Initial-value iterative neural network for solitary wave computations

Jin Song

supervisor： Zhenya Yan

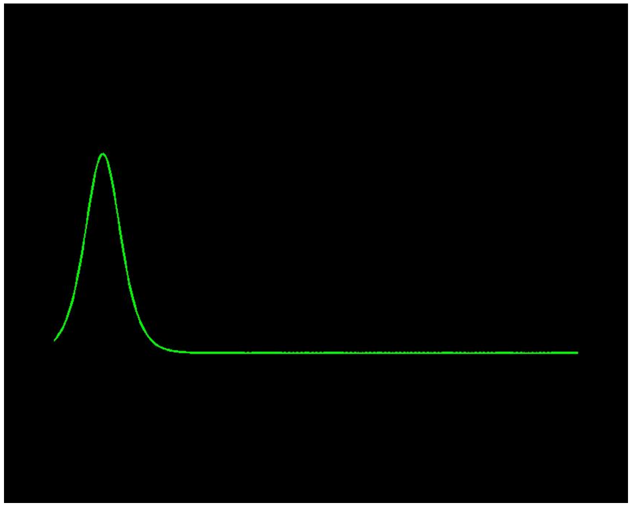
Joint work with Ming Zhong, Zhenya Yan and George Em Karniadakis

KLMM

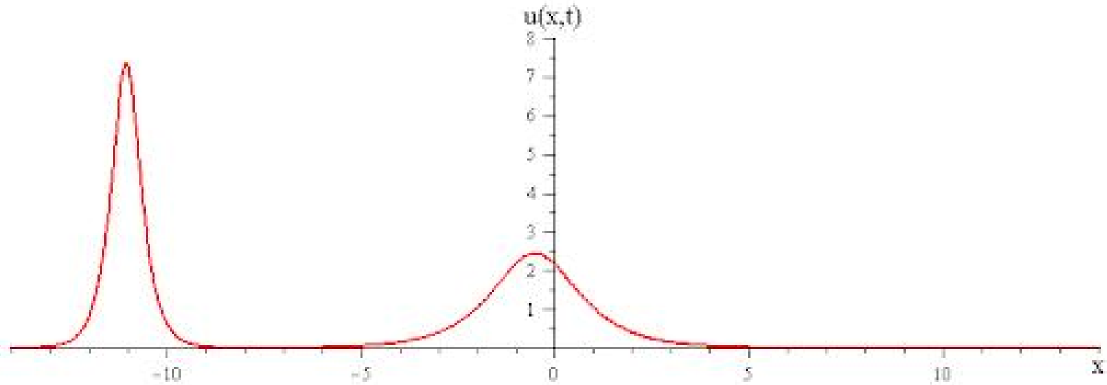
2023/12/12

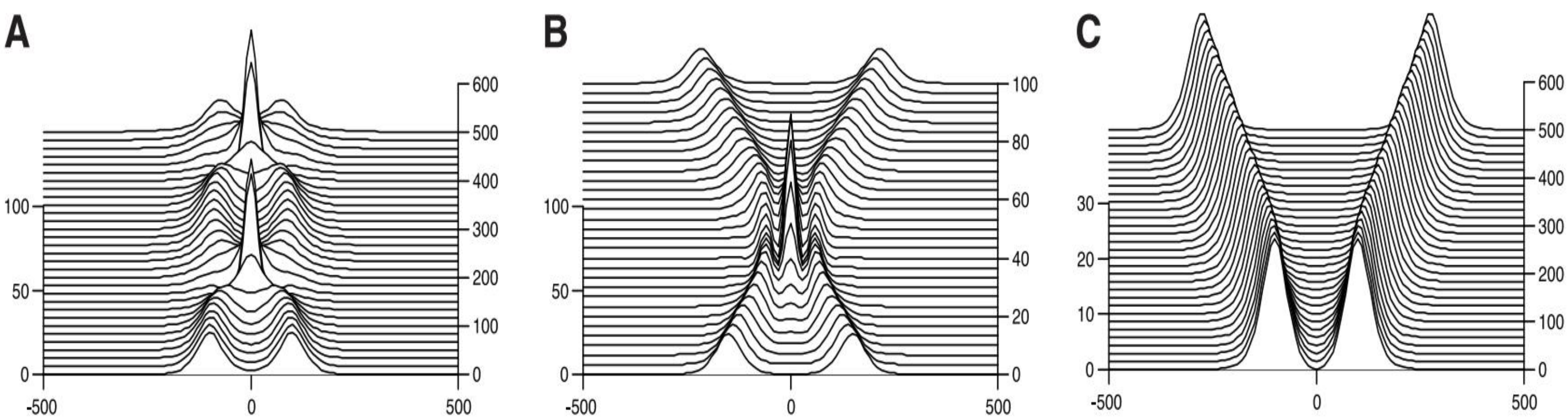
Ⅰ、Introduction

Solitary wave



Soliton





Ⅰ、Introduction

In 1844, John Scott Russell **[Proc. Roy. Soc., Edinburgh, 319 (1844)]**

Boussinesq: $U_{tt} - U_{xx} + 3(U^2)_{xx} - U_{xxxx} = 0$ **[J. Math. Pure. Appl. 17, 55–108 (1872)]**

KdV: $U_t + 6UU_x + U_{xxx} = 0$ **D. J. Korteweg, and G. de Vries [Phil. Mag. 39, 422(1895)]**

NLS: $U_t - U_{xx} - U|U|^2 = 0$

d-dimensional generalized NLS with potential

$$iU_t - \Delta U + V(\mathbf{x})U + \mathcal{N}(\mathbf{x}, |U|^2)U = 0 \tag{1}$$

where $U = U(\mathbf{x}, t)$ is a complex field of the d-dimensional spatial variable $\mathbf{x} \in \mathbb{R}^d$ and time t, $\Delta = \partial_{x_1}^2 + \partial_{x_2}^2 + \cdots + \partial_{x_d}^2$, $V(\mathbf{x})$ a real or complex potential, and $\mathcal{N}(\mathbf{x}, |U|^2)$ a function of $\mathbf{x}$ and intensity $|U|^2$.

$$U(\mathbf{x}, t) = u(\mathbf{x})e^{i\mu t} \tag{2}$$

$$Lu = 0, \quad \text{where} \quad L = -\Delta + V(\mathbf{x}) + \mathcal{N}(\mathbf{x}, |u|^2) - \mu. \tag{3}$$

The system can be written in the following form:

$$\mathbf{L_0 u}(\mathbf{x}) = 0$$

where $\mathbf{L_0}$ is a nonlinear operator, $\mathbf{u}(\mathbf{x}) \in \mathbb{C}^m$ is a complex-valued vector solitary wave solution, and $\mathbf{u} \to 0$ as $|\mathbf{x}| \to \infty$.

Traditional numerical methods

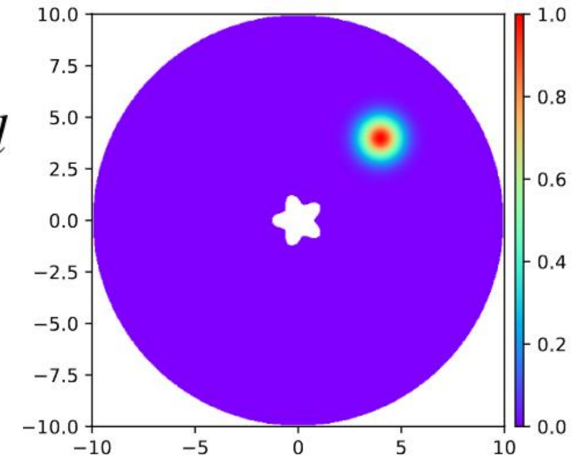
- **Shooting method** **[Commun. ACM, 5(12) 613-614; J. Phys. A- Math. Gen. 20(6) (1987) 1411.]**
- **Iterative methods** $\mathbf{u}_{n+1} = \mathcal{M}_n \mathbf{u}_n$
  - Petviashvili method **[J. Comput. Phys. 226 1668-1692.]**
  - Imaginary-time evolution method **[SIAM J.Sci.Comput.25 1674–1697.]**
  - Squared-operator iteration **[Stud. Appl. Math. 118 153-197.]**
  - Newton-conjugate-gradient **[J. Comput. Phys. 228 7007–7024.]**

Discretization scheme

- Finite different method
- Finite element method
- Discontinuous Galerkin method
- Spectral method
- Spectral elements methods
- … …

High dimensional problem? $M \sim N^d$

Complex computational domains?



Scientific Machine Learning (SciML)

- Deep Ritz method **[Commun. Math. Stat. 6 1-12.]**
- Deep Galerkin method **[J. Comput. Phys. 375 1339-1364.]**
- Physics-informed neural networks **[J. Comput. Phys. 378 686.]**
- Neural Operator **[JMLR 24 1-97]**
- …… ……

Numerical methods	Machine learning methods
Mesh-based	Mesh-free
Discretization	Automatic differentiation
No generalization	Generalization

Problem statement

$$\mathbf{L}_0\mathbf{u}(\mathbf{x}) = 0$$

where $\mathbf{L}_0$ is a nonlinear operator, $\mathbf{u}(\mathbf{x}) \in \mathbb{C}^m$ is a complex-valued vector solitary wave solution, and $\mathbf{u} \to 0$ as $|\mathbf{x}| \to \infty$.

Purpose： NN(x; θ)

$$\mathrm{NN}(\mathbf{x};\theta) \approx \mathbf{u}(\mathbf{x})$$

- **Model:** Build a loss function $\mathcal{L}_1$ according to PDE
- **Optimization:** Minimize the loss function

$$\theta_0 = \operatorname{argmin} \mathcal{L}_1(\theta)$$

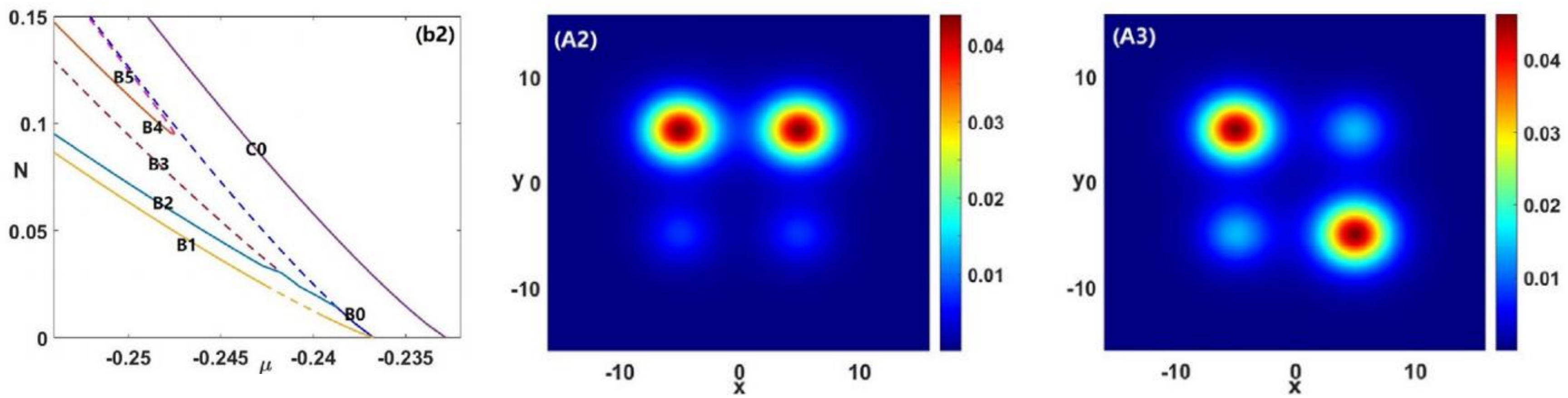
Problem: multi-solution

Problem 1: Trivial solution

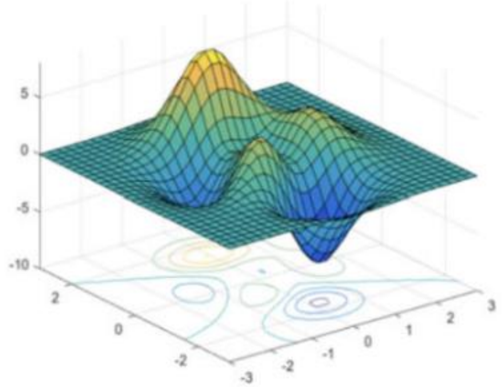
$$\mathbf{u}(\mathbf{x}) \equiv 0$$

Problem 2: Multi-solution

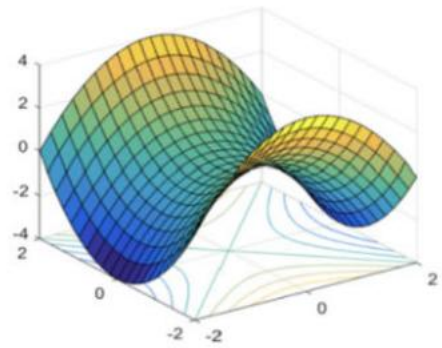
- **Degenerate state**
- **Symmetry breaking bifurcations**



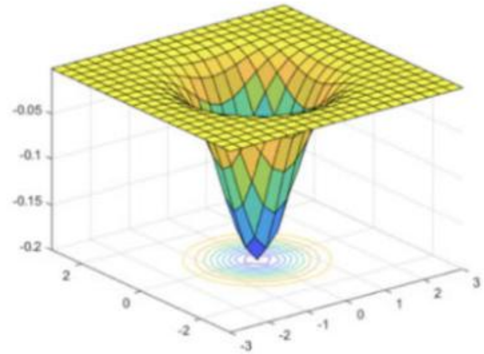
Ⅱ、Methodology



Local minimum

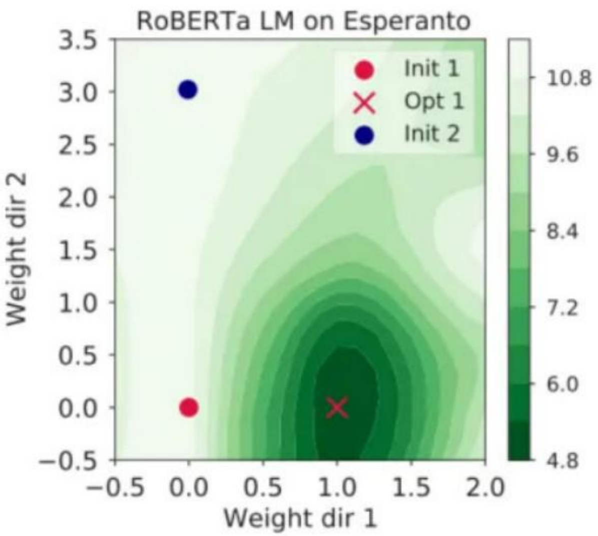


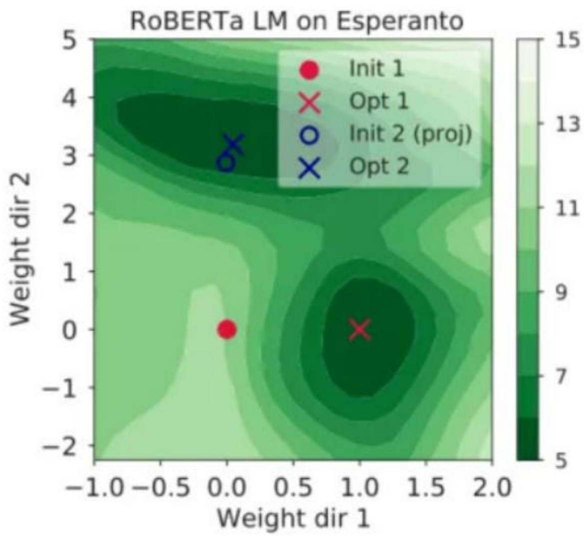
Saddle point



Barren plateaus

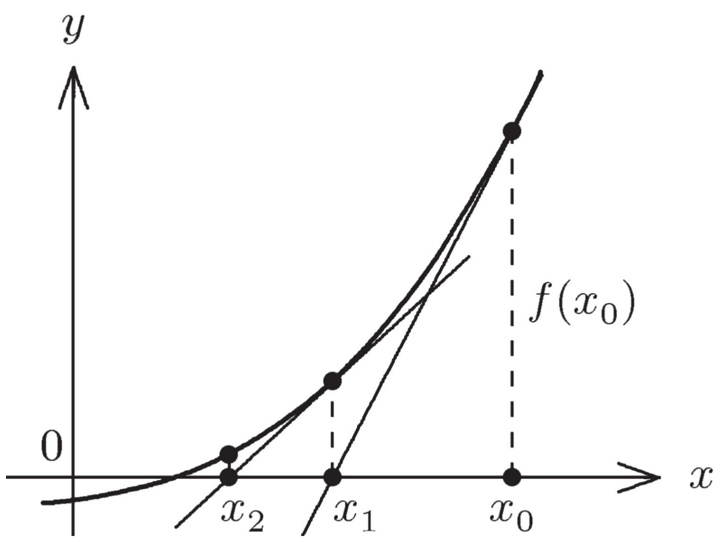
Where can you get to already decided at the beginning

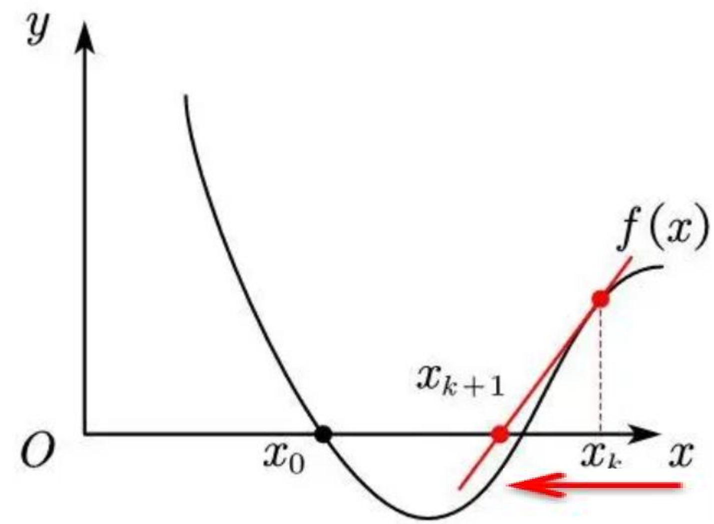




Usually go to the local minimum that is closest to your initial parameter

Analyzing monotonic linear interpolation in neural network loss landscapes. arXiv preprint arXiv:2104.11044.





Newton method

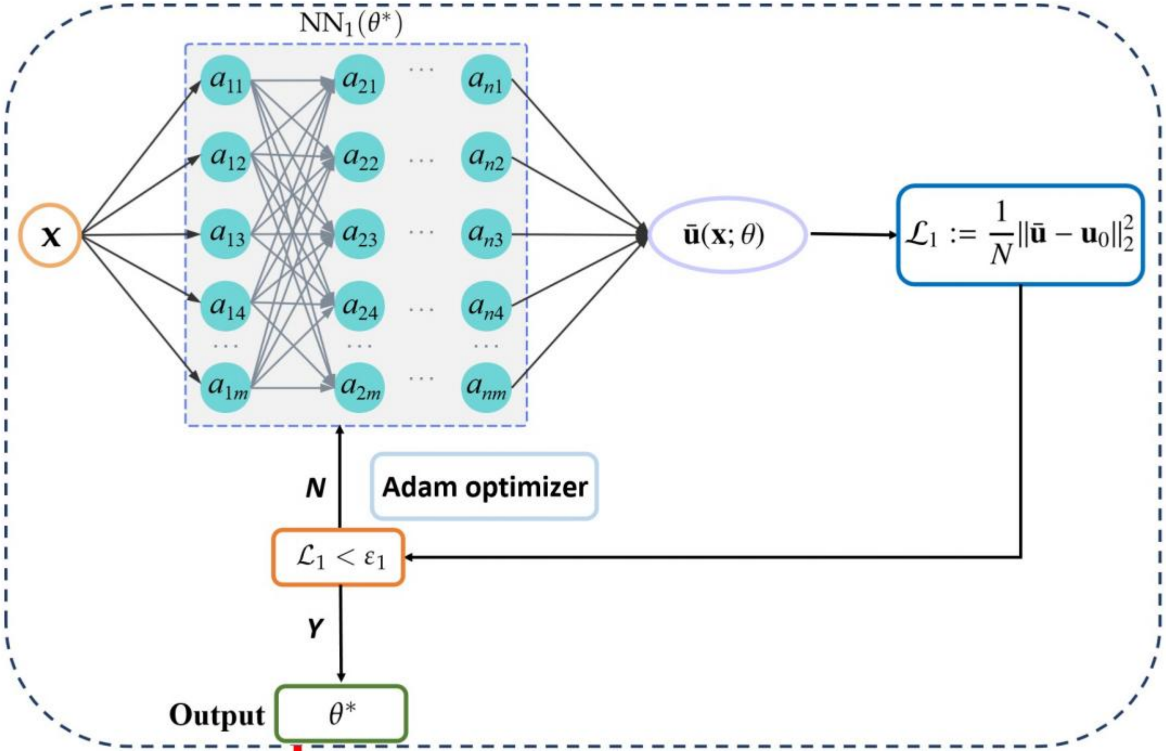
Initial-value

Initial-value iterative neural network (IINN)

Given an appropriate initial value $\mathbf{u}_0$, such that it is sufficiently close to $\mathbf{u}^*$.

$\mathbf{NN}_1$ Train the network parameters θ by minimizing the meansquared error loss.

$$\mathcal{L}_1 := \frac{1}{N}\|\bar{\mathbf{u}} - \mathbf{u}_0\|_2^2 = \frac{1}{N}\sum_{i=1}^{N}|\bar{\mathbf{u}}(\mathbf{x}_i) - \mathbf{u}_0(\mathbf{x}_i)|^2.$$



Initial-value iterative neural network (IINN)

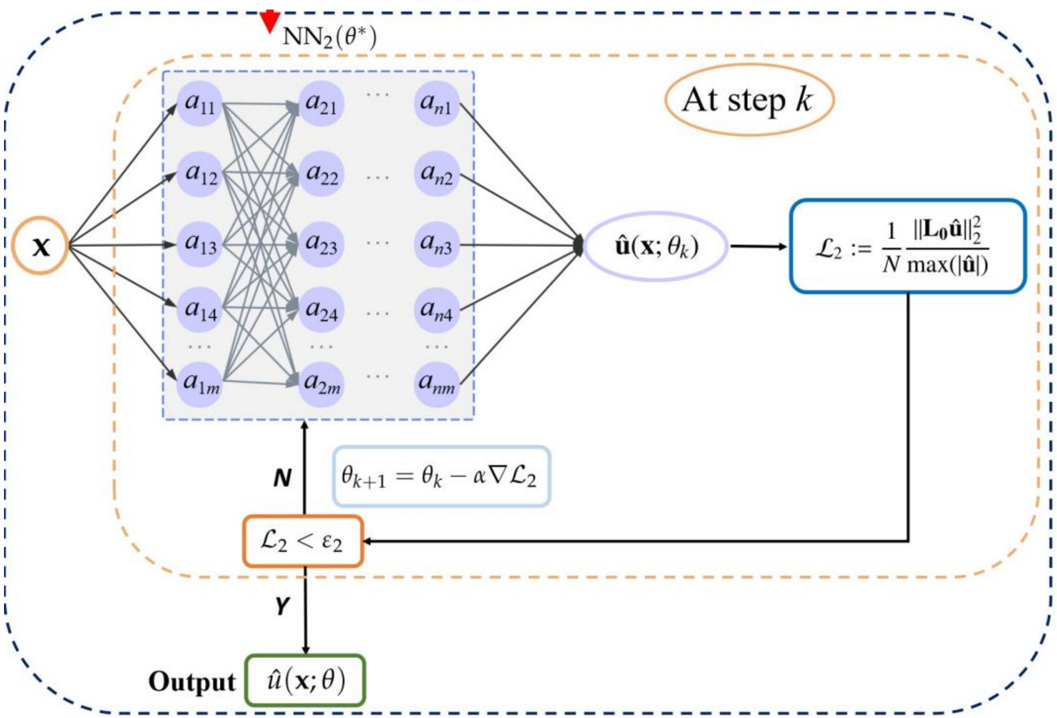
NN_2

Initialize the parameters θ of NN2 with the learned ones from NN1

$$\theta_0 = \operatorname{argmin} \mathcal{L}_1(\theta)$$

Define the loss function L2 as follows
and utilize SGD or Adam optimizer to minimize it

$$\mathcal{L}_2 := \frac{1}{N}\frac{\|\mathbf{L}_0\hat{\mathbf{u}}\|_2^2}{\max(|\hat{\mathbf{u}}|)} = \frac{1}{N}\frac{\sum_{i=1}^{N}|\mathbf{L}_0\hat{\mathbf{u}}(\mathbf{x}_i)|^2}{\max_i(|\hat{\mathbf{u}}(\mathbf{x}_i)|)}$$



Ⅱ、Methodology

Algorithm 1 The framework of initial value iterative neural network (IINN)

Require: Operator $\mathbf{L_0}$ in (2); initial state $\mathbf{u}_0$; error threshold ε_1 and ε_2; training data $\{\mathbf{x}_i, \mathbf{u}_0(\mathbf{x}_i)\}_{i=1}^N$; learning rate α, maximum iteration number K.

Ensure: Output $\hat{\mathbf{u}}$.

For NN_1, randomly initialize the parameters θ_0 s.t. they satisfy the normal distribution. For network output $\bar{\mathbf{u}}(\mathbf{x}, \theta_0)$, $\mathcal{L}_1 := \frac{1}{N}\|\bar{\mathbf{u}} - \mathbf{u}_0\|_2^2$.

for $k = 0 : K$ **do**
  if $\mathcal{L}_1(\theta_k) < \varepsilon_1$ **then**
    $\theta^* = \theta_k$;
    break;
  else
    Apply the Adam optimizer update parameters θ_k;
  end if
end for

For NN_2, initialize the parameters $\theta_0 = \theta^*$ and set $k = 0$. For network output $\hat{\mathbf{u}}(\mathbf{x}, \theta_0)$, $\mathcal{L}_2 := \frac{1}{N}\frac{\|\mathbf{L_0}\hat{\mathbf{u}}\|_2^2}{\max(|\hat{\mathbf{u}}|)}$.

while $\mathcal{L}_2 \geq \varepsilon_2$ **do**
  $\theta_{k+1} = \theta_k - \alpha\nabla\mathcal{L}_2$;
  $k = k + 1$;
end while

Ⅱ、Methodology

Remark 1: From the perspective of numerical iteration, for NN2, we iterate the network parameters with $\hat{\mathbf{u}}(\theta_0)$ as the initial value, such that $\hat{\mathbf{u}}$ satisfies the Eq by minimizing loss function L2.

From a machine learning perspective, the approach is known as transfer learning, where knowledge gained from training one model is transferred to another model, typically when the two models have similar tasks or domains. By initializing NN2 with the parameters of NN1, we can leverage the pre-trained model's learned representations and potentially achieve better performance, especially if the new task or data is related to the original task or data on which NN1 is trained.

Lemma 1 *Let* $\Lambda = \bigcup_{i=1}^{N} \Lambda_i$, *where* $\Lambda_i = \{\theta_i \,|\, \mathbf{L_0}\hat{\mathbf{u}}(\theta_i) = 0, \|\hat{\mathbf{u}}(\theta_i^m) - \hat{\mathbf{u}}(\theta_i^n)\|_2 = 0 \,\text{for}\, m \neq n\}$ *and N is the number of distinct solitary wave solutions.*

Then, $\theta_i \in \Lambda_i$ *is isolated in* Λ_j *for* $i \neq j$.

Theorem 1 *[58] If f is a C^2 function and θ^* be a strict saddle. Assume that learning rate* $0 < \alpha < \frac{1}{\rho}$

then

$$\Pr\left(\lim_{k} \theta_k = \theta^*\right) = 0.$$

Theorem 7 (Theorem III.7, Shub (1987)) *Let* 0 *be a fixed point for the* C^r *local diffeomorphism* $\phi : U \to E$, *where* U *is a neighborhood of* 0 *in the Banach space* E. *Suppose that* $E = E_s \oplus E_u$, *where* E_s *is the span of the eigenvectors corresponding to eigenvalues of magnitude less than or equal to* 1 *of* $D\phi(0)$, *and* E_u *is the span of the eigenvectors corresponding to eigenvalues of magnitude greater than* 1 *of* $D\phi(0)$. *Then there exists a* C^r *embedded disk* W^{cs}_{loc} *that is tangent to* E_s *at* 0 *called the* local stable center manifold. *Moreover, there exists a neighborhood* B *of* 0, *such that* $\phi(W^{cs}_{loc}) \cap B \subset W^{cs}_{loc}$, *and* $\cap_{k=0}^{\infty} \phi^{-k}(B) \subset W^{cs}_{loc}$.

Ⅱ、Methodology

Theorem 2 *For a given soliton state* $\mathbf{u}^*$, *suppose that the initial state* $\mathbf{u}_0$ *is sufficiently close to* $\mathbf{u}^*$ *and the output of* NN_1 $\bar{\mathbf{u}}(\theta^*) = \mathbf{u}_0$. *And* θ^* *satisfies* $d(\theta^*, \Lambda_i) < d(\theta^*, \Lambda_j)$ $(j \neq i)$, *where* i *satisfies* $\hat{\mathbf{u}}(\theta_i) = \mathbf{u}^*$. *Then, the output of* NN_2 $\hat{\mathbf{u}}$ *is sufficiently close to* $\mathbf{u}^*$, *for sufficiently small learning rate* α.

Remark 2: In the scheme of IINN, the choice of initial value $\mathbf{u}_0$ is crucial as it determines the type of solution we ultimately obtain.

Usually, based on the characteristics of the system and our understanding of the system, we can estimate the initial value using physical background knowledge or past experience.

For example, 1D NLS admits sech-type soliton solution. Therefore, we can take $u_0(x) = A\mathrm{sech}(x)$ then by adjusting the coefficient A to make $|Lu_0|$ smaller than a certain threshold.

Example 1.1 (Solitons of the 1D NLS equation with Kerr nonlinearity).

$\mathcal{N}(x,|U|^2)U$ is taken as the Kerr nonlinear term $g|U|^2U$

$$iU_t - U_{xx} + V(x)U + g|U|^2U = 0,$$

$$Lu = 0, \quad L = -\partial_{xx} + V(x) + g|u|^2 - \mu.$$

Case 1.—Bright soliton of the 1D NLS equation with V = 0 and g = −1.

$$u(x) = \sqrt{-2\mu}\,\mathrm{sech}(\sqrt{-\mu}x), \qquad \mu < 0.$$

$$u_0(x) = \mathrm{sech}(x),$$

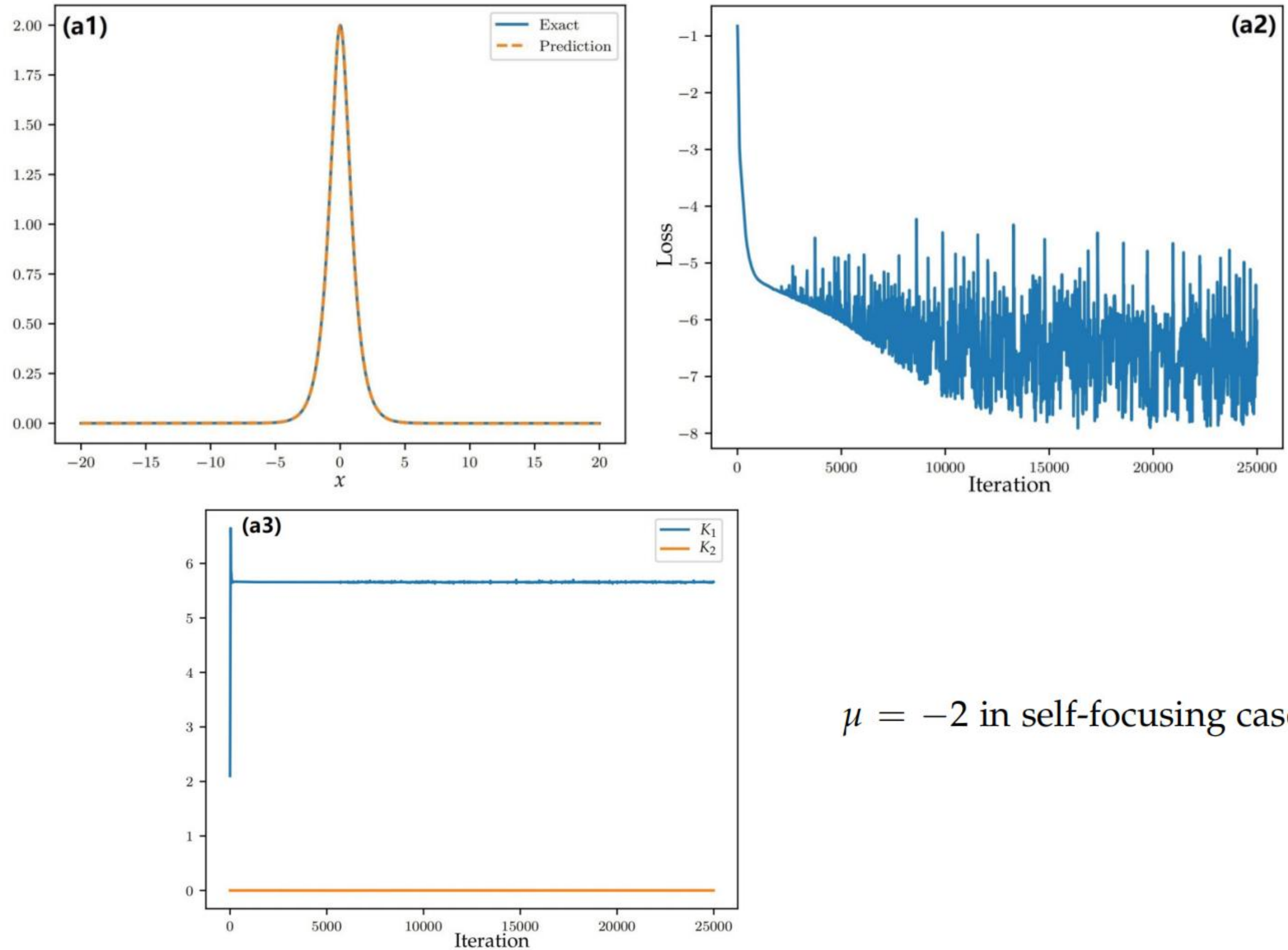
$$\omega_t = J_x,$$

$$(UU^*)_t = i(UU_x^* - U^*U_x)_x,$$

$$(UU_x^*)_t = i(UU_{xx}^* - U_xU_x^* - \frac{1}{2}gU^2U^{*2})_x,$$

$$K_1 = \int_{\mathbf{R}} \omega \mathrm{d}x = \int_{\mathbf{R}} UU^* \mathrm{d}x, \qquad K_2 = \int_{\mathbf{R}} \omega \mathrm{d}x = \int_{\mathbf{R}} UU_x^* \mathrm{d}x$$

Ⅲ、Applications



$\mu = -2$ in self-focusing case

Case 2.—Soliton solution of 1D NLS equation with complex potentials

PT-symmetric Scarf-II potential: $V(x) = V_{\rm re}(x) + iV_{\rm im}(x) = V_0 \,\mathrm{sech}^2(x) + iW_0 \,\mathrm{sech}(x) \tanh(x)$.

$$u(x) = \sqrt{-\frac{2 + V_0 + W_0^2/9}{g}}\,\mathrm{sech}(x) \exp\left[-\frac{iW_0}{3}\arctan(\sinh(x))\right],$$

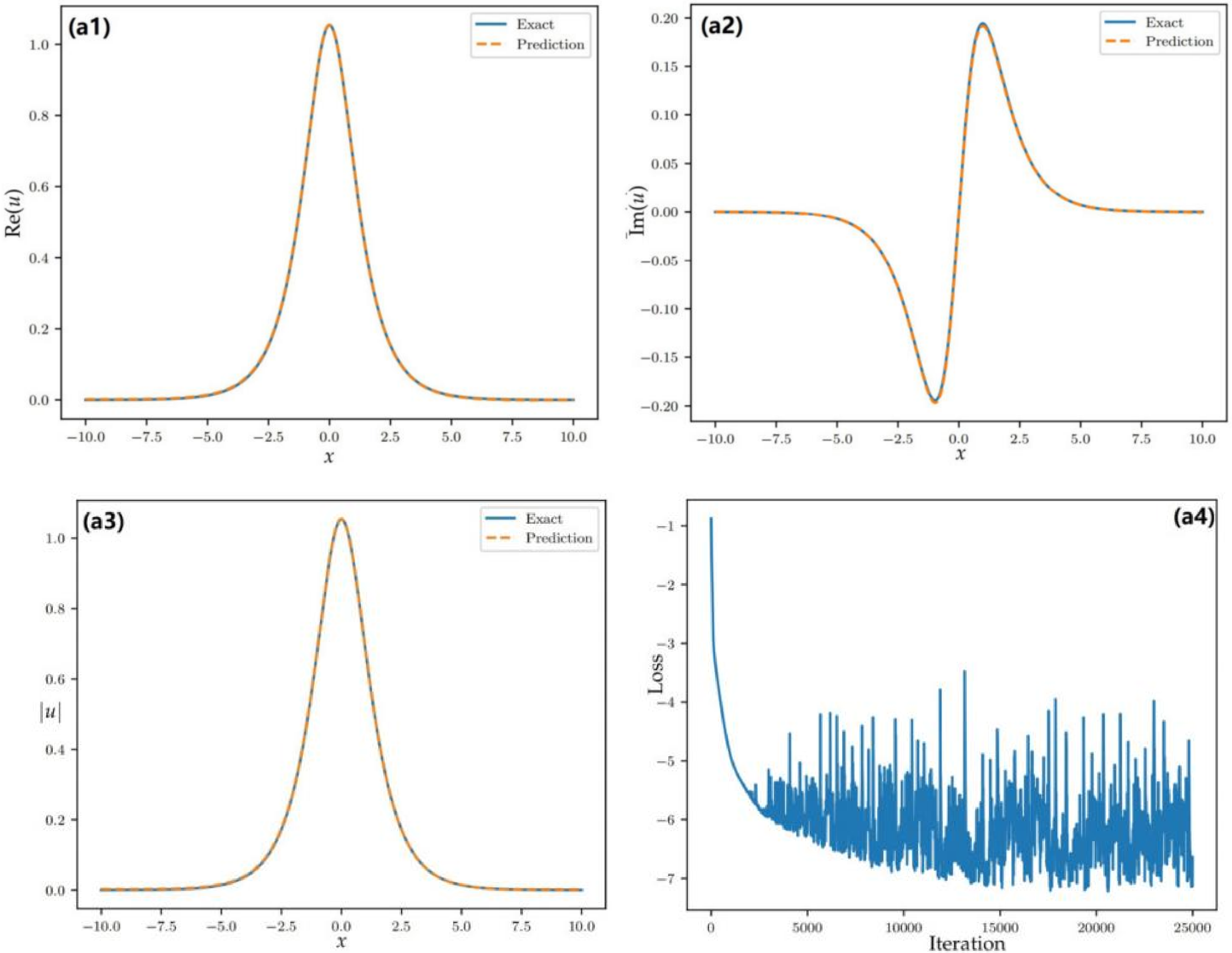
Considering that the solution is a complex-valued function,
in practical we set the network's output $\hat{u}(x) = p(x) + iq(x)$

$$\mathcal{F}_p(x) := -\partial_{xx}p + V_{\rm re}p - V_{\rm im}(x)q + g(p^2+q^2)p - \mu p,$$
$$\mathcal{F}_q(x) := -\partial_{xx}q + V_{\rm re}q + V_{\rm im}(x)p + g(p^2+q^2)q - \mu q.$$

$$\mathcal{L}_2 := \frac{1}{N}\frac{\sum_{i=1}^N \left(|\mathcal{F}_p(x_i)|^2 + |\mathcal{F}_q(x_i)|^2\right)}{\max_i\left(\sqrt{(p(x_i)^2 + q(x_i)^2}\right)}.$$

$$u_0(x) = \mathrm{sech}(x)e^{ix}.$$

Ⅲ、Applications



Example 1.2 (The solitary wave solution of KdV equation).

$$U_t + 6UU_x + U_{xxx} = 0.$$

Considering the traveling wave transform $\xi = x - ct$, then $U(x, t) = u(x - ct) = u(\xi)$

$$-c\frac{\mathrm{d}u}{\mathrm{d}\xi} + 6u\frac{\mathrm{d}u}{\mathrm{d}\xi} + \frac{\mathrm{d}^3u}{\mathrm{d}\xi^3} = 0,$$

We can integrate this with respect to ξ to obtain

$$-cu + 3u^2 + \frac{\mathrm{d}^2u}{\mathrm{d}\xi^2} = A,$$

where A is a constant of integration. Therefore we consider the nonlinear wave system

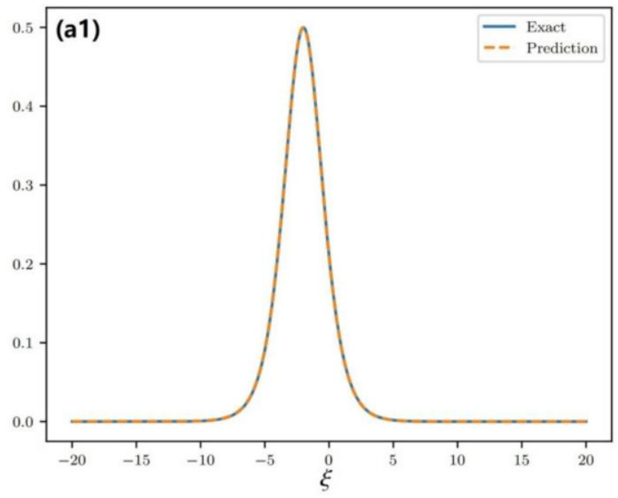
$$Lu - A = 0, \quad L = \frac{\mathrm{d}^2}{\mathrm{d}\xi^2} + 3u - c,$$

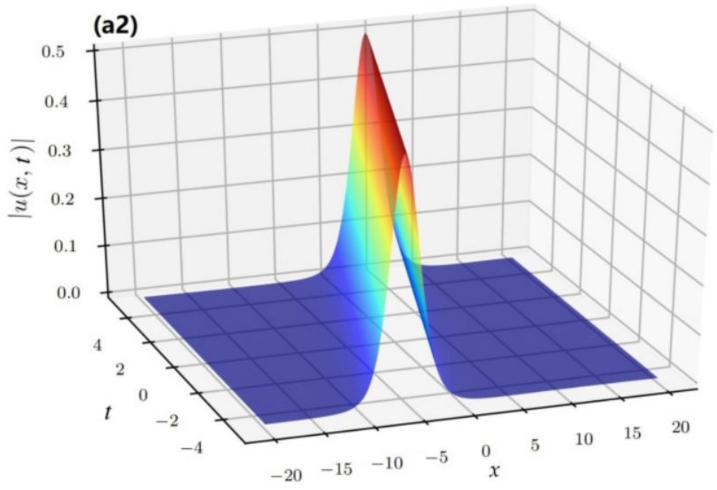
When $A = 0$, the solitary wave solution of the KdV equation can be found,

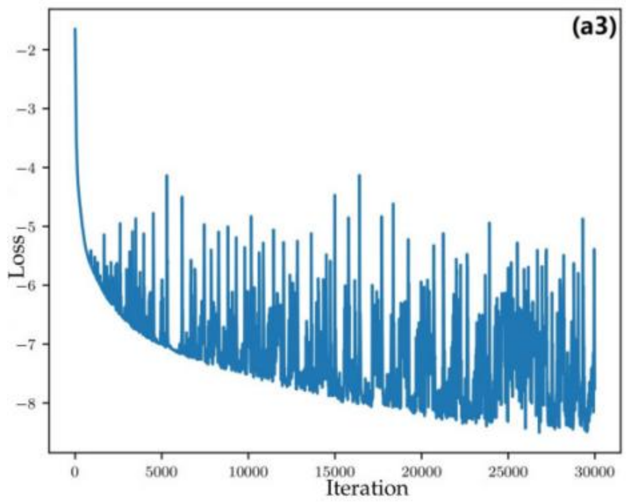
$$u(\xi) = \frac{1}{2}c\,\mathrm{sech}^2\left[\frac{\sqrt{c}}{2}(\xi + a)\right]$$

Ⅲ、Applications

Initial value $u_0(\xi) = \mathrm{sech}^2(\xi + 2)$

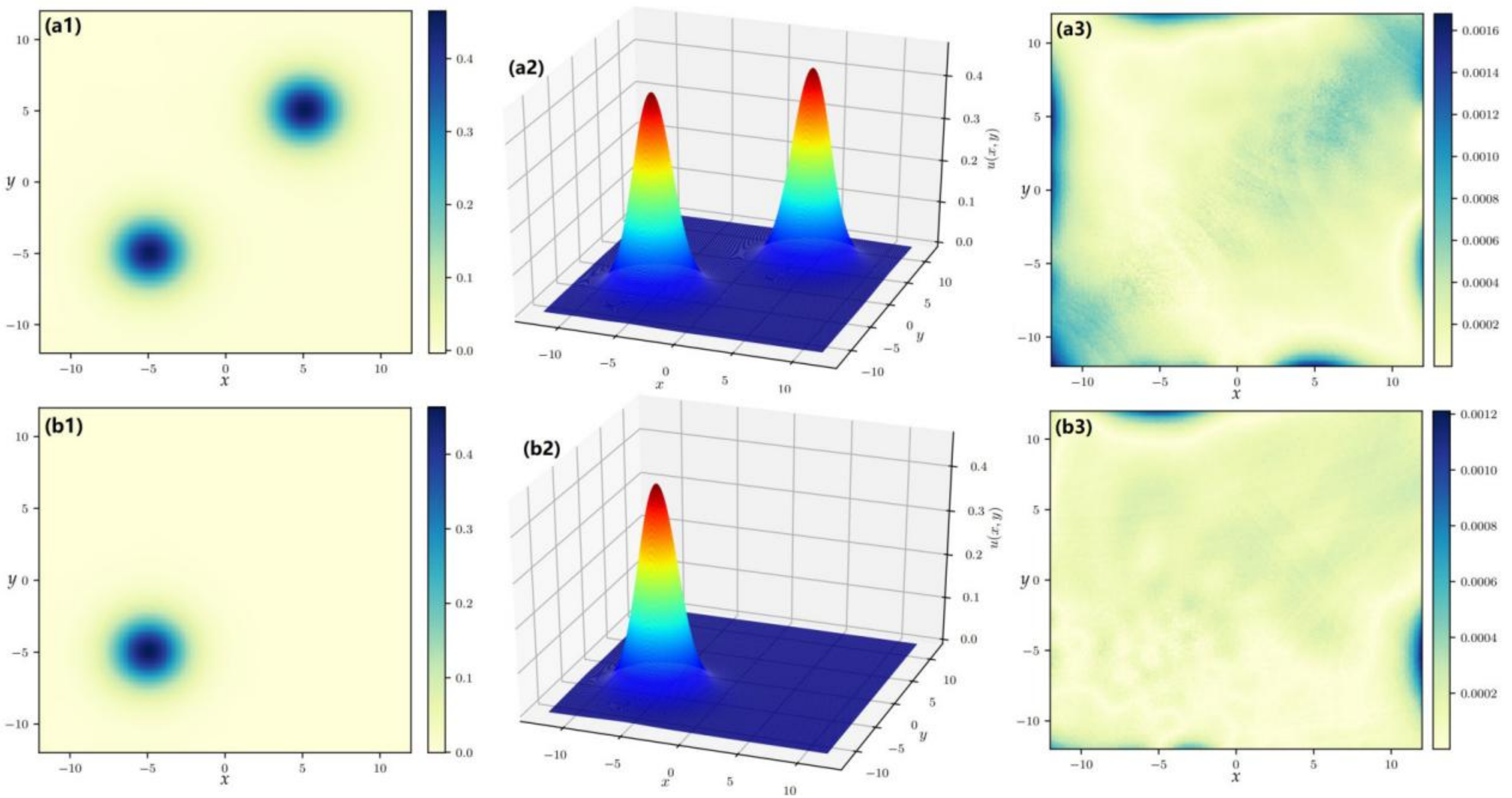






c = 1 and a = 2

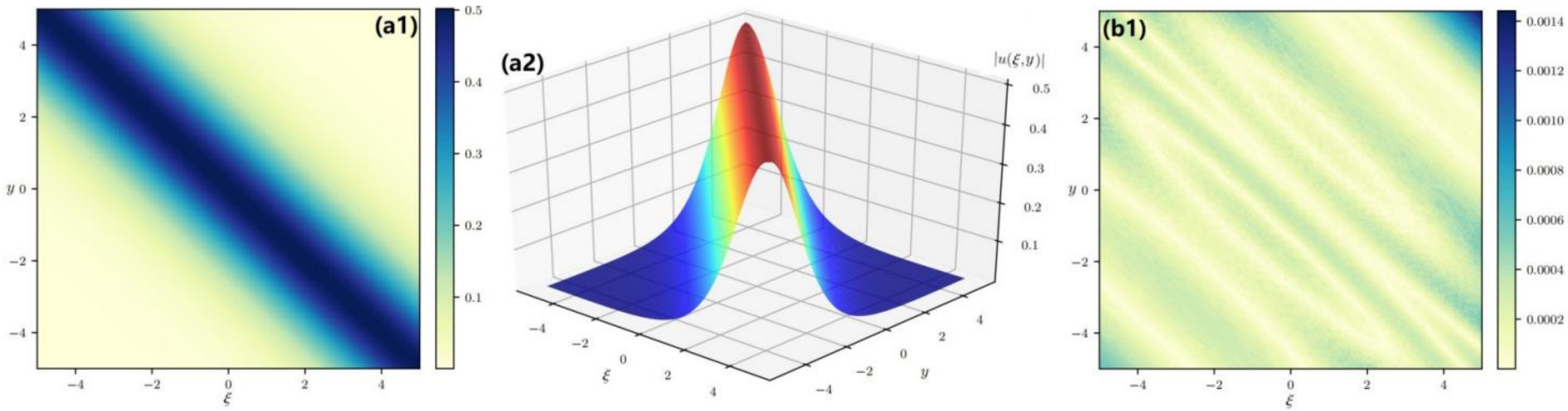
Example 1.3 (Quantum droplets of the 2D amended GP equation with LHY correction and multi-well potential).



Ⅲ、Applications

Example 1.4 (Solitary-wave solution of Kadomtsev-Petviashvili equation).

$$(U_t + 6UU_x + U_{xxx})_x + \alpha U_{yy} = 0, \qquad \alpha \in \mathbb{R}.$$



Example 1.5 (Optical bullets of 3D NLS equation with HO trapping potential).

$$iU_t - \Delta_3 U + V(x,y,z)U - |U|^2 U = 0,$$

$$V(x,y,z) = \frac{1}{2}(x^2 + y^2 + z^2).$$

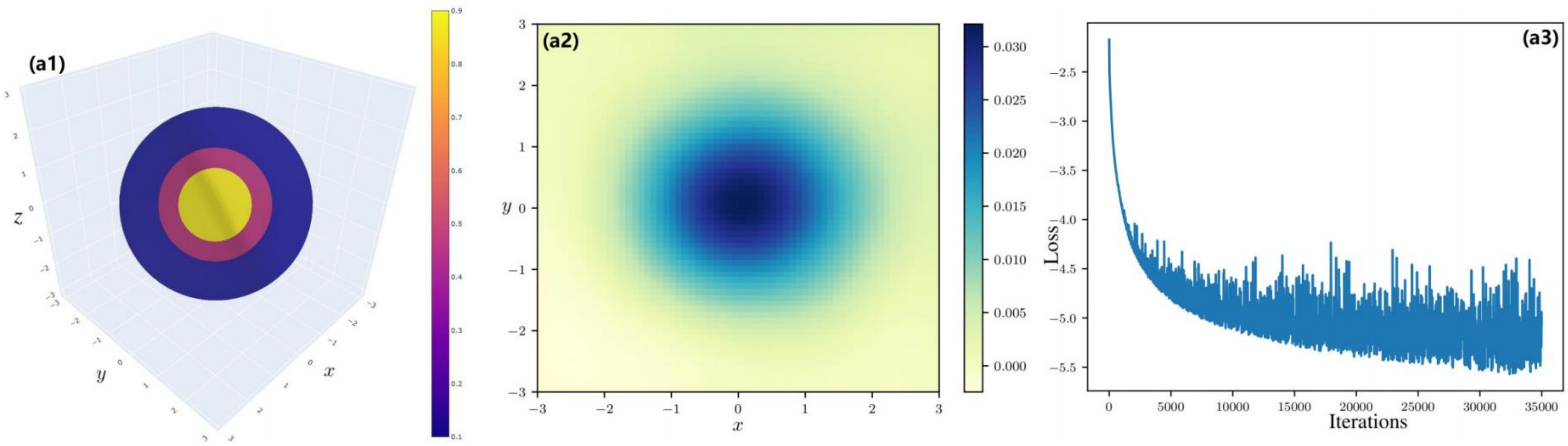
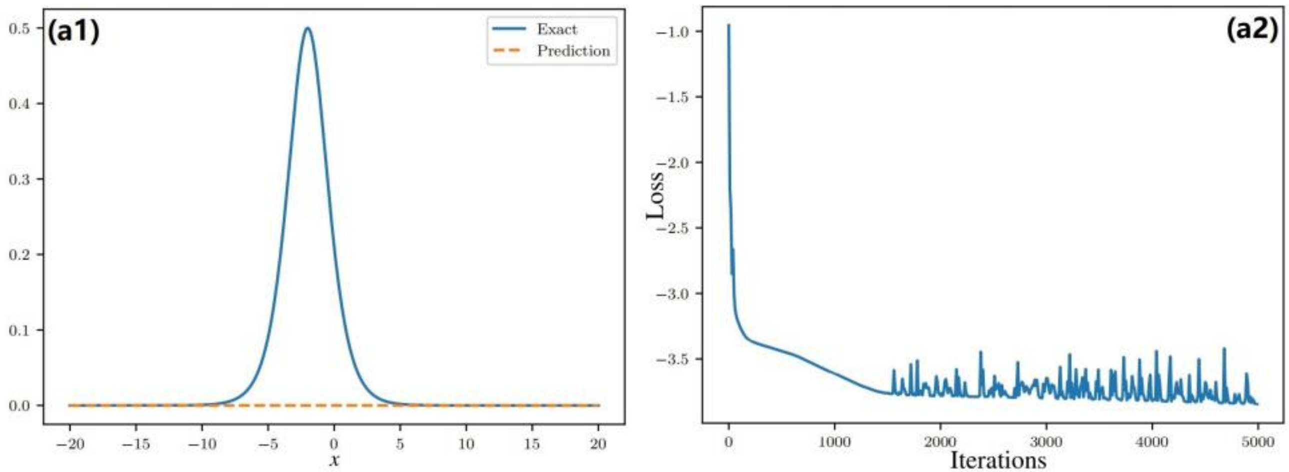


Table 1: The tested some examples and data via the IINN method.

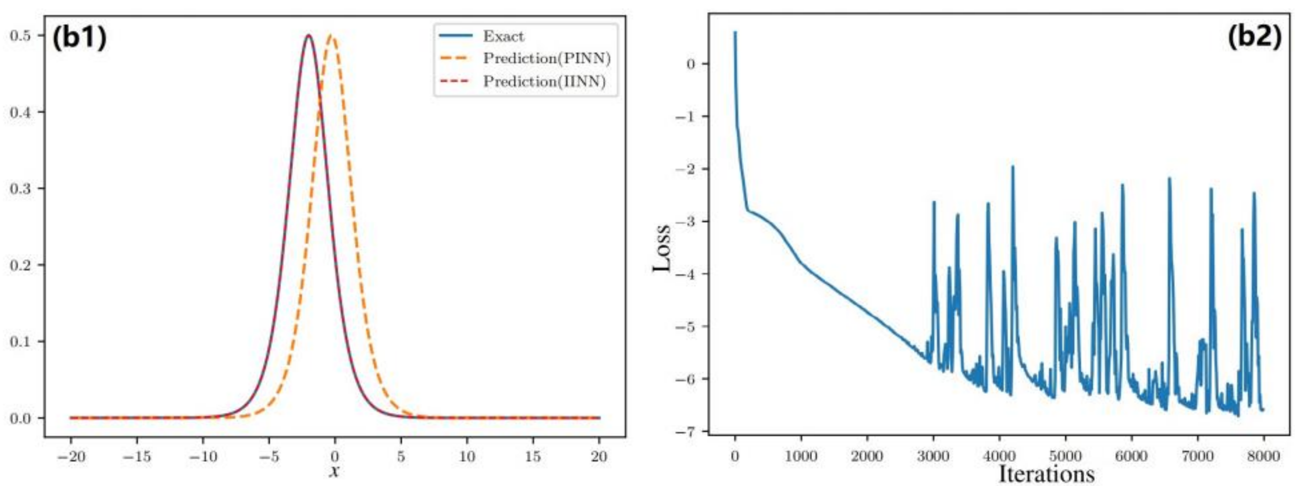
Equation	Potential	Solution	Initial value	Step (NN_1)	Step (NN_2)	N	E_1
1D NLS	/	Bright soliton	$u_0 = \text{sech}(x)$	10000	25000	500	1.63e-03
		Dark soliton	$u_0 = \tanh(x)$	10000	14000	500	3.40e-04
	HG	Ground state	$u_0 = \exp(-x^2)$	5000	30000	200	2.51e-04
		Dipole mode	$u_0 = 4x\exp(-x^2/2)$	10000	25000	200	4.66e-04
	$\mathcal{PT}$ Scarf-II	Soliton (focusing)	$u_0 = \text{sech}(x)\exp(ix)$	2000	25000	200	6.83e-04
		Soliton (defocusing)	$u_0 = \text{sech}(x)\exp(ix)$	2000	25000	200	5.08e-04
1D SNLS	$\mathcal{PT}$ OL	Gap soliton	$u_0 = \text{sech}(x)\cos(x)\exp(ix)$	15000	20000	800	4.00e-04
1D fdCNLS	/	Single-soliton	$\{u_{10}, u_{20}\} = \{2,1\}\text{sech}(x)$	5000	20000	500	1.56e-03
1D KdV	/	Solitary wave	$u_0 = \text{sech}^2(\xi + a)$	12000	30000	500	8.50e-04
2D NLS	HO	Ground state	$u_0 = e^{-0.5(x^2+y^2)}$	2000	20000	20000	4.62e-04
		Vortex soliton	$u_0 = 3re^{-0.5r^2}e^{i\phi}$	10000	20000	20000	1.83e-03
	Periodic	Gap soliton	$u_0 = \text{sech}\left(\sqrt{x^2+y^2}\right)\cos(x)\cos(y)$	20000	40000	20000	6.47e-03
2D GP	Quadruple-well	Branch B1	$u_0 = 0.3\left[e^{-0.1\lvert\mathbf{r}-\mathbf{r}_1\rvert^2} + e^{-0.1\lvert\mathbf{r}-\mathbf{r}_3\rvert^2}\right]$	15000	80000	20000	4.29e-03
		Branch A1	$u_0 = 0.46e^{-0.1\lvert\mathbf{r}-\mathbf{r}_3\rvert^2}$	10000	80000	20000	2.85e-03
2D KP	/	Solitary wave	$u_0 = \text{sech}^2(\xi + y)$	10000	50000	20000	8.35e-04
3D NLS	HO	Bullet	$u_0 = e^{-0.5(x^2+y^2+z^2)}$	10000	35000	40000	8.55e-03

IV、Discussions

- **Limitations of PINNs method**

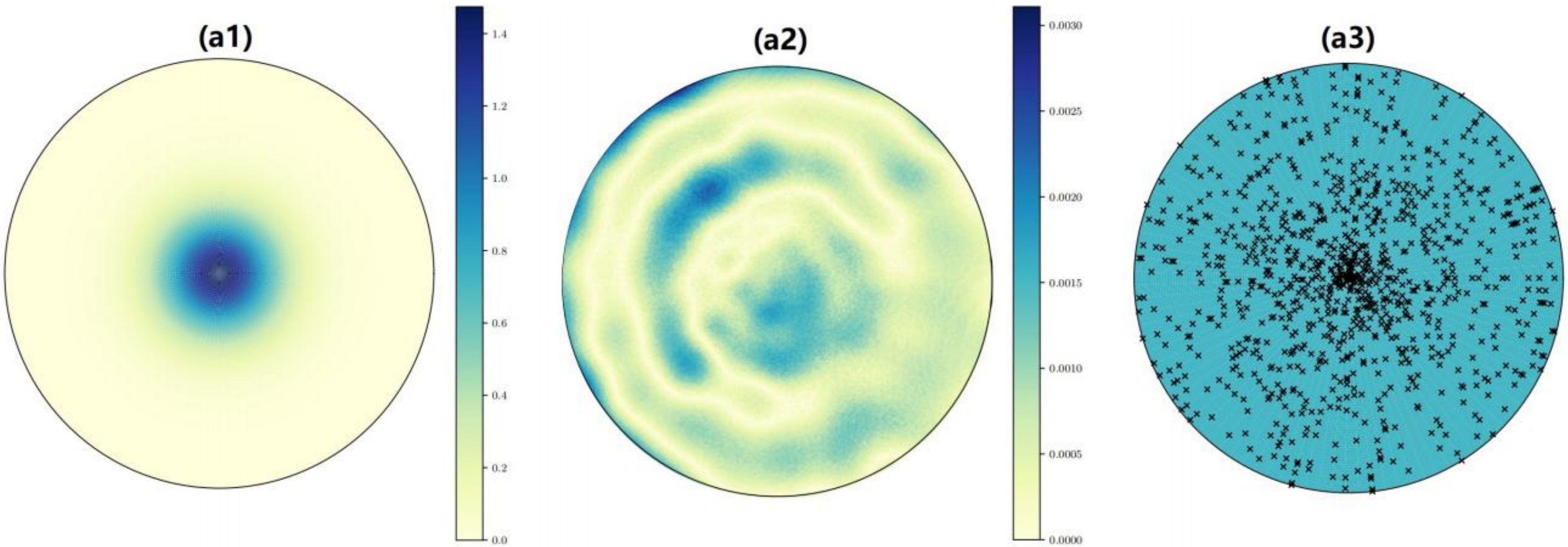


$$\mathcal{L}_3 := \frac{1}{N_f} \frac{\sum_{\ell=1}^{N_f} |\mathbf{L}\hat{\mathbf{u}}(\mathbf{x}_f^\ell)|^2}{\max(|\hat{\mathbf{u}}(\mathbf{x}_f^\ell)|)} + \frac{1}{N_b} \sum_{\ell=1}^{N_b} |\hat{\mathbf{u}}(\mathbf{x}_b^\ell)|^2,$$



IV、Discussions

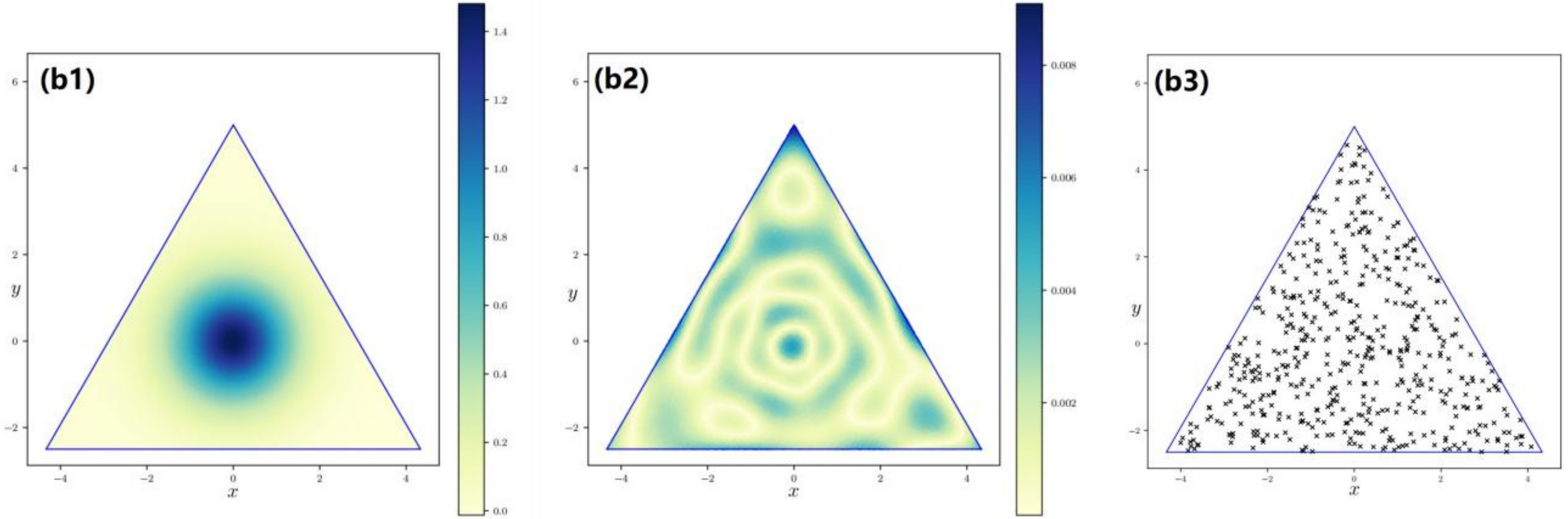
- **Advantages over traditional numerical methods**



The ground state solution u(x) of 2D NLS equation on the disk.

IV、Discussions

- **Advantages over traditional numerical methods**



The ground state solution u(x) of 2D NLS equation on the equilateral triangle region.

Thanks