

Modeling Ground Water Problems Using the Complex Polynomial Method

A. W. Bohannon and
T. V. Hromadka, AS-0020

Abstract

Numerical methods for solving the governing partial differential equations involved with ground water flow continue to be the subject of research and development. A goal in such research is to develop and refine numerical techniques in order to provide an increase in numerical accuracy and reduce the computational effort of applying such techniques. In this paper, the Complex Polynomial Method (CPM) is applied towards approximating problems of ground water flow. Because the CPM exactly solves the governing PDE and does not involve the discretization of the problem domain such as required by use of the typical domain methods of finite differences and finite elements, the CPM affords many advantages over such typically used domain methods.

Keywords:

ground water, numerical methods, Complex Polynomial Method, complex polynomials, modeling

Introduction

The development of new numerical techniques or the extension of existing techniques towards application to problems of ground water flow continues to be the subject of research and development. The vast majority of computer modeling programs in use today for solving ground water flow problems involve application of the domain numerical techniques of finite differences or finite elements. Both numerical techniques involve the discretization of the problem domain and boundary into a myriad of nodal points interconnected by linkages or domain elements. Both the finite differences and finite element methods have the following properties: they do not exactly solve the governing partial differential equations (PDE) that describe ground

water flow over the problem domain; they do not exactly solve the boundary conditions continuously on the problem boundary; and they do not provide a simple approach for evaluating numerical error in the modeling estimates.

In contrast, the CPM exactly solves the governing PDE of the Laplace or Poisson type over the problem domain, and it does not discretize the problem domain. The CPM involves the use of complex variable monomials with complex coefficients determined by fitting the complex polynomial to the prescribed boundary conditions according to a selected measure of fit such as collocation or least squares error minimization, among other measures of fit. Because complex variable monomials add to form a complex variable polynomial, the resulting approximation function is an entire function (analytic over the entire complex plane) and therefore composed of two real valued two-dimensional conjugate functions, representing the equipotential function and the streamline function of the ground water problem solution. These two conjugate functions each exactly solve the Laplace equation over the problem domain, so modeling error analysis is accomplished by using the property that the maximum (and also the minimum) magnitude of modeling error is located on the problem boundary. That is, because the approximation function and the exact solution to the subject ground water problem both exactly solve the Laplace equation (under mild conditions that may involve splitting the problem domain into homogeneous and possibly rescaled domains to accommodate anisotropic domains), the modeling error defined as the difference between these two functions also solves the Laplace equation over the problem domain; therefore, the Maximum Modulus Theorem applies from standard complex variables theory,

resulting in the property that the maximum magnitude of the modeling error must occur on the problem boundary. Consequently, the maximum magnitude of the modeling error throughout the entire problem domain and boundary is determined by simply evaluating the magnitude of the difference between the resulting CPM function and the prescribed boundary conditions along the problem boundary (both functions are known continuously along the problem boundary based on geohydrologic assumptions used to interpolate between ground water level readings on the problem boundary).

The first considerations of using the CPM occurred in the paper of Hromadka and Guymon (1984) [2], but further research and development in the CPM was limited by the computer power available at that time. CPM functions could not be efficiently developed to include high degree terms of complex monomials, resulting in limited applicability to complicated ground water problems; however, the theoretical advantages of using the CPM had been established. With the advent of mathematically oriented programs such as MATLAB, Mathematica, and others, computational power was readily available, providing the ability to work accurately with high degrees of complex monomials and thus solve large non-symmetrical matrices and the resulting matrix systems. Furthermore, such programs contain impressive graphical and internal capabilities that provide considerable demonstrative power and involve surprisingly few programming steps.

In the current paper, the CPM is applied towards modeling a ground water basin where several ground water supply wells are in use. At a well located nearby but outside of the study area, an increase in its extraction rate is analyzed as to its impacts on the exist-

ing groundwater pumping trends at the other ground water wells. In order to mitigate the impacts of the subject increased extraction rate, an increase in ground water banking at a nearby ground water banking percolation basin is examined. The resulting CPM model provides both the equipotentials and the conjugate streamlines for the subject application problem, which are plotted together using Mathematica. The entire computer code, as well as examples of the input and output for this problem, is included as Appendix A, which demonstrates the small computer programming effort needed in order to apply the CPM. The user can directly modify the Mathematica code and see an update to the associated results and graphics, a property available when using such mathematically based computer programs.

Background

The CPM is a numerical procedure that uses a set of complex variable monomials with complex coefficients to form a complex variable polynomial for use as an approximation function. The monomial coefficients are calculated by satisfying the problem boundary conditions. Because complex monomials can be resolved into two, real variable two-dimensional functions, commonly known as the real and imaginary parts of the complex function, both parts are handled as individual functions. Hromadka and Guymon (1984) first developed the CPM variant of the Complex Variable Boundary Element Method (CVBEM) and successfully applied it to a limited set of engineering problems [2]; however, generally available computational power of the time limited the CPM to low degree complex polynomials.

The CPM was recently used to solve PDE of the Laplace equation type using the computer program, Mathematica, which provides a significant increase in numerical accuracy achieved by use of standard computer programs based on computer languages such as FORTRAN and others. As a result of developing the CPM application on Mathematica, Poler et al [4] showed that complex polynomials in excess of degree thirty-five were computationally efficient, and the various graphical features of Mathematica were directly employable to such boundary value problems. This computational success provided a considerable advantage over the other computer solutions applied with the CPM such as seen in

[2] and has returned the CPM as being a strong topic for further research. The computational accuracy provided by such off-the-shelf computer programs such as Mathematica and MATLAB may open the door for more mathematical solutions of such boundary value problems which involve exact solutions of the PDE, and less reliance on numerical techniques that only approximate solutions to the PDE.

Theory

Complex polynomials are entire functions, being analytic over the entire complex plane; therefore, both of the real and imaginary parts of the complex polynomial satisfy the Laplace Equation. Additionally, the real and imaginary parts of the complex function form a conjugate pair which can represent the streamline function and the associated potential function as parts of the solution to the boundary value problem. This feature of the CPM alone provides considerable mathematical advantages over the usual domain numerical methods in common use.

In the current paper, the CPM is extended to use a least-squares error minimization technique to match the problem boundary conditions continuously along the entire problem boundary and not just at a set of points located on the boundary. With this new approach for the CPM, convergence is guaranteed as the complex polynomial degree increases (see Theorem provided in [5]), and the computational advantages afforded by Mathematica apply as reported in [4].

Method

To apply the least-squares error minimization approach to developing the CPM coefficients, the numerical procedure presented in [1] is used. Only a brief description of this Best Approximation technique is presented here for the reader's convenience.

We are given a domain and boundary on which ground water flow is modeled with a complex function subject to the Laplace Equation. The real part of the function models the ground water level, and the imaginary part models the ground water flow streamlines. A set of boundary conditions representing the ground water levels on the boundary are given. We want to determine the best approximation of the governing PDE

using a complex polynomial that is fitted to the problem boundary conditions.

We choose a set of linearly independent basis functions and construct a set of global vectors from those basis functions. We ortho-normalize those global vectors using the Gramm-Schmidt process, and use the resulting basis vectors to approximate the solution as a vector in Hilbert space. The complex coefficients of best fit are the generalized Fourier coefficients.

Using the generalized Fourier coefficients and a back-substitution routine, the coefficients of the approximation function are determined that best approximate the problem boundary conditions in a least squares sense.

Application

An important problem in civil engineering and geohydrology is the analysis of ground water flow impacts due to changes in pumpage or introduction of new ground water wells in an aquifer. Typically, domain type numerical methods are applied towards approximation of the governing PDE describing ground water flow. In the following application problem, the CPM is applied. Use of the CPM considerably reduces data input requirements over the requirements needed by domain type numerical techniques for similar problems.

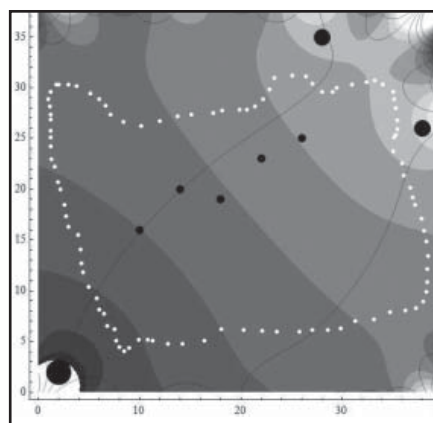
In this application, the various ground water flow properties are assumed to be homogeneous and isotropic throughout the problem domain. It is noted that non-homogeneous or anisotropic properties of the ground water regime, such as parameters for porosity, permeability, and so forth, can be accommodated by re-scaling or by solving simultaneous sub-problems, using the CPM (see [3] for examples using the CVBEM numerical technique). The CPM is applied to an irregular domain covering a 1,200 square mile area where boundary conditions of ground water levels are known from readings obtained from water wells and borings located along the problem boundary. Two large extraction wells are located to the northeast of the subject study area and a large ground water banking percolation basin is located to the southwest of the study area. All CPM models use a complex polynomial of degree fifteen, comprising thirty-one complex monomials. It is noted that the complex monomials are generated internally by the provided Mathematica code [4].

Two CPM models are developed in Figure 1, representing the original problem initial conditions and the steady-state conditions corresponding to a twofold increase in the extraction rate at the most northerly extraction well. Figure 1(a), which is obtained from the Mathematica application developed for this problem, shows the CPM model of ground water levels under the initial scheme for pumping and ground water banking. In comparison, Figure 1(b) shows the steady-state water levels corresponding to the increased extraction rate. The third and fourth CPM models, in Figure 2, determine what increase in ground water banking is needed to mitigate the impacts of the increased ground water extraction. Figure 2(a) shows the model which returns all wells to a level at or above their pre-extraction level. It requires a nearly fifty percent increase in the ground water banking rate at the percolation basin. Figure 2(b) offers an alternative solution, where the original average ground water level is achieved. It requires twenty percent less increase in the ground water banking of the percolation basin compared to the previous option. The boundary values from the CPM of all four models are continuously graphed in Figure 3 for comparison.

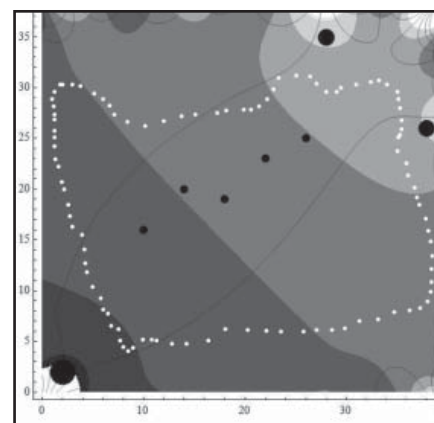
Conclusions

The CPM is applied to ground water problems using the computer program Mathematica, although similar programs such as MATLAB could also be used. The CPM develops a complex polynomial that exactly solves the governing PDE of ground water flow over the problem domain and approximately solves the problem boundary conditions. Because computer programs that use typical finite difference and finite element methods do not exactly solve the governing PDE and, like the CPM, only approximately fit the problem boundary conditions continuously on the problem boundary, the CPM provides a considerable improvement in modeling accuracy. Additionally, such domain methods involve the discretization of the problem domain and boundary, whereas the CPM uses no discretization at all. The CPM provides two real variable two-dimensional conjugate functions, representing the problem streamline function and problem potential function, respectively.

Modeling error analysis is readily accomplished by examining the modeling function fit to the problem bound-

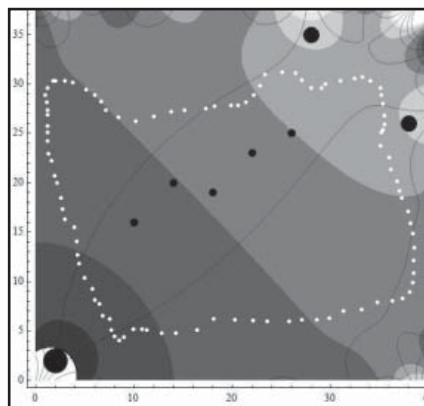


(a) Initial steady state approximate solution of in-situ conditions.

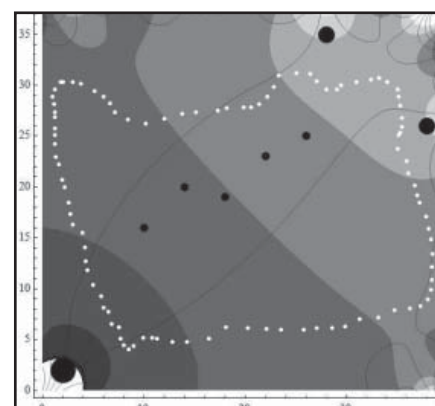


(b) Final steady state approximate solution with impact of increased extraction rate.

Figure 1. CPM models of subject study area before and after the increase in the extraction rate of the most northerly extraction well. The ground water levels are depicted with the contour shading, where lighter shades are the higher levels, and the ground water flow streamlines are shown as solid lines. The boundary conditions from water wells and borings are shown as small white dots. The percolation basin is the large black dot in the bottom-left; the extraction wells are the large black dots in the top-right; and the wells of interest are the black dots inside the subject study area.



(a) First adjusted steady state approximate solution with re-charge.



(b) Second adjusted steady state approximate solution with re-charge.

Figure 2. CPM models of subject study area after the increased in-flow at the ground water banking percolation basin. The same format from Figure 1 is used.

ary conditions because the maximum magnitude of modeling error within the problem domain is less than or equal to the maximum magnitude of error in fitting the problem boundary conditions. The difference in boundary condition estimates corresponding to measured values can be reduced by including additional complex monomials in the resulting complex polynomial approximation function. For this application problem, the complex polynomial degree used is fifteen. It is noted that the CPM model power can be easily increased by simply entering a higher number in the provided program when prompted to enter the number of basis functions.

The presented application problem evaluates the impacts of increasing the extraction rate of ground water at a regional water supply well and then determining the mitigating offset by increasing ground water banking at a banking percolation basin to return the interior water supply well ground water levels close to pre-extraction levels. The Mathematica code and application is included in the Appendix.

Acknowledgements

The authors wish to thank the United States Military Academy at West Point for the support and opportunity to work on this area of research and development. Particularly, the authors wish to

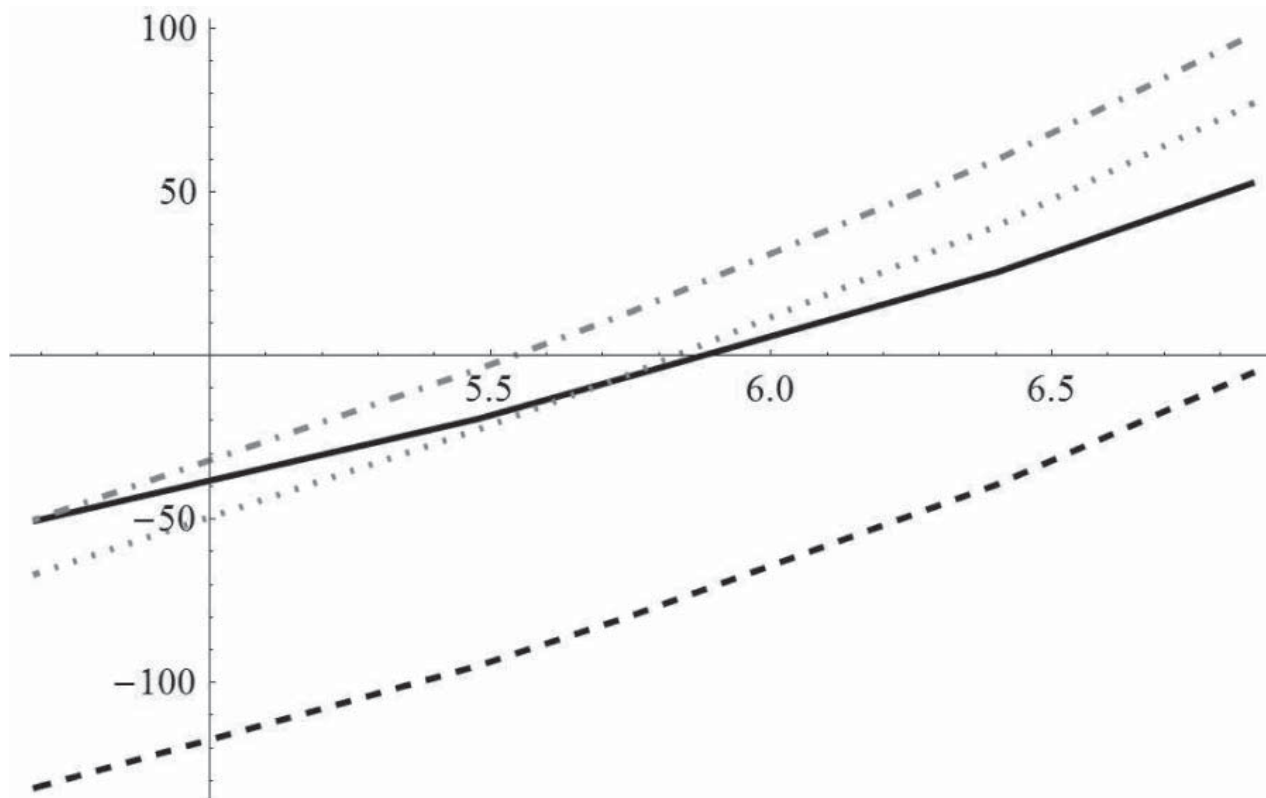


Figure 3. All four CPM models are compared across wells of interest. The horizontal axis represents the linear distance from the percolation basin in miles. The vertical axis represents the ground water level. The original ground water levels are depicted with the solid black line. The ground water levels resulting from the increased extraction rate are depicted with the black dashed line. The first and second adjusted solutions are the gray dot-dashed line and dotted line, respectively.

thank the Head of the Department of Mathematical Sciences, COL Michael Phillips, and Deputy Head, COL Steve Horton, for their support and encouragement and for development of the AIAD opportunities for the summer 2008 program. In particular, we wish to thank LTC Michael Jaye for his constant encouragement, energy, and intellectual excitement in motivating research of this type.

References

1. T. V. Hromadka, 1993, *The Best Approximation Method in Computational Mechanics*, Springer-Verlag, London, 250 pages.
2. V. Hromadka and G. L. Guymon, 1984, Complex Polynomial Approximation of the Laplace Equation, *ASCE Journal of Hydraulic Engineering*, Vol. 110, No. 3, 329-339.
3. T. V. Hromadka and R. J. Whitley, 1998, *Advances in the Complex Variable Boundary Element Method*, Springer-Verlag, New York, 400 pages.
4. A. C. Poler, A. W. Bohannon, S. J. Schowalter, and T. V. Hromadka, 2008, Using the Complex Polynomial Method with Mathematica to Model Problems Involving the Laplace and Poisson Equations, *The Journal of Numerical Methods for Partial Differential Equations*, (in press).
5. R. J. Whitley and T. V. Hromadka, 2005, Approximating Harmonic Functions on R^n With One Function of a Single Complex Variable, *The Journal of Numerical Methods for Partial Differential Equations*, Vol. 21, No. 5, 905-917.

Reviewed by AIPG Associates
Editors: Tom Jordan, CPG-09384,
 Neill Ridgley, CPG-05138, and Doug
 Perisutti, CPG-10055

Microsoft Excel - Adjusted.xls

File Edit View Insert Format Tools Data Window Help Adobe PDF

Type a question for help

100% Arial 10 B I U

Ready with Changes End Review

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1	28.86	-16.0341											
2	1.1	28.12	-18.5199											
3	1.23	27.38	-21.0465											
4	1.23	26.862	-23.1352											
5	1.238	25.752	-27.7966											
6	1.24	25.086	-30.7404											
7	1.246	24.272	-34.4765											
8	1.25	22.94	-40.9702											
9	1.6	22.2	-43.92											
10	1.9	20.72	-51.3186											
11	2.2	19.98	-54.8966											
12	2.6	18.5	-63.1256											
13	2.689	17.32	-70.8874											
14	3	16.28	-77.5851											
15	3.9	15.54	-80.5526											
16	4.18	14.06	-91.7474											
17	4.22	12.713	-103.747											
18	4.4	11.84	-111.552											
19	5	10.36	-124.327											
20	5.738	9.255	-132.199											
21	6	8.14	-143.161											
22	6.48	7.72	-143.735											
23	6.8	6.512	-153.97											
24	7.5	6.216	-148.721											
25	7.69	5.114	-156.253											
26	8	4.44	-157.061											
27	8.5	4.07	-152.279											
28	9	4.366	-143.783											
29	9.9	5.18	-128.079											
30	10.8	5.2	-118.054											
31	11.3	5.11	-113.276											
32	12.78	4.78	-100.203											
33	14.23	4.8	-87.1717											
34	16.43	5.1	-68.5529											
35	18.123	6.2	-52.3965											
36	20.3	6.142	-38.6738											
37	22.1	6.033	-28.5786											
38	23.56	5.987	-20.8585											
39	25.8	6.003	-9.71966											
40	27.1	6.11	-3.42918											
41	28.566	6.168	2.959654											
42	29.88	6.3	8.585022											
43	31.34	6.98	16.55577											
44	33.123	7.214	23.24366											
45	34.789	7.86	30.6607											
46	36.31	8.08	35.27248											
47	37.28	8.333	38.40577											
48	38.004	8.94	42.44116											
49	38.4	9.87	47.36351											
50	38.458	10.86	52.21866											
51	38.46	12.16	58.89763											
52	38.502	13.45	66.09742											
53	38.332	14.86	74.41311											
54	38.14	15.93	81.25401											
55	37.89	17.12	89.54963											
56	37.245	18.46	99.56259											

Sheet1 / Sheet2 / Sheet3

Figure A1. Example Microsoft Excel file for importing into Mathematica. No column headings should be included. The three columns represent the horizontal coordinate, the vertical coordinate, and the ground water level.

x	y	Potential
1	28.86	-16.03413
1.1	28.12	-18.51991
1.23	27.38	-21.04653
1.23	26.862	-23.13524
1.238	25.752	-27.79659
1.24	25.086	-30.74037
1.246	24.272	-34.47645
1.25	22.94	-40.9702
1.6	22.2	-43.91996
1.9	20.72	-51.31856
2.2	19.98	-54.89665
2.6	18.5	-63.12561
2.689	17.32	-70.8874
3	16.28	-77.58513
3.9	15.54	-80.55262
4.18	14.06	-91.74736
4.22	12.713	-103.7472
4.4	11.84	-111.5519
5	10.36	-124.3272
5.738	9.255	-132.1991
6	8.14	-143.1609
6.48	7.72	-143.7354
6.8	6.512	-153.9704
7.5	6.216	-148.7214
7.69	5.114	-156.2532
8	4.44	-157.0613
8.5	4.07	-152.2785
9	4.366	-143.7832
9.9	5.18	-128.0786
10.8	5.2	-118.0535
11.3	5.11	-113.2759
12.78	4.78	-100.2033
14.23	4.8	-87.17173
16.43	5.1	-68.55292
18.123	6.2	-52.39653
20.3	6.142	-38.6738
22.1	6.033	-28.57864
23.56	5.987	-20.8585
25.8	6.003	-9.719663
27.1	6.11	-3.429184
28.566	6.168	2.959664
29.88	6.3	8.585022
31.34	6.98	16.55577
33.123	7.214	23.24366
34.789	7.86	30.6607
36.31	8.08	35.27248
37.28	8.333	38.40577
38.004	8.94	42.44116
38.4	9.87	47.36351
38.458	10.86	52.21866

x	y	Potential
38.46	12.16	58.89763
38.502	13.45	66.09742
38.332	14.86	74.41311
38.14	15.93	81.25401
37.89	17.12	89.54963
37.245	18.46	99.56259
36.999	19.17	105.4776
36.754	20.12	114.161
36.118	21.34	125.5208
35.98	22.567	139.5994
35.1	23.754	147.9014
35.15	25.086	162.0197
35.31	25.382	166.511
35.5	25.9	172.6389
35.44	26.788	174.0469
35.32	27.972	171.0961
35.19	28.86	168.1216
35.1	29.6	166.1738
34	30.34	163.6946
33.2	30.71	163.7237
32.4	30.59	163.2936
31	30.34	161.4119
29.5	29.97	155.9527
29.1	29.6	151.3431
28	29.6	146.5095
27.08	30.414	149.5393
26.4	31.08	151.765
25.1	31.228	140.469
23.3	30.932	119.5397
22.85	29.822	108.2389
22.2	28.86	96.83998
21.4	28.12	86.22891
20.6	27.8684	78.81
19.9	27.824	73.43426
18.2	27.75	61.04279
17.3	27.528	53.93778
15.1	27.38	39.34705
13.8	27.232	31.05445
12	26.714	18.93662
10.2	26.27	7.922375
8.4	26.64	1.237313
7.1	27.38	-1.166709
6.667	28.22	0.444095
6	28.86	0.277848
5.124	29.456	-0.808806
3.8	30.192	-2.84677
3	30.34	-4.948541
2	30.34	-8.012084
1.8	30.34	-8.599612
1.2	29.6	-12.84332

Figure A2. Example data from the first adjusted solution.

Imports a chart from Microsoft Excel in the specified file. The chart contains evaluation points with which to solve the PDE. The user is prompted to the number of complex monomial basis functions, n , to approximate the solution.

```
ImportedData = Flatten[Import["File Path"], 1];
G = Take[ImportedData, All, 2];
Solution = ImportedData[[All, 3]];
n = Input["Number of complex monomial basis functions: "];
```

Generates the real and imaginary parts of the approximate solution with n complex monomial basis functions and $(2n+1)$ unknown coefficients.

$$\phi[x_-, y_-] = \lambda_1 + \text{ComplexExpand}\left[\text{Re}\left[\sum_{m=1}^n (\lambda_{2m} + i \lambda_{2m+1}) (x + i y)^m\right]\right];$$

$$\psi[x_-, y_-] = \text{ComplexExpand}\left[\text{Im}\left[\sum_{m=1}^n (\lambda_{2m} + i \lambda_{2m+1}) (x + i y)^m\right]\right];$$

Constructs a matrix, where each column is a vector with a basis function evaluated at each evaluation point.

```
A = Table[Derivative[0, 1] phi[G[[i, 1]], G[[i, 2]]], {i, Length[G]}, {j, 2 n + 1}];
VectorSet = Table[A[[All, i]], {i, 2 n + 1}];
Do[E_m = VectorSet[[m]], {m, 2 n + 1}]
```

Orthonormalizes the columns of the A-matrix according to Gramm - Schmidt using the standard inner product, the dot product.

```
OrthoNormVectSet = Orthogonalize[VectorSet];
Do[E_hat_m = OrthoNormVectSet[[m]], {m, 2 n + 1}]
```

Calculates the generalized Fourier coefficients by taking the inner product of the given potentials and the orthonormal basis vectors.

```
Do[L_hat_m = Solution.E_hat_m, {m, 2 n + 1}]
```

Reverses the orthonormalization through back substitution to find the coefficients for the approximate solution.

```
Do[C_hat_i = Take[RowReduce[Join[Transpose[Take[VectorSet, i]], Transpose[{E_hat_i}], 2]], {1, i}, -1],
{i, 2 n + 1}]
Do[Lambda_hat_1 = Sum[L_hat_m C_hat_m[[i]], {i, 2 n + 1}],
{m, 1, 2 n + 1}]
```

Outputs the final approximate solution using n complex monomial basis functions.

```
phi[x, y]
psi[x, y]
```

Outputs the potential function, stream function, and evaluation points in a three-dimensional plot.

```
Graph1 = Plot3D[phi[x, y], {x, Min[G[[All, 1]]] - .1, Max[G[[All, 1]]] + .1},
{y, Min[G[[All, 2]]] - .1, Max[G[[All, 2]]] + .1}, PlotStyle -> Hue[0]];
Graph2 = Plot3D[psi[x, y], {x, Min[G[[All, 1]]] - .1, Max[G[[All, 1]]] + .1},
{y, Min[G[[All, 2]]] - .1, Max[G[[All, 2]]] + .1}, PlotStyle -> Hue[0.15]];
Graph3 = Graphics3D[{PointSize[Large], Hue[0.7], Point[ImportedData]}];
Show[Graph1, Graph2, Graph3]
```

Outputs the potential function, stream function, and evaluation points in a contour plot.

```
Basin = {2, 2};
ExtractWells = {{28, 35}, {38, 26}};
Graph4 = ContourPlot[phi[x, y], {x, 0, 40}, {y, 0, 38}, ContourStyle -> None, DisplayFunction -> Identity,
AspectRatio -> Automatic, ColorFunction -> "GrayTones"];
Graph5 = ContourPlot[psi[x, y], {x, 0, 40}, {y, 0, 38}, ContourShading -> False, DisplayFunction -> Identity,
AspectRatio -> Automatic];
Graph6 = Graphics[{PointSize[.01], White, Point[G]}];
Graph7 = Graphics[{PointSize[.04], Black, Point[ExtractWells]}];
Graph8 = Graphics[{PointSize[.06], Black, Point[Basin]}];
Graph9 = Graphics[{PointSize[.02], Black, Point[Wells]}];
Show[Graph4, Graph5, Graph6, Graph7, Graph8, Graph9]
```

Figure A3. Generic CPM Mathematica code. The full file path must be specified (ie. "D:\My Documents\..."). The number of basis functions directly corresponds to the degree of the complex polynomial approximation function.

Outputs the final approximate solution using n complex monomial basis functions.

```

 $\phi[x, y]$ 
 $\psi[x, y]$ 
[-888.499 + 98.9759 x + 43.9996 x^2 - 12.0338 x^3 - 0.987564 x^4 - 0.000115439 x^5 - 0.00497181 x^6 - 0.000322603 x^7 - 7.54293 x 10^-6 x^8 - 5.06693 x 10^-8 x^9 - 6.59549 x 10^-9 x^10 - 1.38901 x 10^-10 x^11 - 1.03678 x 10^-12 x^12 - 2.60535 x 10^-15 x^13 - 7.09349 x 10^-17 x^14 - 2.27422 x 10^-19 x^15 - 321.991 y - 129.043 x y - 4.06305 x^2 y + 3.5658 x^3 y - 0.541743 x^4 y - 0.0271924 x^5 y - 0.000192186 x^6 y - 0.0000764192 x^7 y - 3.18439 x 10^-6 x^8 y - 4.76135 x 10^-8 x^9 y - 2.62395 x 10^-10 x^10 y - 1.76528 x 10^-11 x^11 y - 2.05119 x 10^-13 x^12 y - 7.38497 x 10^-16 x^13 y - 4.04652 x 10^-18 x^14 y - 43.9996 y^2 - 36.1013 x y^2 - 5.92538 x^2 y^2 - 0.00115439 x^3 y^2 - 0.0745772 x^4 y^2 - 0.00677466 x^5 y^2 + 0.000211202 x^6 y^2 - 1.82409 x 10^-6 x^7 y^2 - 2.96797 x 10^-7 x^8 y^2 - 7.63954 x 10^-9 x^9 y^2 - 6.84274 x 10^-11 x^10 y^2 - 2.03217 x 10^-13 x^11 y^2 - 6.45507 x 10^-15 x^12 y^2 - 2.38793 x 10^-17 x^13 y^2 - 1.35435 x^4 y^3 - 3.5658 x^5 y^3 - 1.08349 x^6 y^3 - 0.0906414 x^7 y^3 - 0.000960928 x^8 y^3 - 0.000534934 x^9 y^3 - 0.000029721 x^10 y^3 - 5.71362 x 10^-7 x^11 y^3 - 3.93593 x 10^-9 x^12 y^3 - 3.23635 x 10^-10 x^13 y^3 - 4.51261 x 10^-12 x^14 y^3 - 1.92009 x 10^-14 x^15 y^3 - 1.22744 x 10^-17 x^16 y^3 - 0.987564 y^4 + 0.000577194 x y^4 - 0.0745772 x^2 y^4 - 0.0112911 x^3 y^4 - 0.000528005 x^4 y^4 - 6.38433 x 10^-6 x^5 y^4 - 1.38505 x 10^-8 x^6 y^4 - 4.58373 x 10^-9 x^7 y^4 - 5.13206 x 10^-10 x^8 y^4 - 1.86282 x 10^-12 x^9 y^4 - 7.10058 x 10^-14 x^10 y^4 - 3.10431 x 10^-16 x^11 y^4 - 0.108349 y^5 - 0.0271924 x y^5 - 0.000576557 x^2 y^5 - 0.000534934 x^3 y^5 - 0.0000445815 x^4 y^5 - 1.19986 x 10^-6 x^5 y^5 - 1.10206 x 10^-8 x^6 y^5 - 1.16509 x 10^-9 x^7 y^5 - 2.03068 x 10^-11 x^8 y^5 - 1.05605 x 10^-13 x^9 y^5 - 8.10113 x 10^-17 x^10 y^5 - 0.00497181 y^6 - 0.00225822 x y^6 - 0.000211202 x^2 y^6 - 4.25622 x 10^-6 x^3 y^6 - 1.38505 x 10^-8 x^4 y^6 - 6.41722 x 10^-8 x^5 y^6 - 9.57984 x 10^-10 x^6 y^6 - 4.47078 x 10^-12 x^7 y^6 - 2.13017 x 10^-13 x^8 y^6 - 1.13825 x 10^-15 x^9 y^6 - 0.0000274551 y^7 - 0.0000764192 x y^7 - 0.0000127376 x^2 y^7 - 5.71362 x 10^-7 x^3 y^7 - 7.87186 x 10^-9 x^4 y^7 - 1.16509 x 10^-9 x^5 y^7 - 2.70757 x 10^-11 x^6 y^7 - 1.81037 x 10^-13 x^7 y^7 - 1.73596 x 10^-16 x^8 y^7 - 7.54293 x 10^-6 y^8 - 4.56023 x 10^-7 x^2 y^8 - 2.96797 x^3 y^8 - 2.29186 x 10^-6 x^4 y^8 - 5.13206 x 10^-10 x^5 y^8 - 3.35308 x 10^-12 x^6 y^8 - 2.13017 x 10^-13 x^7 y^8 - 1.46346 x 10^-15 x^8 y^8 - 3.53822 x 10^-7 y^9 - 4.76135 x 10^-8 x y^9 - 1.31198 x 10^-8 x^2 y^9 - 3.23635 x 10^-10 x^3 y^9 - 1.12815 x 10^-11 x^4 y^9 - 1.05605 x 10^-13 x^5 y^9 - 1.35019 x 10^-16 x^6 y^9 - 6.59549 x 10^-9 y^10 - 1.52791 x 10^-8 x y^10 - 6.84274 x 10^-11 x^2 y^10 - 7.45129 x 10^-13 x^3 y^10 - 7.10058 x 10^-14 x^4 y^10 - 6.82949 x 10^-16 x^5 y^10 - 2.38541 x 10^-17 x^6 y^10 - 1.76528 x 10^-11 x^2 y^11 - 1.23071 x 10^-12 x^3 y^11 - 1.92009 x 10^-14 x^4 y^11 - 3.68233 x 10^-17 x^5 y^11 - 1.03678 x 10^-12 x^6 y^11 - 3.38695 x 10^-14 x^7 y^11 - 6.45507 x 10^-13 x^8 y^11 - 1.03477 x 10^-16 x^9 y^11 - 1.57784 x 10^-14 x^10 y^11 - 7.38497 x 10^-16 x^11 y^11 - 2.83256 x 10^-18 x^12 y^11 - 7.09349 x 10^-17 y^14 - 3.41133 x 10^-18 x y^14 - 2.69768 x 10^-20 x^2 y^14]
[-321.991 x + 64.5215 x^2 - 1.35435 x^3 - 0.891449 x^4 - 0.108349 x^5 - 0.00453207 x^6 - 0.0000274551 x^7 - 9.55239 x 10^-6 x^8 - 3.53822 x 10^-7 x^9 - 4.76135 x 10^-8 x^10 - 2.38541 x 10^-11 x^11 - 1.47107 x 10^-13 x^12 - 1.57784 x 10^-16 x^13 - 5.27498 x 10^-17 x^14 - 2.69768 x 10^-20 x^15 - 98.9759 y + 87.9993 x y - 36.1013 x^2 y - 3.55026 x^3 y + 0.000577194 x^4 y - 0.0298309 x^5 y - 0.00225822 x^6 y - 0.0000603434 x^7 y - 4.56023 x 10^-7 x^8 y - 6.59549 x 10^-8 x^9 y - 1.52791 x 10^-8 x^10 y - 1.24413 x 10^-11 x^11 y - 3.38695 x 10^-14 x^12 y - 9.93088 x 10^-16 x^13 y - 3.41133 x 10^-18 x^14 y - 64.5215 y^2 - 4.06305 x y^2 - 5.34869 x^2 y^2 - 1.08349 x^3 y^2 - 0.0679811 x^4 y^2 - 0.000576557 x^5 y^2 - 0.000267467 x^6 y^2 + 0.0000127376 x^7 y^2 - 2.14261 x 10^-7 x^8 y^2 - 1.31198 x 10^-8 x^9 y^2 - 9.70906 x 10^-10 x^10 y^2 - 1.23071 x 10^-12 x^11 y^2 - 4.80023 x 10^-15 x^12 y^2 - 2.83256 x 10^-18 x^13 y^2 - 12.0338 x^4 y^3 - 3.95026 x^5 y^3 - 0.00115439 x^6 y^3 - 0.0994362 x^7 y^3 - 0.0112911 x^8 y^3 - 0.000422404 x^9 y^3 - 4.25622 x 10^-6 x^10 y^3 - 7.91459 x 10^-7 x^11 y^3 - 2.29186 x 10^-8 x^12 y^3 - 2.28091 x 10^-10 x^13 y^3 - 7.45129 x 10^-13 x^14 y^3 - 2.58203 x 10^-14 x^15 y^3 - 1.03477 x 10^-16 x^16 y^3 - 0.891449 y^4 - 0.541743 x y^4 - 0.0679811 x^2 y^4 - 0.000960928 x^3 y^4 - 0.000668668 x^4 y^4 - 0.0000445815 x^5 y^4 - 9.98883 x 10^-7 x^6 y^4 - 7.87186 x 10^-9 x^7 y^4 - 7.2818 x 10^-10 x^8 y^4 - 1.12815 x 10^-11 x^9 y^4 - 5.28026 x 10^-14 x^10 y^4 - 3.68233 x 10^-17 x^11 y^4 - 0.000115439 y^5 - 0.0298309 x y^5 - 0.00677466 x^2 y^5 - 0.000422404 x^3 y^5 - 6.38433 x 10^-6 x^4 y^5 - 1.66206 x 10^-6 x^5 y^5 - 6.41722 x 10^-8 x^6 y^5 - 8.21129 x 10^-10 x^7 y^5 - 3.35308 x 10^-12 x^8 y^5 - 1.42012 x 10^-13 x^9 y^5 - 6.82949 x 10^-16 x^10 y^5 - 0.00453207 y^6 - 0.000192186 x y^6 - 0.000267467 x^2 y^6 - 0.000029721 x^3 y^6 - 9.98883 x 10^-7 x^4 y^6 - 1.10206 x 10^-8 x^5 y^6 - 1.35927 x 10^-9 x^6 y^6 - 2.70757 x 10^-11 x^7 y^6 - 1.58408 x 10^-13 x^8 y^6 - 1.35019 x 10^-14 x^9 y^6 - 0.000322603 y^7 - 0.0000603434 x y^7 - 1.82409 x 10^-6 x^2 y^7 - 7.91459 x 10^-7 x^3 y^7 - 4.58373 x 10^-8 x^4 y^7 - 8.21129 x 10^-10 x^5 y^7 - 4.47078 x 10^-12 x^6 y^7 - 2.43449 x 10^-13 x^7 y^7 - 1.46346 x 10^-15 x^8 y^7 - 9.55239 x 10^-6 y^8 - 3.18439 x 10^-6 x^2 y^8 - 2.14261 x 10^-7 x^3 y^8 - 3.93593 x 10^-9 x^4 y^8 - 7.2818 x 10^-10 x^5 y^8 - 2.03068 x 10^-11 x^6 y^8 - 1.58408 x 10^-13 x^7 y^8 - 1.73596 x 10^-16 x^8 y^8 - 5.06693 x 10^-8 y^9 - 6.59549 x 10^-9 x y^9 - 7.63954 x 10^-8 x^2 y^9 - 2.28091 x 10^-10 x^3 y^9 - 1.86282 x 10^-12 x^4 y^9 - 1.42012 x 10^-13 x^5 y^9 - 1.13825 x 10^-15 x^6 y^9 - 4.76135 x 10^-8 y^10 - 2.62395 x 10^-10 x y^10 - 9.70906 x 10^-11 x^2 y^10 - 4.51261 x 10^-12 x^3 y^10 - 5.28026 x 10^-14 x^4 y^10 - 8.10113 x 10^-13 x^5 y^10 - 1.38901 x 10^-16 x^6 y^10 - 1.24413 x 10^-11 x^2 y^11 - 2.03217 x 10^-13 x^3 y^11 - 2.58203 x 10^-14 x^4 y^11 - 3.10431 x 10^-16 x^5 y^11 - 1.47107 x 10^-12 x^6 y^11 - 2.05119 x 10^-13 x^7 y^11 - 4.80023 x 10^-15 x^8 y^11 - 1.22744 x 10^-17 x^9 y^11 - 2.60535 x 10^-15 x^10 y^11 - 9.93088 x 10^-16 x^11 y^11 - 2.38793 x 10^-17 x^12 y^11 - 5.27498 x 10^-17 y^14 - 4.04652 x 10^-18 x y^14 - 2.27422 x 10^-19 x^2 y^14]

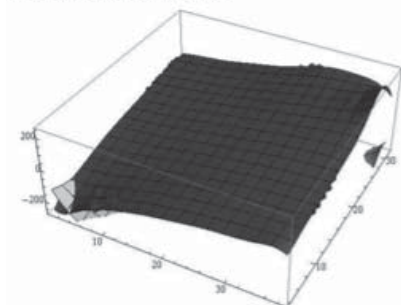
```

Outputs the potential function, stream function, and evaluation points in a three-dimensional plot.

```

Graph1 = Plot3D[ $\phi[x, y]$ , {x, Min[G[[All, 1]]] - .1, Max[G[[All, 1]]] + .1}, {y, Min[G[[All, 2]]] - .1, Max[G[[All, 2]]] + .1}, PlotStyle -> Hue[0]];
Graph2 = Plot3D[ $\psi[x, y]$ , {x, Min[G[[All, 1]]] - .1, Max[G[[All, 1]]] + .1}, {y, Min[G[[All, 2]]] - .1, Max[G[[All, 2]]] + .1}, PlotStyle -> Hue[0.15]];
Graph3 = Graphics3D[{PointSize[Large], Hue[0.7], Point[ImportedData]}];
Show[Graph1, Graph2, Graph3]

```



Outputs the potential function, stream function, and evaluation points in a contour plot.

```

Basin = {2, 2};
ExtractWells = {{28, 35}, {38, 26}};
Graph4 = ContourPlot[ $\phi[x, y]$ , {x, 0, 40}, {y, 0, 38}, ContourStyle -> None, DisplayFunction -> Identity, AspectRatio -> Automatic, ColorFunction -> "GrayTones"];
Graph5 = ContourPlot[ $\psi[x, y]$ , {x, 0, 40}, {y, 0, 38}, ContourShading -> False, DisplayFunction -> Identity, AspectRatio -> Automatic];
Graph6 = Graphics[{PointSize[.01], White, Point[G]}];
Graph7 = Graphics[{PointSize[.04], Black, Point[ExtractWells]}];
Graph8 = Graphics[{PointSize[.06], Black, Point[Basin]}];
Graph9 = Graphics[{PointSize[.02], Black, Point[Wells]}];
Show[Graph4, Graph5, Graph6, Graph7, Graph8, Graph9]

```

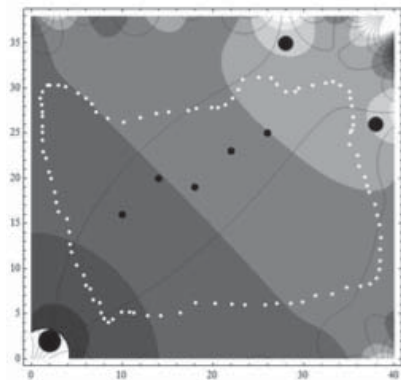


Figure A4. Example Mathematica output of first adjusted solution. Shown are the potential and streamline functions and the 3-D plot and contour plot of the approximate solution.