

B. Tech. Project Report

Development of CONTRADIS

Submitted

By

Man Parvesh Singh Randhawa

Under the supervision of

Dr. T. V. Bharat



**DEPARTMENT OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
APRIL (2016)**

CERTIFICATE

It is certified that the work contained in the project report entitled “**Development of CONTRADIS**”, by **Man Parvesh Singh Randhawa** (1201014039) has been carried out under my supervision and that this work has not been submitted elsewhere for the award of a degree or diploma.

Date:

Dr. T. V. Bharat
Assistant Professor
Department of Civil Engineering
Indian Institute of Technology Guwahati

ACKNOWLEDGEMENT

I would like to thank my supervisor, Dr. T. V. Bharat, for assigning me this project. His friendly nature and supportive attitude was a great motivator for me. From him, I got to lot to learn a lot about research and how to pursue the same.

I dedicate the work to Dr. Bharat and the Department of Civil Engineering, IIT Guwahati.

Date:

Man Parvesh Singh Randhawa

ABSTRACT

CONTRADIS is a software package that is built for the purpose of analyzing the experimental readings obtained from various methods that have been used by the previous studies in the field of contaminant transport in soils. There are different methods that use different parameters to study the flow of different ions/ contaminants through a given sample of soil. This package demonstrates two of those methods, namely, Through Diffusion technique and Half-Cell Method. This software is also capable of using the contaminant transport data from experiments and predicting the diffusion and linear sorption parameters of the soils. The software is verified on the synthetic data obtained from calculated concentration values as an input data.

TABLE OF CONTENTS

	Page no.
Certificate	i
Acknowledgement	ii
Abstract	iii
List of Figures	vi
List of Symbols and Abbreviations	vii
CHAPTER 1: INTRODUCTION	1
1.1 The name	1
1.2 Main objective	1
1.3 Need for this software	1
1.4 Technical Information	1
CHAPTER 2: Theory	2
2.1 Double reservoir or Through Diffusion technique	2
2.2 Half Cell method	5
CHAPTER 3: Solving transcendental equations	7
3.1 Transcendental equation	7
3.2 Transcendental equation used	7
3.3 Standard procedure to solve Transcendental equations	7
3.3.1 Swarm optimization techniques	7
3.3.2 Glowworm swarm optimization	7
3.3.3 Pseudo-code for GSO	8
CHAPTER 4: Inverse analysis	9
4.1 Objective	9
4.2 Particle Swarm Optimization	9

4.2.1 About PSO	9
4.2.2 Pseudo-code for PSO	10
CHAPTER 5: Development of a general .NET application	11
5.1 The .NET Framework	11
5.2 Development of applications based on .NET Framework	11
5.2.1 How to start	11
5.2.2 Types of applications that can be developed	11
5.2.3 Creating a simple forms application using Visual BASIC	11
CHAPTER 6: Development procedure of the proposed software	14
6.1 Creating the User interface	14
6.2 Implementing forward and inverse analysis	14
6.3 Demonstration	14
CHAPTER 7: Future work	18
APPENDIX	19
REFERENCES	19

LIST OF FIGURES

Figure No.	Title	Page No.
2.1	Transient through diffusion set-up (after Bharat, 2012)	2
2.2	Concentration variation from experimental results (after Bharat, 2012)	3
2.3	Comparison of the analytical solution with the experimental results. (after Bharat, 2012)	4
2.4	Infinite case	6
2.5	Finite case	6
5.1	Different options available for types of applications to be developed	12
5.2	Visual Studio interface	12
5.2	Sample user interface in Visual Studio	13
7.1	About section of the software	14
7.2	Introduction tab for Through diffusion	15
7.3	Input tab for Through Diffusion	15
7.4	Results tab for Through Diffusion	16
7.5	Introduction tab for Half Cell method	16
7.6	Input tab for Half Cell method	17
7.7	Results tab for Half Cell method	17

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol	Description
c	Concentration of contaminant
H_s	Equivalent height of source
H_c	Equivalent height of reservoir
α_j	Eigenvalues obtained by solving transcendental equation
D^*	Diffusion coefficient
R_d	Retardation factor
$erfc$	Error function
x	Distance
t	Time
exp	Exponential function
π	Ratio of area of circle to the square of its radius

Introduction

1.1 The name

The name CONTRADIS stands for "CONtaminant TRANsport due to DIffusion in Soils".

1.2 Main objective

This software is built with the vision for the researchers in Geotechnical Engineering who are involved in the research related to contaminant transport phenomenon in soils or similar topics. It would be helpful for them to find the data that should be obtained from their experiments and from their experimental data they would be able to find the diffusion parameters of the soil. From this they would be able to check the correctness of their experiments. Many more uses of this software can be sought that can be beneficial to the researchers in Geotechnical Engineering.

1.3 Need for this software

Currently, POLLUTE is the only one software package available that is widely used by Geo-environmental engineers for contaminant migration analysis. It is being used in the industry since fifteen years. But the drawbacks of this software include non-availability of option for inverse analysis capabilities and high cost, due to which everyone cannot use it.

This software is being built for overcoming the limitations of POLLUTE.

1.4 Technical information

The software package is based on the .NET framework by Microsoft and can run in Windows operating systems, i.e., Windows 8 and later.

Theory

2.1 Double reservoir or Through Diffusion technique

In the double reservoir technique as described by Barone et al., 1992, Rowe et al., 1988 and Shackelford, 1991, the intact plug of barrier material (clay) is placed within the diffusion cell. A source and a collector reservoir is connected to the clay plug as shown in figure 2.1. The solution that is placed in the source solution may be distilled water spiked with the concerned organic species. And the collector solution is of pure distilled water. The initial concentration of the source solution is c_0 . After the introduction of the leachate in the source reservoir mass transport of the chemical constituents takes place by molecular diffusion due to concentration gradient (Shackelford, 1989). Due to the mass transport the concentration of the source reservoir decreases with time and the concentration in the collector reservoir increases with time. The variation of the concentrations in the source and the collector reservoir from experimental results is shown in figure 2.3.

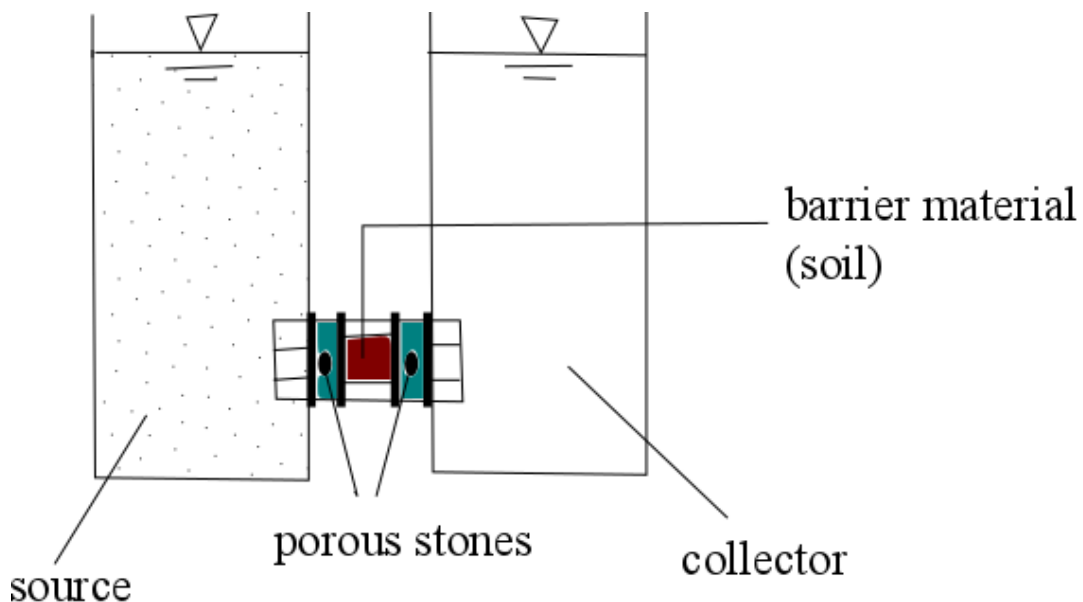


Fig 2.1: Transient through diffusion set-up (after Bharat, 2012)

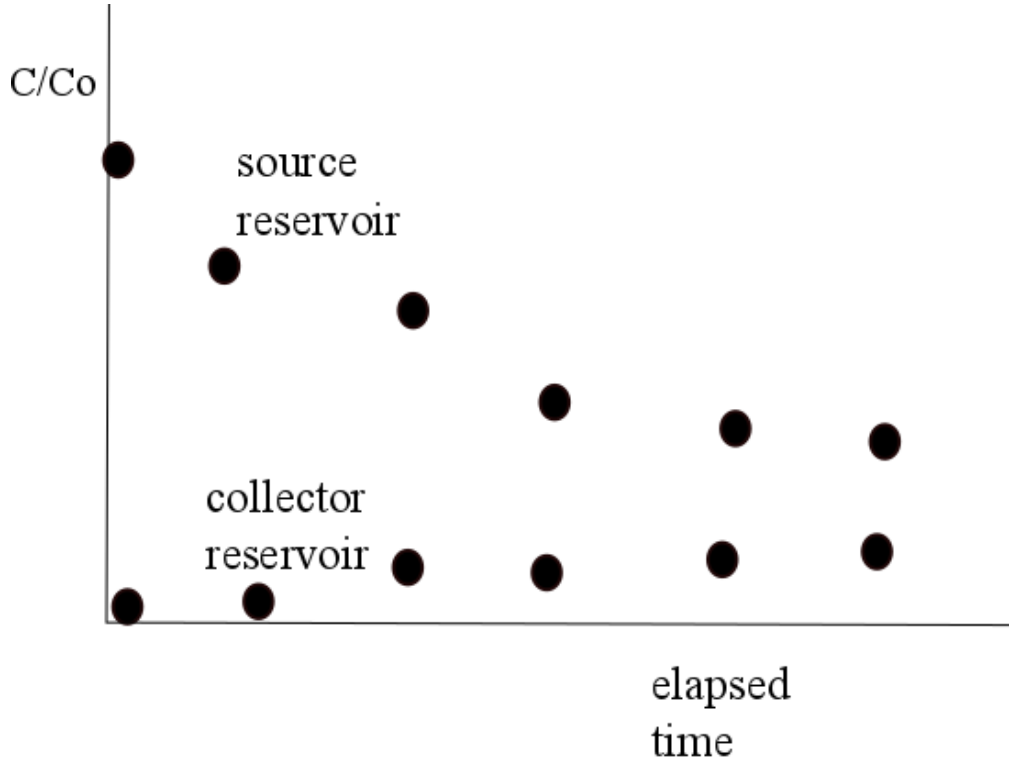


Fig 2.2: concentration variation from experimental results (after Bharat, 2012)

Using mathematical models, mass transport parameters are analyzed from the variation of the observed concentration profiles with time (Bharat, 2012). During the diffusion testing the source and the collector reservoir are monitored periodically to obtain the variation of the organic species concentration with time. Fitting of the theoretical variation of the species with observed experimental variation, it is possible to calculate the value of the diffusion coefficient D^* . The initial and boundary conditions of the model as depicted in fig. 2.2 are:

Initial condition:

$$c(x = 0, t = 0) = c_0$$

$$c(x, t = 0) = 0$$

$$c(x = L, t = 0) = 0$$

Boundary condition:

$$c(x = 0, t) = c_0 + \frac{nD^*}{H_s} \int_0^t \left(\frac{\delta c}{\delta x} \right)_{x=0} dt$$

$$c(x = L, t) = -\frac{nD^*}{H_c} \int_0^t \left(\frac{\delta c}{\delta x} \right)_{x=L} dt$$

where c_0 is the initial concentration of the organic species at time $t = 0$ and H_s and H_c are the equivalent height of the source and collector reservoir respectively.

Using the above initial and boundary conditions the analytical solutions showing the variation of concentration in the source and collector reservoir is given by Bharat, 2013:

$$\frac{c(x = 0, t)}{c_0} = \frac{1}{2 + K} + 2K \sum_{j=1}^{\infty} \frac{(\alpha_j^2 + K^2)e^{-\alpha_j T}}{4K\alpha_j^2(k + 1) - (2K - (\alpha_j^2 - K^2))(\alpha_j^2 - K^2)}$$

$$\frac{c(x = L, t)}{c_0} = \frac{1}{2 + K} - 2K \sum_{j=1}^{\infty} \frac{(\alpha_j^2 + K^2)e^{-\alpha_j T}}{4K\alpha_j^2(k + 1) - (2K - (\alpha_j^2 - K^2))(\alpha_j^2 - K^2)}$$

where $K = Gq = \frac{nRdL}{Hr}$ and $T = \frac{tD^*}{RL^2}$; the values of α_j are obtained by solving the transcendental equation:

$$\tan \alpha_j = \frac{2K\alpha_j}{\alpha_j^2 - K^2}$$

Using several values of D^* and R_d , theoretical curves for the variation of concentration of organic species in the source and the collector reservoir are developed. The curves are best fitted with the curve obtained from experimental data. The best fit curve will give an indication of the D^* value of the particular organic species (Barone, 1992). Figure 4 shows the experimental and theoretical results where the concentration variation are plotted with time and the best fit curve is the one which almost converges.

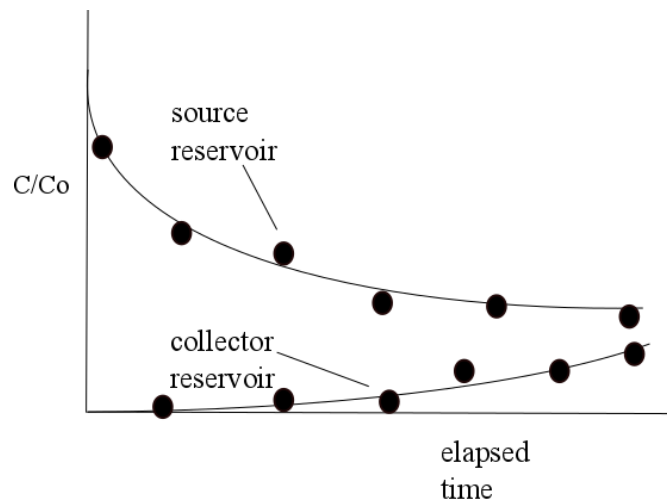


Fig 2.3: comparison of the analytical solution with the experimental results.

(after Bharat, 2012)

2.2 Half Cell method

In order to determine the diffusion coefficient of the contaminants migrating through the barrier material, half-cell method can be employed. Laboratory test simulating one-dimensional diffusive transport is performed through this method. The description of the method can be found in Shackelford (1991). In this method two half cells filled with soil are taken and tagged together. In one of the half-cell the soil is contaminated and in the other half-cell the soil is uncontaminated. As such when both the half cells are connected together there will be migration of contaminants from one half-cell to the other. The contaminated half-cell from which the migration occurs can be termed as source and the other half cell can be termed as receiver. With time the concentration in the source cell decreases as the contaminants diffuses into the receiver cell. After the diffusion test is over the soil mass in each half cell is sliced and analyzed for the variation of the concentration with depth. The analytical solution of diffusion equation given by Shackelford 1991, which is derived from Fick's second law of diffusion equation for soil and is given as:

$$\frac{\delta c}{\delta t} = D^* \frac{\delta^2 c}{\delta x^2} \dots\dots\dots (2.1)$$

where D^* is the effective diffusion coefficient, c is the concentration at any time t and x is the distance. The D^* value is assumed and then theoretical profile is fitted to the experimental data. The solution to equation 2.1 is based upon some initial and boundary conditions where two cases are considered, one is infinite case and the other is finite case.

In the infinite porous medium case the concentration profile does not reach the ends of the half-cell as depicted in figure 2.4, where the origin is taken at the interface of the half cells. The initial and boundary conditions are given as follows:

$$\begin{aligned} c(x \leq 0, t = 0) &= c_0 \\ c(x > 0, t = 0) &= 0 \\ c(x = \infty, t > 0) &= 0 \end{aligned}$$

Corresponding to the above initial and boundary conditions, the solution to equation 2.1 is given by Crank (1975) as:

$$\frac{c}{c_0} = \frac{1}{2} \operatorname{erfc} \left(\frac{x}{2\sqrt{D^*t/R_d}} \right)$$

In the finite porous medium case the concentration profile reaches to the ends of the half-cell as depicted in figure 2.5. The initial and boundary condition for such a case is given as:

$$c(x \leq 0, t = 0) = c_0$$

$$c(x > 0, t = 0) = 0$$

$$\delta c(x = 0, t > 0) / \delta x = 0$$

$$\delta c(x = L, t > 0) / \delta x = 0$$

Corresponding to the above initial and boundary conditions, the solution to equation 1 is given by Carslaw and Jaegar (1959) as:

$$\frac{c}{c_0} = \frac{x_0}{L} + \frac{2}{\pi} \sum_{m=1}^{\infty} \frac{\exp(-D^* m^2 \pi^2 t / R_d L^2)}{m} \cos\left(\frac{m\pi x}{L}\right) \sin\left(\frac{m\pi x_0}{L}\right)$$

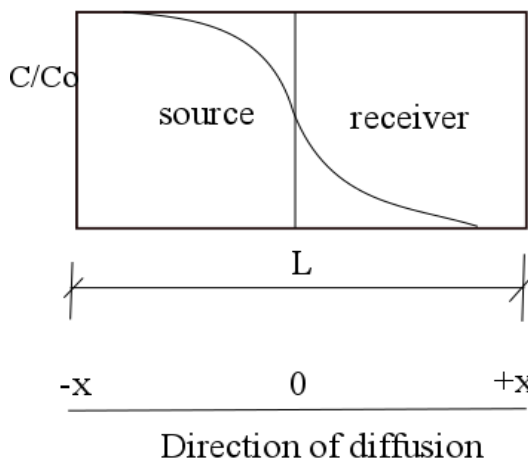


Fig 2.4: Infinite case

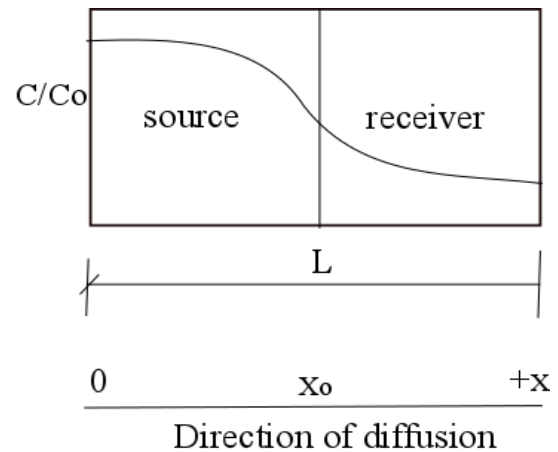


Fig 2.5: Finite case

Solving Transcendental Equations

3.1 Transcendental equation

A transcendental function is an analytic function that does not satisfy a polynomial equation, in contrast to an algebraic function. In other words, a transcendental equation “transcends” algebra in that it cannot be expressed in terms of a finite sequence of the algebraic operations of addition, multiplication, and root extraction.

Examples of a transcendental function include exponential function, the logarithm, and the trigonometric functions.

3.2 Transcendental equation used

In the through diffusion technique, the values of α_j are obtained by solving the transcendental equation:

$$\tan \alpha_j = \frac{2K\alpha_j}{\alpha_j^2 - K^2}$$

3.3 Standard procedure to solve Transcendental equations

3.3.1 Swarm optimization techniques

A variety of iterative techniques, viz. Newton’s method, Newton–Raphson and modified Secant methods, are available for finding the solutions of transcendental equations. These iterative techniques do not yield consistent results due to their dependency on the initial guess solutions. Furthermore, these methods are laborious and inaccurate (Bharat, 2008) for use in inverse analyses where the forward problem will be solved repeatedly several times using different sets of model parameters. A multimodel optimization algorithm (MOA) based on swarm intelligence is used in our work for a quick and accurate estimation of a finite number of transcendental solutions. The description of the algorithm can be found elsewhere (Bharat, 2008, 2009b).

3.3.2 Glowworm swarm

Glowworm swarm optimization (GSO) is a variant for ant-colony optimization and is a powerful technique for finding solutions to optimization of multiple optima continuous functions.

GSO algorithm starts by placing the agents (glowworms) in the search space. Each agent is given with an equal quantity of luminescence initially. Each iteration consists of a luminescence update phase followed by a movement phase based on a transition rule. The luminescence update depends on the function value at the glowworm position. During the luminescence update phase, each glowworm adds to its previous luminescence level, a luminescence quantity proportional to the value of the objective function at that point. Also, a fraction of the previous luminescence value is subtracted from the previous value to simulate the decay in luminescence with time. During the movement phase each glowworm decides, using a probabilistic transition rules to move towards a neighbor that has a luminescence value more than its own. A suitable function is chosen to adaptively update the local-decision domain range of each glowworm.

3.3.3 Pseudo-code for GSO

```

deploy agents randomly
 $\forall i$ , set  $r_i(0) = r_0$ 
set maximum iteration number = iter_max

while ( $t \leq iter\_max$ ) do:
{
  for each glowworm  $i$  do:
  {
     $r_i(t) = find\_dis(i)$ ;
     $f_i(t) = Evaluate\_function$ ;
     $N_i(t) = find\_neighbors(i)$ ;
    for each glowworm  $j \in N_i(t)$  do:
    {
       $pbest = find\_best(j)$ ;
    }
     $x_i(t) \leftarrow x_i(t) + rand( ) * (pbest - x_i(t))$ ;
     $r_i^i(t) = update\_decision\_range(i)$ ;
  }
   $t \leftarrow t+1$ ;
}

```


Inverse Analysis

4.1 Objective

The goal of inverse analysis is to estimate the diffusion and linear sorption parameters of soils such by comparing the experimentally measured and theoretically computed concentration data by minimizing the RMSE (root mean square error). The technique used for inverse analysis is Particle Swarm Optimization (PSO).

4.2 Particle Swarm Optimization

4.2.1 About PSO

PSO is a class of derivative-free, population-based computational method which was developed based on the social behaviors of animals. In this method, each agent representing a potential solution moves in the search space and adaptively updates its velocity and position according to its own flying experience and the flying experience of its neighbors, aiming at a better position for itself. The position of i^{th} particle and j^{th} dimension is represented as $\vec{x}_{ij} = (x_{i1}, x_{i2}, \dots, x_{iD})$. The best position of the i^{th} particle in its history that gives the best fitness value is represented as $\vec{p}_{ij} = (p_{i1}, p_{i2}, \dots, p_{iD})$. The best particle among all the particles in the whole population and in the entire history is represented by $\vec{p}_{gj} = (p_{g1}, p_{g2}, \dots, p_{gD})$. At each iteration step, the position vector of the i^{th} particle x_{ij} is updated by adding an increment vector called velocity v_{ij} as shown below

$$\vec{x}_{ij} = \vec{x}_{ij} + \vec{v}_{ij} \quad (4.1)$$

The velocity of each individual is updated with the best positions acquired for all individuals over generations, and the best positions acquired by the respective individuals over generations. Updating is executed by the following formula.

$$\vec{v}_{ij} = \chi (\omega \vec{v}_{ij} + \phi_1 \text{rand}_1() (\vec{p}_{ij} - \vec{x}_{ij}) + \phi_2 \text{rand}_2() (\vec{p}_{gj} - \vec{x}_{ij})) \quad (4.2)$$

where χ is called constriction coefficient and ω is the inertia weight introduced by Shi and Eberhart in order to improve the performance of the particle swarm optimizer. ϕ_1 and ϕ_2 are two positive values called acceleration constants. $\text{rand}_1()$ and $\text{rand}_2()$ are two independent random numbers that uniformly distribute between 0 and 1.

1 and are used to stochastically vary the relative pull of \vec{p}_{ij} and \vec{p}_{gj} . The positions of respective individuals are updated by every generation, as expressed above.

4.2.2 Pseudo-code for PSO

For each particle

 Initialize particle

END

Do

 For each particle

 Calculate fitness value

 If the fitness value is better than the best fitness value (pBest) in history

 set current value as the new pBest

 End

 Choose the particle with the best fitness value of all the particles as the gBest

 For each particle

 Calculate particle velocity according equation **4.2**

 Update particle position according equation **4.1**

 End

While maximum iterations or minimum error criteria is not attained

Development of a general .NET application

5.1 The .NET Framework

The .NET Framework (pronounced *dot net*), developed by Microsoft and runs primarily on Microsoft Windows, provides a comprehensive programming model for building various kinds of applications, from mobile to web to desktop.

It includes a large class library known as Framework Class Library (FCL) and provides language interoperability across several programming languages like C#, C++, BASIC, etc.

5.2 Development of applications based on .NET Framework

5.2.1 How to start

The primary requirement of building applications on this framework is some prior knowledge of programming and software development. Then a PC, running a latest Operating System (or close), is needed.

The software package required for this purpose is Visual Studio. It is a comprehensive collection of developer tools and services to create applications based the .NET framework.

5.2.2 Types of applications that can be developed

There are several types of application that can be developed in Visual Studio. Some of them include Windows forms applications, windows console applications, windows phone applications, etc.

Visual Studio also allows the developers to use any one of the several supported programming languages to serve their purposes, as demonstrated in fig 5.1.

5.2.3 Creating a simple forms application using Visual BASIC

The mentioned steps can be followed to develop a simple application in Visual Studio.

5.2.3.1 Open Visual Studio and on the menu bar, choose **File, New, Project**. The dialog box should look like shown in Fig 5.1.

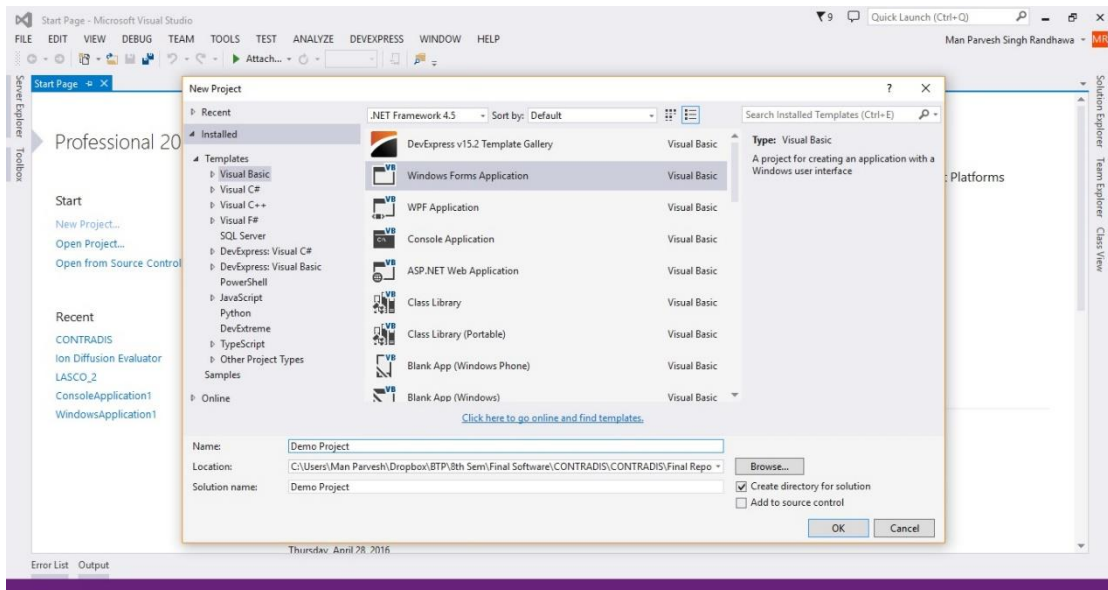


Figure 5.1 Different options available for types of applications to be developed

5.2.3.2 Choose **Visual Basic** in the **Installed Templates** list.

5.2.3.3 In the templates list, choose the **Windows Forms Application** icon. Name your application, and choose the **OK** button.

Visual Studio creates a solution for every program. A solution acts as a container for all of the projects and files needed by the program.

5.2.3.4 Fig 5.2 shows what would be visible in the Visual Studio interface.

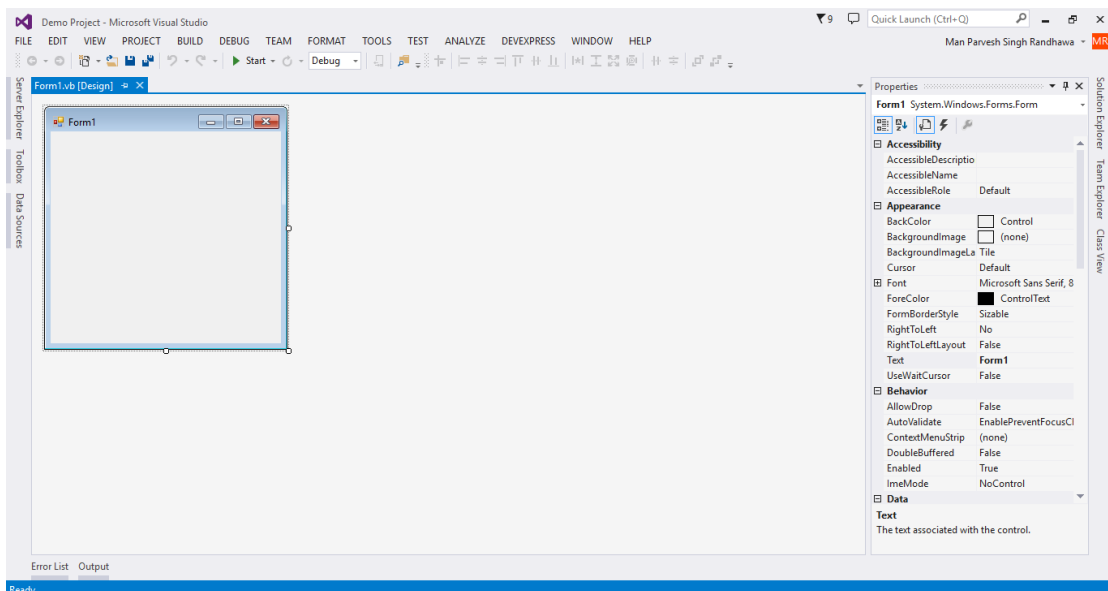


Fig 5.2

The interface contains three windows: a main window, **Solution Explorer**, and the **Properties** window.

5.2.3.5 On the menu bar, choose **File, Save All**.

5.2.3.6 Choose the **F5** key to run the program. Visual Studio runs the program, and a windows as shown in Fig 5.3 would appear.

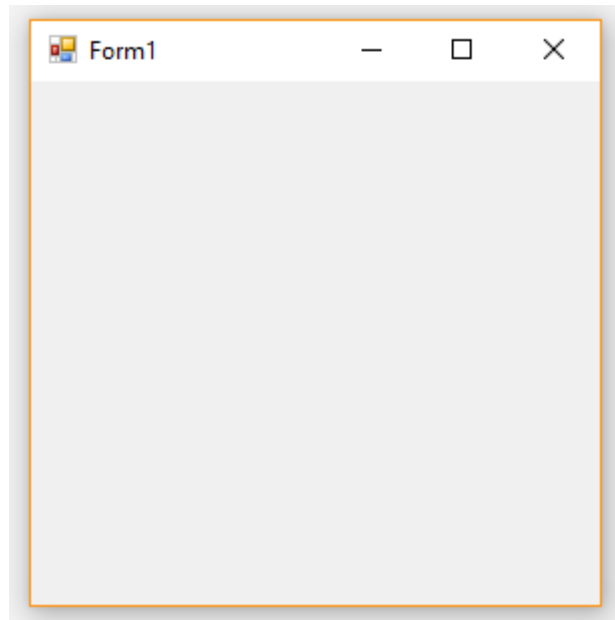


Fig 5.3

5.2.3.7 One can write and edit the code for their program in the code editor in Visual Studio itself.

Development procedure of the proposed software

6.1 Creating the User interface

Using the designer in Visual Studio, all the desirable components, like the tabs, text boxes for input, charts for output, buttons for running the analysis, etc. were added in the user interface.

6.2 Implementing forward and inverse analysis

After this, using the programming language **Visual BASIC**, code snippets were written for both forward and inverse analysis, and added to the main program. After that, code was written to display the results in form of graphs and text.

6.3 Demonstration

A few screenshots of the project that show the working of this software are:

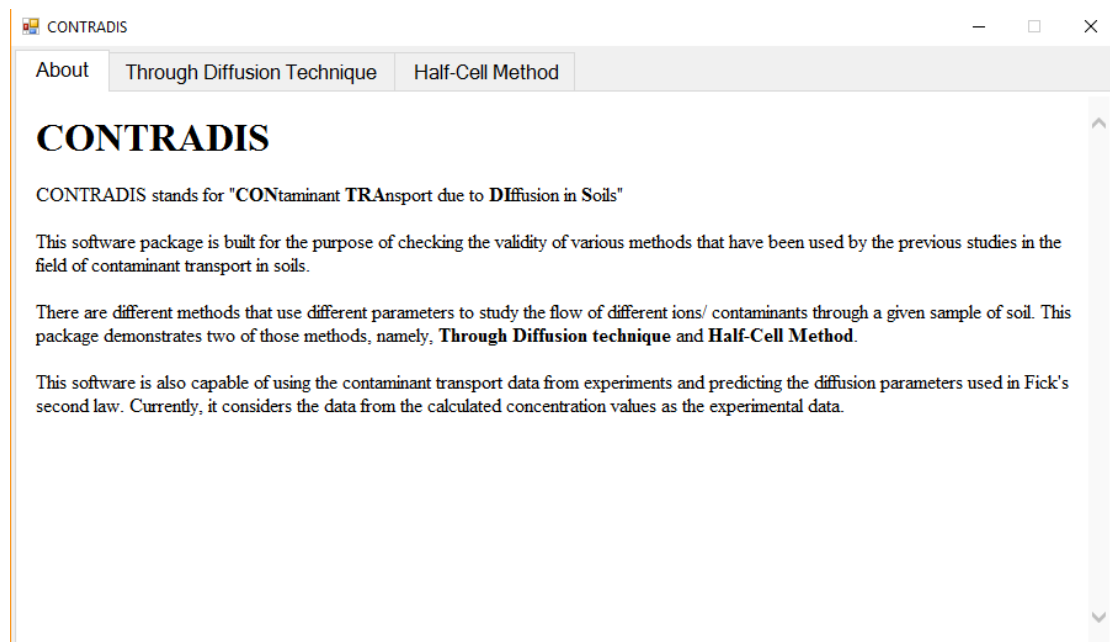


Fig 7.1 About section of the software

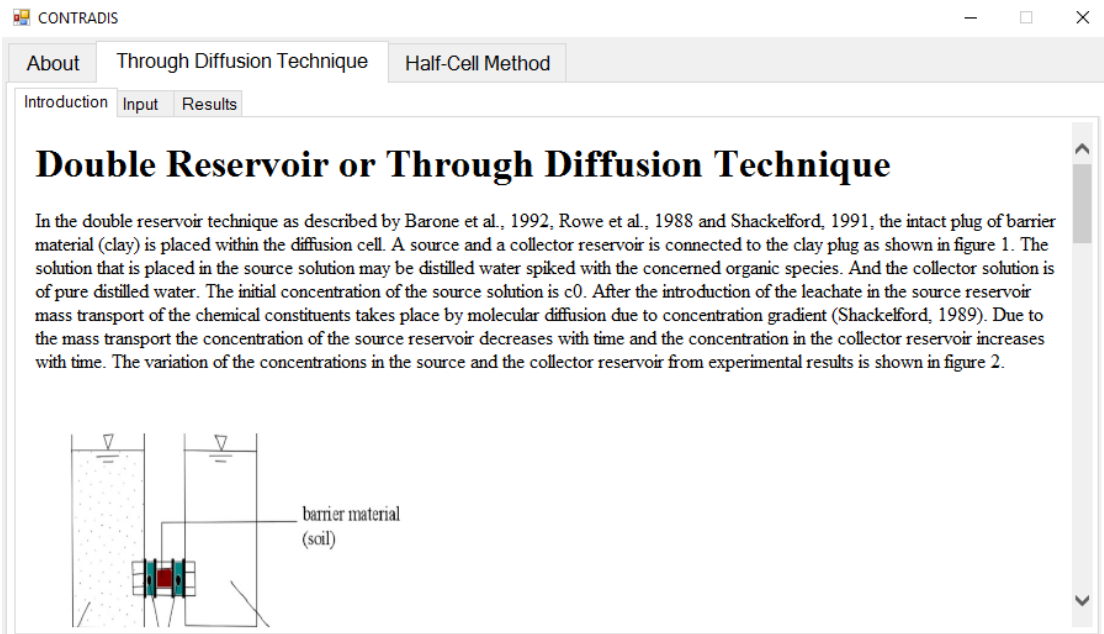


Fig 7.2 Introduction tab for Through diffusion

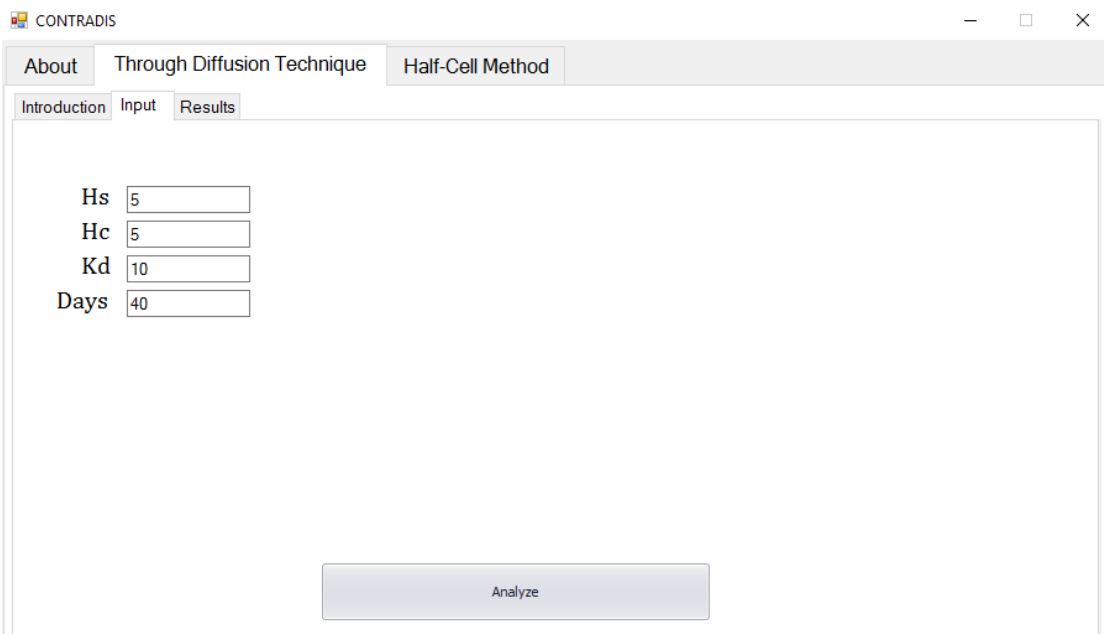


Fig 7.3 Input tab for Through Diffusion

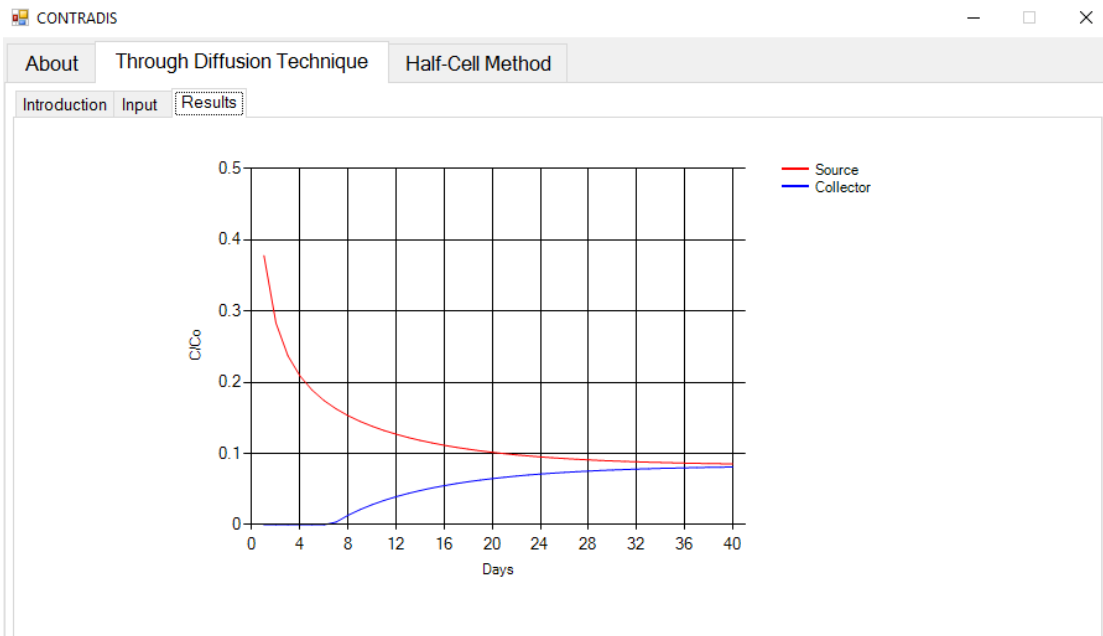


Fig 7.4 Results tab for Through Diffusion

Half-Cell Method

In order to determine the diffusion coefficient of the contaminants migrating through the barrier material, half-cell method can be employed. Laboratory test simulating one-dimensional diffusive transport is performed through this method. The description of the method can be found in Shackelford (1991). In this method two half cells filled with soil are taken and tagged together. In one of the half-cell the soil is contaminated and in the other half-cell the soil is uncontaminated. As such when both the half cells are connected together there will be migration of contaminants from one half-cell to the other. The contaminated half-cell from which the migration occurs can be termed as source and the other half cell can be termed as receiver. With time the concentration in the source cell decreases as the contaminants diffuses into the receiver cell. After the diffusion test is over the soil mass in each half cell is sliced and analyzed for the variation of the concentration with depth. The analytical solution of diffusion equation given by Shackelford 1991, which is derived from Fick's second law of diffusion equation for soil and is given as:

$$\frac{\delta c}{\delta t} = D^* \frac{\delta^2 c}{\delta x^2}$$

where D^* is the effective diffusion coefficient, c is the concentration at any time t and x is the distance. The D^* value is assumed and then theoretical profile is fitted to the experimental data. The solution to equation 1 is based upon some initial and boundary conditions where two cases are considered, one is infinite case and the other is finite case.

Fig 7.5 Introduction tab for Half Cell method

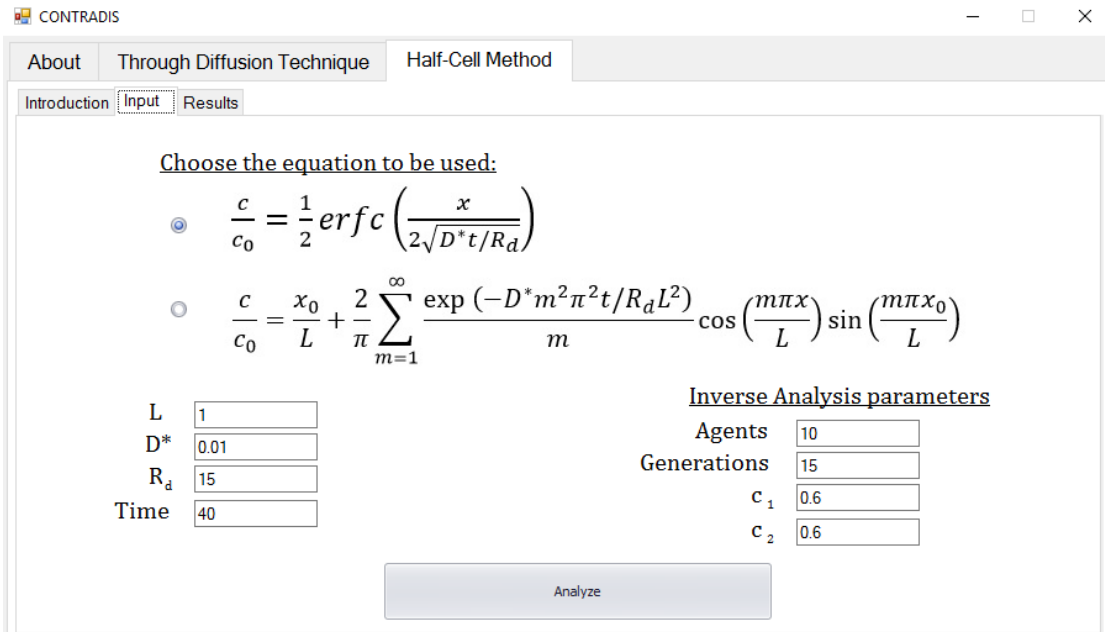


Fig 7.6 Input tab for Half Cell method

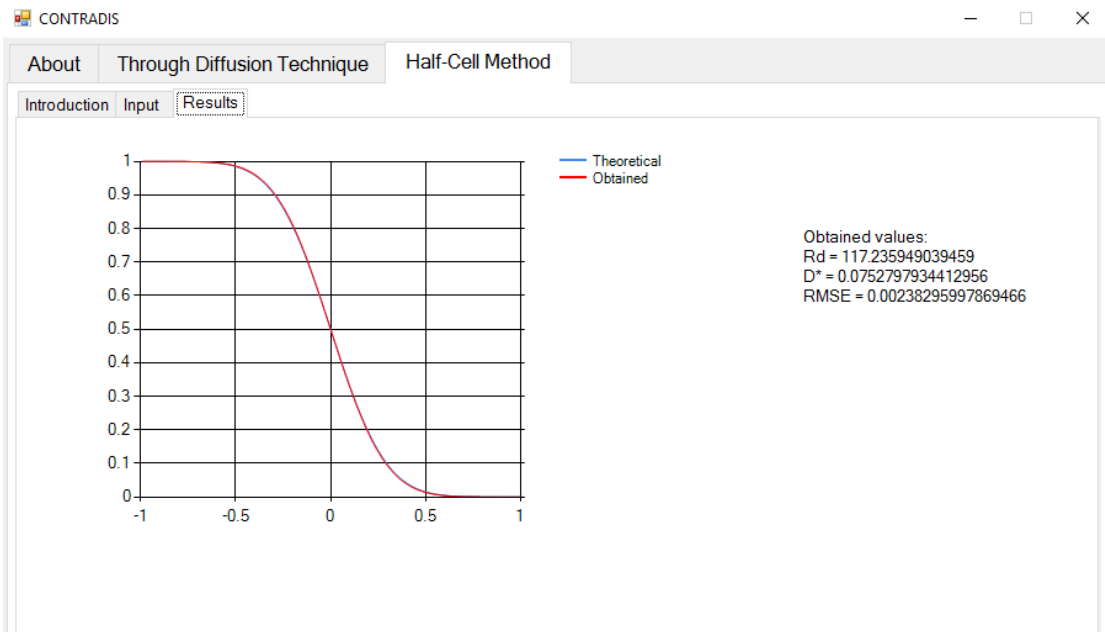


Fig 7.7 Results tab for Half Cell method

Future work

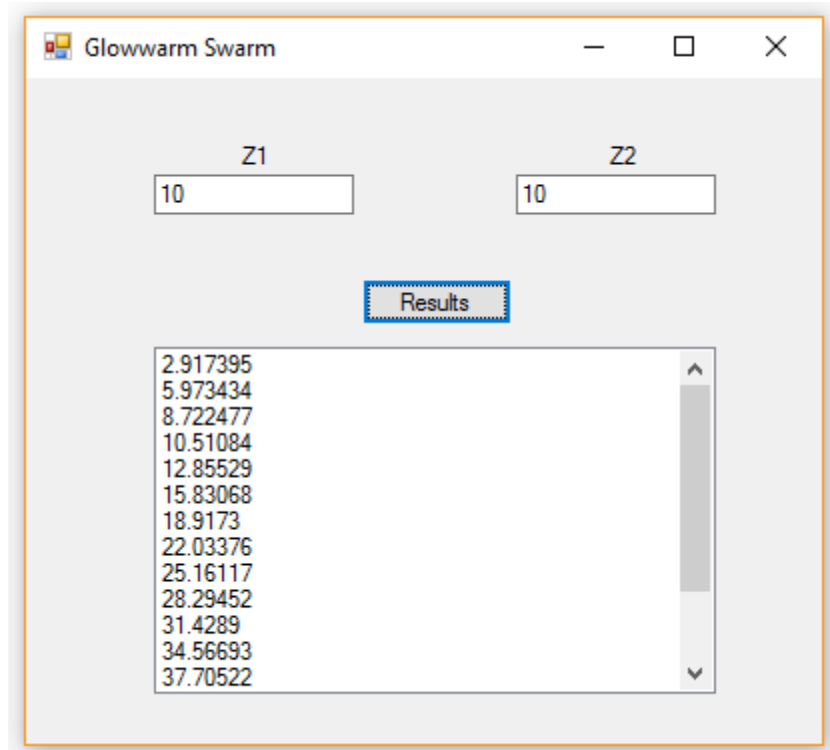
Due to lack of time, the intended goal could not be achieved completely. However, the base for the software is developed and now can be worked upon. The future work planned for this project includes:

- Addition of inverse analysis for Through diffusion technique
- Addition of complete analysis for two more methods, i.e., in-diffusion technique and out-diffusion technique.

APPENDIX

A1: Tests while implementing Glowworm swarm optimization

A graphical user interface (GUI) application was developed to display the results of implementation of Glowworm swarm optimization using data from previous studies. The results obtained were correct.



GUI for Glowworm swarm algorithm

A2: Tests while implementing inverse analysis

A console application was developed to implement inverse analysis using Visual Basic from the data used by Bharat, 2008. The design values for D^* and R_d were taken as $1.5E-6$ and 10.225 respectively. To check the correctness, the root mean square error (RMSE) was computed.

```
D = 1.89252854552332E-06
R = 13.0447863749184

function value at design value = 0.236718105845781
function value at obtained min = 0.217711433871218

RMSE = 0.0309065530188104
```

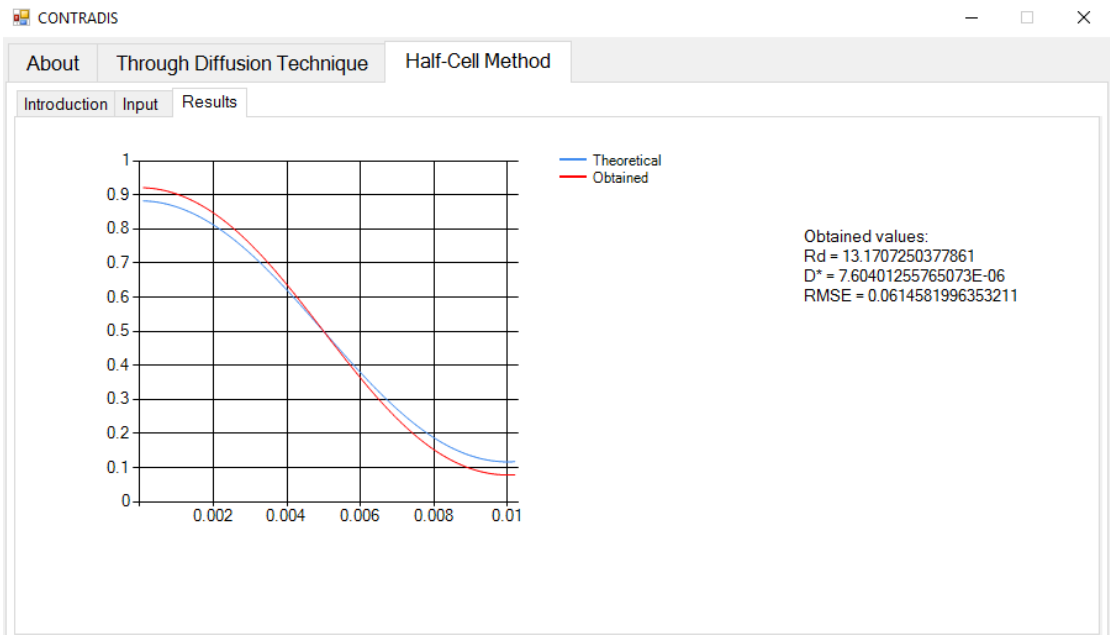
Results of inverse analysis implementation

A3: Validating model from previous data for half-cell method

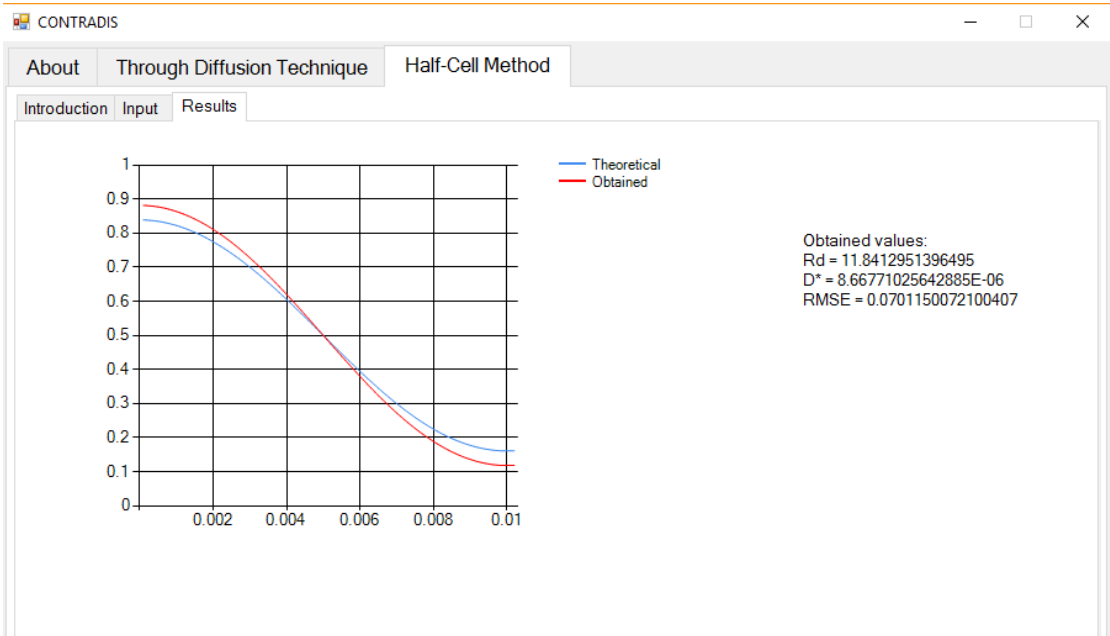
Using the data from the study conducted by Gillham et al. on ^3H in the 50% bentonite mixture, various curves for different diffusion coefficients were obtained.

The results from the proposed software for different D^* and R_d values are:

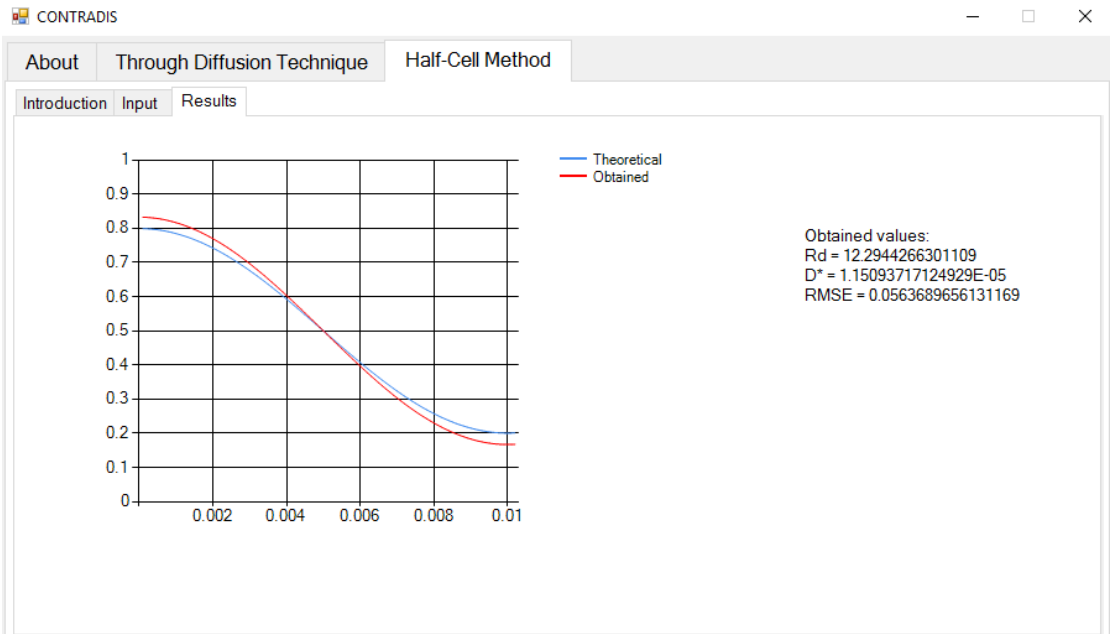
- $D^* = 0.8 * 10^{-5}$ and $R_d = 11$



- $D^* = 1.0 * 10^{-5}$ and $R_d = 11$



- $D^* = 1.2 * 10^{-5}$ and $R_d = 11$



- $D^* = 1.5 * 10^{-5}$ and $R_d = 11$

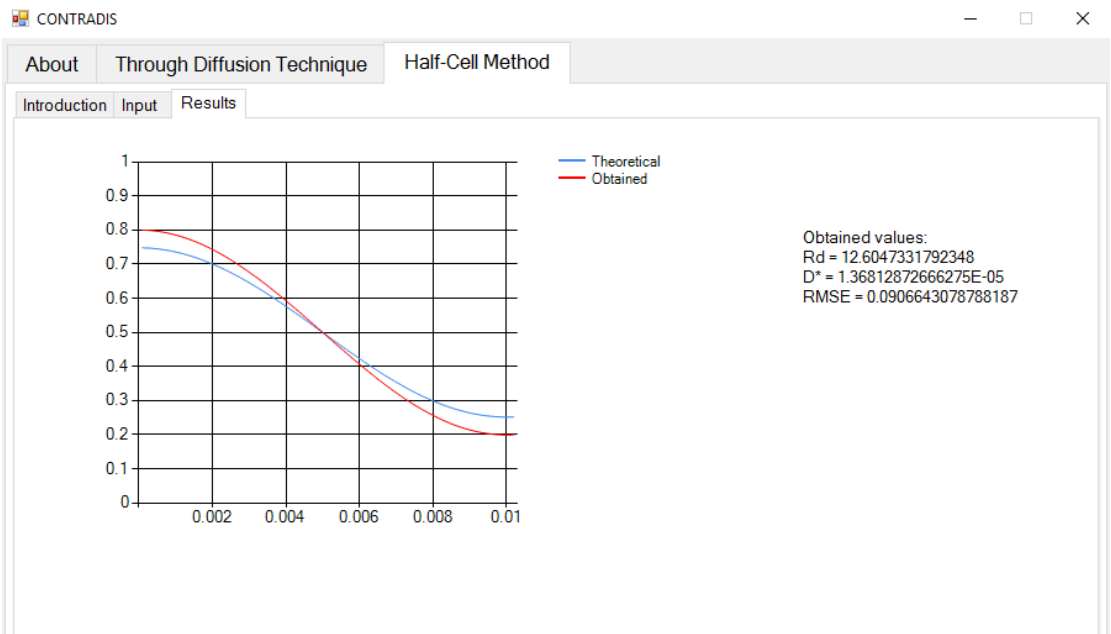
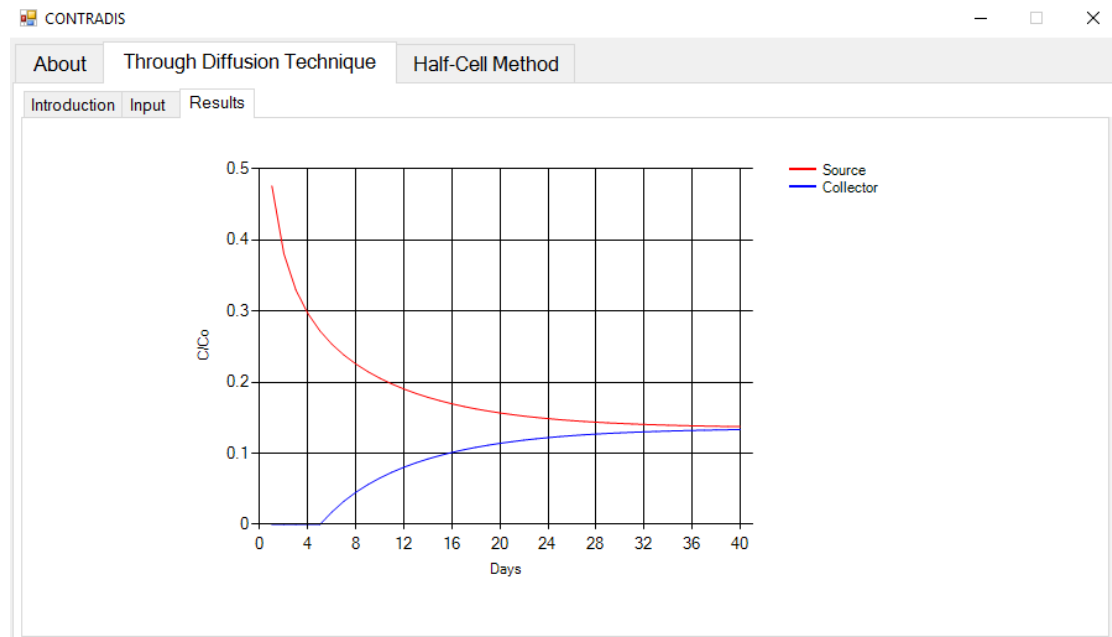


Figure A3.1

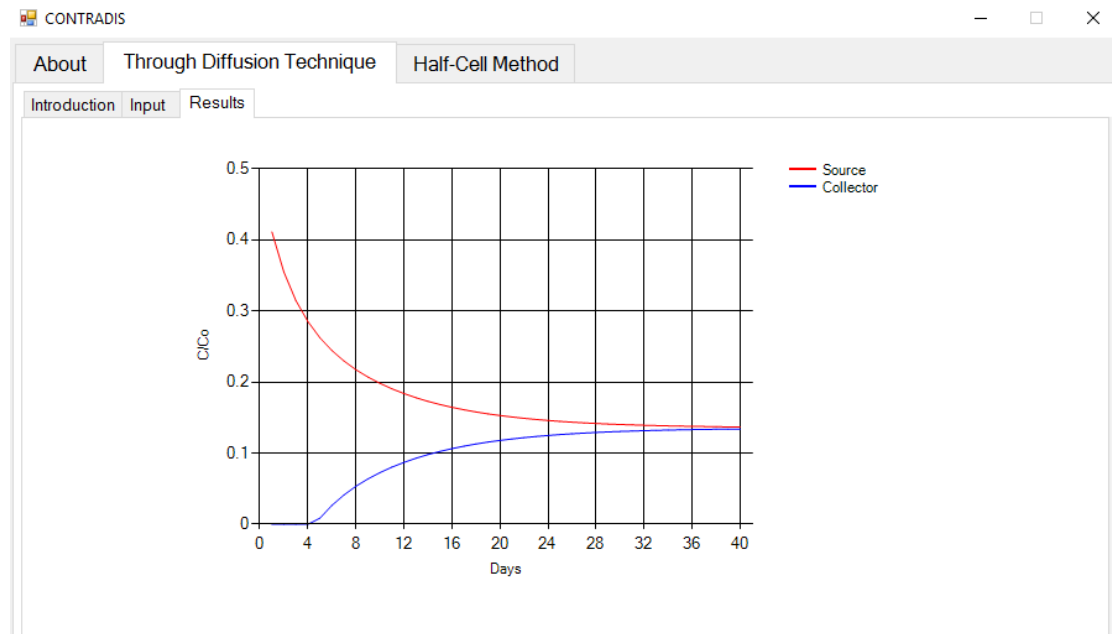
A5: Testing through diffusion technique implementation

A5.1 Using data from study by Bharat, 2013 the output from the software satisfies the results from the study.

For $\alpha = 50$

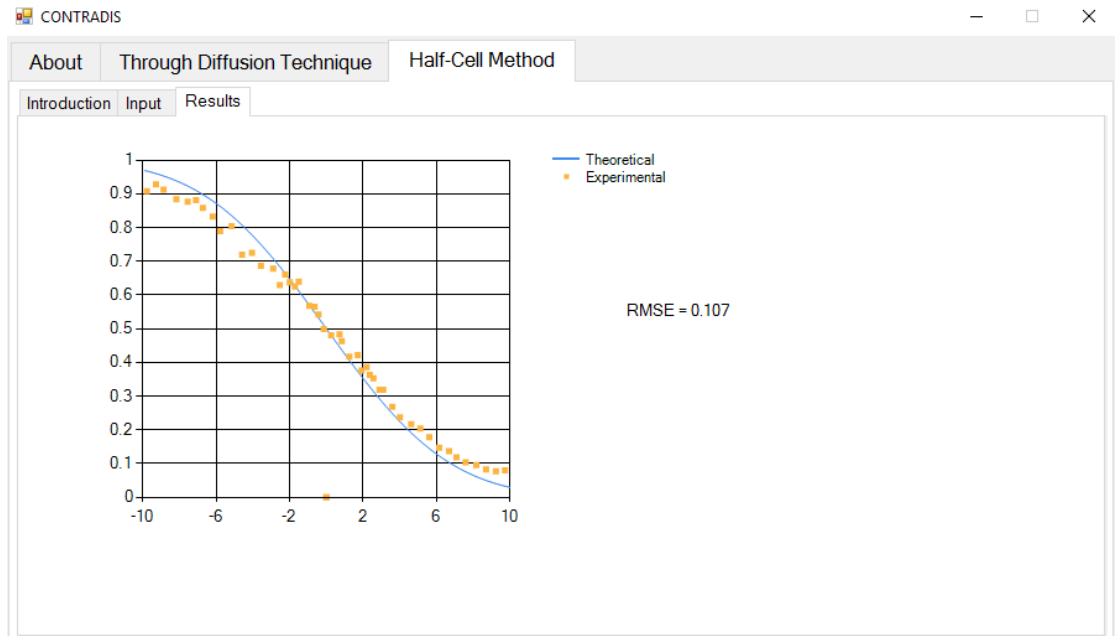


For $\alpha = 100$

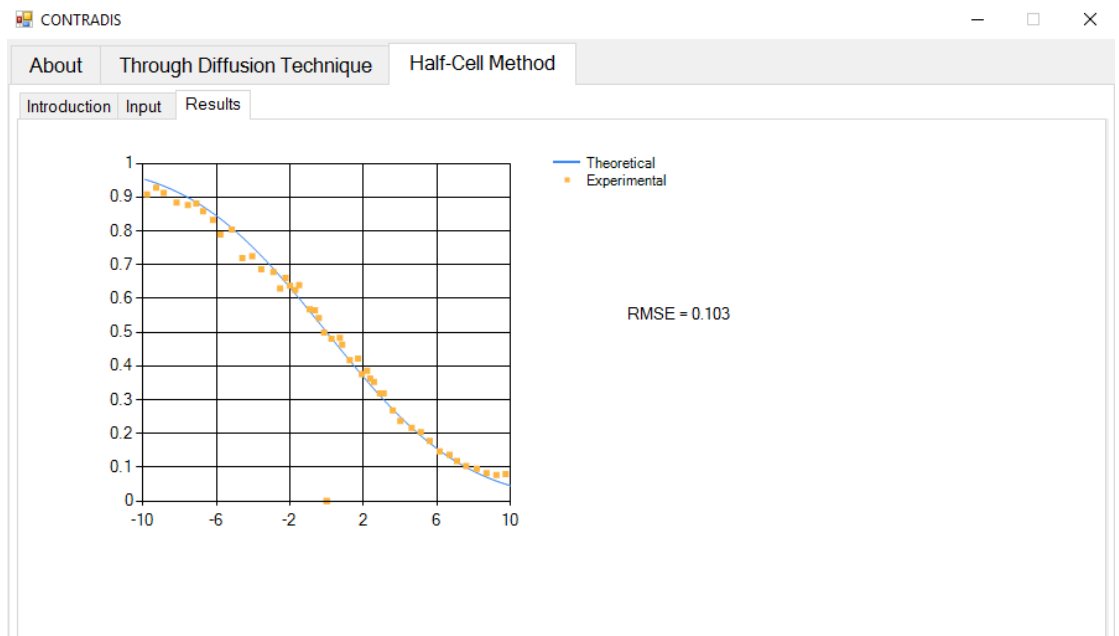


A5.2 Using data from study conducted by Gillham et al., plots were generated using different D^* values to check the fitting of the curve by computing the root mean square error (RMSE).

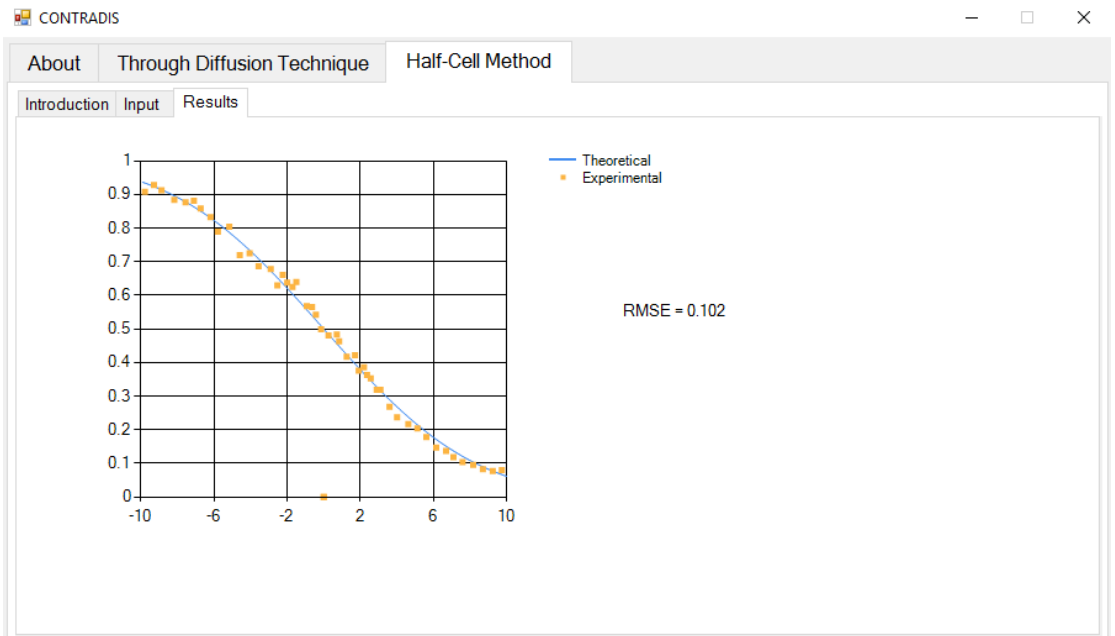
i. For $D^* = 1.0 * 10^{-5}$



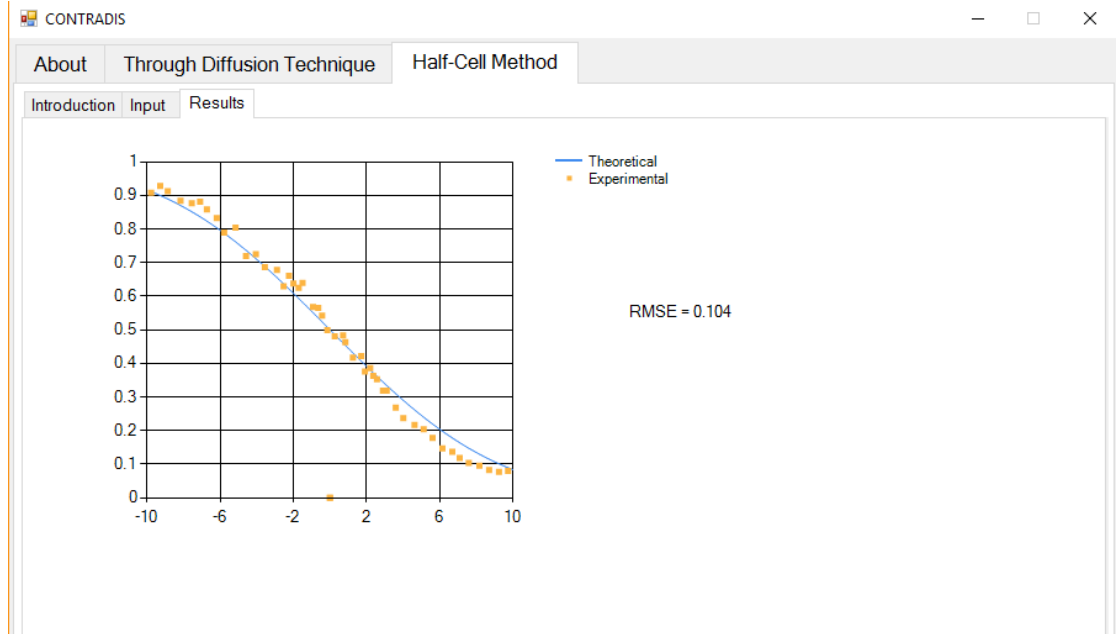
ii. For $D^* = 1.0 * 10^{-5}$



iii. For $D^* = 1.0 * 10^{-5}$



iv. For $D^* = 1.0 * 10^{-5}$



References

- Bharat, Tadikonda Venkata. "Agents based algorithms for design parameter estimation in contaminant transport inverse problems." *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*. IEEE, 2008.
- Bharat, Tadikonda Venkata, Puvvadi V. Sivapullaiah, and Mehter M. Allam. "Swarm intelligence-based solver for parameter estimation of laboratory through-diffusion transport of contaminants." *Computers and Geotechnics* 36.6 (2009): 984-992.
- Bharat, Tadikonda Venkata. "Analytical model for 1-D contaminant diffusion through clay barriers." *Environmental Geotechnics* 1.4 (2014): 210-221.
- Tadikonda Venkata Bharat, Puvvadi V. Sivapullaiah, Mehter M. Allam (2009), "Swarm intelligence-based solver for parameter estimation of laboratory through-diffusion transport of contaminants", *Computers and Geotechnics* 36 (2009) 984–992
- Groundwater software, "Pollute – Contaminant Transport Modeling Software", < <http://www.groundwatersoftware.com/pollute.htm>>
- Microsoft corporation, "Microsoft .NET home", <<https://www.microsoft.com/net/default.aspx>>
- Wikipedia, ".NET Framework" < https://en.wikipedia.org/wiki/.NET_Framework>
- CD Shackelford (1989), "Diffusion of contaminants through waste containment barriers", *Transportation Research Record*, 1219
- Tadikonda Venkata Bharat, Puvvadi V Sivapullaiah , Mehter M Allam (2012), "Robust solver based onmodified particle swarm optimization for improved solution of diffusion transport through containment facilities", *Expert Systems with Applications* 39 (2012) 10812–10820
- F.S. Barone, R.K. Rowe, R.M. Quigley (1992), "A laboratory estimation of diffusion and adsorption coefficients for several volatile organics in a natural clayey soil", *Journal of Contaminant Hydrology*, Volume 10, Issue 3, August 1992, Pages 225-250
- Shackelford, C. and Daniel, D. (1991). "Diffusion in Saturated Soil. I: Background." *J. Geotech. Engrg.*, 10.1061/(ASCE)0733-9410(1991)117:3(467), 467-484.

- J. Crank, Radhey S. Gupta (1975), "Isotherm Migration Method in Two Dimensions", *Int. J. Heat Mass Transfer*. Vol. 18, pp. 1101-1107
- Carslaw, Horatio Scott, and John Conrad Jaeger. "Conduction of heat in solids." Oxford: Clarendon Press, 1959, 2nd ed. (1959).
- Microsoft Corporation, "Getting started with Visual C# and Visual Basic", < <https://msdn.microsoft.com/en-us/library/dd492171.aspx>>
- Microsoft Corporation, "Create a Windows Forms Application Project", < <https://msdn.microsoft.com/en-IN/library/dd492132.aspx>>
- Microsoft Corporation, "Visual Basic Language Reference", < <https://msdn.microsoft.com/en-us/library/sh9ywfdk.aspx>>
- Gillham, R. W., et al. "Diffusion of nonreactive and reactive solutes through fine-grained barrier materials." *Canadian Geotechnical Journal* 21.3 (1984): 541-550.
- Barry, D. A. "Modelling contaminant transport in the subsurface: Theory and computer programs." *Modelling Chemical Transport in Soil: Natural and Applied Contaminants*. Lewis Publishers, Boca Raton, Florida (1992): 105-144.
- Barone, F. S., R. K. Rowe, and Robert M. Quigley. "A laboratory estimation of diffusion and adsorption coefficients for several volatile organics in a natural clayey soil." *Journal of contaminant hydrology* 10.3 (1992): 225-250.
- Rowe, R. K., J. R. Booker, and J. Fraser. "POLLUTE v. 6.3: 1D pollutant migration through a non-homogeneous soil." *Distributed by GAEA Environmental Engineering Ltd, Whitby, Ontario* (1997).