Radial basis meshless methods for unsaturated flow problems

J. Li & C.-S. Chen
Department of Mathematical Sciences,
University of Nevada, Las Vegas,
Las Vegas, Nevada 89154-4020, USA

Abstract

Since its first development by Kansa in 1990, the radial basis function collocation method gains increasing interest recently. In this paper we apply this truly meshless method to both 1-D and 2-D unsaturated flow problems, which are time-dependent and highly nonlinear. Our numerical results demonstrate that the Kansa’s method is accurate and efficient to solve such nonlinear problems.

1 Introduction

Prediction of fluid movement in unsaturated soils is very important in soil science and groundwater hydrology. The governing equation is the classical Richards equation, which can be written in several forms, with either pressure head, moisture content or mixing of them as the independent variable. In this paper, we consider the pressure head based form:

\[ M(\psi) \frac{\partial \psi}{\partial t} - \nabla \cdot K(\psi) \nabla \psi + \frac{\partial K}{\partial z} = f \]  

where \( \psi \) is the pressure head, \( M(\psi) \equiv d\theta/d\psi \) is the specific moisture capacity function, \( \theta = \theta(\psi) \) is the moisture content, \( K(\psi) \) is the unsaturated hydraulic conductivity, \( z \) denotes the vertical dimension positioning positive downward.

The equation (1) is time-dependent and nonlinear, much work has been carried out for developing efficient numerical methods for solving it. Most methods are the finite difference method (FDM) and finite element method (FEM) [4]. Recently there are many interesting radial basis functions (RBFs) based meshless methods developed such as the method of fundamental solution [6, 9], symmetric and
asymmetric collocation method [14, 8, 11, 7, 17, 18, 5, 16]. Some nice features of the above RBF based methods are that they do not require a grid, which is a truly meshless method and can be used to solve complex geometry problems very easily; they can achieve high-order accuracy in a very efficient way (the implementation is very simple compared to other classic numerical methods such as FDM and FEM); they are space-dimension independent due to the dimension independent fact of RBFs, so they are very attractive for solving high-dimensional problems also. We like to point out that there are some other types of meshless methods based upon Galerkin method and other methods, interested readers can find details from some review papers [2, 13] and recent books on meshless such as [1, 12, 10].

The goal of this paper is to extend our previous work on RBF meshless method to solve (1). The rest of the paper is organized as follows. In Section 2 we introduce our RBF meshless algorithm developed for (1). Several numerical examples for both 1-D and 2-D are presented in Section 3. Section 4 concludes our paper.

2 The RBF collocation method

This method was introduced in 1990 by Kansa [14] for solving PDEs by collocation using RBFs. This technique is very general, simple and effective, and it has been applied to many science and engineering areas [15].

For simplicity, we derive our scheme for a 2-D problem with Dirichlet boundary condition

$$\psi(\vec{x}) = g(\vec{x}) \quad \text{on } \partial \Omega.$$  \hfill (2)

It is not difficult to extend its usage to more general boundary condition problems.

Let \( \{ \vec{x}_i \}_{i=1}^N \) be \( N \) distinct collocation points in \( \Omega \), of which \( \{ \vec{x}_i \}_{i=1}^{N_I} \) are interior points; \( \{ \vec{x}_i \}_{i=N_I+1}^N \) are boundary points.

Using a backward Euler method for (1), we have

$$M(\psi_i^{n+1}) \frac{\psi_i^{n+1} - \psi_i^n}{\Delta t} + \frac{\partial K}{\partial z}(\psi_i^{n+1})(1 - \frac{\partial \psi}{\partial z})|_{i}^{n+1} - K(\psi_i^{n+1}) \frac{\partial^2 \psi}{\partial z^2} |_{i}^{n+1}$$

$$- \frac{\partial K}{\partial x}(\psi_i^{n+1}) \frac{\partial \psi}{\partial x} |_{i}^{n+1} - K(\psi_i^{n+1}) \frac{\partial^2 \psi}{\partial x^2} |_{i}^{n+1} = f(\vec{x}_i)$$  \hfill (3)

Because \( M \) and \( K \) are nonlinear functions of \( \psi \), we use the Picard iteration to linearize the problem, which gives us

$$M(\psi_i^{n+1,m}) \frac{\psi_i^{n+1,m} - \psi_i^{n+1,m-1}}{\Delta t} + \frac{\partial K}{\partial z}(\psi_i^{n+1,m}) \frac{\partial \psi}{\partial z} |_{i}^{n+1,m} - K(\psi_i^{n+1,m}) \frac{\partial^2 \psi}{\partial z^2} |_{i}^{n+1,m}$$

$$- \frac{\partial K}{\partial x}(\psi_i^{n+1,m}) \frac{\partial \psi}{\partial x} |_{i}^{n+1,m} - K(\psi_i^{n+1,m}) \frac{\partial^2 \psi}{\partial x^2} |_{i}^{n+1,m}$$

$$= M(\psi_i^{n+1,m}) \psi_i^{n+1,m} - \frac{\partial K}{\partial z}(\psi_i^{n+1,m}) + f(\vec{x}_i) \quad i = 1, \ldots, N_I,$$  \hfill (4)

where \( m \) and \( n \) identify the iteration level and the time level, respectively. Here \( \psi_i \) denotes the numerical solution at the collocation point \( \vec{x}_i \). More specifically we
denote the RBF numerical solution as

$$\psi(\vec{x}) = \sum_{j=1}^{N} c_j \varphi_j(\vec{x}), \quad (5)$$

where \( \{c_j\}_{j=1}^{N} \) are the unknown coefficients to be determined, and \( \varphi_j(\vec{x}) = \varphi(||\vec{x} - \vec{x}_j||) \) can be any radial basis function. Denote \( r = ||\vec{x} - \vec{x}_j|| \) the Euclidean norm between points \( \vec{x} \) and \( \vec{x}_j \). Then some widely used radial basis functions can be represented as: the multiquadric (MQ) \( \varphi(r) = (r^2 + \alpha^2)^{\beta/2} \) (\( \beta \) is a positive odd integer, \( \alpha \) is a free parameter); the Gaussian (GS) \( \varphi(r) = e^{-\alpha r^2} \) (\( \alpha \) is a free parameter); the polyharmonic splines \( \varphi(r) = r^k \log r \) (\( k \) is a positive even integer); and the conical type \( \varphi(r) = r^k \) (\( k \) is a positive odd integer). More details can be found in the recent monograph [3].

Substituting (5) into (4), we end up a system of \( N_I \) equations, which have to be solved together with other \( N-N_I \) equations obtained from the boundary condition (2), i.e.,

$$\sum_{j=1}^{N} \varphi_j(\vec{x}_i) c_j = g(\vec{x}_i), \quad i = N_I + 1, N_I + 2, \ldots, N. \quad (6)$$

Once the unknowns \( c_j \) are obtained, the solution at any point in the domain can be obtained from (5).

3 Numerical results

Below we present several numerical experiments with hypothetical exact solutions. To solve the problem (1), we need a set of soil properties \( \theta(\psi) \) and \( K(\psi) \). The first set we tested is

$$\theta(\psi) = \exp(\alpha \psi), \quad K(\psi) = \beta \exp(\alpha \psi), \quad \alpha = 0.1, \beta = 10. \quad (7)$$

The second set is [4]

$$\theta(\psi) = \theta_s - \theta_r \frac{1}{[1 + (\alpha |\psi|)^n]^m} + \theta_r, \quad (8)$$

$$K(\psi) = K_s \frac{[1 - (\alpha |\psi|)^n-1][1 + (\alpha |\psi|)^n]-m]}{[1 + (\alpha |\psi|)^n]^{m/2}}, \quad (9)$$

where \( \alpha = 0.0335, \theta_s = 0.368, \theta_r = 0.102, n = 2, m = 0.5, K_s = 0.00922 \).

Though MQ is known that exponential accuracy can be obtained, we use the RBF basis \( \varphi(r) = r^9 \) to avoid the choice of that very sensitive free parameter in MQ. We use error tolerance 0.00001 for the iteration convergence.

Example 1. First we consider a 1-D problem of (1) with exact solution \( \psi = -\exp(z - t) \), while the Dirichlet boundary condition, the initial boundary condition and the source term \( f \) are chosen accordingly. The simulation is modeled with 101 uniformly distributed nodes on the domain \([0, 1]\), with time step \( \Delta t = 0.01 \).
The numerical solutions and absolute errors at $t = 1$ using both sets of $\theta(\psi)$ and $K(\psi)$ are presented in Fig.1 and Fig.2 respectively. For comparison purpose, we also presented the results obtained using finite difference method [4] with the same number of nodes. Our results show that the RBF method achieves better accuracy than the finite difference method.

**Example 2.** Here we solve the 2-D problem of (1) with exact solution $\psi = -\exp(x + z - t)$, while the Dirichlet boundary condition, the initial boundary condition and the source term $f$ are chosen accordingly. The physical domain is $[0, 1] \times [0, 1]$. For simplicity, we use 40 uniformly distributed nodes along the domain boundary and time step $\Delta t = 0.01$ in our simulation.

We tested both sets of $\theta$ and $K$, and found that both gave very similar accuracy as in the 1-D problem. Therefore we only presented the results for the first set. In Fig.3 we plotted the numerical solutions and absolute errors obtained by RBF method on 25 uniformly and randomly distributed interior collocation points. Surprisingly, with many runs, we found that the randomly distributed points achieved similar accuracy, but it gave better accuracy than the uniformly distributed points in most cases. Fig. 3 is one of realizations for the case with random points shown in Fig.4 (Left).

Then the problem was solved again by using more interior collocation points. In Fig.5 we plotted the numerical solutions and absolute errors obtained by RBF method on 81 uniformly and randomly distributed interior collocation points. Our numerous experiments showed that the randomly distributed points achieved similar accuracy as the uniformly distributed points. The results presented in Fig.5 is just one realization with the collocation points randomly distributed as in Fig.4 (Right).
Example 3. In order to use our method for practical field application, we investigate the scale effects. Here we resolve Example 2 using 25 uniformly distributed points, but on different sizes of domain \([0, R]^2\). After many experiments, we found that the accuracy really depends on the scale of the physical domain. Results at some collocation points with \(R = 0.1, 0.01, 1, 2\) are presented in Table 1, which shows that the relative errors depends on the modeling domain size. When \(R\) is too small or too large, the solution becomes worse or even diverges. This phenomenon is due to the fact of our basis function \(\psi(r) = r^9\), which can easily result in underflow and overflow when \(R\) becomes too small and too large. Hence for our future investigation of field problems, we have to re-scale the problem domain properly.

4 Conclusions

The RBF collocation meshless method is applied to unsaturated flow problems, which are highly nonlinear time-dependent problems. Numerical examples for both 1-D and 2-D problems demonstrated the efficiency of this so-called Kansa’s method. Practical applications to field problems will be further developed.

Acknowledgements This work was partially supported by U.S. National Science Foundation grant INT-0328186 and Nevada NSF EPSCoR Ring True Program.
Figure 3: Example 2. Numerical solutions and absolute errors obtained by RBF method using 25 uniformly (Top Row) and randomly (Bottom Row) distributed interior collocation points.

References

Figure 4: Example 2. Plots of the randomly distributed collocation points: 25 points (Left) and 81 points (Right).


Figure 5: Example 2. Numerical solutions and absolute errors obtained by RBF method using 81 uniformly (Top Row) and randomly (Bottom Row) distributed interior collocation points.

Table 1: Example 3. Investigation of scale effects on the accuracy. Results are presented at some fixed special collocation points on different sizes of domain \([0, R]^2\).

<table>
<thead>
<tr>
<th>Point ((x, y))</th>
<th>RBF solution</th>
<th>Exact solution</th>
<th>Absolute error</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R = 0.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.0167, 0.0500)</td>
<td>-4.0288e-01</td>
<td>-3.9324e-01</td>
<td>9.6427e-03</td>
<td>0.0245</td>
</tr>
<tr>
<td>(0.0333, 0.0500)</td>
<td>-4.0878e-01</td>
<td>-3.9984e-01</td>
<td>8.9381e-03</td>
<td>0.0224</td>
</tr>
<tr>
<td>(0.0500, 0.0500)</td>
<td>-4.1560e-01</td>
<td>-4.0656e-01</td>
<td>9.0390e-03</td>
<td>0.0222</td>
</tr>
<tr>
<td>(0.0667, 0.0500)</td>
<td>-4.2215e-01</td>
<td>-4.1340e-01</td>
<td>8.7570e-03</td>
<td>0.0212</td>
</tr>
<tr>
<td>(0.0833, 0.0500)</td>
<td>-4.2960e-01</td>
<td>-4.2035e-01</td>
<td>9.2520e-03</td>
<td>0.0220</td>
</tr>
<tr>
<td>(R = 0.01)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.0017, 0.0050)</td>
<td>-3.7869e-01</td>
<td>-3.7034e-01</td>
<td>8.3570e-03</td>
<td>0.0226</td>
</tr>
<tr>
<td>(0.0033, 0.0050)</td>
<td>-3.7861e-01</td>
<td>-3.7095e-01</td>
<td>7.6545e-03</td>
<td>0.0206</td>
</tr>
<tr>
<td>(0.0050, 0.0050)</td>
<td>-3.7935e-01</td>
<td>-3.7157e-01</td>
<td>7.7752e-03</td>
<td>0.0209</td>
</tr>
<tr>
<td>(0.0067, 0.0050)</td>
<td>-3.7983e-01</td>
<td>-3.7219e-01</td>
<td>7.6372e-03</td>
<td>0.0205</td>
</tr>
<tr>
<td>(0.0083, 0.0050)</td>
<td>-3.8113e-01</td>
<td>-3.7281e-01</td>
<td>8.3204e-03</td>
<td>0.0223</td>
</tr>
<tr>
<td>(R = 1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.1667, 0.5000)</td>
<td>-8.1070e-01</td>
<td>-7.1653e-01</td>
<td>9.4175e-02</td>
<td>0.1314</td>
</tr>
<tr>
<td>(0.3333, 0.5000)</td>
<td>-9.7640e-01</td>
<td>-8.4648e-01</td>
<td>1.2992e-01</td>
<td>0.1535</td>
</tr>
<tr>
<td>(0.5000, 0.5000)</td>
<td>-1.1489e+00</td>
<td>-1.0000e+00</td>
<td>1.4897e-01</td>
<td>0.1490</td>
</tr>
<tr>
<td>(0.6667, 0.5000)</td>
<td>-1.3253e+00</td>
<td>-1.1813e+00</td>
<td>1.4402e-01</td>
<td>0.1219</td>
</tr>
<tr>
<td>(0.8333, 0.5000)</td>
<td>-1.5041e+00</td>
<td>-1.3956e+00</td>
<td>1.0854e-01</td>
<td>0.0778</td>
</tr>
<tr>
<td>(R = 2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.3333, 1.0000)</td>
<td>-1.9459e+00</td>
<td>-1.3956e+00</td>
<td>5.5034e-01</td>
<td>0.3943</td>
</tr>
<tr>
<td>(0.6667, 1.0000)</td>
<td>-2.7965e+00</td>
<td>-1.9477e+00</td>
<td>8.4885e-01</td>
<td>0.4358</td>
</tr>
<tr>
<td>(1.0000, 1.0000)</td>
<td>-3.7470e+00</td>
<td>-2.7182e+00</td>
<td>1.0287e+00</td>
<td>0.3784</td>
</tr>
<tr>
<td>(1.3333, 1.0000)</td>
<td>-4.8180e+00</td>
<td>-3.7936e+00</td>
<td>1.0243e+00</td>
<td>0.2700</td>
</tr>
<tr>
<td>(1.6667, 1.0000)</td>
<td>-6.0290e+00</td>
<td>-5.2944e+00</td>
<td>7.3453e-01</td>
<td>0.1387</td>
</tr>
</tbody>
</table>