}}{e_k}\right) \Big/ \log\left(\frac{k}{k-1}\right).$$

In Figure 2.12, we plot the relative error of the quantities $\|\omega_1\|_{L^\infty}$, $\|\omega\|_{L^\infty}$ and E for $t \in \left[0, 1.6 \times 10^{-3}\right]$, and the convergence order of the error in the late time $t \in \left[1 \times 10^{-3}, 1.6 \times 10^{-3}\right]$. We observe a numerical convergence with order slightly higher than 2. The convergence order is quite stable in the time interval of our computation.

mesh size	Sup-norm relative error at $t = 1.6 \times 10^{-3}$			
	ω_1	order	ψ_1	order
256×256	2.545×10^{-1}	-	5.912×10^{-3}	-
512×512	5.478×10^{-2}	2.216	1.168×10^{-3}	2.340
768×768	1.969×10^{-2}	2.524	4.136×10^{-4}	2.560
1024×1024	9.189×10^{-3}	2.655	1.926×10^{-4}	2.656
1280×1280	5.008×10^{-3}	2.720	1.050×10^{-4}	2.719

mesh size	Sup-norm relative error at $t = 1.6 \times 10^{-3}$			
	u^r	order	u^z	order
256×256	2.035×10^{-2}	-	8.095×10^{-3}	-
512×512	3.954×10^{-3}	2.364	1.533×10^{-3}	2.310
768×768	1.405×10^{-3}	2.552	5.793×10^{-4}	2.556
1024×1024	6.540×10^{-4}	2.658	2.699×10^{-4}	2.655
1280×1280	3.594×10^{-4}	2.682	1.472×10^{-4}	2.719

Table 2.1: Relative errors and convergence orders of ω_1, ψ_1, u^r and u^z in sup-norm.

In Table 2.1, we list the relative error and convergence order of the vector fields at $t = 1.6 \times 10^{-3}$. The convergence order stays well above 2, suggesting an at least second-order convergence for our numerical solver of the 3D Euler equations.

Effectiveness of the Adaptive Mesh

Using the initial data (2.12), ω_1 will quickly become very singular near the origin. Our adaptive mesh is designed to resolve the singular profile of ω_1. The idea of the adaptive mesh is to introduce two variables

$$(\rho, \eta) \in [0,1] \times [0,1],$$

and the maps

$$r = r(\rho), \quad z = z(\eta),$$

such that we map the physical domain in (r, z) to a computational domain in (ρ, η). The maps $r = r(\rho)$, $z = z(\eta)$ are designed to resolve the profile of ω_1 well in the plane of (ρ, η). The detailed settings of the adaptive mesh can be found in Appendix A.2. In this section, we test the effectiveness of our adaptive mesh.

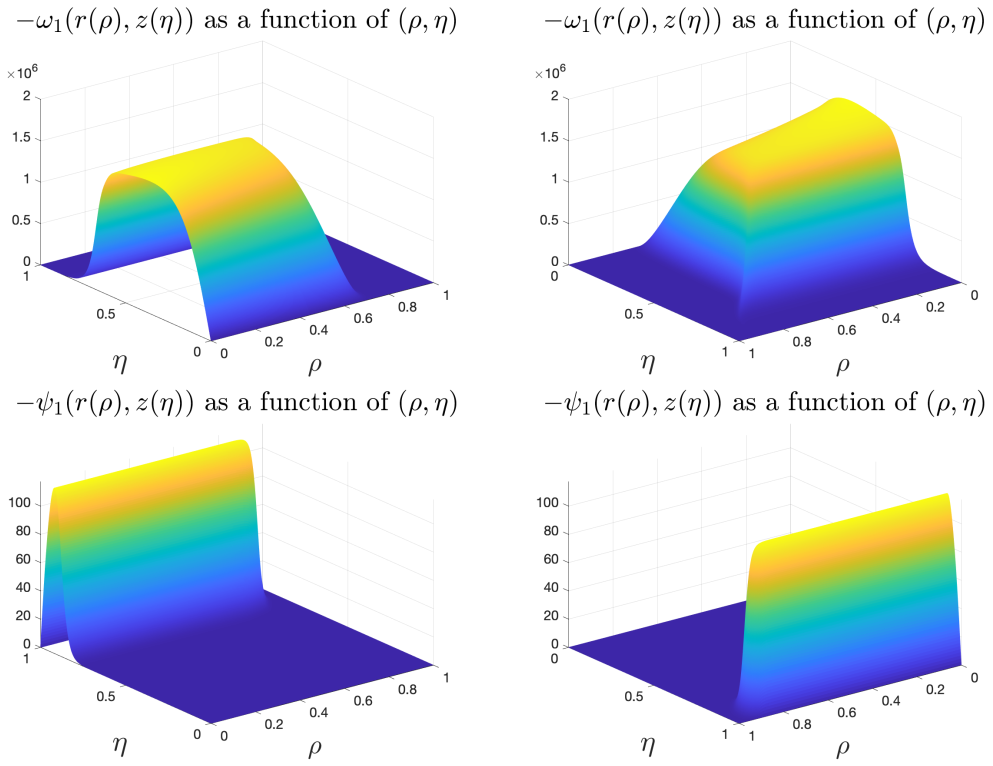


Figure 2.13: Profiles of $-\omega_1(r(\rho), z(\eta))$ and $-\psi_1(r(\rho), z(\eta))$ as functions of (ρ, η) from two different angles at $t = 1.65 \times 10^{-3}$.

We first visualize $\omega_1(r(\rho), z(\eta))$ and $\psi_1(r(\rho), z(\eta))$ as functions of (ρ, η) in Figure 2.13 at $t = 1.65 \times 10^{-3}$. Although ω_1 is very singular as a function of (r, z) as in Figure 2.5, it is clear that ω_1 is well-resolved in the (ρ, η)-plane. We can also see that ψ_1 is already relatively smooth in the (r, z)-plane, especially when far away from the origin. Therefore, we do not need to place many points in the far field to resolve ψ_1.

In Figure 2.14, we visualize the local mesh view of ω_1 and ψ_1 at $t = 1.65 \times 10^{-3}$ from which we can see how the density of the adaptive mesh distributes in different regions. We can see that along the r-direction, the adaptive mesh has three different phases from $r = 0$ to the far field: intermediate density, high density, and low

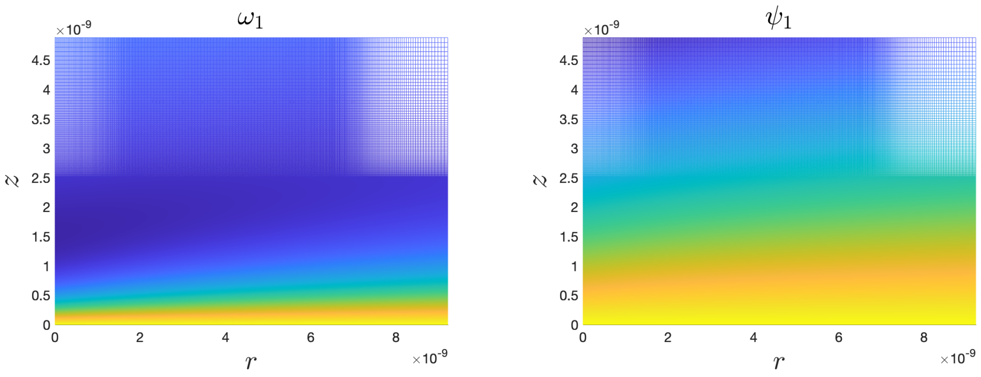


Figure 2.14: Local mesh view of ω_1 and ψ_1 at $t = 1.65 \times 10^{-3}$.

density. Along the z-direction, the adaptive mesh has two phases from $z = 0$ to the far field: high density and low density. Most of the mesh concentrates around $z = 0$ where the solution is most singular in the profiles of ω_1 and ψ_1.

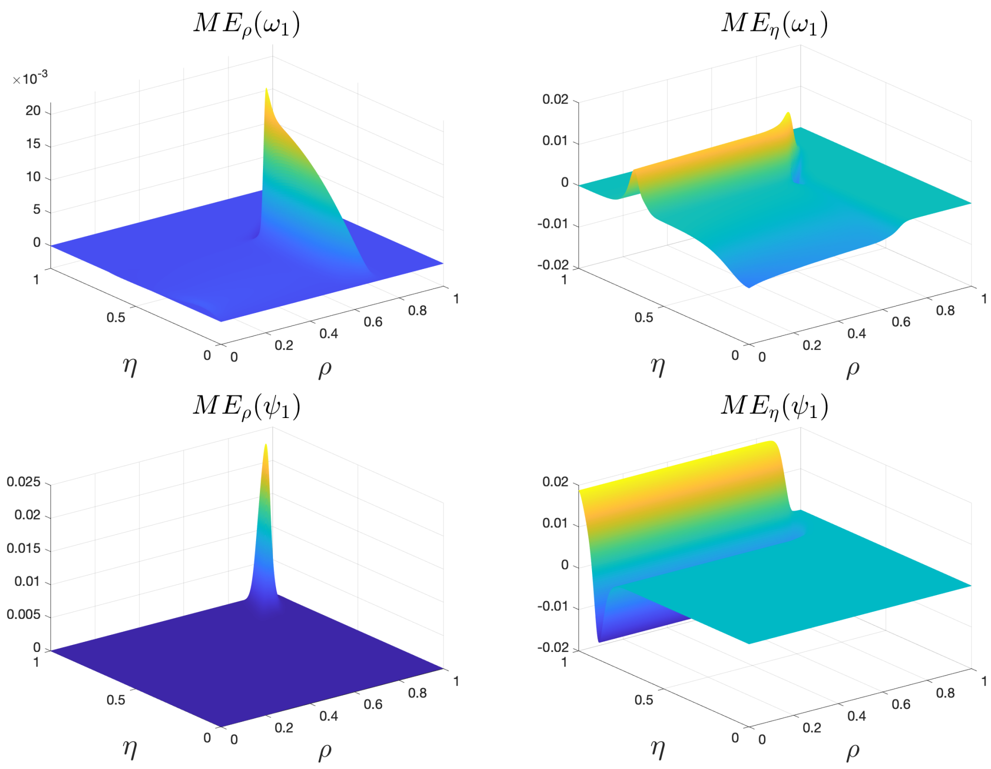


Figure 2.15: Mesh effectiveness functions ME_ρ and ME_η of ω_1 and ψ_1 at $t = 1.65 \times 10^{-3}$.

Following [44], we define the mesh effectiveness functions to quantify the quality of the adaptive mesh. For some function v on the (r, z)-plane, we define

$$\mathrm{ME}_\rho(v) = \frac{h_\rho v_\rho}{\|v\|_{L^\infty}} = \frac{h_\rho r_\rho v_r}{\|v\|_{L^\infty}}, \quad \mathrm{ME}_\eta(v) = \frac{h_\eta v_\eta}{\|v\|_{L^\infty}} = \frac{h_\eta z_\eta v_z}{\|v\|_{L^\infty}},$$

where h_ρ, h_η is the resolution on the plane of (ρ, η). We further define the mesh effectiveness measures as follows:

$$\mathrm{MEM}_\rho(v) = \|\mathrm{ME}_\rho(v)\|_{L^\infty}, \quad \mathrm{MEM}_\rho(v) = \|\mathrm{ME}_\eta(v)\|_{L^\infty}.$$

According to [44], the mesh effectiveness measures estimate the greatest relative growth of a function in a single mesh cell on the (ρ, η)-plane. Small mesh effectiveness measure values indicate that the adaptive mesh has resolved the function well. Thus, we study the mesh effectiveness measures for our adaptive mesh.

We can see from Figure 2.15 that the mesh effectiveness functions are all uniformly bounded. Their magnitude has a relatively small absolute value.

mesh size	Mesh effectiveness measures at $t = 1.65 \times 10^{-3}$			
	$\mathrm{MEM}_\rho(\omega_1)$	$\mathrm{MEM}_\eta(\omega_1)$	$\mathrm{MEM}_\rho(\psi_1)$	$\mathrm{MEM}_\eta(\psi_1)$
256×256	0.087	0.046	0.095	0.075
512×512	0.044	0.027	0.048	0.037
768×768	0.029	0.017	0.032	0.025
1024×1024	0.022	0.014	0.024	0.019
1280×1280	0.017	0.010	0.019	0.015
1536×1536	0.014	0.008	0.016	0.012

Table 2.2: Mesh effectiveness measures MEM_ρ and MEM_η of ω_1 and ψ_1 at different mesh sizes at $t = 1.65 \times 10^{-3}$.

time	Mesh effectiveness measures at mesh size 1024×1024			
	$\mathrm{MEM}_\rho(\omega_1)$	$\mathrm{MEM}_\eta(\omega_1)$	$\mathrm{MEM}_\rho(\psi_1)$	$\mathrm{MEM}_\eta(\psi_1)$
1.60×10^{-3}	0.028	0.009	0.024	0.019
1.61×10^{-3}	0.027	0.010	0.024	0.019
1.62×10^{-3}	0.025	0.010	0.024	0.019
1.63×10^{-3}	0.024	0.010	0.024	0.019
1.64×10^{-3}	0.023	0.011	0.024	0.019
1.65×10^{-3}	0.022	0.014	0.024	0.019

Table 2.3: Mesh effectiveness measures MEM_ρ and MEM_η of ω_1 and ψ_1 at different time instants at mesh size 1024×1024.

In Table 2.2 and 2.3, we list the mesh effectiveness measures MEM_ρ and MEM_η of ω_1 and ψ_1 at different mesh sizes and different time instants. Mesh effectiveness measures decrease as the resolution increases, which is reasonable because higher resolution is expected to be more powerful to resolve the singular part of the solution. Despite that mesh effectiveness measures might slightly increase with the time, they all remain relatively small throughout the computation, which means that our adaptive mesh method has done a good job in resolving the singular solution of the 3D Euler equations.

2.3 Scaling Analysis of the Potential Blow-Up

In this section, we quantify the scaling property of the potential blow-up observed in our computation. This scaling analysis will give more supporting evidence that the potential blow-up satisfies the Beale-Kato-Majda blow-up criterion (**??**). It also uncovers more properties of the potential blow-up.

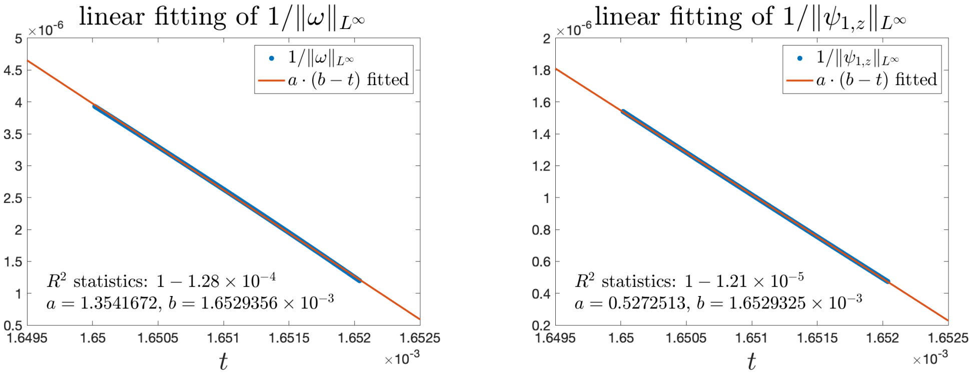


Figure 2.16: Linear fitting of $1/\|\omega\|_{L^\infty}$ and $1/\|\psi_{1,z}\|_{L^\infty}$ with time.

As discussed in (2.7) of Section 2.1, if there is a self-similar blow-up, the scaling invariant property of the 3D Euler equations will ensure that $\|\omega\|_{L^\infty} \sim 1/(T-t)$ and $\|\psi_{1,z}\|_{L^\infty} \sim 1/(T-t)$. Therefore, we examine this property by regressing $\|\omega\|_{L^\infty}^{-1}$ and $\|\psi_{1,z}\|_{L^\infty}^{-1}$ again t, respectively. More specifically, for a quantity v, which is either $\|\omega\|_{L^\infty}^{-1}$ or $\|\psi_{1,z}\|_{L^\infty}^{-1}$, we perform the least square fitting of the model

$$v \sim a\cdot(b-t),$$

in searching for constants a and b, where a is the negated slope of the fitted line, and b can be considered as the estimate time of the blow-up. In Figure 2.16, we visualize the data points and the fitted line using data between $t = 1.6500174\times10^{-3}$ and $t = 1.6520384\times10^{-3}$. The R^2 of the fitting between $\|\omega\|_{L^\infty}^{-1}$ and t is $1-1.28\times10^{-4}$, and the R^2 of the fitting between $\|\psi_{1,z}\|_{L^\infty}^{-1}$ and t is $1-1.21\times10^{-5}$. Such high R^2 values show strong linear relation between $\|\omega\|_{L^\infty}^{-1}$, $\|\psi_{1,z}\|_{L^\infty}^{-1}$ and t. Moreover, the fittings of the two quantities estimate the blow-up time to be $b = 1.6529356 \times 10^{-3}$ and $b = 1.6529325 \times 10^{-3}$ respectively. These two blow-up times agree with each other up to 6 digits. Therefore, Figure 2.16 provides further evidence that the 3D Euler equations develop a potential finite-time singularity.

We next move to fit the scaling factors c_l and c_ω used in the self-similar ansatz (2.3) of the solutions. Since the functions Ω and Ψ are time-independent in (2.3), we should have that

$$Z_1 \sim (T-t)^{c_l}, \quad \|\omega_1\|_{L^\infty}^{-1} \sim (T-t)^{c_\omega},$$

where we recall that $Z_1 = Z_1(t)$ is the z-coordinate of the maximum location of $-\omega_1$. Due to the unknown powers c_l and c_ω, the direct fitting of the above model is nonlinear. Therefore, we turn to a searching algorithm for the power variable. Specifically, for a quantity v, that is either Z_1 or $\|\omega_1\|_{L^\infty}^{-1}$, we search for a power c such that the linear regression of

$$v^{1/c} \sim a \cdot (b-t),$$

has the largest R^2 value up to some error tolerance. We will start with a guessed window of the power c, and then exhaust the value of c within the window up to some error tolerance, and choose c with the largest R^2 value. If the optimal c we searched falls on the boundary of the current window, we then adaptively adjust the window size and location, and repeat the above procedure. When the optimal searched c falls within the interior of the window, we stop the searching.

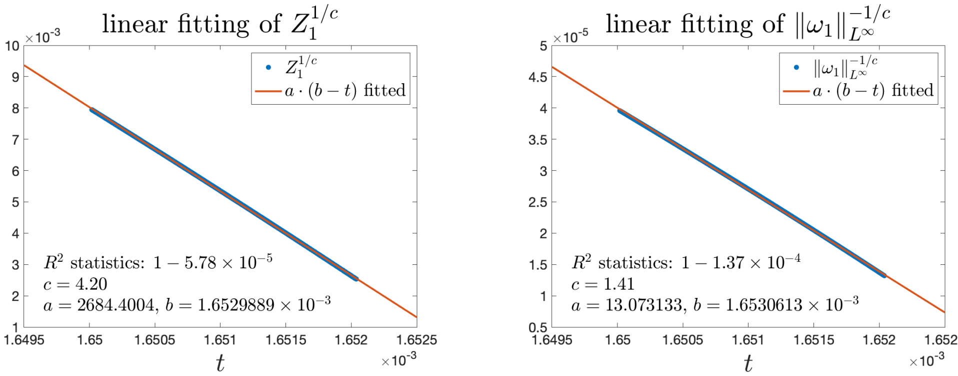


Figure 2.17: Linear fitting of $Z_1^{1/c}$ and $\|\omega_1\|_{L^\infty}^{-1/c}$ with time.

In Figure 2.17, we demonstrate the result of the searching. We can see that with the chosen c, the linear regression achieves a very high R^2 value, suggesting a strong linear relation. The estimated blow-up times only relatively differ from the previous estimate by at most 7.8×10^{-5}. Moreover, the searching suggests that $c_l \approx 4.20$ and $c_\omega \approx 1.41$, and these estimated values of c_l and c_ω satisfy the scaling relation $c_\omega = 1 + \alpha c_l$ as in (2.6) approximately.

It is worth emphasizing that the estimated c_l is well above 1, and this explains the convex curve of $Z_1(t)$ as observed in Figure 2.9 in Section 2.2.

We remark that we did not perform the searching algorithm with $\|\psi_1\|_{L^\infty}$ to find out the scaling factor c_ψ, so that we could check the other scaling relation $c_\psi = 1 - c_l$ in (2.6). This is because $\|\psi_1\|_{L^\infty}$ is mainly affected by the far field behavior of ψ_1, as shown in Figure 2.5. However, the self-similar ansatz (2.3) is only valid in the near field, so such fitting is meaningless. In fact, the good fitting between $\|\psi_{1,z}\|_{L^\infty}^{-1}$ and t already implies that $c_\psi = 1 - c_l$, because the self-similar ansatz suggests that $\|\psi_{1,z}\|_{L^\infty}^{-1} \sim (T - t)^{c_\psi + c_l}$.

mesh size	$1/\|\omega\|_{L^\infty}$		$1/\|\psi_{1,z}\|_{L^\infty}$	
	$10^3 \times b$	R^2	$10^3 \times b$	R^2
1024×1024	1.6529356	0.99987	1.6529325	0.99999
1280×1280	1.6527953	1.00000	1.6528189	1.00000
1536×1536	1.6525824	1.00000	1.6527396	1.00000

mesh size	Z_1			$1/\|\omega_1\|_{L^\infty}$		
	c	$10^3 \times b$	R^2	c	$10^3 \times b$	R^2
1024×1024	4.20	1.6529889	0.99994	1.41	1.6530613	0.99986
1280×1280	4.21	1.6527877	0.99999	1.42	1.6527894	1.00000
1536×1536	4.25	1.6526864	1.00000	1.41	1.6526953	1.00000

Table 2.4: Fitting results of $\|\omega\|_{L^\infty}^{-1}$, $\|\psi_{1,z}\|_{L^\infty}^{-1}$, Z_1 and $\|\omega_1\|_{L^\infty}^{-1}$ at different mesh sizes.

Finally, we perform the above fitting work in different spatial resolutions, and summarize the results in Table 2.4. We can see that the fitting has excellent quality at all spatial resolutions, and the fitted parameters are consistent across different spatial resolutions.

2.4 Dynamic Rescaling Method with Operator Splitting

The Dynamic Rescaling Method

As we have observed in Section 2.2, if we pin the magnitude of maximum of $|\omega_1|$ to a constant, and if we pin the maximum location of $|\omega_1|$ to a fixed point, then the profile of ω_1 seems to remain unchanged over time. This implies that the 3D Euler equations might have a self-similar blow-up with the given initial data (2.12).

In order to better study the potential self-similar singularity, we add extra scaling terms to (2.2) and write

$$\tilde{\omega}_{1,\tau} + \left(\tilde{c}_l \xi + \tilde{u}^{\xi}\right) \tilde{\omega}_{1,\xi} + \left(\tilde{c}_l \zeta + \tilde{u}^{\zeta}\right) \tilde{\omega}_{1,\zeta} = \left(c_{\omega} - (1-\alpha)\tilde{\psi}_{1,\zeta}\right) \tilde{\omega}_1, \tag{2.13a}$$

$$-\tilde{\psi}_{1,\xi\xi} - \tilde{\psi}_{1,\zeta\zeta} - \frac{3}{\xi}\tilde{\psi}_{1,\xi} = \tilde{\omega}_1 \xi^{\alpha-1}, \tag{2.13b}$$

$$\tilde{u}^{\xi} = -\xi\tilde{\psi}_{1,\zeta}, \quad \tilde{u}^{\zeta} = 2\tilde{\psi}_1 + \xi\tilde{\psi}_{1,\xi}, \tag{2.13c}$$

where $\tilde{c}_l = \tilde{c}_l(\tau)$, $\tilde{c}_{\omega} = \tilde{c}_{\omega}(\tau)$ are scalar functions of τ. In (2.13a), the terms $\tilde{c}_l \xi \partial_{\xi}$ and $\tilde{c}_l \zeta \partial_{\zeta}$ stretch the solutions in space to counter the self-similar focusing effect. The term $\tilde{c}_{\omega}\tilde{\omega}_1$ rescales the magnitude of $\tilde{\omega}_1$. The combined effect of these terms dynamically rescales the solutions to capture the potential self-similar singularity. Such dynamic rescaling strategy has widely been used in the study of singularity formation of nonlinear Schrödinger equations as in [73, 61, 63, 60, 77]. And recently it was also used in the study of singularity formation of the 3D Euler and Navier-Stokes equations as in [44, 11, 12].

If we define

$$\tilde{c}_{\psi}(\tau) = \tilde{c}_{\omega}(\tau) + (1+\alpha)\tilde{c}_l(\tau), \tag{2.14}$$

we can check that (2.13) admits the following solution

$$\begin{aligned} \tilde{\omega}_1(\xi,\zeta,\tau) &= \tilde{C}_{\omega}(\tau)\omega_1\left(\tilde{C}_l(\tau)\xi, \tilde{C}_l(\tau)\zeta, t(\tau)\right), \\ \tilde{\psi}_1(\xi,\zeta,\tau) &= \tilde{C}_{\psi}(\tau)\psi_1\left(\tilde{C}_l(\tau)\xi, \tilde{C}_l(\tau)\zeta, t(\tau)\right), \end{aligned} \tag{2.15}$$

where (ω_1, ψ_1) is the solution to (2.2), and

$$\begin{aligned} \tilde{C}_{\omega}(\tau) &= \exp\left(\int_0^{\tau} \tilde{c}_{\omega}(s)\mathrm{d}s\right), \quad \tilde{C}_{\psi}(\tau) = \exp\left(\int_0^{\tau} \tilde{c}_{\psi}(s)\mathrm{d}s\right), \\ \tilde{C}_l(\tau) &= \exp\left(-\int_0^{\tau} \tilde{c}_l(s)\mathrm{d}s\right), \quad t'(\tau) = \tilde{C}_{\psi}(\tau)\tilde{C}_l(\tau) = \tilde{C}_{\omega}(\tau)\tilde{C}_l^{-\alpha}(\tau). \end{aligned}$$

The new equations (2.13) leave us with two degrees of freedom: we are free to choose $\{\tilde{c}_l(\tau), \tilde{c}_{\omega}(\tau)\}$. This allows us to impose the following normalization conditions

$$\tilde{\omega}_1(0,1,\tau) = -1, \quad \tilde{\omega}_{1,\zeta}(0,1,\tau) = 0, \quad \text{for } \tau \geq 0. \tag{2.16}$$

One way to enforce the normalization conditions, as used in many literatures like [44, 67], is to first enforce them at $\tau = 0$ using the scaling invariant relation (2.4), and then enforce their time derivatives to be zero

$$\frac{\partial}{\partial\tau}\tilde{\omega}_1(0,1,\tau) = 0, \quad \frac{\partial}{\partial\tau}\tilde{\omega}_{1,\zeta}(0,1,\tau) = 0, \quad \text{for } \tau \geq 0. \tag{2.17}$$

Using (2.13a), the above conditions are equivalent to

$$\begin{aligned}\tilde{c}_l(\tau) =& -2\tilde{\psi}_1(0,1,\tau) - (1-\alpha)\tilde{\psi}_{1,\zeta\zeta}(0,1,\tau)\frac{\tilde{\omega}_1(0,1,\tau)}{\tilde{\omega}_{1,\zeta\zeta}(0,1,\tau)},\\ \tilde{c}_\omega(\tau) =& (1-\alpha)\tilde{\psi}_{1,\zeta}(0,1,\tau).\end{aligned} \tag{2.18}$$

However, it is hard to evaluate (2.18) accurately, because it requires calculating second-order derivatives. More importantly, due to the complicated nonlinear nature of (2.13a), even if (2.18) can be accurately evaluated, the temporal discretization (Runge-Kutta method) makes it difficult to enforce (2.16) exactly for the next time step. As a result, imposing (2.17) is not as helpful to preserve the normalization conditions (2.16) in the following time steps. The maximum magnitude and location will gradually change in time, which makes it difficult to compute the self-similar profile numerically.

Operator Splitting

To enforce the normalization conditions (2.16) accurately at every time step, we utilize the operator splitting method and rewrite (2.13a) as

$$\tilde{\omega}_{1,\tau} = F(\tilde{\omega}_1) + G(\tilde{\omega}_1), \tag{2.19}$$

where

$$F(\tilde{\omega}_1) = -\tilde{u}^\xi\tilde{\omega}_{1,\xi} - \tilde{u}^\zeta\tilde{\omega}_{1,\zeta} - (1-\alpha)\tilde{\psi}_{1,\zeta}\tilde{\omega}_1,$$

contains the original terms in (2.2a), and

$$G(\tilde{\omega}_1) = -\tilde{c}_l\xi\tilde{\omega}_{1,\xi} - \tilde{c}_l\zeta\tilde{\omega}_{1,\zeta} + \tilde{c}_\omega\tilde{\omega}_1,$$

is the linear part that controls the rescaling. Here we view $\tilde{\psi}_1$ as a function of $\tilde{\omega}_1$ through the Poisson equation (2.13b). The operator splitting method allows us to solve (2.13a) by solving $\tilde{\omega}_{1,\tau} = F(\tilde{\omega}_1)$ and $\tilde{\omega}_{1,\tau} = G(\tilde{\omega}_1)$ alternatively.

We can use the standard Runge-Kutta method to solve $\tilde{\omega}_{1,\tau} = F(\tilde{\omega}_1)$. As for $\tilde{\omega}_{1,\tau} = G(\tilde{\omega}_1)$, we notice that there is a closed form solution for the initial value problem

$$\tilde{\omega}_1(\xi,\zeta,\tau) = \tilde{C}_\omega(\tau)\tilde{\omega}_1\left(\tilde{C}_l(\tau)\xi, \tilde{C}_l(\tau)\zeta, 0\right), \tag{2.20}$$

where $\tilde{C}_\omega(\tau) = \exp\left(\int_0^\tau \tilde{c}_\omega(s)\mathrm{d}s\right)$ and $\tilde{C}_l(\tau) = \exp\left(-\int_0^\tau \tilde{c}_l(s)\mathrm{d}s\right)$.

In the first step, solving $\tilde{\omega}_{1,\tau} = F(\tilde{\omega}_1)$ will violate the normalization conditions (2.16). But we will correct this error in the second step by solving $\tilde{\omega}_{1,\tau} = G(\tilde{\omega}_1)$

with a smart choice of $\tilde{C}_l$ and $\tilde{C}_\omega$ in (2.20). In other words, at every time step when we solve $\tilde{\omega}_{1,\tau} = G(\tilde{\omega}_1)$, we can perfectly enforce (2.16) by properly choosing $\tilde{C}_l$ and $\tilde{C}_\omega$ in (2.20). We could also adopt the Strang's splitting [87] for better temporal accuracy.

Numerical Settings

Now we numerically solve the dynamic rescaling formulation (2.13). For the initial condition, we use the result from the final iteration of the adaptive mesh method in Section 2.2, and use the relation (2.5) to enforce the normalization conditions (2.17). Now that the maximum location of $\tilde{\omega}_1$ is pinned at $(\xi, \zeta) = (0, 1)$, we focus on a large computational domain

$$\mathcal{D}' = \left\{(\xi, \zeta) : 0 \le \xi \le 1 \times 10^5, 0 \le \zeta \le 5 \times 10^4\right\}.$$

This choice of the computational domain implies that the dynamic rescaling formulation effectively solves the original equations in the domain $(r, z) \in [0, 100000Z_1] \times [0, 50000Z_1]$.

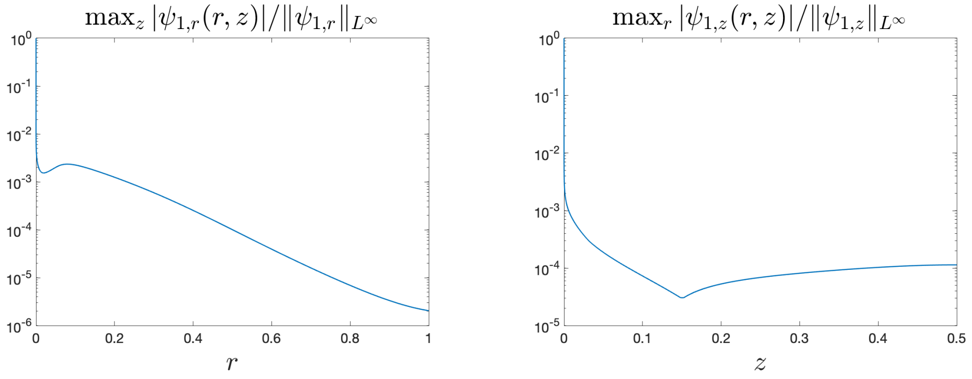


Figure 2.18: Decay of the derivatives of ψ_1.

We adopt the boundary conditions and symmetry of (2.2) in Section 2.1, except the far field boundary conditions for $\tilde{\psi}_1$. Due to extra stretching terms, the far field boundary for $\tilde{\psi}_1$ will no longer correspond to the far field boundary for ψ_1, namely $r = 1$ and $z = 1/2$. However, we notice that $\psi_{1,r}$ decays rapidly with respect to r, and $\psi_{1,z}$ decays rapidly with respect to z. For example, Figure 2.18 shows the decay of $\psi_{1,r}$ as $r \longrightarrow 1$ and the decay of $\psi_{1,z}$ as $z \longrightarrow 1/2$ for the solution to (2.2) at $t = 1.6524635 \times 10^{-3}$. Therefore, it is reasonable to impose the zero Neumann boundary condition at the far field boundaries of $\mathcal{D}'$: $\xi = 100000$ and $\zeta = 50000$. Due to the size of the computation domain $\mathcal{D}'$ and the presence of the vortex stretching terms, the error introduced by this boundary condition will have little influence on the near field around $(\xi, \zeta) = (0, 1)$.

We remark that we still need the adaptive mesh in r- and z-direction, because we not only need to cover a vary large field, but also need to focus around $(\xi, \zeta) = (0, 1)$. The adaptive mesh in solving the dynamic rescaling formulation will not change during the computation, since the dynamically rescaled vorticity has its maximum location fixed at $(\xi, \zeta) = (0, 1)$ for all times instead of traveling toward the origin.

Convergence to the Steady State

We solve (2.13) until it converges to a steady state. In Figure 2.19, we monitor how the normalization conditions (2.17) are enforced. The two normalized quantities are visually fixed at 1. In fact, they deviate from 1 by less than 5.14×10^{-4}. In Figure 2.20, we view the system (2.13) as an ODE of $\tilde{\omega}_1$ as in (2.19), and plot the relative strength of the time derivative

$$\frac{\|\tilde{\omega}_{1,\tau}\|_{L^\infty}}{\|\tilde{\omega}_1\|_{L^\infty}} = \frac{\|F(\tilde{\omega}_1) + G(\tilde{\omega}_1)\|_{L^\infty}}{\|\tilde{\omega}_1\|_{L^\infty}},$$

as a function of time τ. This relative strength of the time derivative has a decreasing trend and drops below 8.18×10^{-6} near the end of the computation, which implies that we are very close to the steady state.

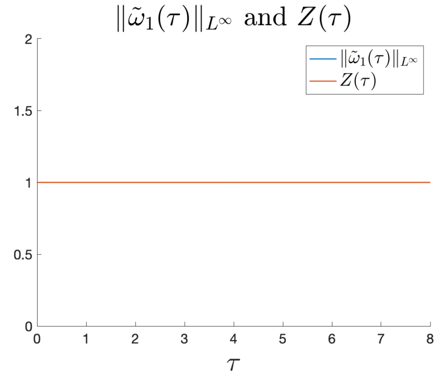


Figure 2.19: Curves of the normalized quantities $\|\tilde{\omega}_1(\tau)\|_{L^\infty}$ and $Z_1(\tau)$.

When the solution of (2.13) converges to a steady state, $\tilde{\omega}_1$ and $\tilde{\psi}_1$ should be independent of the time τ. Therefore, we should have the following relation from (2.15)

$$\omega_1(r, z, t) \sim \tilde{C}_\omega^{-1}(\tau(t))\tilde{\omega}_1\left(\tilde{C}_l^{-1}(\tau(t))r, \tilde{C}_l^{-1}(\tau(t))z\right),$$

$$\psi_1(r, z, t) \sim \tilde{C}_\psi^{-1}(\tau(t))\tilde{\psi}_1\left(\tilde{C}_l^{-1}(\tau(t))r, \tilde{C}_l^{-1}(\tau(t))z\right),$$

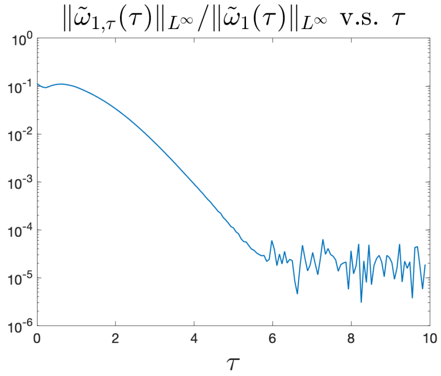


Figure 2.20: Curve of the relative time derivative strength $\|\tilde{\omega}_{1,\tau}(\tau)\|_{L^\infty}/\|\tilde{\omega}_1(\tau)\|_{L^\infty}$.

where $\tau = \tau(t)$ is the rescaled time variable. Comparing the above relation with the ansatz stated in (2.3), we conclude that

$$c_l = -\frac{\tilde{c}_l}{\tilde{c}_\omega + \alpha\tilde{c}_l}, \quad c_\omega = \frac{\tilde{c}_\omega}{\tilde{c}_\omega + \alpha\tilde{c}_l}, \quad c_\psi = \frac{\tilde{c}_\psi}{\tilde{c}_\omega + \alpha\tilde{c}_l}. \tag{2.21}$$

We remark that assuming (2.14), the above relation naturally guarantees that the scaling relation (2.6) holds true.

In Figure 2.21, we show the curves of scaling factors $\tilde{c}_l$, $\tilde{c}_\omega$ for the dynamic rescaling formulation (2.13) and c_l, c_ω for the self-similar ansatz (2.3). We observe a relatively fast convergence to the steady state as time increases. The converged values $c_l = 4.549$ and $c_\omega = 1.455$ are close to the approximate values obtained in Section 2.3. Moreover, they also satisfy the relation (2.6).

The approximate steady states of $\tilde{\omega}_1$ and $\tilde{\psi}_1$ are plotted in Figure 2.22. We see that both $\tilde{\omega}_1$ and $\tilde{\psi}_1$ are quite smooth in ξ, suggesting a possible 1D structure of their profiles. While both functions have weak dependence on ξ, $-\tilde{\omega}_1$ seems to tilt up around $\xi = 0$ a little bit. The shape of the steady states looks similar to the shape of the profiles we obtained via the adaptive mesh at the stopping time in Figure 2.6.

2.5 Hölder Exponent in Potential Blow-Up Formation

Intentionally redacted.

2.6 Anisotropic Scaling of the Potential Self-Similar Solutions

Intentionally redacted.

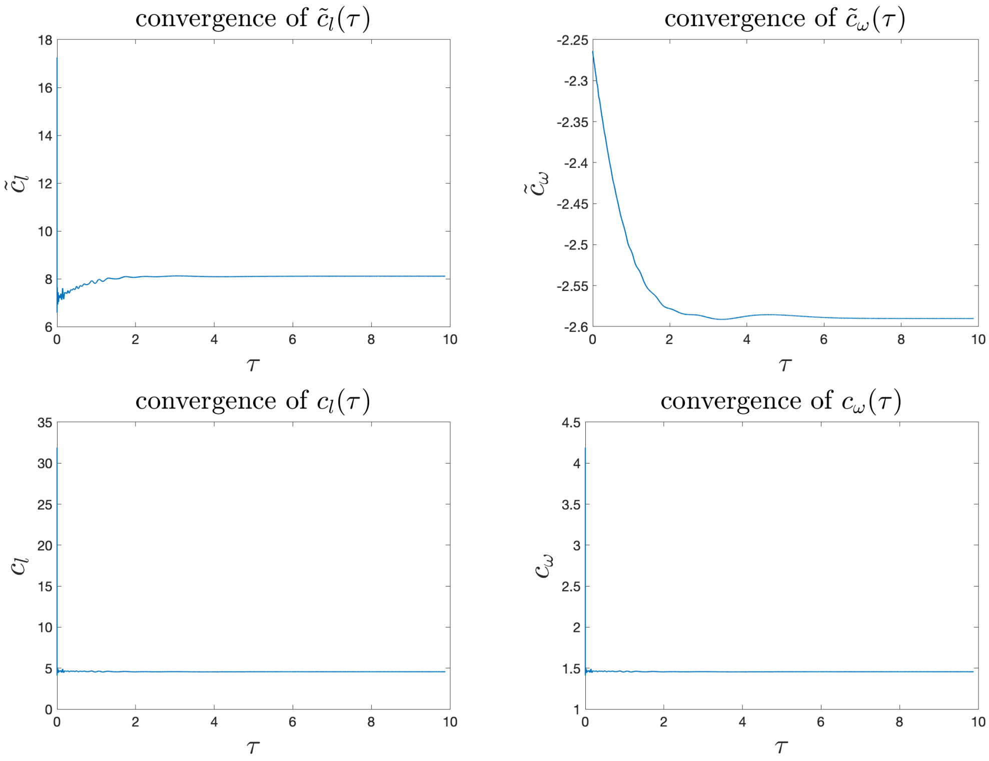


Figure 2.21: Convergence curves of the scaling factors using dynamic rescaling method. Top row: $\tilde{c}_l$ and $\tilde{c}_\omega$. Bottom row: c_l and c_ω.

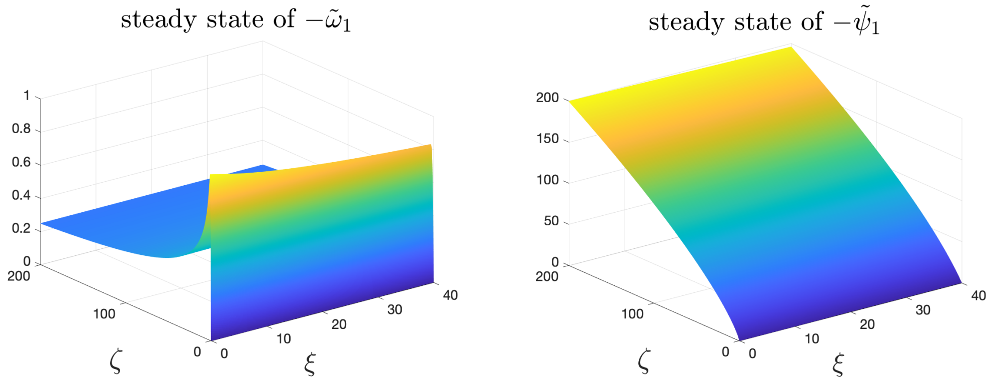


Figure 2.22: Steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$.

2.7 Comparison with Elgindi's Singularity Formation

In this section, we compare our blow-up scenario with the scenario in [30] studied by Elgindi.

Elgindi introduced a polar coordinate system on the (r,z)-plane to construct his blow-up solution. More specifically, he introduced

$$\rho = \sqrt{r^2+z^2}, \quad \theta = \arctan\left(\frac{z}{r}\right).$$

Then for a Hölder exponent α, he introduced a change of variable $R=\rho^\alpha$ and define the variables

$$\Omega(R,\theta) = \omega^\theta(r,z), \quad \Psi(R,\theta) = \frac{1}{\rho^2}\psi^\theta(r,z).$$

In this setting, (**??**) can be rewritten as

$$\Omega_t + (3\Psi + \alpha R\Psi_R)\,\Omega_\theta - (\Psi_\theta - \Psi\tan\theta)\,\Omega_R = (2\Psi\tan\theta + \alpha R\Psi_R\tan\theta + \Psi_\theta)\,\Omega, \tag{2.22a}$$

$$-\alpha^2R^2\Psi_{RR} - \alpha(5+\alpha)R\Psi_R - \Psi_{\theta\theta} + (\Psi\tan\theta)_\theta - 6\Psi = \Omega. \tag{2.22b}$$

We remark that we do not adopt the convention (**??**') as in [30], instead we stick to our convention (**??**).

Elgindi's analysis of (2.22b) establishes the following leading order approximation for small α

$$\Psi(R,\theta) = \frac{1}{4\alpha}\sin(2\theta)L_{12}(\Omega)(R) + \text{lower order terms}, \tag{2.23}$$

where

$$L_{12}(\Omega)(R) = \int_R^\infty \int_0^{\frac{\pi}{2}} \Omega(s,\theta)\frac{K(\theta)}{s}\mathrm{d}s\mathrm{d}\theta,$$

with $K(\theta) = 3\sin\theta\cos^2\theta$. If we plug in the approximation (2.23) to (2.22a), neglecting lower order terms of α, and (time) scaling out some constant factor, we arrive at Elgindi's fundamental model

$$\Omega_t = \frac{1}{\alpha}L_{12}(\Omega)\Omega, \tag{2.24}$$

which admits self-similar finite-time blow-up. In his analysis, Elgindi chose the following self-similar solution of the fundamental model (2.24)

$$\Omega(R,\theta,t) = \frac{c}{1-t}F\left(\frac{R}{1-t}\right)\left(\sin\theta\cos^2\theta\right)^{\alpha/3}, \tag{2.25}$$

where $c>0$ is some fixed constant, and $F(z) = 2z/(1+z)^2$.

One difference between our blow-up scenario and Engindi's blow-up scenario is how the scaling factor c_l depends on α. We rewrite (2.25) as

$$\begin{aligned}\Omega &= \frac{c}{1-t}F\left(\frac{\rho^\alpha}{1-t}\right)\left(\frac{r^2z}{\rho^3}\right)^{\alpha/3} \\ &= \frac{c}{1-t}F\left(\left(\frac{\rho}{(1-t)^{1/\alpha}}\right)^\alpha\right)\left(\frac{r^{2/3}z^{1/3}}{\rho}\right)^\alpha.\end{aligned}$$

If we let $G(z) = F(z^\alpha)$, we see

$$\Omega = \frac{c}{1-t} G\left(\frac{\rho}{(1-t)^{1/\alpha}}\right)\left(\frac{r^{2/3}z^{1/3}}{\rho}\right)^\alpha .$$

Since $r^{2/3}z^{1/3}/\rho$ is homogeneous, we may conclude that the scaling factors for the self-similar blow-up solution (2.25) are

$$c_l = 1/\alpha, \quad c_\omega = 2.$$

Note that this also satisfies the relation $c_\omega = 1 + \alpha c_l$ in (2.6). This implies that c_l decreases as α increases, and c_l will tend to infinity as $\alpha \longrightarrow 0$. However, as shown in Figure **??**, our c_l increases as α increases, and $c_l \longrightarrow +\infty$ when α tends to some α^* below $1/3$, and α^* is approaching $1/3$ as the parameter δ approaches zero.

Furthermore, the regularity of our initial data as a function of ρ is different from that of Elgindi's initial data. Around $(r, z) = (0, 0)$, Elgindi's initial condition has the following leading order behavior

$$\Omega \sim \rho^\alpha \left(\sin\theta\cos^2\theta\right)^{\alpha/3} = r^{2\alpha/3}z^{\alpha/3}.$$

However, our initial condition gives

$$\omega^\theta = r^\alpha \omega_1^\circ \sim r^\alpha z = \rho^{1+\alpha}\cos^\alpha\theta\sin\theta.$$

These two leading order behaviors differ from each other in that

- Elgindi's initial condition has a C^α Hölder continuity in ρ, whereas ours is $C^{1,\alpha}$ smooth,
- Elgindi's initial condition has a Hölder continuity near $z = 0$, whereas ours is smooth in z.

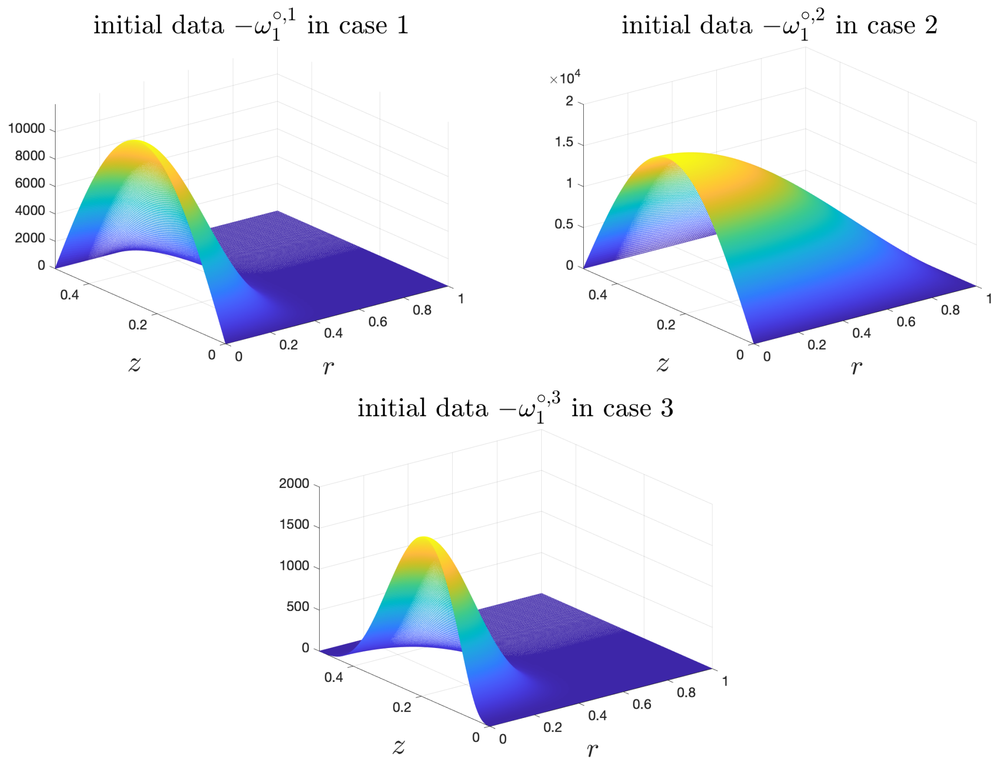
In Conjecture 8 of [27], the authors conjectured that the initial data could be C^∞ in ρ for finite-time blow-up of the 3D axisymmetric Euler equations with no swirl. Our initial data slightly improves the regularity of the initial data in ρ. In the next Section, we will briefly explore the initial data with higher regularity in ρ.

In Lemma 4.33 of [27], the authors stated that the limiting equations at $\alpha = 0$ of (2.22), can blow up in finite time for initial data of Ω that only has a C^α-Hölder continuity near $r = 0$ for $\alpha < 1/3$. Our study shows that even for the original Euler equations, it is not essential to have Hölder continuity in the initial data along the z-direction. The essential driving force for the finite-time blow-up comes from the Hölder continuity of the initial vorticity along the r-direction.

2.8 Sensitivity of the Potential Blow-Up to Initial Data

We also study the sensitivity of the potential self-similar blow-up to initial data. In addition to the initial data (2.12), we consider the following cases,

$$\begin{aligned} \omega_1^{\circ,1} &= -12000\left(1-r^2\right)^{18}\sin(2\pi z), \\ \omega_1^{\circ,2} &= -6000\cos\left(\frac{\pi r}{2}\right)\sin(2\pi z)\left(2+\exp\left(-r^2\sin^2(\pi z)\right)\right), \\ \omega_1^{\circ,3} &= \frac{-12000\,(1-r^2)^{18}\sin(2\pi z)^3}{1+12.5\sin^2(\pi z)}. \end{aligned} \tag{2.26}$$


initial data $-\omega_1^{\circ,1}$ in case 1

initial data $-\omega_1^{\circ,2}$ in case 2

initial data $-\omega_1^{\circ,3}$ in case 3

Figure 2.23: Profiles of the initial data in all three cases.

We show the profiles of these three initial data in Figure 2.23. Here, in case 1, $\omega_1^{\circ,1}$ is a perturbation of ω_1° by setting the denominator to be 1. In case 2, $\omega_1^{\circ,2}$ has a decay rate in r slower than $\left(1-r^2\right)^{18}$, and is no longer a tensor product of r and z. In case 3, $\omega_1^{\circ,3}$ has an improved regularity in ρ near the origin. Indeed, we have, with $\omega_1(r,z,0) = \omega_1^{\circ,3}(r,z)$,

$$\omega^\theta(r,z,0) = r^\alpha \omega_1^{\circ,3}(r,z) \sim r^\alpha z^3 = \rho^{3+\alpha}\cos^\alpha\theta\sin^3\theta.$$

For all three cases, we only solve the 3D Euler equations with $\alpha = 0.3$ and $\delta = 1$, due to the limited computational resources. As shown in Table **??**, for our original initial data, $c_l = 112.8$ is already very large. It suggests that our choice of α and δ is very close to the borderline between the blow-up and non-blow-up. If the blow-up profile of the above initial data agrees with our original initial data well, we then have good confidence that they should have the same behavior for other settings of α and δ.

We solve the 3D Euler equations with the above initial data by first using the adaptive mesh method to get close enough to the potential blow-up time, and then using the dynamic rescaling method to capture the potential self-similar solution.

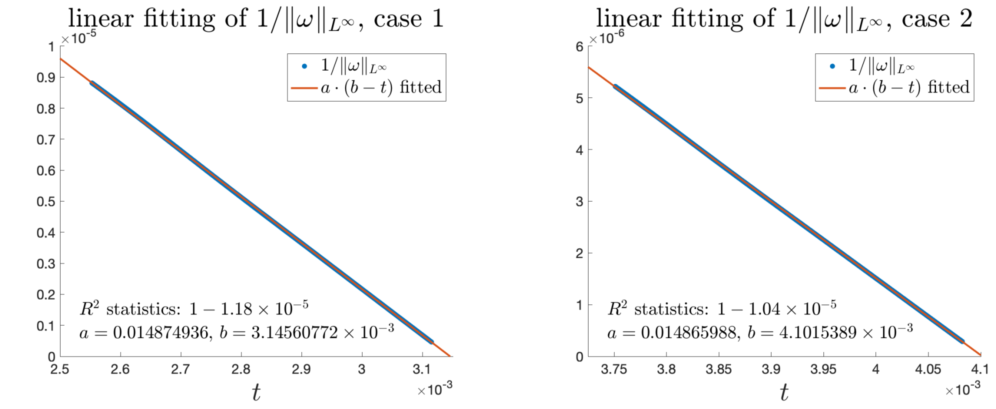


Figure 2.24: Fitting of $1/\|\omega\|_{L^\infty}$ with time t in the first and second cases.

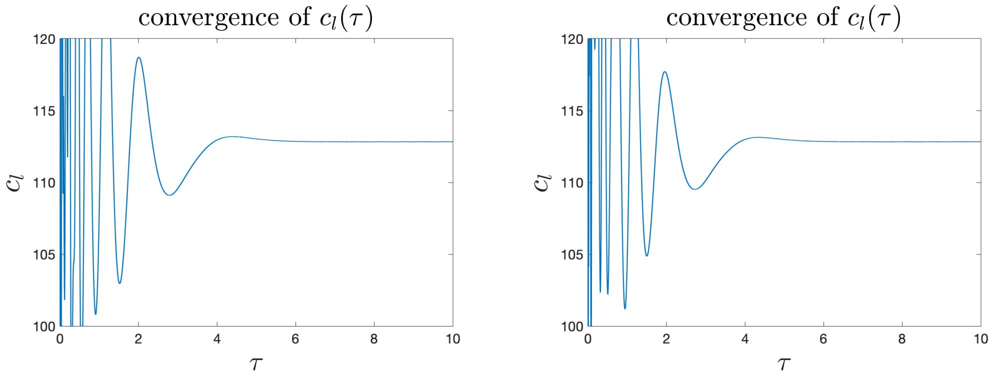


Figure 2.25: Curves of the scaling factor c_l in the first and second cases.

For the first and second cases, we show the fitting of $1/\|\omega\|_{L^\infty}$ with time t in Figure 2.24, and the curve of the scaling factor c_l in Figure 2.25. We can see that in both cases, $\|\omega\|_{L^\infty}$ scales like $1/(T-t)$, which implies a finite-time blow-up. Moreover, c_l converges to 112.8, matching the value of c_l we obtained using the original initial data well. In Figure 2.26, we show the cross sections of the steady state of $-\tilde{\omega}_1$ in comparison with the result obtained using the original initial data. There is no

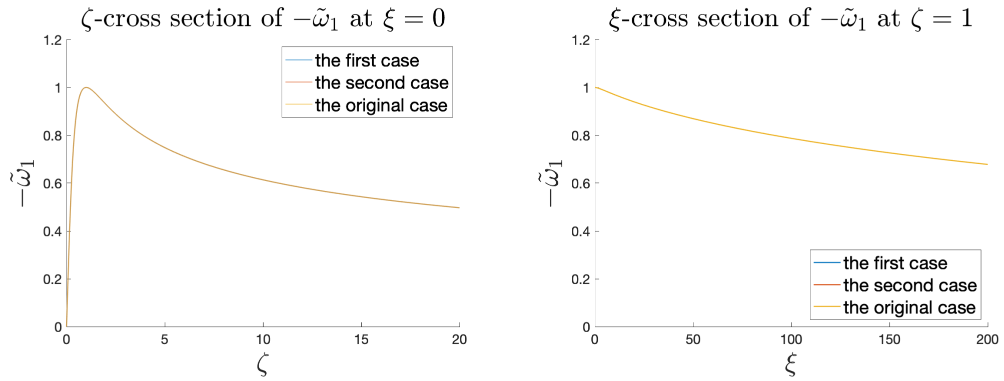


Figure 2.26: Cross sections of the steady states of $-\tilde{\omega}_1$ in the first and second cases.

visible difference between the three steady states presented. In fact, even on the whole computational domain $\mathcal{D}' = \left\{(\xi,\zeta) : 0 \leq \xi \leq 1 \times 10^5, 0 \leq \zeta \leq 5 \times 10^4\right\}$ in the dynamic rescaling computation, the steady states in the first and second cases only differ by 7.03×10^{-10} and 5.29×10^{-10} respectively from the steady state using our original initial data ω_1° in relative sup-norm sense.

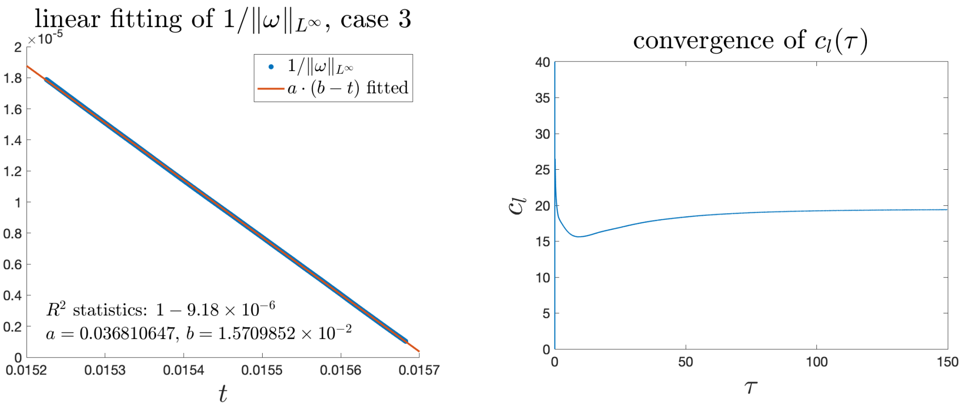


Figure 2.27: Fitting of $1/\|\omega\|_{L^\infty}$ and curve of the scaling factor c_l in the third case.

For the third case, the fitting of $1/\|\omega\|_{L^\infty}$ and the curve of the scaling factor c_l is shown in Figure 2.27. We observe that $1/\|\omega\|_{L^\infty}$ has a good linear fitting with time, suggesting a finite-time blow-up. However, c_l converges to 19.44 which is clearly different from 112.8, suggesting that there might be a new blow-up mechanism. In Figure 2.28, we compare the steady states of $\omega_1^{\circ,3}$ and ω_1° in the 3D profiles and the 2D contours. The steady state of $\omega_1^{\circ,3}$ has a slower change near $z = 0$. This might be caused by the smoothness of the initial data near $z = 0$. We have $\omega_1^{\circ,3} \sim r^\alpha z^3$, in contrast to $\omega_1^\circ \sim r^\alpha z$ near $(r,z) = (0,0)$. The steady state of the third case develops a channel-like structure that is not parallel to either axis.

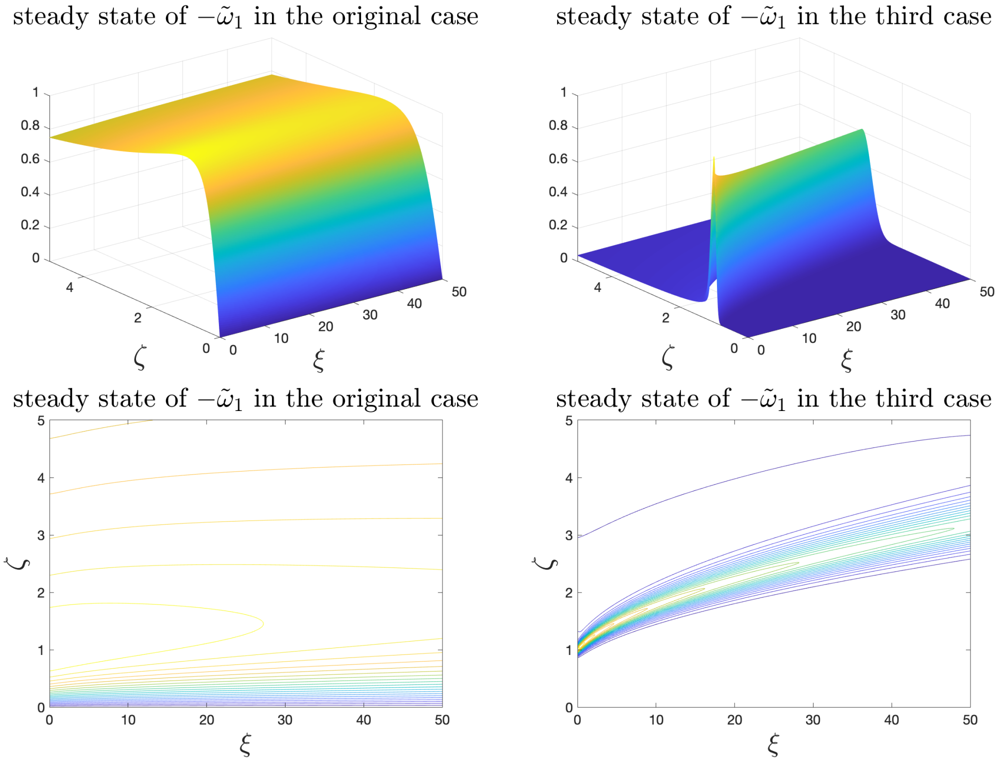


Figure 2.28: Profiles and contours of the steady states of $-\tilde{\omega}_1$ in the original and third cases.

The new blow-up scenario in the third case provides some support of Conjecture 9 of [27], in which the authors conjectured that the 3D Euler equations could still develop a finite-time blow-up for initial data that are C^∞ in ρ. In our future study, we plan to investigate the potential blow-up using a class of initial data of the form

$$\omega_1^{\circ,4} = -12000 \left(1 - r^2\right)^{18} \sin(2\pi z)^{2k+1},$$

with a positive integer k, so that $\omega_1^{\circ,4} \sim r^\alpha z^{2k+1} = \rho^{2k+1+\alpha} \cos^\alpha \theta \sin^{2k+1} \theta$ is C^{2k+1} smooth in ρ.

Chapter 3

SELF-SIMILAR FINITE-TIME SINGULARITY FORMATION FOR HÖLDER CONTINUOUS SOLUTIONS TO THE INCOMPRESSIBLE EULER EQUATIONS ON $\mathbb{R}^n$

3.1 n-D Axisymmetric Euler Equations with No Swirl

In this chapter, we study the self-similar finite-time blow-up for n-D axisymmetric Euler equations with no swirl.

To start with, we introduce the n-dimensional axisymmetric Euler equations. Let

$$u(x,t) : \mathbb{R}^n \times [0,T) \to \mathbb{R}^n,$$

be an n-D vector field of the velocity, and

$$p(x,t) : \mathbb{R}^n \times [0,T) \to \mathbb{R},$$

be an n-D scalar field of the pressure, where $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$. Then the n-D Euler equations are written as

$$u_t + u \cdot \nabla u = -\nabla p, \tag{3.1a}$$

$$\nabla \cdot u = 0. \tag{3.1b}$$

We then consider hyper-cylindrical coordinate system $(r, \theta_1, \dots, \theta_{n-2}, z)$, which is related to the Cartesian coordinate system $(x_1, x_2, \dots, x_n)$ via the following relation

$$\begin{aligned}
x_1 &= r\cos\theta_1,\\
x_2 &= r\sin\theta_1\cos\theta_2,\\
&\vdots\\
x_{n-1} &= r\sin\theta_1\cdots\sin\theta_{n-2}\\
x_n &= z.
\end{aligned}$$

We can see that the hyper-cylindrical coordinate system is simply the direct product of a $(n-1)$-D spherical coordinate system with a 1D Cartesian coordinate system. We write down the frame of the hyper-cylindrical coordinate system in the Cartesian coordinate as

$$\begin{aligned}
e_r &= (\cos\theta_1, \sin\theta_1\cos\theta_2, \dots, \sin\theta_1\cdots\cos\theta_{n-2}, \sin\theta_1\cdots\sin\theta_{n-2}, 0),\\
e_{\theta_1} &= (-\sin\theta_1, \cos\theta_1\cos\theta_2, \dots, \cos\theta_1\cdots\cos\theta_{n-2}, \cos\theta_1\cdots\sin\theta_{n-2}, 0),\\
&\vdots\\
e_{\theta_{n-2}} &= (0, 0, \dots, -\sin\theta_{n-2}, \cos\theta_{n-2}, 0),\\
e_z &= (0, 0, \dots, 0, 0, 1).
\end{aligned}$$

Similar to the 3D case, we call an n-D vector field v to be axisymmetric if the following ansatz applies

$$v = v^r(r,z)e_r + v^{\theta_1}(r,z)e_{\theta_1} + v^z(r,z)e_z,$$

in other words, v^r, v^{θ_1}, and v^z are only functions of (r,z). For such vector field, the calculus on the curvilinear coordinate [25] of $(r,\theta_1,\dots,\theta_{n-2},z)$ gives

$$\begin{aligned}
\nabla\cdot v =& v^r_r + \frac{n-2}{r}v^r + \frac{(n-3)\cot\theta_1}{r}v^{\theta_1} + v^z_z,\\
(v\cdot\nabla)\,v =& \left(v^r v^r_r + v^z v^r_z - \frac{1}{r}v^{\theta_1}v^{\theta_1}\right)e_r + \left(v^r v^{\theta_1}_r + v^z v^{\theta_1}_z + \frac{1}{r}v^r v^{\theta_1}\right)e_{\theta_1}\\
&+ \left(v^r v^z_r + v^z v^z_z\right)e_z,
\end{aligned}$$

where again we use subscripts to denote derivatives for simplicity, except that it does not apply to the unit vectors e_r, e_{θ_1}, and e_z.

We can see that if there is "swirl" $u^{\theta_1} \neq 0$ in the initial condition for dimension $n \neq 3$, then the incompressibility condition $\nabla\cdot u = 0$ will inevitably introduce the dependence on θ_1 for the equations, which implies that we cannot obtain a truly axisymmetric Euler equations for dimension greater than 3. Note that when $n = 3$, the incompressibility condition does not introduce any trouble since the third term in $\nabla\cdot v$ vanishes exactly for $n = 3$ even if there is swirl.

Therefore, to derive the n-dimensional axisymmetric Euler equations with $n > 3$, we need to impose the "no swirl" assumption $u^{\theta_1} = 0$. Luckily, the "no swirl" assumption will be preserved dynamically by the n-D Euler equations. We remark that the axisymmetric Euler equations offer tremendous computational saving, which enables us to investigate potential finite time singularity for the general n-dimensional Euler equations using our current computational resources.

Thus, the axisymmetric n-D Euler equations with no swirl can be written in the vorticity-stream function form as

$$\omega^\theta_t + u^r\omega^\theta_r + u^z\omega^\theta_z = \frac{n-2}{r}u^r\omega^\theta, \tag{3.2a}$$

$$-\psi^\theta_{rr} - \psi^\theta_{zz} - \frac{n-2}{r}\psi^\theta_r + \frac{n-2}{r^2}\psi^\theta = \omega^\theta, \tag{3.2b}$$

$$u^r = -\psi^\theta_z, \quad u^z = \frac{n-2}{r}\psi^\theta + \psi^\theta_r. \tag{3.2c}$$

Similarly, since we plan to use C^α continuous initial data for the angular vorticity ω^θ, we make the following change-of-variables

$$\omega^\theta(r,z) = r^\alpha\omega_1(r,z), \quad \psi^\theta(r,z) = r\psi_1(r,z). \tag{3.3}$$

Using (ω_1, ψ_1), an equivalent form of the n-D axisymmetric Euler equations with no swirl is given below

$$\omega_{1,t} + u^r\omega_{1,r} + u^z\omega_{1,z} = -(n-2-\alpha)\psi_{1,z}\omega_1, \tag{3.4a}$$

$$-\psi_{1,rr} - \psi_{1,zz} - \frac{n}{r}\psi_{1,r} = \omega_1 r^{\alpha-1}, \tag{3.4b}$$

$$u^r = -r\psi_{1,z}, \quad u^z = (n-1)\psi_1 + r\psi_{1,r}. \tag{3.4c}$$

Roughly speaking, the dimension n controls the strength of the vortex stretching term $-(n-2-\alpha)\psi_{1,z}\omega$ and the z-advection speed $u^z = (n-1)\psi_1 + r\psi_{1,r}$. It also modifies the Poisson equation for ψ_1. It seems natural to conjecture that the singularity formation will be more likely in the high-dimensional case because of the stronger vortex stretching.

3.2 Numerical Evidence for the Potential Blow-Up

Settings of the Problem

We provide numerical evidence for the finite-time blow-up in the high-dimensional case. We will use the setting $n = 10$, $\alpha = 0.5$ for this section. More exploration of different parameters will be studied in the Section 3.4 and 3.5.

For this 10-dimensional case, we use the same computational domain $\mathcal{D}$ for (r, z), and the same numerical solver as for the 3D Euler equations, see Appendix A. To expedite the computation, we use a modified version of the initial data (2.12),

$$\omega_1^\circ = \frac{-12000\left(1-r^2\right)^{18}\sin(2\pi z)}{1+12.5\sin^2(\pi z)}, \tag{3.5}$$

because the initial vorticity will concentrate more near the origin than (2.12).

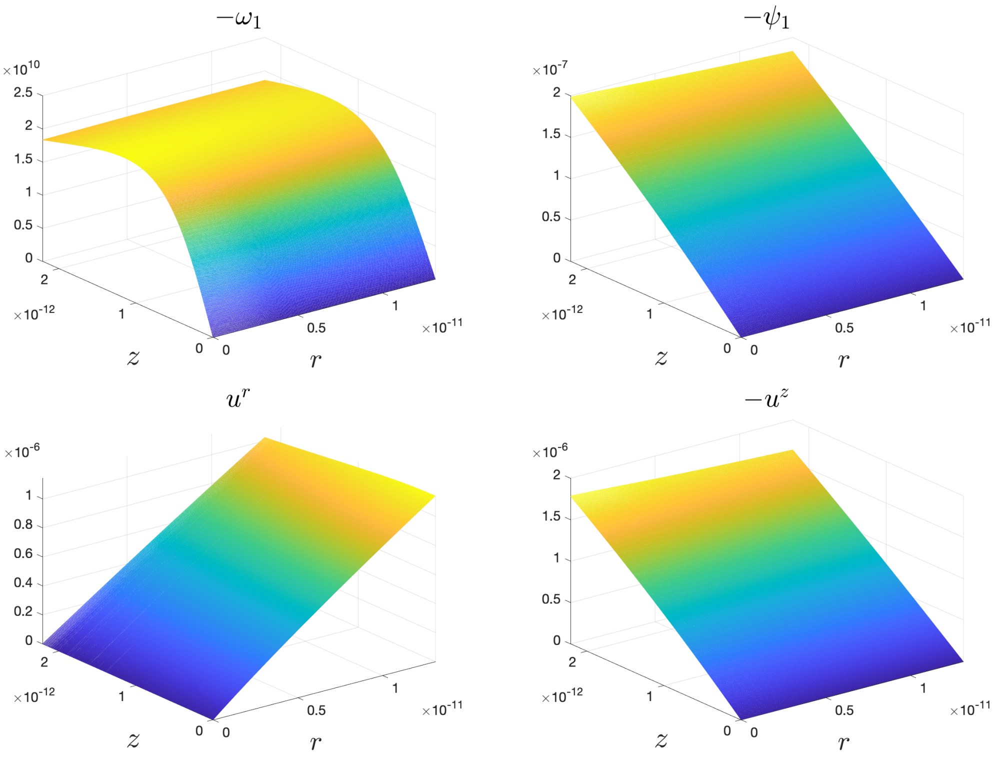


Figure 3.1: Zoomed-in 3D profiles of $-\omega_1$, $-\psi_1$, u^r and $-u^z$ at $t = 7.9582242\times10^{-4}$ near the origin $(0, 0)$.

Adaptive Mesh Method

On 1024×1024 spatial resolution, we solve the equations until $t = 7.9582242\times10^{-4}$, where the solution becomes too singular to be resolved by our numerical method. The zoomed-in profiles of $-\omega_1$, $-\psi_1$, and the velocity fields u^r, $-u^z$ are shown in 3.1. We can see that $-\omega_1$ seems to mostly depend on z instead of r. In Figure 3.2, we show the curves of important quantities of the solution. At the end of the computation, $\|\omega_1\|_{L^\infty}$ has increased by a factor of around 6.5×10^6, and $\|\omega\|_{L^\infty}$ has increased by a factor of around 515. We recall that $(R_1(t), Z_1(t))$ is the maximum location of $|\omega_1|$. We still observe that $R_1(t) = 0$, and $-\omega_1$ becomes very one-dimensional. We observe that $Z_1(t)$ decays very fast towards zero, acting like $(T - t)^c$ with some terminal time T and an exponent $c > 1$.

The double logarithm curve $\log\log\|\omega\|_{L^\infty}$ grows superlinear in time, giving us a first sign that the solution will form potential singularity in finite time. Another sign comes from the Beale-Kato-Majda blow-up criterion (**??**), as the time integral $\int_0^t \|\omega(s)\|_{L^\infty}\mathrm{d}s$ grows rapidly in time.

We remark that the kinetic energy E is also a conservative quantity in the n-

dimensional case. After more than 5.5×10^4 iterations, the change in the kinetic energy is less than 1.17×10^{-4} of its own scale. The conservation of energy provides additional support for the accuracy of our numerical solution.

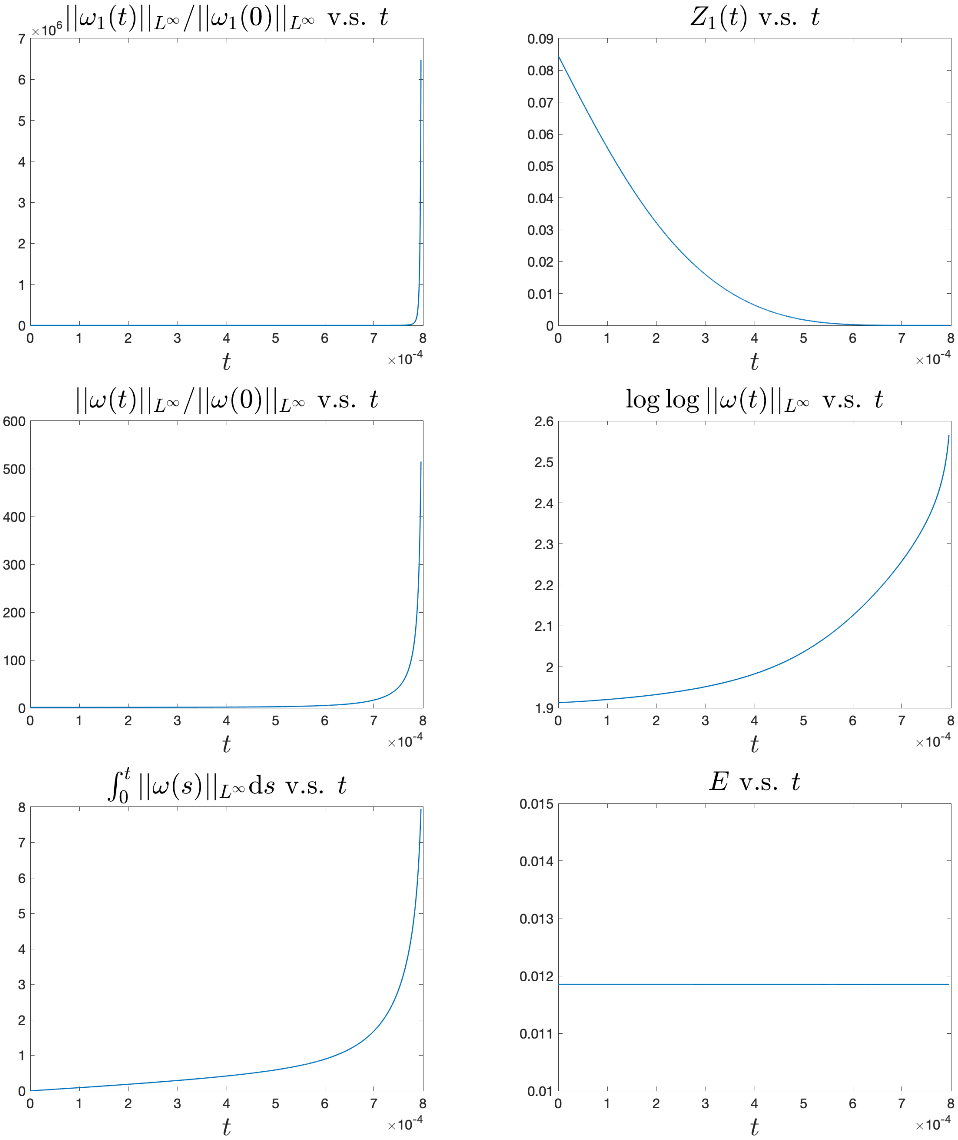


Figure 3.2: Curves of $\|\omega_1\|_{L^\infty}$, Z_1, $\|\omega\|_{L^\infty}$, $\log\log\|\omega\|_{L^\infty}$, $\int_0^t \|\omega(s)\|_{L^\infty}\mathrm{d}s$ and E as a function of time.

Scaling Analysis

In Figure 3.3, we perform scaling analysis for the potential blow-up. Similar to the 3D case, the scaling invariant property of the n-D Euler equations implies that if the self-similar blow-up exists, then $\|\omega\|_{L^\infty}$ and $\|\psi_{1,z}\|_{L^\infty}$ should scale like $1/(T-t)$, where T is the blow-up time. In the top row of Figure 3.3, $1/\|\omega\|_{L^\infty}$ or $1/\|\psi_{1,z}\|_{L^\infty}$

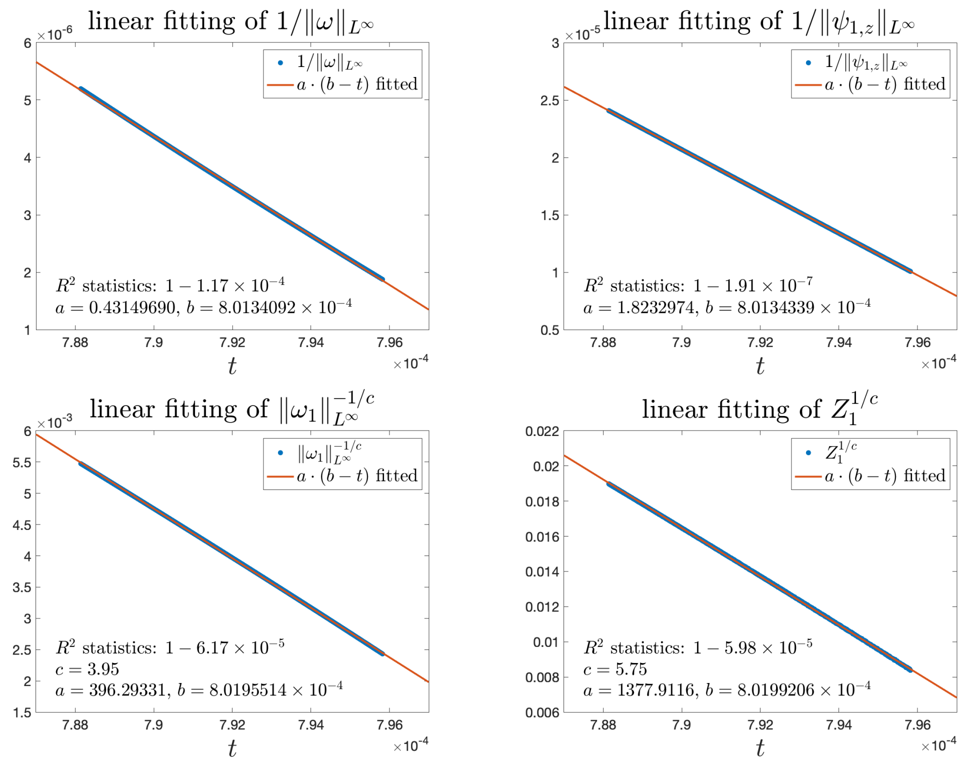


Figure 3.3: Fitting the scales of $\|\omega\|_{L^\infty}$, $\|\psi_{1,z}\|_{L^\infty}$, and $\|\omega_1\|_{L^\infty}$, Z_1.

as a function of t gives excellent linear fitting with R^2 value higher than 0.9998. The blow-up times estimated by the fitting of these two quantities also match each other very well, with one 8.0134092×10^{-4} and another 8.0134974×10^{-4}. It provides further evidence that the our 10-D Euler equations develop a potential finite-time self-similar singularity.

In the second row of Figure 3.3, we use the fitting method described in Section 2.3 to find out the scaling factors c_l and c_ω. The self-similar ansatz implies that $\|\omega_1\|_{L^\infty} \sim 1/(T-t)^{c_\omega}$ and $Z_1 \sim (T-t)^{c_l}$. Therefore, we find the best constant c for $\|\omega\|_{L^\infty}^{-1/c}$ or $Z_1^{1/c}$ such that they achieve the highest R^2 values when fitting with t. Our results give $c_l = 5.75$ and $c_\omega = 3.95$, which satisfies the relation (2.6) $c_\omega = 1 + \alpha c_l$ approximately. Moreover, the estimated blow-up times in both cases agree with each other quite well.

Dynamic Rescaling Method

The scaling analysis suggests that the potential singularity is very likely to be self-similar. Therefore, we use the dynamic rescaling method to study the potential self-similar profile of the solution. In the high-dimensional case, the dynamic rescaling formulation becomes

$$\tilde{\omega}_{1,\tau} + \left(\tilde{c}_l\xi + \tilde{u}^{\xi}\right)\tilde{\omega}_{1,\xi} + \left(\tilde{c}_l\zeta + \tilde{u}^{\zeta}\right)\tilde{\omega}_{1,\zeta} = \left(c_\omega - (n-2-\alpha)\tilde{\psi}_{1,\zeta}\right)\tilde{\omega}_1, \tag{3.6a}$$

$$-\tilde{\psi}_{1,\xi\xi} - \tilde{\psi}_{1,\zeta\zeta} - \frac{n}{\xi}\tilde{\psi}_{1,\xi} = \tilde{\omega}_1\xi^{\alpha-1}, \tag{3.6b}$$

$$\tilde{u}^{\xi} = -\xi\tilde{\psi}_{1,\zeta}, \quad \tilde{u}^{\zeta} = (n-1)\tilde{\psi}_1 + \xi\tilde{\psi}_{1,\xi}. \tag{3.6c}$$

We adopt the same settings for the computational domain $\mathcal{D}'$, the normalization conditions (2.16), and the operator splitting method (2.19) as in Section 2.4.

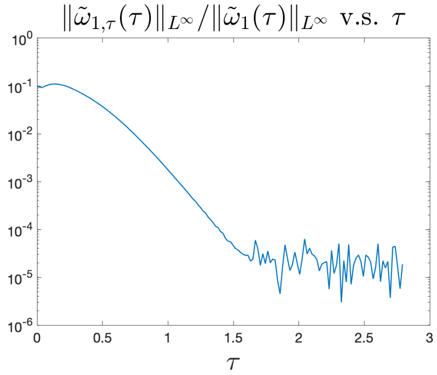


Figure 3.4: Curve of the relative time derivative strength $\|\tilde{\omega}_{1,\tau}(\tau)\|_{L^\infty}/\|\tilde{\omega}_1(\tau)\|_{L^\infty}$.

We observe fast convergence to the steady state using the solution from the last iteration of the adaptive mesh method as the initial condition for the dynamic rescaling formulation. In Figure 3.4, we plot the relative time derivative strength $\|\tilde{\omega}_{1,\tau}(\tau)\|_{L^\infty}/\|\tilde{\omega}_1(\tau)\|_{L^\infty}$. This relative strength of the time derivative has a decreasing trend and goes down below 1.78×10^{-6} near the end of the computation, which provides strong evidence that we are close to the steady state.

In Figure 3.5, we plot the curves of the scaling factors. In the top row, the scaling factors $\tilde{c}_l$, $\tilde{c}_\omega$ that are used in the dynamic rescaling method demonstrate good convergence to a constant value. In the second row, the scaling factors c_l, c_ω that appear in the self-similar ansatz converge to the value of 5.897 and 3.949 respectively. The estimated values for c_l, c_ω not only match the estimated values from the scaling analysis, but also satisfy the relation $c_\omega = 1 + \alpha c_l$ approximately. This provides another verification of the validity of our results.

The steady states of $-\tilde{\omega}_1$ and $\tilde{\psi}_1$ are plotted in Figure 3.6. We can see that $-\tilde{\omega}_1$ is very close to be one-dimensional, with little tilt towards $\xi = 0$.

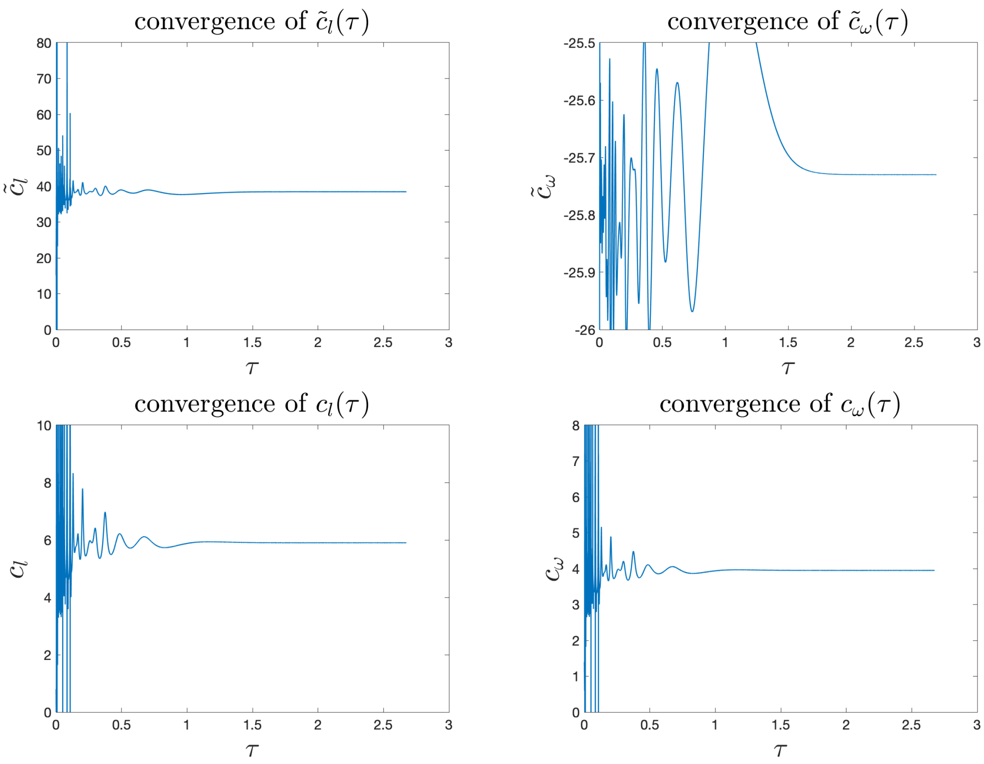


Figure 3.5: Convergence curves of the scaling factors using dynamic rescaling method. Top row: $\tilde{c}_l$; and $\tilde{c}_\omega$. Bottom row: c_l and c_ω.

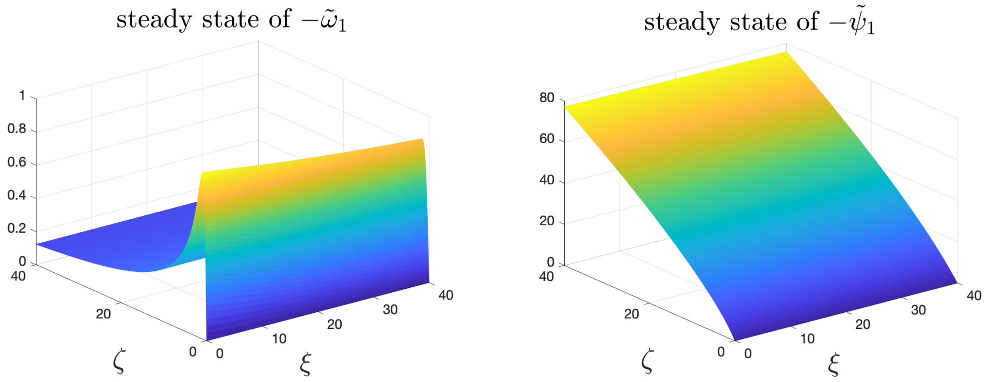


Figure 3.6: Steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$, with $n = 10$ and $\alpha = 0.5$.

3.3 Hölder Exponent, Anisotropic Scaling, and Dimension in the Potential Self-Similar Blow-Up

Intentionally redacted.

3.4 Possible Mechanism of the Potential Singularity Formation

Intentionally redacted.

3.5 A One-Dimensional Model for the Potential Self-Similar Blow-Up

Intentionally redacted.

Chapter 4

FINITE-TIME SINGULARITY FORMATION OF THE WEAK CONVECTION MODEL OF THE IMPRESSIBLE EULER EQUATIONS ON $\mathbb{R}^n$

4.1 Motivation of the Model

In this chapter, we turn to study the axisymmetric Euler equations with smooth initial data. It is known that 3D axisymmetric Euler equations with no swirl have global regularity if the initial data is sufficiently smooth [90]. In fact, as observed by [66], any smooth solution $(u^\theta, \omega^\theta, \psi^\theta)$ of the axisymmetric Euler equations (**??**) (no need to be swirl-free) must satisfy the compatible conditions:

$$u^\theta(0,z,t) = \omega^\theta(0,z,t) = \psi^\theta(0,z,t) = 0.$$

Therefore, we may define the new variables, as introduced by Hou and Li in [45]:

$$u_1 = u^\theta/r, \quad \omega_1 = \omega^\theta/r, \quad \psi_1 = \psi^\theta/r. \tag{4.1}$$

In the new variables, the vorticity equation becomes

$$\omega_{1,t} + u^r\omega_{1,r} + u^z\omega_{1,z} = 2u_1u_{1,z}.$$

If we assume the no swirl condition, the vorticity equation simplifies to

$$\omega_{1,t} + u^r\omega_{1,r} + u^z\omega_{1,z} = 0,$$

which implies that the maximum of $|\omega_1|$ is conserved. At the same time, the vorticity vector in the axisymmetric and no swirl setting can be simplified to $\omega = \omega^\theta e_\theta$. As a result, $\omega = r\omega_1 e_\theta$ would violate the Beale-Kato-Majda blow-up criterion (**??**).

The reason for the global regularity of the 3D axisymmetric Euler equations with no swirl is due to the balance of the vortex stretch term and the convection terms. In the equation of the angular vorticity ω^θ, the vortex stretching term $\frac{1}{r}u^r\omega^\theta$ gets canceled by a part of the convection term $u^r\omega^\theta_r$ after the change-of-variable $\omega^\theta = r\omega_1$.

To better understand the balancing effect between the vortex stretch term and the convection terms, we modify the 3D axisymmetric Euler equations with no swirl so that the vortex stretch term has different strength than the convection terms. We also generalize this design to the n-dimensional case. Recall that the n-D axisymmetric Euler equations with no swirl have the form

$$\omega^\theta_t + u^r\omega^\theta_r + u^z\omega^\theta_z = \frac{n-2}{r}u^r\omega^\theta, \tag{4.2a}$$

$$-\psi^\theta_{rr} - \psi^\theta_{zz} - \frac{n-2}{r}\psi^\theta_r + \frac{n-2}{r^2}\psi^\theta = \omega^\theta, \tag{4.2b}$$

$$u^r = -\psi^\theta_z, \quad u^z = \frac{n-2}{r}\psi^\theta + \psi^\theta_r. \tag{4.2c}$$

We introduce the weak convection model by adding a parameter $\varepsilon \in [0, 1]$ to control the strength of the convection terms:

$$\omega^\theta_t + \varepsilon u^r\omega^\theta_r + \varepsilon u^z\omega^\theta_z = \frac{n-2}{r}u^r\omega^\theta, \tag{4.3a}$$

$$-\psi^\theta_{rr} - \psi^\theta_{zz} - \frac{n-2}{r}\psi^\theta_r + \frac{n-2}{r^2}\psi^\theta = \omega^\theta, \tag{4.3b}$$

$$u^r = -\psi^\theta_z, \quad u^z = \frac{n-2}{r}\psi^\theta + \psi^\theta_r. \tag{4.3c}$$

If we still introduce the change-of-variable (4.1), the equations in (4.3) becomes

$$\omega_{1,t} + \varepsilon u^r\omega_{1,r} + \varepsilon u^z\omega_{1,z} = -(n-2-\varepsilon)\psi_{1,z}\omega_1, \tag{4.4a}$$

$$-\psi_{1,rr} - \psi_{1,zz} - \frac{n}{r}\psi_{1,r} = \omega_1, \tag{4.4b}$$

$$u^r = -r\psi_{1,z}, \quad u^z = (n-1)\psi_1 + r\psi_{1,r}. \tag{4.4c}$$

The case of $\varepsilon = 1$ resumes the original n-D axisymmetric Euler equations with no swirl (3.2). We can see that indeed for the 3D case, the vortex stretching term $\psi_{1,z}\omega$ vanishes when $\varepsilon = 1$.

We remark that the weak convection model (4.3) is different from the models studied in [62, 47, 44, 46]. In [62, 47], the authors directly dropped the convection terms in 3D axisymmetric Euler equations using the variables (ω_1, ψ_1). In [44, 46], the authors generalized the previous model by modifying the Biot-Savart law (4.2c) to

$$u^r = \varepsilon\left(-r\psi_{1,z}\right), \quad u^z = \varepsilon\left((n-1)\psi_1 + r\psi_{1,r}\right).$$

This modification has some good properties like preserving some modified kinetic energy. In the no swirl case, the above modification can be written in the form of the equations in our model (4.4) with only (4.4a) replaced by

$$\omega_{1,t} + \varepsilon u^r\omega_{1,r} + \varepsilon u^z\omega_{1,z} = -\varepsilon(n-3)\psi_{1,z}\omega_1.$$

In other words, the no swirl case of the model in [44, 46] only rescales the time by the parameter ε, and does not change the balance between the convection terms and the vortex stretching term. In contrast, the weak convection model (4.4) gives more degrees of freedom to study the balancing effect.

No Energy Conservation

An important difference of the weak convection model (4.4) to the original n-D axisymmetric Euler equations is that the kinetic energy is not conserved in our weak convection model. In fact, since we would like an imbalance between the convection terms and the vortex stretching terms, it would be hard to maintain the energy conservation.

Specifically, in the n-D axisymmetric setting, we define the kinetic energy E as

$$E = \frac{1}{2}\int_{\mathcal{D}} \left((u^r)^2 + (u^z)^2\right) r^{n-2}\mathrm{d}r\mathrm{d}z, \tag{4.5}$$

where $\mathcal{D}$ is the cylinder domain

$$\mathcal{D} = \{(r,z) : 0 \le r \le 1, 0 \le z \le 1/2\}.$$

We first notice that,

$$\begin{aligned}\int_{\mathcal{D}} \left((u^r)^2 + (u^z)^2\right) r^{n-2}\mathrm{d}r\mathrm{d}z &= \int_{\mathcal{D}} \left((-r\psi_{1,z})^2 + ((n-1)\psi_1 + r\psi_{1,r})^2\right) r^{n-2}\mathrm{d}r\mathrm{d}z \\ &= \int_{\mathcal{D}} \left(\psi_{1,r}^2 + \psi_{1,z}^2\right) r^n \mathrm{d}r\mathrm{d}z \\ &\quad + (n-1)\int_{\mathcal{D}} \left((n-1)\psi_1^2 r^{n-2} + 2\psi_1\psi_{1,r}r^{n-1}\right)\mathrm{d}r\mathrm{d}z.\end{aligned}$$

Using the Poisson equation (4.4b) and integration by part, we have

$$\begin{aligned}\int_{\mathcal{D}} \left(\psi_{1,r}^2 + \psi_{1,z}^2\right) r^n \mathrm{d}r\mathrm{d}z &= \int_{\mathcal{D}} \left(-\psi_{1,rr} - \psi_{1,zz} - \frac{n}{r}\psi_{1,r}\right)\psi_1 r^n \mathrm{d}r\mathrm{d}z \\ &= \int_{\mathcal{D}} \omega_1\psi_1 r^n \mathrm{d}r\mathrm{d}z.\end{aligned}$$

Furthermore, the integration by part also gives

$$\int_{\mathcal{D}} \left((n-1)\psi_1^2 r^{n-2} + 2\psi_1\psi_{1,r}r^{n-1}\right)\mathrm{d}r\mathrm{d}z = \int_{\mathcal{D}} \left(\psi_1^2 r^{n-1}\right)_r \mathrm{d}r\mathrm{d}z = 0.$$

So we have an equivalent formula for the kinetic energy E

$$E = \frac{1}{2}\int_{\mathcal{D}} \omega_1\psi_1 r^n \mathrm{d}r\mathrm{d}z.$$

Now we use (4.4a) and find

$$\begin{aligned}2\frac{\mathrm{d}}{\mathrm{d}t}E &= \int_{\mathcal{D}} \left(-\varepsilon u^r\omega_{1,r} - \varepsilon u^z\omega_{1,z} - (n-2-\varepsilon)\psi_{1,z}\omega_1\right)\psi_1 r^n \mathrm{d}r\mathrm{d}z \\ &= -\varepsilon\int_{\mathcal{D}} \left(u^r\omega_{1,r} + u^z\omega_{1,z}\right)\psi_1 r^n\mathrm{d}r\mathrm{d}z - (n-2-\varepsilon)\int_{\mathcal{D}} \psi_1\psi_{1,z}\omega_1 r^n \mathrm{d}r\mathrm{d}z \\ &= \mathrm{I} + \mathrm{II}.\end{aligned}$$

For the first term I in the right-hand side of the above equation, we do integration by part and plug in the incompressible condition $\nabla \cdot u = u^r_r + \frac{n-2}{r} u^r + u^z_z = 0$:

$$\begin{aligned}
\mathrm{I} =& -\varepsilon \int_{\mathcal{D}} \left(u^r \omega_{1,r} + u^z \omega_{1,z}\right) \psi_1 r^n \mathrm{d}r\mathrm{d}z \\
=&\varepsilon \int_{\mathcal{D}} \left(u^r_r + \frac{n}{r} u^r + u^z_z\right) \omega_1 \psi_1 r^n \mathrm{d}r\mathrm{d}z + \varepsilon \int_{\mathcal{D}} \left(u^r \psi_{1,r} + u^z \psi_{1,z}\right) \omega_1 r^n \mathrm{d}r\mathrm{d}z \\
=&\varepsilon \int_{\mathcal{D}} \frac{2}{r} u^r \omega_1 \psi_1 r^n \mathrm{d}r\mathrm{d}z + \varepsilon \int_{\mathcal{D}} \left(u^r \psi_{1,r} + u^z \psi_{1,z}\right) \omega_1 r^n \mathrm{d}r\mathrm{d}z,
\end{aligned}$$

and then we plug in the Biot-Savart law (4.4c) for u^r, u^z:

$$\begin{aligned}
\mathrm{I} =&\varepsilon \int_{\mathcal{D}} \frac{2}{r} \left(-r\psi_{1,z}\right) \omega_1 \psi_1 r^n \mathrm{d}r\mathrm{d}z \\
&+ \varepsilon \int_{\mathcal{D}} \left(\left(-r\psi_{1,z}\right) \psi_{1,r} + \left((n-1)\psi_1 + r\psi_{1,r}\right) \psi_{1,z}\right) \omega_1 r^n \mathrm{d}r\mathrm{d}z, \\
=&\varepsilon(n-3) \int_{\mathcal{D}} \psi_1 \psi_{1,z} \omega_1 r^n \mathrm{d}r\mathrm{d}z.
\end{aligned}$$

Therefore, we can conclude that

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} E =& \frac{1}{2}\varepsilon(n-3) \int_{\mathcal{D}} \psi_1 \psi_{1,z} \omega_1 r^n \mathrm{d}r\mathrm{d}z - \frac{1}{2}(n-2-\varepsilon) \int_{\mathcal{D}} \psi_1 \psi_{1,z} \omega_1 r^n \mathrm{d}r\mathrm{d}z \\
=& -\frac{1}{2}(n-2)(1-\varepsilon) \int_{\mathcal{D}} \psi_1 \psi_{1,z} \omega_1 r^n \mathrm{d}r\mathrm{d}z.
\end{aligned}$$

We can see that in general the kinetic energy E is conserved only if $n = 2$ or $\varepsilon = 1$.

4.2 Numerical Evidence for the Potential Blow-Up

Settings of the Problem

In this section, we present numerical evidence for the potential blow-up. We will use the setting $n = 3$, $\varepsilon = 0.1$ for this section. More exploration of different parameters will be studied in the Section 4.3.

We use the computational domain $\mathcal{D}$ defined in the previous section for (r, z). We impose a periodic boundary condition in z:

$$\omega_1(r, z, t) = \omega_1(r, z+1, t) = 0.$$

We enforce the initial data for ω_1 to be an odd function of z:

$$\omega_1(r, z, t) = -\omega_1(r, -z, t),$$

and this will be dynamically preserved by the equations. So we only need to solve the equations in a half period in z. We impose a no-flow boundary condition at $r = 1$:

$$\psi_1(0, z, t) = 0.$$

We use the same numerical solver as for the 3D Euler equations, see Appendix A. The initial condition for ω_1 is the same as we used in Chapter 3:

$$\omega_1^\circ = \frac{-12000\,(1-r^2)^{18}\sin(2\pi z)}{1+12.5\sin^2(\pi z)}. \tag{4.6}$$

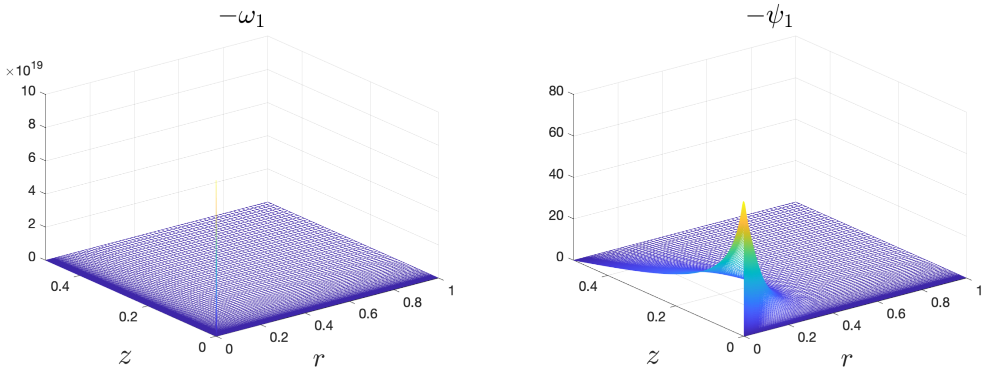


Figure 4.1: 3D profiles of $-\omega_1$, $-\psi_1$ at $t = 1.3002048 \times 10^{-2}$.

Adaptive Mesh Method

On 1024×1024 spatial resolution, we solve the equations for more than 1.3×10^5 iterations until $t = 1.3002048 \times 10^{-2}$, where the solution becomes too singular in space to be resolved by our numerical method due to the round-off error. At the end of the computation, $\|\omega_1\|_{L^\infty}$ has increased by a factor of around 2.9×10^{16}, and the maximum vorticity $\|\omega\|_{L^\infty}$ has increased by a factor of around 2.0×10^{14}. The profiles of $-\omega_1$, $-\psi_1$ are plotted in Figure 4.1. We see that $-\omega_1$ is significantly large near the origin $(r, z) = (0, 0)$. The zoomed-in profiles of $-\omega_1$, $-\psi_1$, and the velocity fields u^r, $-u^z$ are shown in Figure 4.2. It is very interesting to notice that even though the scale of the plotted r axis is 2.5×10^5 larger than the plotted z axis, $-\omega_1$, $-\psi_1$ seem to be very one-dimensional and depend on z only.

We show the curves of important quantities of the solution in Figure 4.3. We observe a clear superlinear curve for $\log\log\|\omega\|_{L^\infty}$ in time. This provides a first sign that the solution will form potential singularity in finite time. Another sign comes from the Beale-Kato-Majda blow-up criterion (**??**), as we observe a remarkably fast growth of the integral $\int_0^t \|\omega(s)\|_{L^\infty}\mathrm{d}s$. Let $Z_1(t)$ be the z-coordinate of the maximum location of $|\omega_1|$, we see that it collapses to zero very fast in time. We also observe that, since we do not have energy conservation here, the kinetic energy E grows rapidly in the late stage of the computation.

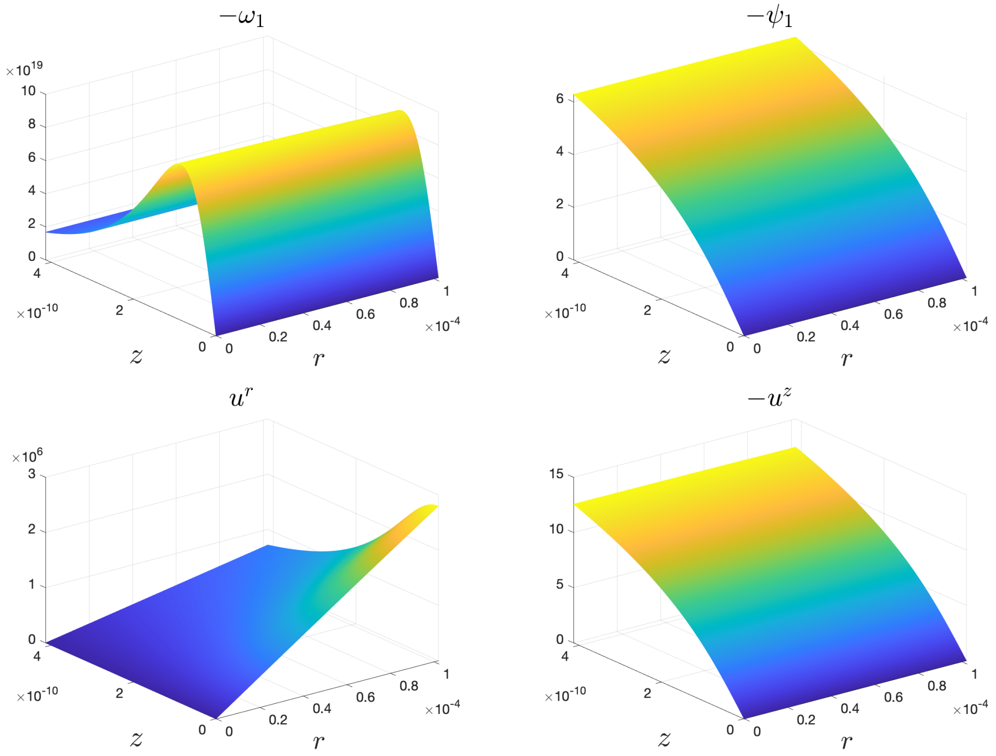


Figure 4.2: Zoomed-in 3D profiles of $-\omega_1$, $-\psi_1$, u^r, and $-u^z$ at $t = 1.3002048\times10^{-2}$ near the origin. Note that the r axis scale is 2.5×10^6 of the z axis scale.

Self-Similar Profile

Since the solution is very one-dimensional and seems to only depend on z, it is reasonable to assume that the self-similar profile, if exists, would be anisotropic. In fact, let $(R(t), Z(t))$ be the the maximum location of $|\omega^\theta|$. In other words, we have $|\omega^\theta(R(t), Z(t))| = \|\omega^\theta(t)\|_{L^\infty}$. In Figure 4.4, we plot the ratio $R(t)/Z(t)$ and the trajectory of $(R(t), Z(t))$ in time t. We can clearly see that R/Z grows very fast in late time, and the value of this ratio even goes beyond 10^7. The trajectory of $(R(t), Z(t))$ also demonstrates a clear anisotropic nature of the solution. The change of $R(t)$ in time is significantly slower than the change of $Z(t)$ in time.

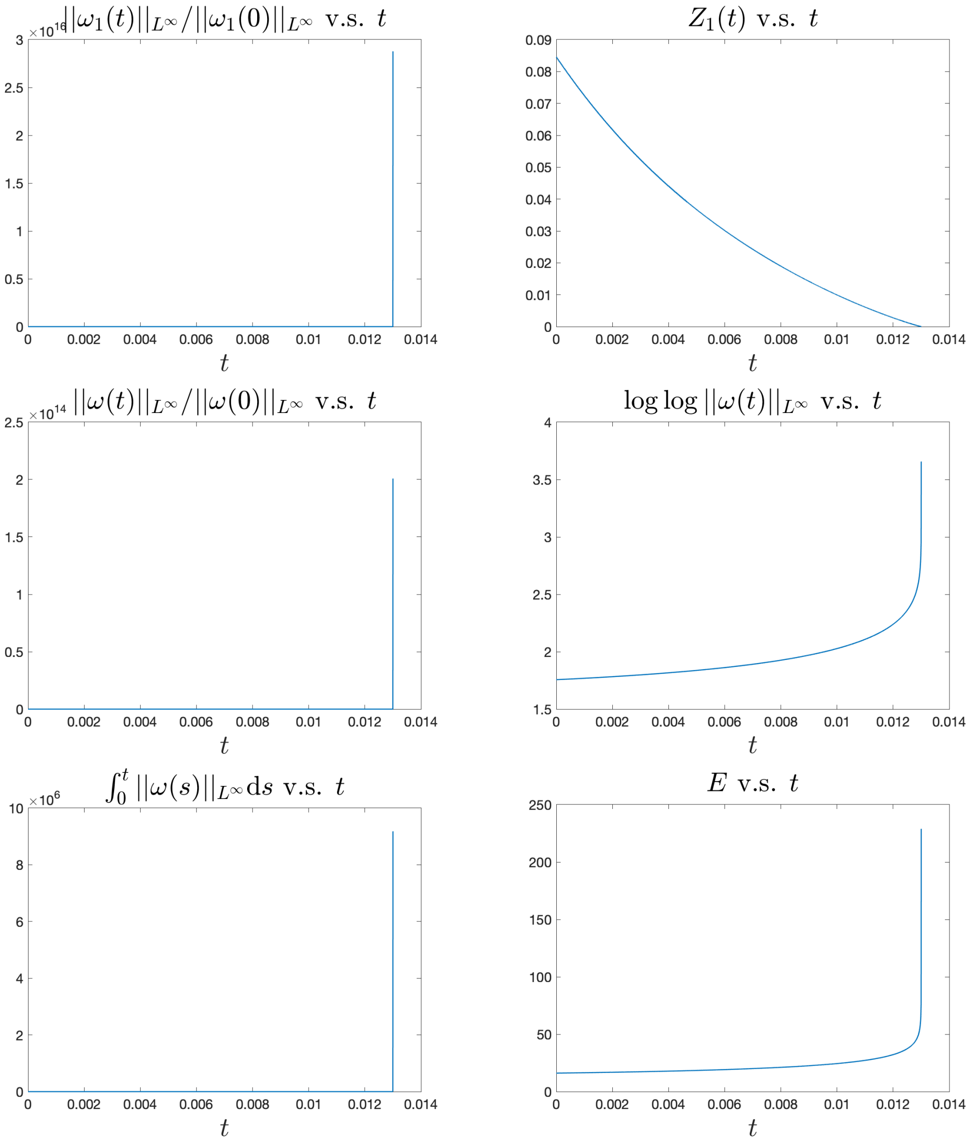


Figure 4.3: Curves of $\|\omega_1\|_{L^\infty}$, Z_1, $\|\omega\|_{L^\infty}$, $\log\log\|\omega\|_{L^\infty}$, $\int_0^t \|\omega(s)\|_{L^\infty}\mathrm{d}s$ and E as functions of time.

Therefore, instead of the self-similar ansatz (2.3), we would now assume

$$\begin{aligned}\omega_1(x,t) &\approx \frac{1}{(T-t)^{c_\omega}}\Omega\left(\frac{r}{(T-t)^{c_l'}}, \frac{z}{(T-t)^{c_l}}\right),\\ \psi_1(x,t) &\approx \frac{1}{(T-t)^{c_\psi}}\Psi\left(\frac{r}{(T-t)^{c_l'}}, \frac{z}{(T-t)^{c_l}}\right),\end{aligned} \tag{4.7}$$

with the parameter $c_l' \leq c_l$. We use the parameter c_l' to model the anisotropic behavior in r and z. We remark that it could also be possible that Ω and Ψ in (4.7) are independent of r. However, we choose the current ansatz (4.7) because it is capable to model all these cases.

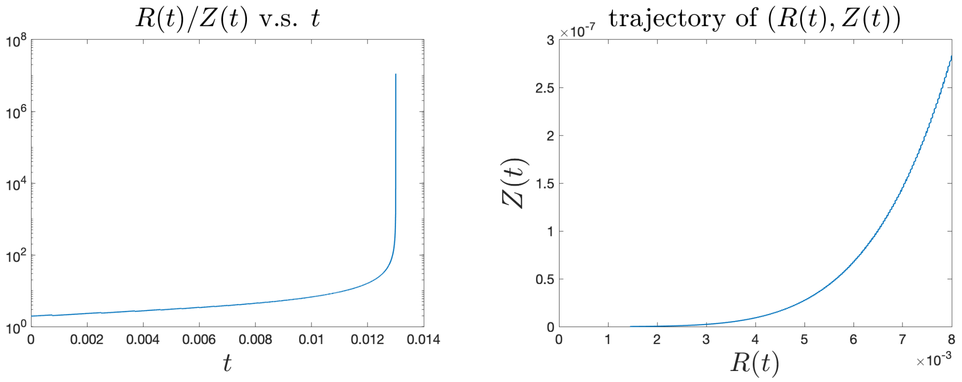


Figure 4.4: Curve of the ratio $R(t)/Z(t)$ as a function of time and the trajectory of $(R(t), Z(t))$ over time.

If we plug in this ansatz back to the weak convection model (4.4), and expect the leading order terms of both sides match with each other, we should still have the scaling relation, as the special case $\alpha = 1$ of (2.6):

$$c_\omega = 1 + c_l, \quad c_\psi = 1 - c_l.$$

As a consequence, we should still expect that

$$\|\psi_{1,z}(t)\|_{L^\infty} \sim \frac{1}{T-t}.$$

However, we no longer have $\|\omega^\theta(t)\|_{L^\infty} \sim 1/(T-t)$, and instead we have

$$\|\omega^\theta(t)\|_{L^\infty} \sim \frac{1}{(T-t)^{c_\omega - c_l'}}.$$

We notice that the exponent $c_\omega - c_l' \geq c_\omega - c_l = 1$, so our ansatz (4.7) still allows the self-similar solution to satisfy the Beale-Kato-Majda blow-up criterion (**??**).

Due to the one-dimensional structure of the solution, we only plot the z-cross section to check if the solution is potentially self-similar. Specifically, we plot

$$\hat{\omega}_1(\zeta, t) = \omega_1(0, \zeta Z_1(t), t)/\|\omega_1\|_{L^\infty},$$

at different time instants. In Figure 4.5, we show the profiles of $-\hat{\omega}_1$ at the 1.0×10^5-th, 1.1×10^5-th, 1.2×10^5-th, 1.3×10^5-th iteration of our computation. We see no visible difference in $-\hat{\omega}_1$. In fact, during this time, the maximum vorticity $\|\omega\|_{L^\infty}$ has increased by a factor of 8.2×10^3. This gives us a strong evidence that there exists a self-similar blow-up profile.

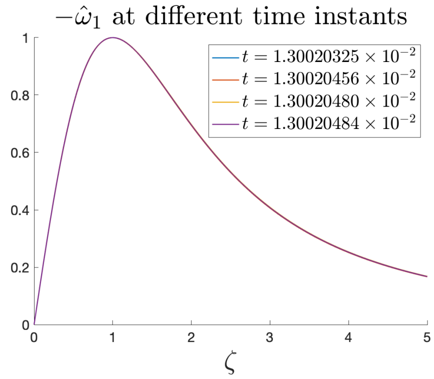


Figure 4.5: Profiles of $-\hat{\omega}_1$ at different time instants.

Scaling Analysis

Similar as we did in Section 2.3 and Section 3.2, we perform scaling analysis on the curves of important quantities.

For $\|\psi_{1,z}\|_{L^\infty}$, since our self-similar ansatz (4.7) predicts it to scale like $1/(T-t)$, we will directly fit $\|\psi_{1,z}\|_{L^\infty}^{-1}$ with t and check the quality of the fitting. For quantities like $\|\omega\|_{L^\infty}$, $\|\omega_1\|_{L^\infty}$ and Z, we do not know their exponents. Therefore, we follow our searching algorithm, described in Chapter 2, for the exponent c, and then fit a linear model.

In Figure 4.6, we can see that the fitting quality is all very good, with very high R^2 values. Besides, the estimated blow-up times from different quantities match each other up to 7 digits. From the fitting of $Z_1^{1/c}$ and $\|\omega_1\|_{L^\infty}^{-1/c}$, we obtain the estimate $c_\omega \approx 2.08$ and $c_l \approx 1.08$. This matches with our scaling relation $c_\omega = 1 + c_l$ approximately. In addition, we notice that $\|\omega\|_{L^\infty}$ scales like $1/(T-t)^{1.88}$. The exponent 1.88 is clearly larger than 1, which is the case for the isotropic self-similar blow-up (2.3). This also matches with our assumption that $c_l' \leq c_l$.

Dynamic Rescaling Method

Numerical results in the previous sections suggest that the potential blow-up of the solution is very likely to be self-similar. In this section, we use the dynamic rescaling method to study this potential self-similar profile. From the zoomed-in profiles of $-\omega_1$ and $-\psi_1$ in Figure 4.2, we can see that there is little dependence on r in $-\omega_1$ and $-\psi_1$. Thus, we tend to think that there is no dependence on r in the self-similar ansatz (4.7). However, in favor of a closed-form equation, we still add same stretching term in r and z, as we did in (2.13). The dynamic rescaling

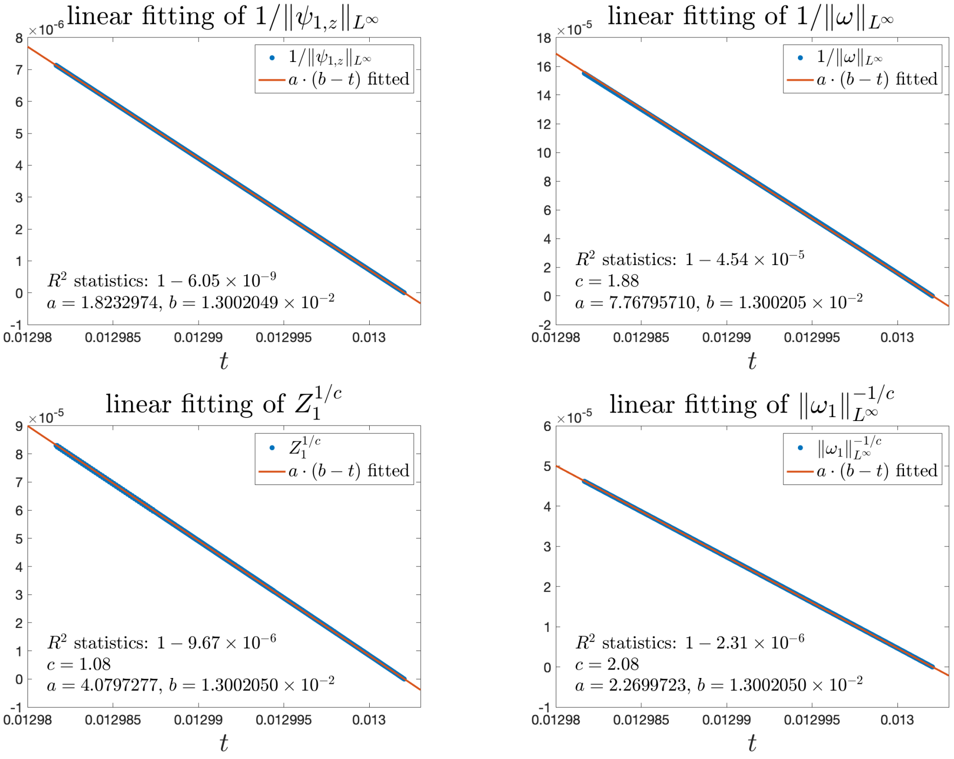


Figure 4.6: Linear fitting of $\|\psi_{1,z}\|_{L^\infty}^{-1}$, $\|\omega\|_{L^\infty}^{-1/c}$, $Z_1^{1/c}$ and $\|\omega_1\|_{L^\infty}^{-1/c}$ with time.

formulation for the weak convection model (4.4) is

$$\tilde{\omega}_{1,\tau} + \left(\tilde{c}_l\xi + \varepsilon\tilde{u}^{\xi}\right)\tilde{\omega}_{1,\xi} + \left(\tilde{c}_l\zeta + \varepsilon\tilde{u}^{\zeta}\right)\tilde{\omega}_{1,\zeta} = \left(\tilde{c}_\omega - (n-2-\varepsilon)\tilde{\psi}_{1,\zeta}\right)\tilde{\omega}_1, \quad \text{(4.8a)}$$

$$-\tilde{\psi}_{1,\xi\xi} - \tilde{\psi}_{1,\zeta\zeta} - \frac{n}{\xi}\tilde{\psi}_{1,\xi} = \tilde{\omega}_1, \quad \text{(4.8b)}$$

$$\tilde{u}^{\xi} = -\xi\tilde{\psi}_{1,\zeta}, \quad \tilde{u}^{\zeta} = (n-1)\tilde{\psi}_1 + \xi\tilde{\psi}_{1,\xi}. \quad \text{(4.8c)}$$

Similarly, if we plug in our new ansatz (4.7) and ignore the lower order terms, we should have following identities

$$c_l = -\frac{\tilde{c}_l}{\tilde{c}_\omega + \tilde{c}_l}, \quad c_\omega = \frac{\tilde{c}_\omega}{\tilde{c}_\omega + \tilde{c}_l}, \quad c_\psi = \frac{\tilde{c}_\psi}{\tilde{c}_\omega + \tilde{c}_l},$$

which can be seen as the special case of $\alpha = 1$ of (2.21).

For our computation of the dynamic rescaling formulation (4.8), we still use the same computational domain $\mathcal{D}'$, the normalization conditions (2.16), and the operator splitting method (2.19) as in Section 2.4.

In Figure 4.7, the curves of the scaling factors are plotted. In the top row, the scaling factors $\tilde{c}_l$ and $\tilde{c}_\omega$ in the dynamic rescaling method demonstrate good convergence to

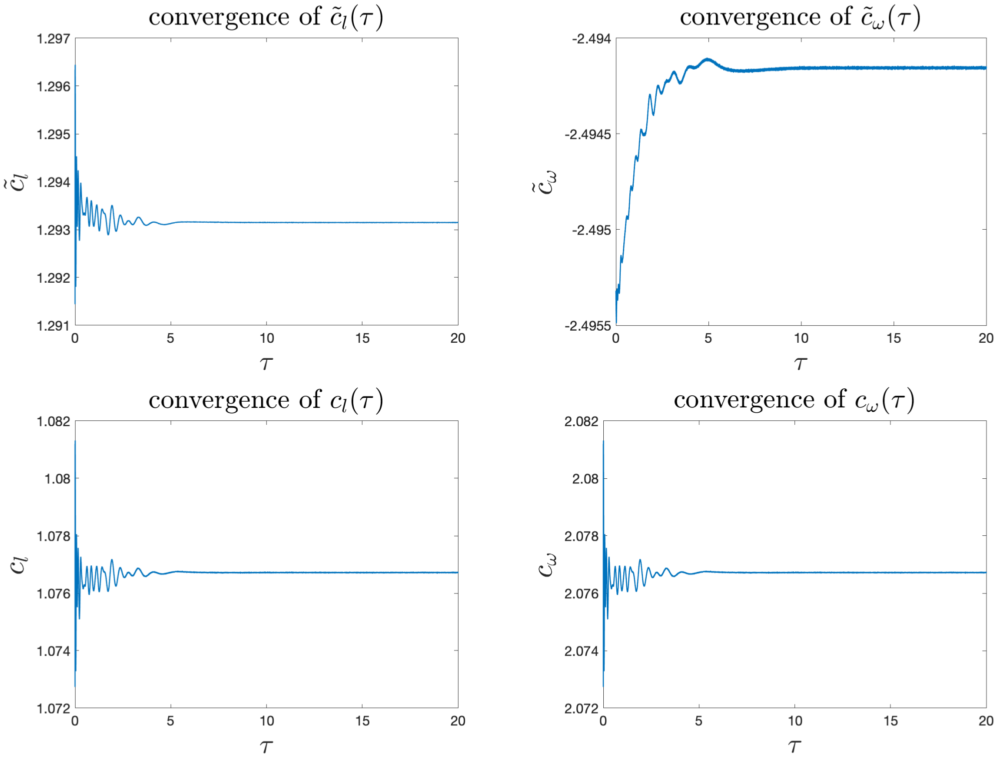


Figure 4.7: Convergence curves of the scaling factors using dynamic rescaling method with $n = 3$ and $\varepsilon = 0.1$. Top row: $\tilde{c}_l$; and $\tilde{c}_\omega$. Bottom row: c_l and c_ω.

a constant value. In the second row, the scaling factors c_l and c_ω in our self-similar ansatz (4.7) converge to the value of 1.077 and 2.077 respectively. The estimated values for c_l and c_ω not only match the estimated values of the scaling analysis, but also satisfy the relation $c_\omega = 1+c_l$ approximately. This provides another verification of the validity of our results.

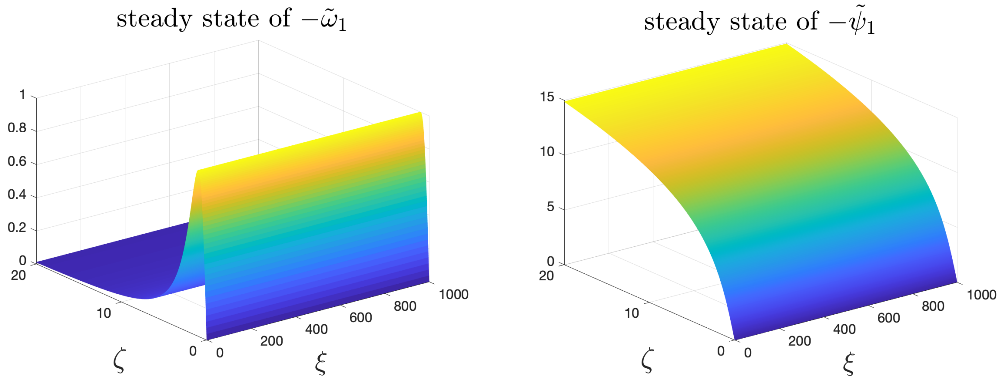


Figure 4.8: Steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$, with $n = 3$ and $\varepsilon = 0.1$.

In Figure 4.8, we show the steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$. Despite that the scales

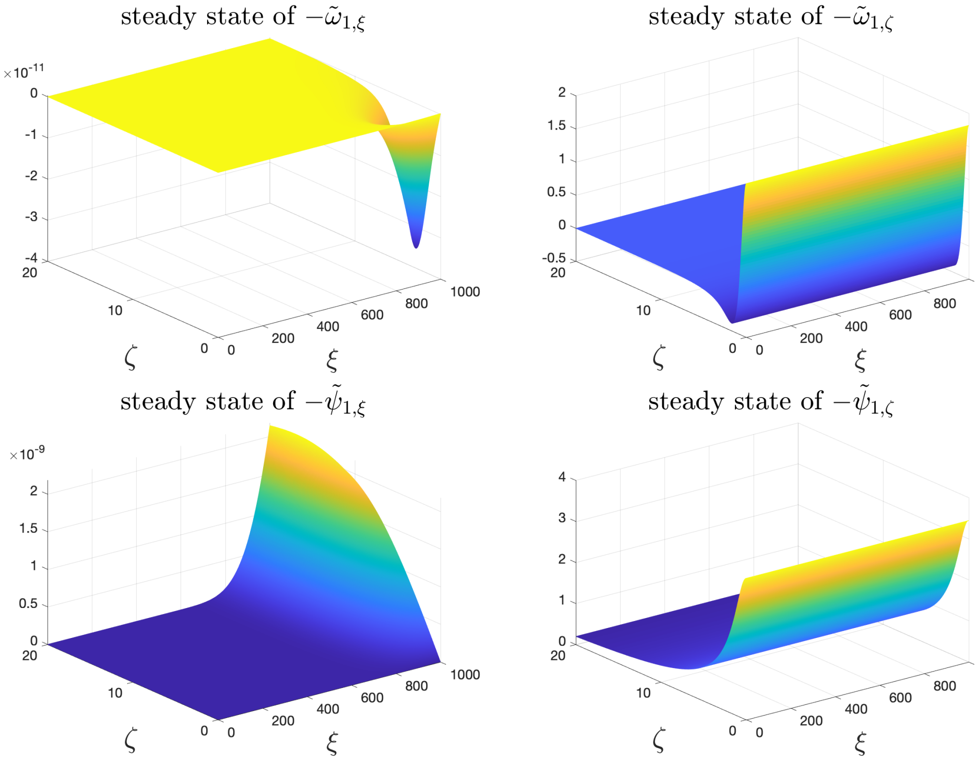


Figure 4.9: Steady states of the derivatives $-\tilde{\omega}_{1,\xi}$, $-\tilde{\omega}_{1,\zeta}$ and $-\tilde{\psi}_{1,\xi}$, $-\tilde{\psi}_{1,\zeta}$, with $n = 3$ and $\varepsilon = 0.1$.

for r axis and z axis are quite different in the figures, the steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$ seem to depend on z only. In other words, the steady states are also very one-dimensional. In Figure 4.9, we show the derivatives of the steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$. The ξ-derivatives, $-\tilde{\omega}_{1,\xi}$ and $-\tilde{\psi}_{1,\xi}$, are at the magnitude of at most 2.5×10^{-9}, much smaller than the magnitude of the ζ-derivatives, $-\tilde{\omega}_{1,\zeta}$ and $-\tilde{\psi}_{1,\zeta}$, who are at the scale of 10^0. Figure 4.9 gives a strong evidence of the one-dimensional profile in the self-similar blow-up. In the meanwhile, the highly one-dimensional self-similar profile we captured also endorses our anisotropic self-similar ansatz we assumed in (4.7), where the collapsing along the r-axis is much weaker than the collapsing along the z-axis.

Sensitivity to Initial Data

The potential self-similar blow-up solutions we described above seem to be robust with the initial data. We perturb the original initial condition (4.6) a lot and design the following initial conditions for ω_1:

$$\omega_1^{\circ,1} = \frac{-6000\cos^2\left(\frac{\pi r}{2}\right)\sin(2\pi z)}{1+12.5\sin^2(\pi z)},$$
$$\omega_1^{\circ,2} = -6000\cos^2\left(\frac{\pi r}{2}\right)\sin(2\pi z)\left(2+\exp\left(-r^2\sin^2(\pi z)\right)\right).$$

In both cases, the decay as $r \to 1$ is much slower than the original $(1-r^2)^{18}$. The second case $\omega_1^{\circ,2}$ also introduces non-tensor-product part of r and z.

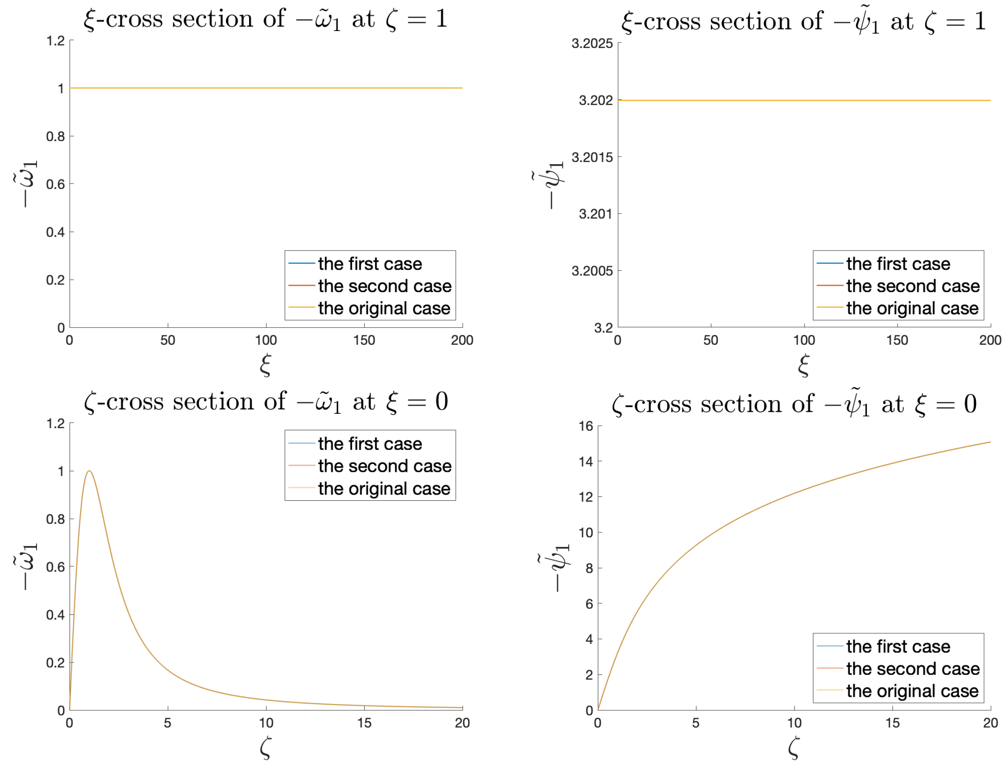


Figure 4.10: Comparison of cross sections of steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$ with different initial data with $n = 3$, $\varepsilon = 0.1$.

Due to the limited computational resources, we only solve the weak convection model (4.4) with $n = 3$ and $\varepsilon = 0.1$. For both cases, we first use the adaptive mesh method to solve the equations (4.4), and then use the solutions in the last iteration as initial conditions to the dynamic rescaling method. In Figure 4.10, we compare the cross sections of steady states of $\tilde{\omega}_1$ and $\tilde{\psi}_1$ in cases 1 and 2 with those of the original case of initial data. It is interesting to notice that the cross sections of all three cases have no visible difference. We report that the sup-norm relative difference of the steady states of $\tilde{\omega}_1$ in $(\xi, \zeta) \in [0, 200] \times [0, 20]$ among these three cases is at most 5.4×10^{-9}. We also report that the scaling factors c_l estimated by these three methods are all 1.722, agreeing each other by at least 4 digits. The steady states and scaling factors provide a strong evidence that the potential self-similar blow-up solution is not sensitive to the choice of the initial condition for ω_1.

4.3 Influence of the Convection Strength and Dimension on Potential Finite-Time Blow-Up

In this section, we explore how the convection strength ε and the dimension n influence the blow-up of the weak convection model (4.4). For each combination of ε and n in the following, the equations (4.4) are first solved by the adaptive mesh method close enough to the potential blow-up time, and then we switch to the dynamic rescaling method to continue the computation in a local region near the origin $(r, z) = (0, 0)$.

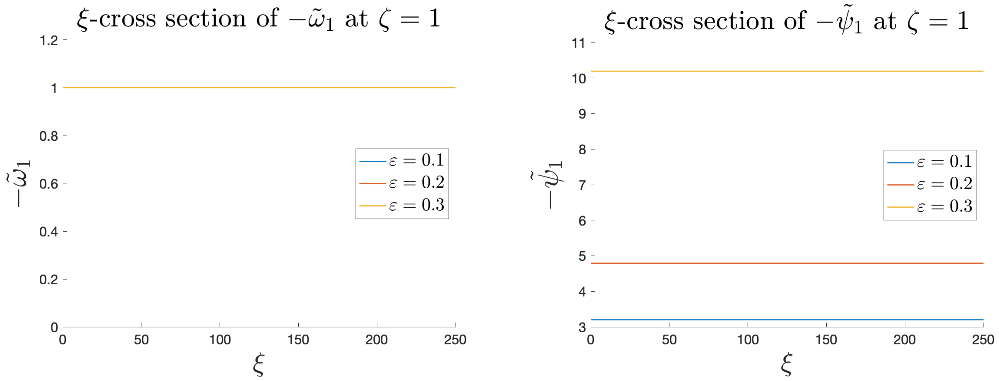


Figure 4.11: Cross sections in ξ of steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$ with different ε in $\mathbb{R}^3$.

In Figure 4.11 and 4.12, we compare the cross sections of steady states with different convection strength ε. We can see from Figure 4.11 that regardless of the convection strength ε, the steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$ seem to be very one-dimensional. From Figure 4.12, we see that the larger ε is, the wider spread $-\tilde{\omega}_1$ is. This could be caused by the stronger convection effect on ω_1 with larger ε. We can also see that the magnitude of $\tilde{\psi}_1$ grows fast with ε, which further implies stronger convection along the z-axis, because $u^\zeta = (n-1)\tilde{\psi}_1$ when $\xi = 0$.

In Figure 4.13 and 4.14, we compare the cross sections of steady states in different dimensions n. Similarly, we still observe that the steady states for both $\tilde{\omega}_1$ and $\tilde{\psi}_1$ are very one-dimensional. In Figure 4.14, we can also see that the steady state of $\tilde{\omega}_1$ becomes slightly more compact, and the magnitude of $\tilde{\psi}_1$ becomes slightly smaller as the dimension n increases. There seems to be a non-trivial limit case as $n \longrightarrow +\infty$.

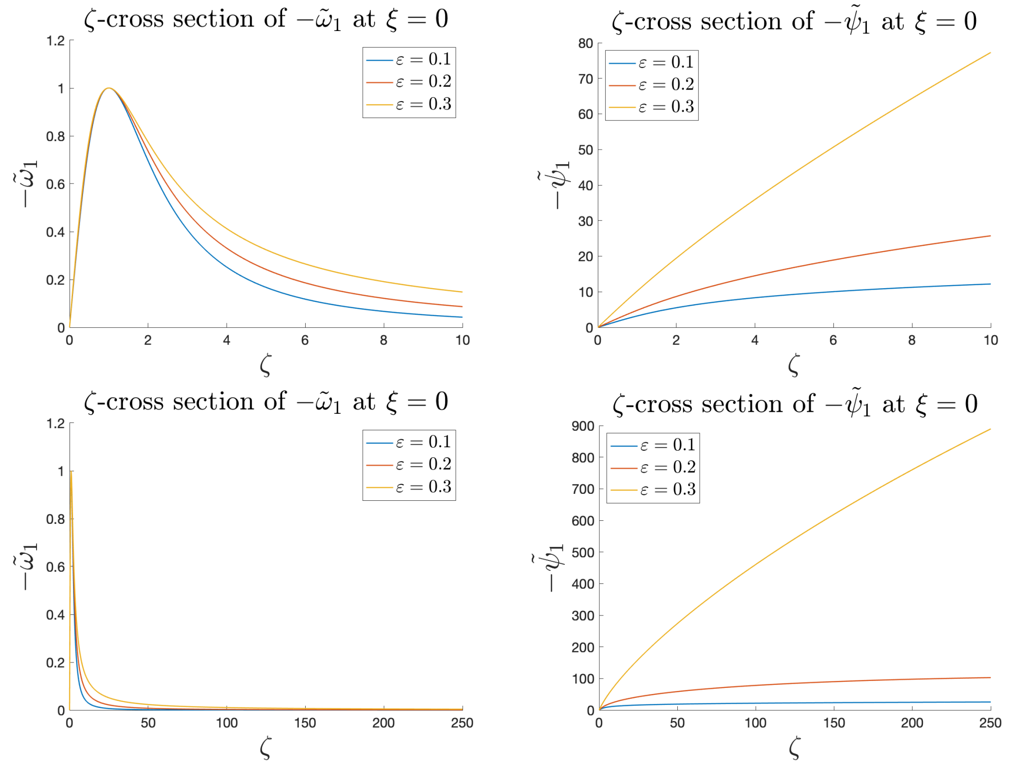


Figure 4.12: Cross sections in ζ of steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$ with different ε in $\mathbb{R}^3$. Top row: on a local window. Bottom row: on a larger window.

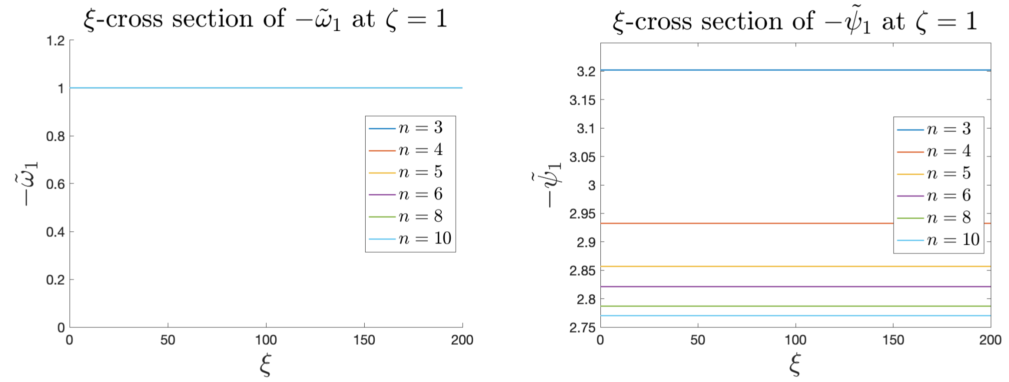


Figure 4.13: Cross sections in ξ of steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$ with different dimensions n with $\varepsilon = 0.1$.

c_l	ε								
n	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
3	0.5000	1.072	2.742	76.61	-	-	-	-	-
10	0.5000	0.7578	1.101	1.588	2.356	3.818	7.949	118.3	-

Table 4.1: The scaling factor c_l for different choice of n and ε.

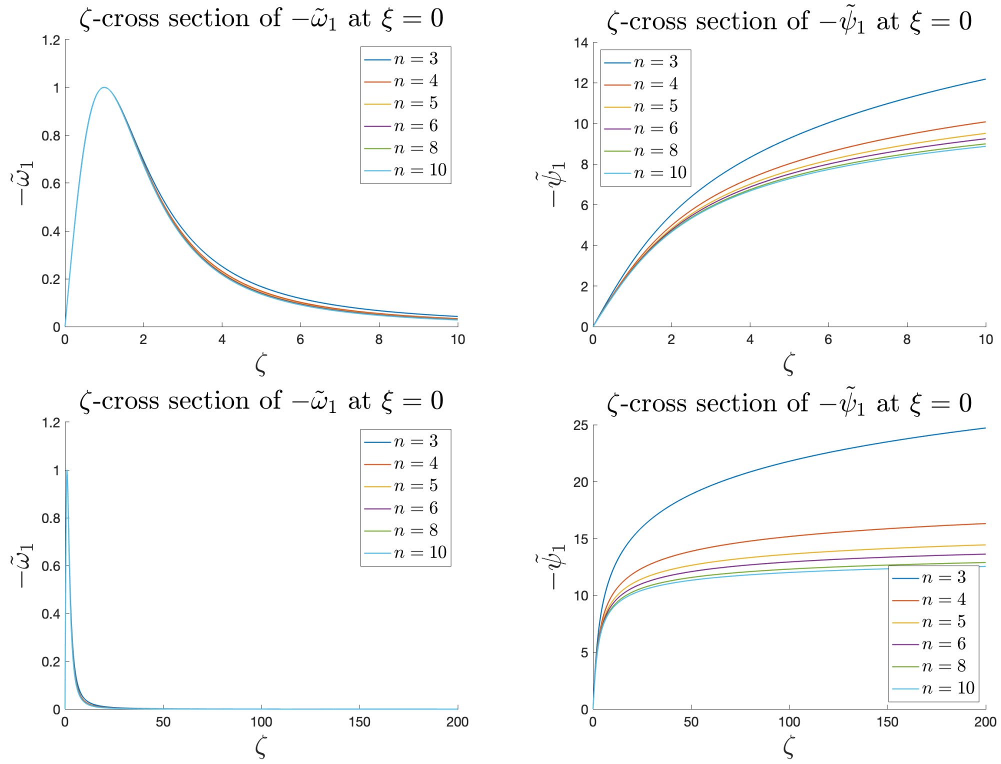


Figure 4.14: Cross sections in ζ of steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$ with different dimensions n with $\varepsilon = 0.1$. Top row: on a local window. Bottom row: on a larger window.

In Table 4.1, we list the estimated scaling factor c_l we observed for different combinations of (n, ε). The empty entries mean that the combination of (n, ε) does not develop finite-time blow-up. The scaling factor c_l grows fast as ε approaches the critical value ε^* that separates the region between the blow-up and non-blow-up. As ε approaches ε^*, it seems that c_l tends to infinity. Based on the data points in Table 4.1, it is natural to conjecture that $\varepsilon^* = 1 - \frac{2}{n}$. In Section 4.4, we will provide a heuristic explanation for this critical value.

4.4 A One-Dimensional Model for the Finite-Time Blow-Up

The numerical results in Section 4.2 strongly suggest that the profiles of ω_1 and ψ_1 can be well approximated by their dependence on z only. Based on this observation, we assume that

$$\omega_1(r, z) = \omega_1(0, z),$$

and derive a one-dimensional model for the weak convection model (4.4).

We simply assume $\partial_r = 0$ for all quantities in the weak convection model (4.4). We slightly abuse the symbols and consider ω_1 and ψ_1 as the 1D profile of the original ω_1 and ψ_1 in z. The 1D model of the weak convection model can be written as

$$\omega_{1,t} + \varepsilon(n-1)\psi_1\omega_{1,z} = -(n-2-\varepsilon)\psi_{1,z}\omega_1, \tag{4.9a}$$

$$-\psi_{1,zz} = \omega_1. \tag{4.9b}$$

In fact, the Poisson equation (4.9b) under the given zero Dirichlet boundary condition has the following closed-form solution:

$$\psi_1(z) = L(L-z)\int_0^{z/L} s\omega_1(Ls)\mathrm{d}s + Lz\int_{z/L}^1 (1-s)\omega_1(Ls)\mathrm{d}s,$$

where $L = 1/2$ is the domain size in z-axis. The 1D model (4.9) is very easy to compute numerically, because there is no need to solve the 2D Poisson equation.

Numerical Verification

We run direct numerical simulation to check how well the 1D model (4.9) approximates the original model (4.4). For the 1D model, we use the z-cross section of the original initial data (4.6) at $r = 0$ as our initial data. Similarly, we first solve the 1D model (4.9) using the adaptive mesh method, and then use the result in the late stage to start the dynamic rescaling computation.

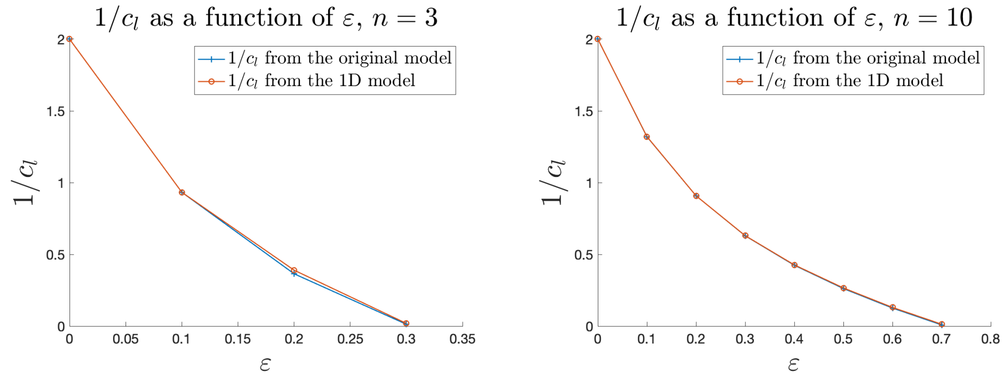


Figure 4.15: Comparison of the estimated scaling factor c_l from the original model and the 1D model.

We first look at their estimate of the scaling factor c_l. In Figure 4.15, we compare the $1/c_l$ curve versus ε. In both cases of $n = 3$ and $n = 10$, the curves from the original model and the 1D model look very close to each other. In Table 4.2, we listed the estimated c_l for different combinations of (n, ε). We can see that the difference is very small when ε is small. However, the scaling factors c_l between our 1D model

$n = 3$	ε			
	0.0	0.1	0.2	0.3
$c_{l,\text{original}}$	0.5000	1.072	2.742	76.61
$c_{l,\text{1D}}$	0.5000	1.072	2.564	52.17

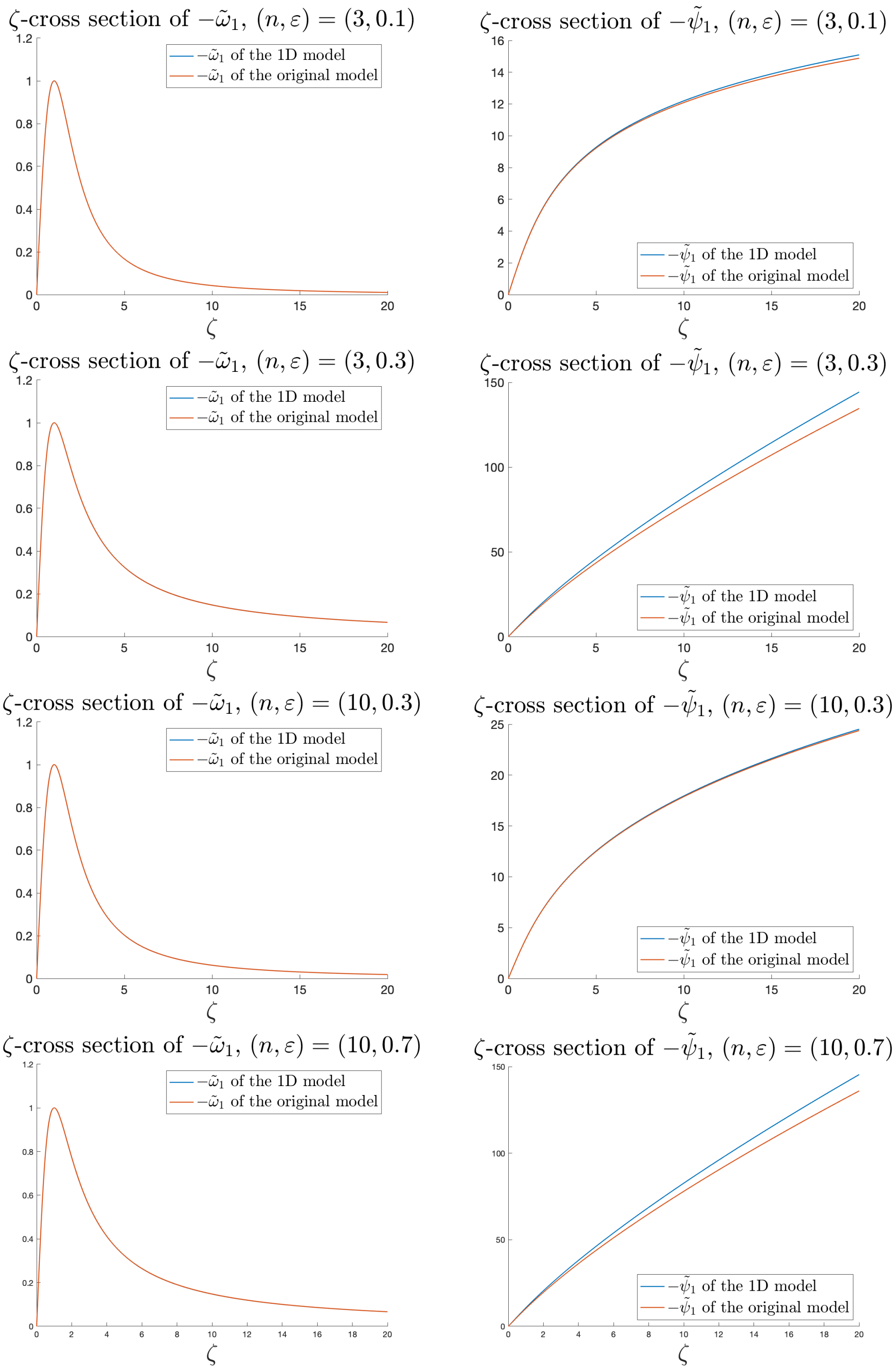
$n = 10$	ε							
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$c_{l,\text{original}}$	0.5000	0.7578	1.101	1.588	2.356	3.818	7.946	118.3
$c_{l,\text{1D}}$	0.5000	0.7572	1.101	1.586	2.343	3.758	7.670	74.13

Table 4.2: The scaling factors c_l from the original model $c_{l,\text{original}}$ and the 1D model $c_{l,\text{1D}}$ for different combinations of (n, ε).

and the original model will become inconsistent when ε is approaching its critical value ε^*.

Next we compare the steady states of their converged solutions. We consider the combinations $(n, \varepsilon) = (3, 0.1)$, $(3, 0.3)$, $(10, 0.3)$, and $(10, 0.7)$, because they cover the situations when ε is small and ε is close to ε^* in both dimensions. We report that for all cases, the steady state of the original model has very good one-dimensional structure. Therefore, in Figure 4.16, we show the comparison between ζ-cross section of the original model (4.8) and the steady state of our 1D model (4.9). We notice that there is no visible difference in the steady states of $-\tilde{\omega}_1$ in all cases. In fact, even for $\zeta \in [0, 1000]$, the steady states of $-\tilde{\omega}_1$ only differ in the relative sup-norm by 2.4×10^{-4}, 4.6×10^{-4}, 4.4×10^{-4}, and 3.3×10^{-4} in four cases respectively. However, the comparison of the steady states of $-\tilde{\psi}_1$ show that the 1D model approximates the original model well only when ε is small. The profiles of the steady states of $-\tilde{\psi}_1$ differ in relative sup-norm by 1.4×10^{-2}, 7.2×10^{-2}, 5.6×10^{-3} and 6.9×10^{-2} in four cases respectively. The difference in the steady states of $-\tilde{\psi}_1$ is mainly located in the far field, especially in the cases of $(n, \varepsilon) = (3, 0.3)$ and $(10, 0.7)$. In fact, the difference grows as ζ increases. An explanation for this phenomenon is that in the far field, the second-order ζ-derivative $-\tilde{\psi}_{1,\zeta\zeta}$ becomes very small, so the ξ-derivative terms $-\tilde{\psi}_{1,\xi\xi} - \frac{n}{\xi}\tilde{\psi}_{1,\xi}$ should not be neglected in the Poisson equation (4.8b). Therefore, in the far field, (4.9b) needs some correction to better approximate (4.4b).

As a conclusion, the numerical simulation verifies that the 1D model (4.9) is a good approximation of the original model (4.4) when ε is small. When the parameter ε is approaching ε^*, the ψ_1 of the original model is different from the 1D model in the far field, although the profiles of ω_1 of both models are very close.



ζ-cross section of $-\tilde{\omega}_1$, $(n,\varepsilon)=(3,0.1)$

ζ-cross section of $-\tilde{\psi}_1$, $(n,\varepsilon)=(3,0.1)$

ζ-cross section of $-\tilde{\omega}_1$, $(n,\varepsilon)=(3,0.3)$

ζ-cross section of $-\tilde{\psi}_1$, $(n,\varepsilon)=(3,0.3)$

ζ-cross section of $-\tilde{\omega}_1$, $(n,\varepsilon)=(10,0.3)$

ζ-cross section of $-\tilde{\psi}_1$, $(n,\varepsilon)=(10,0.3)$

ζ-cross section of $-\tilde{\omega}_1$, $(n,\varepsilon)=(10,0.7)$

ζ-cross section of $-\tilde{\psi}_1$, $(n,\varepsilon)=(10,0.7)$

$-\tilde{\omega}_1$ of the 1D model
$-\tilde{\omega}_1$ of the original model
$-\tilde{\psi}_1$ of the 1D model
$-\tilde{\psi}_1$ of the original model
ζ
0
0.2
0.4
0.6
0.8
1
1.2
2
4
6
8
10
12
14
16
18
20
5
15
25
50
100
150

Figure 4.16: Comparison of the 1D steady states of $-\tilde{\omega}_1$ and $-\tilde{\psi}_1$ for the 1D model and the original model.

A Heuristic Explanation of the Blow-Up

The proposed 1D model (4.9) has a much simpler structure than the original model (4.4), and this gives hope to derive more explicit explanation of the numerical phenomenon.

In (4.9), let us introduce $v = -\psi_{1,z}$, the model can be re-written as

$$\begin{aligned} \omega_{1,t} + \varepsilon(n-1)\psi_1\omega_{1,z} &= (n-2-\varepsilon)v\omega_1, \\ v_z &= \omega_1, \\ -\psi_{1,z} &= v. \end{aligned}$$

We then integrate the dynamic equation for ω_1 from 0 to z. Due to the second equation $v_z = \omega_1$, we have

$$v_t + \varepsilon(n-1)\int_0^z \psi_1\omega_{1,z} = -(n-2-\varepsilon)\int_0^z \psi_{1,z}\omega_1 + C(t), \tag{4.10}$$

where C is a function of time t only.

Now we add $\varepsilon(n-1)\int_0^z \psi_{1,z}\omega_1$ to both sides of (4.10). Using the product rule, the left-hand side of (4.10) becomes,

$$\begin{aligned} v_t + \varepsilon(n-1)\int_0^z \left(\psi_1\omega_{1,z} + \psi_{1,z}\omega_1\right) &= v_t + \varepsilon(n-1)\int_0^z (\psi_1\omega_1)_z \\ &= v_t + \varepsilon(n-1)\psi_1\omega_1 \\ &= v_t + \varepsilon(n-1)\psi_1 v_z. \end{aligned}$$

And using integration by part, the right-hand side of (4.10) becomes

$$\begin{aligned} -(n-2-n\varepsilon)\int_0^z \psi_{1,z}\omega_1 + C(t) &= (n-2-n\varepsilon)\int_0^z v v_z + C(t) \\ &= \frac{1}{2}(n-2-n\varepsilon)\int_0^z \left(v^2\right)_z + C(t) \\ &= \frac{1}{2}(n-2-n\varepsilon)v^2 + C'(t). \end{aligned}$$

The constant in the last line is related to the previous constant by $C'(t) = C(t) - \frac{1}{2}(n-2-n\varepsilon)v(0)^2$. However, we slightly abuse the symbols and still use C for C' in the following. So we now have

$$v_t + \varepsilon(n-1)\psi_1 v_z = \frac{1}{2}(n-2-n\varepsilon)v^2 + C(t). \tag{4.11}$$

To fix the constant $C(t)$, we notice that ψ_1 has the zero Dirichlet boundary condition at both sides $z = 0$ and $z = 1/2$, so $\int_0^{1/2} v(z)\mathrm{d}z = \psi_1(0) - \psi_1(1/2) = 0$ is a constant. Therefore, we integrate (4.11) from $z = 0$ to $z = 1/2$, and we must have that

$$\varepsilon(n-1)\int_0^{1/2} \psi_1(z)v_z(z)\mathrm{d}z = \frac{1}{2}(n-2-n\varepsilon)\int_0^{1/2} v^2(z)\mathrm{d}z + \frac{1}{2}C(t).$$

Using integration by part again, we have

$$\begin{aligned}\int_0^{1/2} \psi_1(z)v_z(z)\mathrm{d}z &= (\psi_1 v)\,|_0^{1/2} - \int_0^{1/2} \psi_{1,z}(z)v(z)\mathrm{d}z,\\ &= \int_0^{1/2} v^2(z)\mathrm{d}z.\end{aligned}$$

Plugging this relation back, we have

$$\begin{aligned}C(t) &= 2\varepsilon(n-1)\int_0^{1/2} \psi_1(z)v_z(z)\mathrm{d}z - (n-2-n\varepsilon)\int_0^{1/2} v^2(z)\mathrm{d}z,\\ &= 2\varepsilon(n-1)\int_0^{1/2} v^2(z)\mathrm{d}z - (n-2-n\varepsilon)\int_0^{1/2} v^2(z)\mathrm{d}z,\\ &= ((3n-2)\varepsilon - n + 2)\int_0^{1/2} v^2(z)\mathrm{d}z.\end{aligned}$$

This gives rise to the following equations for v:

$$v_t + \varepsilon(n-1)\psi_1 v_z = \frac{1}{2}(n-2-n\varepsilon)v^2 + ((3n-2)\varepsilon - n + 2)\int_0^{1/2} v^2(z)\mathrm{d}z, \tag{4.12a}$$

$$-\psi_{1,z} = v. \tag{4.12b}$$

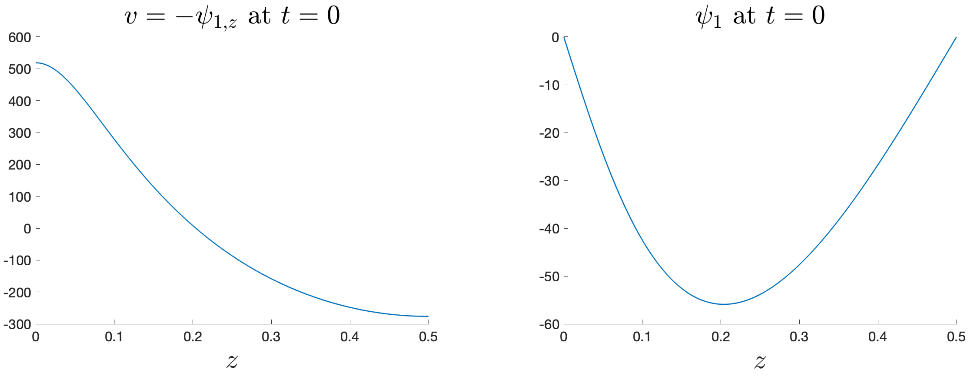


Figure 4.17: 1D profile of $v = -\psi_{1,z}$ and ψ_1 at $t = 0$.

In Figure 4.17, we plot the initial state of v and ψ_1 for the initial data ω_1° we used. We can see that v achieves its maximum at $z = 0$, and it is positive near $z = 0$. The negative value of ψ_1 suggests that at $t = 0$, v is transported towards $z = 0$. In fact, $\omega_1 = v_z$ satisfies the equation

$$\omega_{1,t} + \varepsilon(n-1)\psi_1\omega_{1,z} = -(n-2-\varepsilon)\psi_{1,z}\omega_1.$$

Since the initial condition (4.6) for ω_1 is negative in $(0, 1/2)$ and zero at the boundaries $z = 0$ and $z = 1/2$, ω_1 will remain negative inside the domain and zero at the boundaries. The negativity of of $\omega_1 = v_z$ inside the domain leads to the following lemma:

Lemma 4.4.1. *$v(z,t)$ is always monotonically decreasing in z.*

Therefore, the maximum of v will always locate at $z = 0$. Thus, we let $V(t) = v(0,t)$.

We first drop the constant term $C(t)$. Because $\psi_1(0) = 0$, the convection term vanishes in the equation for V, so we have

$$V_t = \frac{1}{2}(n-2-n\varepsilon)V^2. \tag{4.13}$$

Therefore, if and only if $\varepsilon < 1 - \frac{2}{n}$, V will have a finite-time blow-up near $z = 0$ at the rate of $1/(T-t)$, where T is the blow-up time. Since we have $\omega_1 = v_z$, we naturally claim a finite-time blow-up at $z = 0$ for ω_1 when $\varepsilon < 1 - \frac{2}{n}$. In fact, in Section 4.2, the scaling analysis for self-similar ansatz (4.7) implies that we must have $\|v\|_{L^\infty} = \|\psi_{1,z}\|_{L^\infty} \sim 1/(T-t)$, which has been verified by our numerical simulation in Figure 4.6. This exactly agrees with the corollary here that $V \sim 1/(T-t)$.

In addition, (4.13) suggests that if $\varepsilon > 1 - \frac{2}{n}$, V will decay like $1/(t+c)$ for some constant c. In Figure 4.18, we plot $1/\|v\|_{L^\infty}$ against time t for two no-blow-up cases: $(n,\varepsilon) = (3, 0.4)$ and $(10, 0.9)$. Surprisingly, the behavior of $1/\|v\|_{L^\infty} = 1/\|\psi_{1,z}\|_{L^\infty}$ is quite linear with the time t, which agrees with the prediction from (4.13) fairly well.

Roughly speaking, if there is a focusing self-similar blowup for $v = -\psi_{1,z}$, we can easily show that the growth of its L^2 norm is much slower than $V^2 = v(0,t)^2$. Therefore, the effect of dropping the constant term $C(t)$ can be neglected. In the following, we will show this rigorously.

If $\frac{n-2}{3n-2} \le \varepsilon \le 1 - \frac{2}{n}$, we have $(3n-2)\varepsilon - n + 2 \ge 0$, and as a result

$$C(t) = ((3n-2)\varepsilon - n + 2)\int_0^{1/2} v^2(z)\mathrm{d}z \ge 0.$$

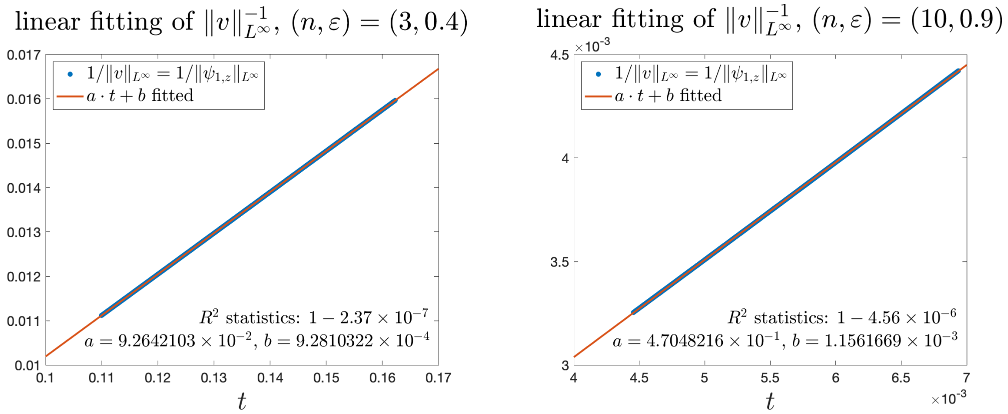


Figure 4.18: Linear fitting of $1/\|\psi_{1,z}\|_{L^\infty}$ with time t for the no-blow-up cases.

Therefore we must have

$$V_t > \frac{1}{2}(n - 2 - n\varepsilon)V^2,$$

with $n - 2 - n\varepsilon > 0$, which implies that $v = -\psi_{1,z}$ will blow-up in finite time.

If $0 \leq \varepsilon < \frac{n-2}{3n-2}$, we still have $n - 2 - n\varepsilon > 0$, but now $C(t) < 0$. We need to establish a bound for the constant term $C(t)$.

Let $W(t) = v(1/2, t)$ be the minimum of v at time t. We first claim that

Lemma 4.4.2. *If $W < 0$, then $\int_0^{1/2} v^2(z)\mathrm{d}z \leq -\frac{1}{2}VW$.*

Proof. From Lemma 4.4.1 we know that v is monotonic decreasing. Since $V > 0$ and $W < 0$, we know that there is exactly one zero of v between 0 and 1/2. Let z^* be this zero. We have that $0 < v \leq V$ for $z < z^*$, and $0 > v \geq W$ for $z > z^*$.

Since $v = -\psi_{1,z}$, and $\psi_1(0) = \psi_1(1/2) = 0$, we know that $\int_0^{1/2} v\mathrm{d}z = 0$. In other words,

$$\int_0^{z^*} v(z)\mathrm{d}z = -\int_{z^*}^{1/2} v(z)\mathrm{d}z.$$

Let $A = \int_0^{z^*} v(z)\mathrm{d}z$, we have that

$$A = \int_0^{z^*} v(z)\mathrm{d}z \leq Vz^*.$$

On the other hand,

$$A = -\int_{z^*}^{1/2} v(z)\mathrm{d}z \leq -W(1/2 - z^*).$$

So we must have

$$A \le \min(Vz^*, -W(1/2 - z^*)).$$

It is not hard to show that

$$\max_{z^* \in (0,1/2)} \min(Vz^*, -W(1/2 - z^*)) = \frac{-VW}{2(V - W)},$$

and therefore, we have $A \le \frac{-VW}{2(V-W)}$.

Finally, we have

$$\begin{aligned} \int_0^{1/2} v^2(z)\mathrm{d}z &= \int_0^{z^*} v^2(z)\mathrm{d}z + \int_{z^*}^{1/2} v^2(z)\mathrm{d}z \\ &\le V \int_0^{z^*} v(z)\mathrm{d}z + W \int_{z^*}^{1/2} v(z)\mathrm{d}z \\ &= (V - W)A \le -\frac{1}{2}VW, \end{aligned}$$

which finishes the proof. □

The equation for W reads

$$W_t = \frac{1}{2}(n - 2 - n\varepsilon)W^2 + C(t).$$

And therefore,

$$\frac{\mathrm{d}}{\mathrm{dt}}(V + W) = \frac{1}{2}(n - 2 - n\varepsilon)\left(V^2 + W^2\right) + 2C(t).$$

Since $\int_0^{1/2} v(z)\mathrm{d}z = 0$, and by the monotonicity of v as stated in Lemma 4.4.1, we must have $W(t) = v(1/2, t)$, as the minimum of v in $[0, 1/2]$, is smaller than zero.

Since $W < 0$, we use Lemma 4.4.2 to bound the right-hand side of the above equation. Noticing that $n - 2 - n\varepsilon > 0$ and $(3n - 2)\varepsilon - n + 2 < 0$, we have

$$\begin{aligned} &\frac{1}{2}(n - 2 - n\varepsilon)\left(V^2 + W^2\right) + 2C(t) \\ =&\frac{1}{2}(n - 2 - n\varepsilon)\left(V^2 + W^2\right) + 2\left((3n - 2)\varepsilon - n + 2\right)\int_0^{1/2} v^2\mathrm{d}z \\ \ge&\frac{1}{2}(n - 2 - n\varepsilon)\left(V^2 + W^2\right) - \left((3n - 2)\varepsilon - n + 2\right)VW \\ =&\varepsilon(n - 1)\left(V^2 + W^2\right) + \frac{1}{2}\left(n - 2 - (3n - 2)\varepsilon\right)(V + W)^2 \ge 0. \end{aligned}$$

And thus we know that $\frac{\mathrm{d}}{\mathrm{dt}}(V+W) \geq 0$. Since at $t = 0$, we have $V(0) + W(0) > 0$ (which can also be verified by Figure 4.17). So $V(t) + W(t) > 0$ for $t \geq 0$, which means that $V > -W$. Together with Lemma 4.4.2, we have that

$$\int_0^{1/2} v^2(z)\mathrm{d}z \leq -\frac{1}{2}VW \leq \frac{1}{2}V^2.$$

As a consequence, we always have

$$C(t) = ((3n-2)\varepsilon - n + 2)\int_0^{1/2} v^2\mathrm{d}z > \frac{1}{2}\left((3n-2)\varepsilon - n + 2\right)V^2.$$

Therefore, we turn back to the governing equation for V, and obtain

$$\begin{aligned} V_t &= \frac{1}{2}(n-2-n\varepsilon)V^2 + C(t) \\ &\geq \frac{1}{2}(n-2-n\varepsilon)V^2 + \frac{1}{2}\left((3n-2)\varepsilon - n + 2\right)V^2 \\ &= \varepsilon(n-1)V^2. \end{aligned}$$

So when $0 < \varepsilon < \frac{n-2}{3n-2}$, V will blow up in finite time.

Lastly, when $\varepsilon = 0$, the 1D model (4.12) can be simplified as

$$v_t = \frac{1}{2}(n-2)v^2 - (n-2)\int_0^{1/2} v^2(z)\mathrm{d}z.$$

Since $W < 0$, we have, by Lemma 4.4.2:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{dt}}(V+W) &= \frac{1}{2}(n-2)\left(V^2+W^2\right) + 2C(t) \\ &= \frac{1}{2}(n-2)\left(V^2+W^2\right) - 2(n-2)\int_0^{1/2} v^2(z)\mathrm{d}z \\ &\geq \frac{1}{2}(n-2)\left(V^2+W^2\right) + (n-2)VW \\ &= \frac{1}{2}(n-2)\left(V+W\right)^2. \end{aligned}$$

Since $V(0)+W(0) > 0$, $V+W$ will become infinite in finite time, and so will $v(0,t)$, because $v(0,t) = V > V+W$. Therefore, we also have a finite-time blow-up for ω_1 in this case.

We conclude our result in the below:

Theorem 4.4.1. *For any $\varepsilon < 1 - \frac{2}{n}$, there exist smooth solutions of our 1D model* (4.9) *in $\mathbb{R}^n$ that form singularity in finite time.*

For the future study, it would be interesting to study the quality of the approximation to the original model (4.3) by our 1D model (4.9), especially for ε close to $1 - \frac{2}{n}$. It is worthwhile to mention that if we could generalize Theorem 4.4.1 to the original model (4.3), then we notice that as $n \to +\infty$, the upper bound $1 - \frac{2}{n}$ for ε that admits finite-time blow-up would approach 1, whose case would asymptotically recover the n-D axisymmetric Euler equations with no swirl. Since in Theorem 4.4.1 we only use a smooth initial condition for ω_1, this would give a numerical answer to the Question 7 of [27] in the infinite-dimensional limit case, where the authors ask if the n-D axisymmetric Euler equations with no swirl can form singularity in finite-time from smooth initial data when $n \geq 4$.

Part II

Sampling of High-Dimensional Distributions by Deep Generative Networks

Chapter 5

MULTISCALE INVERTIBLE GENERATIVE NETWORKS FOR HIGH-DIMENSIONAL DISTRIBUTIONS

5.1 Background Review

In this section, we review several important concepts and recent studies in high-dimensional distribution sampling using deep generative networks.

The Transport Map Approach

The transport map is a deterministic nonlinear transformation that links two probability measures and their samples, and it has become a very popular approach to the Bayesian inverse problem recently. Specifically, a map $T : \mathbb{R}^{d_\gamma} \to \mathbb{R}^d$ is a transport map if it pushes forward a reference probability measure γ in $\mathbb{R}^{d_\gamma}$ to a probability measure of interest ν in $\mathbb{R}^d$, for example, the one defined in (1.6). The push forward relation, denoted as $T_\sharp \gamma = \nu$, means that

$$\nu(A) = \gamma(T^{-1}(A)), \quad \text{where } T^{-1}(A) := \{x \in \mathbb{R}^{d_\gamma} | T(x) \in A\}, \tag{5.1}$$

for any Borel measurable set $A \subset \mathbb{R}^d$. It is well-known that when the target distribution ν is non-atomic, there exists a transport map that pushes forward γ to ν, see [6, 72, 93]. However, the uniqueness is not guaranteed in general.

Compared to the MCMC-type methods [4, 5, 75, 79, 20] and the SVGD-related methods [69, 14, 13], the transport map approach is more advantageous in the efficient sampling process. Given the transport map T between the reference measure γ and the target meaure ν, and given independent and identically distributed (i.i.d.) samples $\{x_k\}_{k=1}^n$ from γ, which is usually chosen as some simple and well-known distribution like Gaussian, we get i.i.d. samples of ν immediately as $\{T(x_k)\}_{k=1}^n$.

If $d_\gamma = d$ and T is a diffeomorphism (invertible, and both T and T^{-1} are differentiable), the density function π of γ is linked with the density function q of ν by the change-of-variable rule. Let $\mathrm{J}_x T$ be the Jacobian of T with respect to x. We have

$$q(x) = \pi(T^{-1}(x)) |\det \mathrm{J}_x T^{-1}(x)|. \tag{5.2}$$

The existence of such an invertible transport map with Jacobian almost everywhere with respect to γ can be guaranteed by the absolute continuity of reference measure γ and target measure ν when $d_\gamma = d$. For example, the Knothe–Rosenblatt rearrangement [81, 56] is a special example of such a transport map.

In Bayesian inverse problems, we design a parametric family $\{T_\theta : \theta \in \Theta\}$ for the transport map and choose the parameter θ via variational inference. Essentially, it seeks the optimizer of a learning objective over the parameter θ, which usually measures the approximation of our working distribution $T_{\theta\sharp}\gamma$ to the target ν such as in (5.1), and applies some regularization or constraint. For example, in [29], the transport map is represented by multi-variate polynomials, and the Kullback-Leibler divergence (which we will denote as the KL divergence for short) is used to assess the approximation quality. The optimization is further regularized by the Wasserstein metric, and the transport map is constrained to be a Knothe–Rosenblatt re-arrangement. In [69], the transport map is a perturbation of the identity map by elements from a reproducing kernel Hilbert space, and the approximation is measured by the KL divergence. Other designs also include certain implicit maps [17, 74]. Sometimes the uniqueness of the optimum is sacrificed because any transport map satisfying (5.1) is sufficient for sample generation.

Deep Generative Network as a Transport Map

The recent development of deep learning techniques offers an alternative to represent the transport map. In fact, the deep generative network has already been very successful in machine learning tasks like natural image synthesis [54, 96], where it is used to model a transport map from a reference distribution to the distribution of natural images. The deep generative network has the advantage of large capacity and that its complexity has weak dimension dependence. It also has flexible scalability, in that the more computational resources we have, the larger the network we can use, and better the result we can expect.

As examples of deep generative networks, generative adversarial network (GAN) [34] and variational autoencoder (VAE) [53] learn a non-invertible transport map, while flow-based generative models (also called invertible flows) [23, 24, 54] represent an invertible transport map whose log determinant of Jacobian is also accessible. The difference in modeling an invertible or non-invertible transport map leads to different available choices of the learning objective. We would like to point out that the GAN and the VAE are both originally designed for sample generation in the natural image synthesis task. The image synthesis task defines the target distribution by giving a large number of i.i.d. samples from that distribution, but the Bayesian inverse problem defines the target by its unnormalized probability density.

One typical flow-based generative model considered here is known as the Glow [54], which is also originally designed for the image synthesis task. The Glow model designs an invertible transport map T by composition of units of the transport map:

$$T = T_1 \circ T_2 \circ \cdots \circ T_n,$$

where T_i is a unit of the transport map called invertible blocks. In [54], each invertible block T_i is a concatenation of three invertible units: actnorm, invertible 1×1 convolution, and affine coupling.

Let x and y both be 3D tensors of size $h \times w \times c$, which is the popular format to store objects of images in original applications of the Glow. In the following, we will use x and y to show the operations encoded in each invertible unit. We remark that c is used to denote the color channel dimension, and so is usually assumed to be a small number.

The actnorm unit essentially performs element-wise shifting and scaling: if y is the output of x through the actnorm unit, then

$$y_{i,j,k} = s_k x_{i,j,k} + b_k,$$

for $i = 1, \ldots, h$, $j = 1, \ldots, w$ and $k = 1, \ldots, c$. Here s and b are the parameters of the actnorm unit, which encode the scaling and shift information. In the reverse direction, to map from y to x, we have

$$x_{i,j,k} = (y_{i,j,k} - b_k)/s_k,$$

as long as elements in b are non-zero. More interestingly, the Jacobian matrix of the forward map from x to y is diagonal, and therefore the log determinant of the Jacobian can be given as $hw \sum_{k=1}^{c} \log |s_k|$. Due to the scaling and shift operation, the actnorm unit can serve as a normalization to increase numerical stability and robustness.

The unit of invertible 1×1 convolution can be seen as the generalization of the actnorm unit. It performs linear transformation on the third dimension: if y is the output of x through the invertible 1×1 convolution unit, then

$$y_{i,j,:} = W x_{i,j,:},$$

for $i = 1, \dots, h$ and $j = 1, \dots, w$. Here W is an invertible $c \times c$ matrix, and $x_{i,j,:}$, $y_{i,j,:}$ are vectors of length c. Due to the fact that the third dimension c is small, the matrix W is easy to invert, so the inversion from y to x can be given by

$$x_{i,j,:} = W^{-1} y_{i,j,:}.$$

Similarly, the Jacobian matrix of the unit of invertible 1×1 convolution is block diagonal with many small-sized blocks, making the log determinant easy to compute: $hw \log \det |W|$. The invertible 1×1 convolution unit not only scales the third dimension, but also introduces permutation along the third dimension. The parameters of the invertible 1×1 convolution unit are the $c \times c$ matrix W, which in practice is stored in its PLU decomposition for easy inversion and easy access to its determinant.

The affine coupling unit is a triangular transform which introduces nonlinearity. We first split the tensor x into x_1 and x_2, so that $\dim(x) = \dim(x_1) + \dim(x_2)$. Splitting can be done by simply cutting the tensor into two parts. For example, we can cut x along the first dimension and get $x_1 = x_{1:h_1,:,:}$ and $x_2 = x_{h_1+1:h,:,:}$ with some number $1 < h_1 < h$. Then we choose two arbitrary nonlinear maps f and g from the space of x_2 to the space of x_1, and map

$$s = f(x_2), \quad b = g(x_2).$$

Finally, we let

$$(y_1)_{i,j,k} = s_{i,j,k} \, (x_1)_{i,j,k} + b_{i,j,k}, \quad y_2 = x_2,$$

for $i = 1, \dots, h$, $j = 1, \dots, w$ and $k = 1, \dots, c$. The output y is the concatenation of y_1 and y_2, so that $\dim(y) = \dim(y_1) + \dim(y_2)$. For example, if y_1 has size $h_1 \times w \times c$, and y_2 has size $(h - h_1) \times w \times c$, then we can concatenate y_1 and y_2 along the first dimension to get an $h \times w \times c$ tensor y. The Jacobian matrix of this affine coupling unit is a block-triangular matrix, and surprisingly the log determinant of the Jacobian can be written as

$$\sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{k=1}^{c} \log \left| (f(x_2))_{i,j,k} \right|.$$

Due to the block-triangular structure, we can also map backward from y to x by first splitting y into y_1 and y_2, then obtaining x_1 and x_2 by

$$(x_1)_{i,j,k} = \left((y_1)_{i,j,k} - b_{i,j,k} \right) / s_{i,j,k}, \quad x_2 = y_2,$$

where $s = f(y_2)$, and $b = g(y_2)$, and in the end, concatenating x_1 and x_2 to recover x. The parameters of the affine coupling unit only appear in the parametric functions f and g. The functions f and g introduce nonlinearity to the affine coupling unit. In practice, the functions f and g are modeled by deep neural networks.

The chain rule allows us to pile up these invertible units together as a transport map unit T_i, and pile up T_i together as the transport map T, with its inverse available and the log determinant of its Jacobian computable. Consequently, the Glow model is a parametric transport map $T = T_\theta$, where parameters θ is the collection of parameters in the invertible units. Since the map $T = T_\theta$ is invertible, we can evaluate the density using (5.2).

The application of deep generative networks to the Bayesian inverse problem is not new to us. The authors in [48] represented the transport map by non-invertible generative networks, and the works in [80, 2, 59, 57] used invertible networks. As shown by (5.2), the invertibility of the map enables the evaluation of density, which in turn allows more effective learning objectives. The works in [80, 2, 59] used learning objectives like the maximum mean discrepancy, or the KL divergence, while the training presented in [57] resembles a GAN. Algorithms in [80, 57] can face potential difficulties in scaling up for high-dimensional problems. Also, due to the tricky non-convex optimization problem and the property of the learning objective, the approaches in both invertible [2, 59] and non-invertible [48] generative networks could encounter some challenges such as mode collapse as the dimension grows. Our proposed method is different from them both in the network architecture and in the training strategy, and targets at the high-dimensional cases of the Bayesian inverse problem.

5.2 Low-Dimensional Structure in the Posterior Distribution

We propose the Multiscale Invertible Generative Network, which we denote as MsIGN for short. The MsIGN makes use of the low-dimensional structure in the posterior distribution and generates samples via the transport map approach. In this section, we discuss the low-dimensional structure in the posterior distribution that motivates our method.

The target Bayesian posterior ν essentially re-factorizes the prior μ with respect to the likelihood $\mathcal{L}$. As implied by (1.6), the Radon-Nikodym derivative $\frac{\mathrm{d}\nu}{\mathrm{d}\mu}(x)$ is proportional to the likelihood function $\mathcal{L}(x; y)$. Roughly speaking, due to the limited number of observations or measurements, which is usually small compared to the problem dimension ($s < d$), the difference between the high-dimensional posterior ν and the prior μ most likely lies in a low-dimensional subspace. We refer to [86, 98] for detailed discussion about this low-dimensional difference. Since the prior μ is usually tractable and easy to sample from, we attack the high-dimensional challenge by exploring the low-dimensional structure in the likelihood, and sequentially approximating the high-dimensional posterior by low-dimensional surrogates.

Surrogate Distribution for the Posterior

A typical and common case of such low-dimensional approximation to the Bayesian posterior arises when the hidden system states $x \in \mathbb{R}^d$ represent the parameterization of some spatial or temporal quantities. An example of such quantity can be the permeability field as a variable of space position in the Darcy flow, or the reaction rate as a variable of time in chemical kinetics. For such spatial or temporal x, it is very common that spatial or temporal variation of x comes from some multiscale structure from coarse (a low-dimensional version of x) to fine (the original x). In other words, the concept of resolution or scale naturally arises in x. To build the low-dimensional approximation, we first introduce a deterministic upscaling operator $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}^{d_c}$ such that $x_c = \mathcal{A}(x)$ links the original fine-scale $x \in \mathbb{R}^d$ to its coarse-scale version $x_c \in \mathbb{R}^{d_c}$. Since the coarse-scale variable is always lower-dimensional, we have $d_c < d$. The upscaling operator $\mathcal{A}$ can either be a linear operator that averages the value of x in individual regions, like the average pooling operator, or be a nonlinear operator that homogenizes the fine-scale variation of x, like methods in [28, 1, 8]. Despite unavoidable inaccuracy due to the information loss, the coarse-scale version x_c should still preserve the ability to give informative prediction of system observables. The corresponding forward map based on the coarse-scale x_c is denoted by $\mathcal{F}_c : \mathbb{R}^{d_c} \to \mathbb{R}^{d_y}$.

We say a Bayesian inverse problem has the *multiscale* property, if the original fine-scale forward map $\mathcal{F}$ in (1.1) can be well approximated by the coarse-scale forward map (5.6) associated with the upscaled coarse-scale variable by $\mathcal{A}$:

$$\mathcal{F}_c(\mathcal{A}(x)) \approx \mathcal{F}(x), \quad \text{for } x \in \mathbb{R}^d. \tag{5.3}$$

To quantitatively define the multiscale property, we introduce the definitions and assumptions

Assumption 5.2.1. *Assume that the forward map $\mathcal{F}$ has a finite bound with respect* *to the prior,*

$$C_0 := \int \|\mathcal{F}(x)\|_\Gamma^2 \rho(x)\mathrm{d}x < +\infty, \tag{5.4}$$

and assume the approximation of (5.3) *satisfies*

$$\delta_{\mathcal{A}} := \int \|\mathcal{F}(x) - \mathcal{F}_c(\mathcal{A}(x))\|_\Gamma^2 \rho(x)\mathrm{d}x < +\infty. \tag{5.5}$$

Using the above notations, the *multiscale* property of a Bayesian inverse problem is characterized by the quantity $\delta_{\mathcal{A}}$.

A popular example of a multiscale Bayesian inverse problem is, for example, in systems like elliptic equations with multiscale coefficients, numerical homogenization provides a very good approximation using coarse-scale features, see [28, 1, 8]. The approximation (5.3) then essentially characterizes the low-dimensional structure in the likelihood function $\mathcal{L}$. A similar idea has also been exploited in other approaches to the Bayesian inverse problem, including [98, 7, 14, 13].

Similar to (1.4), we can define a coarse-scale likelihood function:

$$\mathcal{L}_c(x_c; y) = \exp\left(-\frac{1}{2}\|y - \mathcal{F}_c(x_c)\|_\Gamma^2\right). \tag{5.6}$$

The approximation (5.3) motivates us to define a surrogate posterior distribution $\tilde{\nu}$ by $\frac{\mathrm{d}\tilde{\nu}}{\mathrm{d}\mu}(x) \propto \mathcal{L}_c(\mathcal{A}(x); y)$ such that $\tilde{\nu}$ is close to the target posterior ν. The probability density $\tilde{q}$ of $\tilde{\nu}$ for $x \in \mathbb{R}^d$ is given by

$$\tilde{q}(x) := \frac{1}{\tilde{Z}}\rho(x)\mathcal{L}_c(\mathcal{A}(x); y), \tag{5.7}$$

where the normalizing constant $\tilde{Z}$ is given by

$$\tilde{Z}(y) = \int \rho(x)\mathcal{L}_c(\mathcal{A}(x); y)\mathrm{d}x.$$

Roughtly speaking, since $\mathcal{F}_c(\mathcal{A}(x)) \approx \mathcal{F}(x)$, we expect $\mathcal{L}_c(\mathcal{A}(x); y) \approx \mathcal{L}(x; y)$, and therefore we have

$$\tilde{q}(x) = \frac{1}{\tilde{Z}}\rho(x)\mathcal{L}_c(\mathcal{A}(x); y) \approx \frac{1}{\tilde{Z}}\rho(x)\mathcal{L}(x; y) = \frac{Z}{\tilde{Z}}q(x). \tag{5.8}$$

In [98], authors used a similar low-rank approximation of the target distribution as our $\tilde{\nu}$ in (5.8), whereas they focused on a linear upscaling operator $\mathcal{A}$, and targeted at searching for an optimal low-rank approximation. The surrogate distribution $\tilde{\nu}$

in our framework, however, is designed as an intermediate step to capture the target distribution ν. We will build a transport map to bridge the difference between $\tilde{\nu}$ and ν as in Algorithm 1.

We recall that the definition of the Jeffreys divergence [50] between two distributions q and $\tilde{q}$ is given by

$$D_{\mathrm{J}}(q\|\tilde{q}) := D_{\mathrm{KL}}(q\|\tilde{q}) + D_{\mathrm{KL}}(\tilde{q}\|q), \tag{5.9}$$

where D_{KL} is the Kullback-Leibler divergence, which is given by

$$D_{\mathrm{KL}}(q\|\tilde{q}) = \mathbb{E}_{x\sim q}\left[\log\frac{q(x)}{\tilde{q}(x)}\right], \quad D_{\mathrm{KL}}(\tilde{q}\|q) = \mathbb{E}_{x\sim\tilde{q}}\left[\log\frac{\tilde{q}(x)}{q(x)}\right].$$

In fact, the approximation of the surrogate $\tilde{q}$ to the posterior q can be characterized by the following theorem, inspired by [14]:

Theorem 5.2.1. *Assume that Assumption 5.2.1 holds, and further assume that $\delta_{\mathcal{A}}$ is smaller than some constant $\delta_0 = \delta_0(y, C_0, Z)$, which only depends on the observation data y, the forward map bound in the prior C_0, and the normalizing constant Z. Then the Jeffreys divergence between q in* (1.6) *and $\tilde{q}$ in* (5.8) *is bounded by*

$$D_{\mathrm{J}}(q\|\tilde{q}) \le C\delta_{\mathcal{A}}^{1/2}. \tag{5.10}$$

Here the constant $C = C(y, C_0, Z)$ only depends on y, C_0, and Z.

Proof. Let $I(x) := |\log\mathcal{L}(x;y) - \log\mathcal{L}_c(\mathcal{A}(x);y)|$, we have

$$\begin{aligned}2I(x) &= \left|\|y-\mathcal{F}(x)\|_\Gamma^2 - \|y-\mathcal{F}_c(\mathcal{A}(x))\|_\Gamma^2\right|\\ &= \left|(2y-\mathcal{F}(x)-\mathcal{F}_c(\mathcal{A}(x)))^T\,\Gamma^{-1}\left(\mathcal{F}(x)-\mathcal{F}_c(\mathcal{A}(x))\right)\right|.\end{aligned}$$

Using the Cauchy-Schwarz inequality and the triangular inequality, we have

$$2I(x) \le (2\|y\|_\Gamma + \|\mathcal{F}(x)-\mathcal{F}_c(\mathcal{A}(x))\|_\Gamma + 2\|\mathcal{F}(x)\|_\Gamma) \times \|\mathcal{F}(x)-\mathcal{F}_c(\mathcal{A}(x))\|_\Gamma.$$

Therefore, we proceed to bound the integral $\int I(x)\rho(x)\mathrm{d}x$ as

$$\begin{aligned}\int I(x)\rho(x)\mathrm{d}x \le&\|y\|_\Gamma\int \|\mathcal{F}(x)-\mathcal{F}_c(\mathcal{A}(x))\|_\Gamma\rho(x)\mathrm{d}x\\ &+\frac{1}{2}\int \|\mathcal{F}(x)-\mathcal{F}_c(\mathcal{A}(x))\|_\Gamma^2\rho(x)\mathrm{d}x\\ &+\int \|\mathcal{F}(x)\|_\Gamma\|\mathcal{F}(x)-\mathcal{F}_c(\mathcal{A}(x))\|_\Gamma\rho(x)\mathrm{d}x.\end{aligned} \tag{5.11}$$

On the other hand, the weighted integral Cauchy-Schwarz inequality gives

$$\int \|\mathcal{F}(x) - \mathcal{F}_c(\mathcal{A}(x))\|_\Gamma \rho(x)\mathrm{d}x \le \delta_{\mathcal{A}}^{1/2}.$$

Here we recall that $\delta_{\mathcal{A}} = \int \|\mathcal{F}(x) - \mathcal{F}_c(\mathcal{A}(x))\|_\Gamma^2 \rho(x)\mathrm{d}x$. We also recall that $C_0 = \int \|\mathcal{F}(x)\|_\Gamma^2 \rho(x)\mathrm{d}x$, so we also have

$$\int \|\mathcal{F}(x)\|_\Gamma \|\mathcal{F}(x) - \mathcal{F}_c(\mathcal{A}(x))\|_\Gamma \rho(x)\mathrm{d}x \le C_0^{1/2}\delta_{\mathcal{A}}^{1/2}.$$

Thus the bound (5.11) can be given explicitly

$$\int I(x)\rho(x)\mathrm{d}x \le \|y\|_\Gamma \delta_{\mathcal{A}}^{1/2} + C_0^{1/2}\delta_{\mathcal{A}}^{1/2} + \frac{1}{2}\delta_{\mathcal{A}} = \left(\|y\|_\Gamma + C_0^{1/2} + \frac{1}{2}\delta_{\mathcal{A}}^{1/2}\right)\delta_{\mathcal{A}}^{1/2}.$$

Now for the KL divergence $D_{\mathrm{KL}}(q\|\tilde{q})$, we have

$$\begin{aligned} D_{\mathrm{KL}}(q\|\tilde{q}) &= \int \log\left(\frac{\mathcal{L}(x;y)\tilde{Z}}{\mathcal{L}_c(\mathcal{A}(x);y)Z}\right)\frac{1}{Z}\mathcal{L}(x;y)\rho(x)\mathrm{d}x \\ &= \frac{1}{Z}\int \log\left(\frac{\mathcal{L}(x;y)}{\mathcal{L}_c(\mathcal{A}(x);y)}\right)\mathcal{L}(x;y)\rho(x)\mathrm{d}x + \log\frac{\tilde{Z}}{Z}. \end{aligned}$$

Since $0 < \mathcal{L}(x;y) = \exp(-\frac{1}{2}\|y - \mathcal{F}(x)\|_\Gamma^2) \le 1$, we can go further by

$$\begin{aligned} D_{\mathrm{KL}}(q\|\tilde{q}) &\le \frac{1}{Z}\int \left|\log\left(\frac{\mathcal{L}(x;y)}{\mathcal{L}_c(\mathcal{A}(x);y)}\right)\right|\rho(x)\mathrm{d}x + \log\frac{\tilde{Z}}{Z} \\ &= \frac{1}{Z}\int I(x)\rho(x)\mathrm{d}x + \log\frac{\tilde{Z}}{Z} \\ &\le \frac{1}{Z}\left(\|y\|_\Gamma + C_0^{1/2} + \frac{1}{2}\delta_{\mathcal{A}}^{1/2}\right)\delta_{\mathcal{A}}^{1/2} + \log\frac{\tilde{Z}}{Z}. \end{aligned}$$

For the other KL divergence $D_{\mathrm{KL}}(\tilde{q}\|q)$, similarly we have

$$D_{\mathrm{KL}}(\tilde{q}\|q) \le \frac{1}{\tilde{Z}}\left(\|y\|_\Gamma + C_0^{1/2} + \frac{1}{2}\delta_{\mathcal{A}}^{1/2}\right)\delta_{\mathcal{A}}^{1/2} + \log\frac{Z}{\tilde{Z}}.$$

Putting these estimates together, we have

$$D_{\mathrm{J}}(q\|\tilde{q}) = D_{\mathrm{KL}}(q\|\tilde{q}) + D_{\mathrm{KL}}(\tilde{q}\|q) \le \left(\frac{1}{\tilde{Z}} + \frac{1}{Z}\right)\left(\|y\|_\Gamma + C_0^{1/2} + \frac{1}{2}\delta_{\mathcal{A}}^{1/2}\right)\delta_{\mathcal{A}}^{1/2}. \tag{5.12}$$

Finally, we deal with the constant terms. We observe that

$$|Z - \tilde{Z}| = \left|\int \mathcal{L}(x;y)\rho(x)\mathrm{d}x - \int \mathcal{L}_c(\mathcal{A}(x);y)\rho(x)\mathrm{d}x\right|$$

$$\leq \int |\mathcal{L}(x;y) - \mathcal{L}_c(\mathcal{A}(x);y)|\rho(x)\mathrm{d}x.$$

Noticing that $|e^x - e^y| \leq |x-y|$ for $x, y \leq 0$, and $\log \mathcal{L}(x;y) = -\frac{1}{2}\|y - \mathcal{F}(x)\|_\Gamma^2 < 0$, $\log \mathcal{L}_c(\mathcal{A}(x);y) = -\frac{1}{2}\|y - \mathcal{F}_c(\mathcal{A}(x))\|_\Gamma^2 < 0$. Thus, we further obtain

$$\begin{aligned}|Z - \tilde{Z}| &\leq \int |\log \mathcal{L}(x;y) - \log \mathcal{L}_c(\mathcal{A}(x);y)|\rho(x)\mathrm{d}x = \int I(x)\rho(x)\mathrm{d}x \\ &\leq \left(\|y\|_\Gamma + C_0^{1/2} + \frac{1}{2}\delta_{\mathcal{A}}^{1/2}\right)\delta_{\mathcal{A}}^{1/2}.\end{aligned}$$

This bound suggests that there exists $\delta_0 = \delta_0(y, C_0, Z)$ which only depends on y, C_0 and Z, such that if $\delta_{\mathcal{A}} < \delta_0$, then $|Z - \tilde{Z}| < Z/2$. Therefore, the bound (5.12) can be simplified to

$$\begin{aligned}D_{\mathrm{J}}(q\|\tilde{q}) &\leq \left(\frac{1}{Z/2} + \frac{1}{Z}\right)\left(\|y\|_\Gamma + C_0^{1/2} + \frac{1}{2}\delta_{\mathcal{A}}^{1/2}\right)\delta_{\mathcal{A}}^{1/2} \\ &= \frac{3}{Z}\left(\|y\|_\Gamma + C_0^{1/2} + \frac{1}{2}\delta_{\mathcal{A}}^{1/2}\right)\delta_{\mathcal{A}}^{1/2}.\end{aligned}$$

Let $C = \frac{3}{Z}\left(\|y\|_\Gamma + C_0^{1/2} + \frac{1}{2}\delta_0^{1/2}\right)$, which only depends on Z, C_0 and y. We conclude that when $\delta_{\mathcal{A}} < \delta_0$, $D_{\mathrm{J}}(q\|\tilde{q}) \leq C\delta_{\mathcal{A}}^{1/2}$. □

Scale Decoupling

When x follows the prior μ, the distribution of $x_c = \mathcal{A}(x)$ is $\mu_c = \mathcal{A}_\sharp\mu$, which is the push-forward of μ by $\mathcal{A}$. Let ρ_c be the density function of $\mathcal{A}_\sharp\mu$, the conditional probability rule gives that $\rho(x|\mathcal{A}(x)) = \rho(x)/\rho_c(\mathcal{A}(x))$. So we conclude

$$\rho(x) = \rho_c(\mathcal{A}(x))\rho(x|\mathcal{A}(x)). \tag{5.13}$$

We interpret (5.13) as a recovery game for sample generation. To sample x from ρ, one can first sample its coarse-scale version $\mathcal{A}(x)$ from ρ_c, and then recover missing fine-scale details while preserving the coarse-scale structure by sampling from the conditional distribution $\rho(x|\mathcal{A}(x))$.

With the coarse-scale prior μ_c and the coarse-scale likelihood $\mathcal{L}_c$ in (5.6), we define a coarse-scale posterior ν_c by $\frac{\mathrm{d}\nu_c}{\mathrm{d}\mu_c}(x_c) \propto \mathcal{L}_c(x_c;y)$, whose density function is

$$q_c(x_c) = \frac{1}{Z_c(y)}\rho_c(x_c)\mathcal{L}_c(x_c;y). \tag{5.14}$$

where the constant is given by

$$Z_c(y) = \int \rho_c(x_c)\mathcal{L}_c(x_c;y)\mathrm{d}x_c.$$

An important observation is that the coarse-scale posterior (5.14) and the surrogate posterior in (5.8) can be bridged by our conditional prior (5.13):

$$\begin{aligned}\tilde{q}(x) &= \frac{1}{\tilde{Z}}\rho(x)\mathcal{L}_c(\mathcal{A}(x);y) = \frac{1}{\tilde{Z}}\rho_c(\mathcal{A}(x))\rho(x|\mathcal{A}(x))\mathcal{L}_c(\mathcal{A}(x);y)\\ &= \frac{Z_c}{\tilde{Z}}\rho(x|\mathcal{A}(x))q_c(\mathcal{A}(x)) \propto \rho(x|\mathcal{A}(x))q_c(\mathcal{A}(x)).\end{aligned} \tag{5.15}$$

The scale decoupling of the surrogate distribution in (5.15) can be used to construct samples from the target posterior ν as summarized in Algorithm 1. Since it assumes the capture of the coarse-scale distribution ν_c, Algorithm 1 only serves as a conceptual guideline of our strategy.

Algorithm 1 An Ideal Sampling Algorithm

```
    Output: Sample $x$ from the target distribution $\nu$
1:  Sample $x_c$ from the coarse-scale distribution $\nu_c$.
2:  Sample $\tilde{x}$ from the prior conditional distribution $\rho(\tilde{x}|\mathcal{A}(\tilde{x}) = x_c)$.
3:  Learn a transport map $F$ that pushes forward $\tilde{\nu}$ to $\nu$.
4:  Obtain sample $x$ from the target distribution $\nu$ by $x = F(\tilde{x})$.
```

As a remark, in step 2, $\tilde{x}$ ideally will follow the surrogate distribution $\tilde{\nu}$. Since Theorem 5.2.1 implies that $\tilde{\nu}$ is not far away from ν, there exists a transport map F that is close to the identity map. We introduce the details of the learning of the transport map F in Sections 5.3 and 5.4.

Comparison with the Low-Dimensional Structure in Other Works

In [78], the authors proposed a similar formulation as (5.15). In their setting, a likelihood function has the multiscale structure, if there exists a coarse-scale *random variable* γ of dimension d_c with $d_c < d$ and a likelihood $\mathcal{L}_c$ such that

$$\mathcal{L}(x,\gamma;y) = \mathcal{L}_c(\gamma;y). \tag{5.16}$$

Here $\mathcal{L}(x,\gamma;y)$ is the joint likelihood of (x,γ) given the observation y. Then the *joint posterior distribution* of the fine- and coarse-scale parameters (x,γ) can be decoupled as

$$\begin{aligned}q(x,\gamma) \propto \rho(x,\gamma)\mathcal{L}(x,\gamma;y) &\overset{(i)}{=} \rho(x,\gamma)\mathcal{L}_c(\gamma;y)\\ &\overset{(ii)}{=} \rho(x|\gamma)\rho(\gamma)\mathcal{L}_c(\gamma;y) \overset{(iii)}{=} \rho(x|\gamma)q_c(\gamma),\end{aligned} \tag{5.17}$$

with normalizing constants omitted in the equivalence relations. Here, we use the definition of multiscale structure (5.16) in [78] in (i). For (ii) we apply the conditional probability rule $\rho(x,\gamma) = \rho(x|\gamma)\rho(\gamma)$. In (iii), the authors in [78] defined the posterior in the coarse scale as $q_c(\gamma) := \rho(\gamma)\mathcal{L}_c(\gamma; y)$.

There are two important differences in these two definitions. First of all, our coarse-scale parameter x_c is a deterministic function of the fine-scale parameter x, while in [78], γ is a random variable that may contain extra randomness outside x (as demonstrated in numerical examples in [78]). This difference in definition results in significant difference in modeling: our invertible model has d-dimensional random noise z as input to approximate the target posterior $q(x)$, while models in [78] have $(d + d_c)$-dimensional random noise as input to approximate the joint-posterior $q(x,\gamma)$. Another consequence is that users need to define the joint prior $\rho(x,\gamma)$ in [78], while in our definition the prior of x_c is naturally induced by the prior of x.

Secondly, our multiscale structure (5.8) is an approximate relation and we use an invertible flow in our MsIGN to model this approximation, while in [78] the multiscale structure (5.17) is an exact relation and authors treat the prior-upsampled solution $\rho(x|\gamma)q_c(\gamma)$, which is in the right-hand side of (5.17), as the final solution. Our approximate multiscale relation and further treatment by the invertible flow enable us to apply the method recursively in a multiscale fashion, while in [78] the proposed method is essentially a two-scale method and there is not further correction based on the prior-upsampled solution $\rho(x|\gamma)q_c(\gamma)$ at the fine-scale.

Finally, the invertible model in [78] consists of multivariate polynomials, which suffer from the exponential growth of polynomial coefficients as dimension grows. In our work, the invertible model is deep generative networks, whose parameter dimension has a weak dependence on the problem dimension.

We also observe that [86, 14, 13] seek a best low-rank approximation of the posterior, and treat the approximation as the final solution with no extra modification. As we will see in Section 5.5, the true posterior could still be far away from the prior-upsampled solution, especially in the first few coarse scales.

In addition, while in [2], flow-based generative models are also used to approximate the distribution of inverse problems, their definition of posterior is not equivalent to ours, as they assume no error in measurement (1.1). Furthermore, as their training strategy looks to capture the target distribution while simultaneously learning the forward map $\mathcal{F}$, they mainly focus on low-dimensional Bayesian inverse problems, in contrast with our high-dimensional setting here.

5.3 Network Architecture of Multiscale Invertible Generative Networks

To carry out the ideal step 1 of Algorithm 1, we apply the above procedure recursively until the dimension of the coarsest scale is small enough so that the corresponding coarsest-scale posterior ν_c can be easily sampled by a standard method. To elaborate this hierarchical divide-and-conquer strategy, we introduce the following rule of notations to distinguish between scales: let L be the number of scales, and for $1 \leq l \leq L$, let $x_l \in \mathbb{R}^{d_l}$ be the quantity of interest at scale l, and $\mathcal{A}_l : \mathbb{R}^{d_l} \to \mathbb{R}^{d_{l-1}}$ be the upscaling operator connecting scale l and $l-1$: $x_{l-1} = \mathcal{A}_l(x_l)$. The final scale x_L coincides with our original target x, and the dimension goes up as l increases: $d_1 < d_2 < \ldots < d_L = d$.

At the final scale L, x_L inherits the original prior distribution $\mu_L = \mu$, whose density is $\rho_L = \rho$, likelihood function $\mathcal{L}_L(x_L; y) = \mathcal{L}(x; y)$ and the posterior distribution $\nu_L = \nu$, whose density is $q_L = q$, as defined in (1.4) and (1.6). At scale l, $1 \leq l \leq L-1$, the forward map $\mathcal{F}_l(x_l) : \mathbb{R}^{d_l} \to \mathbb{R}^s$ is a coarse-scale approximation of the next scale: $\mathcal{F}_l(\mathcal{A}_{l+1}(x_{l+1})) \approx \mathcal{F}_{l+1}(x_{l+1})$, analog to (5.3). Following (5.14), we define the posterior ν_l at scale l by $\frac{\mathrm{d}\nu_l}{\mathrm{d}\mu_l} \propto \mathcal{L}_l(x_l; y)$, whose density is

$$q_l(x_l) = \frac{1}{Z_l(y)} \rho_l(x_l) \mathcal{L}_l(x_l; y), \tag{5.18}$$

with

$$Z_l(y) = \int \rho_l(x_l) \mathcal{L}_l(x_l; y) \mathrm{d}x_l.$$

Here ρ_l is the density of the prior distribution μ_l at scale l, which is defined as: $\mu_l := \mathcal{A}_{l+1\sharp}\mu_{l+1} = (\mathcal{A}_{l+1} \circ \cdots \circ \mathcal{A}_L)_\sharp \, \mu_L$. And $\mathcal{L}_l(x_l; y)$ is the likelihood at scale l:

$$\mathcal{L}_l(x_l; y) := \exp\left(-\frac{1}{2}\|y - \mathcal{F}_l(x_l)\|_\Gamma^2\right).$$

In an analogous manner, the surrogate distribution $\tilde{\nu}_l$ is an approximation to ν_l for $2 \leq l \leq L$. Its density $\tilde{q}_l$ is given by

$$\tilde{q}_l(x_l) = \frac{1}{\tilde{Z}_l(y)} \rho_l(x_l) \mathcal{L}_{l-1}(\mathcal{A}_l(x_l); y), \tag{5.19}$$

with

$$\tilde{Z}_l(y) = \int \rho_l(x_l) \mathcal{L}_{l-1}(\mathcal{A}_l(x_l); y) \mathrm{d}x_l.$$

Following (5.15), $\tilde{\nu}_l$ is closely connected to the last-scale posterior ν_{l-1} by a prior conditional distribution:

$$\tilde{q}_l(x_l) \propto \rho_l(x_l | \mathcal{A}_l(x_l)) q_{l-1}(\mathcal{A}_l(x_l)).$$

Using the notations above, a conceptually workable modification of Algorithm 1 can be summarized in Algorithm 2.

Algorithm 2 The Hierarchical Sampling Strategy of the MsIGN

```
**Output**: Sample $x = x_L$ from the target distribution $\nu = \nu_L$
1: Sample $x_1$ from the coarsest-scale distribution $\nu_1$ by a standard method.
2: **for** $l \leftarrow 2$ to $L$ **do**
3:     Sample $\tilde{x}_l$ from the conditional distribution $\rho_l(\tilde{x}_l | \mathcal{A}_l(\tilde{x}_l) = x_{l-1})$.
4:     Learn a transport map $F_l$ that pushes forward $\tilde{\nu}_l$ to $\nu_l$.
5:     Obtain sample $x_l$ from the distribution $\nu_l$ by $x_l = F_l(\tilde{x}_l)$.
6: **end for**
```

Sampling from the Coarsest-Scale Distribution

In the step 1 of Algorithm 2, we learn a transport map F_1 that pushes forward γ_1, the d_1-dimensional standard Gaussian distribution, to ν_1. Because the problem dimension d_1 can be chosen to be very small, many standard methods of transport maps can be applied here, like [80, 48, 57].

Sampling from the Conditional Distribution

In the step 3 of Algorithm 2, the prior conditional distribution (5.13) only depends on the original prior $\mu_L = \mu$ and $\mathcal{A}_{l'}$, $l \le l' \le L$, which are all known in advance to the observation y. Therefore, we can compute a transport map $PC_l : \mathbb{R}^{d_{l-1}} \times \mathbb{R}^{d_l - d_{l-1}} \to \mathbb{R}^{d_l}$ named as prior conditional map, such that a sample $\tilde{x}_l \in \mathbb{R}^{d_l}$ from the prior conditional distribution $\rho_l(\tilde{x}_l | \mathcal{A}_l(\tilde{x}_l) = x_{l-1})$ can be generated by $\tilde{x}_l = PC_l(x_{l-1}, z_l)$, where $z_l \in \mathbb{R}^{d_l - d_{l-1}}$ follows γ_l, the $(d_l - d_{l-1})$-dimensional standard Gaussian distribution. And $x_{l-1} \in \mathbb{R}^{d_{l-1}}$ is the sample from the last scale posterior q_l. Again, we remark that the sample $\tilde{x}_l$ generated in this way will follow the surrogate distribution $\tilde{\nu}_l$.

Since the prior conditional map PC_l only depends on the prior distribution ρ_l and upscaling operator $\mathcal{A}_l$, it can be pre-computed before giving the observation y, and will be fixed afterwards. The existence of the map PC_l is guaranteed by the following theorem.

Theorem 5.3.1. *If there is a map $\mathcal{B}_l : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l - d_{l-1}}$ such that the map $C_l : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ given by*

$$C_l(x) = \begin{bmatrix} \mathcal{A}_l(x) \\ \mathcal{B}_l(x) \end{bmatrix},$$

is a diffeomorphism, then there exists a bijective transport map $S_l : \mathbb{R}^{d_l} \longrightarrow \mathbb{R}^{d_l}$, such that $S_{l\sharp}\mu_l = \mu_{l-1} \otimes \gamma_l$, and $P_{l-1} \circ S_l = \mathcal{A}_l$, where $P_{l-1} : \mathbb{R}^{d_l} \longrightarrow \mathbb{R}^{d_{l-1}}$ is the linear projector to the first d_{l-1} dimension.

Our prior conditional map PC_l can then be taken as $PC_l = S_l^{-1}$. Specifically, $\tilde{x}_l$ and (x_{l-1}, z_{l-1}) have one-to-one correspondence under S_l. Thus

$$\{S^{-1}(x_{l-1}, z_l) | z_l \in \mathbb{R}^{d_l - d_{l-1}}\} = \{\tilde{x}_l | \mathcal{A}_l(\tilde{x}_l) = x_{l-1}\},$$

which means the one-to-one correspondence between $\{(x_{l-1}, z_l) | z_l \in \mathbb{R}^{d_l - d_{l-1}}\}$ and the conditional set $\{\tilde{x}_l | \mathcal{A}_l(\tilde{x}_l) = x_{l-1}\}$ under S_l. Since $z_l \sim \gamma_l$ and

$$S_{l\sharp}^{-1}(\mu_{l-1} \otimes \gamma_l) = \mu_l,$$

$S_l^{-1}(x_{l-1}, z_l)$ should follow the conditional distribution $\rho_l(\tilde{x}_l | \mathcal{A}_l(\tilde{x}_l) = x_{l-1})$.

We give a constructional proof of Theorem 5.3.1 below:

Proof. Consider the new distribution $\hat{\mu}_l = C_{l\sharp}\mu_l$. Since the first d_{l-1} dimensions of C_l is $\mathcal{A}_l$, and $\mu_{l-1} = \mathcal{A}_{l\sharp}\mu_l$ by definition, we know that the marginal distribution of $\hat{\mu}_l$ in the first d_{l-1} dimension is exactly μ_{l-1}. Let $(\hat{\mu}_l)_i$, $(\mu_{l-1} \otimes \gamma_l)_i$ be their marginal distributions in the first i dimensions. We have $(\hat{\mu}_l)_{d_{l-1}} = \mu_{l-1}$.

Now we consider constructing a triangular map $R_l : \mathbb{R}^{d_l} \longrightarrow \mathbb{R}^{d_l}$ that pushes forward $\hat{\mu}_l$ to $\mu_{l-1} \otimes \gamma_l$. Our construction mimics the way to construct the K-R rearrangement [82, 93]. We start by setting $R_{l,d_{l-1}} = \mathrm{id}_{d_{l-1}}$ to be the d_{l-1}-dimensional identity map. We have $R_{l,d_{l-1}\sharp}(\hat{\mu}_l)_{d_{l-1}} = (\mu_{l-1} \otimes \gamma_l)_{d_{l-1}}$ because both sides are μ_{l-1}.

Our construction works recursively. Suppose we have $R_{l,i} : \mathbb{R}^i \longrightarrow \mathbb{R}^i$ that pushes forward $(\hat{\mu}_l)_i$ to $(\mu_{l-1} \otimes \gamma_l)_i$ for $i \geq d_{l-1}$. Since $\hat{\mu}_l$, $\mu_{l-1} \otimes \gamma_l$ are non-atomic, we can find a non-decreasing map $\hat{R}_{l,i+1} : \mathbb{R}^{i+1} \longrightarrow \mathbb{R}$ such that for any fixed $x_1, \ldots, x_i$,

$$\hat{R}_{l,i+1}(x_1, \ldots, x_i, \cdot)_\sharp \left((\hat{\mu}_l)_{i+1}(\mathrm{d}x_{i+1} | x_1, \ldots, x_i)\right) = (\mu_{l-1} \otimes \gamma_l)_{i+1}(\mathrm{d}x_{i+1} | x_1, \ldots, x_i).$$

Specifically, let $F_{i+1}(x_{i+1}; x_1, \ldots, x_i)$ be the cumulative density function of the 1D distribution $(\hat{\mu}_l)_{i+1}(\mathrm{d}x_{i+1} | x_1, \ldots, x_i)$, $G_{i+1}(x_{i+1}; x_1, \ldots, x_i)$ be the cumulative density function of the 1D distribution $(\mu_{l-1} \otimes \gamma_l)_{i+1}(\mathrm{d}x_{i+1} | x_1, \ldots, x_i)$, so they are both monotonically increasing because we assume all distributions here are absolutely continuous to the Lebesgue measure. We could simply set

$$\hat{R}_{l,i+1}(x_1, \ldots, x_i, x_{i+1}) = F_{i+1}(x_{i+1}; x_1, \ldots, x_i)^{-1} G_{i+1}(x_{i+1}; x_1, \ldots, x_i).$$

This construction also ensures that $\hat{R}_{l,i+1}(x_1, \dots, x_i, x_{i+1})$ is monotonically increasing in x_{i+1}. Therefore, setting $R_{l,i+1} : \mathbb{R}^{i+1} \longrightarrow \mathbb{R}^{i+1}$ as

$$R_{l,i+1}(x_1, \dots, x_i, x_{i+1}) = \begin{bmatrix} R_{l,i}(x_1, \dots, x_i) \\ \hat{R}_{l,i+1}(x_1, \dots, x_i, x_{i+1}) \end{bmatrix},$$

we can verify that $R_{l,i+1\sharp}\,(\hat{\mu}_l)_{i+1} = (\mu_{l-1} \otimes \gamma_l)_{i+1}$, as in Section 2.3 of [82].

This recursive construction for $d_{l-1} \leq i < d_l$ finally gives us a lower-triangular bijective map $R_l : \mathbb{R}^{d_l} \longrightarrow \mathbb{R}^{d_l}$ such that $R_{l\sharp}\hat{\mu}_l = \mu_{l-1} \otimes \gamma_l$, and $P_{l-1} \circ R_l = P_{l-1}$, because $R_{l,d_{l-1}} = \mathrm{id}_{d_{l-1}}$.

Finally we consider $S_l = R_l \circ C_l$. On the one hand, we have

$$S_{l\sharp}\mu_l = (R_l \circ C_l)_\sharp\, \mu_l = R_{l\sharp}\left(C_{l\sharp}\mu_l\right) = R_{l\sharp}\hat{\mu}_l = \mu_{l-1} \otimes \gamma_l,$$

and on the other hand, when we apply the linear projector P_{l-1}, we get

$$P_{l-1} \circ S_l = P_{l-1} \circ (R_l \circ C_l) = (P_{l-1} \circ R_l) \circ C_l = P_{l-1} \circ C_l = \mathcal{A}_l.$$

By construction, S_l is a bijective, and from [82], given the regularity of μ_l and μ_{l-1}, the constructed R_l is a diffeomorphism and so is S_l. Finally we remark that the uniqueness of S_l is not necessarily guaranteed. □

It is interesting to notice that in the special case of linear upscaling operator $\mathcal{A}_l(x_l) = A_l x_l$ with $A_l \in \mathbb{R}^{d_{l-1} \times d_l}$, and the Gaussian prior $\mu_l = \mathcal{N}(0, \Sigma_l)$, there is a closed form formula for the map PC_l.

Theorem 5.3.2. *For linear upscaling operator $\mathcal{A}_l(x_l) = A_l x_l$ with a matrix $A_l \in \mathbb{R}^{d_{l-1} \times d_l}$ that has full row rank, and a Gaussian prior $\mu_l = \mathcal{N}(0, \Sigma_l)$ with Σ_l symmetric positive definite, the transport map PC_l can be chosen as*

$$\tilde{x}_l = PC_l(x_{l-1}, z_l) = U_{l-1} x_{l-1} + V_l z_l, \tag{5.20}$$

where $U_{l-1} = \Sigma_l A_l^T (A_l \Sigma_l A_l^T)^{-1}$, $V_l \in \mathbb{R}^{d_l \times (d_l - d_{l-1})}$ is any matrix satisfying

$$V_l V_l^T = \Sigma_l - \Sigma_l A_l^T (A_l \Sigma_l A_l^T)^{-1} A_l \Sigma_l,$$

and the existence of V_l is guaranteed. Furthermore, PC_l defined here is a bijective between $\tilde{x}_l$ and (x_{l-1}, z_l).

Theorem 5.3.2 is very helpful in pre-computing the map PC_l in the popular cases of a Gaussian prior. The transport map PC_l is called the prior conditioning layer in our MsIGN framework.

To prove Theorem 5.3.2, we first notice that since $A_l \in \mathbb{R}^{d_{l-1}\times d_l}$ $(d_{l-1} < d_l)$, we can always find a matrix $B_l \in \mathbb{R}^{(d_l-d_{l-1})\times d_l}$, such that $A_l B_l^T = 0$. We now state the following lemma, called the partition of unity.

Lemma 5.3.1. *Since $A_l \in \mathbb{R}^{d_{l-1}\times d_l}$, $B_l \in \mathbb{R}^{(d_l-d_{l-1})\times d_l}$, $A_l B_l^T = 0$ and the covariance matrix Σ_l is symmetric positive definite, we have the following decomposition of the identity matrix $I_{d_l} \in \mathbb{R}^{d_l\times d_l}$:*

$$I_{d_l} = \Sigma_l^{\frac{1}{2}} A_l^T (A_l \Sigma_l A_l^T)^{-1} A_l \Sigma_l^{\frac{1}{2}} + \Sigma_l^{-\frac{1}{2}} B_l^T (B_l \Sigma_l^{-1} B_l^T)^{-1} B_l \Sigma_l^{-\frac{1}{2}}. \tag{5.21}$$

Proof. Consider the following matrix $\Omega_l \in \mathbb{R}^{d_l\times d_l}$:

$$\Omega_l = [\Sigma_l^{\frac{1}{2}} A_l^T (A_l \Sigma_l A_l^T)^{-\frac{1}{2}} \quad \Sigma_l^{-\frac{1}{2}} B_l^T (B_l \Sigma_l^{-1} B_l^T)^{-\frac{1}{2}}].$$

We claim that Ω_l is an orthonormal matrix because

$$\Omega_l^T \Omega_l = \begin{bmatrix} I_{d_{l-1}} & * \\ *^T & I_{d_l-d_{l-1}} \end{bmatrix} = I_{d_l},$$

as $* = (A_l \Sigma_l A_l^T)^{-\frac{1}{2}} A_l B_l^T (B_l \Sigma_l^{-1} B_l^T)^{-\frac{1}{2}} = 0$ due to the assumption $A_l B_l^T = 0$.

Therefore, Ω_l is a $d_l \times d_l$ orthonormal matrix, and $\Omega_l \Omega_l^T = I_{d_l}$, which means

$$\begin{aligned} I_{d_l} = \Omega_l \Omega_l^T &= \left[\Sigma_l^{\frac{1}{2}} A_l^T (A_l \Sigma_l A_l^T)^{-\frac{1}{2}} \quad \Sigma_l^{-\frac{1}{2}} B_l^T (B_l \Sigma_l^{-1} B_l^T)^{-\frac{1}{2}}\right] \begin{bmatrix} (A_l \Sigma_l A_l^T)^{-\frac{1}{2}} A_l \Sigma_l^{\frac{1}{2}} \\ (B_l \Sigma_l^{-1} B_l^T)^{-\frac{1}{2}} B_l \Sigma_l^{-\frac{1}{2}} \end{bmatrix} \\ &= \Sigma_l^{\frac{1}{2}} A_l^T (A_l \Sigma_l A_l^T)^{-1} A_l \Sigma_l^{\frac{1}{2}} + \Sigma_l^{-\frac{1}{2}} B_l^T (B_l \Sigma_l^{-1} B_l^T)^{-1} B_l \Sigma_l^{-\frac{1}{2}}. \end{aligned}$$

□

Finally, we prove Theorem 5.3.2.

Proof. First we notice that $A_l U_{l-1} = I_{d_{l-1}}$ and $A_l V_l = 0$, so $A_l \tilde{x}_l = x_{l-1}$. Following Theorem 5.3.1 and the remark in Section 5.2, we now only need to prove that $PC_{l\sharp}(\mu_{l-1} \otimes \gamma_l) = \mu_l$ to show that $\tilde{x}_l = PC_l(x_{l-1}, z_l)$ follows the conditional distribution $\rho_l(\tilde{x}_l | \mathcal{A}_l(\tilde{x}_l) = x_{l-1})$ for a fixed x_{l-1} and $z_l \sim \gamma_l$.

Since the map PC_l is linear, and x_{l-1}, z_l are independent Gaussians, both sides of (5.13) are Gaussian distributions. It remains to check that their moments match each other, which translates to

$$\Sigma_l = U_{l-1}A_l\Sigma_l A_l^T U_{l-1}^T + V_l V_l^T. \tag{5.22}$$

Here $A_l\Sigma_l A_l^T$ is the covariance matrix for $x_{l-1} = A_l\tilde{x}_l$.

Recalling the definitions of V_l and U_{l-1} in Theorem 5.3.2, we have $V_lV_l^T = \Sigma_l - \Sigma_l A_l^T(A_l\Sigma_lA_l^T)^{-1}A_l\Sigma_l$ and $U_{l-1}A_l\Sigma_lA_l^TU_{l-1}^T = \Sigma_lA_l^T(A_l\Sigma_lA_l^T)^{-1}A_l\Sigma_l$. So (5.22) holds and the first part is proved. We are left to show the existence of V_l and the invertibility of PC_l.

Let $\Sigma_{l|l-1} = V_lV_l^T = \Sigma_l - \Sigma_lA_l^T(A_l\Sigma_lA_l^T)^{-1}A_l\Sigma_l$. We notice that

$$\Sigma_{l|l-1} = \Sigma_l - \Sigma_l^T(A_l\Sigma_lA_l^T)^{-1}A_l\Sigma_l = \Sigma_l^{\frac{1}{2}}\left(I_{d_l} - \Sigma_l^{\frac{1}{2}}A_l^T(A_l\Sigma_lA_l^T)^{-1}A_l\Sigma_l^{\frac{1}{2}}\right)\Sigma_l^{\frac{1}{2}}.$$

Using the partition of unity in Lemma 5.3.1, we have

$$\Sigma_{l|l-1} = \Sigma_l^{\frac{1}{2}}\Sigma_l^{-\frac{1}{2}}B_l^T(B_l\Sigma_l^{-1}B_l^T)^{-1}B_l\Sigma_l^{-\frac{1}{2}}\Sigma_l^{\frac{1}{2}} = B_l^T(B_l\Sigma_l^{-1}B_l^T)^{-1}B_l.$$

Therefore, the existence of $V_l \in \mathbb{R}^{d_l\times(d_l-d_{l-1})}$ such that $\Sigma_{l|l-1} = V_lV_l^T$ is guaranteed, because taking any orthonormal matrix $P_l \in \mathbb{R}^{(d_l-d_{l-1})\times(d_l-d_{l-1})}$, the construction $V_l = B_l^T(B_l\Sigma_l^{-1}B_l^T)^{-\frac{1}{2}}P_l$ satisfies the requirement.

To show the invertibility of PC_l, let $\tilde{x}_l = PC_l(x_{l-1}, z_l) = U_{l-1}x_{x-l} + V_lz_l$. We claim that the inversion can be given by

$$x_{l-1} = A_l\tilde{x}_l, \quad \text{and} \quad z_l = P_l^T(B_l\Sigma_l^{-1}B_l^T)^{-\frac{1}{2}}B_l\Sigma_l^{-1}\tilde{x}_l,$$

where P_l is the orthonormal matrix such that $V_l = B_l^T(B_l\Sigma_l^{-1}B_l^T)^{-\frac{1}{2}}P_l$. To see this, we compute, under the above claim,

$$\begin{aligned} U_{l-1}x_{l-1} + V_lz_l &= U_{l-1}A_l\tilde{x}_l + V_lP_l^T(B_l\Sigma_l^{-1}B_l^T)^{-\frac{1}{2}}B_l\Sigma_l^{-1}\tilde{x}_l \\ &= (U_{l-1}A_l + V_lP_l^T(B_l\Sigma_l^{-1}B_l^T)^{-\frac{1}{2}}B_l\Sigma_l^{-1})\tilde{x}_l. \end{aligned}$$

We will show $U_{l-1}A_l + V_lP_l^T(B_l\Sigma_l^{-1}B_l^T)^{-\frac{1}{2}}B_l\Sigma_l^{-1} = I_{d_l}$ to complete the proof.

Since $U_{l-1} = \Sigma_lA_l^T(A_l\Sigma_lA_l^T)^{-1}$ and $V_l = B_l^T(B_l\Sigma_l^{-1}B_l^T)^{-\frac{1}{2}}P_l$, we have $U_{l-1}A_l = \Sigma_lA_l^T(A_l\Sigma_lA_l^T)^{-1}A_l$ and $V_lP_l^T(B_l\Sigma_l^{-1}B_l^T)^{-\frac{1}{2}}B_l\Sigma_l^{-1} = B_l^T(B_l\Sigma_l^{-1}B_l^T)^{-1}B_l\Sigma_l^{-1}$. So we proceed to obtain

$$
\begin{aligned}
&U_{l-1}A_l + V_l P_l^T (B_l \Sigma_l^{-1} B_l^T)^{-\frac{1}{2}} B_l \Sigma_l^{-1} \\
=&\Sigma_l A_l^T (A_l \Sigma_l A_l^T)^{-1} A_l + B_l^T (B_l \Sigma_l^{-1} B_l^T)^{-1} B_l \Sigma_l^{-1} \\
=&\Sigma_l^{\frac{1}{2}} \left(\Sigma_l^{\frac{1}{2}} A_l^T (A_l \Sigma_l A_l^T)^{-1} A_l \Sigma_l^{\frac{1}{2}} + \Sigma_l^{-\frac{1}{2}} B_l^T (B_l \Sigma_l^{-1} B_l^T)^{-1} B_l \Sigma_l^{-\frac{1}{2}} \right) \Sigma_l^{-\frac{1}{2}} \\
=&\Sigma_l^{\frac{1}{2}} I_{d_l} \Sigma_l^{-\frac{1}{2}} = I_{d_l}.
\end{aligned}
$$

Here in the last line we invoked the partition of unity in Lemma 5.3.1. □

Sampling from the Transport Map

In the step 4 of Algorithm 2, due to the resemblance between $\tilde{\nu}_l$ and ν_l as shown by Theorem 5.2.1, the transport map F_l that modifies $\tilde{x}_l \sim \tilde{\nu}_l$ to $x_l \sim \nu_l$ can be seen as a perturbation of the identity map. Therefore, we stack multiple invertible blocks of the Glow [54] introduced in Section 5.1 as the invertible flow F_l, and initialize it to be an identity map in $\mathbb{R}^{d_l}$, which is quite different from [54]. Specifically, for every invertible block in F_l, the parameters s and b in the actnorm unit will be initialized as the all-one vector and all-zero vector respectively. In the invertible 1×1 convolution unit, the matrix W is initialized as the identity matrix. And in the affine coupling unit, we initialize the parameterized map f and g such that f returns a constant all-one tensor, and g returns a constant all-zero tensor.

Overall Architecture

We conclude the network architecture for the pipeline described in Algorithm 2 in Figure 5.1. The network architecture is also formally written in Table 5.1. We recall that for $1 \leq l \leq L$, γ_l is the $(d_l - d_{l-1})$-dimensional standard Gaussian distribution, assuming $d_0 = 0$, and z_l is a sample to γ_l.

scale	distribution perspective	sample perspective
$l = 1$	$F_{1\sharp}\gamma_1 = \nu_1$	$x_1 = F_1(z_1)$
$2 \leq l \leq L$	$PC_{l\sharp}(\gamma_l \otimes \nu_{l-1}) = \tilde{\nu}_l$ $F_{l\sharp}\tilde{\nu}_l = \nu_l$	$\tilde{x}_l = PC_l(x_{l-1}, z_l)$ $x_l = F_l(\tilde{x}_l)$

Table 5.1: The multiscale strategy of the MsIGN to approximate $\nu = \nu_L$ and generate a sample x_L from it.

Let $T_l = F_l \circ PC_l$ for $l \geq 2$ and $T_1 = F_1$ be the transport map at scale $l = 1$. The hierarchical sampling strategy in Algorithm 2 can now be formulated as learning T_l

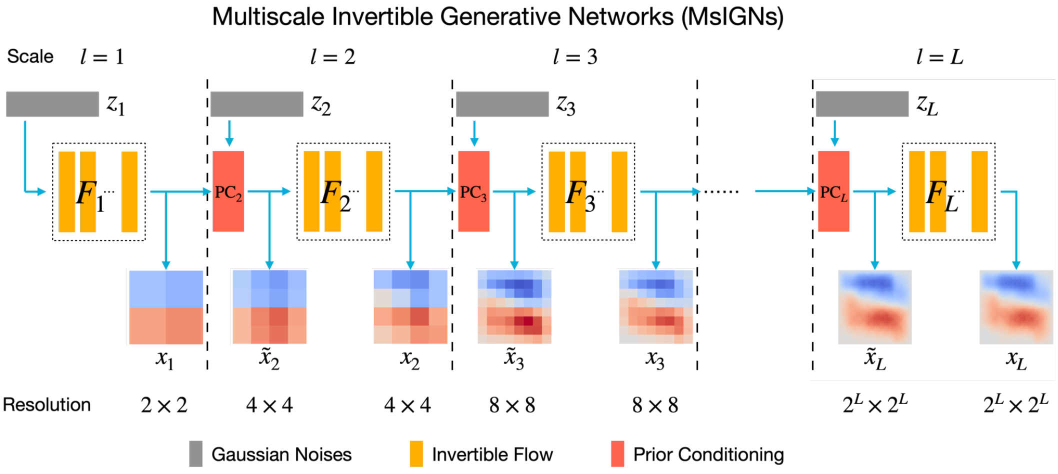


Figure 5.1: A diagram of the network architecture of the MsIGN.

such that

$$T_{1\sharp}\gamma_1 = \nu_1, \quad T_{l\sharp}\left(\gamma_l \otimes \nu_{l-1}\right) = \nu_l, \quad 2 \leq l \leq L. \tag{5.23}$$

The overall transport map represented by the MsIGN can then be written as $T = \tilde{T}_L \circ \cdots \circ \tilde{T}_1$, where $\tilde{T}_l : \mathbb{R}^d \to \mathbb{R}^d$ is the extension of $T_l : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ to the full dimensions by the direct product with the identity map for the dimensions not considered by T_l. Setting $\gamma = \gamma_L \otimes \cdots \otimes \gamma_2 \otimes \gamma_1$, we have $T_\sharp \gamma = \nu_L$.

5.4 Training Strategy of Multiscale Invertible Generative Networks

The MsIGN adopts the variational inference approach to learn the network parameters. To avoid abuse of notations, we omit the scale indicator (subscript l in Section 5.3) when there is no ambiguity, since the whole pipeline in Algorithm 2 is processed scale by scale. We use q as the density function of the target distribution, which can be any ν_l for $1 \leq l \leq L$. And similarly we use p_θ as the density function of the working distribution, which is $F_{1\sharp}\gamma_1$ when $l = 1$, or $F_{l\sharp}\tilde{\nu}_{l-1}$ when $l \geq 2$. Here θ denotes the network parameter of the invertible flow F_l, and belongs to a proper set Θ. Variational inference learns the parameter θ by solving the optimization

$$\min_{\theta\in\Theta} D\left(p_\theta, q\right) \tag{5.24}$$

for some hand-picked discrepancy D.

We remark that the prior conditional layer PC does not need training. With the existence guaranteed by Theorem 5.3.1, we can learn PC offline as it only depends on our choice of the prior μ and upscaling operator $\mathcal{A}$. Particularly, for linear upscaling operator $\mathscr{A}$ and Gaussian prior μ, it has a closed form according to Theorem 5.3.2.

Choice of Learning Objective

Multiple choices of the discrepancy D are available here for the training of the MsIGN, since the invertibility allows the density evaluation of p_θ by (5.2). In the literature of Bayesian inverse problems, the Kullback-Leibler (KL) divergence is easy to compute and has been widely used as the learning objective in variational inference. However, its landscape could admit local minima that don't favor the optimization. The authors in [76] suggest that minimizing the KL divergence $D_{\mathrm{KL}}(p_\theta \| q) = \mathbb{E}_{x\sim p_\theta}\left[\log\frac{p_\theta(x)}{q(x)}\right]$ is zero-forcing, meaning that it mostly enforces that p_θ is small whenever q is small, because when p_θ is large and q is small, both the density p_θ and the weight $\log\frac{p_\theta(x)}{q(x)}$ are significant, resulting in a large objective value. But when p_θ is small and q is large, $D_{\mathrm{KL}}(p_\theta\|q)$ can still be small, as we have little weight p_θ in this area. As a consequence, $D_{\mathrm{KL}}(p_\theta\|q)$ primarily penalizes p_θ in the less important region of q. However, the case of mode missing, where p_θ is small but q is large, can still be a local minimum. Therefore, we turn to the Jeffreys divergence [50] which is a symmetrization of the KL divergence:

$$\begin{aligned} D_{\mathrm{J}}(p_\theta\|q) &= D_{\mathrm{KL}}(p_\theta\|q) + D_{\mathrm{KL}}(q\|p_\theta) \\ &= \mathbb{E}_{x\sim p_\theta}\left[\log\frac{p_\theta(x)}{q(x)}\right] + \mathbb{E}_{x\sim q}\left[\log\frac{q(x)}{p_\theta(x)}\right]. \end{aligned} \tag{5.25}$$

We use a toy example of a 1D Gaussian mixture model to illustrate this observation. Given $\sigma > 0$, let q be the density of a Gaussian mixture model with parameter $\mu = (\mu_1, \mu_2)$ unknown but fixed:

$$q(x) = \frac{1}{2}\left(\mathcal{N}(x;\mu_1,\sigma^2) + \mathcal{N}(x;\mu_2,\sigma^2)\right). \tag{5.26}$$

Here $\mathcal{N}(x;\mu,\sigma^2)$ is the density function of a Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$. Our working distribution admits a density p that is also a 1D Gaussian mixture model with parameter $\theta = (\theta_1,\theta_2)$ to be determined:

$$p_\theta(x) = \frac{1}{2}\left(\mathcal{N}(x;\theta_1,\sigma^2) + \mathcal{N}(x;\theta_2,\sigma^2)\right). \tag{5.27}$$

Setting $\mu_1 = -\mu_2 = 1.5$, and $\sigma = 0.25$, we plot the landscapes of the KL divergences $D_{\mathrm{KL}}(p_\theta\|q)$ and $D_{\mathrm{KL}}(q\|p_\theta)$, and the Jeffreys divergence $D_{\mathrm{J}}(p_\theta\|q)$ as functions of θ in Figure 5.2. We mark the global minima (ground-truth) by golden crosses, and other local minima by green crosses. Notice the difference of scale as shown by the color bar. We can see that $D_{\mathrm{KL}}(p_\theta \| q)$ admits two undesired local minima as compared to $D_{\mathrm{J}}(p_\theta \| q)$. This suggests that using the KL divergence $D_{\mathrm{KL}}(p_\theta \| q)$ as the learning objective here can lead to a mode collapse, while the Jeffreys divergence $D_{\mathrm{J}}(p_\theta \| q)$ can capture both modes correctly.

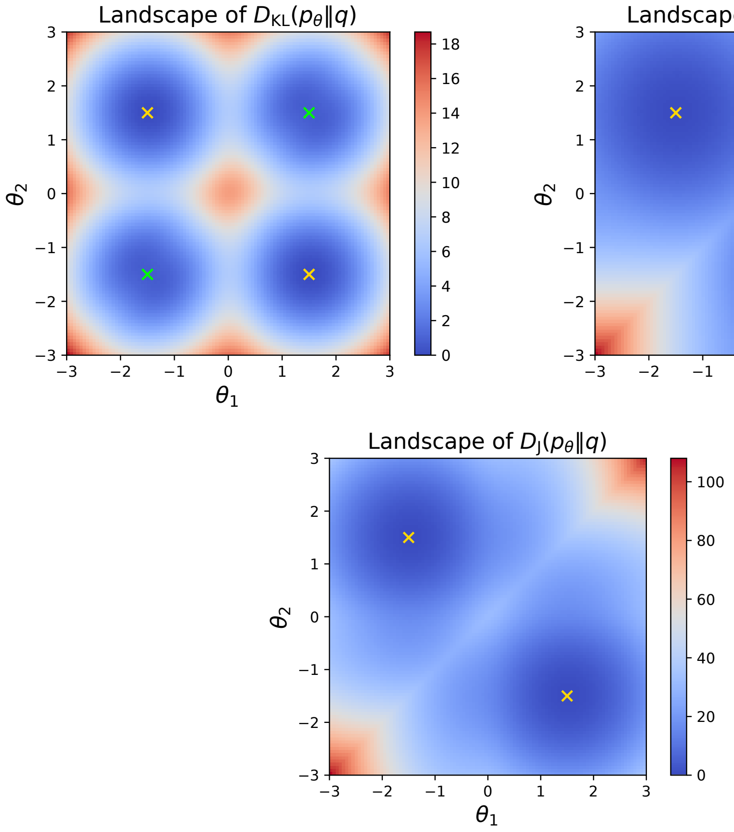


Figure 5.2: Landscapes of the KL divergences and Jeffreys divergence between p_θ and q.

We also mention here that the other single-sided KL divergence $D_{\mathrm{KL}}(q \| p_\theta)$ alone could lead to undesired local minima. Similar to what we discussed above, if p_θ captures all modes in q but also contains some extra modes, described as "zero-avoiding" in [76], we could also observe a small value of $D_{\mathrm{KL}}(q \| p_\theta) = \mathbb{E}_{x \sim q}\left[\log \frac{q(x)}{p_\theta(x)}\right]$, which can be a potential local minimum of the objective landscape. Therefore, we choose the Jeffreys divergence as a robust learning objective to capture multi-modes.

Multi-Stage Optimization

Estimating the Jeffreys divergence in (5.9) requires computing the expectation $\mathbb{E}_{x \sim q}\left[\log \frac{q(x)}{p_\theta(x)}\right]$, where the distribution is the target q. Thus it is usually prohibitively expensive. Since the MsIGN constructs a good approximation $\tilde{q}$ to q, as in (5.8), we do importance sampling for the q-expectation part in the Jeffreys divergence. Therefore, we can use the Monte Carlo method to estimate

$$D_{\mathrm{J}}(p_\theta \| q) = \mathbb{E}_{x\sim p_\theta}\left[\log \frac{p_\theta(x)}{q(x)}\right] + \mathbb{E}_{x\sim\tilde{q}}\left[\frac{q(x)}{\tilde{q}(x)} \log \frac{q(x)}{p_\theta(x)}\right]. \tag{5.28}$$

Moreover, we can estimate the derivatives of the Jeffreys divergence to parameters θ by

$$\frac{\partial}{\partial\theta} D_{\mathrm{J}}(p_\theta \| q) = \mathbb{E}_{x\sim p_\theta}\left[\left(1 + \log \frac{p_\theta(x)}{q(x)}\right) \frac{\partial \log p_\theta(x)}{\partial\theta}\right] - \mathbb{E}_{x\sim\tilde{q}}\left[\frac{q(x)}{\tilde{q}(x)} \frac{\partial \log p_\theta(x)}{\partial\theta}\right]. \tag{5.29}$$

Here in (5.29) we use the identity $\mathbb{E}_{x\sim p_\theta}\left[\partial_\theta \log p_\theta(x)\right] = 0$ as $\lim_{\|x\|\to+\infty} p_\theta(x) = 0$. We remark that due to the limited number d_y of observations in (1.1), q and $\tilde{q}$ are both equivalent to ρ. As a result, q is equivalent to $\tilde{q}$ and the validity of the importance sampling is guaranteed. We also remark that the normalizing constant Z of q in (1.6) is usually unknown, so we can only evaluate $Zq(x)$ as a whole. Fortunately, (5.28) and (5.29) are invariant to such multiplicative constant since we have

$$\begin{aligned} D_{\mathrm{J}}(p_\theta \| q) &= \mathbb{E}_{x\sim p_\theta}\left[\log \frac{p_\theta(x)}{q(x)}\right] + \mathbb{E}_{x\sim\tilde{q}}\left[\frac{q(x)}{\tilde{q}(x)} \log \frac{q(x)}{p_\theta(x)}\right] \\ &= \mathbb{E}_{x\sim p_\theta}\left[\log \frac{p_\theta(x)}{Zq(x)}\right] + \mathbb{E}_{x\sim\tilde{q}}\left[\frac{q(x)}{\tilde{q}(x)} \log \frac{Zq(x)}{p_\theta(x)}\right]. \end{aligned} \tag{5.30}$$

Here we do not need to worry about the multiplicative constant in the importance weight $q(x)/\tilde{q}(x)$ since it can be eliminated by importance sampling with self-normalization weights. Similarly, (5.29) is also invariant to the normalizing constant Z.

Finally we solve the optimization (5.24) by stochastic gradient descent. Optimization strategies for $D_{\mathrm{J}}(p_\theta \| q)$ are summarized in Algorithm 3.

As shown by Algorithm 2, the multiscale strategy in the sample generation process of the MsIGN enables a coarse-to-fine multi-stage training, which also benefits our importance sampling strategy of the Jeffreys divergence. At stage l of Algorithm 2, we target at capturing q_l, and only train invertible flows before or at this scale: $F_{l'}, l' \leq l$. (5.8) implies that q_l can be well approximated by the surrogate $\tilde{q}_l$, which is the conditional upsampling from q_{l-1} as in (5.15). So we use $\tilde{q}_l$ to initialize our model by setting $F_{l'}, l' < l$ as the trained model at stage $l-1$ and setting F_l as the identity map. To initialize the multi-stage training at scale $l = 1$, the

Algorithm 3 Optimization of the Jeffreys divergence

```
**Input**: Unnormalized density $q$ of $\nu$, density $\tilde{q}$ of $\tilde{\nu}$, sample size $N$, learning rate $\eta$, initializer $\theta_0$, number of iterations $M$
**Output**: Optimizer $\theta$ for (5.24) with $D$ being the Jeffreys divergence
for $t \leftarrow 0$ to $M-1$ do
    Sample $\{x_i^*\}_{i=1}^N$ i.i.d. from $p_{\theta_t}$, sample $\{x'_j\}_{j=1}^N$ i.i.d. from $\tilde{q}$
    Evaluate $p_i^* = p_{\theta_t}(x_i^*)$, $q_i^* = q(x_i^*)$, $p'_j = p_{\theta_t}(x'_j)$, $q'_j = q(x'_j)$, and $\tilde{q}'_j = \tilde{q}(x'_j)$ for $i, j = 1, \dots, N$
    Compute the self-normalized importance weight $w'_j = \hat{w}'_j / W'$, for $j = 1, \dots, N$, where $\hat{w}'_j = q'_j / \tilde{q}'_j$ and $W' = \sum_{j=1}^N \hat{w}'_j$
    Obtain $g_i^* = \partial \log p_{\theta_t}(x_i^*) / \partial \theta_t$ and $g'_j = \partial \log p_{\theta_t}(x'_j) / \partial \theta_t$ by the back propagation of the MsIGN for $i, j = 1, \dots, N$
    Estimate the gradient $G_t$ of the Jeffreys divergence (5.29) by
        $$G_t = \frac{1}{N} \sum_{i=1}^N \left(1 + \log p_i^* - \log q_i^*\right) g_i^* - \frac{1}{N} \sum_{j=1}^N w'_j g'_j$$
    Update the parameter by $\theta_{t+1} = \theta_t - \eta G_t$
end for
```

Jeffreys divergence $D_{\mathrm{J}}(p_\theta \| q)$ is directly estimated by the Monte Carlo method with samples from distribution p_θ and q. The p_θ samples come from the model itself and the q samples come from a pretrained MCMC chain, or other standard methods. We remark that at $l = 1$, the problem dimension is very low (in our example the dimension $d_1 = 4$), so it should be easy for standard methods to capture q_1. Numerical experiments demonstrate that such multi-stage strategy significantly stabilizes the training process and improves the performance. The overall training strategy at one scale is summarized in Algorithm 4.

5.5 Numerical Experiment on the Bayesian Inverse Problem

In this section, we present numerical experiments on two high-dimensional Bayesian inverse problems. To make the high-dimensional inference more challenging, we design the target distributions to have at least two equally important modes. In one of the problems called the synthetic Bayesian inverse problem, true samples to the target distribution are available. In another problem named the elliptic Bayesian inverse problem, true samples are not available but the problem is close to real-world applications in the subsurface flow study. We also report the ablation study of the MsIGN in Section 5.6 on these two Bayesian inverse problems, where we compare the performance of different network architectures and training methods to

Algorithm 4 Training Process of the MsIGN at Stage l

```
Input: Well trained $F_{l'}, l' < l$ and pre-computed $PC_{l'}, l' \leq l$
Output: Well trained $F_l$
if $l = 1$ then
    Sample from the coarsest-scale $q_1$ by standard methods
    Learn $F_1$ by solving $\min_{\theta \in \Theta} D_{\mathrm{J}}(p_\theta \| q_1)$ by stochastic gradient descent, the gradient is estimated by the Monte Carlo method
else
    Initialize $F_l$ as an identity map to model the working distribution $p_\theta$, together with layers trained from last stage: $F_{l'}, l' < l$, and $PC_{l'}, , l' \leq l$
    Duplicate $F_{l'}, l' < l$ and $PC_{l'}, l' \leq l$ to model the surrogate $\tilde{q}_l$
    Learn $F_l$ by solving $\min_{\theta \in \Theta} D_{\mathrm{J}}(p_\theta \| q_l)$ by stochastic gradient descent as in Algorithm 3 with the help of the surrogate $\tilde{q}_l$
end if
```

demonstrate the effectiveness of our proposed strategy.

General Settings

In our numerical examples of two high-dimensional Bayesian inverse problems, the target posterior ν is a distribution of discretized 2D field on the unit square $\Omega = [0, 1]^2$ which can be seen as a vector of $64 \times 64 = 4096$ dimension. For both examples, we place a centered Gaussian with a Laplacian-type covariance as the prior:

$$\mu = \mathcal{N}\left(0, \beta^2(-\Delta)^{-1-\alpha}\right), \tag{5.31}$$

which is very common in geophysics and electric tomography. Here the covariance operator admits zero Dirichlet boundary condition. The likelihoods will be specified individually for each problem. As illustrated by Algorithm 2, we plan to learn the 4096-dimensional posterior $\nu = \nu_L$ at the end of $L = 6$ scales, and set problem dimension at each scale as $d_l = 2^l \times 2^l = 4^l$. Since we are interested in the inference of a 2D field, it is natural to set the pooling operator $\mathcal{A}$ as the local average operator. Specifically, on the discretized field, $\mathcal{A}$ gives the local average in every non-overlapping 2×2 patch, and reduces the dimension by 4.

We equip our target distribution ν with multimodality by designing a symmetric distribution with carefully-chosen parameters. Combining properties of the prior μ defined above and the likelihood $\mathcal{L}$ defined afterwards, the posterior is designed to be mirror-symmetric:

$$q(x) = q(x'), \quad \text{if } x(s_1, s_2) = x'(s_1, 1 - s_2) \text{ for every } (s_1, s_2) \in \Omega. \tag{5.32}$$

We carefully select the prior and the likelihood so that our posterior q has at least two modes, which will be demonstrated for both problems. The two modes are mirror-symmetric to each other and possess equal importance.

We benchmark our MsIGN from some state-of-the-art approaches in the literature: the Hamilton Monte Carlo (HMC) [75], the Stein variational gradient descent (SVGD) [69], the projected Stein variational gradient descent (pSVGD) [13], and the amortized Stein variational gradient descent (A-SVGD) [31]. The HMC is a typical MCMC approach that is empirically successful in high-dimensional problems. The SVGD is a recent particle-based method that moves particles along the gradient of the Kullback-Leibler divergence with respect to the target distribution. And the pSVGD further applies adaptive dimensional reduction in the process of the SVGD. The A-SVGD is another deep generative network approach that trains the network by the gradient signal in the SVGD. We use the Glow model [54] for the network architecture in the A-SVGD, which is proven to be very successful in other sample generation tasks, like image synthesis. The number of parameters of the network for the A-SVGD is guaranteed to be comparable to that for the MsIGN, so that the A-SVGD can serve as a fair benchmark in deep generative network approaches.

We measure the computational cost by the number of forward simulations, because simulating the forward map $\mathcal{F}$ in (1.1) contributes to most of the training time, especially for the elliptic Bayesian inverse problem (more than 75% of the wall clock time). For each method, we budget the same number of forward simulations to generate the same number of target samples for fair comparison.

More parameter settings of network architecture and training strategy can be found in Appendix B.

The Synthetic Bayesian Inverse Problem

The synthetic Bayesian inverse problem allows access to ground-truth samples of the target distribution ν so the comparison is clear and solid. The prior μ is set as (5.31) with $\alpha = 0.1, \beta = 2.0$. As for the likelihood, the forward map is given by

$$\mathcal{F}(x) = \langle \varphi, x \rangle^2 = \left(\int_\Omega \varphi(s) x(s) \mathrm{d}s \right)^2 ,$$

with $\varphi(s) = \sin(\pi s_1)\sin(2\pi s_2)$ for $s = (s_1, s_2) \in \Omega$. We remark that due to the symmetry $\varphi(s_1, s_2) = \varphi(1 - s_1, s_2)$, we have $\mathcal{F}(x) = \mathcal{F}(x')$ if $x(s_1, s_2) = x'(s_1, 1 - s_2)$ for $(s_1, s_2) \in \Omega$. Therefore, the likelihood is guaranteed to be mirror symmetric. The ground-truth for x is $x(s_1, s_2) = \sin(\pi s_1)\sin(2\pi s_2)$, see Figure 5.3.

We generate the observed data y by (1.1) with $\varepsilon \sim \mathcal{N}(0, 0.04)$. The computation budget in the number of forward simulations is fixed at 8×10^6 for generating 2500 samples in every computation.

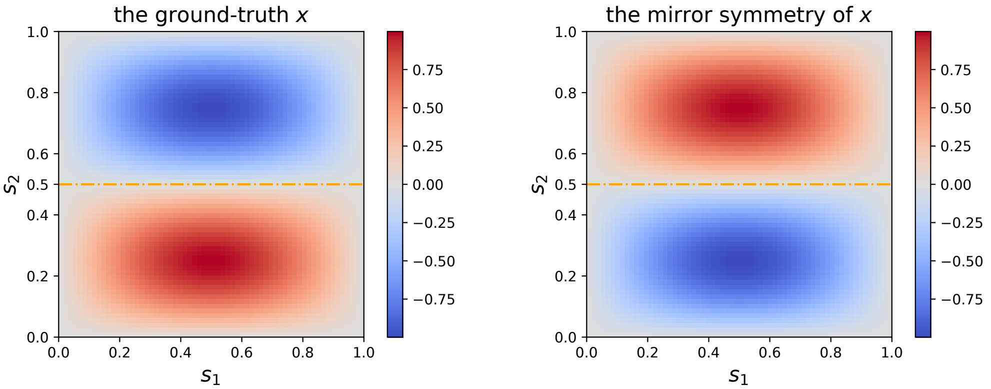


Figure 5.3: The ground-truth x and its mirror symmetry. The dashed line is the symmetry axis.

With the prior and likelihood designed above, our posterior at all scales can be factorized into one-dimensional sub-distributions, namely

$$q_l(x) = \prod_{k=1}^{d_l} q_{l,k}(\langle w_{l,k}, x \rangle), \qquad \text{for } 1 \leq l \leq 6,$$

for some orthonormal basis $\{w_{l,k}\}_{k=1}^{d_l}$. In fact, $w_{l,k}$ can be taken as the first few 2D Fourier basis functions, because they are the eigenvectors of the covariance matrix of the prior (5.31), and the measurement function φ is one of their members. This property gives us access to true samples via the inversion cumulative function sampling along each direction w_k. In fact, all these 1D sub-distributions are unimodal Gaussians except that there is one with two symmetric modes. This double-modal sub-distribution is the marginal distribution $q_{l,*}$ along direction $w_{l,*} = \varphi$. This confirms our construction of two equally important modes.

For each method, we estimate a quantity of interest Q using the Monte Carlo method with generated samples. We run each method for multiple times, and obtain estimates Q_k, $k = 1, \ldots, K$. To assess the distribution approximation, we consider the root mean square error of the estimation, which is defined as

$$\sqrt{\frac{1}{K} \sum_{k=1}^{K} |Q - Q_k|^2}.$$

We may generalize this definition for vector or tensor Q, with now the absolute value $|\cdot|$ replaced by the Frobenius norm. We report the root mean square errors of each method for the following quantity of interest in Figure 5.4: the sub-distributional mean, which is a vector in $\mathbb{R}^{d_l}$ with entries $\mathbb{E}\left[\langle w_{l,k}, x\rangle\right]$ for $k = 1, \ldots, d_l$ at scale l, the sub-distributional standard deviation, which is a vector in $\mathbb{R}^{d_l}$ with entries $\mathrm{Sd}\left[\langle w_{l,k}, x\rangle\right]$ for $k = 1, \ldots, d_l$ at scale l, and the sub-distributional correlation, which is a $d_l \times d_l$ matrix with entries $\mathrm{Corr}\left[\langle w_{l,k}, x\rangle, \langle w_{l,k'}, x\rangle\right]$ for $k, k' = 1, \ldots, d_l$ at scale l. To better compare between different scales, these root mean square errors are divided by their dimensions. The bar range in Figure 5.4 indicates the variation of the error estimates in 5 independent runs.

We observe that the MsIGN is more accurate among other methods in distribution approximation, especially at finer scale when the problem dimension is high. We also remark that the pSVGD achieves a very good result because the target distribution is intrinsically low-rank. In Table 5.2, we report the Jeffreys divergence between the target distribution q and the distributions p_θ captured by the A-SVGD and our MsIGN, since they both allow density evaluation of p_θ. We can see that the MsIGN has superior accuracy in distribution approximation, especially in high-dimensional problems.

Scale	Dimension	MsIGN	A-SVGD
$l = 1$	$d_l = 2^2 = 4$	$(3.00 \pm 0.07) \times 10^{-1}$	$(8.59 \pm 5.45) \times 10^{-1}$
$l = 2$	$d_l = 4^2 = 16$	$(4.85 \pm 0.28) \times 10^{-1}$	$(2.87 \pm 0.21) \times 10^{+0}$
$l = 3$	$d_l = 8^2 = 64$	$(8.49 \pm 0.87) \times 10^{-1}$	$(9.21 \pm 0.29) \times 10^{+0}$
$l = 4$	$d_l = 16^2 = 256$	$(2.74 \pm 0.13) \times 10^{+0}$	$(3.90 \pm 0.67) \times 10^{+1}$
$l = 5$	$d_l = 32^2 = 1024$	$(1.34 \pm 0.04) \times 10^{+1}$	$(4.26 \pm 0.49) \times 10^{+2}$
$l = 6$	$d_l = 64^2 = 4096$	$(5.89 \pm 0.17) \times 10^{+1}$	$(3.62 \pm 0.36) \times 10^{+3}$

Table 5.2: Distribution approximation errors by Jeffreys divergence. The values in the parenthesis indicate the fluctuation in 5 independent runs.

To visualize the mode capture, we plot the marginal distributions along the critical direction $w_{l,*}$, from which we expect to observe double-modality by our construction. The marginal distribution is reconstructed by the kernel density estimation from 2500 generated samples for each method. As our MsIGN works by sequentially capturing q_l for l from 1 to 6, we show the mode capture results at each scale in Figure 5.5. We can see that as the dimension increases, the A-SVGD and the SVGD become less robust in mode capture and eventually collapse to a single mode. Moreover, the HMC becomes imbalanced between modes, and the marginal distribution of the

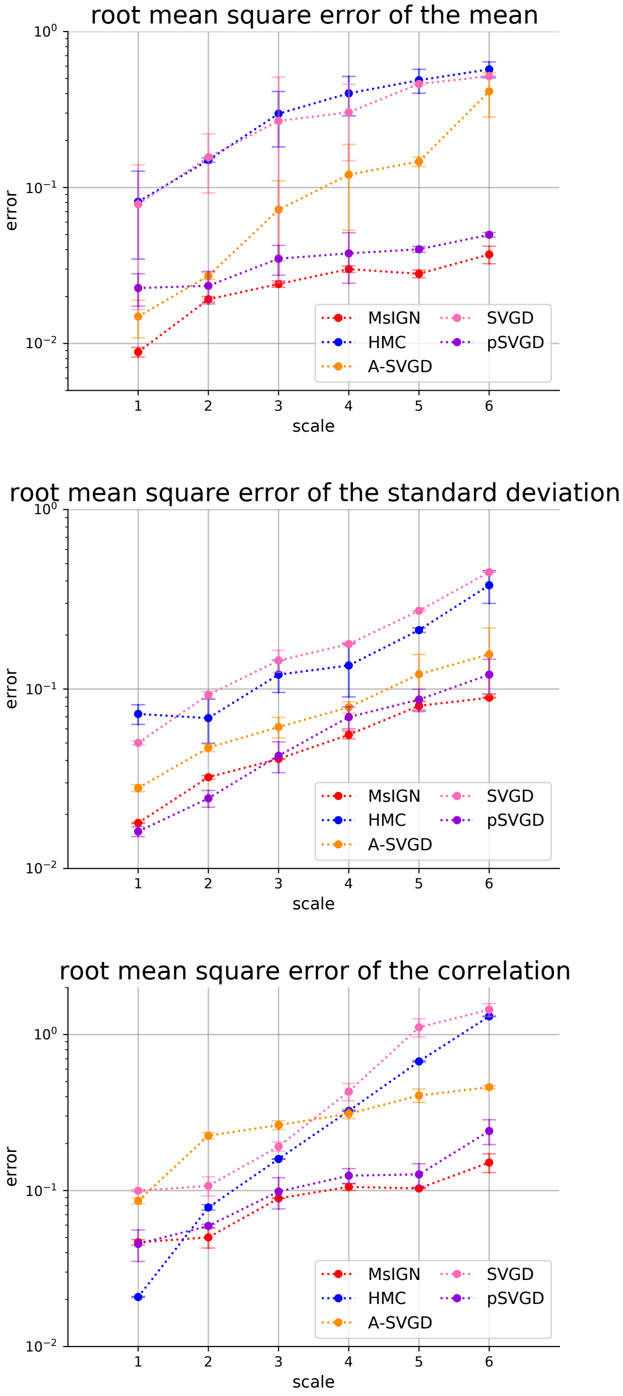


Figure 5.4: Root mean square errors of sub-distributional statistics at different scales $1 \leq l \leq 6$.

pSVGD is a bit biased for $l = 6$. In contrast, our MsIGN successfully captures these two modes and its marginal distribution is the best among these methods when compared with the ground-truth.

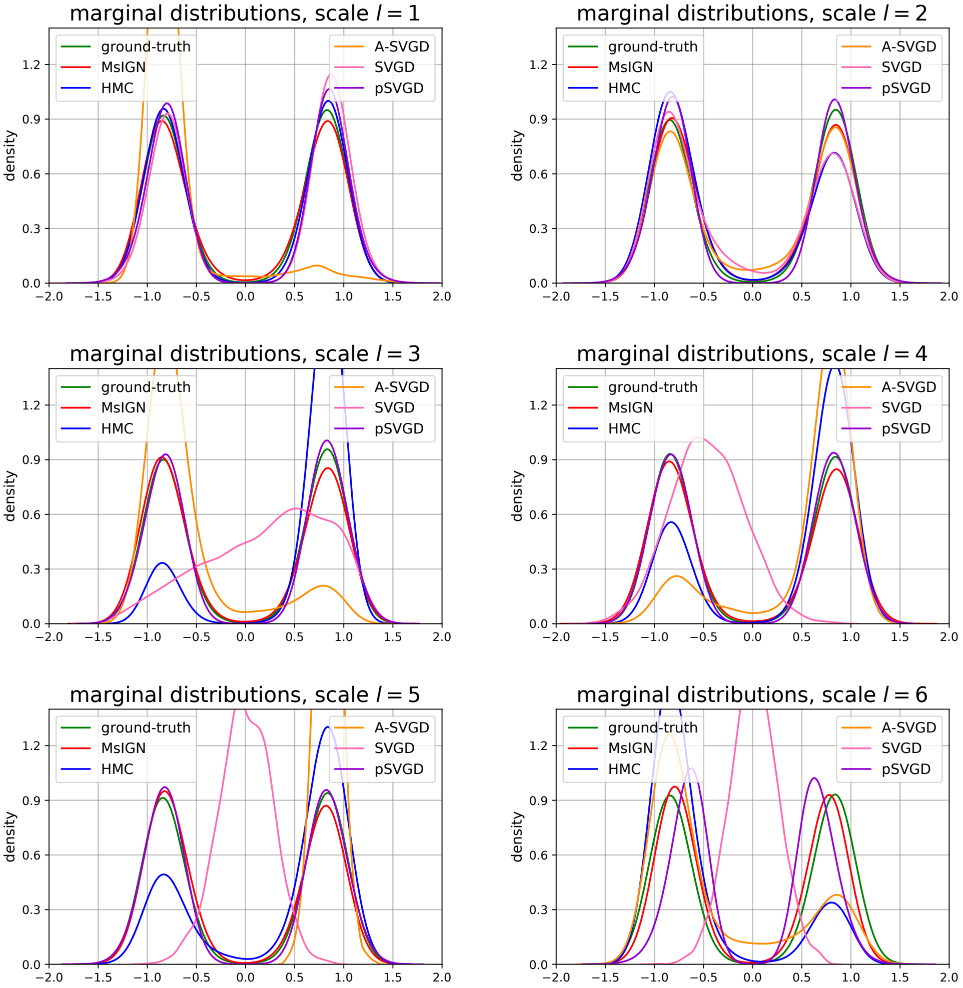


Figure 5.5: Comparison of the marginal distributions along the critical direction $w_{l,*}$ for the synthetic Bayesian inverse problem at all scales $l = 1, \dots, 6$.

We remark that since our MsIGN is a transport map approach, the generated samples are naturally independent. The sample correlation trouble that could potentially occur to the MCMC-type or the SVGD-related methods does not appear in our approach.

The Elliptic Bayesian Inverse Problem

The elliptic Bayesian inverse problem originates from geophysics and fluid dynamics. It is known to be very challenging due to its high-dimensionality and its complicated forward map $\mathcal{F}$. The prior μ is set as (5.31) with $\alpha = 0.5$, $\beta = 2.0$. The forward map is given by linear measurement of the solution to an elliptic partial differential equation (PDE) associated with x. We define

$$\mathcal{F}(x) = \Big[\int_\Omega \varphi_1(s)u(s)\mathrm{d}s \quad \int_\Omega \varphi_2(s)u(s)\mathrm{d}s \quad \dots \quad \int_\Omega \varphi_{15}(s)u(s)\mathrm{d}s\Big]^T \in \mathbb{R}^{15}, \tag{5.33}$$

where $\varphi_k(s), 1 \leq k \leq 15$, are measurement functions, and $u(s)$ is the solution of the following elliptic PDE with zero Dirichlet boundary condition

$$-\nabla \cdot \left(e^{x(s)}\nabla u(s)\right) = f(s), \qquad s \in \Omega. \tag{5.34}$$

The measurement functions φ_k are designed to be the characteristic functions of certain regions. As shown in Figure 5.6, for $1 \leq k \leq 10$, φ_k is the characteristic function of two red squares that are mirror-symmetric to each other. For $11 \leq k \leq 15$, φ_k is the characteristic function of one red square that is mirror-symmetric to itself. The force term f, also shown in Figure 5.6, is chosen as

$$f(s) = \frac{100}{\pi}e^{-10\|s-f_1\|^2} + \frac{100}{\pi}e^{-10\|s-f_2\|^2} - \frac{50}{\pi}e^{-10\|s-f_3\|^2} - \frac{50}{\pi}e^{-10\|s-f_4\|^2},$$

where $f_1 = (0.25, 0.3)$, $f_2 = (0.25, 0.7)$, $f_3 = (0.7, 0.3)$, $f_4 = (0.7, 0.3)$, and $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^2$. The force term is also mirror symmetric along the s_2 direction. We assume the error ε in (1.1) follows $\mathcal{N}(0, (0.02)^2 I_{15})$ and generate our observational data y using the same ground-truth of x shown in Figure 5.3. We set a budget of 5×10^5 number of forward simulations on our computation cost. We remark that the PDE (5.34) is always solved by the finite element method with fixed mesh size $1/64$, regardless of the resolution of x.

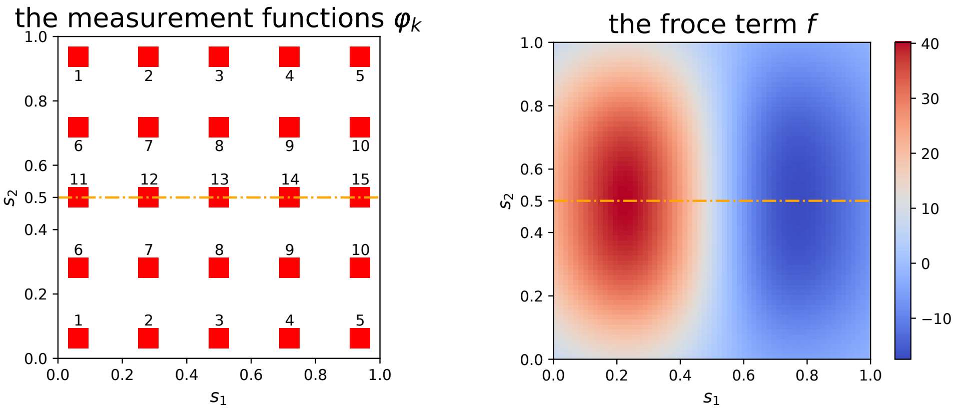


Figure 5.6: The measurement functions φ_k, $k = 1, 2, \dots, 15$, and the force term f for the elliptic Bayesian inverse problem. The dashed lines are the symmetry axes.

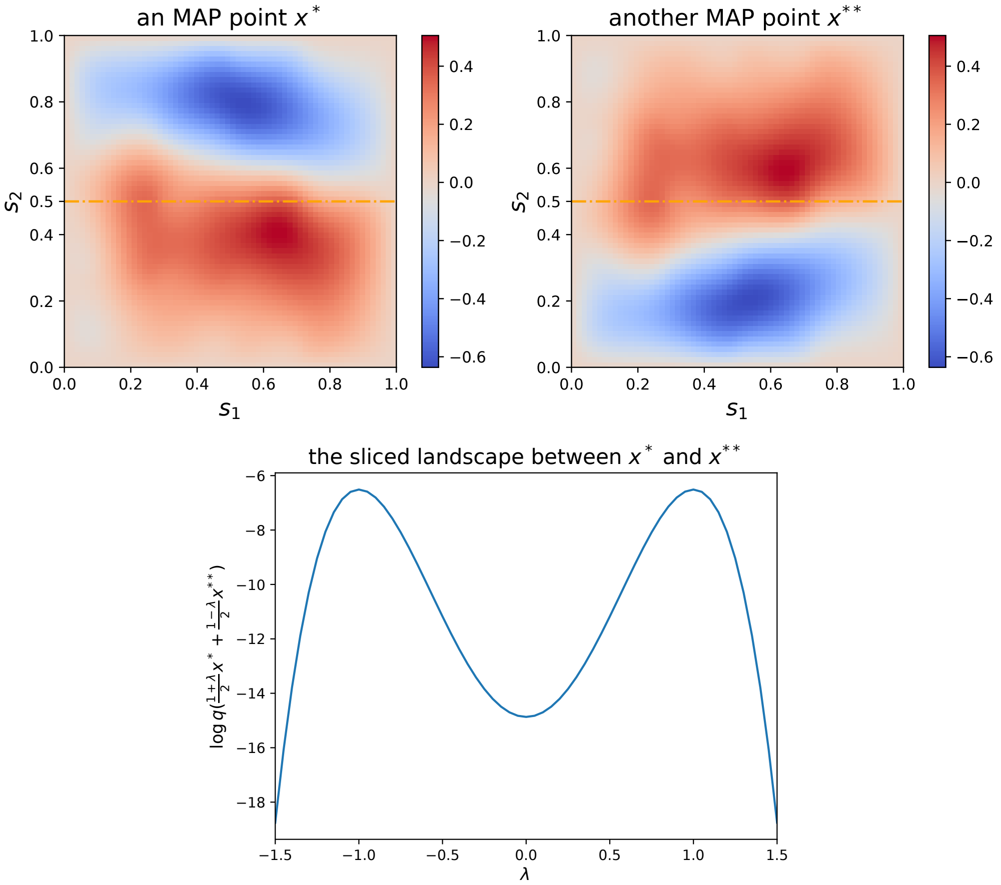


Figure 5.7: The two intrinsically different MAP points x^* and x^{**}, and the sliced landscape between x^* and x^{**}.

In the problem design, the trick of imposing symmetry condition (5.32) guarantees at least two equally important modes in the posterior. One evidence of the existence of two equally important modes is from the Maximum-A-Posterior (MAP) search. We use the gradient descent method from randomly generated points to search for the MAP point. In other words, we solve the optimization

$$\arg\max_x \log q(x).$$

Two intrinsically different MAP points x^* and x^{**} are identified from numerical computation with the same $\log q$ value in Figure 5.7. We see that the two MAP points x^* and x^{**} are mirror symmetric to each other. We also plot the sliced landscape of $\log q$ between x^* and x^{**} in Figure 5.7, which is the curve of $\log q\left(\frac{1+\lambda}{2}x^* + \frac{1-\lambda}{2}x^{**}\right)$ against λ. We can clearly see a double-modal feature of the landscape, suggesting that these two MAP points are highly possible to be representatives from two different modes.

Due to the lack of ground-truth samples, we first compare sample means obtained by different methods. In Figure 5.8 we plot the mean estimates of different methods using 2500 samples. They all look similar and mirror-symmetric along the s_2-direction. The comparison of sample means indirectly suggests the effective target posterior approximation of our MsIGN.

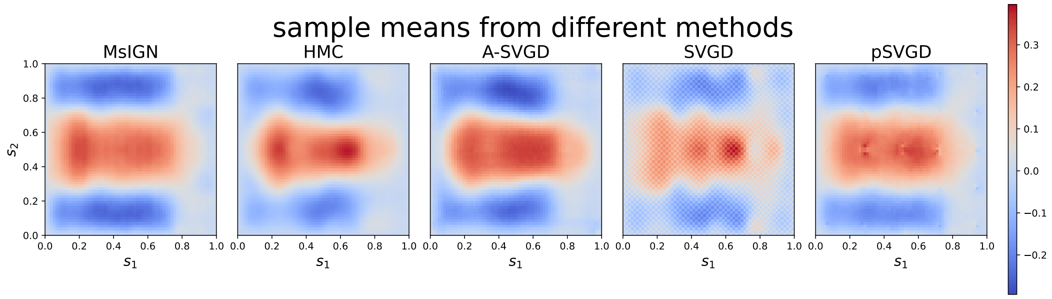


Figure 5.8: The means of the samples generated from different methods.

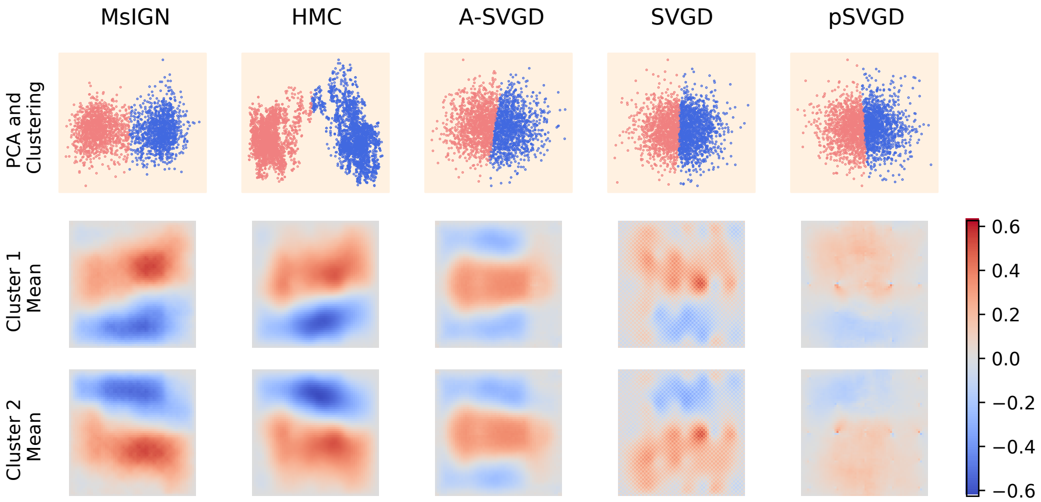


Figure 5.9: Visualization of samples from each method by dimension reduction.

Since the posterior is designed to have at least two mirror symmetric modes with equal importance, we examine the multiple modes capture of different methods. We report the K-means clustering result of 2500 generated samples and means of each cluster in Figure 5.9. In the first row, we embed the high-dimensional samples to a 2D plane by the Principle Component Analysis (PCA) method, and mark the cluster result by the K-means algorithm using colors of red and blue. We can see that samples of the MsIGN and the HMC capture two well-separated modes in the target posterior distribution, but the others fail. Moreover, the HMC captures two modes but is less balanced than the MsIGN and shows an undesired sample correlation. In the second and third rows, we show the means of each cluster. It is interesting to see that two cluster means from all methods are approximately mirror-symmetric to each other. Meanwhile, the means from the MsIGN agree with the HMC. This result also supports that the MsIGN captures double modes of the posterior distribution.

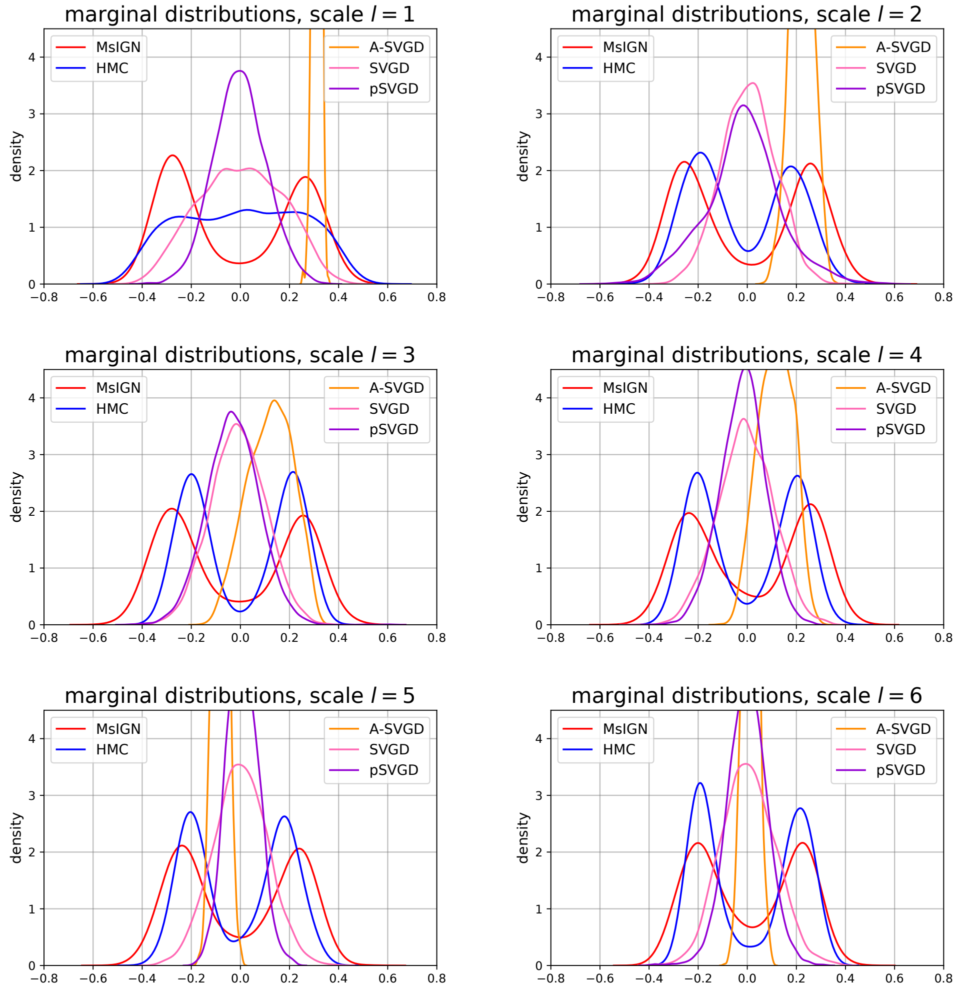


Figure 5.10: Comparison of the marginal distributions along the critical direction $w_{l,*}$ for the elliptic Bayesian inverse problem at all scales $l = 1, \ldots, 6$.

We also check the marginal distributions of the posterior along eigenvectors of the prior covariance matrix, and pick a particular one, which is the eigenvector corresponding to $\sin(\pi s_1)\sin(2\pi s_2)$, to demonstrate that we can capture double modes. The choice of this eigenvector is because the ground-truth of x, as shown in Figure 5.3, is exactly $\sin(\pi s_1)\sin(2\pi s_2)$. Therefore, it is highly likely that the two modes of the posterior are very close to the ground-truth of x and its mirror-symmetry, so along this direction the marginal distribution is likely to be double-modal. Furthermore, we observe that some of the tested methods can stably capture multiple modes in the marginal distribution along this direction. We show the mode capture results for all scales $l = 1, \dots, 6$ in Figure 5.10. All methods except the MsIGN and the HMC failed in detecting all modes, and could even get stuck in the middle. The HMC has acceptable performance, but still suffers from imbalanced modes at some scales. We remark that when $l = 1$, the HMC also fails to capture both modes. This phenomenon might be caused by the aliasing effect. Very rough resolution at this scale pushes the prior to penalize the smoothness too much, and also adds the sensitivity of the likelihood function. Therefore, there is a larger log density gap between modes in the posterior q_1 than other scales, which adds up to the difficulty of multiple modes capture.

5.6 Ablation Study on the Bayesian Inverse Problem

Our MsIGN algorithm introduces new network architecture, as in Section 5.3, and new training strategy, as in Section 5.4. To better under the mechanism of the mode capture ability of the MsIGN and the interplay between network architecture and training strategy, we run extensive experiments with different combinations of choices of network architecture and training strategy. For this purpose, we plot the critical sample marginal distribution along $w_{l,*}$ to verify the mode capture, as we did in Figure 5.10 in Section 5.5.

For the network architecture, we replace the prior conditioning layer by two direct alternatives:

- the split and squeeze layer in the original design of the Glow [54],
- a stochastic nearest-neighbor upsampling layer, which is the prior conditioning layer if assuming the prior is a standard Gaussian distribution whose covariance matrix is an identity matrix.

In the first case, our MsIGN model essentially recovers the Glow. However, for the Glow model, we cannot use the Jeffreys divergence as the objective, because it is infeasible to estimate $D_{\mathrm{KL}}(q\|p_\theta)$. Instead, we use the KL divergence $D_{\mathrm{KL}}(p_\theta\|q)$ as the objective, as in the original design of the Glow [54]. In the second case, the stochastic nearest-neighbor upsampling layer ignores the prior information when it upscales the samples. We call the model in this case as the MsIGN-SNN.

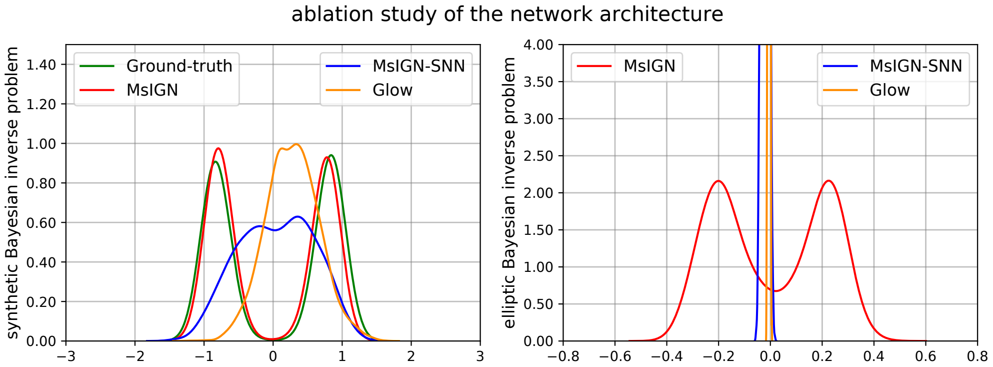


Figure 5.11: Comparison of the marginal distributions at the finest scale ($l = 6$) of models with different network architectures.

Figure 5.11 shows that the prior conditioning layer design is crucial to the performance of the MsIGN on both problems, because neither the MsIGN-SNN nor the Glow has a successful mode capture.

As for the training strategy, we study the effectiveness of the Jeffreys divergence objective and multi-stage training. For the choice of objective function, we try replacing the Jeffreys divergence by:

- the Kullback-Leibler (KL) divergence, which is defined as

$$D_{\mathrm{KL}}(p\|q) = \mathbb{E}_{x\sim p}\left[\log\frac{p(x)}{q(x)}\right],$$

- the kernelized Stein (KS) discrepancy, which is defined as

$$\mathrm{KSD}(p,q) = \mathbb{E}_{x,y\sim p}\left[\delta_{q,p}(x)^T k(x,y)\delta_{q,p}(x)\right],$$

  for some positive kernel k, where $\delta_{q,p}(x) = \nabla\log q(x) - \nabla\log p(x)$, see [68].

For the choice of training manner, in addition to the default multi-stage training, we also consider:

- the single-stage training that directly trains our model on the fine-scale problem.

In other words, the single-stage training does not go from the coarse-scale problem to the fine-scale problem. We remark that the single-stage training using the Jeffreys divergence is infeasible because of the difficulty to estimate $D_{\mathrm{KL}}(q\|p_\theta)$ in the Jeffreys divergence (5.9). We combine different objective choices with different training manner choices, and obtain new methods named the MsIGN-KL, the MsIGN-KL-S, the MsIGN-KS, and the MsIGN-KS-S. For example, the MsIGN-KL trains our MsIGN using the KL divergence as the objective in a multi-stage manner, while the MsIGN-KL-S uses the KL divergence as the objective in a single-stage manner. Similarly, the MsIGN-KS, and the MsIGN-KS-S are two methods using the kernelized Stein discrepancy as the objective in multi-stage manner and single-stage manner, respectively.

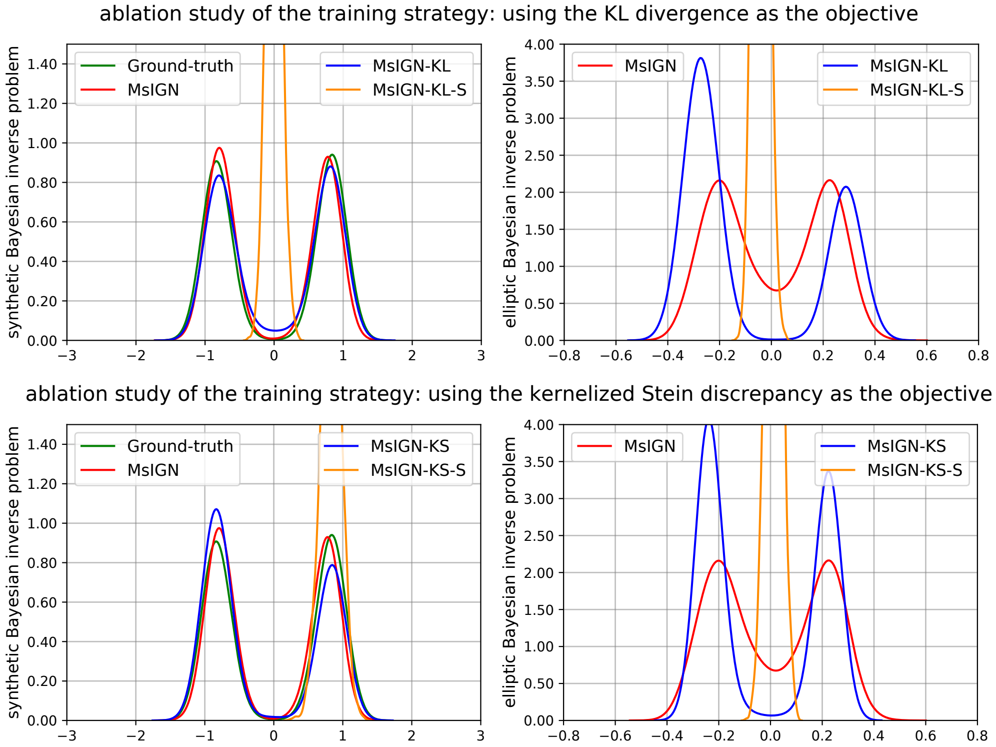


Figure 5.12: Comparison of the marginal distributions at the finest scale ($l = 6$) of models with different training strategy.

Figure 5.12 shows that all models trained in the single-stage manner (the MsIGN-KL-S, the MsIGN-AS-S) face serious mode collapse. Furthermore, our multi-stage training strategy can benefit the training of other objectives, because the MsIGN-KL and the MsIGN-AS are much better than the MsIGN-KL-S and the MsIGN-AS-S respectively. The Jeffreys divergence objective, as used in the MsIGN, leads to more balanced samples in both modes, especially for the complicated elliptic Bayesian inverse problem.

5.7 Multiscale Invertible Generative Networks for the Image Synthesis Task

In this section, we turn to the image synthesis task. When we adopt the same network architecture from the Multiscale Invertible Generative Network (MsIGN) described in Section 5.3, there is no known prior distributions ρ for images, and the density function of the target distribution q is also unknown. In fact, information of the image distribution q is given by its samples $\{x_i\}_{i=1}^N$.

Modification to the Network Architecture

Despite the absence of the prior distribution for images, it is still reasonable to think that the image distribution q has some multiscale structure. Roughly speaking, images of different resolutions usually convey the same set of information to human eyes, but only with different details. For example, let the deterministic upscaling operator $\mathcal{A}$ be the local average operator that downsamples an image from resolution $d \times d$ to resolution $d/2 \times d/2$. In Figure 5.13, we show a 128×128 sample image from the data set CelebA, and the downsampled images after recursively applying $\mathcal{A}$. We can see that as resolution goes down, details in the image is gradually lost, but we could still recognize a human face in the images.

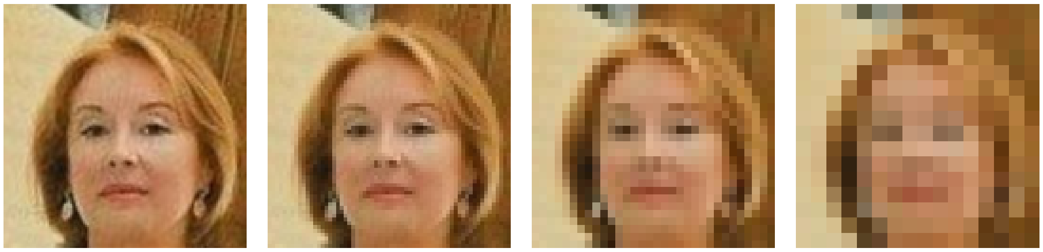


Figure 5.13: One 128×128 sample image from CelebA data set, and its lower resolution versions in 64×64, 32×32 and 16×16, from left to right.

Thus, we recall the definition of the surrogate distribution (5.7) and the scale decoupling relation (5.15):

$$\tilde{q}(x) \propto \rho(x|\mathcal{A}(x))q_c(\mathcal{A}(x)) \approx q(x). \tag{5.35}$$

Here $\tilde{q}$ is the surrogate distribution, $\mathcal{A}$ is the local average operator, and q_c is the target distribution at the coarser resolution. In Bayesian inverse problems, (5.35) holds true because we assume the likelihood has some multiscale structure.

In the image synthesis task, q and q_c are taken as the distributions of images in fine and low resolutions. We should still expect (5.35) to hold true because the low resolution image $\mathcal{A}(x)$ has already outlined objects in the original image x. In other words, $\mathcal{A}(x)$ is already very close to the original image x, with only details in pixel level needed to recover. Here we consider ρ as the distribution that approximates the pixel-level recovery of x from $\mathcal{A}(x)$, and we will use a Gaussian distribution to model ρ. We remark that we do not need to find the optimal ρ, because the difference between the surrogate $\tilde{q}$ and the target q will be corrected by a transport map, as in step 3 of Algorithm 1.

Similar to Algorithm 2, we apply the relation (5.35) recursively until the resolution of image is small enough, so that the coarest-scale image distribution can be easily captured by standard methods. In other words, we still use the settings and notations in Section 5.3. Let L be the number of scales, and for $1 \le l \le L$, let $x_l \in \mathbb{R}^{d_l}$ be the image at scale l, and $\mathcal{A}_l : \mathbb{R}^{d_l} \to \mathbb{R}^{d_{l-1}}$ be the upscaling operator connecting scale l and $l-1$: $x_{l-1} = \mathcal{A}_l(x_l)$. For the target distribution q_l at scale l, although we cannot define it via density as in (5.18), it can be defined as the distribution of downsampled image data x_l at scale l from the data set. The final scale x_L has the same resolution as the target image x, and the dimension and resolution increase as l decreases: $d_1 < d_2 < \ldots < d_L = d$.

To construct the prior conditional layer PC_l at scale l , we assume a simple Gaussian prior $\rho_l = \mathcal{N}(0, \Sigma_l)$ with an isotropic covariance matrix for natural images, i.e, $\Sigma_l = \sigma_l^2 I_{d_l}$ for some $\sigma_l > 0$. Then we determine σ_l from the data set and construct our prior conditioning layer by Theorem 5.3.2.

To determine the prior standard deviation σ_l and construct the prior conditional layer PC_l, we notice that the upscaling operator used here is the local average operator $A_l \in \mathbb{R}^{d_{l-1} \times d_l}$, which gives average on every non-overlapping local 2×2 patch. In other words, $A_l x_l$ is the Frobenius inner product of

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

and the local 2×2 patch of x_l. Therefore, the orthogonal completion $B_l \in \mathbb{R}^{(d_l - d_{l-1}) \times d_l}$ of the matrix A_l on this local 2×2 patch can be equivalent to

$$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \quad \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

We properly choose a normalized version of A_l and B_l, so that $\begin{bmatrix} A_l^T & B_l^T \end{bmatrix}$ is an orthonormal matrix. The orthonormality gives that $A_l^T A_l + B_l^T B_l = I_{d_l}$.

Now using Theorem 5.3.2, we can find an explicit form for $\Sigma_{l|l-1}$, $l \geq 2$:

$$\begin{aligned}\Sigma_{l|l-1} &= \Sigma_l - \Sigma_l A_l^T (A_l \Sigma_l A_l^T)^{-1} A_l \Sigma_l = \sigma_l^2 I_{d_l} - \sigma_l^2 A_l^T (A_l A_l^T)^{-1} A_l \\ &= \sigma_l^2 I_{d_l} - \sigma_l^2 A_l^T A_l = \sigma_l^2 B_l^T B_l.\end{aligned}$$

Therefore, we can choose $V_l = \sigma_l B_l$, such that $\Sigma_{l|l-1} = V_l V_l^T$.

Now we are only left to estimate the scalar σ_l for each $l \geq 2$. We estimate σ_l according to Theorem 5.3.2:

$$x_l = U_{l-1} x_{l-1} + V_l z_l = U_{l-1} x_{l-1} + \sigma_l B_l z_l, \quad z_l \sim \mathcal{N}(0, I_{d_l - d_{l-1}}), \tag{5.36}$$

where x_l, x_{l-1} are the natural images at the resolution $d_l \times d_l$ and $d_{l-1} \times d_{l-1}$. Here U_{l-1} by definition is $U_{l-1} = \Sigma_l A_l^T (A_l \Sigma_l A_l^T)^{-1} = \sigma_l^2 A_l^T (\sigma_l^2 A_l A_l^T)^{-1} = A_l^T$. Plugging it back to (5.36), we have $x_l = A_l^T x_{l-1} + \sigma_l B_l^T z_l$. Now multiplying both sides with B_l, noticing that $B_l B_l^T = I_{d_l - d_{l-1}}$ and $B_l A_l^T = 0$, we obtain

$$B_l x_l = \sigma_l z_l.$$

Now we estimate σ_l by moment matching of both sides. The covariance of the right-hand side is simply $\sigma_l^2 I_{d_l - d_{l-1}}$, because we assume $z_l \sim \mathcal{N}(0, I_{d_l - d_{l-1}})$. The covariance of the left-hand side can be estimated from the data set using the Monte Carlo method, because B_l is known, and x_l is the natural image at resolution d_l. For example, we use 10000 randomly sampled images from each data set and we report our estimates of σ_l in Table 5.3. The estimate of σ_l is quite robust with the random images chosen from the data set.

data set	MNIST	CIFAR-10	CelebA 64	ImageNet 32	ImageNet 64
σ_2	0.67	0.48	0.22	0.32	0.28
σ_3	–	0.46	0.30	0.42	0.36

Table 5.3: Estimates of σ_l for different data sets and scale l.

Modification to the Training Strategy

Since there is no density function for image distribution q, we modify our training strategy for the MsIGN for the image synthesis task. We do not use the Jeffreys divergence $D_{\mathrm{J}}(q\|p_\theta)$ here because we are not given the density of target distribution q to evaluate the KL divergence $D_{\mathrm{KL}}(p_\theta\|q) = \mathbb{E}_{x\sim p_\theta}\left[\log \frac{p_\theta(x)}{q(x)}\right]$.

The learning objective is now set as the KL divergence $D_{\mathrm{KL}}(q\|p_\theta)$, and we search for the optimal network parameter $\theta \in \Theta$ that minimizes the KL divergence. We remark that this is essentially the maximal likelihood estimation, i.e,

$$\max_{\theta \in \Theta} \mathbb{E}_{x \sim q} \left[\log p_\theta(x) \right],$$

where p_θ is the distribution modeled by the MsIGN. The expected value with respect to image distribution q can be estimated by the Monte Carlo method using the samples of q from the data set. We still maintain the hierarchical sampling and multi-stage training strategy as in Algorithms 2 and 4.

5.8 Numerical Experiment on the Image Synthesis Task

We test our model on various data sets, for example, the MNIST, a 28×28-resolution grey-scale handwriting digit data set of 60000 images, the CIFAR-10, a 32×32-resolution color image data set of 60000 images in 10 selected classes, the CelebA 64, a 64×64-resolution color celebrity face data set of more than 2×10^5 images, the ImageNet 32 and the ImageNet 64, 32×32-resolution and 64×64-resolution color image data sets of more than 1×10^6 images in 1000 object classes respectively. We remark that due to the presence of color channels ("RGB" format), the problem dimension is, for example, $d = 3 \times 64 \times 64 = 12288$ for the data set CelebA 64 and ImageNet 64. Parameter settings of network architecture and training strategy of the MsIGN can be found in Appendix B.

We report the bits-per-dimension (BPD) value, which is a shifted and scaled version of the KL divergence:

$$\mathrm{BPD} := -\frac{1}{d} \mathbb{E}_{x \sim q} \left[\log p_\theta(x) \right] = \frac{1}{d} \left(D_{\mathrm{KL}}(q \| p_\theta) - \mathbb{E}_{x \sim q} \left[\log q(x) \right] \right).$$

Because $\mathbb{E}_{x \sim q} \left[\log q(x) \right]$ is a constant that only depends on the target distribution q, a small BPD value means better approximation quality of p to the target distribution q. We compare the BPD value of our MsIGN with our baseline models of flow-based generative networks in Table 5.4. Our MsIGN is superior in the BPD value among the baseline models in almost every data set. Moreover, the MsIGN is more efficient in terms of parameter size: for example, in the experiments shown in Table 5.4, the MsIGN uses 24.4% fewer parameters than the Glow for the CelebA 64 data set, and uses 37.4% fewer parameters than the Residual Flow for the ImageNet 64 data set.

In Figure 5.14 and Figure 5.15, we show synthesized images from the MsIGN after training on the MNIST data set or the CelebA data set. The synthesized images in Figure 5.14 look like real human handwriting digits. And the synthesized human

Model	MNIST	CIFAR-10	CelebA 64	ImageNet 32	ImageNet 64
Real NVP [24]	1.06	3.49	3.02	4.28	3.98
Glow [54]	1.05	3.35	2.20*	4.09	3.81
FFJORD [35]	0.99	3.40	–	–	–
Flow++ [38]	–	3.29	–	–	–
i-ResNet [3]	1.05	3.45	–	–	–
Residual Flow [15]	0.97	**3.28**	–	**4.01**	3.76
MsIGN	**0.93**	**3.28**	**2.15**	4.03	**3.73**

Table 5.4: Comparison of the bits-per-dimension value with baseline models of flow-based generative networks. *: Score obtained by our own reproducing experiment. –: Score not reported.

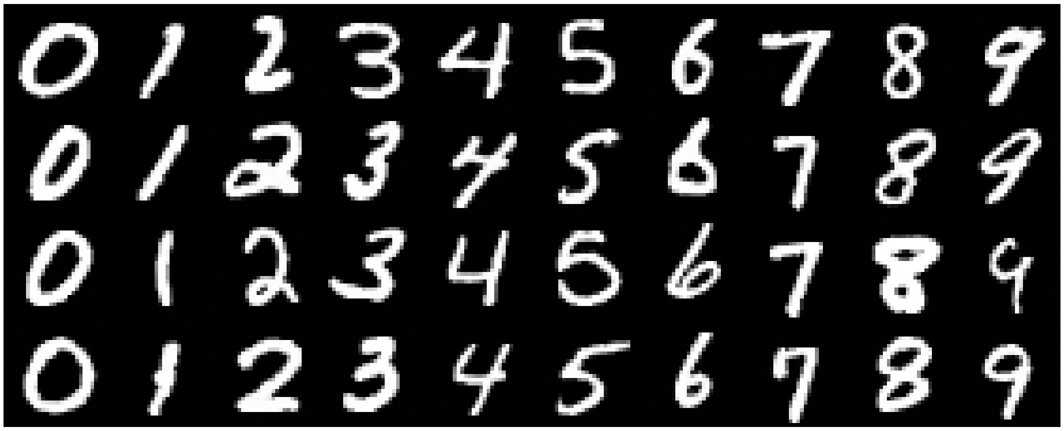


Figure 5.14: Synthesized hand writing digits from the MsIGN after training on the MNIST data set.

faces in Figure 5.15 are hard to distinguish from real human faces without careful watching. It shows good performance of the MsIGN in capturing the data set distribution and synthesizing natural images.

In Figure 5.16, we show linear interpolation of real images from the CelebA in the latent feature space. In each row, the images at both ends, denoted as x_1 and x_2, are randomly chosen from the data set. Recall that $T = T_\theta$ is the invertible transport map represented by the MsIGN, where θ is the network parameter. We do interpolation in the latent space between the latent representation $z_1 = T^{-1}(x_1)$ and $z_2 = T^{-1}(x_2)$ of x_1 and x_2 respectively, and plot images $T(\lambda z_1 + (1 - \lambda)z_2)$ in the intermediate columns with $\lambda = 1/8, 2/8, \dots, 7/8$. In each row, the interpolated images in the middle look like human faces with shared features of the faces at both ends. The latent space seems to be semantically meaningful, which implies that the MsIGN

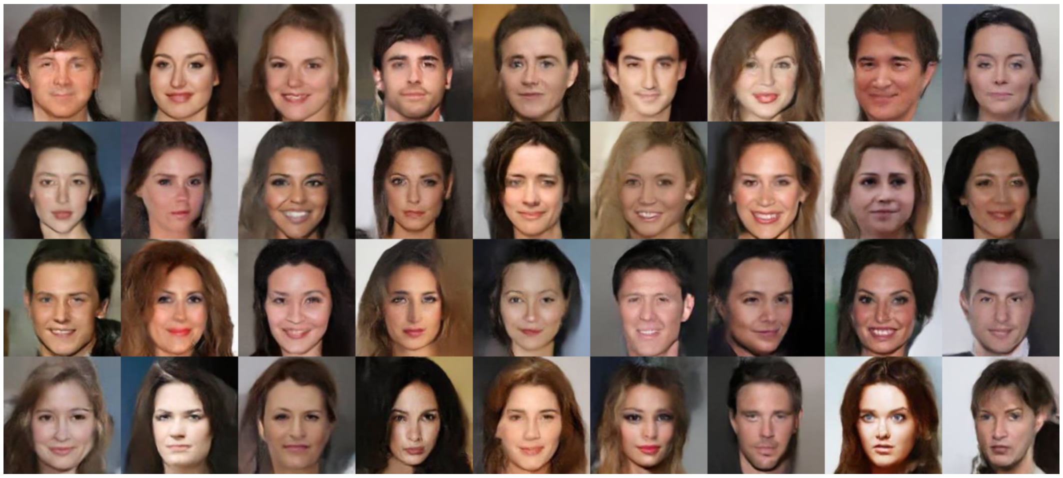


Figure 5.15: Synthesized human faces from the MsIGN after training on the CelebA data set.

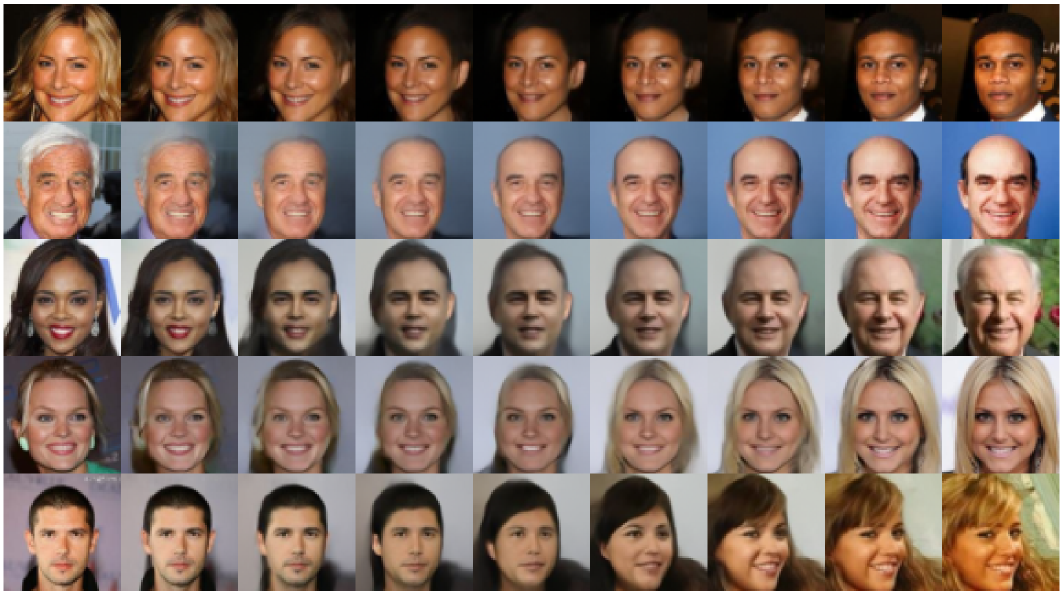


Figure 5.16: Linear interpolation of real images in latent space. In each row, images from left to right correspond to $\lambda = 0, 1/8, \ldots, 7/8, 1$.

captures the data manifold well.

We visualize snapshots at internal checkpoints when the MsIGN maps Gaussian noises to images in Figure 5.17. As Table 5.1 shows, the MsIGN generates a image in the following way: first we generate Gaussian noises $z_1, \ldots, z_L$, then we have $x_1 = F_1(z_1)$, and recursively do

$$\tilde{x}_l = PC_l(x_{l-1}, z_l), \quad x_l = F_l(\tilde{x}_l), \quad \text{for } 2 \leq l \leq L.$$

Here PC_l is the prior conditioning layer, and F_l is the invertible flow at scale l. The output x_L at the final scale is the image generated. The invertible flow F_l is a stack of multiple invertible blocks introduced in Section 5.1. In other words,

$$F_l = f_{1,l} \circ f_{2,l} \circ \cdots \circ f_{n,l},$$

where $f_{i,l}$, $i = 1, \dots, n$ are the invertible blocks, and n is the number of invertible blocks. To visualize this process, from top to bottom in Figure 5.17, we plot four snapshots at each scale l:

- $\tilde{x}_l$ from the surrogate distribution $\tilde{q}_l$,
- the intermediate state when $\tilde{x}_l$ go through $1/3$ of the invertible blocks in F_l,
- the intermediate state when $\tilde{x}_l$ go through $2/3$ of the invertible blocks in F_l,
- x_l from the target distribution q_l.

Notice that at $l = 1$, $\tilde{x}_1$ is not defined, but we can use z_1 instead. Therefore, in each column of Figure 5.17, we plot $4 \times L = 4 \times 4 = 16$ images. It demonstrates how the MsIGN maps Gaussian noises at the top of the column to the images at the bottom of the column.

The first five columns in Figure 5.17 show how the MsIGN maps the latent representation to the image randomly chosen from the data set. The last five columns in Figure 5.17 shows how new images are synthesized from Gaussian noises. From top to bottom we can observe how Gaussian noise is transformed into a human face, and how it grows from a low-resolution one (32×32) to a high-resolution one (128×128). It demonstrates excellent interpretability of internal neurons of the MsIGN, which to the best of our knowledge has not been reported for flow-based generative models before.

5.9 Future Study and Discussion

The Multiscale Invertible Generative Network (MsIGN) and its associated training algorithms make use of the low-dimensional structure to approximate high-dimensional distributions. The hierarchical structure and the multi-stage training strategy benefit the approximation of high-dimensional distributions and help avoid mode collapse. We demonstrate the potential of the MsIGN in the high-dimensional problems like the Bayesian inference problem and the image synthesis task, leaving several interesting topics to follow up.

Due to the constraint from the convolution kernel in the invertible flow, in order to apply the MsIGN to the Bayesian inverse problem, the physical space needs to be discretized as a uniform grid. Besides, the ratio between two adjacent scales needs to be an integer like 2. In fact, the MsIGN only approximates the finite-dimensional discretization of the posterior distribution in the Bayesian inverse problem on a uniform grid. The consistency of the discretizated posterior to the posterior in the function space is guaranteed by the settings of the Bayesian inverse problem, which holds true for most cases in practice. For example, see [18, 88, 21, 49, 89, 42]. It is then natural to ask if the MsIGN can be generalized to other settings of discretization, which would give more flexibility to the MsIGN. To do so, we need to generalize the prior conditional layer and the invertible flow. The generalization of the prior conditional layer is not difficult, but it would be interesting to see if there is a localized prior conditional layer, like the case for the uniform grid presented in this chapter. A localized prior conditional layer would be very beneficial to control the computational cost. On the other hand, the generalization of the invertible flow is non-trivial. Building an invertible flow on a general mesh that is computationally efficient would be an attractive topic to explore.

Recently, the authors in [70, 97] established estimates of the capacity of deep generative networks needed to approximate distributions in the Wasserstein distance, maximum mean discrepancy, or kernelized Stein discrepancy. In the network architecture of the MsIGN, the deep generative network is designed to bridge the difference between the surrogate distribution $\tilde{q}$ and the target distribution q. It would be interesting to study the capacity of deep generative networks needed to push forward $\tilde{q}$ to q up to certain error tolerance. This would require sharper estimates of the difference between $\tilde{q}$ and q in different metrics.

It is also interesting to ask if we could establish a better estimate of the Jeffery divergence, or propose other choices of objective. Using the Jeffreys divergence as the objective function is very helpful to avoid mode collapse in our numerical examples. However, when the problem dimension is too high, or the target posterior is too singular, the importance sampling strategy will become less effective. Thus, building a better estimate of the Jeffery divergence, or proposing other choices of objective, would be very helpful in the training of the MsIGN.

We are also interested in applying the MsIGN to other Bayesian inference problems, for example, more challenging problems as considered in [49, 20], and the data assimilation problems with multiscale structure in the temporal variation, such as those considered in [33]. It is interesting and looks promising to see how deep neural networks can help attack the high-dimensional cases of the Bayesian inference problem.

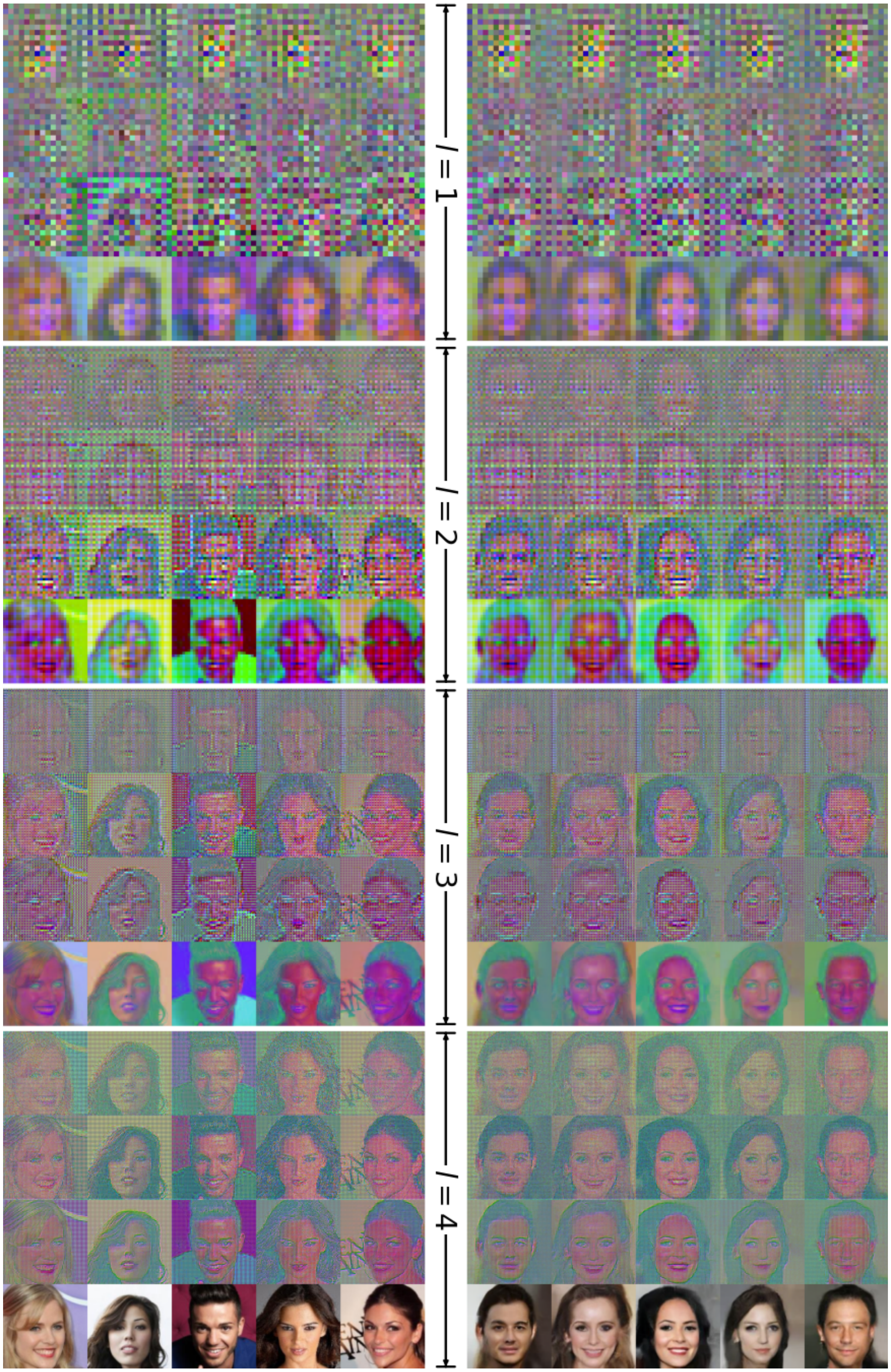


Figure 5.17: Visualization of snapshots at internal checkpoints of the MsIGN when it maps Gaussian noises to images. In each column, from top to bottom, we show how the MsIGN progressively generates new samples from low to high resolution.

BIBLIOGRAPHY

[1] Todd Arbogast. “Numerical subgrid upscaling of two-phase flow in porous media”. In: *Numerical treatment of multiphase flows in porous media*. Springer, 2000, pp. 35–49.

[2] Lynton Ardizzone et al. “Analyzing inverse problems with invertible neural networks”. In: *arXiv preprint arXiv:1808.04730* (2018).

[3] Jens Behrmann et al. “Invertible residual networks”. In: *International Conference on Machine Learning*. 2019, pp. 573–582.

[4] JE Besag. “Comments on “Representations of knowledge in complex systems” by U. Grenander and MI Miller”. In: *J. Roy. Statist. Soc. Ser. B* 56 (1994), pp. 591–592.

[5] Alexandros Beskos et al. “MCMC methods for diffusion bridges”. In: *Stochastics and Dynamics* 8.03 (2008), pp. 319–350.

[6] Yann Brenier. “Polar factorization and monotone rearrangement of vector-valued functions”. In: *Communications on pure and applied mathematics* 44.4 (1991), pp. 375–417.

[7] Michael Brennan et al. “Greedy inference with structure-exploiting lazy maps”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[8] G. A. Chechkin, A. L. Piatnitski, and A. S. Shamaev. *Homogenization: methods and applications*. Vol. 234. American Mathematical Society, 2007.

[9] Changyou Chen, Nan Ding, and Lawrence Carin. “On the convergence of stochastic gradient MCMC algorithms with high-order integrators”. In: *Advances in Neural Information Processing Systems*. 2015, pp. 2278–2286.

[10] Changyou Chen et al. “A unified particle-optimization framework for scalable bayesian sampling”. In: *arXiv preprint arXiv:1805.11659* (2018).

[11] Jiajie Chen and Thomas Y Hou. “Finite Time Blowup of 2D Boussinesq and 3D Euler Equations with $C^{1,\alpha}$ Velocity and Boundary”. In: *Communications in Mathematical Physics* 383.3 (2021), pp. 1559–1667.

[12] Jiajie Chen, Thomas Y Hou, and De Huang. “On the finite time blowup of the De Gregorio model for the 3D Euler equations”. In: *Communications on pure and applied mathematics* 74.6 (2021), pp. 1282–1350.

[13] Peng Chen and Omar Ghattas. “Projected Stein Variational Gradient Descent”. In: *arXiv preprint arXiv:2002.03469* (2020).

[14] Peng Chen et al. “Projected Stein variational Newton: A fast and scalable Bayesian inference method in high dimensions”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 15104–15113.

[15] Tian Qi Chen et al. "Residual flows for invertible generative modeling". In: *Advances in Neural Information Processing Systems*. 2019, pp. 9913–9923.

[16] Tianqi Chen, Emily Fox, and Carlos Guestrin. "Stochastic gradient hamiltonian monte carlo". In: *International conference on machine learning*. 2014, pp. 1683–1691.

[17] Alexandre J Chorin and Xuemin Tu. "Implicit sampling for particle filters". In: *Proceedings of the National Academy of Sciences* 106.41 (2009), pp. 17249–17254.

[18] Simon L Cotter, Masoumeh Dashti, and Andrew M Stuart. "Approximation of Bayesian inverse problems for PDEs". In: *SIAM journal on numerical analysis* 48.1 (2010), pp. 322–345.

[19] Simon L Cotter et al. "MCMC methods for functions: modifying old algorithms to make them faster". In: *Statistical Science* 28.3 (2013), pp. 424–446.

[20] Tiangang Cui, Kody JH Law, and Youssef M Marzouk. "Dimension independent likelihood informed MCMC". In: *Journal of Computational Physics* 304 (2016), pp. 109–137.

[21] Masoumeh Dashti, Stephen Harris, and Andrew Stuart. "Besov priors for Bayesian inverse problems". In: *arXiv preprint arXiv:1105.0889* (2011).

[22] Masoumeh Dashti and Andrew M Stuart. "The Bayesian approach to inverse problems". In: *arXiv preprint arXiv:1302.6989* (2013).

[23] Laurent Dinh, David Krueger, and Yoshua Bengio. "NICE: Non-linear independent components estimation". In: *arXiv preprint arXiv:1410.8516* (2014).

[24] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using real nvp". In: *arXiv preprint arXiv:1605.08803* (2016).

[25] Manfredo P Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.

[26] Tim J Dodwell et al. "A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow". In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1 (2015), pp. 1075–1108.

[27] Theodore D Drivas and Tarek M Elgindi. "Singularity formation in the incompressible Euler equation in finite and infinite time". In: *arXiv preprint arXiv:2203.17221* (2022).

[28] Yalchin Efendiev and Thomas Y Hou. *Multiscale finite element methods: theory and applications*. Vol. 4. Springer Science & Business Media, 2009.

[29] Tarek A El Moselhy and Youssef M Marzouk. "Bayesian inference with optimal maps". In: *Journal of Computational Physics* 231.23 (2012), pp. 7815–7850.

[30] Tarek M Elgindi. “Finite-time singularity formation for $C^{1,\alpha}$ solutions to the incompressible Euler equations on $\mathbb{R}^3$”. In: *Annals of Mathematics* 194.3 (2021), pp. 647–727.

[31] Yihao Feng, Dilin Wang, and Qiang Liu. “Learning to draw samples with amortized stein variational gradient descent”. In: *arXiv preprint arXiv:1707.06626* (2017).

[32] Stuart Geman and Donald Geman. “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images”. In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), pp. 721–741.

[33] Michael B Giles. “Multilevel monte carlo path simulation”. In: *Operations research* 56.3 (2008), pp. 607–617.

[34] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[35] Will Grathwohl et al. “Ffjord: Free-form continuous dynamics for scalable reversible generative models”. In: *arXiv preprint arXiv:1810.01367* (2018).

[36] Alex Graves. “Generating sequences with recurrent neural networks”. In: *arXiv preprint arXiv:1308.0850* (2013).

[37] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[38] Jonathan Ho et al. “Flow++: Improving flow-based generative models with variational dequantization and architecture design”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2722–2730.

[39] Viet Ha Hoang, Christoph Schwab, and Andrew M Stuart. “Complexity analysis of accelerated MCMC methods for Bayesian inversion”. In: *Inverse Problems* 29.8 (2013), p. 085010.

[40] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[41] Klaus Höllig. *Finite element methods with B-splines*. SIAM, 2003.

[42] Bamdad Hosseini and Nilima Nigam. “Well-posed Bayesian inverse problems: Priors with exponential tails”. In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1 (2017), pp. 436–465.

[43] Thomas Y Hou. “The nearly singular behavior of the 3D Navier-Stokes equations”. In: *arXiv preprint arXiv:2107.06509* (2021).

[44] Thomas Y Hou, Tianling Jin, and Pengfei Liu. “Potential singularity for a family of models of the axisymmetric incompressible flow”. In: *Journal of Nonlinear Science* 28.6 (2018), pp. 2217–2247.

[45] Thomas Y Hou and Congming Li. “Dynamic stability of the three-dimensional axisymmetric Navier-Stokes equations with swirl”. In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 61.5 (2008), pp. 661–697.

[46] Thomas Y Hou, Pengfei Liu, and Fei Wang. “Global regularity for a family of 3D models of the axi-symmetric Navier–Stokes equations”. In: *Nonlinearity* 31.5 (2018), p. 1940.

[47] Thomas Y Hou et al. “On finite time singularity and global regularity of an axisymmetric model for the 3D Euler equations”. In: *Archive for Rational Mechanics and Analysis* 212.2 (2014), pp. 683–706.

[48] Thomas Y Hou et al. “Solving Bayesian inverse problems from the perspective of deep generative networks”. In: *Computational Mechanics* 64.2 (2019), pp. 395–408.

[49] Marco A Iglesias, Kui Lin, and Andrew M Stuart. “Well-posed Bayesian geometric inverse problems arising in subsurface flow”. In: *Inverse Problems* 30.11 (2014), p. 114001.

[50] Harold Jeffreys. *Scientific inference*. Read Books Ltd, 2011.

[51] Diederik P Kingma, Max Welling, et al. “An introduction to variational autoencoders”. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[52] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[53] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *ICLR*. 2014.

[54] Durk P Kingma and Prafulla Dhariwal. “Glow: Generative flow with invertible 1x1 convolutions”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 10215–10224.

[55] Durk P Kingma et al. “Improved variational inference with inverse autoregressive flow”. In: *Advances in neural information processing systems* 29 (2016).

[56] Herbert Knothe et al. “Contributions to the theory of convex bodies.” In: *The Michigan Mathematical Journal* 4.1 (1957), pp. 39–52.

[57] Nikola Kovachki et al. “Conditional Sampling With Monotone GANs”. In: *arXiv preprint arXiv:2006.06755* (2020).

[58] Hendrik Anthony Kramers. “Brownian motion in a field of force and the diffusion model of chemical reactions”. In: *Physica* 7.4 (1940), pp. 284–304.

[59] Jakob Kruse et al. “Benchmarking Invertible Architectures on Inverse Problems”. In: *Thirty-sixth International Conference on Machine Learning*. 2019.

[60] Michael Landman et al. “Stability of isotropic self-similar dynamics for scalar-wave collapse”. In: *Physical Review A* 46.12 (1992), p. 7869.

[61] Michael J Landman et al. “Rate of blowup for solutions of the nonlinear Schrödinger equation at critical dimension”. In: *Physical Review A* 38.8 (1988), p. 3837.

[62] Zhen Lei and Thomas Y Hou. “On the stabilizing effect of convection in three-dimensional incompressible flows”. In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 62.4 (2009), pp. 501–564.

[63] BJ LeMesurier et al. “Focusing and multi-focusing solutions of the nonlinear Schrödinger equation”. In: *Physica D: Nonlinear Phenomena* 31.1 (1988), pp. 78–102.

[64] Chang Liu and Jun Zhu. “Riemannian Stein variational gradient descent for Bayesian inference”. In: *Thirty-second aaai conference on artificial intelligence*. 2018.

[65] Jian-Guo Liu and Wei-Cheng Wang. “Characterization and regularity for axisymmetric solenoidal vector fields with application to Navier–Stokes equation”. In: *SIAM Journal on Mathematical Analysis* 41.5 (2009), pp. 1825–1850.

[66] Jian-Guo Liu and Wei-Cheng Wang. “Convergence analysis of the energy and helicity preserving scheme for axisymmetric flows”. In: *SIAM journal on numerical analysis* 44.6 (2006), pp. 2456–2480.

[67] Pengfei Liu. “Spatial Profiles in the Singular Solutions of the 3D Euler Equations and Simplified Models”. PhD thesis. California Institute of Technology, 2017.

[68] Qiang Liu, Jason Lee, and Michael Jordan. “A kernelized Stein discrepancy for goodness-of-fit tests”. In: *International Conference on Machine Learning*. 2016, pp. 276–284.

[69] Qiang Liu and Dilin Wang. “Stein variational gradient descent: A general purpose bayesian inference algorithm”. In: *Advances In Neural Information Processing Systems*. 2016, pp. 2378–2386.

[70] Yulong Lu and Jianfeng Lu. “A universal approximation theorem of deep neural networks for expressing probability distributions”. In: *Advances in neural information processing systems* 33 (2020), pp. 3094–3105.

[71] Guo Luo and Thomas Y Hou. “Formation of finite-time singularities in the 3D axisymmetric Euler equations: a numerics guided study”. In: *SIAM Review* 61.4 (2019), pp. 793–835.

[72] Robert J McCann et al. “Existence and uniqueness of monotone measure-preserving maps”. In: *Duke Mathematical Journal* 80.2 (1995), pp. 309–324.

[73] David W McLaughlin et al. "Focusing singularity of the cubic Schrödinger equation". In: *Physical Review A* 34.2 (1986), p. 1200.

[74] Matthias Morzfeld et al. "A random map implementation of implicit filters". In: *Journal of Computational Physics* 231.4 (2012), pp. 2049–2066.

[75] Radford M Neal et al. "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo* 2.11 (2011).

[76] Frank Nielsen and Richard Nock. "Sided and symmetrized Bregman centroids". In: *IEEE transactions on Information Theory* 55.6 (2009), pp. 2882–2904.

[77] GC Papanicolaou et al. "The focusing singularity of the Davey-Stewartson equations for gravity-capillary surface waves". In: *Physica D: Nonlinear Phenomena* 72.1-2 (1994), pp. 61–86.

[78] Matthew Parno, Tarek Moselhy, and Youssef Marzouk. "A multiscale strategy for Bayesian inference using transport maps". In: *SIAM/ASA Journal on Uncertainty Quantification* 4.1 (2016), pp. 1160–1190.

[79] Noemi Petra et al. "A computational framework for infinite-dimensional Bayesian inverse problems, Part II: Stochastic Newton MCMC with application to ice sheet flow inverse problems". In: *SIAM Journal on Scientific Computing* 36.4 (2014), A1525–A1555.

[80] Danilo Jimenez Rezende and Shakir Mohamed. *Variational Inference with Normalizing Flows*. 2015. arXiv: `1505.05770 [stat.ML]`.

[81] Murray Rosenblatt. "Remarks on a multivariate transformation". In: *The annals of mathematical statistics* 23.3 (1952), pp. 470–472.

[82] Filippo Santambrogio. "Optimal transport for applied mathematicians". In: *Birkäuser, NY* 55.58-63 (2015), p. 94.

[83] Yang Song and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution". In: *Advances in Neural Information Processing Systems* 32 (2019).

[84] Yang Song et al. "Score-based generative modeling through stochastic differential equations". In: *arXiv preprint arXiv:2011.13456* (2020).

[85] Alessio Spantini, Daniele Bigoni, and Youssef Marzouk. "Inference via low-dimensional couplings". In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 2639–2709.

[86] Alessio Spantini et al. "Optimal low-rank approximations of Bayesian linear inverse problems". In: *SIAM Journal on Scientific Computing* 37.6 (2015), A2451–A2487.

[87] Gilbert Strang. "On the construction and comparison of difference schemes". In: *SIAM journal on numerical analysis* 5.3 (1968), pp. 506–517.

[88] Andrew M Stuart. “Inverse problems: a Bayesian perspective”. In: *Acta Numerica* 19 (2010), pp. 451–559.

[89] Tim J Sullivan. “Well-posed Bayesian inverse problems and heavy-tailed stable quasi-Banach space priors”. In: *arXiv preprint arXiv:1605.05898* (2016).

[90] MR Ukhovskii and VI Iudovich. “Axially symmetric flows of ideal and viscous fluids filling the whole space: PMM vol. 32, no. 1, 1968, pp. 59–69”. In: *Journal of applied mathematics and mechanics* 32.1 (1968), pp. 52–62.

[91] Aaron Van den Oord et al. “Conditional image generation with pixelcnn decoders”. In: *Advances in neural information processing systems* 29 (2016).

[92] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. “Pixel recurrent neural networks”. In: *International conference on machine learning*. PMLR. 2016, pp. 1747–1756.

[93] Cédric Villani. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.

[94] Max Welling and Yee W Teh. “Bayesian learning via stochastic gradient Langevin dynamics”. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 681–688.

[95] Dawn B Woodard, Scott C Schmidler, and Mark Huber. “Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions”. In: *The Annals of Applied Probability* 19.2 (2009), pp. 617–640.

[96] Tao Xu et al. “AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks”. In: *arXiv preprint arXiv:1711.10485* (2017).

[97] Yunfei Yang, Zhen Li, and Yang Wang. “On the capacity of deep generative networks for approximating distributions”. In: *Neural Networks* 145 (2022), pp. 144–154.

[98] Olivier Zahm et al. “Certified dimension reduction in nonlinear Bayesian inverse problems”. In: *arXiv preprint arXiv:1807.03712* (2018).

Appendix A

DETAILS OF THE ADAPTIVE MESH METHOD

A.1 Adaptive Mesh Method

Since the solutions of Euler equations quickly become very singular and concentrate in a rapidly shrinking region, despite that the initial data are very smooth, we use the adaptive mesh method to resolve the singular profile of the solutions.

A detailed description of the adaptive mesh method can be found in [44, 71]. Here we will give a brief introduction of the adaptive mesh method. In Appendix A.2, we will list the parameter setting used for the experiments in Part 1.

We will take the following equivalent form of the axisymmetric Euler equations with no swirl as an example in this section,

$$\omega_{1,t} + u^r\omega_{1,r} + u^z\omega_{1,z} = -(n-2-\varepsilon)\psi_{1,z}\omega_1, \tag{A.1a}$$

$$-\psi_{1,rr} - \psi_{1,zz} - \frac{n}{r}\psi_{1,r} = \omega_1, \tag{A.1b}$$

$$u^r = -r\psi_{1,z}, \quad u^z = (n-1)\psi_1 + r\psi_{1,r}, \tag{A.1c}$$

which is (**??**) after making the change of variables: $\omega_1 \to \frac{1}{r}\omega^\theta$, $\psi_1 \to \frac{1}{r}\psi^\theta$. The equations that we solve in Part 1 might be slightly different from this equation, but it does not affect the numerical treatment.

The Euler equations (A.1) are posted as an initial-boundary value problem on the computational domain $(r,z) \in [0,1]\times[0,1/2]$. We introduce two variables $(\rho,\eta) \in [0,1]\times[0,1]$, and the maps

$$r = r(\rho), \quad z = z(\eta).$$

Here we assume these two maps and their derivatives are all analytically known. We also assume that these two maps are monotonically increasing. We will use these two maps to map the physical domain in (r,z) to a computational domain in (ρ,η), so that $\omega_1(r(\rho),z(\eta))$ and $\psi_1(r(\rho),z(\eta))$ as functions of (ρ,η) are relatively smooth.

Let n_ρ, n_η be the mesh resolutions along the r- and z- direction respectively. And let $h_\rho = 1/n_\rho$, $h_\eta = 1/n_\eta$ be the mesh sizes along the r- and z- direction respectively.

We place a uniform mesh on the computation domain of (ρ, η):

$$\mathcal{M}_{(\rho,\eta)} = \left\{(ih_\rho, jh_\eta) : 0 \leq i \leq n_\rho, 0 \leq j \leq n_\eta\right\},$$

This is equivalent to covering the computation domain of (r, z) with the tensor-product mesh:

$$\mathcal{M}_{(r,z)} = \left\{(r(ih_\rho), z(jh_\eta)) : 0 \leq i \leq n_\rho, 0 \leq j \leq n_\eta\right\}.$$

The Vorticity Equation and The Velocity Equation

Let v be some solution variable (either ω_1 or ψ_1), and let $v_{i,j} = v(r(ih_\rho), z(jh_\eta))$ be the discretization of v on the mesh. We can use the following formula to get second-order (in space) approximation of the spatial derivatives of v using the central difference scheme:

$$\begin{aligned}(v_r)_{i,j} &= \frac{\left(v_\rho\right)_{i,j}}{\left(r_\rho\right)_i} \approx \frac{1}{\left(r_\rho\right)_i} \cdot \frac{v_{i+1,j} - v_{i-1,j}}{2h_\rho},\\ (v_z)_{i,j} &= \frac{\left(v_\eta\right)_{i,j}}{\left(z_\eta\right)_j} \approx \frac{1}{\left(z_\eta\right)_j} \cdot \frac{v_{i,j+1} - v_{i,j-1}}{2h_\eta}.\end{aligned}$$

At the boundary of the domain, we need to extend the discretization v beyond the boundary to use the formula above. This can be done by using the symmetry and the boundary conditions. For example, since v is an odd function of z at $z = 0$ and at $z = 1/2$, we have

$$v_{i,-1} = -v_{i,1}, \quad v_{i,n_\eta+1} = -v_{i,n_\eta-1}, \quad 0 \leq i \leq n_\rho.$$

Since v is an even function of r at $r = 0$, we have

$$v_{-1,j} = v_{1,j}, \quad 0 \leq j \leq n_\eta.$$

At $r = 1$, we could extend v by extrapolation:

$$v_{n_\rho+1,j} = 3v_{n_\rho,j} - 3v_{n_\rho-1,j} + v_{n_\rho-2,j}, \quad 0 \leq j \leq n_\eta.$$

With the spatial derivatives available, we can solve the velocity equation (A.1c). For the vorticity equation (A.1a), the time evolution is solved by a second-order explicit Runge–Kutta method.

The Stream Function Equation

We use a B-spline-based Galerkin Poisson solver to solve ψ_1 from (A.1b) on the computation domain of (ρ, η). To start with, we rewrite (A.1b) in the following way:

$$-\frac{1}{r^n r_\rho}\left(r^n \frac{\psi_{1,\rho}}{r_\rho}\right)_\rho - \frac{1}{z_\eta}\left(\frac{\psi_{1,\eta}}{z_\eta}\right)_\eta = \omega_1. \tag{A.2}$$

Next, we multiply both sides with $r^n r_\rho z_\eta \phi_1$ for some suitable test function $\phi_1 \in V$ to be specified below, and integrate both sides over the domain $(\rho, \eta) \in [0,1] \times [0,1]$. After integration by part, we obtain the weak form of the equation (A.1b) for ψ_1: letting

$$a(\psi_1, \phi_1) := \int_{[0,1]^2} \left(\frac{\psi_{1,\rho}}{r_\rho}\frac{\phi_{1,\rho}}{r_\rho} + \frac{\psi_{1,\eta}}{z_\eta}\frac{\phi_{1,\eta}}{z_\eta}\right) r^n r_\rho z_\eta \mathrm{d}\rho \mathrm{d}\eta,$$

and

$$f(\phi_1) := \int_{[0,1]^2} \omega_1 \phi_1 r^n r_\rho z_\eta \mathrm{d}\rho \mathrm{d}\eta,$$

we look for $\psi_1 \in V$ such that for any $\phi_1 \in V$,

$$a(\psi_1, \phi_1) = f(\phi_1).$$

Considering the symmetry and boundary conditions of ψ_1, we define the function space V as

$$\begin{aligned} V = \mathrm{span}\Big\{\phi_1 \in H^1\left([0,1]^2\right) : &\phi_1(-\rho, \eta) = \phi_1(\rho, \eta), \phi_1(1, \eta) = 0, \\ &\phi_1(\rho, -\eta) = -\phi_1(\rho, \eta), \phi_1(\rho, 1-\eta) = -\phi_1(\rho, \eta)\Big\}. \end{aligned}$$

We establish a finite-dimensional subspace $V^k_{w,h}$ of the space V using weighted uniform B-splines of even order k by

$$V^k_{w,h} = \mathrm{span}\left\{w(\rho) B^k_{i,h_\rho}(\rho) B^k_{j,h_\eta}(\eta), 0 \le i \le n_\rho, 0 \le j \le n_\eta\right\}.$$

The weight function $w(\rho) = 1 - \rho^2$ is to enforce the zero Dirichlet boundary condition of ψ_1 at $r = 1$. The function $B^k_{i,h}$ is the shifted and scaled uniform B-spline of order k adjusted to satisfy the boundary condition. Specifically, we have

$$\begin{aligned} B^k_{i,h_\rho}(\rho) &= \frac{b^k_{i,h_\rho}(\rho) + b^k_{i,h_\rho}(-\rho)}{1 + \delta_{i0}}, \\ B^k_{j,h_\eta}(\eta) &= \sum_{M \in \mathbb{Z}} \left(b^k_{j,h_\eta}(2M + \eta) - b^k_{j,h_\eta}(2M - \eta)\right), \end{aligned}$$

where δ_{i0} is the discrete Dirac delta function. We will see that $b^k_{i,h}$ has compact support of size kh, so the infinite sum in $B^k_{j,h_\eta}(\eta)$ will only have finite number of non-zero terms. The function $b^k_{i,h}$ is the shifted and scaled uniform B-spline of order k:

$$b^k_{i,h}(s) = b^k\left(\frac{s}{h} - i + \frac{k}{2}\right), \quad s \in [0,1], i \in \mathbb{Z}, k \text{ is an even number,}$$

such that it is centered at ih. And b^k is the standard B-spline functions, defined recursively [41] by:

$$b^1(x) = \begin{cases} 1, & x \in [0,1] \\ 0, & x \notin [0,1] \end{cases}, \quad b^k(x) = \int_{x-1}^{x} b^{k-1}(x)\mathrm{d}x, \quad k \geq 2.$$

The Galerkin finite element method then discretizes and solves the Poisson equation for ψ_1 by finding $\psi_{1h} \in V^k_{w,h}$, such that for any $\phi_{1h} \in V^k_{w,h}$,

$$a\left(\psi_{1h}, \phi_{1h}\right) = f\left(\phi_{1h}\right).$$

Since $V^k_{w,h}$ is a finite-dimensional space, the above equation can be converted to a sparse linear system, and solved by developed sparse linear solvers.

In our computation, we use $k = 2$, which balances the computational cost with the accuracy. We also remark that, when the dimension n is high, the weight r^n in $a(\psi_1, \phi_1)$ and $f(\phi_1)$ is quite small, and this will make the linear system quite ill-conditioned. To overcome this difficulty, for $n > 5$, we will multiply both sides of (A.2) with $r^m r_\rho z_\eta \phi_1$, and this yields the weak form: letting

$$a\left(\psi_1, \phi_1\right) := \int_{[0,1]^2} \left(\frac{\psi_{1,\rho}}{r_\rho}\frac{\phi_{1,\rho}}{r_\rho} + \frac{\psi_{1,\eta}}{z_\eta}\frac{\phi_{1,\eta}}{z_\eta} + \frac{m-n}{r}\frac{\psi_{1,\rho}}{r_\rho}\phi_1\right) r^m r_\rho z_\eta \mathrm{d}\rho\mathrm{d}\eta,$$

and

$$f\left(\phi_1\right) := \int_{[0,1]^2} \omega_1 \phi_1 r^m r_\rho z_\eta \mathrm{d}\rho\mathrm{d}\eta,$$

we look for $\psi_1 \in V$ such that for any $\phi_1 \in V$,

$$a\left(\psi_1, \phi_1\right) = f\left(\phi_1\right).$$

If $m \neq n$, the bilinear form $a\left(\psi_1, \phi_1\right)$ is no longer symmetric, which would introduce extra computational cost when solving the linear system. However, we still observe robust second-order convergence in the Poisson equation solver. We choose $m = 1$ for the case $n > 5$ in our experiments.

Adaptive Mesh

The key ingredient of the adaptive mesh method is to properly design the map $r = r(\rho)$ and $z = z(\eta)$. In general, we want ω_1 to be smooth as a function of (ρ, η) throughout the computational time.

In the following, we will use y to represent r or z, and x to represent ρ or η, because the construction of the adaptive mesh is the same for these two variables. The only difference is the parameter settings. Following [44], we design the map as

$$y(x) = c \int_0^x p(s) \mathrm{d}s, \tag{A.3}$$

where c is a constant to adjust the size of the domain in y, p is chosen from a parametric family of positive functions. In our practice, we use two parametric families for p. The first parametric family is $p(s) = p(s, x_1, x_2, y_1, y_2)$, where there are four parameters x_1, x_2, y_1, y_2 and $0 < x_1 < x_2 < 1$, $0 < y_1 < y_2 < 1$. The second parametric family is $p(s) = p(s, x_1, x_2, x_3, y_1, y_2, y_3)$, where there are six parameters x_1, x_2, x_3, y_1, y_2, y_3, and $0 < x_1 < x_2 < x_3 < 1$, $0 < y_1 < y_2 < y_3 < 1$.

We take the first parametric family as an example to illustrate the idea. Our design principle is to enforce the following relation to hold approximately:

$$y(x_1) \approx y_1, \quad y(x_2) \approx y_2, \tag{A.4}$$

while still guaranteeing the boundary conditions:

$$y(0) = 0, \quad y(1) = 1, \tag{A.5}$$

if y represents r, and

$$y(0) = 0, \quad y(1) = 1/2, \tag{A.6}$$

if y represents z.

Specifically, we have the following representation of p:

$$p(s, x_1, x_2, y_1, y_2) = p_0 + p_1 q(s - x_1) + p_2 q(s - x_2),$$

where p_0, p_1, p_2 are coefficients to be determined by x_1, x_2, and y_1, y_2, and

$$q(x) = \frac{(1 + x)^{60}}{1 + (1 + x)^{60}},$$

is a smooth function that well approximates the Heaviside step function:

$$q(x) \approx \begin{cases} 1 & x \geq 0, \\ 0 & x < 0. \end{cases}$$

The above approximation suggests that

$$p(s, x_1, x_2, y_1, y_2) \approx \begin{cases} p_0 & s \in [0, x_1)\,, \\ p_0 + p_1 & s \in [x_1, x_2)\,, \\ p_0 + p_1 + p_2 & s \in [x_2, 1]\,. \end{cases}$$

And this property simplifies the approximation relation (A.4) and the constraint (take (A.5) for example) to

$$\begin{cases} p_0 x_1 = y_1, \\ p_0 x_1 + (p_0 + p_1)(x_2 - x_1) = y_2, \\ p_0 x_1 + (p_0 + p_1)(x_2 - x_1) + (p_0 + p_1 + p_2)(x_3 - x_1) = 1, \end{cases}$$

where we use the design (A.3), and assume $c = 1$ for now. We solve this linear system for p_0, p_1, p_2. It is worth noting that the above system is just an approximation, because q is not the exact Heaviside step function. Therefore, we need to choose an appropriate constant c to enforce that $y(1) = 1$. In other words, we could let

$$c = \frac{1}{\int_0^1 \left[p_0 + p_1 q(s - x_1) + p_2 q(s - x_2)\right] \mathrm{d}s}.$$

The construction of the second parametric family $p(s) = p(s, x_1, x_2, x_3, y_1, y_2, y_3)$ is very similar to the first one. We refer the reader to [44] for detailed description.

The approximation relation (A.4) suggests that we could carefully design the parameters x_1, x_2, y_1, y_2, to “zoom-in” to the singular region of the solution. We will describe the strategy to choose the parameters in Section A.2.

A.2 Update of the Adaptive Mesh

Experiments in Chapter 2 and 3

For the axisymmetric Euler equations with no swirl and with Hölder continuous initial data, we use the first parametric family for the adaptive meshes both in r and z direction. The initial setting for the adaptive mesh is,

$$x_1 = 0.012, x_2 = 0.1, y_1 = 0.6, y_2 = 0.9, \qquad \text{for } r = r(\rho),$$
$$x_1 = 0.12, x_2 = 0.1, y_1 = 0.6, y_2 = 0.9, \qquad \text{for } z = z(\eta).$$

Because ω_1 will develop a potential self-similar blow-up near the origin, the maximum location (P, H) of $|\omega_1(r(\rho), z(\eta))|$ as a function of (ρ, η) will be monotonically pushing toward the origin $(0, 0)$. Since our computational mesh $\mathcal{M}_{(\rho,\eta)}$ is uniform, and $\omega_1 = 0$ at the origin, when P or H is too close to 0, there will be few points between the origin and the maximum location, and thus ω_1 could become unresolved.

Therefore, we update our adaptive mesh as long as

$$H < 0.2.$$

Since the singularity formulation is self-similar, this criterion can also monitor the singularity formation along the r-direction.

The new adaptive mesh has the following parameters: for $r = r(\rho)$:

$$\begin{aligned} &x_1 = 1.5r(0.2), x_2 = 10r(0.2), y_1 = 0.6, y_2 = 0.9, && \text{for } r = r(\rho),\\ &x_1 = 1.5z(0.2), x_2 = 10z(0.2), y_1 = 0.6, y_2 = 0.9, && \text{for } z = z(\eta). \end{aligned}$$

This update rule guarantees that, take the z-direction for example, there will be approximately 60% of the points placed between the origin and $1.5H$ after the update. Moreover, there will be approximately 90% of the points placed between the origin and $10H$.

When we update the adaptive mesh, we use a fourth-order piece-wise polynomial interpolation to interpolate the solutions from the old mesh to the new mesh.

Experiments in Chapter 4

For the weak convection model with smooth initial data, we use the second parametric family for the adaptive meshes in r and the first parametric family for the adaptive meshes in z direction. This is because the solution will soon develop a very one-dimensional structure, as shown in Figure 4.2. The smoothness of ω_1 along the $r-$ and $z-$ directions becomes quite anisotropic. The initial setting for the adaptive mesh is,

$$\begin{aligned} &x_1 = 0.002, x_2 = 0.012, x_3 = 0.1, y_1 = 0.05, y_2 = 0.6, y_3 = 0.9, && \text{for } r = r(\rho),\\ &x_1 = 0.12, x_2 = 0.1, y_1 = 0.6, y_2 = 0.9, && \text{for } z = z(\eta). \end{aligned}$$

Since ω_1 is quite anisotropic in r and z, we update them separately.

We update the adaptive mesh in z direction as long as

$$H < 0.2.$$

The new adaptive mesh for z has the following parameters:

$$x_1 = 1.5z(0.2), x_2 = 10z(10), y_1 = 0.6, y_2 = 0.9.$$

As for the r direction, Figure 4.2 suggests that $-\omega_1$ is very flat in r near $r = 0$, but then decays to zero at the far field. The most singular part of $-\omega_1$ is not near the $r = 0$. Instead, we consider designing the new adaptive mesh to smooth the derivative $\omega_{1,r}$. Let $\chi(\rho) = |\omega_{1,r}(r(\rho), z(H))|$. Our numerical observation shows that χ is a unimodal function: $\chi(0) = 0$, and it monotonically increases to its maximum, and drops down a very small value. Letting $X = \max_{\rho\in[0,1]} \chi(\rho)$, we define P_1, P_2 as

$$P_1 = \inf\left\{\rho : \chi(\rho) > \frac{1}{5}X\right\}, \quad P_2 = \sup\left\{\rho : \chi(\rho) > \frac{1}{5}X\right\}.$$

Then we update the adaptive mesh in r direction as long as

$$P_2 - P_1 < 0.2.$$

The new adaptive mesh for r has the following parameters:

$$x_1 = r(P_1), x_2 = r(P_2), x_3 = 2r(P_2) - r(P_1), y_1 = 0.2, y_2 = 0.6, y_3 = 0.9.$$

Roughly speaking, if there are too few points near the maximum location of χ, which is where $-\omega_1$ drops fastest in r, we update the adaptive mesh in r to place more points in that region.

Similarly, when we update the adaptive mesh, we use a fourth-order piece-wise polynomial interpolation to interpolate the solutions from the old mesh to the new mesh.

Appendix B

EXPERIMENTAL SETTINGS OF THE MULTISCALE INVERTIBLE GENERATIVE NETWORKS

We describe detailed settings of the network architecture and training strategy of the MsIGN in our numerical experiments in Part 2 in this section.

As shown in Figure 5.1, the MsIGN has L scales, and at each scale l, the MsIGN consists of two parts: the prior conditioning layer PC_l, and the invertible flow F_l. The prior conditioning layer PC_l is intrinsically a linear transformation. Theorem 5.3.2 gives the closed form formula for the prior conditioning layer PC_l. The invertible flow F_l is a stack of multiple invertible blocks of Glow [54]. We will use K as the number of invertible blocks in each invertible flow. Each invertible block consists of three invertible units: actnorm, invertible 1×1 convolution and affine coupling. In each affine coupling unit, functions f and g are modeled by deep neural networks to introduce nonlinearity to the unit. Following the practice in Glow [54], the network structure of functions f and g is the concatenation of 3 convolution neural networks and 2 ReLU activation layers in turn. The hidden channel size H controls the capacity of the deep neural networks modeling functions f and g.

As for the training of the MsIGN, the multi-stage training of the MsIGN follows the Algorithm 4. At stage $l > 1$, the specific training of the invertible flow of the MsIGN follows the Algorithm 3, where we need to specify the sample size (also called minibatch size) N, learning rate η. We remark that for the Bayesian inverse problem, the number of iterations M is calculated based on the computation budget for that problem. For the image synthesis task, the number of iterations M is characterized by the number of epochs E, which is the number of times that the whole data set has been fed into the model.

We list the values of L, K, H, N, η for the Bayesian inverse problem in Table B.1. The column starting with "Synthetic" gives the settings for the synthetic Bayesian inverse problems. The column starting with "Elliptic" gives the settings for the elliptic Bayesian inverse problems. We also list the values of L, K, H, N, η, E for different data sets in the image synthesis task in Table B.2.

Parameter	Synthetic	Elliptic
L	6	6
K	16	32
H	32	64
N	100	100
η	1×10^{-4}	1×10^{-4}

Table B.1: Parameter settings for the MsIGN in the Bayesian inverse problem in Section 5.5.

Parameter	MNIST	CIFAR-10	CelebA 64	ImageNet 32	ImageNet 64
L	2	3	3	3	3
K	32	32	32	32	32
H	512	512	512	512	512
N	400	400	200	400	400
η	1×10^{-5}	1×10^{-5}	1×10^{-4}	1×10^{-4}	1×10^{-4}
E	2000	2000	1000	400	200

Table B.2: Parameter settings for the MsIGN on different data sets in the image synthesis tasks in Section 5.8.