Academic Science Journal

Solving Inverse Cauchy Problems of the Helmholtz Equation Using Conjugate Gradient Based-Method: BICG, BICGSTAB And PCG

Ibtihal Thabit Jameel, Athraa Falih Hasan, Duaa Jasim and Fatima M. Aboud

Department of Mathematics – College of Science – University of Diyala, Iraq

scimathms2101@uodiyala.edu.iq

Received: 5 August 2022 **Accepted: 12 October 2022**

DOI: https://doi.org/10.24237/ASJ.01.04.652D

Abstract

An inverse Cauchy problem on an under-specified portion of the boundary of a two-dimensional domain occupied by a material satisfying Helmholtz-type equations from additional Cauchy data on the remaining accessible portion of the boundary has been considered. The solution is approached using a polynomial expansion to obtain a linear system; this linear system has been solved by conjugate gradient based-method. The efficiency of the proposed method has been confirmed by applying it for different examples and by comparing the obtained results by using three different numerical methods.

Keywords: Inverse Cauchy problems, Helmholtz equation, polynomial expansion, conjugate gradient based-method.

Academic Science Journal

حل مسائل كوشي العكسية لمعادلة هيلمهولتز باستخدام طرائق تعتمد التدرج المترافق: BICG, BICGSTAB and PCG

ابتهال ثابت جميل و عذراء فالح حسن و دعاء جاسم و فاطمة محمد عبود

قسم الرياضيات، كلية العلوم، جامعة ديالى، العراق

الخلاصة

تم دراسة ايجاد الحل لمسألة كوشي معرفة على حدود قليلة البيانات او معدومة البيانات لمجال ثنائي الأبعاد تشغله مادة تحقق معادلات من نوع هيلمهولتز وذلك بالاستفادة من بيانات كوشي الإضافية على الجزء المتبقي من الحدود. تم تقريب الحل بمفكوك متعدد الحدود للحصول على نظام خطي تم حل هذا النظام الخطي باستخدام طرائق تعتمد التدرج المقترن. تم تأكيد كفاءة الطريقة المقترحة باختبارها لامثلة مختلفة وبمقارنة النتائج لثلاث طرق عددية مختلفة.

الكلمات المفتاحية: مسائل كوشي المعكوسة، معادلة هيلمهولتز، مفكوك كثير الحدود، طرائق تعتمد التدرج المقترن.

Introduction

Direct boundary value problems for Helmholtz equations have been widely studied in the last century. It is a kind of elliptic Partial Differential Equation (PDE) with time-independent solutions of the wave equation. Helmholtz-type equations can be appear naturally in physical applications related to wave propagation, vibration phenomena, and heat transfer [8], in the acoustic cavity problem [13], the radiation wave [15], and the heat conduction in fins [26].

A problem is well-posed in the sense of Hadamard, in case the existence, uniqueness, and stability of the solution is A guaranty, [14], otherwise if one of these conditions is not satisfied by the solution then the problem is ill-posed, and hence an inverse problem is formulated to be able to solve the problem. The inverse problems are more difficult to be solved than the direct problems. It is well known that inverse problems are in general unstable, [14], i.e. a small error of measurement in the input data may imply to an important error in the solution. In recent years, inverse problems have been extensively treated in several branches of science, [27]. The Cauchy problem is one of the examples of inverse problem [7], [12], [16], [21], [28], [29] and

[34]. For this kind of problems, the boundary conditions (Dirichlet, Neumann) are given just on some part of the boundary, and on the rest part there is no information, and this part is known as the un-accessible part of the boundary. Unfortunately, the ill-posed ness of Cauchy problem for the Helmholtz equation is known to be very severe in the sense of Hadamard [14].

For all these reasons, the choice of a suitable algorithm must be taken in consideration to reduce the ill-posedness of this kind of problem. In the last two decades, several methods have been proposed for solving the Cauchy problem for the Helmholtz equation. In the following we recall some of these methods, the truncation method [37], the conjugate gradient method [33], the meshless generalized finite difference method [17], the Landweber method [37].

In fact, the approximation of numerical solutions of direct Helmholtz equation depends on k and this physical parameter has an important effect on the quality of approximation, to have an idea about the dependence of the quality of the numerical solution on the wave number k see [19-20]. Some authors have proposed some methods to solve Cauchy Helmholtz equation for the big value of k [9-11]. A new efficient alternating algorithm based on the relaxation of alternating algorithms has proposed by Jourhmane&Nachaoui [23]. An effective relaxed alternating procedure proved the convergence for all values of wave number k in the case of Helmholtz equation and accelerate the convergence in the case of modified Helmholtz equation by Berdawood et. al. [4-6], the authors proved that for any value wave number k we find an interval of relaxation parameter in which the convergence is assured.

This paper has as aim the exploration of a method to solve a Cauchy problem for a Helmholtz-type equation in a bounded domain enclosed by a smooth boundary by using an approximation of the solution by a polynomial. This method was proposed in Rasheed et. al [35] to solve an inverse Cauchy problem. This paper consists of 5 sections. In Section 2, we recall Inverse Cauchy problems for the Helmholtz equation. In section 3, we present the approximation of the solution by a polynomial expansion. The linear system and numerical method are given in the fourth Section. To conclude, we present numerical results and discussion in Section 5.

Inverse Cauchy problems for the Helmholtz equation

In our study, we consider the domain

$$\Omega = \{(x,y) \in \mathbb{R}^2 \,/\, x^2 + y^2 < r^2\}$$

with its boundary

$$\Gamma_1 = \{(r,\theta) : r = \rho(\theta), 0 \leq \theta \leq \beta\pi\}$$

and

$$\Gamma_2 = \{(r,\theta): r = \rho(\theta), \beta\pi \leq \theta < 2\pi, 0 < \beta < 2\}$$

The inverse Cauchy problem for the Helmholtz equation is given by:

Find the unknown function $T(x,y)$ such that, given $T(x,y)$ and $\partial_n T(x,y)$, on Γ_1

$$(\Delta + k^2)\mathrm{T} = 0 \; (x,y) \in \Omega \quad \text{(1)}$$

$$\mathrm{T}(\rho,\theta) = \tilde{\mathrm{T}}(\theta) \; 0 \leq \theta \leq \beta\pi \quad \text{(2)}$$

$$\partial_n T(\rho,\theta) \equiv \Phi(\rho,\theta) = \tilde{\Phi}(\theta) \; 0 \leq \theta \leq \beta\pi \quad \text{(3)}$$

We start by noting, that the normal derivative equals $\frac{\partial \mathrm{T}}{\partial \mathrm{n}} = \nabla \mathrm{T}.\vec{\mathrm{n}}$ which can be expressed in the following form:

$$\partial_{\mathrm{n}}\mathrm{T}(\rho,\theta) = \eta(\theta)\left[\frac{\partial \mathrm{T}(\rho,\theta)}{\partial \rho} - \frac{\mathrm{p}'}{\rho^2}\frac{\partial \mathrm{T}(\rho,\theta)}{\partial \theta}\right] \quad \text{(4)}$$

$$\eta(\theta) = \frac{\rho(\theta)}{\sqrt{\rho^2(\theta) + [\rho'(\theta)]^2}} \quad \text{(5)}$$

Also, $\partial_n T(x,y)$ can be expressed in terms of $\partial x T$ and $\partial y T$ by

$$\partial_{\mathrm{n}}\mathrm{T} = \eta(\theta)\left[\cos(\theta) - \frac{\rho'}{\rho^2}\sin(\theta)\right]\partial_x \mathrm{T} + \eta(\theta)\left[\sin(\theta) - \frac{\rho'}{\rho^2}\cos(\theta)\right]\partial_y \mathrm{T} \quad \text{(6)}$$

Polynomial expansion

The solution $\mathrm{T}(x,y)$ is expanded by

$$\mathrm{T}(x,y) = \sum_{i=1}^{m} \sum_{\mathrm{j}=1}^{i} c_{ij}\, x^{i-\mathrm{j}} y^{\mathrm{j}-1} \quad \text{(7)}$$

We note that the number of the coefficients c_{ij} is equal to $n = \frac{m(m+1)}{2}$ and these coefficients must be determined. The maximum order of the polynomial given in (7) is $m-1$.

Using equation (7), we find $\partial_x T$, $\partial_y T$ and ΔT

$$\partial_x T(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{i} c_{ij}(i-j)x^{i-j-1}y^{j-1} \quad \text{.......... (8)}$$

$$\partial_y T(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{i} c_{ij}(j-1)x^{i-j}y^{j-2} \quad \text{.......... (9)}$$

so

$$\Delta T(x,y) + K^2T(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{i} c_{ij}[(i-j)(i-j-1)x^{i-j-2}y^{j-1} + (j-1)(j-2)x^{i-j}y^{j-3} + K^2(x^{i-j}y^{j-1})] \quad \text{.......... (10)}$$

The coefficients c_{ij} in equation (7) can be formulated as a vector of dimension n, say c where c_k are the components of c, by the formula $\frac{i(i-1)}{2} + j$.

$T(x,y)$ has the following form as an inner product of a vector of variables a with a vector of coefficients c,

$$T(x,y) = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 & x^3 & x^2y & xy^2 & y^3 & \dots \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{bmatrix}$$

$$= \mathrm{a}^{\mathrm{T}}\mathrm{c} \quad \text{.......... (11)}$$

Noting that the dimensions of the vectors $\mathbf{a}^{\mathrm{T}}, \mathbf{c}$ are taken to be able to do the product.

Including (8) and (9) into (6) gives us an expression of $\partial_n T$. (Normal derivative)

$$e_\ell = \eta(\theta)\left[(i-j)x^{i-j-1}y^{j-1}\left(cos(\theta)-\frac{\rho'}{\rho^2}sin(\theta)\right)+(j-1)x^{i-j}y^{j-2}\left(sin(\theta)-\frac{\rho'}{\rho^2}cos(\theta)\right)\right] \quad (12)$$

For the indices of the coefficients and the variables of $T(x,y)$.

Now, for some internal points of the domain Ω, the Laplacian $\Delta T(x,y)$ can be given as an inner product of a vector d with component d_ℓ of the following form:

$$d_\ell = (i-j)(i-j-1)x^{i-j-2}y^{j-1} + (j-1)(j-2)x^{i-j}y^{j-3} + K^2\left(x^{i-j}y^{j-1}\right) \quad (13)$$

with c, for l= 1, . . . , n_{1a}.

We begin by choosing n_{1a} points $(x_\ell, y_\ell), \ell = 1, \dots, n_{1a}$ on Γ_1 for which the boundary conditions $(2-3)$ are satisfied. Also, for some points $(x_j, y_j), j = 1, \cdots, n_2$ in the domain Ω to satisfy the differential equation given in (1). The obtained equations form a system of linear by solving it, the n coefficients c_{ij} are obtained. Now, this klinear system is as follows

$$Ac = b \quad (14)$$

With a vector b of order $2n_{1a} + n_2$ and A a $2n_{1a} + n_2 \times \frac{m(m+1)}{2}$ matrix given respectively by

$$b = \begin{pmatrix} \tilde{T}(\theta_1) \\ \vdots \\ \tilde{T}(\theta_{n1a}) \\ \tilde{\Phi}(\theta_1) \\ \vdots \\ \tilde{\Phi}(\theta_{n1a}) \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad A = \begin{pmatrix} a_1^T \\ \vdots \\ a_{n1a}^T \\ e_1^T \\ \vdots \\ e_{n1a}^T \\ d_1^T \\ \vdots \\ d_{n2}^T \end{pmatrix} \quad (15)$$

Thus, to solve our inverse Cauchy problem is equivalent to solve the linear system given in (14).

Numerical Methods to Solve the linear system

1- Numerical methods (BiCG), ((BiCGSTAB) and (PCG)

In the following, we recall some of the conjugate gradient-based methods (see [3]), which are well-known iterative method for solving linear system Ax = b,

- The bi-conjugate gradients (BiCG) is an algorithm, that was developed for generalizing the conjugate gradient (CG) method for solving the non-symmetric linear systems.
- The bi-conjugate gradients stabilized (BiCGSTAB) is an algorithm that was developed for improving the BiCG algorithm
- The preconditioned conjugate gradients method (PCG) is an algorithm that was developed to exploit the structure of symmetric positive definite matrices.

The PCG is used to be able to accelerate the iterative method, in fact, for this method we introduce the so-called pre-conditioner , say P, to the linear system Ax = b. This is used in case that the matrix A is ill-conditioned (with a big condition number of κ(A)), so we choose some pre-conditioner P, which implies that the condition number of PA or AP will satisfy cond(PA) << cond(A) or cond(AP) << cond(A). Therefore ,we solve PAx = Pb or APx = Pb by iterative method instead of Ax = b. The Preconditioned Conjugate Gradient (PCG) method is one of these kinds methods.

2- Stopping criterion and initial guess

The simplest and most common stopping criteria are stated by

$$\|r_i\| < Tol \qquad (16)$$

$$\frac{\|r_i\|}{\|b\|} < Tol \qquad (17)$$

where Tol is a user-provided error tolerance. For the initial guess, is chosen to be the zero vector at the initial iteration.

Numerical results and discussion

In this section, we present the numerical results for the inverse Cauchy problem for Helmholtz equation stated in (1-3)

Example 1: We solve the problem (1-3) at the domain bounded by $\rho(\theta) = 1$ and Γ_1 is defined taking $\beta = 1$ with an exact solution is $T(x, y) = x^4 + y^4$. To discretize the boundary, we take $n_{1a} = 100$, $n_{1b} = 100$, and the number of internal domain points is $n_2 = 2000$. The linear system is solved by varying $m = 1, \dots, 12$ and by using the algorithms, $BiCG$, $BiCGSTAB$, PCG with $Tol = 10^{-12}$. In the following the numerical results for different cases of the physical parameter k.

Table 1: $k = \sqrt{15}$

M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	3	0.529096876	1.5	0.529096876	3	0.529096876
3	7	0.358686915	5	0.358686915	7	0.358686915
4	11	0.360590266	11	0.360590266	11	0.360590266
5	**22**	9.69725200E-11	**27**	1.60840098E-13	**23**	3.59471523E-14
6	41	1.22004838E-09	89.5	2.20783540E-12	42	1.44287085E-12
7	88	3.05841475E-09	351.5	1.15074272E-10	85	1.75652567E-11
8	164	2.51138316E-09	940.5	8.70072295E-12	159	8.94092322E-12
9	340	2.39243441E-08	3037.5	6.59980432E-10	304	8.68077326E-11
10	832	3.13441769E-07	11921.5	7.28863188E-10	548	1.39322307E-10
11	1202	0.006475063	6135	3.99616891E-09	976	4.19081426E-09
12	4599	0.006976981	21782	1.09717714E-08	2207	4.64071953E-08

In table (1), from m = 5, a good approximation is obtained for all the methods with roughly a same number of iterations, which is coherent with the given data. In fact, the exact solution is a polynomial function of degree 4 and it is logic to obtain the best accuracy by approximating a polynomial of degree 4 by a polynomial of degree 4. We also observe that the PCG method is more accurate than the $BiCG$, $BiCGSTAB$ and that for a big value of m, the PCG is faster for the same accuracy.

Table 2: $k = \sqrt{25.5}$

M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	2	0.653007484	1.5	0.653007484	2	0.653007484
3	7	0.256190455	4.5	0.256190455	7	0.256190455
4	11	0.255888263	11	0.255888263	11	0.255888263
5	21	1.21995356E-12	30.5	2.43228119E-14	21	8.24990586E-14
6	38	1.88529541E-10	54.5	1.31041973E-10	36	1.42904524E-10
7	82	3.14522081E-10	288.5	9.01315759E-12	76	2.90224296E-12
8	165	2.99339594E-08	950.5	9.73327222E-12	162	6.53785676E-12
9	320	6.27882455E-08	7627.5	1.03926826E-07	334	4.09496706E-11
10	734	0.000199628	8020	1.67332700E-09	662	9.10602603E-09
11	1809	0.004934644	3801	1.27763794E-09	1336	1.01629112E-09
12	4719	0.003778616	8755	1.66952969E-08	2578	2.30932110E-08

For table 2, we note that the same remark as table 1.

Table 3: $k = \sqrt{52}$

M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	2	0.74597044	1.5	0.74597044	2	0.74597044
3	7	0.238066269	4	0.238066269	7	0.238066269
4	10	0.23825077	8	0.23825077	10	0.23825077
5	20	1.01523784E-11	29	2.30310135E-14	20	1.26432210E-14
6	34	2.38414631E-11	49	6.58263536E-12	34	6.55124581E-12
7	61	5.43136554E-10	248.5	8.28453558E-10	68	1.29824124E-12
8	107	9.24083016E-10	465.5	1.28247892E-10	102	4.28143446E-11
9	259	5.08284184E-09	3894.5	1.56914567E-09	239	1.55116778E-09
10	709	1.52873300E-06	31829	5.01934527E-09	541	6.13902432E-09
11	3400	0.000633603	3201	3.49565595E-08	1558	2.17668824E-08
12	6531	0.000812414	1641	5.78665717E-07	2962	6.92160375E-07

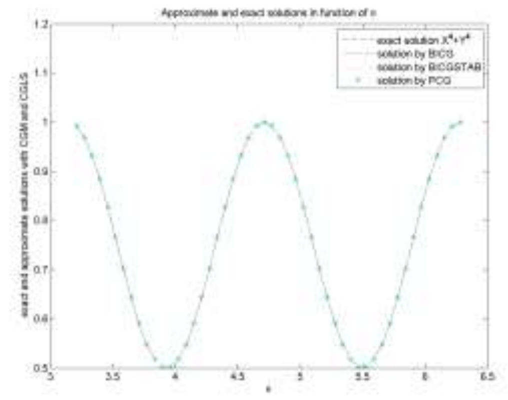
For table 3, we note the same remark as table 1,2, but with more accuracy.

Table 4: $k = 100$

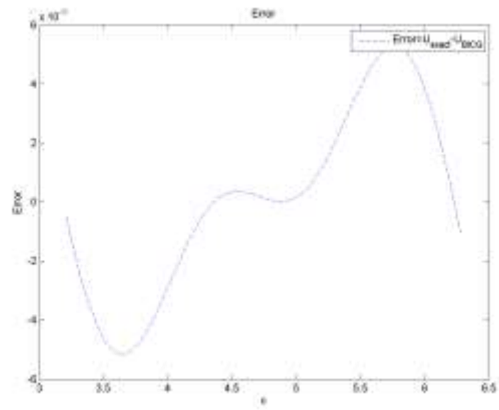
M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	2	0.789484743	1.5	0.789484743	2	0.789484743
3	4	0.281152413	3	0.281152413	4	0.281152413
4	9	0.281355523	8	0.281355523	9	0.281355523
5	21	3.96850485E-12	24.5	3.97236822E-15	20	4.37785145E-15
6	31	3.71444631E-11	47.5	3.61250702E-11	31	2.52186990E-11
7	71	7.51976440E-11	232	1.79198865E-12	65	6.95681632E-14
8	100	6.82114551E-11	384.5	3.24259143E-11	99	3.34686584E-11
9	224	1.44371905E-03	5530	5.86962665E-10	209	5.92917781E-10
10	513	8.56975337E-08	3832	8.29657989E-11	406	5.09891896E-09
11	1145	2.91160113E-04	1401	4.05298672E-08	947	4.05118186E-08
12	3387	3.90553496E-05	8666	1.27338810E-07	1758	1.62936379E-07

For table 4, we note the same remark as tables (1-3), with more accuracy. In fact, we note that when k be bigger the accuracy be better.

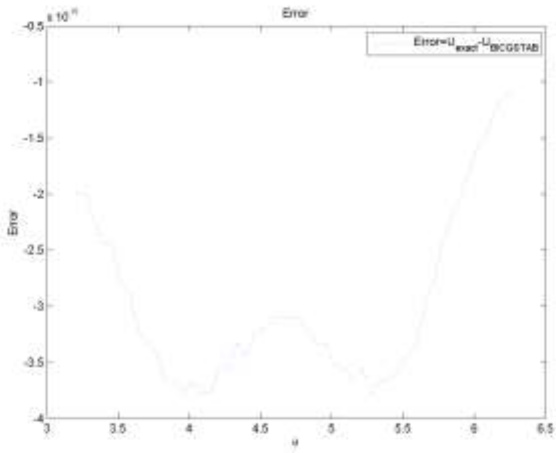
Exact solution $\boldsymbol{T_{ex} = x^4 + y^4}$ with the approximate solutions by $\boldsymbol{CGM}$ and $\boldsymbol{CGLS}$ with $\boldsymbol{\beta = 1}$, on $\boldsymbol{\Gamma_2}$



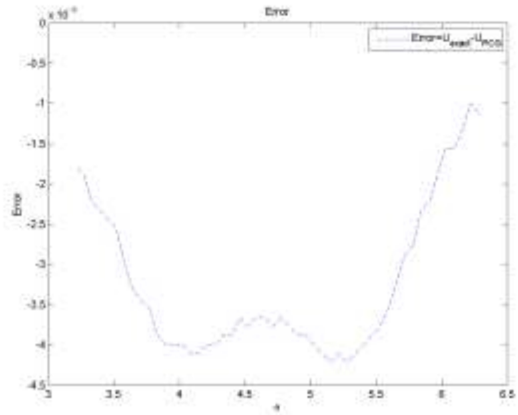
A: The exact and the approximate solution calculated by BiCG, BiCGSTAB, PCG



B: The error between the exact solution and the approximate one calculated by BiCG



C: The error between the exact and the approximate one calculated by BiCGSTAB



D: The error between the exact and the approximate one calculated by PCG

Figure 1: Results for the case of exact solution $T(x, y) = x^4 + y^4$, $\beta = 1$, with $n_{1a} = 100$, $n_{1b} = 100$ $n_2 = 2000$. Approximate solutions calculated by the algorithms, $BiCG$, $BiCGSTAB$, PCG with $Tol = 10^{-12}$.

In the following, we study a non-polynomial example

Example 2: We solve the problem (1-3) at the domain bounded by $\rho(\theta) = 1$ and Γ_1 is defined taking $\beta = 1$ with an exact solution is $T(x, y) = \exp(x)\cos(y)$. To discretize the boundary, we take $n_{1a} = 200$, $n_{1b} = 200$, and the number of internal domain points is $n_2 = 4000$. The linear system is solved by varying $m = 1, \dots, 10$ and by using te algorithms, $BiCG$, $BiCGSTAB$, PCG with $Tol = 10^{-12}$.

Table 5: $k = \sqrt{15}$

M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	3	0.289689244	2	0.289689244	3	0.289689244
3	7	0.094099455	6	0.094099455	7	0.094099455
4	11	0.02333162	15.5	0.02333162	11	0.02333162
5	23	0.004676945	32.5	0.004676945	22	0.004676945
6	43	0.000794595	92	0.000794595	43	0.000794595
7	90	0.000123725	273.5	0.000123724	86	0.000123724
8	162	1.89417243E-05	865.5	1.89296535E-05	159	1.89296540E-05
9	326	6.67042416E-06	2258.5	6.83792522E-06	287	6.83751416E-06
10	769	2.31431418E-06	16044.5	1.83586827E-06	523	1.83517913E-06
11	1458	2.39220632E-04	14434	3.43871041E-07	1016	3.43329433E-07
12	4123	1.26862202E-03	8548	2.20533229E-05	1837	2.22190094E-05

In table (5), the best approximation is obtained with BICG for $m = 10$ and with BiCGSTAB and PCG for $m = 10$. The PCG has less number of iteration for the same accuracy.

Table 6: $k = \sqrt{25.5}$

M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	3	0.289623396	2	0.289623396	3	0.289623396
3	7	0.093962295	6	0.093962295	7	0.093962295
4	11	0.023221393	11	0.023221393	11	0.023221393
5	22	0.004626346	27.5	0.004626346	22	0.004626346
6	41	0.000772314	77	0.000772314	41	0.000772314
7	81	0.000111478	208.5	0.000111478	80	0.000111478
8	169	1.45127820E-05	1052.5	1.45123047E-05	157	1.45121836E-05

9	368	1.81533481E-06	2410.5	1.83691481E-06	328	1.83691490E-06
10	829	6.18819171E-06	9554	4.26124193E-07	663	4.22547519E-07
11	1691	3.65583380E-04	11910	1.02309765E-07	1266	9.98263388E-08
12	3670	5.90709242E-04	27443	7.45231322E-06	2261	7.56741618E-06

In table 6, we note that the BiCG attends the best accuracy with m=9, but the BiCGSTAB and PCG attend it for m=11. The number of iterations of PCG is less than that for the other two algorithms for the same accuracy.

Table 7: k= $\sqrt{52}$

M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	3	0.289596921	2	0.289596921	3	0.289596921
3	7	0.093906953	6	0.093906953	7	0.093906953
4	11	0.023175159	10	0.023175159	11	0.023175159
5	23	0.004604848	26.5	0.004604848	22	0.004604848
6	39	0.000765177	51.5	0.000765177	38	0.000765177
7	73	0.00010925	191.5	0.00010925	70	0.00010925
8	128	1.36839001E-05	550.5	1.36835714E-05	122	1.36835762E-05
9	337	1.53288974E-06	2066	1.52939937E-06	278	1.52939405E-06
10	631	3.47482551E-07	9724	1.69525232E-07	536	1.70979992E-07
11	2108	1.63486693E-05	14375	3.33366025E-07	1343	3.38084033E-07
12	13893	5.89988875E-05	7943	3.98346800E-07	2656	1.03657946E-06

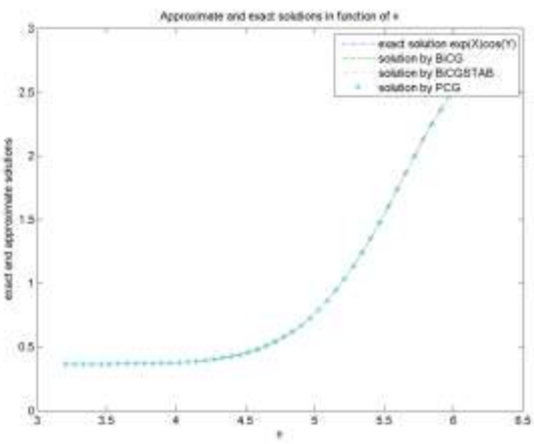
In table 7, the best accuracy is obtained for m=10 with less iteration for the PCG.

Table 8: k= $\sqrt{100}$

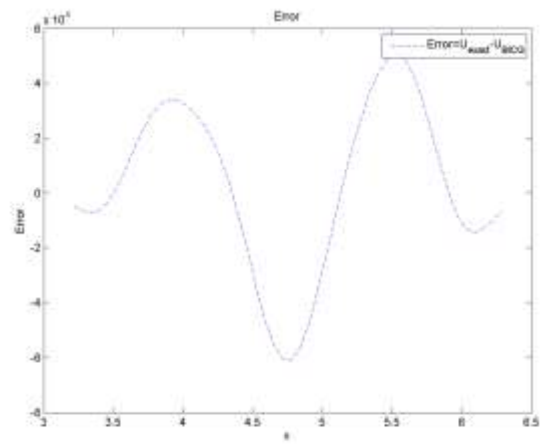
M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	3	0.289596921	2	0.289596921	3	0.289596921
3	7	0.093906953	6	0.093906953	7	0.093906953
4	11	0.023175159	10	0.023175159	11	0.023175159
5	23	0.004604848	26.5	0.004604848	22	0.004604848
6	39	0.000765177	51.5	0.000765177	38	0.000765177
7	73	0.00010925	191.5	0.00010925	70	0.00010925
8	128	1.36839001E-05	550.5	1.36835714E-05	122	1.36835762E-05
9	337	1.53288974E-06	2066	1.52939937E-06	278	1.52939405E-06
10	631	3.47482551E-07	9724	1.69525232E-07	536	1.70979992E-07
11	2108	1.63486693E-05	14375	3.33366025E-07	1343	3.38084033E-07
12	13893	5.89988875E-05	7943	3.98346800E-07	2656	1.03657946E-06

For table 8, the same remark as the previous case is obtained.

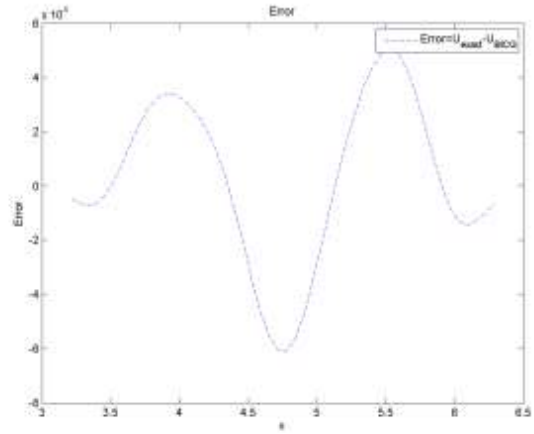
Exact solution $\boldsymbol{T_{ex}} = \mathbf{exp}(\boldsymbol{x})\,\mathbf{cos}(\boldsymbol{y})$ with the approximate solutions by $\boldsymbol{CGM}$ and $\boldsymbol{CGLS}$ with $\boldsymbol{\beta} = \mathbf{1}$, on $\boldsymbol{\Gamma_2}$



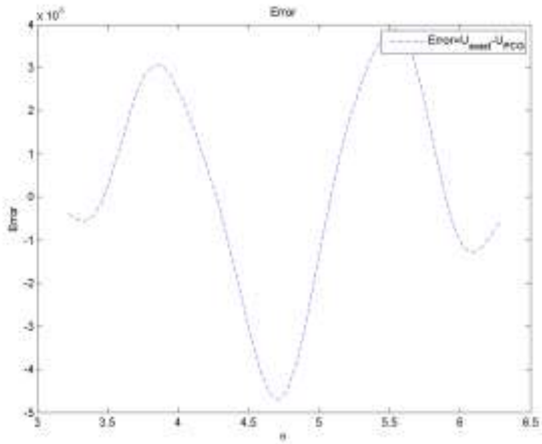
(A): Exact and approximate solution calculated by BiCG, BiCGSTAB, PCG



(B): The error between the exact and the approximate one calculated by BiCG



(C): The error between the exact and the approximate one calculated by BiCGSTAB



(D): The error between the exact and the approximate one calculated by PCG

Figure2: Results for the case of exact solution $T(x, y) = \exp(x)\cos(y)$, $\beta = 1$, with $n_{1a} = 100$, $n_{1b} = 100$ $n_2 = 2000$. Approximate solutions calculated by the algorithms, $BiCG$, $BiCGSTAB$, PCG with $Tol = 10^{-12}$.

In the following, we study an example with a domain of less accessible part.

Example3: We solve the problem (1-3) at the domain bounded by $\rho(\theta) = 1$ and Γ_1 is defined taking $\beta = 0.5$ with an exact solution is $T(x, y) = \exp(x)\cos(y)$. To discretize the boundary, we take $n_{1a} = 200$, $n_{1b} = 200$, and the number of internal domain points is $n_2 = 4000$. The linear system is solved by varying $m = 1, \ldots, 12$ and by using algorithms, $BiCG$, $BiCGSTAB$, PCG with $Tol = 10^{-12}$.

Table 9: $k = \sqrt{15}$

M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	3	0.316725761	2.5	0.316725761	3	0.316725761
3	7	0.107537005	6.5	0.107537005	7	0.107537005
4	12	0.026431743	13.5	0.026431743	12	0.026431743
5	22	0.005392613	28.5	0.005392613	23	0.005392613
6	47	0.000899636	75.5	0.000899637	46	0.000899637
7	96	0.000141111	263.5	0.000141111	91	0.000141111
8	199	3.34802581E-05	697.5	3.34894797E-05	178	3.34894942E-05
9	559	1.13387208E-05	5095.5	1.11085087E-05	402	1.11085281E-05
10	1318	2.05547803E-04	9268	2.96549883E-06	966	2.99460030E-06
11	2937	3.96103444E-03	21705	8.95364710E-05	2081	8.95710383E-05
12	13913	3.89378094E-03	14451	1.91838196E-03	3276	1.92550776E-03

For $m = 9$, the BiCG attend its best accuracy, but the BiCGSTAB and PCG attend a better accuracy for $m = 10$, and the PCG still with less number of iteration for the same accuracy.

Table 10: $k = \sqrt{25.5}$

M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	3	0.31667775	2.5	0.31667775	3	0.31667775
3	7	0.107422891	6.5	0.107422891	7	0.107422891
4	12	0.026363773	14.5	0.026363773	12	0.026363773
5	23	0.005360712	31.5	0.005360712	23	0.005360712
6	44	0.000886733	71	0.000886733	44	0.000886733
7	90	0.000129136	212.5	0.000129136	87	0.000129136
8	223	1.62730298E-05	748.5	1.62733949E-05	182	1.62734186E-05
9	433	2.36713062E-06	5590.5	2.22084865E-06	379	2.22096166E-06
10	1631	4.62038847E-03	21197	8.44033337E-07	937	8.46672620E-07
11	3216	6.58248110E-04	6784	2.11467237E-06	2001	2.17787588E-06
12	14234	8.22605957E-04	15893	8.29292564E-06	3397	1.30346166E-04

For $m = 9$, the BiCG attend its best accuracy, but the BiCGSTAB and PCG attend a better accuracy for $m = 10$, and for the PCG still with less number of iteration for the same accuracy.

Tabel 11: $k = \sqrt{52}$

M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	3	0.316658482	2.5	0.316658482	3	0.316658482
3	7	0.107375467	6.5	0.107375467	7	0.107375467
4	11	0.026335962	13.5	0.026335962	12	0.026335962
5	21	0.005346983	28.5	0.005346983	21	0.005346983
6	41	0.000882249	65.5	0.000882249	40	0.000882249
7	82	0.000127681	174.5	0.000127681	81	0.000127681
8	171	1.58612162E-05	502.5	1.58613437E-05	164	1.58613435E-05
9	381	1.78888008E-06	1344.5	1.78816139E-06	307	1.78860034E-06
10	1075	2.60260577E-06	5921	1.78821258E-07	856	1.79767711E-07
11	4796	1.49784697E-04	24770	1.33075451E-07	2226	1.88835521E-08
12	4341	2.85413617E-04	19673	5.37527193E-06	2402	5.37996591E-06

We note that when $m = 9$, the BiCG attend its best accuracy, but the BiCGSTAB and PCG attend a better accuracy for $m = 11$, and for the PCG still faster for the same accuracy.

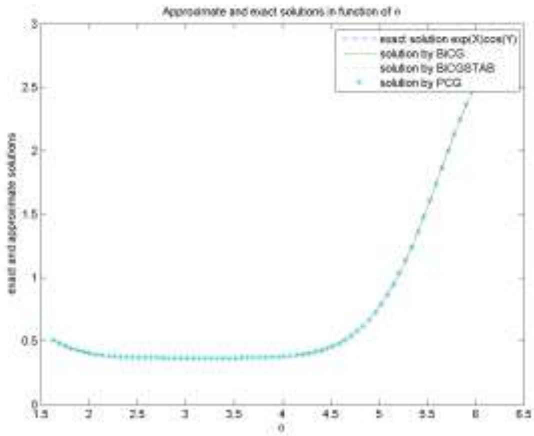
Tabel 12: $k = \sqrt{100}$

M	Iter of BICG	Error of BICG	Iter of BICGSTAB	Error BICGSTAB	Iter of PCG	Error of PCG
2	3	0.316654033	2.5	0.316654033	3	0.316654033
3	7	0.107364277	5.5	0.107364277	7	0.107364277
4	11	0.026329422	13.5	0.026329422	11	0.026329422
5	22	0.005343008	28.5	0.005343008	22	0.005343008
6	39	0.000881183	61	0.000881183	39	0.000881183
7	78	0.00012742	161.5	0.00012742	70	0.00012742
8	155	1.58016930E-05	480.5	1.58015524E-05	155	1.58016525E-05
9	341	1.77496189E-06	1638.5	1.77474857E-06	323	1.77483572E-06
10	817	1.84713125E-07	6928	1.76211436E-07	548	1.77158114E-07
11	3198	1.24987793E-06	23798	1.19310203E-07	1214	1.23236474E-07
12	3444	4.96142356E-06	8024	4.90458413E-07	1991	5.86571403E-07

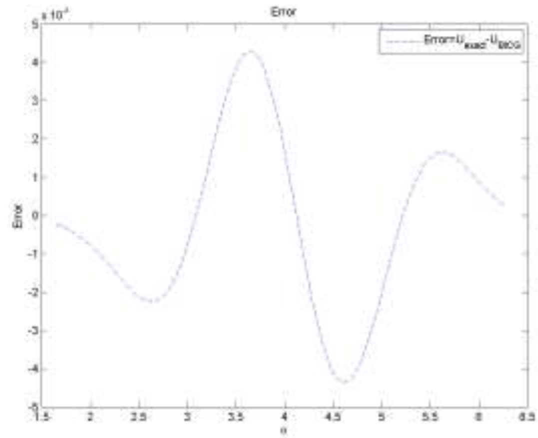
We note that when $m = 10$, the BiCG attend its best accuracy, but the BiCGSTAB and PCG attend a better accuracy for $m = 11$, and for the PCG still faster for the same accuracy.

Exact solution $\boldsymbol{T_{ex} = \exp(x)\cos(y)}$ with the approximate solutions by $\boldsymbol{CGM}$ and $\boldsymbol{CGLS}$ with $\boldsymbol{\beta = 0.5}$, on $\boldsymbol{\Gamma_2}$

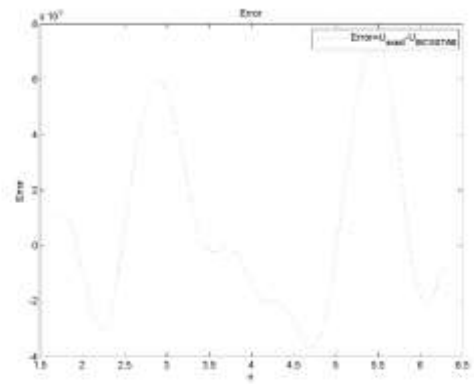
Academic Science Journal



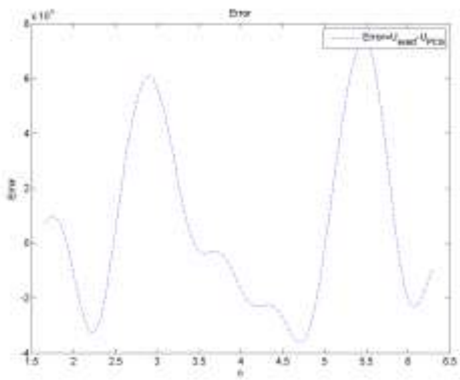
(A): Exact and approximate solution calculated by BiCG, BiCGSTAB, PCG



(B): The error between the exact and the approximate one calculated by BiCG



(C): The error between the exact and the approximate one calculated by BiCGSTAB



(D): The error between the exact and the approximate one calculated by PCG

Figure3: Results for the case of exact solution $T(x,y) = \exp(x)\cos(y)$, $\beta = 0.5$, with $n_{1a} = 100$, $n_{1b} = 100$ $n_2 = 2000$. Approximate solutions calculated by the algorithms, $BiCG$, $BiCGSTAB$, PCG with $Tol = 10^{-12}$.

Conclusion

An inverse Cauchy problem for Helmholtz equations in a regular domain is solved by benefiting from the over-specified boundary conditions, these extra data help to recover the missing data on another inaccessible part. Using polynomial expansion following the method of Rasheed et al., the problem is transformed to solve a direct problem of linear system. To illustrate the proposed method, we apply it for some examples with polynomial and non-polynomial exact solution, for all the cases we obtained approximations with good accuracy for all the proposed conjugate gradiant-based algoriths: BiCG, BiCGSTAB, PCG with the best accuracy and the faster one is the PCG.

Acknowledgement

The authors warmly thank Professor A. Nachaoui of Nantes University for having proposed and co-supervised this work.

References

1. F. Aboud, A. Nachaoui, M. Nachaoui, J. Phys.: Conf. Ser. , 1743(012003) (2021).
2. G. Alessandrini, L. Rondi, E. Rosset, S. Vessella, Inverse problems, 25(123004), (2009)
3. R. Barrett, M. Berry, T.F. Chan, SIAM, Philadelphia, (1994)
4. K. A. Berdawood, A. Nachaoui, R. Saeed, M. Nachaoui, F. Aboud, Advanced Mathematical Models & Applications, 5(1), 131-139(2020)
5. K. A. Berdawood, A. Nachaoui, M. Nachaoui, F. Aboud, Numer. Methods Partial Differential Eq., (2021)
6. K. A. Berdawood, A. Nachaoui, R. Saeed, M. Nachaoui, F. Aboud, Discrete & Continuous Dynamical Systems-S, (2021)
7. A. Bergam, A. Chakib, A. Nachaoui, M. Nachaoui, Appl. Math. Comput., 346, 865–878(2019)
8. DE. Beskos, ASME Appl Mech Rev 50, 986–1996(1997)
9. F. Berntsson, V. Kozlov, L. Mpinganzima, B. O. Turesson, Inverse Probl. Sci. Eng., 26, 1062–1078(2018)

10. F. Berntsson, V. A. Kozlov, L. Mpinganzima, B. O. Turesson, Comput. Math. Appl., 73(1), 163–172(2017)
11. F. Berntsson, V. A. Kozlov, L. Mpinganzima, B. O. Turesson, Inverse Probl. Sci. Eng., 22, 45–62(2014)
12. A. Chakib, A. Nachaoui, M. Nachaoui, H. Ouaissa, Inverse Problems, 35, 015008(2018)
13. JT. Chen, FC. Wong, J Sound Vibration, 217, 75–95(1998)
14. J. Hadamard, Lectures on Cauchy's problem in linear partial differential equations, (Dover Publications Dover Publications, New York, 1953)
15. I. Harari, PE. Barbone, M. Slavutin, R. Shalom, Int J Numer Meth Engng 41,1105–1131(1998)
16. E. Hernandez-Montero, A. Fraguela-Collar, J. Henry, Math. Model. Nat. Phenom., 14(2), (2019)
17. Q. Hua, Y. Gu, W. Qu, W. Chen, C. Zhang, Eng. Anal. Bound. Elem., 82, 162–171(2017)
18. C. H. Huang, W. C. Chen, International Journal of Heat and Mass Transfer, 43(17), 3171-3181, (2000)
19. F. Ihlenburg, I. Babuska, SIAM J. Numer. Anal., 34, 315–358, (1997)
20. F. Ihlenburg, I. Babuska, Comput. Math. Appl., 30, 9–37(1995)
21. V. Isakov, Inverse problems for partial differential equations, Vol127, Applied Mathematical Sciences, Third ed., Springer, Cham, 2017)
22. M. Jourhmane, A. Nachaoui, A relaxation algorithm for solving a Cauchy problem, Proc. of the 2nd Internat. Conf., 30, 151-158(1996)
23. M. Jourhmane, A. Nachaoui, An alternating method for an inverse Cauchy problem, Numerical Algorithms, Proc. of the 2nd Internat. Conf., 21(1-4), 247-260(1999)
24. V. A. Kozlov, V. G Maz'ya, A. V. Fomin, Zh. Vychisl. Mat. I Mat. Fiz., 31, 64–74(1991)
25. V. A. Kozlov, V. G Maz'ya, A. V. Fomin, Zh. Vychisl. Mat. I Mat. Fiz., 31(1), 64–74(1991)
26. AD. Kraus, A. Aziz, J. Welty, An iterative method for solving the Cauchy problem for elliptic equations, Extended surface heat transfer. John Wiley & Sons, New York, 2001)
27. S. Kubo, JSME Int J, 31, 157–166(1988)
28. M. M. Lavrent'ev, V. G. Romanov, S. P. Shishatski, Ill-posed problems of mathematical physics and analysis, volume 64 of Translations of Mathematical Monographs. American Mathematical Society, Providence, RI. Translated from the Russian by J. R. Schulenberger, Translation edited by Lev J. Leifman, (1986)
29. C. S. Liu, F. Wang, Comput. Math. Appl., 76, 1837–1852(2018)
30. J. C. Liu, T. Wei, Appl. Math. Comput., 219(23),10866–10881(2013)

31. C.S. Liu, C. L. Kuo, Engineering Analysis with Boundary Elements, 62, 35-43(2016)
32. L. Marin, J. Comput. Mech., 39, 25-40(2006)
33. L. Marin, L. Elliott, P. J. Heggs, D. B. Ingham, D. Lesnic, X. Wen, Comput. Mech., 31, 367–377(2003)
34. A. Nachaoui, M. Nachaoui, A. Chakib, M. A. Hilal, J. Comput. Appl. Math., 381, 113030(2021)
35. S. M. Rasheed, A. Nachaoui, M. F. Hama, A.K. Jabbar, Advanced Mathematical Models & Applications, 6(2), 89-105(2021)
36. R. Aster, B. Brian, T. Clifford, Parameter Estimation and Inverse Problems, (Academic Press, 2012)
37. F. Yang, P. Zhang, X. X. Li, Appl. Anal., 98, 991-1004(2019)