Journal of Computational and Applied Mathematics 475 (2026) 116998

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A spectral Levenberg–Marquardt-Deflation method for multiple solutions of semilinear elliptic systems

Lin Li [a], Yuheng Zhou [a], Pengcheng Xie [b,*], Huiyuan Li [c]

[a] School of Mathematics and Physics, University of South China, Hengyang, China
[b] Applied Mathematics and Computational Research Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, 94720, CA, USA
[c] State Key Laboratory of Computer Science/Laboratory of Parallel Computing, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

MSC:
65N35
65N22
65F05
65L10

Keywords:
Multiple solutions
Legendre–Galerkin method
Levenberg–Marquardt method
Deflation

ABSTRACT

Numerous practical problems give rise to nonlinear differential equations that may exhibit multiple nontrivial solutions relevant to applications. Efficiently computing these solutions is crucial for a profound understanding of these problems and enhancing various applications. Therefore, the development of a numerical method capable of finding multiple solutions efficiently is imperative. Additionally, the provision of an efficient iteration process is vital for promptly obtaining multiple solutions. In the current paper, we introduce a novel algorithm for identifying multiple solutions of semilinear elliptic systems, where the trust region Levenberg–Marquardt method, combined with the deflation technique, is designed to compute multiple solutions for the first time. Based on several numerical experiments, our algorithm demonstrates efficacy in efficiently identifying multiple solutions, even when the nonlinear term appearing in these equations involves solely the first derivative. Moreover, we validate the efficiency of our algorithm and unveil previously undiscovered solutions in the existing literature

1. Introduction

Many nonlinear differential equations arising from practical problems permit multiple nontrivial solutions, which can be observed in physics, mechanics, biology, energy and engineering and so on. The in-depth study of multiple solutions is helpful to improve the understanding and application of these problems [1–3]. It is worth pointing out that most of these equations do not have explicit solutions. Moreover, the PDE theory only provides some powerful analytic solution techniques for special cases, such as the radial symmetrical case. Therefore, the development of efficient numerical methods to find multiple solutions will be very meaningful and is attracting the attention of many studies around the world.

Recently, in [4], we proposed a spectral trust-region deflation method for finding multiple solutions of a single nonlinear equation. For some coupled nonlinear equations with multiple solutions, the spectral trust-region deflation method presented in [4] does not seem to be very effective without any further improvement. To be specific, we identify multiple solutions to the following problem

$$-\Delta\vec{u} = \vec{G}(\boldsymbol{x}, \vec{u}), \qquad \boldsymbol{x} \in \Omega \tag{1.1}$$

* Corresponding author.
E-mail addresses: lilinmath@usc.edu.cn (L. Li), 15321363156@163.com (Y. Zhou), pxie@lbl.gov, pxie98@gmail.com (P. Xie), huiyuan@iscas.ac.cn (H. Li).

https://doi.org/10.1016/j.cam.2025.116998
Received 13 October 2024; Received in revised form 23 April 2025
Available online 9 August 2025

supplemented with some boundary conditions, where Ω is a bounded domain in $\mathbb{R}^d$, $d = 1, 2, \ldots$, and $\vec{G}$ is a nonlinear vector function of $\vec{u}$. Compared to identifying multiple solutions of a single nonlinear equation, efficiently identifying multiple solutions of (1.1) will be more challenging due to the substantial increase in the number of discrete unknown variables. For example, when considering the case $d = 2, \vec{u} := (u, v)$ with homogeneous Dirichlet boundary conditions, we assume that the numerical solution (denoted by $u_N(x, y)$ and $v_N(x, y)$) is expanded as

$$u_N(x,y) = \sum_{i,j=0}^{N} \tilde{u}_{ij}\phi_i(x)\phi_j(y) \quad \text{and} \quad v_N(x,y) = \sum_{i,j=0}^{N} \tilde{v}_{ij}\phi_i(x)\phi_j(y), \tag{1.2}$$

where $\phi_i(x) = L_i(x) - L_{i+2}(x)$, $L_i(x)(0 \le i \le N+2)$ are Legendre polynomials, and $\{\tilde{u}_{ij}\}$ and $\{\tilde{v}_{ij}\}$ are expanding coefficients to be solved. From (1.2), we need to solve $2(N+1)^2$ unknown variables. The computational complexity will increase dramatically with increasing N, indicating that identifying multiple solutions of (1.1) (or solving these unknown variables in (1.2)) will become more and more difficult. On the other hand, for certain coupled partial differential equations (PDEs) with multiple solutions, including the two types of noncooperative systems in [5], the function $\vec{G} \in C^1[-1, 1]$ defined in (1.1) is only satisfied under specific parameter conditions. The algorithms utilizing second derivative information cannot support multiple solutions of (1.1), including the spectral trust-region deflation method given in [4]. Therefore, in the current paper, our aim is to improve the spectral trust-region deflation method presented in [4], and to design a new numerical method for finding multiple solutions of (1.1) efficiently.

Before we introduce our approach, it is necessary to elaborate on the relevant background and motivations. In 1993, Choi and Mckenna [6] proposed the mountain pass algorithm (MPA) for multiple solutions of semilinear elliptic problems. Subsequently, in [7], Xie et al. pointed out that the MPA was feasible for finding two solutions of mountain pass type with Morse index 1 or 0. However, as said in [8], the MPA may fail to locate the sign-changing solutions, where Ding et al. proposed a high linking algorithm (HLA) for such solutions. In 2001, Li and Zhou [9] proposed the minimax algorithm (MNA) for finding multiple solutions of nonlinear equations, and more recent advancements are presented in [10–13]. It is worth pointing out that all of these methods above require that nonlinear differential equations have a variational structure, where the variational structure plays an important role in designing the algorithm to find multiple solutions. However, many differential equations with multiple solutions have no variational structure, which makes the methods above not applicable. This also leads to the second category of the existing methods for multiple solutions, i.e., some numerical methods (e.g., spectral method or finite difference) are used to discretize differential equations with multiple solutions. Then, some iterative methods are provided to find multiple solutions of the resulting nonlinear algebraic system (NLAS). Along this line, the search-extension method (SEM) was proposed by Xie et al. [14], and some improvements to SEM have been found in [15,16]. In [17,18], with the finite difference discretization, Allgower et al. proposed the homotopy continuation method for finding multiple solutions of the NLAS, and some interesting recent works have been inspired by this method, e.g. [19–21]. It is worth pointing out that in these methods, the Newtonian method is often chosen to iterate the NLAS with an initial guess. However, as we know, the main disadvantage of the Newtonian method is that the iteration is sensitive to the initial guess, and the inverse of the Jacobian matrix should exist. In [4], we first used the trust-region method to replace the Newtonian method for finding multiple solutions, where numerical results concerning computational efficiency have been greatly improved. This indicates that the trust-region method is worthy of extensive application in computing multiple solutions. However, if the trust-region method presented in [4] is directly used to iterate the NLAS arising from (1.1), the computational efficiency will be greatly affected due to the expensive computational cost of the Hessian matrix. More importantly, if the Hessian matrix does not exist, the trust-region method given in [4] cannot be used. Therefore, in the current paper, combined with the deflation technique, we will design a more efficient iteration and substantially improve the efficiency for finding multiple solutions of (1.1). To be specific, since the solution $\vec{u}$ considered in (1.1) is assumed to be smooth, the spectral Legendre–Galerkin method is mainly used to discretize (1.1) for high precision. To address the residual with only first-order derivative information, we first employ the Levenberg–Marquardt method, a trust-region algorithm, to iteratively solve the resulting NLAS and deflated system. Here, our method is dubbed the spectral trust region LM-Deflation method. Compared with the existing methods, the main differences and advantages of the spectral trust region LM-Deflation method reside in two respects:

- When the number of unknown variables to be solved increases significantly, the spectral trust region LM-Deflation method efficiently reduces computational complexity by simplifying the Hessian matrix. Furthermore, the method is applicable for solving (1.1) with low regularity (i.e., $\vec{G} \in C^1[a, b]$).
- In contrast to various existing methods, our method demonstrates the ability to efficiently and reliably converge to multiple solutions, which allows us to start even with the same initial guesses for multiple solutions based on the deflation technique.

The remainder of this paper is organized as follows. In Section 2, we describe the spectral trust region LM-Deflation method. In Section 3, numerical experiments are provided to demonstrate the efficiency of the method. In Section 4, we conclude the paper with some remarks.

2. An algorithm for computing multiple solutions

The main purpose of this section is to propose an efficient algorithm for computing multiple solutions of (1.1). The simple flow configuration for the algorithm is presented in Fig. 2.1, where the Spectral-Galerkin method is used to discretize (1.1), the trust region Levenberg–Marquardt method is designed to iterate the resulted NLAS and the deflated system, resulting in multiple solutions. For added clarity, the forthcoming section is divided into four parts: the first part — the Spectral-Galerkin method to (1.1), the second part — the trust region Levenberg–Marquardt method for iterating the resulted NLAS and the deflated system, the third part — the deflation technique for finding multiple solutions and the fourth part - a summary of our algorithm.

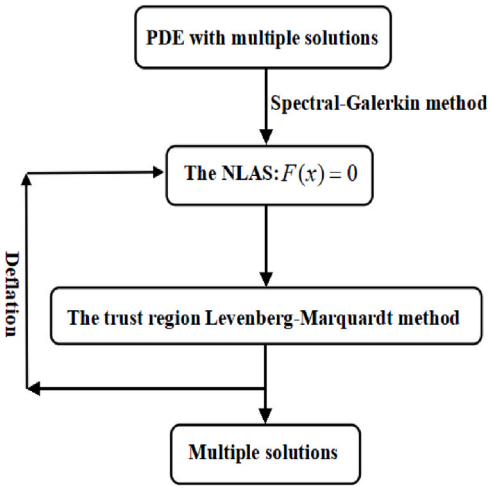


Fig. 2.1. The simple flow configuration of our algorithm.

2.1. The spectral-Galerkin method

We consider the following general system of partial differential equations

$$\mathcal{L}\boldsymbol{u} + \mathcal{N}\boldsymbol{u} = \boldsymbol{0}, \qquad \boldsymbol{x} \in \Omega, \tag{2.1}$$

where $\boldsymbol{u} := (u_1(\boldsymbol{x}), u_2(\boldsymbol{x}), \dots, u_n(\boldsymbol{x}))^\top$ is a vector function of $\boldsymbol{x}$, $\mathcal{L}$ and $\mathcal{N}$ are linear and nonlinear operator, respectively. Here the linear operator $\mathcal{L}$ may be $-\Delta$ or other operators, and in this paper we mainly focus on the former. The nonlinear operator $\mathcal{N} := (\mathcal{N}_1(\boldsymbol{u}), \mathcal{N}_2(\boldsymbol{u}), \dots, \mathcal{N}_n(\boldsymbol{u}))^\top$ is defined. When the basis functions $\{\phi_k\}_{k=0}^N$ satisfied corresponding boundary conditions are chosen, the resulting $\{u_i(\boldsymbol{x})\}_{i=1}^n$ can be expanded as follows

$$u_i(\boldsymbol{x}) = \sum_{j=0}^{N} a_j^i \phi_j(\boldsymbol{x}), \qquad 1 \le i \le n. \tag{2.2}$$

Based on the Spectral-Galerkin method, substituting (2.2) into (2.1) yields the resulting nonlinear algebraic system, i.e.,

$$\mathcal{A}\boldsymbol{a} + \mathcal{F}(\boldsymbol{a}) = \boldsymbol{0}, \tag{2.3}$$

where

$$\boldsymbol{a} = (a_0^1, a_1^1, \dots, a_N^1, a_0^2, \dots, a_N^2, \dots, a_0^n, \dots, a_N^n)^T,$$

$$[\boldsymbol{S}]_{ij} = -\int_\Omega \Delta\phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x})d\boldsymbol{x}, \qquad \mathcal{A} = \begin{bmatrix} \boldsymbol{S} & & & \\ & \ddots & & \boldsymbol{0} \\ & & \ddots & \\ & \boldsymbol{0} & & \boldsymbol{S} \end{bmatrix},$$

$$\mathcal{F}(\boldsymbol{a}) = \Big(\int_\Omega \mathcal{N}_1(\boldsymbol{u})\phi_0(\boldsymbol{x})d\boldsymbol{x}, \dots, \int_\Omega \mathcal{N}_1(\boldsymbol{u})\phi_N(\boldsymbol{x})d\boldsymbol{x}, \int_\Omega \mathcal{N}_2(\boldsymbol{u})\phi_0(\boldsymbol{x})d\boldsymbol{x}, \dots, \int_\Omega \mathcal{N}_n(\boldsymbol{u})\phi_N(\boldsymbol{x})d\boldsymbol{x}\Big)^\top.$$

Here it is worth pointing out that $\boldsymbol{a}$ in (2.3) is the unknown vector to be solved. To increase the clarity of the above process, as an illustrative example, we consider (2.1) with homogeneous Dirichlet boundary conditions, $d = 2$ (i.e. $\Omega = (x, y)$), $\boldsymbol{u} = (u_1(x, y), u_2(x, y))$ and $\mathcal{N} = (\mathcal{N}_1(\boldsymbol{u}), \mathcal{N}_2(\boldsymbol{u}))$, and the Legendre–Galerkin method is used to discretize it. Let $\mathbb{P}_N$ be the set of the polynomials of degree at most N, and $\mathbb{P}_N^0 = \{\phi \in \mathbb{P}_N \ : \ \phi(\pm 1) = 0\}$. The Legendre–Galerkin approximation is to find $(u_N^1, u_N^2) \in X_N^0 = (\mathbb{P}_N^0)^2$ such that

$$\begin{cases} (\nabla u_N^1, \nabla\phi_N) = (I_N\mathcal{N}_1, \phi_N), \\ (\nabla u_N^2, \nabla\phi_N) = (I_N\mathcal{N}_2, \phi_N), \end{cases} \tag{2.4}$$

for $\forall \phi_N \in \mathbb{P}_N^0$, where I_N is the Legendre–Gauss–Lobatto tensorial interpolation operator with $N+1$ points in each coordinate direction. Based on the property of the Legendre polynomial [22], we introduce the basis of $\mathbb{P}_N^0$ as

$$\phi_k(x) = L_k(x) - L_{k+2}(x), \qquad 0 \le k \le N,$$

and write

$$\begin{aligned}
&u_N^1 = \sum_{k,j=0}^{N} a_{kj}^1 \phi_k(x)\phi_j(y), && u_N^2 = \sum_{k,j=0}^{N} a_{kj}^2 \phi_k(x)\phi_j(y),\\
&\hat{a}_{kj} = \int_I \phi_j'(x)\phi_k'(x)dx, && A = (\hat{a}_{kj})_{k,j=0,1,\ldots,N},\\
&b_{kj} = \int_I \phi_j(x)\phi_k(x)dx, && B = (b_{kj})_{k,j=0,1,\ldots,N},\\
&\mathcal{F}_{1,kj} = (I_N\mathcal{N}_1, \phi_k(x)\phi_j(y)), && \mathcal{F}_{2,kj} = (I_N\mathcal{N}_2, \phi_k(x)\phi_j(y)).
\end{aligned}$$

$\boldsymbol{S}, \boldsymbol{a}$ and $\mathcal{F}$ in (2.3) become

$$\boldsymbol{S} = A \otimes B + B \otimes A^{\top},$$

$$\boldsymbol{a} = (a_{00}^1, a_{10}^1, \ldots, a_{N0}^1, a_{01}^1, \ldots, a_{N1}^1, \ldots, a_{NN}^1, a_{00}^2, \ldots, a_{NN}^2)^{\top},$$

$$\mathcal{F} = (\mathcal{F}_{1,00}, \mathcal{F}_{1,10}, \ldots, \mathcal{F}_{1,N0}, \ldots, \mathcal{F}_{1,NN}, \mathcal{F}_{2,00}, \mathcal{F}_{2,10}, \ldots, \mathcal{F}_{2,N0}, \ldots, \mathcal{F}_{2,NN})^{\top}.$$

Here $\otimes$ denotes the tensor product operator, i.e. $A \otimes B = (Ab_{ij})_{i,j=0,1,\ldots,N}$. It is worth pointing out that when N is increased, (2.3) (or (2.4)) will become a large-scale nonlinear system, indicating that solving it will be more challenging. Moreover, when the nonlinear operator in (2.1) is only first-order differentiable with respect to $\boldsymbol{u}$, $\mathcal{F}$ in (2.3) is also only first-order differentiable with respect to $\boldsymbol{a}$. In other words, algorithms with second-order derivative information fail to solve (2.3). Motivated by [4], the trust region Levenberg–Marquardt method is designed to overcome these difficulties for the first time. In addition, in the computational process, we only need to evaluate $\mathcal{F}_{1,kj}$ and $\mathcal{F}_{2,kj}$ for given $\boldsymbol{u}$ that can be implemented efficiently by the pseudo-spectral technique described in [22, Ch. 4] in terms of coefficients.

2.2. The trust region Levenberg–Marquardt method

Firstly, we rewrite (2.3) in a more elegant form, i.e.,

$$\boldsymbol{F} = \left(F_1,\ F_2,\ \cdots,\ F_n\right)^T \tag{2.5}$$

with the unknown vector $\boldsymbol{a} = (a_1, \ldots, a_n)^T$, where $F_i\ (i = 1, \ldots, n)$ are functions of the unknown vector $\boldsymbol{a}$. Here our aim is to find $\boldsymbol{a}$ to satisfy $\boldsymbol{F}(\boldsymbol{a}) \equiv \boldsymbol{0}$. Motivated by [4], we reformulate the zero-finding problem (2.5) as a nonlinear optimization problem using the nonlinear least-squares method:

$$\min_{\boldsymbol{a}\in\mathbb{R}^n} Q(\boldsymbol{a}), \quad Q(\boldsymbol{a}) := \frac{1}{2}\left\|\boldsymbol{F}(\boldsymbol{a})\right\|_2^2 = \frac{1}{2}\sum_{i=1}^{n} F_i^2(\boldsymbol{a}). \tag{2.6}$$

Before presenting an efficient method to solve (2.6), we should introduce the Jacobian matrix, the gradient and Hessian matrix as follows:

$$\boldsymbol{J}(\boldsymbol{a}) = \boldsymbol{F}'(\boldsymbol{a}) = (\nabla F_1(\boldsymbol{a}), \nabla F_2(\boldsymbol{a}), \ldots, \nabla F_n(\boldsymbol{a}))^T,$$

$$\boldsymbol{g}(\boldsymbol{a}) = \nabla Q(\boldsymbol{a}) = \boldsymbol{J}^T(\boldsymbol{a})\boldsymbol{F}(\boldsymbol{a}) \quad \text{and} \quad \boldsymbol{G}(\boldsymbol{a}) = \nabla^2 Q(\boldsymbol{a}) = \boldsymbol{J}^T(\boldsymbol{a})\boldsymbol{J}(\boldsymbol{a}) + \boldsymbol{S}(\boldsymbol{a}),$$

where $\boldsymbol{S}(\boldsymbol{a}) = \sum_{i=1}^n F_i(\boldsymbol{a})\nabla^2 F_i(\boldsymbol{a})$. so far, in term of updating the unknown vector $\boldsymbol{a}$ in (2.6), there may exist three formats as follows:

$$\begin{aligned}
&\textbf{(I)} \quad \boldsymbol{a}_{k+1} = \boldsymbol{a}_k - \boldsymbol{G}(\boldsymbol{a}_k)^{-1}\boldsymbol{g}(\boldsymbol{a}_k), && \text{(Newtonian method)}\\
&\textbf{(II)} \quad \boldsymbol{a}_{k+1} = \boldsymbol{a}_k - (J(\boldsymbol{a}_k)^T J(\boldsymbol{a}_k))^{-1}\boldsymbol{g}(\boldsymbol{a}_k), && \text{(Gauss–Newtonian method)}\\
&\textbf{(III)} \quad \boldsymbol{a}_{k+1} = \boldsymbol{a}_k - (J(\boldsymbol{a}_k)^T J(\boldsymbol{a}_k) + \mu_k I)^{-1}\boldsymbol{g}(\boldsymbol{a}_k), && \text{(Levenberg–Marquardt method)}
\end{aligned} \tag{2.7}$$

where $\mu_k \ge 0$ is a parameter being updated from iteration to iteration. Obviously, the main disadvantage of the Newtonian method is that the second-order term $\boldsymbol{S}(\boldsymbol{a})$ in the Hessian matrix $\boldsymbol{G}(\boldsymbol{a})$ is difficult or expensive to compute. Further, as said in the introduction above, the Hessian matrix $\boldsymbol{G}(\boldsymbol{a})$ does not exist in some situations, which means that the Newtonian method cannot be used to solve (2.6). When ignoring $\boldsymbol{S}(\boldsymbol{a})$, we use the format **II** to solve (2.6), where it is also required that Jacobian matrix $\boldsymbol{J}(\boldsymbol{a})$ has full column rank. In other words, if $\boldsymbol{J}(\boldsymbol{a})$ is rank-deficient, then the format **II** cannot work well. In addition, based on our experience from [4], we have realized that computing a full Hessian matrix can reduce computational efficiency. Therefore, here we use the Levenberg–Marquardt method to solve (2.6) in the process of seeking multiple solutions. Let

$$\boldsymbol{s}_k := \boldsymbol{a}_{k+1} - \boldsymbol{a}_k = -(\boldsymbol{J}(\boldsymbol{a}_k)^T\boldsymbol{J}(\boldsymbol{a}_k) + \mu_k I)^{-1}\boldsymbol{g}(\boldsymbol{a}_k), \tag{2.8}$$

then it is easy to verify that the Levenberg–Marquardt step s_k in (2.8) is the solution of the following optimization problem

$$\min_{s\in R^n} \|F(a_k) + J(a_k)s\|_2^2 + \mu_k\|s\|_2^2. \tag{2.9}$$

Meanwhile, the Levenberg–Marquardt method is also one of the trust-region methods. To be specific, the following trust-region model is considered:

$$\min_{s\in\mathbb{B}_{h_k}} q^{(k)}(s) := Q(a_k) + g(a_k)^T s + \frac{1}{2}s^T J(a_k)^T J(a_k)s, \tag{2.10}$$

where Δ_k represents the trust-region radius and the trust region $\mathbb{B}_{h_k} := \{s := x - x_k \in \mathbb{R}^n : \|s\|_2 \le \Delta_k = \|s_k\|_2\}$[23–26]. Then we can derive that s_k in (2.8) is the global optimal solution of (2.10) [27]. Moreover, for the trust-region method, the global convergence with a local superlinear rate of convergence is as follows:

Theorem 2.1 (*See [27]*)**.** *Assume that*

(i) *the function $Q(x)$ is bounded below on the level set*

$$S := \{x \in \mathbb{R}^n : Q(x) \le Q(x^{(0)})\}, \quad \forall x^{(0)} \in \mathbb{R}^n, \tag{2.11}$$

and is Lipschitz continuously differentiable in S;

(ii) *the Hessian matrices $G(x^{(k)})$ are uniformly bounded in 2-norm, i.e., $\|G(x^{(k)})\| \le \beta$ for any k and some $\beta > 0$.*

If $g(x^{(k)}) \ne 0$, then

$$\liminf_{k\to\infty} \|g(x^{(k)})\| = 0. \tag{2.12}$$

Moreover, if $g(x^*) = 0$, and $G(x^*)$ is positive definite, then the convergence rate of the trust-region method is quadratic.

In addition, it is worth pointing out that the main difference between the Levenberg–Marquardt method (i.e. (2.10)) and other trust region methods is that other trust region methods update the trust region radius Δ_k directly, while the Levenberg–Marquardt method updates the parameter μ_k, which in turn modifies the value Δ_k implicitly. Moreover, we can see that $(J(a_k)^T J(a_k) + \mu_k I)$ in (2.7) is also positive definite, indicating that its inverse can be well defined. When μ_k in (2.7) is changed, the Levenberg–Marquardt method allows choosing any direction between the Gauss–Newtonian direction and the steepest descent direction, i.e., when $\mu_k = 0$, it reduces to the Gauss–Newtonian direction. While μ_k is large enough, the produced direction is close to the steepest descent direction. Furtherly, in [27], Yuan et al. have described the relation between s and μ, i.e., let $s = s(\mu)$ is a solution of

$$(J(a)^T J(a) + \mu I)s = -g(a), \tag{2.13}$$

Then we can conclude the following theorem.

Theorem 2.2. *When μ increases from zero, $\|s(\mu)\|_2$ in (2.10) will decrease strictly monotonically.*

To obtain a proper trust-region radius Δ_k for determining s_k and μ_k, we define

$$r_k = \frac{Q(a_k) - Q(a_{k+1})}{q^{(k)}(0) - q^{(k)}(s_k)} \tag{2.14}$$

for $q^{(k)}(s_k)$ and the objective function value $Q(a_{k+1})$. (2.14) is viewed as an indicator for the expansion and contraction of the trust region Δ_k. If r_k is close to 1, it means there is good agreement, we can expand the trust-region radius for the next iteration. Based on the relation between s_k and μ_k in Theorem 2.2, we should reduce μ_{k+1} for this case, and choose $\mu_{k+1} = 0.1\mu_k$. If r_k is close to zero or negative, the trust-region radius should be shrunk, indicating the corresponding value of μ_{k+1} should be increased, and $\mu_{k+1} = 10\mu_k$ is chosen. Otherwise, we do not alter the trust-region radius.

Next, to the best of our knowledge, we will present the results on the convergence of the Levenberg–Marquardt method in the existing literature. For the convenience of subsequent presentation, the definition of the local error bound should be introduced as follows:

Definition of the local error bound. Let X^* be the solution set of (2.5) and $a^* \in X^*$, if there exists a positive constant $c > 0$ such that

$$\|F(a)\|_2 > c\,\mathrm{dist}(a, X^*), \qquad \forall a \in N(a^*, b) = \{a \mid \|a - a^*\|_2 \le b\} \cap X^* \ne \emptyset, \tag{2.15}$$

where

$$\mathrm{dist}(a, X^*) = \inf_{y\in X^*} \|a - y\|_2.$$

Then, we call that $F(a)$ is a local error bound on some neighborhood of $a^* \in X^*$. As said in [28], if the Jacobian matrix is nonsingular at the solution a^* of (2.5) and if the initial guess is chosen sufficiently close to a^*, then the Levenberg–Marquardt method has a quadratic convergence. To improve the requirement of the nonsingularity, Yamashita and Fukushima [29] show that under the local error bound condition, if $\mu_k = \|F(a_k)\|_2^2$ and if the initial guess is chosen sufficiently close to the solution set X^*, the Levenberg–Marquardt method converges quadratically to the solution set X^*. In [30], Fan et al. considered $\mu_k = \|F(a_k)\|_2^\delta$ with $\delta \in [1, 2]$ and

proved that the Levenberg–Marquardt method still achieves the quadratic convergence under the same conditions. In [28], the cubic convergence of the modified Levenberg–Marquardt method was also proved under the local error bound condition. Finally, some progressive and interesting studies can be seen in [31].

2.3. The deflation technique for finding multiple solutions

In fact, the deflation technique is used to find multiple solutions of (2.5). Its main idea is to successively modify (2.5) under consideration to eliminate known solutions, and discover additional distinct solutions. We now state the deflation process for computing multiple solutions to the nonlinear algebraic system (2.5).

Step 1. Give an initial guess $\boldsymbol{a}^0 \in R^n$;
Step 2. Based on the Levenberg–Marquardt method, a solution of (2.5) is computed with the initial guess $\boldsymbol{a}^0$, and the solution is denoted by $\boldsymbol{r}_1$;
Step 3. A deflation operator is defined as

$$\Psi(\boldsymbol{a};\boldsymbol{r}_1) = \left(\frac{1}{\|\boldsymbol{a}-\boldsymbol{r}_1\|_q^p} + \alpha\right)\mathbb{I}, \tag{2.16}$$

where $\mathbb{I}$ is the identity operator, $\alpha \geq 0$ is a shift scalar, $p \in [1,\infty)$ is the deflation exponent.
The norm $\|\cdot\|_q$ should be chosen for the solution space, and here we choose $q = 2$. A deflated system $\hat{\boldsymbol{F}}(\boldsymbol{a})$ can be formed by applying (2.16) to (2.5) as

$$\hat{\boldsymbol{F}}(\boldsymbol{a}) = \Psi(\boldsymbol{a};\boldsymbol{r}_1)\boldsymbol{F}(\boldsymbol{a}); \tag{2.17}$$

Step 4. With the same initial guess $\boldsymbol{a}^0$, the deflated system $\hat{\boldsymbol{F}}(\boldsymbol{a})$ is solved using the Levenberg-Marquardt method, and the resulting solution is denoted by $\boldsymbol{r}_2$;
Step 5. A multiple deflation operator needs to be defined as follows:

$$\Psi(\boldsymbol{a};\boldsymbol{r}_1,\boldsymbol{r}_2) = \prod_{i=1}^{2}\Psi(\boldsymbol{a};\boldsymbol{r}_i), \tag{2.18}$$

and a new deflated system $\hat{\hat{\boldsymbol{F}}}(\boldsymbol{a})$ can be obtained:

$$\hat{\hat{\boldsymbol{F}}}(\boldsymbol{a}) = \Psi(\boldsymbol{a};\boldsymbol{r}_1,\boldsymbol{r}_2)\boldsymbol{F}(\boldsymbol{a}); \tag{2.19}$$

Step 6. Similar to **Step 4**, and the resulting solution is denoted by $\boldsymbol{r}_3$;
Step 7. Continue the cycle from Step 5 to Step 6, multiple solutions can be obtained, and these solutions are denoted by $X^* = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots\}$.

Remark 2.1. To have a better understanding and application of the deflation, we make some necessary remarks as follows:

In Step 3, the main purpose of introducing the deflation operator (2.16) is to find another distinct solution. To be specific, based on the results in [4,32,33], the deflated system $\hat{\boldsymbol{F}}(\boldsymbol{a})$ satisfies two properties: (i) The preservation of solutions of $\hat{\boldsymbol{F}}(\boldsymbol{a})$ should be hold, i.e. for $\boldsymbol{a} \neq \boldsymbol{r}_1$, $\hat{\boldsymbol{F}}(\boldsymbol{a}) = \boldsymbol{0}$ iff $\boldsymbol{F}(\boldsymbol{a}) = \boldsymbol{0}$; (ii) The Levenberg–Marquardt method applied to $\hat{\boldsymbol{F}}(\boldsymbol{a})$ will not find $\boldsymbol{r}_1$ again due to the deflation operator, i.e.,

$$\liminf_{\boldsymbol{a}\to\boldsymbol{r}_1} \|\hat{\boldsymbol{F}}(\boldsymbol{a})\|_q > 0,$$

indicating that with the deflation operator, another distinct solution can be obtained, even with the same initial guess. In addition, for simplicity, in the current paper $p = 2$ and $\alpha = 1$ in (2.16) are chosen.

In Step 5, it is worth mentioning that when we consider more than two solutions, the multiple deflation operator is constructed, which may increase the computational complexity in the nonlinear iteration. The single deflation operator with the initial guess presented in Step 4 may be considered to replace the multiple deflation operator, which is helpful in reducing the computational complexity.

2.4. Summary of the spectral Levenberg–Marquardt-Deflation algorithm

In summary, based on the methods above, a new and novel algorithm for computing multiple solutions of (1.1) can be designed, which is presented in the following **Algorithm 1**.

Algorithm 1 - An algorithm of computing multiple solutions of (1.1)

Input: $0 < \epsilon \ll 1$, $0 < \delta_1 < \delta_2 < 1$, $\mu_0 = 0.01$, and initial solution set $S \Leftarrow$ the empty set Φ.
Output: S

1: Given nonlinear discrete system $\boldsymbol{F}(\boldsymbol{a})$ by the Legendre–Galerkin method in the section 2.1.
2: **while** multiple solutions not reached **do**
3: **while** first-order optimality threshold or failure criterion not met **do**
4: Compute $\boldsymbol{J}(\boldsymbol{a}_k)$ and $\boldsymbol{g}(\boldsymbol{a}_k)$; ▷ Levenberg–Marquardt iteration in section 2.2.
5: If $\|\boldsymbol{g}(\boldsymbol{a}_k)\| < \epsilon$ and $|Q(\boldsymbol{a}_k)|^{1/2} < \epsilon$, stop;
6: Approximately solve $\boldsymbol{s}_k$ by (2.8);
7: Compute r_k by (2.14);
8: If $r_k \geq \delta_1$, then $\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + \boldsymbol{s}_k$; Otherwise, set $\boldsymbol{a}_{k+1} = \boldsymbol{a}_k$;
9: If $r_k < \delta_1$, then $\mu_{k+1} := 10\mu_k$;
If $\delta_1 \leq r_k \leq \delta_2$, then $\mu_{k+1} := \mu_k$;
Otherwise, set $\mu_{k+1} := 0.1\mu_k$;
10: Update $q^{(k)}$, set $k := k + 1$, go to step 4;
11: end
12: **if** Convergence threshold met **then**
13: Add converged solution $\boldsymbol{r}$ to S;
14: $\hat{\boldsymbol{F}}(\boldsymbol{a}) \leftarrow \Psi(\boldsymbol{a}; \boldsymbol{r})\boldsymbol{F}(\boldsymbol{a})$ ▷ the deflation technique presented in section 2.3.
15: Go back to step 3;
16: **end**
17: **end**
18: return S

Some additional remarks about the implementation of the **Algorithm 1** are listed, and the corresponding numerical tests will be conducted in the forthcoming section.

- For the Levenberg–Marquardt iteration (see lines 3-11), if both $\boldsymbol{F}$ and its Jacobi $\boldsymbol{J}$ are Lipschitz continuous, i.e., there exist positive constants L_1 and L_2 such that

$$\|\boldsymbol{F}(\boldsymbol{y}) - \boldsymbol{F}(\boldsymbol{x})\|_2 \leq L_2\|\boldsymbol{y} - \boldsymbol{x}\|_2, \qquad \forall \boldsymbol{x}, \boldsymbol{y}, \tag{2.20}$$

and

$$\|\boldsymbol{J}(\boldsymbol{y}) - \boldsymbol{J}(\boldsymbol{x})\|_2 \leq L_1\|\boldsymbol{y} - \boldsymbol{x}\|_2, \qquad \forall \boldsymbol{x}, \boldsymbol{y}. \tag{2.21}$$

Let $\boldsymbol{s}_k$ be computed by (2.8), then the inequality

$$Q_k(0) - Q_k(\boldsymbol{s}_k) \geq \|\boldsymbol{g}_k\|_2 \min\{\|\boldsymbol{s}_k\|_2, \frac{\|\boldsymbol{g}_k\|_2}{\|\boldsymbol{J}_k^T\boldsymbol{J}_k\|_2}\} \tag{2.22}$$

holds for all $k \geq 1$. Moreover, the sequence generated by the **Algorithm 1** satisfies

$$\lim_{k\to\infty} \|\boldsymbol{g}_k\|_2 = 0. \tag{2.23}$$

For the local convergence, $\{\boldsymbol{a}_k\}$ will converge to $\boldsymbol{a}^*$ when $\boldsymbol{F}$ satisfies the local error bound given in (2.15). In addition, when the conditions of convergence above cannot be satisfied, the Levenberg–Marquardt method combined with the deflation technique may diverge. To improve it, we consider the choice of a random deviation from the obtained solution as the initial guess based on the previous research conclusion [4].
- As highlighted previously, the Levenberg–Marquardt method is first introduced. Compared with the Newtonian method, the Levenberg–Marquardt method enables us to overcome the sensitivity of initial guesses and substantially improve the computational efficiency. This will be verified by subsequent numerical experiments.
- It is worth pointing out that when increasing N, we can use the matrix–vector multiplication to save storage space in a computer.

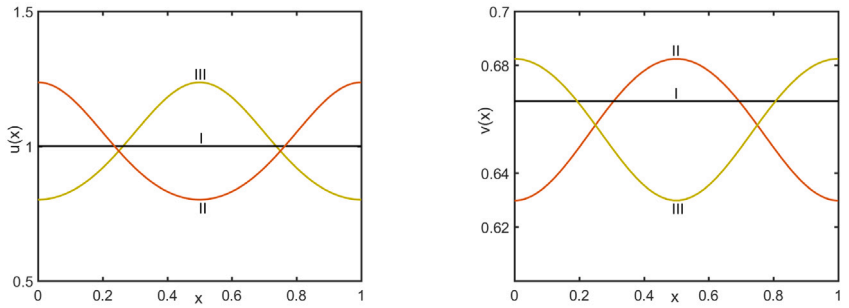
3. Numerical experiments

In this section, with several numerical experiments, we focus on the efficiency of **Algorithm 1** for finding multiple solutions. 1-D and 2-D numerical experiments are shown in Sections 3.1–3.2, respectively. We choose $\delta_1 = 0.25, \delta_2 = 0.75$ and $\epsilon = 10^{-13}$ in **Algorithm 1**, and all programs are carried out on a server with Intel(R) Core(TM) i7-7500U (2.90 GHz) and 120 GB RAM.

3.1. ODE examples

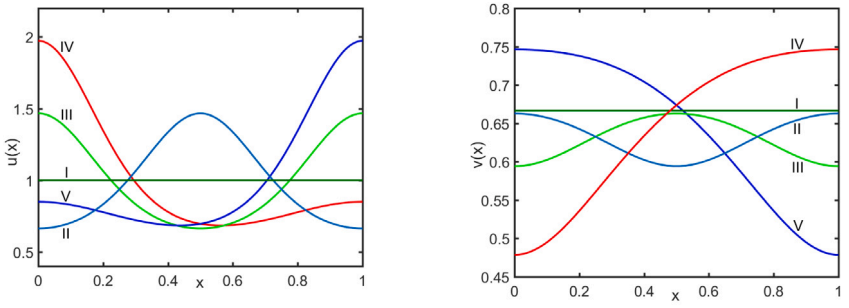
We consider the following problem:

$$\begin{cases} d_1 \frac{d^2u}{dx^2} = g_1(u, v) \\ d_2 \frac{d^2v}{dx^2} = g_2(u, v) \end{cases} \quad \text{on } \Omega = (0, 1) \tag{3.1}$$



(a) $u(x)$ (b) $v(x)$

Fig. 3.1. Multiple solutions of the Schnakenberg model with $d_2 = 50$.



(a) $u(x)$ (b) $v(x)$

Fig. 3.2. Multiple solutions of the Schnakenberg model with $d_2 = 70$.

with the no-flux boundary conditions: $\frac{du}{dx} = 0$ and $\frac{dv}{dx} = 0$ in $\partial\Omega$, where d_1 and d_2 are constants to be specified later. We focus on the Schnakenberg model (i.e. $g_1(u,v) = c(u - a - u^2v)$, $g_2(u,v) = c(u^2v - b)$) and the Gray–Scott model (i.e. $g_1(u,v) = (\mu + \rho)u - vu^2$, $g_2(u,v) = vu^2 - \rho(1 - v)$), where a, b, c, μ and ρ are constants.

3.1.1. The Schnakenberg model

As said in [34], the Schnakenberg model is a Turing model, where u is an activator and v is a substrate. With numerical tests, we consider two cases $d_2 = 50$ and $d_2 = 70$ with fixed parameters $d_1 = 1, a = 1/3, b = 2/3$ and $c = 200$, respectively. In Figs. 3.1–3.2, multiple solutions of the Schnakenberg model on $u(x)$ and $v(x)$ are presented, where we list the number of solutions labeled by I, II, ⋯, V. These multiple solutions are in agreement with that presented in [34]. The efficiency of our spectral trust region LM-Deflation method for finding these multiple solutions is considered emphatically, and the corresponding numerical results are also included in Tables 1–2. Some observations and highlights are as follows.

- When $N = 8, 16$ and 24, the accuracies of our spectral trust region LM-Deflation method are presented in Table 1, where we compare the errors in maximum norm with the numerical solution $(\hat{u}, \hat{v})$ obtained with a relatively large N, indicating that our method is quite accurate.
- Based on the following initial guesses, a comparison of $\|\boldsymbol{F}(\boldsymbol{x})\|_2$ between our method and other methods is given in Table 2.

$$\mathrm{IG}_1 : \begin{cases} \tilde{u}^{(0)} = -\mathrm{ones}(N+1,1) \\ \tilde{v}^{(0)} = -\mathrm{ones}(N+1,1) \end{cases} \quad \mathrm{IG}_2 : \begin{cases} \tilde{u}^{(0)} = -\sin(\mathrm{ones}(N+1,1)) \\ \tilde{v}^{(0)} = -\mathrm{ones}(N+1,1) \end{cases} \quad \mathrm{IG}_3 := \sin(\mathrm{IG}_1), \tag{3.2}$$

  where $N = 24$. The Newtonian iterations fail to converge for these initial inputs, while for the LSTR method and our method, the values of $\|\boldsymbol{F}(\boldsymbol{x})\|_2$ descend very fast. However, it is noteworthy that the computational times (denoted by T) of our method are much less than those of the LSTR method, indicating that the trust region Levenberg–Marquardt iteration in Section 2.2 plays an important role in improving the efficiency of our method (see Table 3).

3.1.2. The Gray–Scott model

In the chemical reaction, the Gray–Scott model is used to describe the autocatalytic reaction between the activator and the substrate. With the bootstrapping method, Hao et al. [34] studied multiple solutions of stationary spatial patterns on the Gray–Scott

Table 1
Performance of our spectral trust region LM-Deflation method with $d_2 = 50$.

	I			II			III		
	N	$\|\|\boldsymbol{F}\|\|_2$	CPU	N	$\|\|\boldsymbol{F}\|\|_2$	CPU	N	$\|\|\boldsymbol{F}\|\|_2$	CPU
$\|\|u_N - \hat{u}\|\|_\infty$	8	1.06e−14	0.34	8	5.78e−2	0.51	8	2.06e−2	0.45
	16	2.94e−15	1.65	16	3.29e−7	2.54	16	9.20e−6	3.01
	24	2.37e−15	1.89	24	8.14e−13	3.49	24	5.21e−12	3.12
$\|\|v_N - \hat{v}\|\|_\infty$	8	4.69e−13	–	8	3.96e−3	–	8	8.34e−2	–
	16	7.93e−14	–	16	5.20e−7	–	16	4.62e−5	–
	24	1.02e−15	–	24	6.74e−13	–	24	3.02e−12	–

Table 2
A comparison of $\|\boldsymbol{F}(\boldsymbol{x})\|_2$ between our method and other methods.

Newtonian iteration in [33]				LSTR method in [4]				**Algorithm 1**			
n_{it}	IG_1	IG_2	IG_3	n_{it}	IG_1	IG_2	IG_3	n_{it}	IG_1	IG_2	IG_3
5	2.78e14	3.81e13	4.98e16	5	5.03e9	3.20e7	4.08e10	5	2.08e8	5.21e10	9.38e6
10	4.51e17	9.06e18	5.92e19	10	4.78e5	3.01e3	5.78e7	10	2.51e5	9.38e5	4.78e−1
15	3.01e23	9.43e23	8.49e26	15	3.92e1	8.29e−1	6.93e2	15	7.53e−1	2.04e−2	4.80e−7
20	4.83e32	9.50e31	8.71e30	20	3.19e−5	9.85e−7	4.32e−5	20	5.97e−6	3.01e−7	5.83e−10
	–	–	–	25	5.29e−10	6.05e−10	3.50e−10	25	6.35e−11	3.01e−13	5.49e−13
T	–	–	–		4.82	3.72	5.73		3.02	2.93	3.01

Table 3
Performance of our spectral trust region LM-Deflation method with $d_2 = 70$.

	I			II			III		
	N	$\|\|\boldsymbol{F}\|\|_2$	CPU	N	$\|\|\boldsymbol{F}\|\|_2$	CPU	N	$\|\|\boldsymbol{F}\|\|_2$	CPU
$\|\|u_N - \hat{u}\|\|_\infty$	8	3.03e−13	0.40	8	1.97e−3	0.72	8	3.92e−2	0.63
	16	7.96e−15	2.01	16	5.82e−8	1.83	16	8.37e−6	2.95
	24	6.35e−15	2.45	24	9.16e−13	3.02	24	1.46e−13	3.87
$\|\|v_N - \hat{v}\|\|_\infty$	8	5.82e−13	–	8	5.89e−3	–	8	2.71e−2	–
	16	9.26e−15	–	16	6.92e−8	–	16	4.57e−6	–
	24	4.15e−15	–	24	8.31e−15	–	24	8.47e−14	–
	IV			V					
	N	$\|\|\boldsymbol{F}\|\|_2$	CPU	N	$\|\|\boldsymbol{F}\|\|_2$	CPU			
$\|\|u_N - \hat{u}\|\|_\infty$	8	2.57e−3	0.25	8	1.49e−3	0.49			
	16	4.58e−8	1.82	16	7.47e−8	1.75			
	24	6.79e−13	2.79	24	6.73e−14	3.03			
$\|\|v_N - \hat{v}\|\|_\infty$	8	2.07e−3	–	8	9.45e−3	–			
	16	2.69e−9	–	16	8.41e−8	–			
	24	9.36e−12	–	24	3.24e−13	–			

Table 4
Accuracy of our spectral trust region LM-Deflation method on $||u_N - \hat{u}||_\infty$.

N	I	II	III	IV	V	VI	VII	VIII	IX	X
8	1.03e−3	5.21e−2	5.93e−2	6.30e−2	6.29e−4	3.92e−3	6.55e−2	1.71e−3	3.18e−2	7.06e−2
16	7.57e−6	4.45e−7	4.89e−7	6.46e−6	9.50e−8	3.44e−8	1.86e−7	3.44e−6	7.91e−8	3.17e−7
24	5.73e−10	4.38e−11	6.93e−12	5.93e−11	9.52e−11	1.03e−12	8.41e−10	6.55e−11	9.21e−13	2.09e−12
N	XI	XII	XIII	XIV	XV	XVI	XVII	XVIII	XIX	XX
8	2.76e−2	4.98e−2	9.68e−3	8.42e−2	6.99e−3	1.63e−3	9.59e−2	6.55e−3	1.19e−2	2.43e−3
16	5.43e−5	6.83e−6	2.54e−7	8.14e−7	9.29e−8	2.55e−7	5.47e−7	8.14e−6	8.28e−7	7.34e−6
24	5.83e−13	1.65e−12	2.67e−10	6.66e−13	4.98e−12	1.38e−11	9.59e−12	7.51e−13	2.53e−12	2.08e−12

model on a 1D domain $[0, 1]$. Our spectral trust region LM-Deflation method is also used to find these multiple solutions, and we choose $d_1 = 2.5\times10^{-4}, d_2 = 5\times10^{-4}, \rho = 0.04$ and $\mu = 0.065$, which is the same as in [34]. Moreover, the property of the solutions on the Gray–Scott model is introduced, i.e., the solutions of the Gray–Scott model are symmetric to the center of the domain $x = 0.5$. To be specific, if $u(x)$ is a solution, then $u((x + 0.5) \bmod 1)$ is also a solution.

In Fig. 3.3, we show multiple solutions of the Gray–Scott model in a 1D domain $[0, 1]$, and the number of solutions are labeled by I, II, …, where these multiple solutions are agreement with that obtained in [34], and the solutions are symmetric with respect to the center of the domain $x = 0.5$ (such as I and III, II and IV). Numerical tests on I, … , XX are considered to show the efficiency of our spectral trust region LM-Deflation method. In Tables 4–5, with increasing N, the accuracies on $\|u_N - \hat{u}\|_\infty$ and $\|v_N - \hat{v}\|_\infty$ are

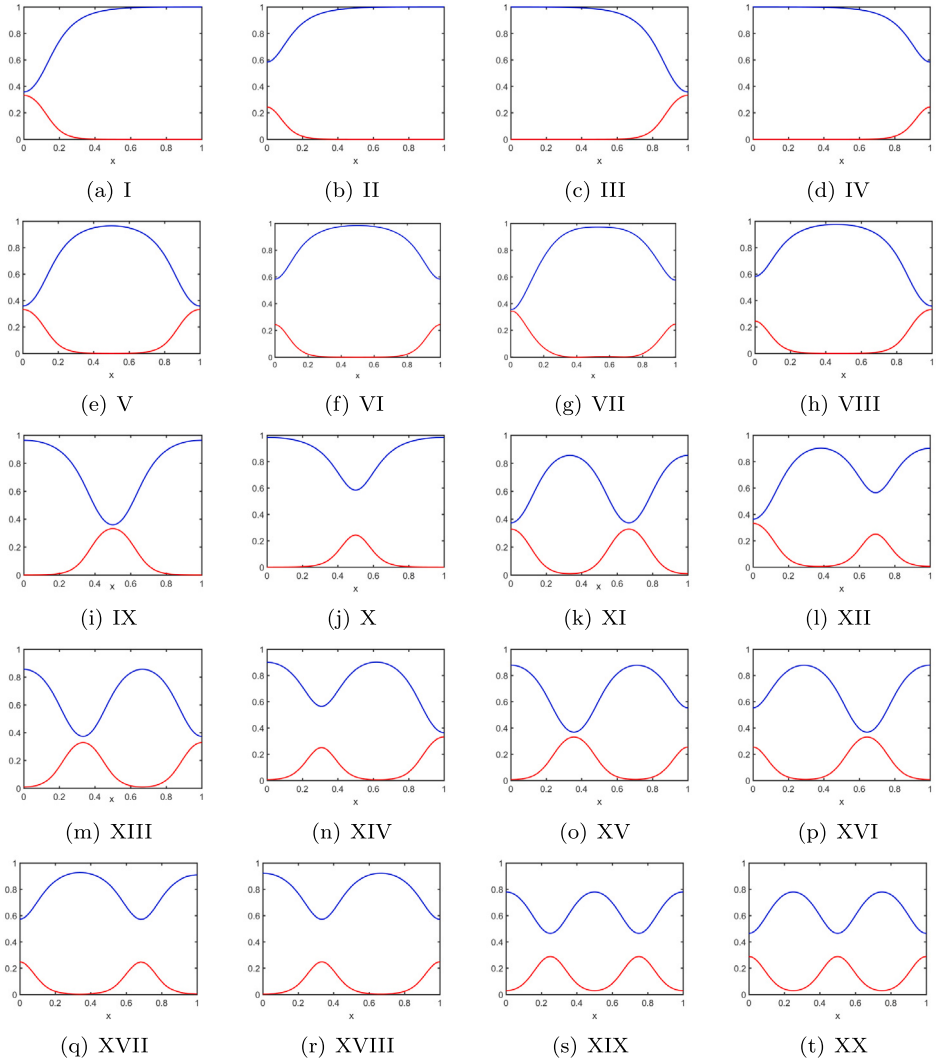


Fig. 3.3. Multiple solutions of the Gray–Scott model (Red: $u_{24}(x)$; Blue: $v_{24}(x)$).

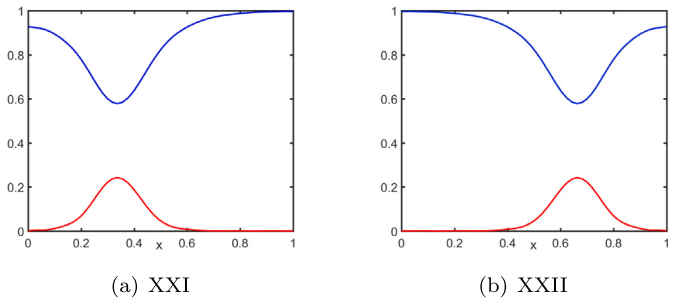


Fig. 3.4. New solutions of the Gray–Scott model (Red: $u_{24}(x)$; Blue: $v_{24}(x)$).

Table 5
Accuracy of our spectral trust region LM-Deflation method on $||v_N - \hat{v}||_\infty$.

N	I	II	III	IV	V	VI	VII	VIII	IX	X
8	3.49e−3	3.51e−2	7.58e−2	1.29e−3	1.96e−2	8.30e−3	7.57e−2	5.39e−3	5.68e−3	2.51e−3
16	2.51e−8	3.81e−7	5.30e−8	4.69e−6	1.19e−8	2.51e−7	6.16e−8	4.73e−7	9.17e−7	5.33e−6
24	5.85e−13	3.37e−11	1.62e−12	6.01e−13	8.25e−12	3.11e−11	6.89e−13	4.42e−11	8.17e−12	7.74e−13
N	XI	XII	XIII	XIV	XV	XVI	XVII	XVIII	XIX	XX
8	8.68e−3	4.31e−2	1.36e−3	8.53e−2	7.59e−2	8.44e−1	8.69e−3	3.50e−2	2.39e−2	1.23e−3
16	5.77e−5	5.13e−7	3.50e−6	1.36e−8	3.99e−6	1.81e−7	5.79e−7	1.44e−6	1.83e−8	6.22e−6
24	2.64e−13	4.90e−12	9.02e−12	4.96e−12	1.31e−13	2.34e−11	3.53e−11	8.21e−12	4.30e−12	9.64e−11

Table 6
A comparison of $\|\boldsymbol{F}(\boldsymbol{x})\|_2$ between our method and other methods for the Gray–Scott model.

Newtonian iteration in [33]				LSTR method in [4]				**Algorithm 1**			
n_{it}	IG_1	IG_2	IG_3	n_{it}	IG_1	IG_2	IG_3	n_{it}	IG_1	IG_2	IG_3
5	1.68e9	5.47e10	3.06e10	5	5.03e6	3.20e7	4.08e8	5	4.09e7	4.81e7	4.96e8
10	1.88e14	4.86e16	5.10e14	10	5.56e3	3.81e3	2.75e4	10	2.51e4	3.85e4	7.78e4
15	3.21e17	8.17e16	5.10e15	15	3.85e0	8.09e−1	7.93e0	15	4.83e1	2.67e−1	4.29e−1
20	6.86e23	4.50e25	3.68e20	20	8.41e−5	4.56e−7	4.52e−5	20	3.39e−6	8.83e−7	4.21e−6
	–	–	–	25	5.93e−10	1.98e−10	3.67e−10	25	9.12e−11	9.36e−13	7.73e−13
T	–	–	–		5.09	5.31	5.12		4.21	3.39	4.19

Table 7
Performance of our spectral trust region LM-Deflation method to new solutions of the Gray–Scott model.

N	XXI			XXII		
	8	16	24	8	16	24
L^∞ error	2.8401e−4	3.9201e−8	4.0923e−12	3.6520e−5	1.9302e−9	1.2084e−13
$\|\boldsymbol{F}(\boldsymbol{x})\|_2$	4.0398e−12	6.3028e−13	6.9281e−12	1.9026e−12	7.9036e−12	2.0538e−13

given, indicating that our spectral trust region LM-Deflation method is reliable and effective. Since we are not trying to provide the Hessian matrix exactly, the efficiency of our spectral trust region LM-Deflation method in the nonlinear iteration is well behaved in Table 6. Furthermore, two new solutions which were not reported in the literature are also found (see Fig. 3.4), and they are labeled by XXI and XXII. The corresponding numerical information is also given in Table 7.

3.2. PDE examples

We consider the model problem:

$$\begin{cases} -\triangle u = G_1(u(x,y), v(x,y)), \\ -\triangle v = G_2(u(x,y), v(x,y)), \\ u = 0,\ v = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{3.3}$$

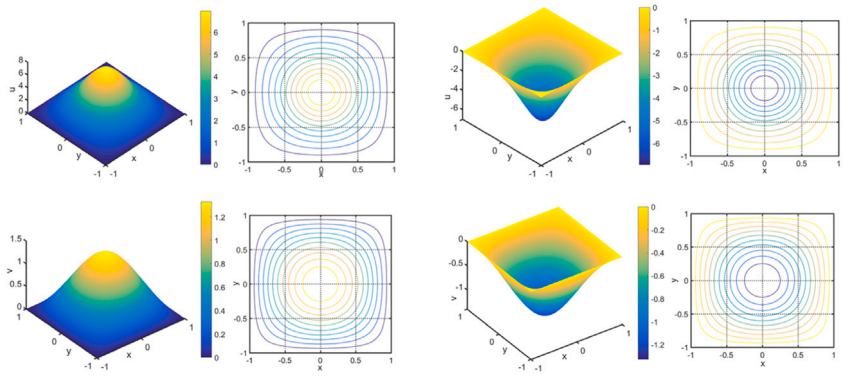
where $(x, y) \in \Omega$, and $G_1(u(x, y), v(x, y)), G_2(u(x, y), v(x, y))$ and Ω are to be specified later. Here we focus on three cases as follows:

Case 1: The problem (3.3) with $G_1(u(x, y), v(x, y)) = \lambda u - \delta v + |u|^{p-1}u$ and $G_2(u(x, y), v(x, y)) = \delta u + \gamma v - |v|^{q-1}v$ is known as the noncooperative system of definite type [5], which together with the domain $\Omega = (-1, 1)\times(-1, 1)$. We choose $p = q = 3, \lambda = -0.5, \gamma = -0.5, \delta = 5$. In Figs. 3.5–3.7, we present multiple solutions of the noncooperative system of definite type and their contours, where we list the number of solutions labeled by I, II, … , VI. The solutions with 1-peak or 2-peak are in agreement with those presented in [5]. While the solutions with multiple peaks are found and shown for the first time. The accuracies of these multiple solutions are presented in Table 8. Based on different initial guesses in (3.4) ($N = 24$), We further consider a comparison of $\|\boldsymbol{F}(\boldsymbol{x})\|_2$ between our method and other methods (see Table 9), indicating that our method is very effective again.

$$IG_1: \begin{cases} \tilde{u}^{(0)} = -\text{ones}(N+1, N+1) \\ \tilde{v}^{(0)} = -\text{ones}(N+1, N+1), \end{cases} \quad IG_2: \begin{cases} \tilde{u}^{(0)} = -\sin(\text{ones}(N+1, N+1)) \\ \tilde{v}^{(0)} = -\sin(\text{ones}(N+1, N+1)). \end{cases} \tag{3.4}$$

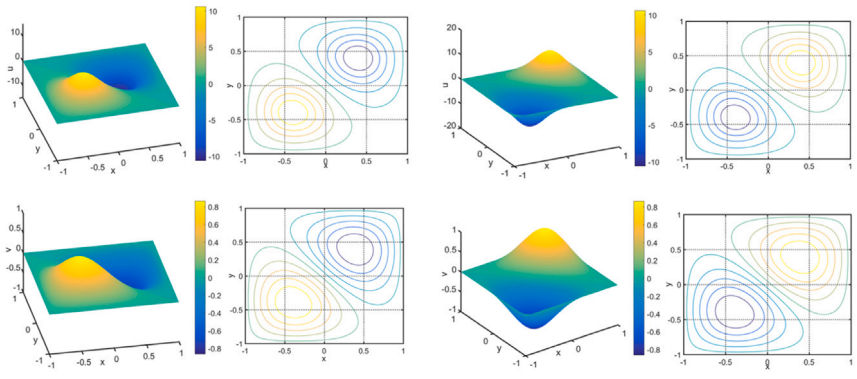
When $p = 2$ and $q = 2$, from (3.3) and Fig. 3.8, we can conclude that G_1, $G_2 \in C^1[-1, 1]$ is only satisfied. As said in the introduction, the method using second derivative information cannot be used to solve (3.3). While the trust region Levenberg–Marquardt method introduced in Section 2.2 has great advantages, and numerical results are presented in Figs. 3.9–3.11 and Table 10, indicating that our algorithm is very effective again.

Case 2: The problem (3.3) with $G_1(u(x), v(x)) = \lambda u - \delta v + |u|^{p-1}u$ and $G_2(u(x), v(x)) = \delta u + \gamma v + |v|^{q-1}v$ is known as the noncooperative system of indefinite type [5], which together with the domain $\Omega = (-3, 3)\times(-3, 3)$. The parameters $p = q = 3, \lambda = -0.5, \gamma = -1, \delta = 0.5$



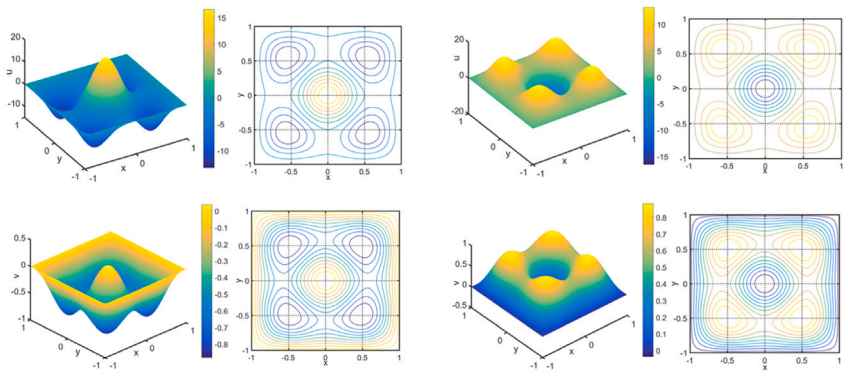
(a) I (b) II

Fig. 3.5. Profiles of 1-peak solutions for $p = 3$ and $q = 3$.



(a) III (b) IV

Fig. 3.6. Profiles of 2-peak sign-changing solutions for $p = 3$ and $q = 3$.



(a) V (b) VI

Fig. 3.7. Profiles of multi-peak solutions for $p = 3$ and $q = 3$.

are chosen. In Figs. 3.12–3.14, several multiple solutions and their contours are presented, where some asymmetric or multi-peak solutions are also found and shown. As for the efficiency of our algorithm, the situation is similar to the case above, and we do not repeat and show it.

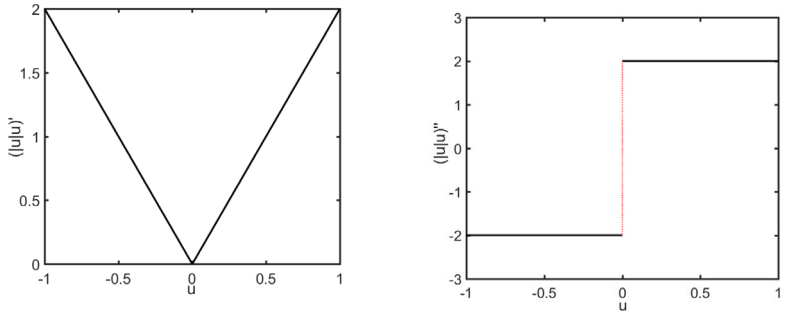
Similar to the case 1, $G_1, G_2 \in C^1[-1, 1]$ is only satisfied when $p = 2$ and $q = 2$. Here our aim is to test the efficiency of our algorithm, not to find multiple solutions as many as possible. Numerical results are presented in Figs. 3.15–3.16, and the performance of the efficiency in finding multiple solutions is similar to the case 1 above. Here we do not repeat and show it.

Table 8
Accuracy of the **Algorithm 1** for $p = 3$ and $q = 3$.

	N	I	II	III	IV	V	VI
$\|\|u_{24} - \hat{u}\|\|_\infty$	8	6.92e−1	7.65e−2	4.97e−1	1.62e−1	2.55e−2	7.09e−2
	16	1.86e−5	6.79e−6	9.59e−7	9.50e−6	7.65e−6	7.95e−5
	24	4.38e−9	3.81e−8	5.85e−9	5.05e−10	4.45e−10	1.62e−8
$\|\|v_{24} - \hat{v}\|\|_\infty$	8	8.14e−1	6.16e−2	7.58e−1	5.68e−2	2.85e−2	2.43e−2
	16	1.96e−7	5.67e−8	2.51e−5	9.34e−8	1.19e−7	4.69e−6
	24	3.49e−10	9.29e−11	2.56e−10	7.79e−9	3.49e−8	4.73e−9

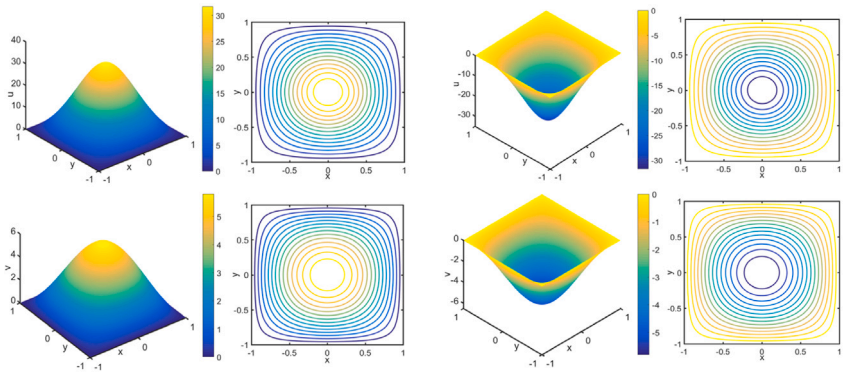
Table 9
A comparison of $\|\boldsymbol{F}(\boldsymbol{x})\|_2$ between **Algorithm 1** and other methods for $p = 3$ and $q = 3$.

Newtonian iteration in [33]			LSTR method in [4]			**Algorithm 1**		
n_{it}	IG_1	IG_2	n_{it}	IG_1	IG_2	n_{it}	IG_1	IG_2
5	2.24e10	8.17e15	5	5.92e8	8.32e10	5	2.51e9	6.92e12
10	1.44e14	8.38e19	15	2.91e5	4.82e7	15	9.91e7	6.93e8
15	9.13e20	7.48e23	25	9.62e1	7.71e2	25	4.73e3	9.53e4
20	7.81e27	7.74e29	35	3.19e−5	8.50e−4	35	2.85e−3	6.71e−4
	–	–	45	5.97e−10	3.92e−10	45	4.52e−11	7.83e−12
T	–	–		9.34	9.83		6.31	5.38



(a) $(|u|u)'$ vs. u (b) $(|u|u)''$ vs. u

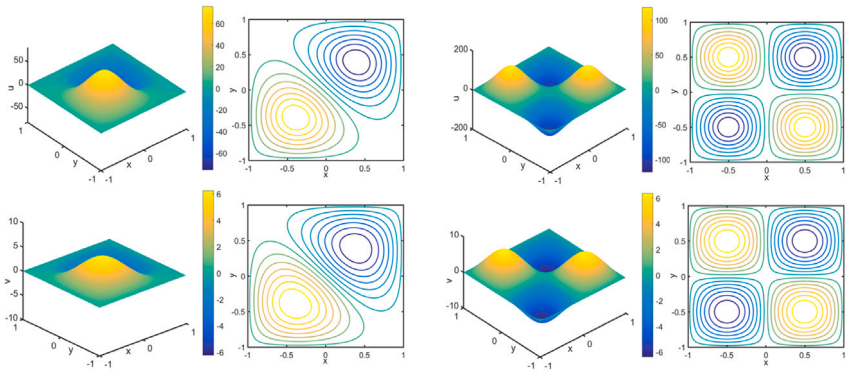
Fig. 3.8. Profiles of $(|u|u)'$ and $(|u|u)''$ vs. u on $[-1, 1]$.



(a) I (b) II

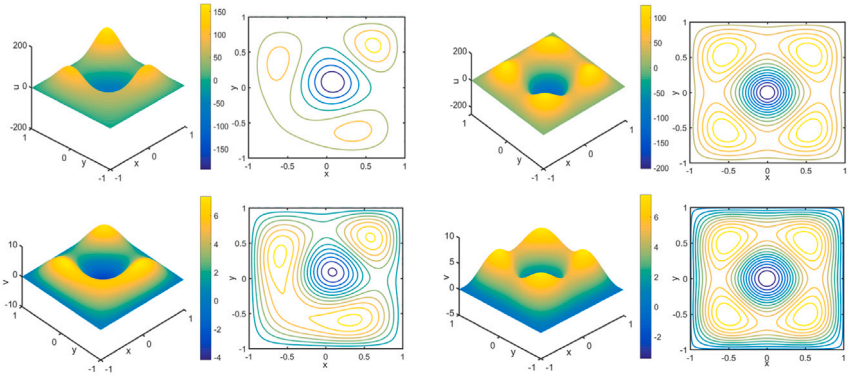
Fig. 3.9. Profiles of 1-peak solutions for $p = 2, q = 2$.

Case 3: The problem (3.3) with $G_1(u(x), v(x)) = \eta_1 u + \mu_1 u^3 + \beta u v^2$ and $G_2(u(x), v(x)) = \eta_2 v + \mu_2 v^3 + \beta u^2 v$ arises from the Bose–Einstein condensates (BEC) [35], which together with the domain $\Omega = (0, 1) \times (0, 1)$. The solution set of the problem has additional structural symmetries, that is, if (u, v) is a solution, then $(\pm u, \pm v)$ are solutions. With numerical tests, we choose $\eta_1 = \eta_2 = -1, \mu_1 = \mu_2 = 1, \beta = -5$. In Figs. 3.17–3.20, we present multiple solutions, and some multiple solutions also show the symmetry. For example, if (u, v)



(a) III (b) IV

Fig. 3.10. Profiles of 2-peak sign-changing solutions for $p = 2, q = 2$.



(a) V (b) VI

Fig. 3.11. Profiles of multi-peak solutions for $p = 2, q = 2$.

Table 10
Accuracy of the **Algorithm 1** for $p = 2$ and $q = 2$.

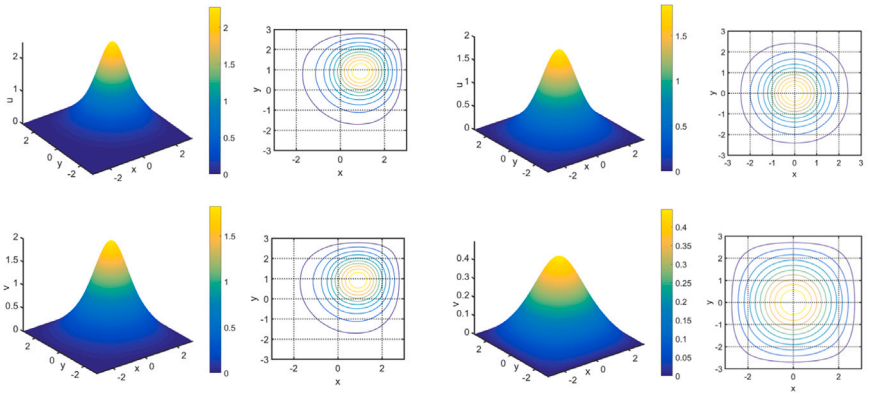
	N	I	II	III	IV	V	VI
$\|\|u_{24} - \hat{u}\|\|_\infty$	8	5.45e−2	1.79e−2	5.93e−3	3.74e−1	5.86e−1	9.32e−1
	16	4.82e−6	3.81e−5	9.65e−7	2.73e−6	4.82e−6	9.15e−5
	24	3.92e−9	8.19e−8	2.87e−9	9.04e−10	8.73e−10	6.16e−8
$\|\|v_{24} - \hat{v}\|\|_\infty$	8	6.53e−2	7.30e−2	7.81e−1	8.47e−2	1.83e−1	5.26e−3
	16	4.82e−8	7.73e−8	6.73e−5	6.10e−7	1.09e−9	5.83e−7
	24	8.27e−10	6.14e−12	6.97e−11	9.94e−9	5.83e−11	4.03e−10

represents the type-I solution in Fig. 3.17a, the type-II solution in Fig. 3.17b is $(-u, -v)$. A similar situation is also observed in Fig. 3.18. Multiple solutions presented in Figs. 3.19–3.20 also have the symmetry, and here we will not repeat them. In Tables 11–12 with different initial guesses in (3.5), numerical results on these multiple solutions are given, where the efficiency of our method is shown once again.

$$\mathrm{IG}_1 : \begin{cases} \tilde{u}^{(0)} = \mathrm{ones}(N+1, N+1) \\ \tilde{v}^{(0)} = -\mathrm{ones}(N+1, N+1), \end{cases} \qquad \mathrm{IG}_2 : \begin{cases} \tilde{u}^{(0)} = -\sin(\mathrm{ones}(N+1, N+1)) \\ \tilde{v}^{(0)} = -\cos(\mathrm{ones}(N+1, N+1)). \end{cases} \tag{3.5}$$

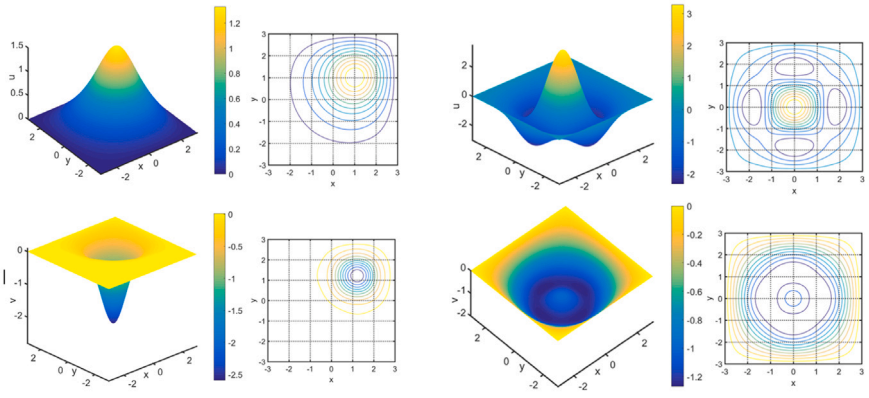
4. Concluding remarks

In this paper, an efficient trust region LM-Deflation method was proposed to compute multiple solutions of semilinear elliptic systems. Based on numerical experiments presented in Section 3, the trust region LM-Deflation method has great advantages in finding multiple solutions of semilinear elliptic systems compared with other methods given in [4,33]. In addition, our method can also work well for equations with only first-order derivative information. In subsequent studies, we will continue to apply and extend the proposed method to address more challenging problems in the future, including its convergence analysis.



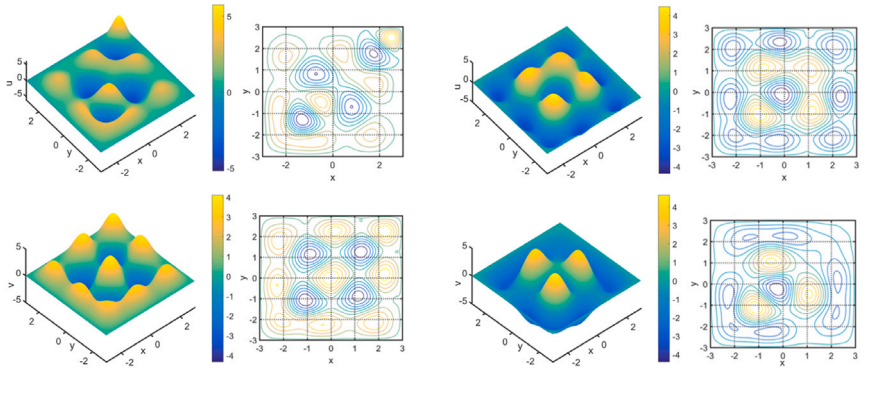
(a) I (b) II

Fig. 3.12. Profiles of an asymmetric positive solution (a) and a symmetric positive solution (b).



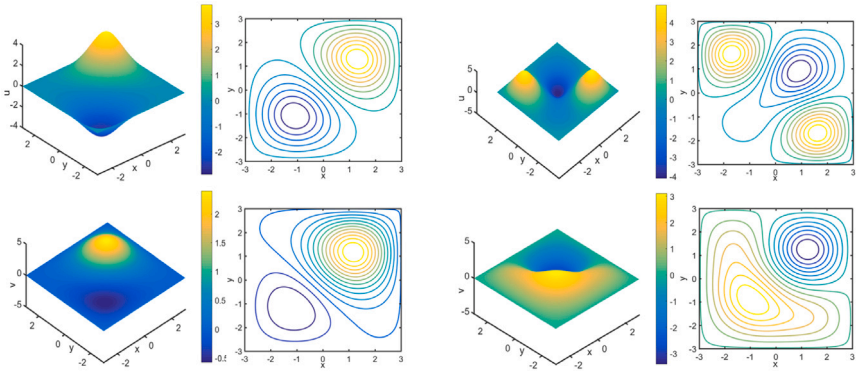
(a) III (b) IV

Fig. 3.13. Profiles of an asymmetric positive solution (a) and a symmetric sign-changing solution (b).



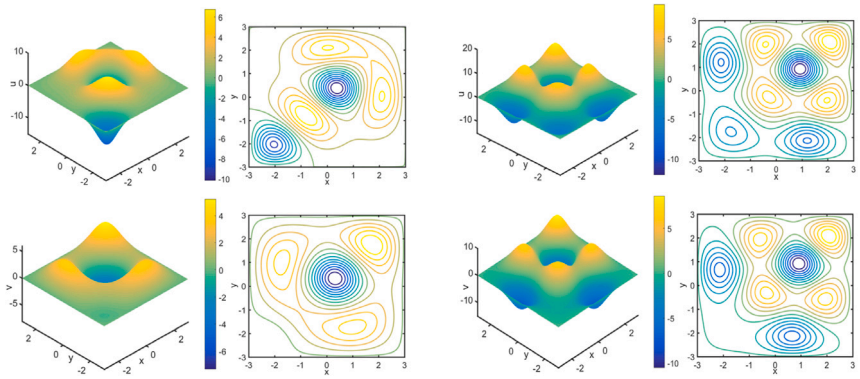
(a) V (b) VI

Fig. 3.14. Profiles of multi-peak solutions to case 2 for $p = 3$ and $q = 3$.



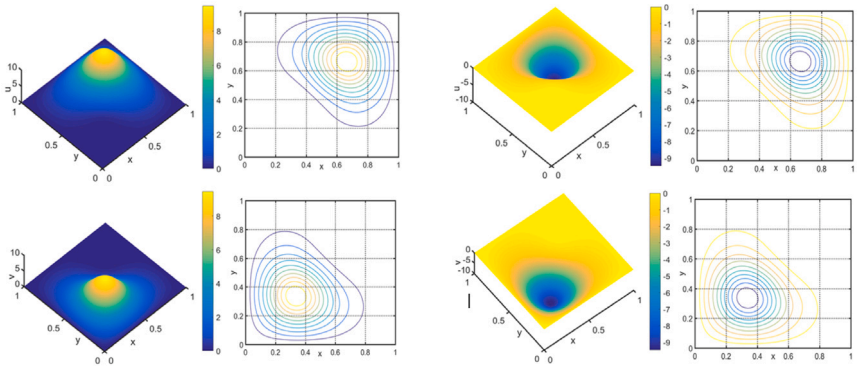
(a) I (b) II

Fig. 3.15. Profiles of solutions I and II for $p = 2$ and $q = 2$.



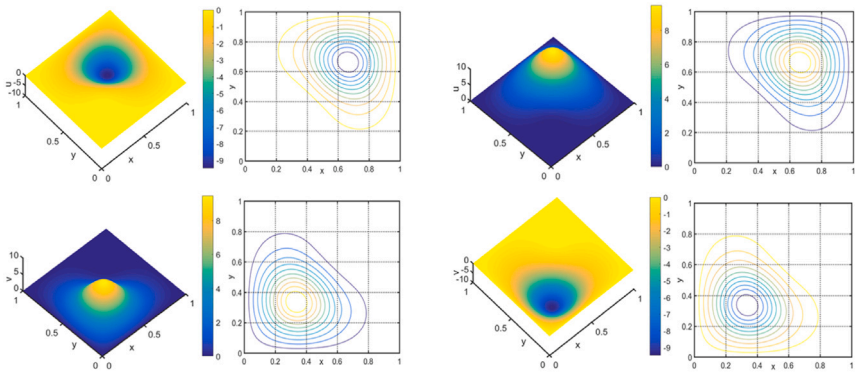
(a) III (b) IV

Fig. 3.16. Profiles of solutions III and IV for $p = 2$ and $q = 2$.



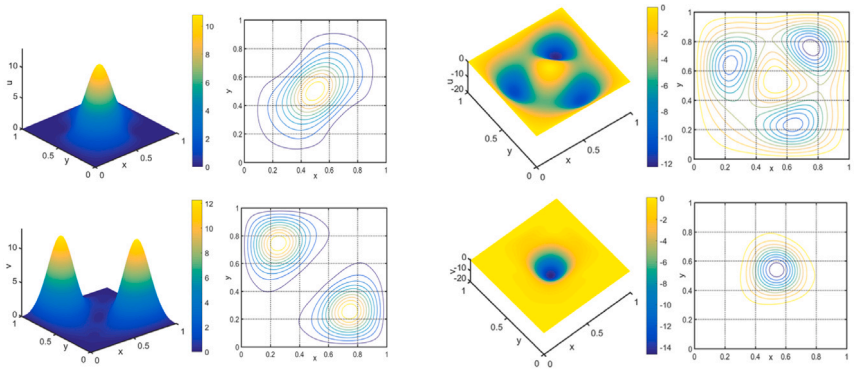
(a) I (b) II

Fig. 3.17. Profiles of solution set 1.



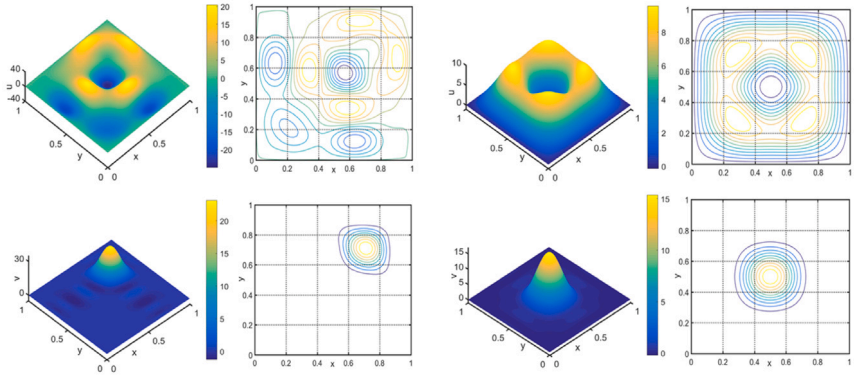
(a) III (b) IV

Fig. 3.18. Profiles of solution set 2.



(a) V (b) VI

Fig. 3.19. Profiles of solution set 3.



(a) VII (b) VIII

Fig. 3.20. Profiles of solution set 4.

Table 11
Accuracy of **Algorithm 1** to case 3.

	N	I	II	III	IV	V	VI	VII	VIII
$\|\|u_{24} - \hat{u}\|\|_\infty$	8	3.14e−1	7.06e−2	8.23e−1	4.38e−1	4.89e−3	7.43e−1	3.18e−2	6.94e−2
	16	3.17e−7	3.81e−6	4.45e−5	3.92e−6	2.76e−6	7.65e−7	4.61e−6	7.95e−6
	24	6.55e−9	5.52e−9	1.86e−9	7.09e−11	7.54e−10	9.73e−9	2.76e−11	6.94e−10
$\|\|v_{24} - \hat{v}\|\|_\infty$	8	7.13e−1	5.05e−2	2.54e−1	8.14e−2	9.59e−2	6.79e−2	2.55e−1	8.40e−1
	16	2.76e−7	1.62e−7	1.19e−6	6.99e−7	1.49e−6	5.47e−6	6.79e−6	5.85e−5
	24	3.80e−11	7.79e−11	5.30e−12	2.51e−11	7.53e−11	1.19e−11	7.58e−10	4.69e−10

Table 12
A comparison of $\|\boldsymbol{F}(\boldsymbol{x})\|_2$ between our method and other methods for case 3.

Newtonian iteration in [33]			LSTR method in [4]			**Algorithm 1**		
n_{it}	IG_1	IG_2	n_{it}	IG_1	IG_2	n_{it}	IG_1	IG_2
5	1.02e8	6.01e10	5	4.50e6	8.25e8	5	4.42e8	1.52e9
10	7.94e15	2.62e17	10	1.06e3	9.61e3	10	9.13e4	7.79e5
15	3.11e23	6.98e25	15	6.54e1	2.28e0	15	6.89e1	6.54e0
20	7.01e27	9.13e35	20	9.96e−3	7.81e−3	20	7.81e−5	5.28e−6
	–	–	25	5.28e−7	2.28e−7	25	4.63e−10	1.65e−11
T	–	–		8.01	9.21		6.09	5.97

Acknowledgment

The work of H. Li was partially supported by the National Natural Science Foundation of China (Grant Nos. 12471348, 12131005). This work was done before the corresponding author joined LBNL.

Data availability

No data was used for the research described in the article.

References

[1] U. Frisch, S. Matarrese, R. Mohayaee, A. Sobolevski, A reconstruction of the initial conditions of the Universe by optimal mass transportation, Nat. 417 (2002) 260–262.
[2] H.T. Davis, Introduction to Nonlinear Differential and Integral Equations, US Atomic Energy Commission, 1960.
[3] E. Tadmor, A review of numerical methods for nonlinear partial differential equations, B. Am. Math. Soc. 49 (2012) 507–554.
[4] L. Li, L.L. Wang, H.Y. Li, An efficient spectral trust-region deflation method for multiple solutions, J. Sci. Comput. 32 (2023) 1–23.
[5] X.J. Chen, J.X. Zhou, A local min-max-orthogonal method for finding multiple solutions to noncooperative elliptic systems, Math. Comp. 79 (2010) 2213–2236.
[6] Y.S. Choi, P.J. McKenna, A mountain pass method for the numerical solution of semilinear elliptic problems, Nonlinear. Anal. 20 (1993) 417–437.
[7] Z.Q. Xie, C.M. Chen, Y. Xu, An improved search-extension method for computing multiple solutions of semilinear PDEs, IMA J. Numer. Anal. 25 (2005) 549–576.
[8] Z.H. Ding, D. Costa, G. Chen, A high-linking algorithm for sign-changing solutions of semilinear elliptic equations, Nonlinear. Anal. 38 (1999) 151–172.
[9] Y.X. Li, J.X. Zhou, A minimax method for finding multiple critical points and its applications to semilinear PDEs, SIAM J. Sci. Comput. 23 (2001) 840–865.
[10] X.D. Yao, J.X. Zhou, A minimax method for finding multiple critical points in Banach spaces and its application to quasi-linear elliptic PDEs, SIAM J. Sci. Comput. 26 (2005) 1796–1809.
[11] X.D. Yao, J.X. Zhou, Numerical methods for computing nonlinear eigenpairs: Part I. Iso-homogeneous cases, SIAM J. Sci. Comput. 29 (2007) 1355–1374.
[12] X.D. Yao, J.X. Zhou, Numerical methods for computing nonlinear eigenpairs: Part II. Non-iso-homogeneous cases, SIAM J. Sci. Comput. 30 (2008) 937–956.
[13] J.X. Zhou, Instability analysis of saddle points by a local minimax method, Math. Comp. 74 (2004) 1391–1411.
[14] C.M. Chen, Z.Q. Xie, Search extension method for multiple solutions of a nonlinear problem, Comp. Math. Appl. 47 (2004) 327–343.
[15] Z.Q. Xie, C.M. Chen, Y. Xu, An improved search-extention method for solving semilinear PDEs, Acta. Math. Sci. 26 (2006) 757–766.
[16] Z.Q. Xie, W.F. Yi, J.X. Zhou, An augmented singular transform and its partial Newton method for finding new solutions, J. Comput. Appl. Math. 286 (2015) 145–157.
[17] E.L. Allgower, A.J. Sommese D.J. Bates, C.W. Wampler, Solution of polynomial systems derived from differential equations, Comput. 76 (2006) 1–10.
[18] E. Allgower, S.G. Cruceanu, S. Tavener, Application of numerical continuation to compute all solutions of semilinear elliptic equations, Adv. Geom. 9 (2009) 371–400.
[19] W.R. Hao, J.D. Hauenstein, B. Hu, A.J. Sommese, A bootstrapping approach for computing multiple solutions of differential equations, J. Comput. Appl. Math. 258 (2014) 181–190.
[20] Y. Wang, W. Hao, G. Lin, Two-level spectral methods for nonlinear elliptic equations with multiple solutions, SIAM J. Sci. Comput. 40 (2018) B1180–B1205.
[21] X.P. Zhang, J.T. Zhang, B. Yu, Eigenfunction expansion method for multiple solutions of semilinear elliptic equations with polynomial nonlinearity, SIAM J. Numer. Anal. 51 (2013) 2680–2699.
[22] J. Shen, T. Tang, L.L. Wang, Spectral Methods: Algorithms, Analysis and Applications, vol. 41, Springer Science & Business Media, 2011.
[23] P. Xie, Y. Yuan, Derivative-free optimization with transformed objective functions (DFOTO) and the algorithm based on the least Frobenius norm updating quadratic model, J. Oper. Res. Soc. China (2024).
[24] P. Xie, Sufficient conditions for error distance reduction in the ℓ^2-norm trust region between minimizers of local nonconvex multivariate quadratic approximates, J. Comput. Appl. Math. 453 (2025) 116146.

[25] P. Xie, Y. Yuan, Least H^2 norm updating of quadratic interpolation models for derivative-free trust-region algorithms, IMA J. Numer. Anal. (2025) drae106.
[26] P. Xie, S.M. Wild, ReMU: Regional minimal updating for model-based derivative-free optimization, 2025, arXiv preprint arXiv:2504.03606.
[27] W.Y. Sun, Y.X. Yuan, Optimization Theory and Methods: Nonlinear Programming, vol. 1, Springer Science & Business Media, 2006.
[28] J.Y. Fan, The modified Levenberg–Marquardt method for nonlinear equations with cubic convergence, Math. Comp. 81 (2012) 447–466.
[29] N. Yamashita, M. Fukushima, On the Rate of Convergence of the Levenberg–Marquardt Method, vol. 15, Computing (Supplement 15), 2001.
[30] J.Y. Fan, Y.X. Yuan, On the quadratic convergence of the Levenberg–Marquardt method without nonsingularity assumption, Comput. 74 (2019) 23–39.
[31] J.Y. Fan, J.C. Huang, J.Y. Pan, An adaptive multi-step Levenberg–Marquardt method, J. Sci. Comput. 78 (2019) 531–548.
[32] K.M. Brow, W.B. Gearhart, Deflation techniques for the calculation of further solutions of a nonlinear system, Numer. Math. 16 (1971) 334–342.
[33] P.E. Farrell, A. Birkisson, S.W. Funke, Deflation techniques for finding distinct solutions of nonlinear partial differential equations, SIAM J. Sci. Comput. 37 (2015) A2026–A2045.
[34] W.R. Hao, C. Xue, Spatial pattern formation in reaction–diffusion models: a computational approach, J. Math. Biol. 80 (2020) 521–543.
[35] X.P. Zhang, J. Zhang, B. Yu, Finding multiple solutions to elliptic systems with polynomial nonlinearity, Numer. Methods Partial Differential Equations 36 (2019) 1074–1097.