

Searching globally optimal parameter sequence for defeating Runge phenomenon by immunity genetic algorithm



Hongwei Lin*, Linjie Sun

Department of Mathematics, State Key Lab. of CAD&CG, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Keywords:

Data interpolation
Runge phenomenon
Parametric curve
Parameter sequence
Immunity genetic algorithm

ABSTRACT

Data interpolation is a fundamental data processing tool in scientific studies and engineering applications. However, when interpolating data points on an equidistant grid using polynomials, the so-called Runge phenomenon may occur, making polynomial interpolation unreliable. Although there are some methods proposed to defeat the Runge phenomenon, it is still an open problem which parameter sequence is the globally optimal for overcoming the Runge phenomenon. In this paper, we develop an immunity genetic algorithm based method to solve this problem. Specifically, we first model the Runge-phenomenon-defeating problem as an optimization in which the objective function is the energy of the parametric curve. An immunity genetic algorithm is then devised to determine the best IGA parameter sequence, which minimizes the objective function. The resulting parametric curve overcomes the Runge phenomenon. By performing the proposed immunity genetic searching algorithm starting with some groups of randomly generated parameter sequences, the resulted parameter sequences closely oscillate around the Chebyshev parameter sequence. Therefore, the Chebyshev parameter sequence is most likely the globally optimal sequence conquering the Runge phenomenon.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Data interpolation is a fundamental tool in scientific studies and engineering applications. However, polynomial interpolation on equidistant grids is unreliable because of the famous Runge phenomenon, which was discovered independently by Carl Runge, C. Méray, and Emilie Borel more than a century ago [1–5]. Specifically, given an analytical function $f(x) = 1/(1 + 25x^2)$, $x \in [-1, 1]$, we sample some points at the following equidistant nodes:

$$x_{i,N} = -1 + 2i/N, \quad i = 0, 1, \dots, N, \quad N = 1, 2, \dots, \quad (1)$$

and interpolate these sampling points with the Lagrange interpolation polynomial $f_N(x)$, i.e.,

$$f_N(x) = \sum_{i=0}^N f(x_{i,N})L_i(x), \quad (2)$$

* Corresponding author. Tel.: +86 571 87951860 8304; fax: +86 571 88206681.

E-mail addresses: hwlin@zju.edu.cn, hwlin@zjucadcg.cn (H. Lin), sunlinjie@zjucadcg.cn (L. Sun).

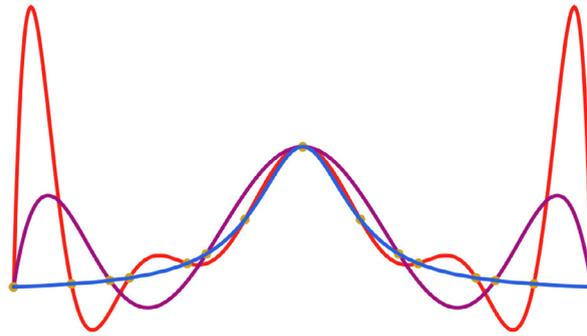


Fig. 1. Runge phenomenon generated by Lagrange interpolation at $N = 7$ and $N = 11$ equidistant sampling points, respectively. In this figure, the blue curve is the function $f(x) = 1/(1 + 25x^2)$, $x \in [-1, 1]$, the purple curve is the Lagrange interpolation polynomial $f_N(x)$ (2) for $N = 7$, and the red curve is the Lagrange interpolation polynomial $f_N(x)$ (2) for $N = 11$.

where,

$$L_i(x) = \frac{\prod_{j \neq i} (x - x_{j,N})}{\prod_{j \neq i} (x_{i,N} - x_{j,N})}.$$

The Runge phenomenon means that $f_N(x)$ will diverge near the endpoints $x = \pm 1$ when $N \rightarrow \infty$. Fig. 1 illustrates the Runge phenomenon for the Lagrange interpolation polynomial $f_N(x)$ (2), where $N = 7$ and $N = 11$, respectively.

Currently, there are a number of methods for defeating the Runge phenomenon (refer to Section 2). Nearly all of them deal with a polynomial function interpolator (2). To the best of our knowledge, no work has considered defeating the Runge phenomenon using a parametric curve interpolator. In fact, after sampling some points equidistantly,

$$\mathbf{P}_i = (x_{i,N}, f(x_{i,N})) = (x_{i,N}, 1/(1 + 25x_{i,N}^2)), \quad i = 0, 1, \dots, N, \quad N = 1, 2, \dots,$$

and assigning a parameter t_i to each of these points, where $-1 = t_0 < t_1 < \dots < t_N = 1$, they can be interpolated by a parametric curve $\mathbf{P}(t)$, i.e.,

$$\mathbf{P}(t) = \sum_{i=0}^N \mathbf{P}_i L_i(t), \quad t \in [-1, 1],$$

where $L_i(t)$ is the Lagrange basis function,

$$L_i(t) = \frac{\prod_{j \neq i} (t - t_{j,N})}{\prod_{j \neq i} (t_{i,N} - t_{j,N})}, \quad i = 0, 1, \dots, N. \quad (3)$$

The parametric curve $\mathbf{P}(t)$ is just the interpolator of the sampling points \mathbf{P}_i , i.e., $\mathbf{P}(t_i) = \mathbf{P}_i$, $i = 0, 1, \dots, N$.

As stated in Section 2, the existing methods for defeating the Runge phenomenon usually design or calculate a sequence of suitable sampling points so that the Runge phenomenon disappears for the polynomial interpolating these sampling points. However, they cannot answer the following question: Which is the globally optimal parameter sequence for defeating the Runge phenomenon?

In this paper, we endeavor to find the optimal parameters that make the shape of the parametric interpolation curve as desirable as possible, thus defeating the Runge phenomenon in *interpolating equidistant sampling points*. Specifically, we employ an energy (refer to Section 3) to measure the quality of the shape of the interpolation curve. Thus, searching for optimal parameters to make the interpolation curve as desirable as possible is equivalent to minimizing the energy of the curve. Because the energy of the interpolation curve is highly non-linear, traditional optimization methods are either inefficient or invalid. Therefore, we develop an immunity genetic algorithm (IGA)-based approach to solve this highly non-linear optimization problem. Moreover, by investigating dozens of groups of optimal parameters, we find the interesting result that they oscillate closely around the Chebyshev parameters.

The structure of this paper is as follows. In Section 2, we survey the related work on defeating the Runge phenomenon, and genetic algorithm. After introducing the parametric curve interpolator and fitness function in Section 3, the IGA-based optimal parameter search algorithm is developed in Section 4. Moreover, some best IGA parameter sequences are presented and discussed in Section 5. Finally, this paper is concluded in Section 6.

2. Related work

In this section, we will briefly review some methods of defeating the Runge phenomenon and related work on genetic algorithms.

Methods for defeating the Runge phenomenon: As stated above, some techniques for defeating the Runge phenomenon have been presented. The regularization method defeats the Runge phenomenon by minimizing a cost function, which is the sum of a residual item and a smoothness item [6,7]. Because the Runge phenomenon never occurs when the interpolation points are sampled according to Chebyshev nodes, the mock-Chebyshev method re-samples the equidistant interpolation points to a subset, which approximates the distribution of the Chebyshev grid [8,9]. The three-interval method proposed by Boyd is an efficient approach to conquering the Runge phenomenon. This technique employs one method on the interval $[-1 + \varepsilon, 1 - \varepsilon]$, and a different method on $[-1, -1 + \varepsilon]$ and $[1 - \varepsilon, 1]$ [10,11]. In addition, the least-squares fitting method can also weaken the Runge effect by fitting n samples with a polynomial of degree d ($d \ll n - 1$) [8,12].

There are many other methods of defeating the Runge phenomenon, including those based on radial basis functions [13,14] and methods that overcome the Gibbs phenomenon [15,16]. For more details on Runge phenomenon-defeating methods, please refer to [7,17,18].

However, these methods aforementioned focus on developing sampling rules for sampling suitable points so that the Runge phenomenon is defeated in polynomial interpolation. In this paper, we study the problem on defeating the Runge phenomenon in interpolating the equidistant sampling points using parametric curves, and devise an IGA-based algorithm to search the globally optimal parameter sequence.

Genetic algorithm: Proposed by John Holland [19], genetic algorithms (GAs) have been widely employed in machine learning, artificial intelligence, and adaptive control [20,21]. In particular, GAs have been successfully applied to highly non-linear optimization problems [21–23]. Unlike traditional optimization methods, which always handle single points in the search space, GAs maintain a collection of solutions, and perform a multidirectional search. The collection of solutions is a population of individuals, each of which is represented by a chromosome. The population of individuals evolves via selection mechanisms. The fittest individuals are selected to reproduce, and the weaker individuals are eliminated. In each generation, new individuals are generated from one or two parents by mutation and crossover. This evolution process terminates when the best solution has been found.

An IGA [24] is an improved version of the classical GA. Based on the biological theory of immunity, the IGA constructs an immune operator by vaccination and immune selection, thus greatly improving the search ability and convergence speed.

GA and IGA have been extensively employed in nonlinear optimization. In [25], a hybrid approach combining two heuristic optimization techniques, particle swarm optimization and GA, is introduced to solve nonlinear optimization problem. In [26], GAs are employed in settling large-scale nonlinear optimization problems encountered in earth sciences. Moreover, in data fitting with B-spline curve or surface, GAs are utilized to solve the nonlinear optimization problems for selecting the desirable B-spline knot sequence [27,28]. Moreover, in Ref. [29], the problem of minimizing the fitting error by choosing suitable knot sequences is modeled as a nonlinear optimization problem, and figured out using IGA.

In this paper, we devise an IGA-based optimal parameter search algorithm for defeating the Runge phenomenon. To our knowledge, this approach has never been used to search for optimal parameters to defeat the Runge phenomenon.

3. Parametric curve interpolator and the fitness function

As stated above, if we interpolate the points sampled from the analytical function, $f(x) = 1/(1 + 25x^2), x \in [-1, 1]$, at the equidistant nodes (1), using the Lagrange polynomial interpolator (2), the Runge phenomenon will appear when $N \rightarrow \infty$ (Fig. 1), due to the divergence of the Lagrangian basis functions $L_i(x)$ (2) when $N \rightarrow \infty$. Nearly all methods of defeating the Runge phenomenon strive to avoid it using the polynomial interpolator (2) to interpolate the equidistant sampling points.

However, when we sample the function $f(x) = 1/(1 + 25x^2), x \in [-1, 1]$ on the Chebyshev grid

$$x_i^c = \cos(\pi i/N), i = 0, 1, \dots, N, N = 1, 2, \dots, \tag{4}$$

and apply the Lagrange interpolation polynomial $f_N^c(x)$ defined on (4), i.e.,

$$f_N^c(x) = \sum_{i=0}^N f(x_{i,N})L_i^c(x), \quad \text{where, } L_i^c(x) = \frac{\prod_{j \neq i} (x - x_{j,N}^c)}{\prod_{j \neq i} (x_{i,N}^c - x_{j,N}^c)}, \tag{5}$$

the Runge phenomenon does not occur.

Clearly, the Runge phenomenon is caused by the divergence of the Lagrangian basis functions, which are determined by the distribution of the nodes on which they are defined. Therefore, the key to defeating the Runge phenomenon is to make the Lagrangian basis functions non-divergent.

In this paper, a parametric curve is employed to interpolate the sampling points. Suppose the sampling points for the function $f(x) = 1/(1 + 25x^2), x \in [-1, 1]$ at equidistant nodes are

$$\mathbf{P}_i = (x_{i,N}, f(x_{i,N})) = (-1 + 2i/N, f(-1 + 2i/N)), \tag{6}$$

$$i = 0, 1, \dots, N, N = 1, 2, \dots$$

Each sampling point \mathbf{P}_i can be assigned a parameter $t_i, i = 0, 1, \dots, N$. The parameter for each sampling point can be chosen arbitrarily, with the prerequisite that the parameter sequence

$$\mathbf{T} = \{t_0, t_1, \dots, t_N\} \tag{7}$$

is strictly increasing, i.e., $t_0 < t_1 < \dots < t_N$. In our implementation, we fix $t_0 = -1$ and $t_N = 1$. A parametric Lagrange interpolation curve is then constructed:

$$\mathbf{P}(t) = \sum_{i=0}^N \mathbf{P}_i L_i(t), \quad t \in [-1, 1], \quad \text{where, } L_i(t) = \frac{\prod_{j \neq i} (t - t_j)}{\prod_{j \neq i} (t_i - t_j)}. \quad (8)$$

The parametric curve $\mathbf{P}(t)$ (8) interpolates the sampling points, i.e., $\mathbf{P}(t_i) = \mathbf{P}_i, i = 0, 1, \dots, N$.

Note that the Lagrange basis functions (8) are determined by the parameter sequence (7). If appropriate parameters are selected, the Lagrange basis functions are non-divergent, and then the Runge phenomenon is defeated. It is well known that if the Chebyshev grid (4) is taken to give the parameters, i.e., $t_i = x_i^c, i = 0, 1, \dots, N$ (named the *Chebyshev parameter sequence*), the Runge phenomenon is defeated even if the sampling points (6) are at the equidistant nodes (1). In this paper, we study the Runge phenomenon-defeating problem from a new perspective, i.e., we seek an *optimal* parameter sequence that satisfies the following conditions:

- (1) The distance between the parametric curve $\mathbf{P}(t)$ (8) and the curve of the analytical function $f(x) = 1/(1 + 25x^2), x \in [-1, 1]$ is as small as possible;
- (2) $\mathbf{P}(t)$ (8) is as smooth as possible.

Clearly, if the curve $\mathbf{P}(t)$ (8) meets the conditions stated above, the Runge phenomenon is defeated.

Specifically, the first condition guarantees that $\mathbf{P}(t)$ (8) is not divergent when $N \rightarrow \infty$. This is ensured by the minimization of

$$E_{\text{error}}(\mathbf{T}) = \int_{t_0}^{t_N} \|\mathbf{P}(t) - \mathbf{I}(t)\|^2 dt, \quad (9)$$

where \mathbf{T} is the unknown parameter sequence, and $\mathbf{I}(t)$ is the polyline generated by connecting the sampling points, defined as

$$\mathbf{I}(t) = \left(1 - \frac{t - t_i}{t_{i+1} - t_i}\right) \mathbf{P}_i + \frac{t - t_i}{t_{i+1} - t_i} \mathbf{P}_{i+1}, \quad t \in [t_i, t_{i+1}], i = 0, 1, \dots, N - 1. \quad (10)$$

The second condition is satisfied by minimizing the following function:

$$E_{\text{smooth}}(\mathbf{T}) = \int_{t_0}^{t_N} \|\mathbf{P}''(t)\|^2 dt. \quad (11)$$

Consequently, to satisfy these conditions simultaneously, we must minimize the following energy function E with variables t_1, t_2, \dots, t_{N-1} :

$$E(\mathbf{T}) = E(t_1, t_2, \dots, t_{N-1}) = \alpha E_{\text{error}}(\mathbf{T}) + \beta E_{\text{smooth}}(\mathbf{T}), \quad (12)$$

where $\alpha > 0$ and $\beta > 0$ are weights for balancing the two items E_{error} and E_{smooth} . On one hand, if the weight α is fixed, the larger the weight β is, the smoother the curve $\mathbf{P}(t)$ (8) is. On the other hand, with the fixed weight β , the larger the weight α is, the smaller the distance between $\mathbf{P}(t)$ (8) and $f(x) = 1/(1 + 25x^2), x \in [-1, 1]$ is. However, because the object function E (12) is highly non-linear, traditional optimization methods are inefficient or invalid. Therefore, we develop an IGA-based algorithm to search for optimal parameters by minimizing $E(\mathbf{T}) = E(t_1, t_2, \dots, t_{N-1})$ (12), which is taken as the *fitness* function in the IGA. Interestingly, lots of experimental data show that the optimal parameter sequence \mathbf{T} fluctuates closely around the Chebyshev parameter sequence.

4. Optimal parameter search algorithm based on IGA

Although IGA and GA have been extensively employed in solving nonlinear optimization, the applications of IGA and GA are problem-oriented. Therefore, the algorithms of IGA and GA are different for different problems. For example, in Ref. [29], IGA is utilized to solve the nonlinear optimization problem, which minimizes the B-spline surface fitting error by selecting suitable bidirectional B-spline knot sequences. In the IGA method developed in Ref. [29], the fitness function is the fitting error of the B-spline surface, and the initial population consists of the bidirectional knot sequences that are generated by randomizing an initial bidirectional knot sequence produced by the accumulated chord length method. Consequently, the crossover, mutation, vaccination, and the probability-regulating formula (15) are all different from those in the IGA algorithm developed in this paper. To our knowledge, it is the first time for IGA to be employed in defeating the Runge phenomenon.

The IGA-based optimal parameter search algorithm is outlined in Algorithm 1, and will be explained step by step in this section.

Step 1: Generation of the initial random population:

The first step of the IGA is to generate the initial random population. Each individual of the population is a parameter sequence \mathbf{T} (7). As the Runge phenomenon occurs near the end points, the parametric domain $[-1, 1]$ is divided into three subintervals, i.e.,

$$[-1, -t_m], (-t_m, t_m), \quad \text{and } [t_m, 1], \quad \text{where } 0 < t_m < 1. \quad (13)$$

Algorithm 1: Outline of the optimal parameter search algorithm based on IGA

- 1 Generate the initial random population;
- 2 **while** *The termination condition is not satisfied* **do**
- 3 Calculate the fitness function for each individual;
- 4 Choose individuals with duplication restraining;
- 5 Perform crossover, mutation, and vaccination operations;
- 6 **end**

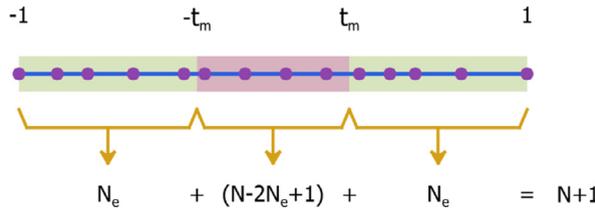


Fig. 2. Structure of an individual.

In each of the two end subintervals $[-1, -t_m]$ and $[t_m, 1]$, we assign N_e initial random parameters; in the middle subinterval $(-t_m, t_m)$, $N - 2N_e + 1$ initial random parameters are generated. Therefore, each individual T consists of three segments, i.e.,

$$\begin{aligned}
 T_I &= \{t_0, t_1, \dots, t_{N_e-1}\}, \\
 T_{II} &= \{t_{N_e}, t_{N_e+1}, \dots, t_{N-N_e}\}, \\
 T_{III} &= \{t_{N-N_e+1}, t_{N-N_e+2}, \dots, t_N\},
 \end{aligned} \tag{14}$$

located in the respective subintervals (Fig. 2). The total number of parameters in each individual T is $N + 1$. In our implementation, we employ the Mersenne Twister algorithm [30] to generate pseudo-random numbers.

Step 2: Calculation of the fitness function:

The fitness function $E(t_1, t_2, \dots, t_{n-1})$ (12) includes two items, E_{error} (9) and E_{smooth} (11). The smaller the fitness value, the better the individual. Because the curve $I(t)$ in the integrand of E_{error} (9) is defined piecewise, we calculate E_{error} (9) piece by piece, i.e.,

$$E_{error}(T) = \int_{t_0}^{t_N} \|P(t) - I(t)\|^2 dt = \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} \|P(t) - I(t)\|^2 dt.$$

In our implementation, the integrals in E_{error} (9) and E_{smooth} (11) are calculated by the Gauss–Lobatto method [31].

The weights α and β in the fitness function $E(t_1, t_2, \dots, t_{n-1})$ (12) are used to balance the two items E_{error} and E_{smooth} . When interpolating points sampled from the function $f(x) = 1/(1 + 25x^2)$, $x \in [-1, 1]$, desirable results are generated by taking $\alpha = 1000$ and $\beta = 1$ in a set of experiments with different combinations of α and β .

[Remark:] If several adjacent parameters in an individual are very close, the calculation of the fitness function (12) may lead to an arithmetic overflow or NaN fault. Once such an individual is found in the IGA iterations, it is immediately re-initialized.

Step 3: Individual selection with duplication restraining:

In our implementation, we employ *roulette-wheel selection* [32] to choose individuals. Suppose e_i is the fitness value (9) of the i th individual. To perform the roulette-wheel selection, let

$$f_i = e_{max} - e_i, \text{ for each individual in the current generation,}$$

where $e_{max} = \max_i \{e_i\}$. The probability g_i of the i th individual being selected is then defined as

$$g_i = \frac{f_i}{\sum_j f_j}.$$

Moreover, the *concentration* of an individual is the number of individuals in a *similar set*, in which the distance between any two individuals $T^{(i)}$ and $T^{(j)}$ satisfies

$$d(T^{(i)}, T^{(j)}) = \sqrt{\sum_{l=0}^N (t_l^{(i)} - t_l^{(j)})^2} < \varepsilon_s.$$

The threshold ε_s should be small enough to ensure that an individual is not contained in two or more similar sets. In our implementation, ε_s is taken as 10^{-6} . If there are k individuals in a similar set, the concentration of each individual in the similar set is taken as k . We denote the concentration of the i th individual as k_i , and the average concentration as k_a .

It is well-known that high concentrations of some individuals may lead to premature convergence of the algorithm. To overcome this problem and improve the performance of the algorithm, the duplication of individuals with high concentrations

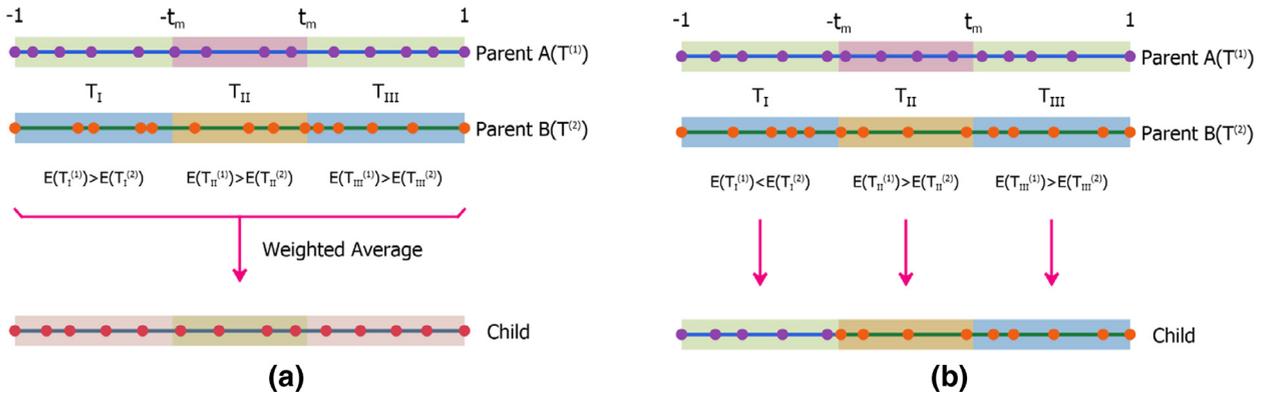


Fig. 3. Two crossover operators. (a) Weighted average operator: If the three segments of one individual are all better than those of the other individual, the new individual is generated as their weighted average. (b) Extraction operator: If not all of the segments of one individual are better than those of the other individual, we extract the better segments from the two individuals to form the new individual.

Table 1
Parameters employed in the IGA-based searching algorithm.

Parameters	$N = 20$	$N = 30$
Population size	200	200
The maximum generation	300	500
Selection rate	0.6	0.6
Cross rate	0.3	0.3
Mutation rate	0.3	0.3
t_m (13)	$\frac{1}{3}$	$\frac{1}{3}$
N_e (14)	8	12
α, β (12)	1000, 1	1000, 1
α_r (15)	0.05	0.05
ε_t	50	50

should be restrained to suppress the number of identical or similar individuals. To this end, the probability g_i is further regulated as

$$p_i = \begin{cases} \frac{\alpha_r k_a}{k_i} g_i, & k_i > k_a, \\ g_i, & k_i \leq k_a, \end{cases} \tag{15}$$

where α_r is a weight (refer to Table 1).

Furthermore, supposing the population size in one generation is M , $\lceil \frac{M}{100} \rceil$ individuals with small fitness values are selected to be stored in the immunological memory.

Step 4: Crossover, mutation, and vaccination:

Because each individual \mathbf{T} is an ascending parameter sequence, the basic requirement for the crossover and mutation operators is that the sequence generated by them is also ascending.

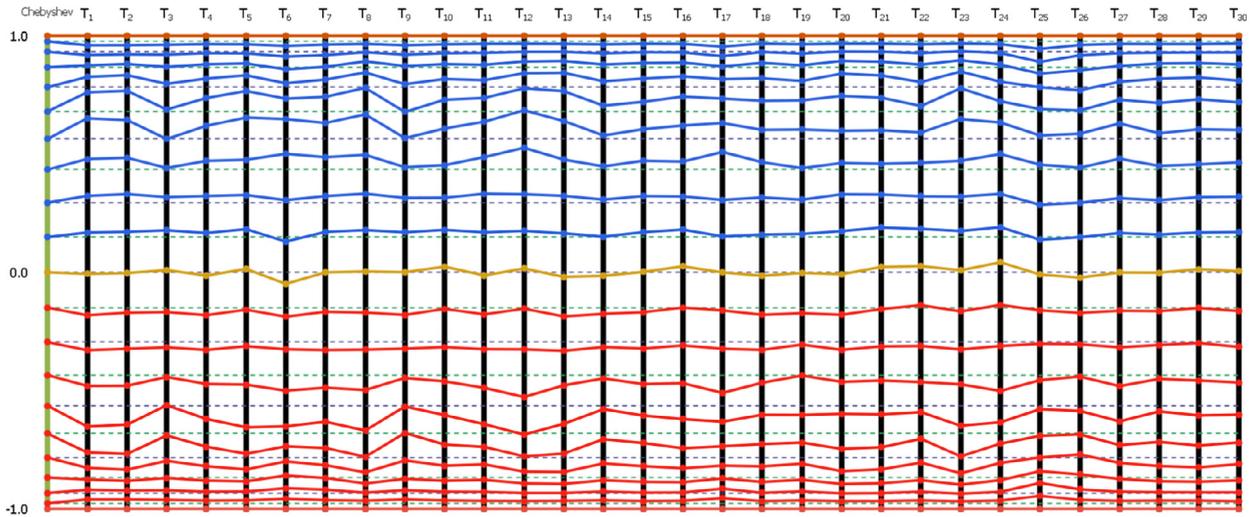
Moreover, to measure the fitness of each segment of the individual \mathbf{T} , we define the fitness functions E_I, E_{II} , and E_{III} for the segments $\mathbf{T}_I, \mathbf{T}_{II}$, and \mathbf{T}_{III} as follows:

$$E_I(\mathbf{T}_I) = \alpha \int_{t_0}^{t_{N_e-1}} \|\mathbf{P}(t) - \mathbf{I}(t)\|^2 dt + \beta \int_{t_0}^{t_{N_e-1}} \|\mathbf{P}'(t)\|^2 dt, \tag{16}$$

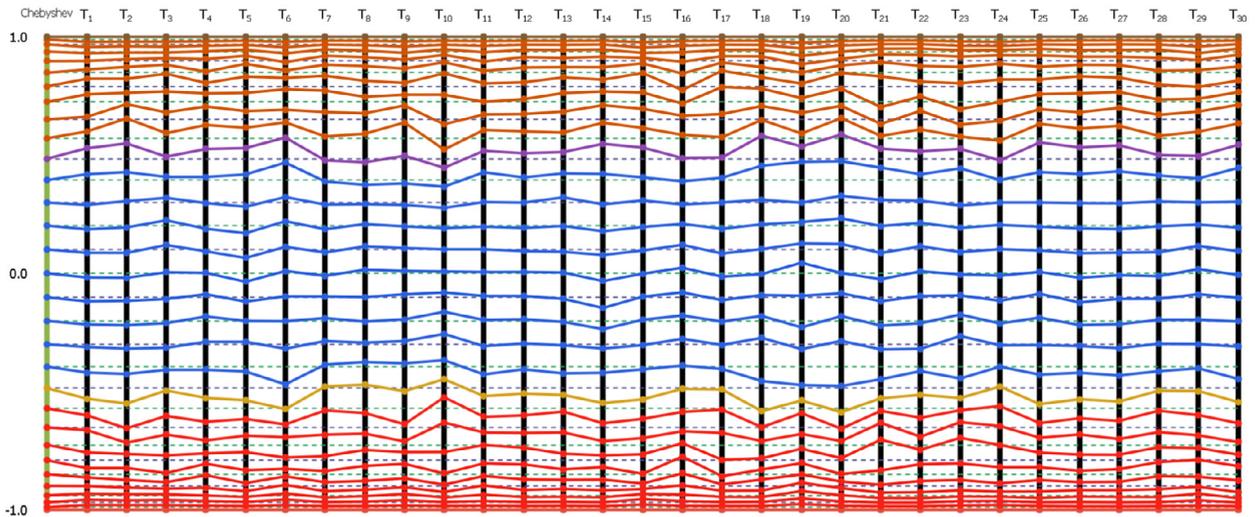
$$E_{II}(\mathbf{T}_{II}) = \alpha \int_{t_{N_e}}^{t_{N-N_e}} \|\mathbf{P}(t) - \mathbf{I}(t)\|^2 dt + \beta \int_{t_{N_e}}^{t_{N-N_e}} \|\mathbf{P}'(t)\|^2 dt, \tag{17}$$

$$E_{III}(\mathbf{T}_{III}) = \alpha \int_{t_{N-N_e+1}}^{t_N} \|\mathbf{P}(t) - \mathbf{I}(t)\|^2 dt + \beta \int_{t_{N-N_e+1}}^{t_N} \|\mathbf{P}'(t)\|^2 dt. \tag{18}$$

Crossover: First, the individuals chosen for crossover are randomly distributed into two arrays with the same size. Then, the two individuals in the two arrays with the same index are paired for the crossover operation. Suppose the individuals $(\mathbf{T}^{(1)}, \mathbf{T}^{(2)})$ are paired for the crossover operation. We develop two crossover operators. The first is the *weighted average* operator (Fig. 3(a)), and the second is the *extraction* operator (Fig. 3(b)). If the three segments of one individual, for example $\mathbf{T}^{(1)}$, are all better than



(a) $N = 20$.



(b) $N = 30$.

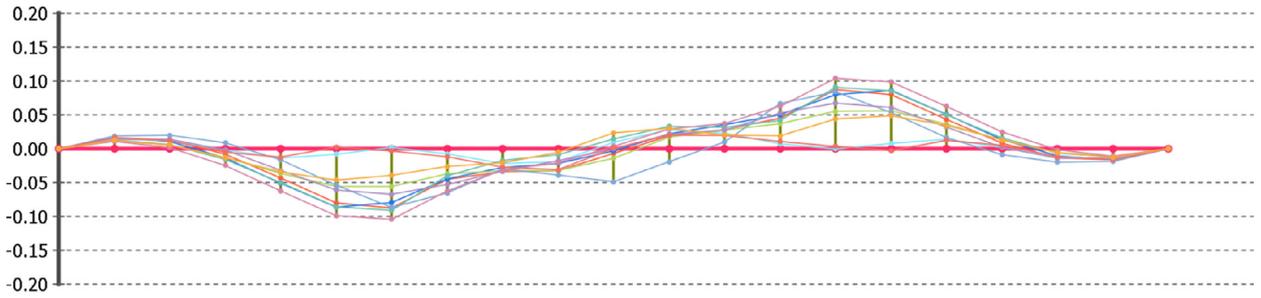
Fig. 4. Chebyshev parameter sequence and 30 best IGA parameter sequences with $N = 20$ (a) and $N = 30$ (b), respectively. The parameter sequences are illustrated as the points on the vertical line segments, and parameters with the same index are connected as a polyline.

those of the other individual $\mathbf{T}^{(2)}$, the new individual \mathbf{T}^{avg} is generated as the weighted average of them (Fig. 3), i.e.,

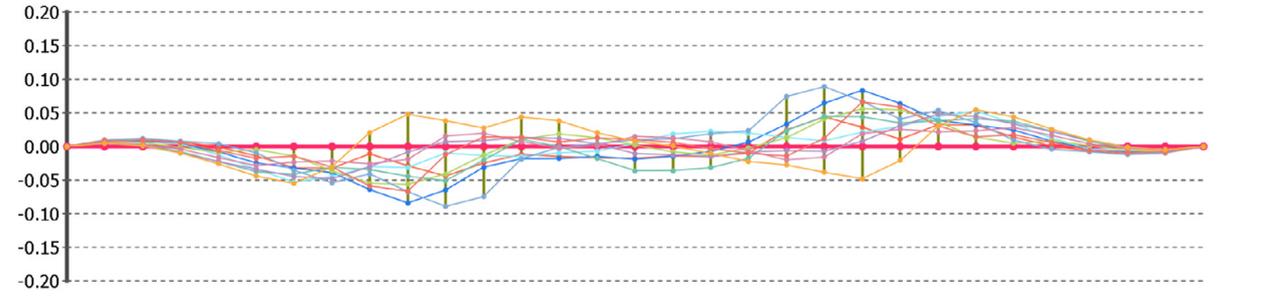
$$\mathbf{T}^{avg} = \left(1 - \frac{E(\mathbf{T}^{(1)})}{E(\mathbf{T}^{(1)}) + E(\mathbf{T}^{(2)})}\right) \mathbf{T}^{(1)} + \frac{E(\mathbf{T}^{(1)})}{E(\mathbf{T}^{(1)}) + E(\mathbf{T}^{(2)})} \mathbf{T}^{(2)}.$$

On the other hand, if not all of the segments of one individual are better than those of the other, we extract the better segments from the two individuals to form the new individual \mathbf{T}^{ext} . For example, segment $\mathbf{T}_I^{(1)}$ of $\mathbf{T}^{(1)}$ is better than segment $\mathbf{T}_I^{(2)}$ of $\mathbf{T}^{(2)}$, but segments $\mathbf{T}_{II}^{(1)}$ and $\mathbf{T}_{III}^{(1)}$ of $\mathbf{T}^{(1)}$ are worse than segments $\mathbf{T}_{II}^{(2)}$ and $\mathbf{T}_{III}^{(2)}$ of $\mathbf{T}^{(2)}$. Thus, we extract segment $\mathbf{T}_I^{(1)}$ of $\mathbf{T}^{(1)}$, and segments $\mathbf{T}_{II}^{(2)}$ and $\mathbf{T}_{III}^{(2)}$ of $\mathbf{T}^{(2)}$, to constitute the new individual \mathbf{T}^{ext} (Fig. 3(b)).

Mutation: In this paper, we develop two mutation operators. The number of individuals selected for each mutation operation is one second of the total number of individuals for mutation.



(a) $N = 20$.



(b) $N = 30$.

Fig. 5. Error distribution curves of the 30 best IGA parameter sequences with $N = 20$ (a) and $N = 30$ (b), respectively. The error distribution curve of a parameter sequence is a polyline. In this figure, the points on the x -axis indicate the indices of the parameters, and the y -axis denotes the error between each parameter of an best IGA parameter sequence and the parameter of the Chebyshev parameter sequence that has the same index.

The first mutation operator increases the diversity of the population. We randomly choose some parameters t_i from a selected individual T , avoiding the two end points t_0 and t_N , and perform the linear combination

$$t_i^{new} = \begin{cases} (1 - \lambda)t_{i-1} + \lambda(-t_m), & i = N_e - 1, \\ (1 - \lambda)(-t_m) + \lambda t_{i+1}, & i = N_e, \\ (1 - \lambda)t_{i-1} + \lambda t_m, & i = N - N_e, \\ (1 - \lambda)t_m + \lambda t_{i+1}, & i = N - N_e + 1, \\ (1 - \lambda)t_{i-1} + \lambda t_{i+1} & otherwise, \end{cases}$$

where $0 < \lambda < 1$ is a random number.

The second mutation operator is intended to interpolate the symmetric data point set, e.g., the point set $\{P_i, i = 0, 1, \dots, N\}$ (6) considered in this paper. Specifically, for a selected individual T , we first check the fitness of segments T_I and T_{III} , using Eqs. (16) and (18), respectively. If the fitness of segment T_I is less than that of segment T_{III} , segment T_{III} is replaced by the mirror transformation of T_I , i.e.,

$$t_{N-i}^{new} = -t_i, \quad i = 0, 1, \dots, N_e - 1.$$

However, if the fitness of segment T_{III} is less than that of T_I , segment T_I is replaced by the mirror transformation of T_{III} , i.e.,

$$t_i^{new} = -t_{N-i}, \quad i = 0, 1, \dots, N_e - 1.$$

Vaccination:

Finally, a vaccination operation is performed on the resulting individuals to generate the next iteration of the population. Vaccination means that individuals with higher fitness values are replaced by some individuals in the immunological memory.

In this way, the next iteration of the population is generated.

Step 5: Termination condition:

If the maximum generation is reached, or the smallest fitness value of the current generation is less than a predefined threshold ε_t , the algorithm terminates. Otherwise, we return to Step 2.

5. Results and discussion

The IGA-based optimal parameter search algorithm described above has been tested using different numbers of sampling points, i.e., different values of N in Eq. (6). In this section, we present experimental data with $N = 20$ and $N = 30$.

Table 2

The six best IGA parameter sequences and Chebyshev parameter sequence when $N = 20$.

Sequence	Chebyshev	T_1	T_2	T_3	T_4	T_5	T_6
	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
	-0.9777	-0.9612	-0.9614	-0.9632	-0.9662	-0.9652	-0.9587
	-0.9335	-0.9201	-0.9227	-0.9219	-0.9280	-0.9272	-0.9134
	-0.8685	-0.8766	-0.8822	-0.8703	-0.8814	-0.8843	-0.8597
	-0.7840	-0.8273	-0.8350	-0.7979	-0.8206	-0.8341	-0.8004
	-0.6821	-0.7621	-0.7682	-0.6902	-0.7375	-0.7679	-0.7359
	-0.5649	-0.6524	-0.6443	-0.5631	-0.6203	-0.6556	-0.6516
	-0.4351	-0.4803	-0.4797	-0.4427	-0.4722	-0.4751	-0.5004
	-0.2956	-0.3294	-0.3233	-0.3174	-0.3280	-0.3128	-0.3253
	-0.1495	-0.1819	-0.1712	-0.1684	-0.1816	-0.1582	-0.1883
	0.0000	-0.0069	-0.0031	0.0095	-0.0141	0.0144	-0.0488
	0.1495	0.1685	0.1715	0.1778	0.1666	0.1826	0.1304
	0.2956	0.3238	0.3311	0.3188	0.3231	0.3273	0.3057
	0.4351	0.4803	0.4853	0.4424	0.4722	0.4769	0.5020
	0.5649	0.6524	0.6448	0.5646	0.6203	0.6556	0.6491
	0.6821	0.7621	0.7684	0.6900	0.7381	0.7677	0.7354
	0.7840	0.8273	0.8346	0.7980	0.8203	0.8341	0.8004
	0.8685	0.8766	0.8820	0.8703	0.8810	0.8844	0.8597
	0.9335	0.9201	0.9229	0.9218	0.9280	0.9272	0.9136
	0.9777	0.9612	0.9616	0.9632	0.9662	0.9652	0.9590
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
E_{error}	0.0003	0.0004	0.0004	0.0004	0.0004	0.0004	0.0005
E_{smooth}	1069	91.96	98.47	99.24	100.4	103.3	103.9

Table 3

The six best IGA parameter sequences and Chebyshev parameter sequence when $N = 30$.

Sequence	Chebyshev	T_1	T_2	T_3	T_4	T_5	T_6
	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
	-0.9897	-0.9802	-0.9822	-0.9822	-0.9826	-0.9849	-0.9805
	-0.9693	-0.9581	-0.9612	-0.9593	-0.9634	-0.9673	-0.9573
	-0.9390	-0.9319	-0.9353	-0.9353	-0.9404	-0.9466	-0.9311
	-0.8990	-0.8997	-0.9068	-0.9112	-0.9075	-0.9215	-0.8947
	-0.8497	-0.8636	-0.8742	-0.8835	-0.8544	-0.8872	-0.8592
	-0.7918	-0.8236	-0.8238	-0.8448	-0.8059	-0.8330	-0.8273
	-0.7257	-0.7580	-0.7645	-0.7703	-0.7619	-0.7567	-0.7794
	-0.6522	-0.6631	-0.7164	-0.6821	-0.7067	-0.6865	-0.6930
	-0.5720	-0.6007	-0.6558	-0.6034	-0.6282	-0.6163	-0.6392
	-0.4859	-0.5298	-0.5506	-0.4956	-0.5268	-0.5367	-0.5750
	-0.3949	-0.4196	-0.4257	-0.4083	-0.4079	-0.4151	-0.4694
	-0.2997	-0.3112	-0.3180	-0.3146	-0.2891	-0.2893	-0.3182
	-0.2016	-0.2164	-0.2195	-0.2110	-0.1828	-0.2016	-0.2026
	-0.1013	-0.1181	-0.1160	-0.1082	-0.0884	-0.1193	-0.0971
	0.0000	-0.0174	-0.0189	0.0043	0.0020	-0.0359	0.0100
	0.1013	0.08731	0.0869	0.1201	0.09333	0.06547	0.1126
	0.2016	0.1865	0.1944	0.2241	0.1888	0.1700	0.2206
	0.2997	0.2902	0.3062	0.3197	0.2977	0.2822	0.3231
	0.3949	0.4196	0.4283	0.4083	0.4079	0.4187	0.4694
	0.4859	0.5298	0.5504	0.4943	0.5268	0.5307	0.5750
	0.5720	0.6007	0.6558	0.5934	0.6282	0.6163	0.6392
	0.6522	0.6631	0.7163	0.6821	0.7067	0.6872	0.6930
	0.7257	0.7580	0.7645	0.7703	0.7619	0.7641	0.7794
	0.7918	0.8236	0.8238	0.8448	0.8059	0.8330	0.8273
	0.8497	0.8636	0.8742	0.8835	0.8544	0.8874	0.8592
	0.8990	0.8997	0.9068	0.9112	0.9075	0.9216	0.8947
	0.9390	0.9319	0.9353	0.9353	0.9404	0.9466	0.9311
	0.9693	0.9581	0.9612	0.9593	0.9634	0.9672	0.9573
	0.9897	0.9802	0.9822	0.9822	0.9826	0.9847	0.9805
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
E_{error}	6.991e-05	1.063e-04	1.369e-04	1.212e-04	7.574e-005	1.158e-04	1.528e-04
E_{smooth}	4619	181.8	197.9	243.0	261.4	281.7	284.8

In the case of $N = 20$, the 30 best IGA parameter sequences are generated by running the IGA-based search algorithm 30 times. Each best IGA parameter sequence T is produced after 300 evolutions with a population size of 200. The three subintervals are taken as $[-1, -\frac{1}{3}]$, $(-\frac{1}{3}, \frac{1}{3})$, and $[\frac{1}{3}, 1]$, i.e., $t_m = \frac{1}{3}$ (13). The numbers of parameters in the three subsegments T_I , T_{II} , and T_{III} are 8, 5, 8, respectively, i.e., $N_e = 8$ (14). The parameters employed in the IGA-based search algorithm are listed in Table 1.

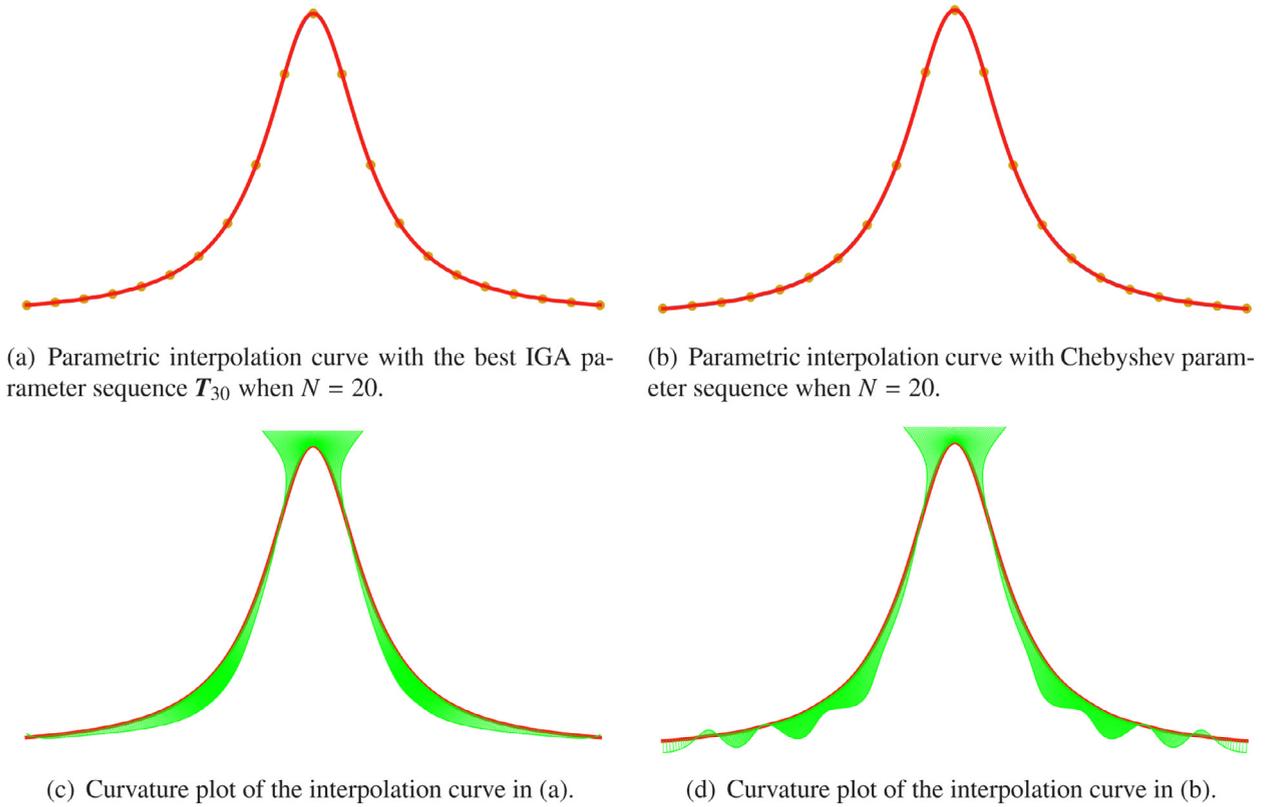


Fig. 6. The parametric interpolation curves with the best IGA parameter sequence T_{30} (a) and Chebyshev parameter sequence when $N = 20$, and their curvature plots (c,d). In this figure, the parametric interpolation curve is in red, and the curve of $f(x) = 1/(1 + 25x^2)$, $x \in [-1, 1]$ is in blue. (Note that the interpolation curve (in red) totally overlaps the curve of $f(x)$ (in blue).) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

Moreover, the Chebyshev parameter sequence and the 30 best IGA parameter sequences T_1, T_2, \dots, T_{30} with $N = 20$ are illustrated in Fig. 4(a). In this figure, the Chebyshev parameter sequence is represented by the points on the leftmost vertical line segment, and the best IGA parameter sequence T_i is represented as the points on the vertical line segment, labeled T_i . The 30 best IGA parameter sequences T_i , $i = 1, 2, \dots, 30$ are arranged in ascending order of their energy $E(T_i)$ (12). Furthermore, parameters with the same index are connected as a polyline, resulting in 21-piece polylines.

It can be seen from these 21-piece polylines that the best IGA parameter sequences closely oscillate around the Chebyshev parameter sequence. This conclusion is more clearly validated by the error distribution of the deviation between the best IGA parameter sequence T_i and the Chebyshev parameter sequence, as shown in Fig. 5(a). If the Chebyshev parameter sequence is $T_c = \{t_0^c, t_1^c, \dots, t_N^c\}$, and the best IGA parameter sequence T_i is $T_i = \{t_0^i, t_1^i, \dots, t_N^i\}$, the error distribution of T_i is defined as

$$error_i = \{t_0^i - t_0^c, t_1^i - t_1^c, \dots, t_N^i - t_N^c\}. \tag{19}$$

In Fig. 5(a), the points on the x-axis indicate the indices of the parameters in the parameter sequence, and the y-axis is the error given by Eq. (19). Thus, the error distributions $error_i$ are illustrated as polylines in Fig. 5(a). These again indicate that the best IGA parameter sequences T_i , $i = 1, 2, \dots, 20$ oscillate around the Chebyshev parameter sequence, and the errors are not more than 0.10. The error distribution curves presented in Fig. 5(a) exhibit a clear symmetrical structure, which is caused by the second mutation operator.

Furthermore, in the case of $N = 30$, the IGA-based search algorithm is run 30 times to generate 30 best IGA parameter sequences. Each best IGA parameter sequence is produced after 500 evolutions with a population size of 200. In this case of $N = 30$, $t_m = \frac{1}{3}$ (13), i.e., the interval $[-1, 1]$ is subdivided into three subintervals, $[-1, -\frac{1}{3}]$, $(-\frac{1}{3}, \frac{1}{3})$, $[\frac{1}{3}, 1]$, and the numbers of parameters in the three subsegments T_I , T_{II} , and T_{III} are 12, 7, 12, respectively. In other words, $N_e = 12$ (14). The parameters employed in the IGA-based search algorithm for the case $N = 30$ can be found in Table 1.

Similarly, the 30 best IGA parameter sequences with $N = 30$ and the corresponding Chebyshev parameter sequence are illustrated in Fig. 4(b), and their error distribution curves are shown in Fig. 5(b). These again validate the conclusion that the optimal parameter sequences oscillate around the Chebyshev parameter sequence. Therefore, the Chebyshev parameter sequence is most likely the globally optimal parameter sequence for defeating the Runge phenomenon.

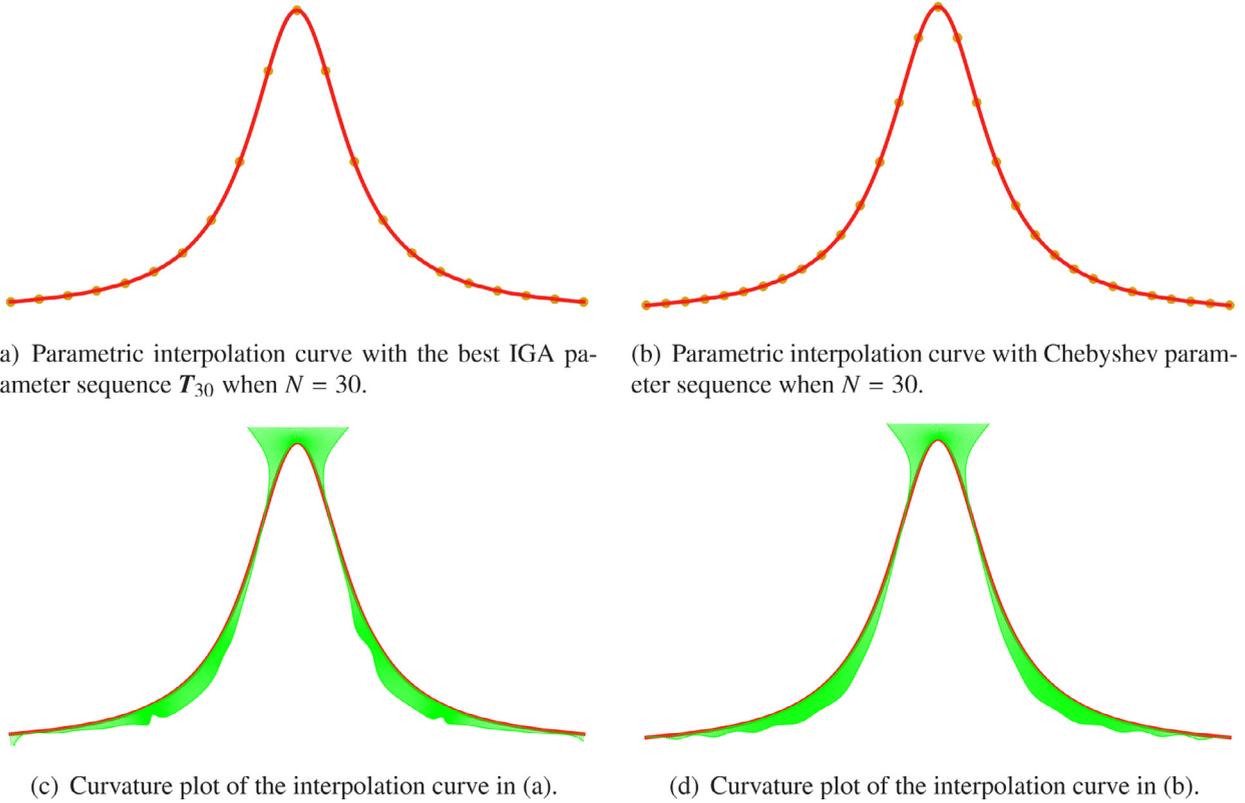


Fig. 7. The parametric interpolation curves with the best IGA parameter sequence T_{30} (a) and Chebyshev parameter sequence (b) when $N = 30$, and their curvature plots (c, d). In this figure, the parametric interpolation curve is in red, and the curve of $f(x) = 1/(1 + 25x^2)$, $x \in [-1, 1]$ is in blue. (Note that the interpolation curve (in red) totally overlaps the curve of $f(x)$ (in blue).) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

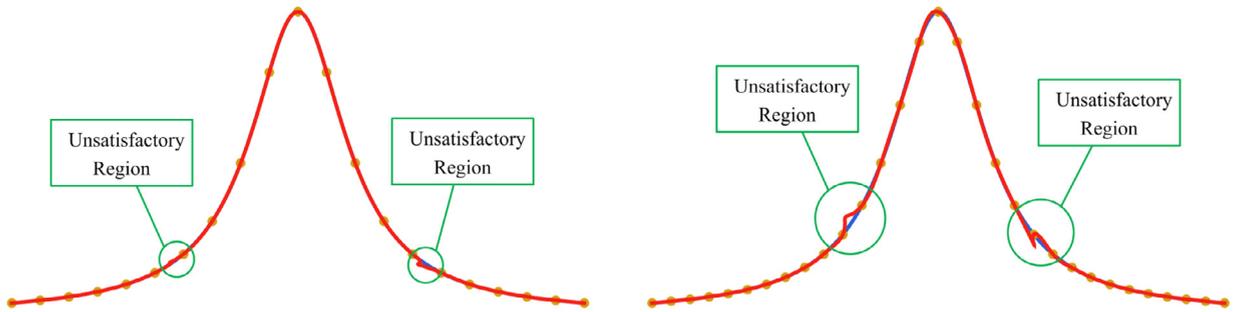
The six best IGA parameter sequences, i.e., T_1, T_2, \dots, T_6 , and Chebyshev parameter sequence, with $N = 20$ and $N = 30$, respectively, are listed in Tables 2 and 3.

To give an intuitive impression, we illustrate the parametric interpolation curves $P(t)$ with best IGA parameter sequence T_{30} and Chebyshev parameter sequence, when $N = 20$ (in Fig. 6), and $N = 30$ (in Fig. 7), respectively. In Figs. 6 and 7(a) and (b) are the parametric curves with optimal parameter sequence T_{30} and Chebyshev parameter sequence, respectively, and (c) and (d) are the curvature plots of the parametric curves in (a) and (b). Curvature plot is a frequently-used tool in geometric design and related fields to evaluate the smoothness of a parametric curve [33]. While the smoothness of the parametric curve with the best IGA parameter sequence T_{30} (Fig. 6(c)) is better than that of the parameter curve with Chebyshev parameter sequence (Fig. 6(d)) when $N = 20$, the smoothness of the curve with T_{30} (Fig. 7(c)) is worse than that of the curve with Chebyshev parameter sequence (Fig. 7(d)) when $N = 30$.

Finally, it should be pointed out that an important factor influencing the parametric interpolation curve is the numbers of parameters assigned to the three subsegments T_I, T_{II} , and T_{III} . For the Chebyshev parameter sequence, the three subsegments have 8, 5, 8 parameters ($N_e = 8$) when $N = 20$, and 12, 7, 12 parameters ($N_e = 12$) when $N = 30$. The best IGA parameter sequences obtained in this paper are determined for $N_e = 8$ and $N_e = 12$, same as Chebyshev parameter sequences. However, if we change the numbers of parameters in the three subsegments a little, the best IGA parameter sequence generated by the IGA-based search algorithm makes the shape of the parametric interpolation curve unsatisfactory. For example, in the case of $N = 20$, we change the numbers of parameters in the three subsegments T_I, T_{II} , and T_{III} to 7,7,7 (Fig. 8(a)); in the case of $N = 30$, we change them to 11,9,11 (Fig. 8(b)). As illustrated in Fig. 8, the best IGA parameter sequences generated with so little changes make the shapes of the parametric interpolation curves unsatisfactory near the two end points of T_{II} .

5.1. Influence of the parameters used in IGA

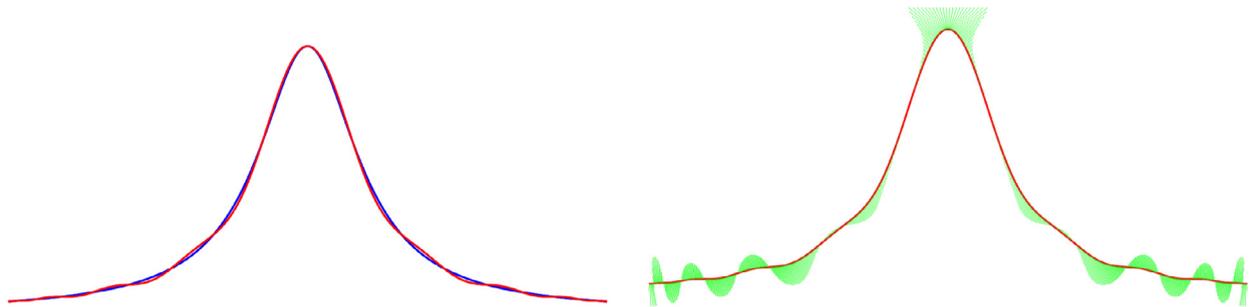
To demonstrate the influence of the parameters used in IGA, we take the experiment with parameters listed in the second column of Table 1 (i.e., the column beginning with $N = 20$) as a benchmark, and change the parameters one by one while the other parameters are fixed. The data E_{error} and E_{smooth} of the generated solution corresponding to each set of parameters are listed in Tables A.4–A.12 in Appendix A. Specifically, for each set of parameters, we run the IGA 10 times, generating 10 solutions with



(a) The parametric interpolation curve with the best IGA parameter sequence where the numbers of the parameters in the three sub-segments are 7, 7, 7.

(b) The parametric interpolation curve with the best IGA parameter sequence where the numbers of the parameters in the three sub-segments are 11, 9, 11.

Fig. 8. The shape of the parametric interpolation curve is made unsatisfactory if the numbers of the parameters in the three subsegments are different from those of the Chebyshev parameter sequence.



(a) The result (in red) of the mock-Chebyshev method and the curve (in blue) of $f(x) = 1/(1 + 25x^2)$.

(b) The curvature plot of the mock-Chebyshev result.

Fig. 9. The result of the mock-Chebyshev method with $E_{error} = 0.0014$ and $E_{smooth} = 368.1010$.

10 sets of E_{error} and E_{smooth} . The data E_{error} and E_{smooth} which are the closest to the averages of the 10 sets of E_{error} and E_{smooth} are listed in Tables A.4–A.12.

From the data listed in Tables A.4–A.12, we can see that the parameter with the most influence on IGA is the number of parameters (N) (i.e., number of samples). Refer to Table A.4, when N changes from 20 to 40, E_{error} varies from 4.7531×10^{-4} to 1.4223×10^{18} , and E_{smooth} from 2.1732×10^2 to 1.2601×10^{23} . Next, the parameter with the second most influence on IGA is N_e , the number of parameters in the two end subintervals. Refer to Table A.9, when N_e varies from 5 to 9, E_{error} alters from 94.339 to 5.0184×10^4 , and E_{smooth} from 1.6438×10^6 to 1.1771×10^2 . In addition, the other parameters have relatively small influences on IGA. With them, E_{error} is in the order of magnitude of 10^{-4} , and E_{smooth} is in the order of magnitude of 10^2 .

5.2. Comparison with existing methods

In general, the Runge phenomenon defeating methods can be categorized into the following classes [7]: regularization methods [6], least-squares fitting methods [8], methods using non-polynomial basis [13,14] (such as radial basis functions, Fourier extensions, etc.), methods mimicking the Chebyshev sampling [8] (such as mock-Chebyshev method), etc. The aforementioned methods usually construct a definite solution that can defeat the Runge phenomenon. However, they cannot answer the question which the best solution for defeating the Runge phenomenon is. For this purpose, the objective of the IGA-based method developed in this paper is to search the best solution for defeating the Runge phenomenon.

Because the IGA-based method developed in this paper and the mock-Chebyshev method [8] both interpolate the samples using polynomials, we compared the IGA-based method with the mock-Chebyshev method. The mock-Chebyshev method chooses a subset from equidistant samples where the samples in the subset approach the Chebyshev samples with the same number as close as possible. And then, the polynomial interpolator to the samples in the subset can defeat the Runge phenomenon. In our implementation, we equidistantly take 100 samples from the interval $[-1, 1]$, and select 21 samples that are closest to the 21 Chebyshev samples. Fig. 9 illustrates the results of the mock-Chebyshev method with $E_{error} = 0.0014$ and $E_{smooth} = 368.1010$. As stated above, the solution of the mock-Chebyshev method can defeat the Runge phenomenon, but it is not the best solution for defeating the Runge phenomenon. Compare with the result generated by the IGA-based method

Table A.4

The influence of the number of parameters (N) (i.e., the number of samples).

#Parameters	$N = 20$	$N = 25$	$N = 30$	$N = 35$	$N = 40$
E_{error}	4.7531e-04	4.2432e-02	2.1303e+04	2.603e+09	1.4223e+18
E_{smooth}	2.1732e+02	3.136e+03	1.7744e+08	3.0125e+15	1.2601e+23

Table A.5

The influence of the population size (M).

Population size	$M = 150$	$M = 200$	$M = 250$	$M = 300$	$M = 350$
E_{error}	3.9040e-04	4.7531e-04	4.1485e-04	4.5459e-04	4.3165e-04
E_{smooth}	1.7099e+02	2.1732e+02	1.7401e+02	1.7870e+02	1.6215e+02

Table A.6

The influence of the maximum generation.

Max. generation	200	250	350	400	450
E_{error}	4.7531e-04	3.8225e-04	3.9804e-04	3.8673e-04	4.2971e-04
E_{smooth}	2.1732e+02	2.3100e+02	1.7318e+02	1.4968e+02	1.5958e+02

Table A.7

The influence of the selection rate, cross rate and mutation rate.

Selection rate	0.2	0.3	0.4	0.6	0.8
Cross rate	0.1	0.15	0.2	0.3	0.4
Mutation rate	0.1	0.15	0.2	0.3	0.4
E_{error}	6.1174e-04	4.6655e-04	4.1482e-04	4.7531e-04	4.2988e-04
E_{smooth}	3.1001e+02	2.7779e+02	2.1412e+02	2.1732e+02	1.4328e+02

Table A.8

The influence of t_m (Eq. (13)).

t_m	1/6	1/3	1/2	2/3	5/6
E_{error}	4.7145e-04	4.7531e-04	3.9713e-04	4.1331e-04	3.9450e-04
E_{smooth}	3.0343e+02	2.1732e+02	2.2611e+02	1.6654e+02	2.1222e+02

Table A.9

The influence of N_e (Eq. (14)).

N_e	5	6	7	8	9
E_{error}	9.4339e+01	7.6948e-02	1.6742e-03	4.7531e-04	5.0184e-04
E_{smooth}	1.6438e+06	2.1672e+03	5.9903e+02	2.1732e+02	1.1771e+02

Table A.10

The influence of (α, β) (Eq. (12)).

(α, β)	(10,1)	(100,1)	(1000,1)	(10000,1)	(100000,1)
E_{error}	4.8626e-04	5.7086e-04	4.7531e-04	4.8865e-04	4.9179e-04
E_{smooth}	2.5860e+02	2.8965e+02	2.1732e+02	2.8043e+02	2.7554e+02

(Fig. 6(a) and (c)), which interpolates 21 equidistant samples, with $E_{error} = 0.0008$ and $E_{smooth} = 152.6$. Clearly, the solution generated by the IGA-based method is better than that by the mock-Chebyshev method.

6. Conclusion

In this paper, we developed an IGA-based optimal parameter search algorithm for defeating the Runge phenomenon. Unlike traditional methods, which attempt to defeat the Runge phenomenon using polynomial interpolation, we employed a parametric interpolation curve. First, the Runge phenomenon was modeled as an optimization problem, with the objective function given by the energy of the parametric curve. Because this objective function is highly non-linear, traditional optimization methods are inefficient or invalid. Thus, we developed an IGA-based search algorithm to seek the globally optimal parameter sequence that allows the parametric curve to defeat the Runge phenomenon. It is shown that the optimal parameter sequences oscillate closely

Table A.11
The influence of α_r (Eq. (15)).

α_r	0.02	0.03	0.04	0.05	0.06
E_{error}	4.0078e-04	3.8653e-04	4.2457e-04	4.7531e-04	6.5504e-04
E_{smooth}	1.5467e+02	1.8305e+02	1.7349e+02	2.1732e+02	1.9344e+02

Table A.12
The influence of the number of individuals in the immunological memory.

#Individuals	1	2	3	4	5
E_{error}	4.6612e-04	4.7531e-04	4.0224e-04	3.9845e-04	4.6836e-04
E_{smooth}	1.7466e+02	2.1732e+02	1.8200e+02	1.3966e+02	1.9034e+02

around the Chebyshev parameter sequence. So the Chebyshev parameter sequence is most likely the globally optimal parameter sequence.

Acknowledgment

This paper is supported by National Natural Science Foundation of China (nos. 61379072, 60970150).

Appendix A. The influence of the parameters in IGA

In this appendix, we list the data E_{error} and E_{smooth} of the solutions generated by changing the parameters employed in IGA one by one in Tables A.4–A.10.

References

- [1] C. Runge, Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten, Z. Math. Phys. 46 (224–243) (1901) 20.
- [2] C. Méray, Observations sur la légitime de l'interpolation, Ann. Sci. Econ. Norm. Super. 1 (1884) 165–176.
- [3] C. Méray, Nouvelles exemples d'interpolation illusoire, Bull. Sci. Math. 20 (1896) 266–270.
- [4] J.F. Epperson, On the Runge example, Am. Math. Monthly 94 (4) (1987) 329–341.
- [5] T. Mills, S.J. Smith, Lagrange interpolation polynomials based equidistant nodes, Math. Sci. 16 (1991) 107–117.
- [6] J.P. Boyd, Defeating the Runge phenomenon for equispaced polynomial interpolation via Tikhonov regularization, Appl. Math. Lett. 5 (1992) 57–59.
- [7] J.P. Boyd, J.R. Ong, Exponentially-convergent strategies for defeating the Runge phenomenon for the approximation of non-periodic functions, Part I: single-interval schemes, Commun. Comput. Phys. 5 (2–4) (2009) 484–497.
- [8] J.P. Boyd, F. Xu, Divergence (Runge phenomenon) for least-squares polynomial approximation on an equispaced grid and mock-Chebyshev subset interpolation, Appl. Math. Comput. 210 (1) (2009) 158–168.
- [9] G. Stengle, Chebyshev interpolation with approximate nodes of unrestricted multiplicity, J. Approx. Theory 57 (1) (1989) 1–13.
- [10] J.P. Boyd, Exponentially accurate Runge-free approximation of non-periodic functions from samples on an evenly spaced grid, Appl. Math. Lett. 20 (9) (2007) 971–975.
- [11] R.B. Platte, A. Gelb, A hybrid Fourier–Chebyshev method for partial differential equations, J. Sci. Comput. 39 (2) (2009) 244–264.
- [12] E. Rakhmanov, Bounds for polynomials with a unit discrete norm, Ann. Math. (2007) 55–88.
- [13] R.B. Platte, T.A. Driscoll, Polynomials and potential theory for Gaussian radial basis function interpolation, SIAM J. Numer. Anal. 43 (2) (2005) 750–766.
- [14] B. Fornberg, C. Piret, A stable algorithm for flat radial basis functions on a sphere, SIAM J. Sci. Comput. 30 (1) (2007) 60–80.
- [15] D. Gottlieb, C.-W. Shu, On the Gibbs phenomenon and its resolution, SIAM Rev. 39 (4) (1997) 644–668.
- [16] J.P. Boyd, Trouble with Gegenbauer reconstruction for defeating Gibbs phenomenon: Runge phenomenon in the diagonal limit of Gegenbauer polynomial approximations, J. Comput. Phys. 204 (1) (2005) 253–264.
- [17] R.B. Platte, L.N. Trefethen, A.B. Kuijlaars, Impossibility of fast stable approximation of analytic functions from equispaced samples, SIAM Rev. 53 (2) (2011) 308–318.
- [18] J.P. Boyd, J.R. Ong, Exponentially-convergent strategies for defeating the Runge phenomenon for the approximation of non-periodic functions, Part two: Multi-interval polynomial schemes and multidomain Chebyshev interpolation, Appl. Numer. Math. 61 (4) (2011) 460–472.
- [19] J. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, MI, 1975.
- [20] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (2) (1988) 95–99.
- [21] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Professional, 1989.
- [22] D. Beasley, R. Martin, D. Bull, An overview of genetic algorithms: Part 1. Fundamentals, Univ. Comput. 15 (1993) 58–58.
- [23] P. Bentley, An introduction to evolutionary design by computers, Evol. Des. Comput. 1 (1999) 1–79.
- [24] L. Jiao, L. Wang, A novel genetic algorithm based on immunity, IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. 30 (5) (2000) 552–561.
- [25] W. Abd-El-Wahed, A. Mousa, M. El-Shorbagy, Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems, J. Comput. Appl. Math. 235 (5) (2011) 1446–1453.
- [26] K. Gallagher, M. Sambridge, Genetic algorithms: a powerful tool for large-scale nonlinear optimization problems, Comput. Geosci. 20 (7) (1994) 1229–1236.
- [27] F. Yoshimoto, T. Harada, Y. Yoshimoto, Data fitting with a spline using a real-coded genetic algorithm, Comput. Aided Des. 35 (8) (2003) 751–760.
- [28] J.-T. Tsai, W.-H. Ho, T.-K. Liu, J.-H. Chou, Improved immune algorithm for global numerical optimization and job-shop scheduling problems, Appl. Math. Comput. 194 (2) (2007) 406–424.
- [29] X. Zhao, C. Zhang, L. Xu, B. Yang, Z. Feng, IGA-based point cloud fitting using B-spline surfaces for reverse engineering, Inform. Sci. 245 (2013) 276–289.
- [30] M. Matsumoto, T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator, ACM Trans. Model. Comput. Simulat. 8 (1) (1998) 3–30.
- [31] W. Gander, W. Gautschi, Adaptive quadrature-revisited, BIT Numer. Math. 40 (1) (2000) 84–101.
- [32] R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, John Wiley & Sons, 2004.
- [33] N.S. Sapidis, G.D. Koras, Visualization of curvature plots and evaluation of fairness: an analysis of the effect of 'scaling', Comput. Aided Geometr. Des. 14 (4) (1997) 299–311.