Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 33(3): 1836–1857.
DOI: 10.3934/era.2025083
Received: 22 December 2024
Revised: 26 February 2025
Accepted: 13 March 2025
Published: 31 March 2025

Research article

An optimized potential formula of the $m \times n$ apple surface network and its application of potential in path planning

Yangming Xu[1], Yanpeng Zheng[1,*], Xiaoyu Jiang[2,*], Zhaolin Jiang[3] and Zhibin Liu[1]

[1] School of Automation and Electrical Engineering, Linyi University, Linyi 276000, China
[2] School of Information Science and Engineering, Linyi University, Linyi 276000, China
[3] School of Mathematics and Statistics, Linyi University, Linyi 276000, China

* **Correspondence:** Email: zhengyanpeng0702@sina.com, jxy19890422@sina.com.

Abstract: An optimized potential formula for the $m \times n$ apple surface network has been introduced in this paper. Compared with the original potential formula, this method significantly enhances the efficiency required for rapid and large-scale numerical simulations. Based on the optimized potential function, we proposed a metaheuristic algorithm suitable for apple surface environment path planning. Chebyshev polynomials of the first class were employed to represent the potential function. Subsequently, a fast algorithm for calculating the potential utilizing the first kind of discrete sine transform (DST-I) was devised. We proposed potential formulas for several cases to visually present the distribution of the potential and illustrated them using three-dimensional graphs. We also conducted simulation experiments on the computational efficiency of the original and optimized formulas at different data scales, verifying the superiority of the optimized formulas. These findings provided new perspectives and tools for the computation of resistor networks and the design of path planning algorithms. Experiments were conducted to analyze the efficiency and availability of various techniques for computing potential.

Keywords: resistor network; potential; path planning; tridiagonal matrices

1. Introduction

The method of constructing resistor network models to simulate problems has been applied across various fields. Many studies have shown that establishing a resistor network model can effectively solve numerous complex problems [1].

Since the introduction of Kirchhoff's laws [2], significant progress has been made in the study of resistor networks. Subsequently, many research methods emerged based on this. These include the Lattice Green's functions method [3] for solving infinite networks, Perrier and Girault use elementary

methods for solving symmetric bracelet resistor networks [4] and rotational invariance in resistor networks [5], and Wu's Laplacian matrix method [6] for solving arbitrary networks. Tan [7] creatively came up with the innovative recursive transformation (RT) method, which is essential for figuring out the equivalent resistance and potential of resistor networks. Among the RT methods are the branch current-based RT-I method and the node voltage-based RT-V method. Tan's approach, which differs from the Laplacian matrix method, is simpler and more accurate, making his RT method the foundation for most studies of resistor networks. Furthermore, Luo and Luo [8] conducted in-depth research on LC networks and cascade circuit networks. It can be seen from the literature that the tridiagonal Toeplitz matrix and the perturbed tridiagonal Toeplitz matrix play a key role in the modeling of resistance networks. To date, tridiagonal matrices have been undergoing extensive research [9, 10]. Moreover, the resistance distance is another name for the equivalent resistance in graph theory [11]. As resistor networks demonstrate high value in various scientific fields, applying them in engineering will be particularly important and significant. To advance this work, achieving rapid computation of large-scale resistor networks is necessary. Jiang et al. [12, 13] have been dedicated to this work.

Furthermore, planning optimal paths [14, 15] has been consistently a challenging task. Path planning is widely used today in fields like autonomous driving [16], robot path planning [17], and UAV trajectory optimization and control [18]. Extensive research on path planning for particular forms has produced ongoing new findings. For instance, Kulathunga [19] and Mazaheri [20] have studied path planning in the 3D environment, Xue et al. [21] has studied path planning on cylindrical storage tanks, and UGUR has studied path planning on cuboids [22]. The significance of path design for special shapes has been amply illustrated by these studies.

In 2023, Tan proposed an apple surface network [23], which connects the two poles of a sphere to one pole. Due to the shape of this structure resembling an apple, Tan [23] named this model the apple surface network, as shown in Figure 1. In this paper, the original potential formula is reformulated with the first kind of Chebyshev polynomials to enhance computational efficiency along with numerical algorithms developed for calculating the potential in large-scale resistor networks. Figure 1 is called an arbitrary $m \times n$ apple surface network model, where m (excluding the pole 0) and n are the number of nodes along the longitude and latitude directions (the circular lines latitude), and r_1 and r are the resistances in the respective longitude and latitude directions. Assuming the current J is constant, it flows from the input point $d_1(x_1, y_1)$ to the output point $d_2(x_2, y_2)$, where the voltage in the local area of the network are shown in Figure 2. The potential formula of any point $d(x, y)$ in an $m \times n$ apple surface network is as follows

$$\frac{\mathbf{U}_{m\times n}(x,y)}{J} = \frac{2r}{m+1}\sum_{i=1}^{m}\frac{g^{(i)}_{x_1,x}\sin(y_1\theta_i) - g^{(i)}_{x_2,x}\sin(y_2\theta_i)}{\lambda_i^n + \overline{\lambda}_i^n - 2}\sin(y\theta_i), \tag{1.1}$$

where $\theta_i = i\pi/(m+1)$ and

$$g^{(i)}_{x_s,x} = F_{n-|x_s-x|} + F_{|x_s-x|}, s = 1, 2 \tag{1.2}$$

$$F^{(i)}_k = \frac{(\lambda_i^k - \overline{\lambda}_i^k)}{(\lambda_i - \overline{\lambda}_i)}, \tag{1.3}$$

$$\begin{aligned}\lambda_i &= 1 + \nu - \nu\cos\theta_i + \sqrt{(1+\nu-\nu\cos\theta_i)^2 - 1},\\ \overline{\lambda}_i &= 1 + \nu - \nu\cos\theta_i - \sqrt{(1+\nu-\nu\cos\theta_i)^2 - 1}.\end{aligned} \tag{1.4}$$

$$\nu = \frac{r}{r_1}. \tag{1.5}$$

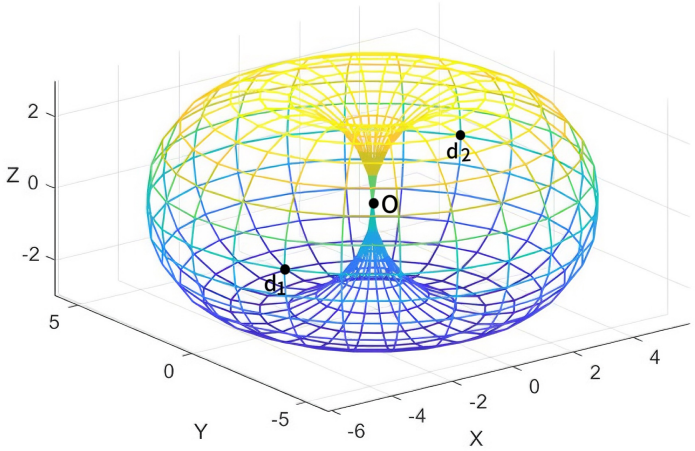

Figure 1. An arbitrary $m \times n$ apple surface network model, in which the resistance values on the longitude and latitude lines are r_1 and r, respectively, and the nodal numbers at the longitude (apart from pole O) and atitude are m and n [23].

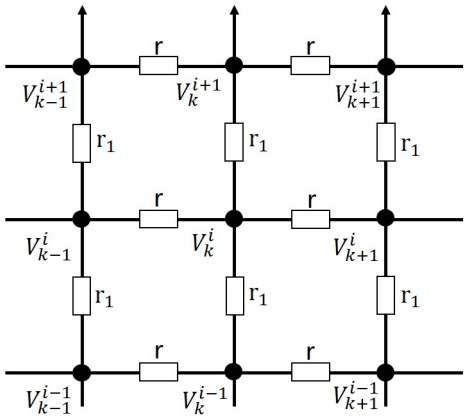

Figure 2. Segment of apple surface resistor network with potential parameters [23].

2. Optimized potential formulas using Chebyshev polynomials

In this section, we give an optimized potential formula for the apple surface resistor network. The potential formula optimized by the first-class Chebyshev polynomial reduces CPU runtime.

Assume the current J is input at $d_1(x_1, y_1)$ and output at $d_2(x_2, y_2)$. The optimized potential formula

of any point $d(x, y)$ in the $m \times n$ apple surface network is

$$\frac{\mathbf{U}_{m\times n}(x,y)}{J} = \frac{2r}{m+1}\sum_{\iota=1}^{m}\frac{\mu_{x_1,x}^{(\iota)}\sin(y_1\theta_\iota) - \mu_{x_2,x}^{(\iota)}\sin(y_2\theta_\iota)}{T_n^{(\iota)} - T_{n-2}^{(\iota)} - 2}\sin(y\theta_\iota), \tag{2.1}$$

where

$$\theta_\iota = \frac{\iota\pi}{m+1}, \tag{2.2}$$

$$\mu_{x_s,x}^{(\iota)} = T_{n-|x_s-x|-1}^{(\iota)} + T_{|x_s-x|-1}^{(\iota)}, s = 1, 2, \tag{2.3}$$

$$\omega_\iota = 2 + 2h - 2h\cos\frac{\iota\pi}{m+1}, \tag{2.4}$$

$$h = \frac{r}{r_1}, \tag{2.5}$$

$$T_n^{(\iota)} = \frac{\sinh(n\rho_\iota)}{\sqrt{\left(\frac{\omega_\iota}{2}\right)^2 - 1}}, \cosh\rho_\iota = \frac{\omega_\iota}{2}. \tag{2.6}$$

Assuming that the voltage at the origin, O, is zero, i.e., $V_0^{(0)} = 0$ the potential at any node concerning the origin can be expressed using Ohm's law:

$$\frac{\mathbf{U}_{m\times n}(x,y)}{J} = \frac{V_x^{(y)} - V_0^{(0)}}{J}, \tag{2.7}$$

where $V_x^{(y)}$ is the node voltage.

3. Horadam sequence and the first-class discrete sine transform

In this section, we employ the Chebyshev polynomials of the first kind to represent the Horadam sequence, improve the efficiency of calculating the potential of large-scale resistor networks.

Horadam sequence [24] is defined by the following recurrence relation

$$W_n = pW_{n-1} - qW_{n-2}, W_0 = a, W_1 = b, \tag{3.1}$$

where $p, q, a, b \in \boldsymbol{C}$, $n \in \boldsymbol{N}, n \geq 2$, $\boldsymbol{C}$ is the set of all complex numbers, and $\boldsymbol{N}$ is the set of all natural numbers.

The analytical expression of the Horadam sequence [24] in Eq (3.1) is as follows

$$W_n = \frac{A\beta^n - B\bar{\beta}^n}{\beta - \bar{\beta}}, \tag{3.2}$$

where $A = b - a\bar{\beta}, B = b - a\beta, \beta$ and $\bar{\beta}$ are the roots of the associated characteristic equation for $\beta^2 - p\beta + q = 0$.

The expression of Eq (3.1) can be represented using the Chebyshev polynomials of the first kind [25], resulting in the following equation

$$W_n = \frac{2\sqrt{E}q^{\frac{n}{2}}}{\sqrt{p^2 - 4q}}\cos(n\tau + \phi), \tag{3.3}$$

where

$$E = pab - qa^2 - b^2, \cos\tau = \frac{p}{2\sqrt{q}}, \cos\phi = \frac{a\sqrt{p^2-4q}}{2\sqrt{E}}, \tau, \phi \in \boldsymbol{C}. \tag{3.4}$$

Remark 1 The derivation of Eq (1.3) using the Chebyshev polynomial of the first class will be presented.

$$F_\nu^{(\iota)} = \omega_\iota F_{\nu-1}^{(\iota)} - F_{\nu-2}^{(\iota)}, F_0^{(\iota)} = 0, F_1^{(\iota)} = 1. \tag{3.5}$$

Based on Eqs (1.3), (3.1) (3.3) and (3.5), the following is obtained

$$F_n^{(\iota)} = T_n^{(\iota)} = \frac{2\sqrt{-1}}{\sqrt{\omega_\iota^2 - 4}} \cos(n\tau_\iota + \frac{\pi}{2}), \tag{3.6}$$

$$\cos\tau_\iota = \frac{\omega_\iota}{2}. \tag{3.7}$$

After taking into account the physical background of the resistor network, Eq (3.6) is written as

$$F_n^{(\iota)} = T_n^{(\iota)} = \frac{\sinh(n\rho_\iota)}{\sqrt{\left(\frac{\omega_\iota}{2}\right)^2 - 1}}, \tag{3.8}$$

where

$$\cosh\rho_\iota = \frac{\omega_\iota}{2}, i\rho_\iota = \tau_\iota, \tag{3.9}$$

and ω_ι is given in Eq (2.4).

Remark 2 The derivation of $\lambda_\iota^n + \bar{\lambda}_\iota^n$ using the Chebyshev polynomial of the first class will be provided.

Let

$$B_n^{(\iota)} = \lambda_\iota^n + \bar{\lambda}_\iota^n, \tag{3.10}$$

where $B_0^{(\iota)} = 2, B_1^{(\iota)} = \omega_\iota$.

After that, the expression for the recursive relation of $B_n^{(\iota)}$ is

$$B_n^{(\iota)} = \omega_\iota B_{n-1}^{(\iota)} - B_{n-2}^{(\iota)}, B_0^{(\iota)} = 2, B_1^{(\iota)} = \omega_\iota, \tag{3.11}$$

where ω_l and $B_n^{(\iota)}$ are expressed in Eqs (2.4) and (3.10), respectively.

By Eqs (3.8) and (3.11), $B_n^{(\iota)}$ as follows,

$$B_n^{(\iota)} = \lambda_\iota^n + \bar{\lambda}_\iota^n = \omega_\iota T_{n-1}^{(\iota)} - 2T_{n-2}^{(\iota)} = T_n^{(\iota)} - T_{n-2}^{(\iota)}. \tag{3.12}$$

Based on Eqs (1.1), (1.2), (3.8), and (3.12), we re-express the potential formula using the Chebyshev polynomial of the first kind, resulting in Eq (2.1).

To obtain a quick method for the potential in large-scale resistor networks, we diagonalize the tridiagonal Toeplitz matrix $\mathbf{A}_m$ using the first kind of discrete sine transform.

$$A_m = \begin{pmatrix} 2+2h & -h & 0 & \cdots & \cdots & 0 \\ -h & 2+2h & -h & \ddots & & \vdots \\ 0 & -h & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -h & 0 \\ \vdots & & \ddots & -h & 2+2h & -h \\ 0 & \cdots & \cdots & 0 & -h & 2+2h \end{pmatrix}_{m\times m}, \tag{3.13}$$

$$h = \frac{r}{r_1} \tag{3.14}$$

Based on the node voltage and Kirchhoff's laws, Tan [7] provided the following matrix equation model

$$\vec{V}_{k+1} = A_{m\times m}\vec{V}_k - \vec{V}_{k-1} - r\vec{I}_x\delta_{y,i}, \tag{3.15}$$

where

$$\vec{V}_k = \left[\vec{V}_k^{(1)}, \vec{V}_k^{(2)}, \cdots, \vec{V}_k^{(\vec{m})}\right]^T, \tag{3.16}$$

$$I_x^{(i)} = J(\delta_{x_1,k} - \delta_{x_2,k}), \tag{3.17}$$

$$\delta_{k,x} = \begin{cases} 1, & x = k. \\ 0, & x \neq k. \end{cases} \tag{3.18}$$

We are going to conduct a matrix transformation.

Let

$$\mathbb{S}_m^I = \sqrt{\frac{2}{m+1}}\left(\sin\frac{jk\pi}{m+1}\right)_{k,j=1}^{m}. \tag{3.19}$$

The matrix $\mathbb{S}_m^I$ is clearly an orthogonal matrix, representing the first kind of discrete sine transform, with both its transpose and inverse equal to $\mathbb{S}_m^I$, and it possesses the following property:

$$(\mathbb{S}_m^I)^{-1} = (\mathbb{S}_m^I)^T = \mathbb{S}_m^I. \tag{3.20}$$

For Eq (3.13), perform the following orthogonal diagonalization

$$(\mathbb{S}_m^I)^{-1}\mathbf{A}_m(\mathbb{S}_m^I) = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m) \tag{3.21}$$

i.e.,

$$\mathbf{A}_m = (\mathbb{S}_m^I)\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)(\mathbb{S}_m^I)^{-1}, \tag{3.22}$$

where

$$\lambda_\iota = 2 + 2h - 2h\cos\frac{\iota\pi}{m+1}, \iota = 1, 2, \ldots, m, \tag{3.23}$$

From Eq (3.21), the eigenvalues of matrix $\mathbf{A}_m$ is λ_ι.

By left-multiplying Eq (3.21) by $\mathbb{S}_m^I$, the following equation is obtained

$$\mathbf{A}_m\mathbb{S}_m^I = (\mathbb{S}_m^I)\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m), \tag{3.24}$$

i.e.,

$$\mathbf{A}_m\left(\zeta^{(1)}, \zeta^{(2)}, \cdots, \zeta^{(m)}\right) = \left(\zeta^{(1)}, \zeta^{(2)}, \cdots, \zeta^{(m)}\right)\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m), \tag{3.25}$$

where $\zeta^{(j)} = (\zeta_1^{(j)}, \ldots, \zeta_m^{(j)})^T$,

$$\zeta_k^{(j)} = \sqrt{\frac{2}{m+1}}\sin\frac{jk\pi}{m+1}, j = 1, 2, \ldots, m, k = 1, 2, \ldots, m, \tag{3.26}$$

Eq (3.25) can be written as follows

$$\mathbf{A}_m\zeta^{(i)} = \lambda_\iota\zeta^{(i)}, \iota = 1, 2, \ldots, m. \tag{3.27}$$

According to Eq (3.27), we get the eigenvector $\zeta^{(j)} = (\zeta_1^{(j)}, \ldots, \zeta_m^{(j)})^T$ corresponding to λ_ι.

Let

$$\mathrm{C}_k = \mathbb{S}_m^I \mathrm{V}_k, \tag{3.28}$$

i.e.,

$$\mathrm{V}_k = \mathbb{S}_m^I \mathrm{C}_k, \tag{3.29}$$

C_k is a column vector of $m \times 1$,

$$\mathbf{C}_k = [c_k^{(1)}, c_k^{(2)}, ..., c_k^{(m)}]^T, (0 \le k \le n). \tag{3.30}$$

In order to design a fast algorithm, it is necessary to compute V_0 and V_1. For this purpose, we present the following conclusion.

Remark 3 The node voltage formula for the apple surface network was provided by Tan [7] as follows.

$$V_k^{(\mathrm{y})} = J\frac{2r}{m+1}\sum_{i=1}^{m}\frac{g_{x_1,x}^{(i)}\sin(y_1\theta_i) - g_{x_2,x}^{(i)}\sin(y_2\theta_i)}{\lambda_i^n + \lambda_i^n - 2}\sin(y\theta_i), \tag{3.31}$$

where $g_{x_1,x}^{(i)}$ is expressed in Eq (1.2), λ_i^n and $\bar{\lambda}_i^n$ in Eq (4.16), $\theta_i = i\pi/(m+1)$, $k = 0, 1$. Using the first-class Chebyshev polynomial, we re-express the point voltage formula as follows using Eqs (1.2), (3.8), (3.12), and (3.31).

$$\mathrm{V}_x^{(y)} = \frac{2rJ}{m+1}\sum_{\iota=1}^{m}\frac{\mu_{x_1,x}^{(\iota)}\sin(y_1\theta_\iota) - \mu_{x_2,x}^{(\iota)}\sin(y_2\theta_\iota)}{T_n^\iota + T_{n-2}^\iota - 2}\sin(y\theta_\iota), \tag{3.32}$$

where $\theta_\iota = \iota\pi/m + 1$, $\mu_{x_s,x}^{(\iota)}$, and $T_n^{(\iota)}$ are same as Eqs (2.3) and (2.6), respectively.

According to Eqs (3.28)–(3.30), and (3.32), we can get the exact formula of $\mathbf{c}_k^{(\iota)}$ as

$$\mathbf{c}_x^{(\iota)} = \frac{\sqrt{2}rJ}{\sqrt{m+1}}\left(\frac{\mu_{x_1,x}^{(\iota)}\sin(y_1\theta_\iota) - \mu_{x_2,x}^{(\iota)}\sin(y_2\theta_\iota)}{T_n^{(\iota)} - T_{n-2}^{(\iota)} - 2}\right), (0 \le x \le n). \tag{3.33}$$

4. Displaying the potential formulas for some special cases

Based on the derived resistor network potential formula (2.1), the potential formula is analyzed from two aspects. Corresponding variables are assigned based on specific conditions, and 3D views are used to demonstrate them.

Influence of current input node on potential formulas

Case 1. In the event when the current J enters at $d_1(x_1, y_1)$ and exits at $d_2(x_2, y_2) = O_{(0,0)}$, the apple surface network's optimized potential formula may be recast as

$$\frac{\mathbf{U}_{m\times n}(x,y)}{J} = \frac{2r}{m+1}\sum_{\iota=1}^{m}\frac{\mu_{x_1,x}^{(\iota)}\sin(y_1\theta_\iota)}{T_n^{(\iota)} - T_{n-2}^{(\iota)} - 2}\sin(y\theta_\iota), \tag{4.1}$$

$$\mu_{x_1,x}^{(\iota)} = T_{n-|x_1-x|-1}^{(\iota)} + T_{|x_1-x|-1}^{(\iota)}, \tag{4.2}$$

$$T_n^{(\iota)} = \frac{\sinh(n\rho_\iota)}{\sqrt{\left(\frac{\omega_\iota}{2}\right)^2 - 1}}, \tag{4.3}$$

$$\cosh \rho_\iota = \frac{\omega_\iota}{2}, \tag{4.4}$$

$$\omega_\iota = 2 + 2h - 2h \cos \frac{\iota\pi}{m+1}, h = \frac{r}{r_1}, \tag{4.5}$$

$$\theta_\iota = \iota\pi/m + 1. \tag{4.6}$$

Let $m = n = 60, J = 1, x_1 = y_1 = 20, x_2 = y_2 = 0$ and $r_1 = r = 1$ in Eq (4.1). The following is the special potential formula for the apple surface network.

$$\mathbf{U}_{60\times60}(x, y) = \frac{2}{61} \sum_{\iota=1}^{60} \frac{\mu_{20,x}^{(\iota)} \sin(20\theta_\iota)}{T_{60}^{(\iota)} - T_{58}^{(\iota)} - 2} \sin(y\theta_\iota), \tag{4.7}$$

where

$$\mu_{20,x}^{(\iota)} = T_{59-|20-x|}^{(\iota)} + T_{|20-x|-1}^{(\iota)}, \tag{4.8}$$

$$T_{60}^{(\iota)} = \frac{\sinh(60\rho_\iota)}{\sqrt{\left(\frac{\omega_\iota}{2}\right)^2 - 1}}, T_{58}^{(\iota)} = \frac{\sinh(58\rho_\iota)}{\sqrt{\left(\frac{\omega_\iota}{2}\right)^2 - 1}}, \tag{4.9}$$

$$\cosh \rho_\iota = \frac{\omega_\iota}{2}, \tag{4.10}$$

$$\omega_\iota = 4 - 2\cos \frac{\iota\pi}{61}, \tag{4.11}$$

$$\theta_\iota = \iota\pi/61. \tag{4.12}$$

The 3D potential distribution view of Eq (4.7) is depicted as shown in Figure 3.

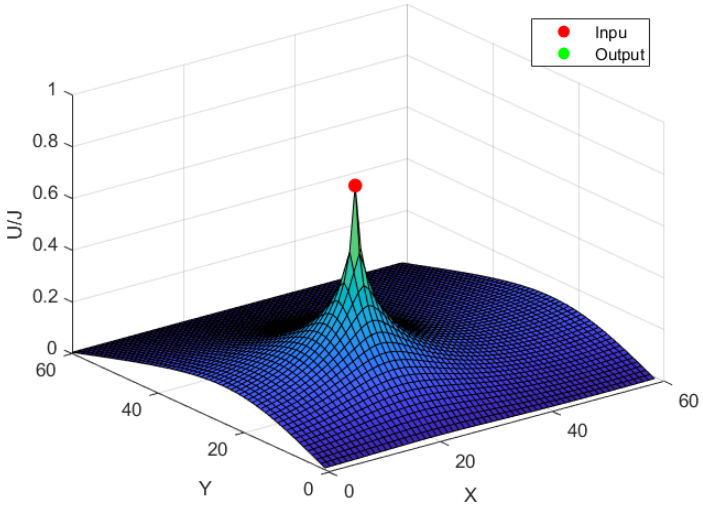

Figure 3. The 3D distribution view of $\mathbf{U}_{60\times60}(x, y)/J$ in Eq (4.7).

Case 2. If the input point of current J is $d_1(x_1, y_1)$ and the output point is $d_2(x_2, y_1)$, a novel potential formula for the apple surface network can be rewritten as.

$$\frac{\mathbf{U}_{m\times n}(x,y)}{J} = \frac{2r}{m+1}\sum_{\iota=1}^{m}\frac{(\mu_{x_1,x}^{(\iota)} - \mu_{x_2,x}^{(\iota)})\sin(y_1\theta_\iota)}{T_n^{(\iota)} - T_{n-2}^{(\iota)} - 2}\sin(y\theta_\iota), \tag{4.13}$$

$$\mu_{x_s,x}^{(\iota)} = T_{n-|x_s-x|-1}^{(\iota)} + T_{|x_s-x|-1}^{(\iota)}, s = 1, 2, \tag{4.14}$$

$T_n^{(\iota)}$, ρ_ι, ω_ι and θ_ι are given by Eqs (4.3)–(4.6), respectively.

Let $m = n = 60, J = 1, x_1 = y_1 = y_2 = 20, x_2 = 40$, and $r_1 = r = 1$ in Eq (4.7). After that, the apple surface network's idiosyncratic potential formula is presented by

$$\mathbf{U}_{60\times 60}(x,y) = \frac{2}{61}\sum_{\iota=1}^{60}\frac{(\mu_{20,x}^{(\iota)} - \mu_{40,x}^{(\iota)})\sin(20\theta_\iota)}{T_{60}^{(\iota)} - T_{58}^{(\iota)} - 2}\sin(y\theta_\iota), \tag{4.15}$$

$$\begin{aligned}\lambda_i &= 1 + \nu - \nu\cos\theta_i + \sqrt{(1+\nu-\nu\cos\theta_i)^2 - 1},\\ \overline{\lambda}_i &= 1 + \nu - \nu\cos\theta_i - \sqrt{(1+\nu-\nu\cos\theta_i)^2 - 1}.\end{aligned} \tag{4.16}$$

$$\mu_{20,x}^{(\iota)} = T_{59-|20-x|}^{(\iota)} + T_{|20-x|-1}^{(\iota)}, \mu_{40,x}^{(\iota)} = T_{59-|40-x|}^{(\iota)} + T_{|40-x|-1}^{(\iota)}, \tag{4.17}$$

$T_{60}^{(\iota)}$ and $T_{58}^{(\iota)}$, ρ_ι, ω_ι, and θ_ι, are given by Eqs (4.9)–(4.12).

The 3D potential distribution view of Eq (4.15) is depicted as shown in Figure 4.

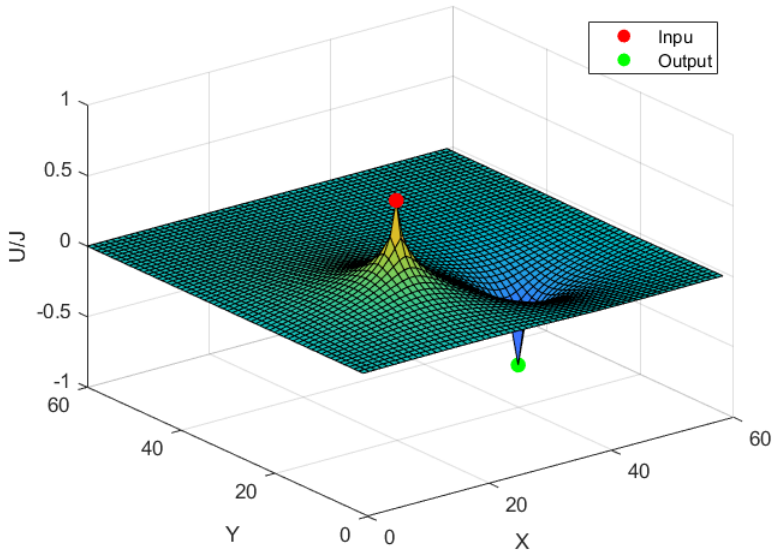

Figure 4. The 3D distribution view of $\mathbf{U}_{60\times 60}(x,y)/J$ in Eq (4.15).

Effect of resistivity $h(h = \frac{r}{r_1})$ on potential formulas.

In this section, the potential formula is affected by h, as depicted in 3D.

Let $m = n = 60, J = 1, x_1 = y_1 = 20, x_2 = y_2 = 40$, and $r = 1$ in Eq (2.1). Then a special potential formula of the apple surface network is expressed by

$$\mathbf{U}_{60\times 60}(x,y) = \frac{2}{61}\sum_{\iota=1}^{60}\frac{\mu_{20,x}^{(\iota)}\sin(20\theta_\iota) - \mu_{40,x}^{(\iota)}\sin(40\theta_\iota)}{T_{60}^{(\iota)} - T_{58}^{(\iota)} - 2}\sin(y\theta_\iota), \tag{4.18}$$

$$\mu_{20,x}^{(\iota)} = T_{59-|20-x|}^{(\iota)} + T_{|20-x|-1}^{(\iota)}, \mu_{40,x}^{(\iota)} = T_{59-|40-x|}^{(\iota)} + T_{|40-x|-1}^{(\iota)}, \tag{4.19}$$

$$T_{60}^{(\iota)} = \frac{\sinh(60\rho_\iota)}{\sqrt{\left(\frac{\omega_\iota}{2}\right)^2 - 1}}, T_{58}^{(\iota)} = \frac{\sinh(58\rho_\iota)}{\sqrt{\left(\frac{\omega_\iota}{2}\right)^2 - 1}}, \tag{4.20}$$

$$\cosh\rho_\iota = \frac{\omega_\iota}{2}, \tag{4.21}$$

$$\omega_\iota = 2 + 2h - 2h\cos\frac{\iota\pi}{m+1}, h = \frac{1}{r_1}, \tag{4.22}$$

$$\theta_\iota = \iota\pi/61. \tag{4.23}$$

The change in h affects only ω and T; therefore, in the following idiosyncratic potential formulas 3, 4, and 5, only the variations in ω and T are considered.

Case 3. When $r_1 = 1, h = 1$, ω_ι and $T_n^{(\iota)}$ are obtained as follows.

$$\mathbf{U}_{60\times 60}(x,y) = \frac{2}{61}\sum_{\iota=1}^{60}\frac{\mu_{20,x}^{(\iota)}\sin(20\theta_\iota) - \mu_{40,x}^{(\iota)}\sin(40\theta_\iota)}{T_{60}^{(\iota)} - T_{58}^{(\iota)} - 2}\sin(y\theta_\iota), \tag{4.24}$$

$$\omega_\iota = 4 - 2\cos\frac{\iota\pi}{61}, \tag{4.25}$$

$\mu_{20,x}^{(\iota)}$ and $\mu_{40,x}^{(\iota)}$, $T_{60}^{(\iota)}$ and $T_{58}^{(\iota)}$, ρ_ι, and θ_ι, are given by Eqs (4.19)–(4.21), and (4.23).

The 3D potential distribution view of Eq (4.24) is depicted as shown in Figure 5.

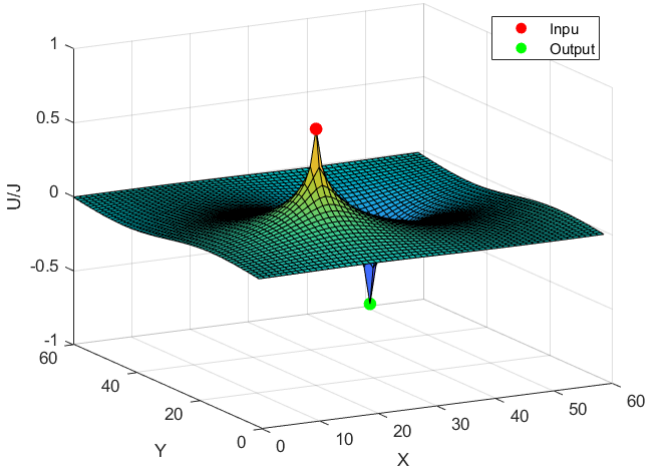

Figure 5. The 3D distribution view of $\mathbf{U}_{60\times 60}(x,y)/J$ in Eq (4.24).

Case 4. When $r_1 = 10, h = 0.1$, ω_ι and $T_n^{(\iota)}$ are obtained as follows.

$$\mathbf{U}_{60\times60}(x,y) = \frac{2}{61}\sum_{\iota=1}^{60}\frac{\mu_{20,x}^{(\iota)}\sin(20\theta_\iota) - \mu_{40,x}^{(\iota)}\sin(40\theta_\iota)}{T_{60}^{(\iota)} - T_{58}^{(\iota)} - 2}\sin(y\theta_\iota), \tag{4.26}$$

$$\omega_\iota = 2.2 - 0.2\cos\frac{\iota\pi}{61}, \tag{4.27}$$

$\mu_{20,x}^{(\iota)}$ and $\mu_{40,x}^{(\iota)}$, $T_{60}^{(\iota)}$ and $T_{58}^{(\iota)}$, ρ_ι, and θ_ι, are given by Eqs (4.19)–(4.21), and (4.23).

The 3D potential distribution view of Eq (4.26) is depicted as shown in Figure 6.

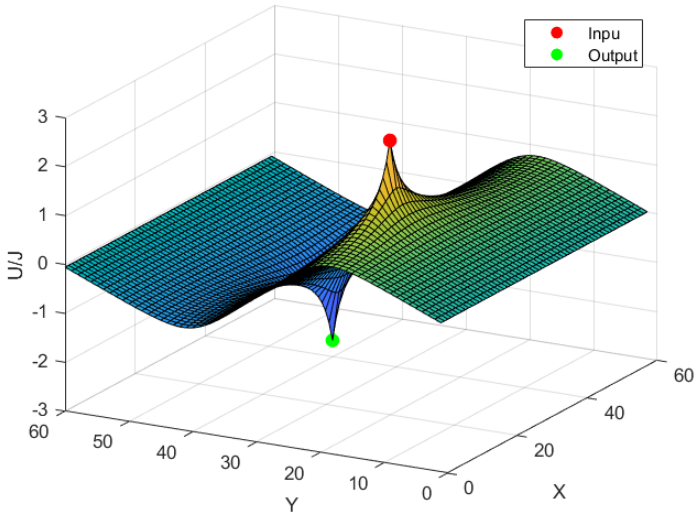

Figure 6. The 3D distribution view of $\mathbf{U}_{60\times60}(x,y)/J$ in Eq (4.26).

Case 5. When $r_1 = 0.1, h = 10$, ω_ι and $T_n^{(\iota)}$ are obtained as follows

$$\mathbf{U}_{60\times60}(x,y) = \frac{2}{61}\sum_{\iota=1}^{60}\frac{\mu_{20,x}^{(\iota)}\sin(20\theta_\iota) - \mu_{40,x}^{(\iota)}\sin(40\theta_\iota)}{T_{60}^{(\iota)} - T_{58}^{(\iota)} - 2}\sin(y\theta_\iota), \tag{4.28}$$

$$\omega_\iota = 22 - 20\cos\frac{\iota\pi}{61}, \tag{4.29}$$

$\mu_{20,x}^{(\iota)}$ and $\mu_{40,x}^{(\iota)}$, $T_{60}^{(\iota)}$ and $T_{58}^{(\iota)}$, ρ_ι, and θ_ι, are given by Eqs (4.19)–(4.21), and (4.23).

The 3D potential distribution view of Eq (4.28) depicted as shown in Figure 7.

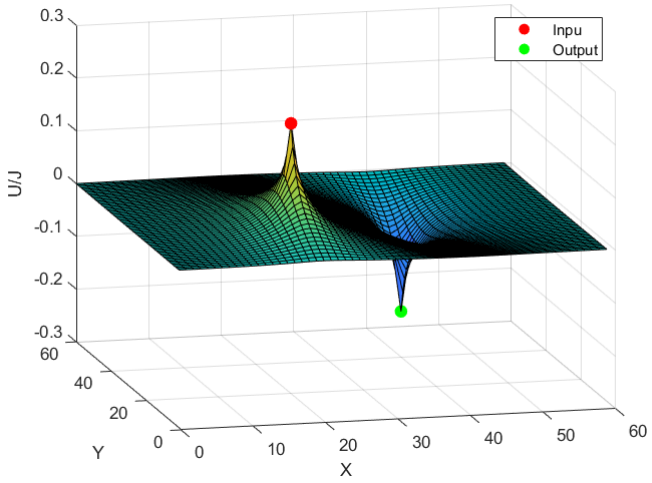

Figure 7. The 3D distribution view of $\mathbf{U}_{60\times60}(x, y)/J$ in Eq (4.28).

5. Numerical methods for potential computation

Combining the DST-I and Eqs (2.4), (2.6), (2.7), (3.15), (3.29), and (3.33), we present a numerical algorithm for fast algorithm of large-scale potential in the resistor network. We use a fast algorithm to obtain the numerical values of the potential at each point of the resistance network.

Based on the scale presented in this paper, the algorithm's computational complexity is analyzed. As is widely known, the computational complexity of the matrix vector multiplication with the modified tridiagonal Toeplitz matrix is $O(m)$, which is equivalent to the computational complexity of Algorithm 1. Moreover, the computational complexity of DST-I is $O(m\log_2 m)$. Additionally, during the process of recursive computation, Algorithm 1 is nested, resulting in a computational complexity of $O(mn)$ for this stage. Taking into account the aforementioned analysis, the computational complexity of the potential calculated by the Algorithm 2 is $m\log_2 m + O(mn)$.

Algorithm 1: Fast matrix-vector multiplication $\mathbf{A}_m\mathbf{q} = \mathbf{y}$
Step1: Compute y_1 by Eq $y_1 = (2 + 2h)q_1 - hq_2$;
Step2: Cycle computing y_ι by Eqs $y_\iota = (-h)q_{\iota-1} + (2 + 2h)q_\iota - hq_{\iota+1}, \iota = 2, \ldots, m - 1$;
Step3: Compute y_m by Eq $y_m = (-h)q_{m-1} + (2 + 2h)q_m$.

Algorithm 2: Algorithm for calculating $\mathbf{U}_{m\times n}(x, y)/J$
Step1: Compute ω_ι by Eq (2.4), $\iota = 1, 2, \ldots, m$;
Step2: Compute ρ_ι by $\cosh\rho_\iota = \frac{\omega_\iota}{2}$, $\iota = 1, 2, \ldots, m$;
Step3: Compute $T_n^{(\iota)}$ by Eq (2.6), $n = n - \lvert x_s - x\rvert - 1, \lvert x_s - x\rvert - 1, n, n - 2, s = 1, 2, \iota = 1, 2, \ldots, m$;
Step4: Compute $\mathbf{c}_0^{(\iota)}$ and $\mathbf{c}_1^{(\iota)}$ by Eq (3.33), $\iota = 1, \ldots, m$;
Step5: Compute $\mathbf{V}_k$ by Eq (3.29), and DST-I, $k = 0, 1$;
Step6: Compute $\mathbf{A}_m\mathbf{V}_k$ by Algorithm 1, $k = 1, 2, \ldots, ;$
Step7: Multiple recursive calculation $\mathbf{V}_k$ by Eq (3.15), $k = 2, 3, \ldots, ;$
Step8: Compute $\mathbf{U}_{m\times n}(x, y)/J$ by Eq (2.7).

Example1. When the scale is $m = 1000$ and $n = 10$, Algorithm 2 yields the potential distribution depicted in Figure 8, where the current enters at node (x_1, y_1) $(x_1 = 3, y_1 = 200)$ and exits at node (x_2, y_2) $(x_2 = 7, y_2 = 800)$, $r = 1, r_1 = 100$, and $J = 10$.

Example2. When the scale is $m = 1400$ and $n = 10$, Algorithm 2 yields the potential distribution depicted in Figure 9, where the current enters at node (x_1, y_1) $(x_1 = 5, y_1 = 400)$ and exits at node (x_2, y_2) $(x_2 = 9, y_2 = 1000)$, $r = 1, r_1 = 100$, and $J = 10$.

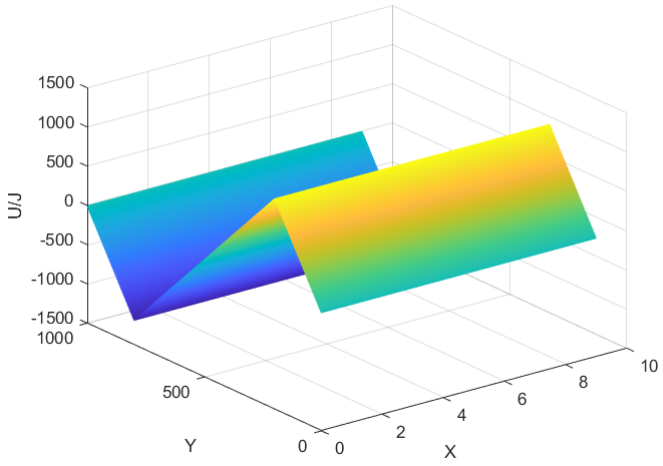

Figure 8. A 3D image display of $\mathbf{U}_{1000\times10}(x, y)/J$ by the fast Algorithm 2.

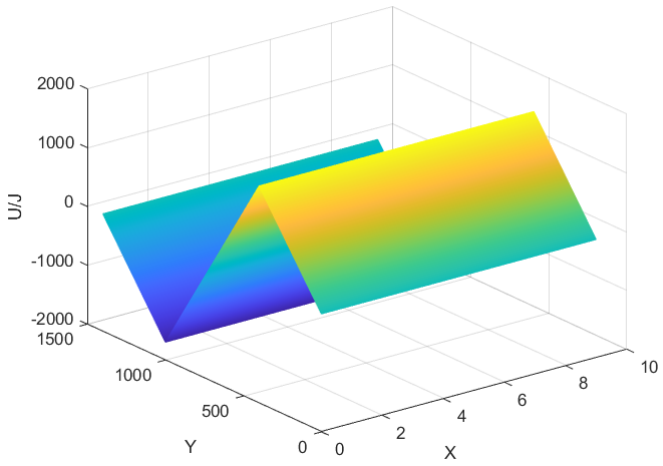

Figure 9. A 3D image display of $\mathbf{U}_{1400\times10}(x, y)/J$ by the fast Algorithm 2.

6. Efficiency of calculation method

In this section, examples are shown that demonstrate the computational efficiency of the potential computed by formulas (1.1), (2.1), and Algorithm 2. In the $m\times n$ rectangular resistor network, $d_1(x_1, y_1)$ and $d_2(x_2, y_2)$ represent the current input and output nodes, respectively. The horizontal axis represents

the scale of the resistor network. The CPU processing times t_1, t_2, and t_3 represent the time required to calculate the formulas (1.1) and (2.1), and Algorithm 2. The time is measured in seconds, and "–" signifies that the calculation scale or time exceeds 1200 seconds.

The AMD R7-4800H laptop used for these tests has a CPU clock of 2.9 GHz.

Remark 4 In Tables 1–4, it is obvious that the calculation efficiency of potential function (2.1) is much faster than that of potential function (1.1) for a square and rectangle with different resistivity. Furthermore, the experimental data from Tables 5 further suggest that as the value of r/r_1 decreases, the computational efficiency of both the original and optimized formulas will be enhanced to some degree when processing resistor networks of the same scale.

Remark 5 The significantly improved efficiency of the fast Algorithm 2 in calculating the potential is evident from the data presented in Table 6. Compared to the analytic potential functions (1.1) and (2.1), the fast Algorithm 2 demonstrates significantly higher efficiency. Furthermore, both the fast Algorithm 2 and the analytic potential function (2.1) can handle larger calculation scales compared to the analytic potential function (1.1)).

Table 1. The computation time and by formulas (1.1), (2.1), respectively, for $r/r_1 = 1$.

$m \times n$	(x_1, y_1)	(x_2, y_2)	r/r_1	t_1	t_2
100×100	$(1, 1)$	$(10, 10)$	1	0.132	0.074
300×300	$(1, 1)$	$(10, 10)$	1	2.099	1.440
500×500	$(1, 1)$	$(10, 10)$	1	12.240	5.031
1000×1000	$(1, 1)$	$(10, 10)$	1	81.067	33.812
2000×2000	$(1, 1)$	$(10, 10)$	1	600.351	199.650

Table 2. The computation time and by formulas (1.1), (2.1), respectively, for $r/r_1 = 1$.

$m \times n$	(x_1, y_1)	(x_2, y_2)	r/r_1	t_1	t_2
1000×10	$(1, 1)$	$(10, 10)$	1	0.964	0.621
3000×30	$(1, 1)$	$(10, 10)$	1	11.319	3.314
5000×50	$(1, 1)$	$(10, 10)$	1	52.217	13.579
10000×100	$(1, 1)$	$(10, 10)$	1	830.103	310.639
20000×200	$(1, 1)$	$(10, 10)$	1	-	-

Table 3. The computation time and by formulas (1.1), (2.1), respectively, for $r/r_1 = 0.1$.

$m \times n$	(x_1, y_1)	(x_2, y_2)	r/r_1	t_1	t_2
100×100	$(1, 1)$	$(10, 10)$	0.1	0.121	0.065
300×300	$(1, 1)$	$(10, 10)$	0.1	1.753	0.980
500×500	$(1, 1)$	$(10, 10)$	0.1	10.050	4.121
1000×1000	$(1, 1)$	$(10, 10)$	0.1	76.707	31.185
2000×2000	$(1, 1)$	$(10, 10)$	0.1	580.363	183.071

Table 4. The computation time and by formulas (1.1), (2.1), respectively, for $r/r_1 = 0.1$.

$m \times n$	(x_1, y_1)	(x_2, y_2)	r/r_1	t_1	t_2
1000×10	$(1, 1)$	$(10, 10)$	0.1	0.132	0.074
3000×30	$(1, 1)$	$(10, 10)$	0.1	2.099	1.440
5000×50	$(1, 1)$	$(10, 10)$	0.1	12.240	5.031
10000×100	$(1, 1)$	$(10, 10)$	0.1	81.067	33.812
20000×200	$(1, 1)$	$(10, 10)$	0.1	600.351	199.650

Table 5. The computation time and by formulas (1.1) and (2.1) when r/r_1 change.

$m \times n$	(x_1, y_1)	(x_2, y_2)	r/r_1	t_1	t_2
1000×1000	$(1, 1)$	$(10, 10)$	100	90.894	38.053
1000×1000	$(1, 1)$	$(10, 10)$	10	85.682	36.683
1000×1000	$(1, 1)$	$(10, 10)$	1	80.067	33.812
1000×1000	$(1, 1)$	$(10, 10)$	0.1	77.034	29.682
1000×1000	$(1, 1)$	$(10, 10)$	0.01	75.352	26.795

Table 6. The computation time and by formulas (1.1), (2.1) and fast algorithm 2, respectively, for $r/r_1 = 1$.

$m \times n$	(x_1, y_1)	(x_2, y_2)	r/r_1	t_1	t_2	t_3
1000×10	$(1, 1)$	$(10, 10)$	1	0.964	0.621	0.065
3000×10	$(1, 1)$	$(10, 10)$	1	6.591	2.278	0.226
5000×10	$(1, 1)$	$(10, 10)$	1	17.391	7.573	0.499
10000×10	$(1, 1)$	$(10, 10)$	1	136.673	51.536	1.639
20000×10	$(1, 1)$	$(10, 10)$	1	430.672	150.130	7.491

7. Application in path planning

In this section, based on the potential formula, a path planning algorithm suitable for apple surface environment is preliminary designed.

Algorithm 3: This algorithm is a heuristic method that completes path planning by simulating a potential drop. Compared with the traditional artificial potential field method, the potential formula algorithm is more suitable for path planning on the apple surface environment, especially since it can adapt to the periodi characteristics of the horizontal direction on apple surface environment. The path planning algorithm is described as follows.

Algorithm 3: Path Planning Algorithm

Step1: Establish robot working environment maps based on grid-based methods (determine obstacles and the starting point and target point of the robot);

Step2: Let the starting point and target point correspond to the input point and output point of the current in apple surface network, respectively, and calculate the potential $\frac{\mathbf{U}(x,y)}{J}$ of each node in the apple surface resistor network according to formula (8);

Step3: At the corresponding position of the obstruction in the apple surface resistor network, a set increment is applied to the potential. After the fixed increment is added, the potential is $\frac{\mathbf{U}(x,y)}{J} + 0.3\frac{\mathbf{U}(x_1,y_1)}{J}$ (where (x_1, y_1) is the potential of starting point);

Step4: Use $\min\{U_{x+1,y+1}, U_{x+1,y}, U_{x+1,y-1}, U_{x,y-1}, U_{x-1,y-1}, U_{x-1,y}, U_{x-1,y+1}, U_{x,y+1}\}$ to select the corresponding node with the smallest potential;

Step5: The autonomous mobile robot moves to the corresponding node according to the node found in Step 4 and updates the present position. Repeat Step 4 until the current node reaches the goal point;

Step6: Determine whether the current position of the robot is the target point, and if so, terminate; otherwise, execute Step 4;

Step7: The algorithm ends.

Let $x_1 = 5$, $y_1 = 3$, $x_2 = 7$, $y_2 = 9$, $r = 1$, $r_1 = 1$, and $J = 1$. According to function (2.1), the potential distribution views of the apple surface resistor network, including only the starting and ending point, can be obtained with a potential distribution diagram, as shown in Figures 10 and 11.

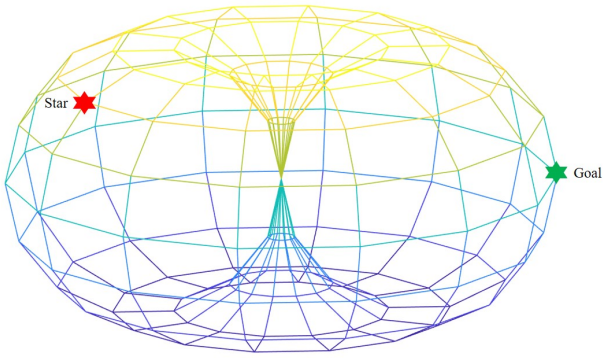

Figure 10. 15×15 apple surface environment with obstacles.

The potential value at the location of an obstacle is increased by a fixed increment when the obstacle is present in the environment. With this modification, the robot may successfully avoid obstacles during pathfinding by navigating around them. Moving from high-potential to low-potential locations is therefore the best way to plan a course. Figures 12–15 depict the robot's path planning in a node-weighted potential distribution map that corresponds to the actual apple environment.

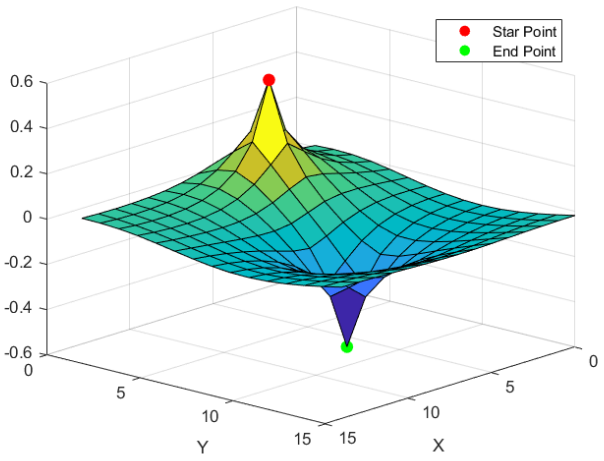

Figure 11. Potential distribution diagram.

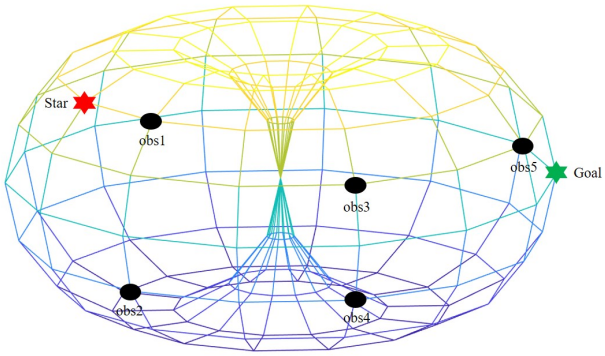

Figure 12. 15 × 15 apple surface environment after adding obstacles.

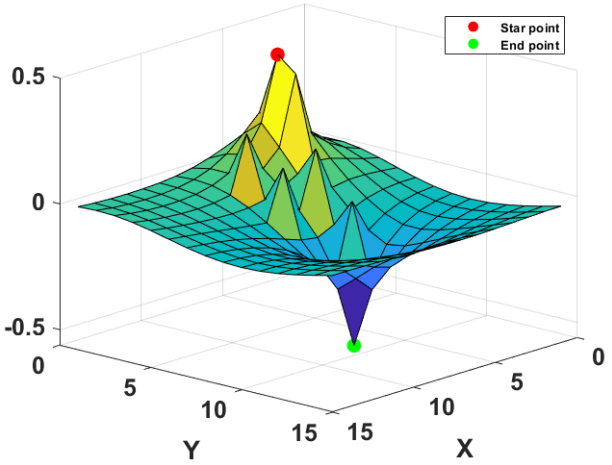

Figure 13. Potential distribution diagram after node-weighted adjustments for obstacles.

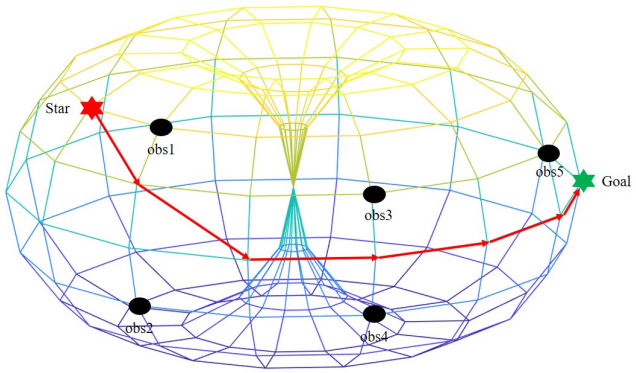

Figure 14. Robot path planning in 15×15 apple surface environment.

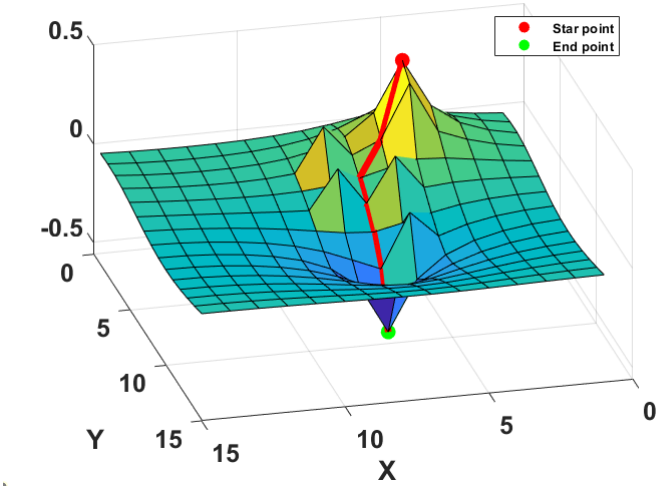

Figure 15. Path planning in a node-weighted potential distribution diagram.

Algorithm 4: Next, we use the potential function and gradient descent method to give an optimal path algorithm. This algorithm differs from Algorithm1 robot navigation approaches that can explore only in eight directions, as it is based on gradient descent, with each step taken in the direction of the steepest gradient descent, thus achieving both obstacle avoidance and path planning for the robot. The weighting potential function in path planning is expressed as follows

$$F(x, y) = \frac{\mathbf{U}_{m\times n}(x, y)}{J} + f_{\text{OBS}}(x, y), \tag{7.1}$$

where $f_{\text{OBS}}(x, y)$ is the obstacle function, defined as

$$f_{\text{OBS}}(x, y) = \frac{4}{5\pi}\left(\sum_{i=1}^{N_o} \exp(-P_i[\ x - X_i \quad y - Y_i\ ]A^{-1}[\ \begin{matrix} x - X_i \\ y - Y_i \end{matrix}\ ])\right), \tag{7.2}$$

where N_o is the number of obstacles, $A = [\ \begin{matrix} 15 & 0 \\ 0 & 15 \end{matrix}\ ]$, P_i and (X_i, Y_i) represent the size and coordinate

position of the ith obstacle, respectively.

Let $x_1 = 5$, $y_1 = 3$, $x_2 = 7$, $y_2 = 9$, $r = 1$, $r_1 = 1$, and $J = 1$. According to Algorithm 4, path planning in a node-weighted potential distribution diagram are shown in Figure 16, and a robot path planning in 15×15 apple surface environment is shown in Figure 17.

Algorithm 4: Path Planning Algorithm

Step1: Extract the necessary information of the robot and obstacles, i.e., N_o, S_x, S_y, G_x, G_y, P_i, X_i, Y_i, $i = 1, ..., N_o$, where (S_x, S_y) and (G_x, G_y) are the starting point and target point coordinates of the robot, respectively;

Step2: Construct an environment occupancy map;

Step3: Let the input and output points of the current be (S_x, S_y) and (G_x, G_y), respectively;

Step4: Calculate $F(x, y)$ by Eq (2.1), (7.1) and (7.2);

Step5: Initialize robot position $(R_x, R_y) = (S_x, S_y)$, and set a threshold value ε

WHILE norm$((R_x, R_y), (G_x, G_y)) > \varepsilon$ **DO**

$a = 1/2(\text{norm}(\nabla F(R_x, R_y)))$

$(R_x, R_y) = (R_x, R_y) - a \times \nabla F(R_x, R_y)$

END

Go to Step 6;

Step6: Map the obtained path in the apple surface environment;

Step7: Stop.

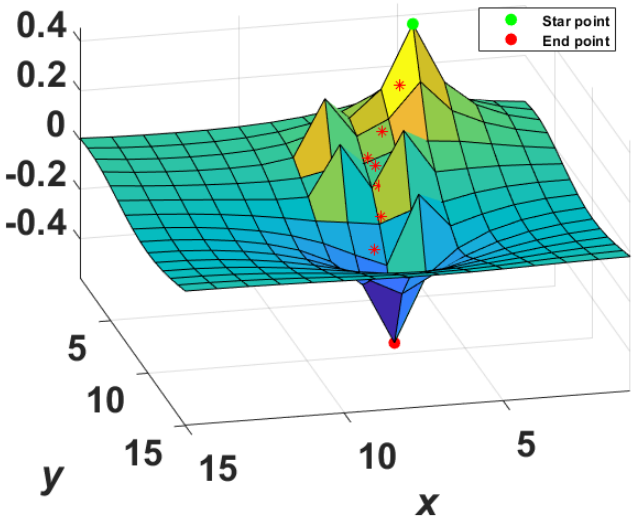

Figure 16. Path planning in a node-weighted potential distribution diagram.

Remark 6 As we know, the path obtained by gradient descent method must be optimal. Therefore, Algorithm 4 gives the optimal path. Algorithm 3 is used to find the path using the law of natural decline of potential from high to low; however, at present, we cannot prove that this path is optimal, as it merely represents an exploratory application of the potential function.

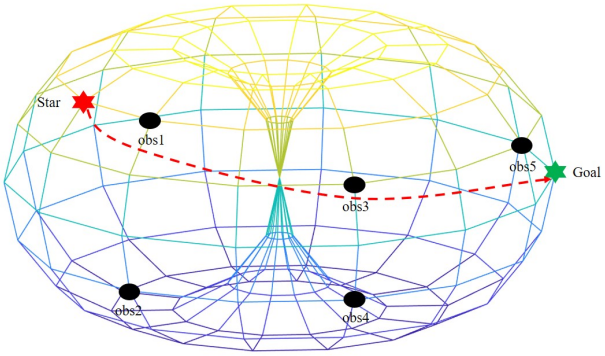

Figure 17. Robot path planning in a 15×15 apple surface environment.

8. Conclusions

The potential formula for the $m \times n$ apple surface network is altered in this paper. Unlike traditional methods for solving resistor networks, we use the Horadam sequence represented by Chebyshev polynomials of the first kind to derive explicit formulas for potential. Moreover, a fast algorithm for solving the potential is developed using the current recursive formula and DST-I. We also present 3D views of the discrete distribution of potential in several special cases and compare the calculation efficiency of the precise potential formula with the fast algorithm. Last, an algorithm for path planning is created based on the apple surface network's potential energy formula.

Formulas (1.1) and (2.1) and Algorithm 2 are applicable only to this particular type of resistance network and is not a universal algorithm for general resistance networks. In addition, Algorithm 2 will experience a sharp increase in error with the increase of iteration times, and when $n > 10$, it will not meet the engineering requirements. In the future, we will continue to explore better numerical algorithms.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgements

This research was supported by the Shandong provincial (Grant No. ZR2023QF031) and (Grant No. 2023KJ214).

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. X. Jiang, D. Meng, Y. Zheng, Z. Jiang, A dynamic cobweb resistance network solution based on a structured zeroing neural network and its applications, *Sci. Rep.*, **15** (2025), 5222. https://doi.org/10.1038/s41598-025-89471-6
2. G. Kirchhoff, Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird, *Ann. Phys.*, **148** (1847), 497–508. https://doi.org/10.1002/andp.18471481202
3. J. Cserti, Application of the lattice Green's function for calculating the resistance of an infinite network of resistors, *Am. J. Phys.*, **68** (2000), 896–906. https://doi.org/10.1119/1.1285881
4. F. Perrier, F. Girault, Rotational invariance in resistor networks: Two-point resistances around an n-fold corner, *Eur. J. Phys.*, **42** (2021), 025803. https://doi.org/10.1088/1361-6404/abc63d
5. F. Perrier, F. Girault, Two-point resistances in symmetric bracelet resistor networks: Accurate estimates and exact expressions, *Eur. J. Phys.*, **45** (2024), 025805. https://doi.org/10.1088/1361-6404/ad242a
6. F. Y. Wu, Theory of resistor networks: the two-point resistance, *J. Phys. A: Math. Gen.*, **37** (2004), 6653. https://doi.org/10.1088/0305-4470/37/26/004
7. Z. Tan, Theory of an $m \times n$ apple surface network with special boundary, *Commun. Theor. Phys.*, **75** (2023), 065701. https://doi.org/10.1088/1572-9494/accb82
8. F. Luo, L. Luo, Study on the electrical characteristics of $6 \times n$ cobweb cascaded circuit network model, *Results Phys.*, **33** (2022), 105160. https://doi.org/10.1016/j.rinp.2021.105160
9. J. Wang, Y. Zheng, Z. Jiang, Norm equalities and inequalities for tridiagonal perturbed Toeplitz operator matrices, *J. Appl. Anal. Comput.*, **13** (2023), 671–683. https://doi.org/10.11948/20210489
10. Y. Fu, X. Jiang, Z. Jiang, S. Jhang, Properties of a class of perturbed Toeplitz periodic tridiagonal matrices, *Comput. Appl. Math.*, **39** (2020), 146. https://doi.org/10.1007/s40314-020-01171-1
11. H. Chen, F. Zhang, Resistance distance local rules, *J. Math. Chem.*, **44** (2008), 405–417. https://doi.org/10.1007/s10910-007-9317-8
12. X. Jiang, G. Zhang, Y. Zheng, Z. Jiang, Explicit potential function and fast algorithm for computing potentials in $\alpha \times \beta$ conic surface resistor network, *Expert Syst. Appl.*, **238** (2024), 122157. https://doi.org/10.1016/j.eswa.2023.122157
13. Z. Jiang, Y. Zhou, X. Jiang, Y. Zheng, Analytical potential formulae and fast algorithm for a horn torus resistor network, *Phys. Rev. E*, **107** (2023), 044123. https://doi.org/10.1103/PhysRevE.107.044123
14. S. M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006. https://doi.org/10.1017/CBO9780511546877
15. C. Mahulea, M. Kloetzer, R. González, *Path Planning of Cooperative Mobile Robots Using Discrete Event Models*, Wiley-IEEE Press, 2019. https://doi.org/10.1002/9781119486305
16. Y. Ji, L. Ni, C. Zhao, C. Lei, Y. Du, W. Wang, Tripfield: A 3D potential field model and its applications to local path planning of autonomous vehicles, *IEEE Trans. Intell. Transp. Syst.*, **24** (2023), 3541–3554. https://doi.org/10.1109/TITS.2022.3231259

17. Z. Yu, J. Yuan, Y. Li, C. Yuan, S. Deng, A path planning algorithm for mobile robot based on water flow potential field method and beetle antennae search algorithm, *Comput. Electr. Eng.*, **109** (2023), 108730. https://doi.org/10.1016/j.compeleceng.2023.108730
18. Z. Pan, C. Zhang, Y. Xia, H. Xiong, X. Shao, An improved artificial potential field method for path planning and formation control of the multi-UAV systems, *IEEE Trans. Circuits Syst.-II: Express Briefs*, **69** (2022), 1129–1133. https://doi.org/10.1109/TCSII.2021.3112787
19. G. Kulathunga, A reinforcement learning based path planning approach in 3D environment, *Procedia Comput. Sci.*, **212** (2022), 152–160. https://doi.org/10.1016/j.procs.2022.10.217
20. H. Mazaheri, S. Goli, A. Nourollah, Path planning in three-dimensional space based on butterfly optimization algorithm, *Sci. Rep.*, **14** (2024), 2332. https://doi.org/10.1038/s41598-024-52750-9
21. J. Xue, J. Li, J. Chen, C. Tu, A. Stancu, X. Wang, Wall-climbing robot path planning for cylindrical storage tank inspection based on modified A-star algorithm, in *2021 IEEE Far East NDT New Technology & Application Forum (FENDT)*, (2021), 191–195. https://doi.org/10.1109/FENDT54151.2021.9749634
22. A. Uğur, Path planning on a cuboid using genetic algorithms, *Inf. Sci.*, **178** (2008), 3275–3287. https://doi.org/10.1016/j.ins.2008.04.005
23. Z. Tan, Electrical property of an $m \times n$ apple surface network, *Results Phys.*, **47** (2023), 106361. https://doi.org/10.1016/j.rinp.2023.106361
24. A. F. Horadam, Basic properties of a certain generalized sequence of numbers, *Fibonacci Q.*, **3** (1965), 161–176. https://doi.org/10.1080/00150517.1965.12431416
25. G. Udrea, A note on the sequence $(W_n)_{n\geq 0}$ of A. F. Horadam, *Port. Math.*, **53** (1996), 143–155.

AIMS Press