Academic Science Journal

Solving Bi-harmonic Cauchy problem using a meshless collocation method

Baydaa Khaleel Mostafa, Duaa Jasim , Athraa Falih Hasan , Ibtihal Thabit Jameel, Fatima M. ABOUD

Department of Mathematics – College of Science – Diyala University

scimathms2107@uodiyala.edu.iq

Received: 29 July 2022 **Accepted: 10 October 2022**

DOI: https://doi.org/10.24237/ASJ.02.01.649E

Abstract

In this article a fourth order differential boundary value problem is solved using a mesh less collocation method. The efficiency of the proposed methods is illustrated by solving problems with some examples of a polynomial and Non polynomial exact solutions and by using Conjugate gradient method and Conjugate gradient Least square algorithms and the numerical stability is verified by using a noise for the input boundary data.

Keywords: Bi-Laplacian differential equation, Inverse Cauchy problem, mesh less method. Conjugate Gradient Method, Conjugate Gradient Least Square.

حل مسألة كوشي ثنائية التوافق باستخدام طريقة التجميع بدون شبكة

بيداء خليل مصطفى ,دعاء جاسم, عذراء فالح حسن, ابتهال ثابت, فاطمة محمد عبود

قسم الرياضيات - كلية العلوم - جامعة ديالى

الخلاصة

في هذا البحث تم حل مشكلة القيمة الحدية التفاضلية من الدرجة الرابعة باستخدام طريقة التجميع غير الشبكية. تم التأكد من كفاءة الطريقة المقترحة من خلال حل مشكلات لبعض الأمثلة مع حلول مضبوطة من نوع متعددات الحدود وغير متعددات الحدود وباستخدام خوارزميات طريقة التدرج المترافق و طريقة التدرج المترافق للتربيعات الصغرى كما تم التحقق من الاستقرار العددي باستخدام ضوضاء لبيانات الحدود.

الكلمات المفتاحية : المعادلات التفاضلية ثنائية لابلاس, مسألة كوشي العكسية, طريقة لا شبكية, طريقة التدرج المترافق, طريقة التدرج المترافق للتربيعات الصغرى

Introduction

The fourth order differential equation has many applications in physics (in the fields of fluid and solid mechanics), mathematics, engineering mathematics and computing sciences. In the last decades several iterative and non-iterative methods have been developed where the Dirichlet and or Dirichlet and Neumann conditions are satisfied on the boundary. Some authors have treated the fourth order problem directly and solved the problems in its original form, other authors are preferred to split the problem in two problems of second order, i.e. a couple of problems with Laplace equation, this permit them to benefit from the advantage of the second order equation and all the results lied to them. Recently, a new iterative method has been proposed based on the transformation of the bi-harmonic equation with the Dirichlet and Neumann boundary conditions to an optimization problem similar to an optimal control one [1].Some authors proposed some numerical techniques based on finite difference method (FDM) (see [1]. In [3] the authors proposed some numerical techniques based on finite difference method (FDM) (see [3] by splitting the bi-harmonic problem into two decoupled Poisson equations). Some other works based on the finite element method (FEM), or on a mixed finite volume method (see [4] and the references cited therein). A mesh less Multi-quadric (MQ) collocation method to solve bi-harmonic problems with discontinuous boundary conditions has been proposed in [9]. An iterative method based on the fixed point theory to solve bi-harmonic type equation with mixed boundary conditions see [7]. The authors in [17] have proposed a meshless spectral element method based on the Legendre-Galerkin approximation to solve the two-dimensional bi-harmonic equation. In [10], [11], [12], [13], [14] the authors have given some accurate techniques to solve the inverse problems.

In this work, we propose a meshless collection method following the same method proposed by Rashid et al. in [16]. For this we present the solution as a polynomial expansion and by verifying the Bi-Laplacian differential equation and the boundary condition with the given Cauchy data, we obtain a linear system, we solve this linear system by using CGM and CGLS algorithms by applying the proposed method for some examples with polynomial and non-polynomial exact solutions and we verify the stability of the numerical results by applying some noise on the given data.

The plan for the rest of this article is as follows, In section 2, the Inverse Cauchy problem bi-harmonic equation is stated. A numerical method based on the approximation of the solution by a polynomial expansion is proposed in section 3. The application of the proposed method on some examples with the numerical results in section 4. The stability of the numerical method is checked in section 5. In section 6 we give our conclusion.

Inverse Cauchy problem bi-harmonic equation

We consider the inverse Cauchy problem of bi harmonic equation defined on an annular domain $\Omega \backslash D_\beta \subset R^2$with

$$\Omega = \{(r,\theta) : 0 \le r < 1, \quad 0 \le \theta \le 2\pi\}$$

$$D_\beta = \{(r,\theta) : 0 \le r < \beta, \quad 0 < \beta < 1, \; 0 \le \theta \le 2\pi\}$$

With the boundary $\Gamma_1 \cup \Gamma_2$

$$\Gamma_1 = \{(r,\theta) \quad : r = \rho_e(\theta) \quad 0 \le \theta \le 2\pi\}$$

$$\Gamma_2 = \{(r,\theta) \quad : r = \rho_i(\theta) \quad 0 \le \theta \le 2\pi\}$$

Where $0 < \rho_e(\theta) \le 1$and $0 < \rho_i(\theta) < 1$. The problem is given as follows:

$$\Delta^2 u = F(x,y) \text{ in } \Omega \tag{1}$$

$$u(\rho,\theta) = u_0(\theta), on\ \Gamma_1 \tag{2}$$

$$\partial_n u(\rho,\theta) = h_0(\theta), on\ \Gamma_1 \tag{3}$$

$$\Delta \text{u}(\rho,\theta) = w_0, on\ \Gamma_1 \tag{4}$$

$$\partial_n \Delta \text{u}(\rho,\theta) = w'_0, on\ \Gamma_1 \tag{5}$$

The Cauchy data $u, \partial_n u, \Delta u, \partial_n \Delta u$ are given onΓ_1 which is called asthe accessible part of the boundary, this part of the boundary is over determined (in which there are four boundary conditions), but unfortunately there is no data on Γ_2 (no boundary condition in this part) so this part is known as under-determined or inaccessible part of the boundary. For these raison an inverse Cauchy problem (see [7], [8]) for the bi-Laplacian is formulated to determine the unknown function u on the interior under-determined boundary Γ_2.

We recall that ∂_n is the outer normal derivative which is given by Rasheed et. al. [16].

$$\partial_n \mathrm{u}(\rho,\theta) = \eta(\theta)\left[\frac{\partial u(\rho,\theta)}{\partial\theta} - \frac{\rho'}{\rho^2}\frac{\partial u\,(P,\theta)}{\partial\theta}\right] \tag{6}$$

$$\text{with}\eta(\theta) = \frac{\rho(\theta)}{\sqrt{\rho^2(\theta) + [\rho'(\theta)]^2}} \tag{7}$$

In general cases the radiusρ can be taken as a function ρ, θ i.e. it is a variable, so we can calculate its derivative.

In this paper, we take a special case in which ρ is a constant, so ρ', θ =0.

In fact, the normal derivatives is given by the inner product of the gradient with the normal vector, i.e. $\frac{\partial u}{\partial n}$=$\nabla u.\vec{n}$,so we can express the normal derivate in terms of the derivative with respect to $x$and y.We take the polar coordinates, here the points (x, y)are the points with respect to the ρ, θ.

$$\partial_n \mathrm{u}(\rho,\theta) = \eta(\theta)\left[\cos(\theta) - \frac{\rho'}{\rho^2}\sin(\theta)\right]\partial \mathrm{x}(\Delta \mathrm{u}) + \eta(\theta)[\sin(\theta) - \frac{\rho'}{\rho^2}\cos(\theta)]\,\partial \mathrm{y}(\Delta \mathrm{u}) \tag{8}$$

Expression of Solution as a polynomial expansion

We consider that the solution $u(x, y)$ is expressed as the following polynomial expansion:

$$u(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{i} c_{ij} x^{i-j} y^{j-1} \tag{9}$$

Now, we express the problem in (1-5) in form of the expansion in (9). To do so, we find

$$\partial_x u(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{i} c_{ij}(i-j)\, x^{i-j} y^{j-1} \tag{10}$$

$$\partial_y u(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{i} c_{ij}(j-1)\, x^{i-j} y^{j-2} \tag{11}$$

From which the different degree of derivatives are calculated to find $\Delta u, \partial_n \Delta, \Delta^2 u$.

$$\Delta u(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{i} c_{ij} \left[(i-j)(i-j-1)x^{i-j-2}y^{j-1} + (j-1)(j-2)x^{i-j}y^{j-3}\right] \tag{12}$$

Then

$$\partial_x(\Delta u) = \sum_{i=1}^{m} \sum_{j=1}^{i} c_{ij} \left[(i-j)(i-j-1)(i-j-2)x^{i-j-3}y^{j-1} + (i-j)(j-1)(j-2)\, x^{i-j-1}y^{j-3}\right] \tag{13}$$

$$\partial_y(\Delta u) = \sum_{i=1}^{m} \sum_{j=1}^{i} c_{ij} \left[(i-j)(i-j-1)(j-1)x^{i-j-2}\, y^{j-2} + (j-1)(j-2)(j-3)x^{i-j}y^{j-4}\right] \tag{14}$$

By using (8), the normal derivative of the Laplacian is given by the following:

$$\partial_{\mathrm{n}}\,(\Delta u)(x,y)=\sum_{i=1}^{m}\sum_{j=1}^{i}c_{ij}\eta(\theta)[\cos(\theta) -\frac{\bar{\rho}}{\rho^2}\sin(\theta)][(i-j)(i-j-1)(i-j-2)x^{i-j-3}\,y^{j-1} + (j-1)(j-2)(i-j)x^{i-j}y^{j-3}]+\ \eta(\theta)[\sin(\theta) -\frac{\bar{\rho}}{\rho^2}\cos(\theta)](i-j)(i-j-1)(j-1)x^{i-j-2}y^{j-2} + (j-1)(j-2)(j -3)x^{i-j}y^{j-4}\ ] \tag{15}$$

and

$$\Delta^2(x,y)=\sum_{i=1}^{m}\sum_{j=1}^{i}c_{ij}(\,i-j\,)(i-\ j-1\,)(\,i-j-\ 2\,)(i-j-3)x^{(i-j-4)}y^{(j-1)} + 2(i-j)(i-j-1)(j-1)(j-2)x^{(i-j-2)}y^{(j-3)} +(j-1)(j-2)(j-3)(j-4)x^{(i-j)}y^{(j-5)} \tag{16}$$

The coefficients c_{ij}must be determined, the number of these coefficientsc_{ij},is= $\frac{m(m+1)}{2}$. To obtain a linear system we express c_{ij} as a vector $\boldsymbol{c}$of length n where the index ij is used to obtain the index of the components of $\boldsymbol{c}=[\boldsymbol{c}_\iota]_{\boldsymbol{n}\times\boldsymbol{1}}$ by taking $\iota=\frac{i(i-1)}{2}+j$, so the unknowns function$u(x,y)$ can be expressed as an inner product of a row of variables, say $\boldsymbol{v}'$, with a column of coefficient vector $\boldsymbol{c}$, i.e.

$$u=\boldsymbol{v}'.\boldsymbol{c} \tag{17}$$

where

$$\boldsymbol{v}'=[1,x,y\,,x^2,xy,y^2\,,x^3,\cdots],\qquad \boldsymbol{c}=\begin{bmatrix}c_1\\c_2\\c_3\\\vdots\\c_n\end{bmatrix}$$

Similarly, the normal derivative ∂_n u can be represented as a scalar product of a row of variables, say $\boldsymbol{e}$, with a column of coefficient vector $\boldsymbol{c}$, i.e. $u = \boldsymbol{e}.\boldsymbol{c}$ where the component of$\boldsymbol{e}$are given by

$$e_k = \eta(\theta)[(\mathrm{i} - \mathrm{j})x^{i-j-1}y^{j-1}\left(\cos(\theta) - \frac{\bar{\rho}}{\rho^2}\sin(\theta) + (\mathrm{j}-1)\, x^{i-j}y^{j-2}\left(\sin(\theta) - \frac{\bar{\rho}}{\rho^2}\cos(\theta)\right)\right)]. \quad (18)$$

Also, the Laplacian can be expressed as an inner product of a row of variables, say d, with a column of coefficient vector $\boldsymbol{c}$, i.e.$\Delta u = \boldsymbol{d}.\boldsymbol{c}$ where the component of $\boldsymbol{d}$ are given by

$$\boldsymbol{d_k} = [(\boldsymbol{i}-\boldsymbol{j})(\boldsymbol{i}-\boldsymbol{j}-\mathbf{1})\boldsymbol{x}^{\boldsymbol{i-j-2}}\,\boldsymbol{y}^{\boldsymbol{j-1}} + (\boldsymbol{j}-\mathbf{1})(\boldsymbol{j}-\mathbf{2})\boldsymbol{x}^{\boldsymbol{i-j}}\boldsymbol{y}^{\boldsymbol{j-3}}] \quad (19)$$

and the normal derivative of the Laplacian can be expressed as an inner product of a row of variables, say ϱ, with a column of coefficient vector $\boldsymbol{c}$, i.e.$\partial_n \Delta u = \varrho.\boldsymbol{c}$ where the component of ϱ are given by

$$\begin{aligned} \boldsymbol{e_k} &= \eta(\theta)[\cos(\theta) - \frac{\bar{\rho}}{\rho^2}\sin(\theta)][(i-j)(i-j-1)(i-j-2)x^{i-j-3}\,y^{j-1} \\ &\quad +(j-1)(j-2)(i-j)x^{i-j-1}y^{j-3}] \\ &\quad + \eta(\theta)[\sin(\theta) - \frac{\bar{\rho}}{\rho^2}\cos(\theta)](i-j)(i-j-1)(j-1)x^{i-j-2}y^{j-2} \\ &\quad + (j-1)(j-2)(j-3)x^{i-j}y^{j-4}\,] \end{aligned} \quad (20)$$

Finally, the bi-Laplacian can be expressed as an inner product of a row of variables, say ξ, with a column of coefficient vector $\boldsymbol{c}$, i.e.$\Delta^2 u = \xi.\boldsymbol{c}$where the component of$\varrho$are given by

$$\begin{aligned} \xi_k &= (j-1)(i-j-1)(i-j-2)(i-j-3)x^{i-j-4}y^{j-1} \\ &\quad + 2(i-j)(i-j-1)(j-1)(j-2)x^{i-j-2}y^{j-3} \\ &\quad + (j-1)(j-2)(j-3)(j-4)x^{i-j}y^{j-5} \end{aligned} \quad (21)$$

Now, we are ready to construct the linear system

$$Ac = b \tag{22}$$

is constructed such that A, b are matrices with 5 blocks:

1. The first one is constructed by satisfying the first boundary condition given in (2) using the formula in (17) for some given function u_0 and for some selected points on Γ_1,
2. The second block by satisfying the second boundary condition in (3) using the formula in (18) for some given function h_0 and for some selected points on Γ_1,
3. The third block by satisfying the third boundary condition in (4) using the formula in (19) for some given function w_0 and for some selected points on Γ_1,
4. The fourth block by satisfying the fourth boundary condition in (5) using the formula in (20) for some given function w'_0 and for some selected points on Γ_1,
5. The fifth block by satisfying the bi-Laplacian differential equation in (1) using the formula in (21) for some given function F and for some selected points in the domain $\Omega\backslash D_\beta$.

For this we selectn_1 points on the boundary Γ_1, say $(xi, yi), i = 1, \dots, n_1$ to satisfy the condition (2-5) and we selectn_2 points in the domain $\Omega\backslash D_\beta$, say $(xj, yj), j = 1, \dots, n_2$ to satisfy the equation (1). So the vector b is of order $4n_1 + n_2$ and A is $(4n_1 + n_2) \times n$ matrix and the vector c is of order$n = \frac{m(m+1)}{2}$.

$$A = \begin{bmatrix} v'_1 \\ \vdots \\ v'_{n1} \\ e'_1 \\ \vdots \\ e'_{n1} \\ d'_1 \\ \vdots \\ d'_{n1} \\ \varrho'_1 \\ \vdots \\ \varrho'_{n1} \\ \xi'_1 \\ \vdots \\ \xi'_{n2} \end{bmatrix} b = \begin{bmatrix} u_0(\theta_1) \\ \vdots \\ u_0(\theta_{n1}) \\ h_0(\theta_1) \\ \vdots \\ h_0(\theta_{n1}) \\ w_0(\theta_1) \\ \vdots \\ w_0(\theta_{n1}) \\ w'_0(\theta_1) \\ \vdots \\ w'_0(\theta_{n1}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{23}$$

So the inverse Cauchy problem for the bi-Laplacian equation is reduced to solve the linear system in (23).

Solving the linear system by using CGM and CGLS algorithms

Consider the linear system given in (22), to solve this system, we use two well-known algorithms, which are the Conjugate Gradient method (CGM) and the Conjugate Gradient least square method (CGLS) (see [16], [2]).

Stopping criterion and Initial guess

An important thing that we need to care about to start and stop a numerical method, is that the initial guess (we take it the zero vector with a propitiate order) and for stop these algorithms is the following stopping criteria:

$$\|r_i\| < Tol \tag{24}$$

$$\frac{\|r_i\|}{\|b\|} < Tol \tag{25}$$

Numerical results and discussion

For illustrating the efficacy of the proposed method, we consider some examples with polynomial and non-polynomial exact solutions. The given exact solution is used to calculate:

- the function F in the domain $\Omega \backslash D_\beta$
- the trace of the exact solution is equal to $u_0$on $\Gamma 1$
- the normal derivative the exact solution is equal to h_0 on Γ_1
- the Laplacian of the exact solution is equal to w_0 on Γ_1
- the normal derivative of the Laplacian is equal to w'_0on $\Gamma 1$

in addition to these data, we use the zero initial guess and the CGM and $CGLS$ like-methods are used with some propitiate tolerance and stopping criteria. For the algorithms of CGM and CGLS see [16].

Example(1)

We consider the problem (1-5) with an exact solution is $u(x\,,y) = x^4 - y^4$ with an annular domain bounded by$\rho_e(\theta) = 1, \rho_i(\theta) = 0.5$, the outer part of the boundary is the accessible part $\Gamma 1$ is taken with $\rho_e(\theta) = 1$ and β= 2, the number of points on the outer boundary is taken to be n_1=100 and the number of the internal domain points isn_2=1000 . We vary$m$ from 2 to 10 for both algorithm CGM and CGLS.

$u(x,y) = x^4 - y^4$ whenn_1 =100 andn_2 =1000 with $Tol = 10^{-10}$				
M	No. of Iteration for CGM	Relative Error with CGM	No. of Iteration for CGLS	Relative Error with CGLS
2	-	-	-	-
3	-	-	-	-
4	-	-	-	-
5	**2**	**1.8e-12**	**3**	**2.1189e-12**
6	2	9.8e-12	4	3.2864-12
7	8	6.3e-09	8	0.22957
8	11	3.6e-09	12	0.22957
9	23	1.5e-07	17	0.0094068
10	28	2.7e-08	18	0.0094068
11	45	0.00042	36	0.32837

We note that when we take $m = 2,3,4$, the both algorithms do not attend an accepted accuracy of convergence. The exact solution is a polynomial of degree 4, so the ideal approximation obtained for $m = 5$, i.e. we approximate a polynomial of degree 4 by a polynomial of degree 4, so when we take $m = 5$, the convergence is attended with 3 iterations with a relative error **2.1189e-12**for CGLS and with 2 iteration with a relative error**1.8e-12**for CGM which are very ideal.

Now we take the case with n_1=400 and n_2=8000,$Tol = 10^{-10}$

$\boldsymbol{u(x,y) = x^4 - y^4}$ **when** n_1**=400 and** n_2**=8000**, $\boldsymbol{Tol} = 10^{-10}$				
M	No. of Iteration for CGM	Relative Error with CGM	No. of Iteration for CGLS	Relative Error with CGLS
2	-	-	-	-
3	-	-	-	-
4	-	-	-	-
5	**2**	**3.7e-12**	**3**	**4.4697e-14**
6	3	2.7e-12	4	1.4576e-11
7	9	2.4e-10	10	0.22957
8	11	2.6e-09	15	0.22957
9	24	5.6e-08	17	0.0097726
10	32	9.4e-09	27	0.0097726
11	55	0.00042	42	0.033654

Academic Science Journal

Similarly to the previous case, we note that when we take $m = 2,3,4$, the both algorithms do not converge. In fact, our exact solution is a polynomial of degree 4, so the ideal approximation obtained for $m = 5$, i.e. we approximate a polynomial of degree 4 by a polynomial of degree 4, so when we take $m = 5$ the number of iteration equal to 4 and the relative error is equal 4.4697e-14 for CGLS and 3.7e-12 for CGM which are very ideal.

The following figures show the errors and the comparison of the exact solution and the approximate solution calculated by CGM.

In the following the figure of the exact and approximate solutions by CGM and CGLS.

In the following the error by CGM.

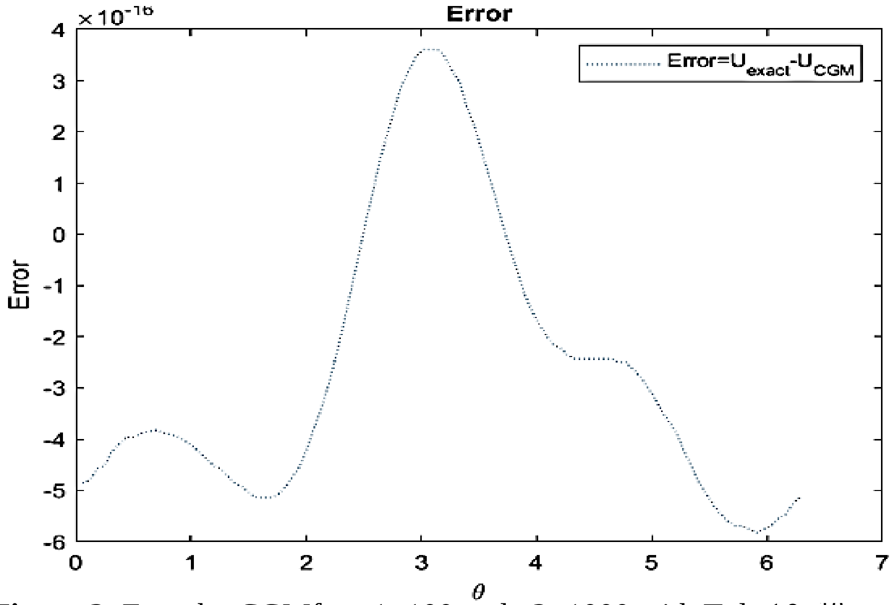

Figure 2: Error by CGMfor n1=100 and n2=1000 with Tol=10^{-10}

Example (2):

In this example we suppose that the exact solution is $u(x\,,y) = x^4 + y^4$the domain is bounded by$\rho(\theta)$=1 and Γ1 is defined taking β= 2 the number of boundary collection used for discretizing the boundary is taken to be n1=100 and n_r=10 and the number of internal collectionn_2 =1000 .

Case 1:n_1=100 and n_2=1000 with $Tol = 10^{-10}$

$u(x,y) = x^4 - y^4$ when n_1 =100 and n_2 =1000 with $Tol = 10^{-10}$				
M	No. of Iteration for CGM	Relative Error with CGM	No. of Iteration for CGLS	Relative Error with CGLS
2	-	-	-	-
3	-	-	-	-
4	-	-	-	-
5	**6**	**5.6678e-14**	**3**	**2.1189e-12**
6	11	6.4118e-13	4	3.2864-12
7	20	1.6091e-12	8	0.22957
8	27	2.4387e-12	12	0.22957
9	88	9.3526e-10	17	0.0094068
10	126	1.2239e-09	18	0.0094068

For m=5 the number of iteration =6 and the relative error is equal **5.6678e-14 for CGM** and**2.1189e-12 for CGLS** that is a good approximation.

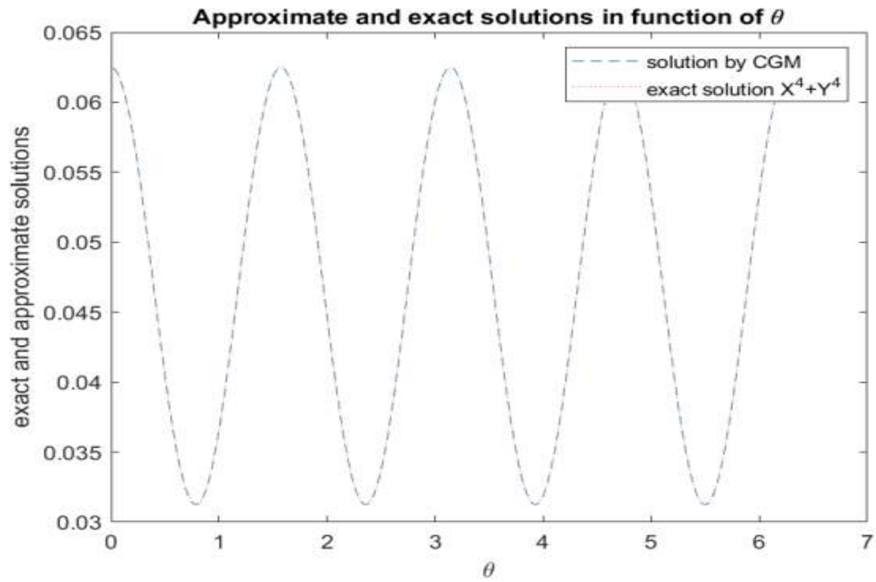

Figure 3: Approximate Solution by CGM and exact solution $u(x,y) = x^4 + y^4$for $n_1 = 400$ and $n_2 = 8000$ with $Tol = 10^{-10}$

In the following the figure of the error:

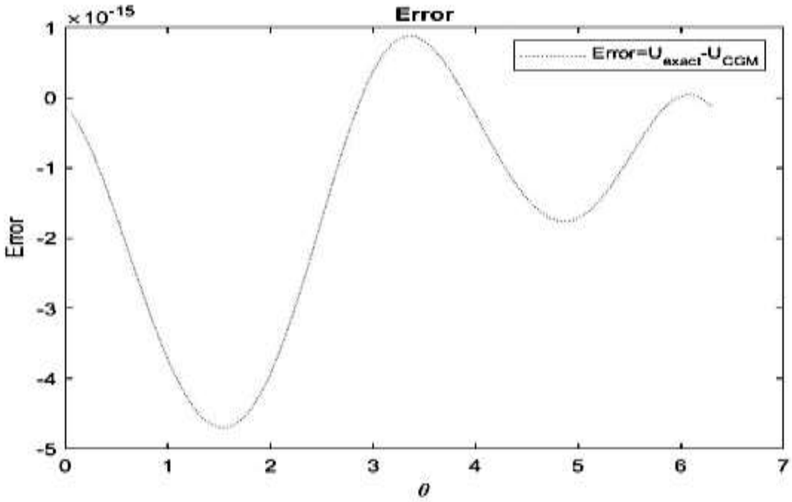

Figure 4: Error by CGM for $n_1 = 400$ and $n_2 = 8000$ with $Tol = 10^{-10}$

Example(3):

In this example we solve the problem (1-5) by supposing that the exact solution is

$$u(x, y) = \exp(x)\, cos(y) + x^4$$

with the same domain and boundary for the previous examples .The number of boundary on the boundary is taken to be $n_1 = 200$ and the number of the points in the interior domain is $n_2 = 4000$. As the previous examples we study the cases of $m = 2, \dots, 11$,for both CGM and CGLS algorithms, with a tolerance $Tol = 10^{-12}$.

$u(x, y) = \exp(x)\, cos(y) + x^4$ when n_1 =200 and n_2=4000 with $Tol = 10^{-12}$				
M	No. of Iteration for CGM	Relative Error with CGM	No. of Iteration for CGLS	Relative Error with CGLS
2	-	-	-	-
3	-	-	-	-
4	-	-	-	-
5	13	0.026025063244862	13	0.026025063245107
6	21	0.0245600183899321	25	0.024560018387322
7	**43**	**0.023827973140670**	**44**	**0.023827973190222**
8	73	0.032073059303643	76	0.032073059309326
9	172	0.032816995991338	154	0.032816994769686
10	370	0.032398126449167	312	0.032398123460117

Academic Science Journal

In the following the figure of the exact and approximate solutions by CGM and CGLS.

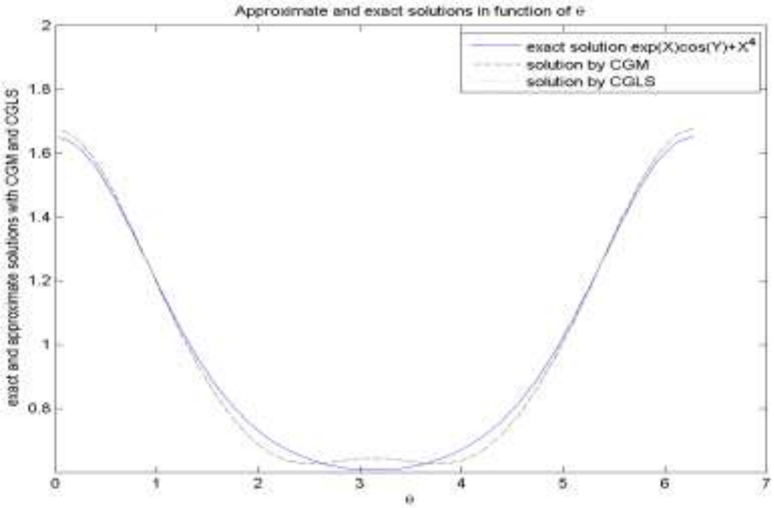

Figure 5: Exact and approximate solutions by CGM and CGLS for $u(x, y) = \exp(x)\, cos(y) + x^4$ when n1=200 and n2=4000 with Tol=10^{-12}

In the following the error by CGM.

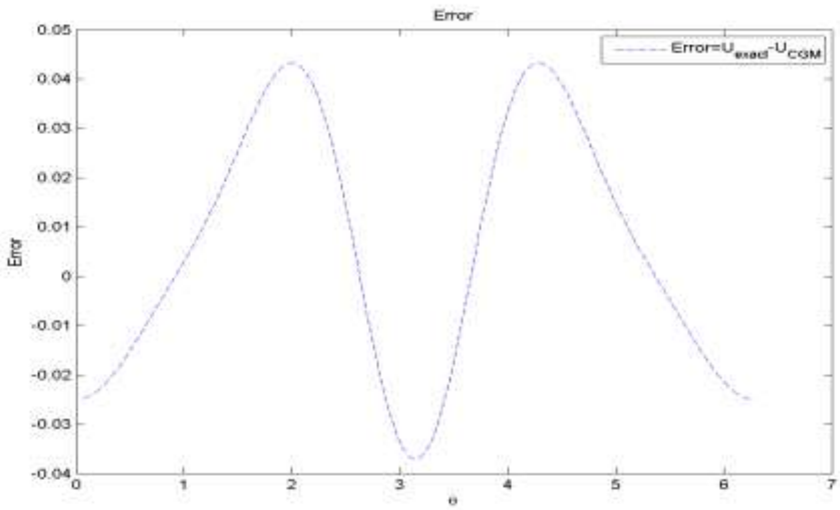

Figure 6: Error between exact solution and the calculate approximation by CGM

In the following the figure of the error CGLS.

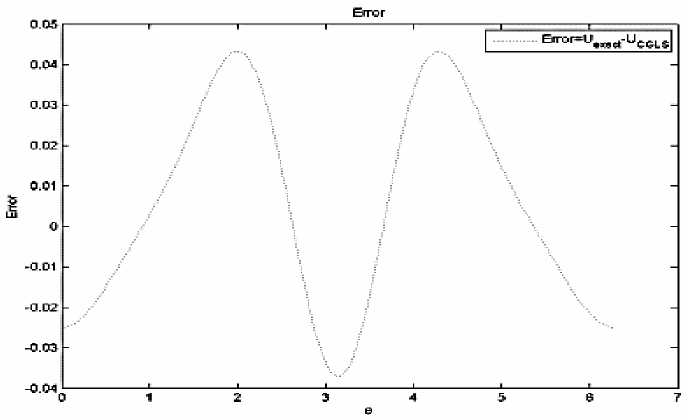

Figure 7: Error between exact solution and the calculate approximation by CGLS

Stability and effect of a noise

It is well-known that the inverse problem depends on the data that may have some error due to the measurement error. So, we check the effect of any noise of the data on the calculate solution. Here, we give the following form of the noise on the Cauchy data:

$$u_0(\theta) = u_{ex}(\rho, \theta) + \sigma * rand$$

Where $rand$ is the Gauss random error and σ is the deviation of measurement errors. σ is the noise level, it takes the values 0.001, 0.01, 0.05 and 0.1.

We test the effect of the noise for example 3.

$u(x, y) = \exp(x)\, cos(y) + x^4$ when n_1 =200 and n_2=4000 with $Tol = 10^{-12}$				
σ	No. of Iteration for CGM	Relative Error with CGM	No. of Iteration for CGLS	Relative Error with CGLS
0.1	**58**	0.048048673438369	**61**	0.048048673438261
0.01	61	0.023294161460917	65	0.023294161455813
0.05	60	0.029669222287058	62	0.029669222287195
0.001	57	0.023741709140837	64	0.023741709141370

In the following the figures of each case:

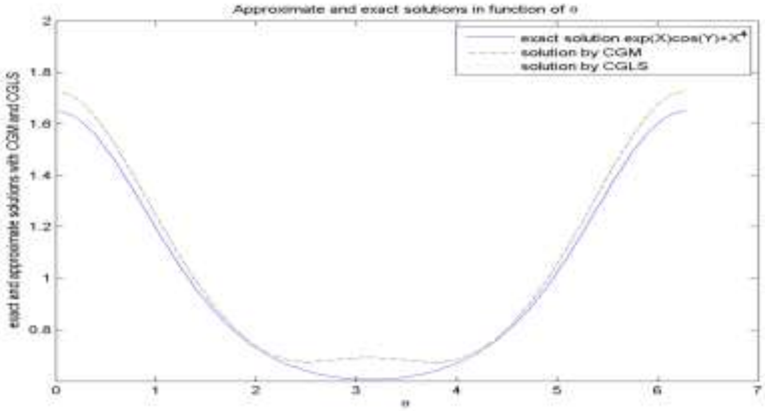

Figure 8: Noise parameter$\sigma = 0.1$

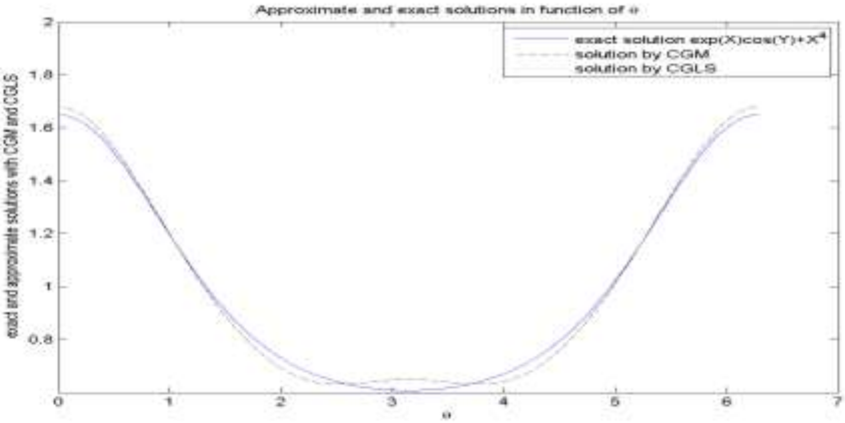

Figure 9 : Noise paramètre $\sigma = 0.01$

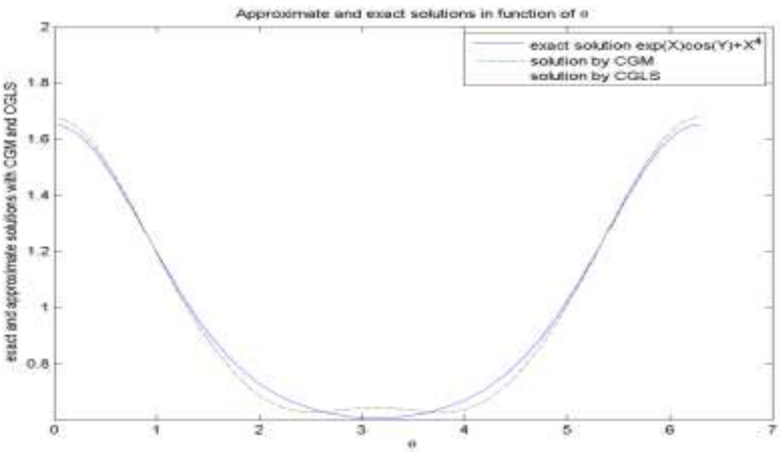

Figure 10 : Noise paramètre $\sigma = 0.05$

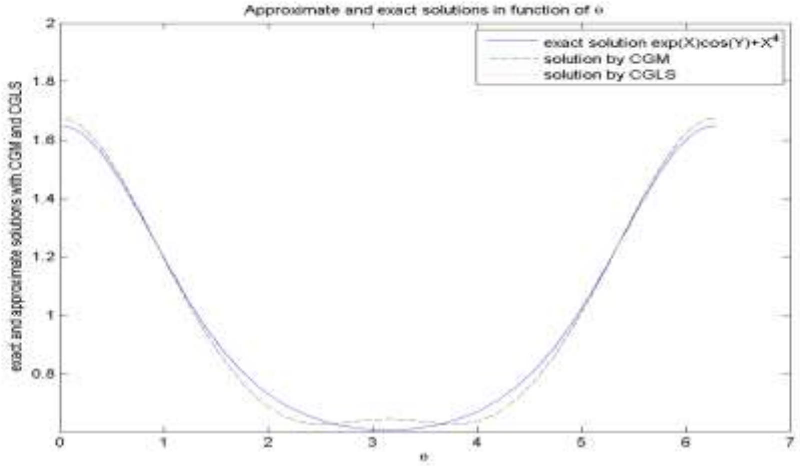

Figure 12 : Noise paramètre $\sigma = 0.001$

In fact, the problem in example 3 is ill-posed with Condition number about 2.355744917801085e+003. Figures present the exact and calculate approximate solutions obtained by both CGM and CGLS on the boundary Γ_2.These figures show that for a big value $\sigma = 0.1$ the approximate solution move away from the exact solution and the error multiplied by 2 and by reducing the value of σuntil $\sigma = 0.001$the approximate solution be closer to the exact solution, the most important for all these cases the solution still stable for both CGM and CGLS. These both approximations still good, even for a high value of noise until $\sigma = 0.1$ relative random parameter,in fact our problem is highly ill-conditioned with condition number2.355744917801085e+003.

Conclusion

We solve the inverse Cauchy problem of bi-Laplacian differential equation in an annular domain, the unknown data on a part of the boundary are recovered

from the over-specified Cauchy boundary conditions. The inverse Cauchy problem is reformulated to solve a direct problem benefiting from a polynomial expansion of the solution. Different kind of numerical examples with polynomial and non-polynomial exact solution are presented to confirm that our proposed method overcome the severe ill-posedness of the inverse Cauchy problem. The stability of the method is checked by applying a different value of noise on the Cauchy data.

References

1. F. Aboud, A. Nachaoui, M. Nachaoui, On the approximation of a Cauchy problem in a non-homogeneous medium, J. Phys.: Conf. Ser, 1743 012003(2021)
2. F. Aboud, I. Th. Jameel, A. F. Hasan, B. Kh. Mostafa and A. Nachaoui, Polynomial approximation of an inverse Cauchy problem for Helmholtz type equations, Advanced Mathematical Models and Applications, 7 , 306-322(2022)
3. S. Andrieux, T. N. Baranger, A. Ben Abda, Solving Cauchy problems by minimizing an energy-like functional, Inverse Problems, 22, 115 - 133(2006)
4. K. A. Berdawood, A. Nachaoui, R. Saeed, M. Nachaoui, F. Aboud, An alternating procedure with dynamic relaxation for Cauchy problems governed by the modified helmholtz equation, Advanced Mathematical Models & Applications, 5 (1), 131-139(2020)
5. L. Boudjaj, A. Naji, F. Ghafrani, Solving bi-harmonic equation as an optimal control problem using localized radial basis functions collocation method, Engineering Analysis with Boundary Elements, 10, 208–217(2019)
6. Q. A. Dang, Mixed Boundary-Domain Operator in Approximate Solution of Biharmonic Type Equation, Vietnam Journal of Mathematics, 26(3), 243-252(1998)
7. J. Hadamard, Lectures on Cauchy's problem in linear partial differential equations, (Dover Publications, New York, 1953)
8. V. Isakov, Inverse problems for partial differential equations, (Springer, Cham, 2017)
9. S. Chantasiriwan, Solutions to harmonic and bi-harmonic problems with discontinuous boundary conditions by collocation methods using multi-quadrics as basis functions, Int Commun Heat Mass Transf, 34(3),313–20(2007)
10. C. S. Liu, S. N. Atluri, A highly accurate technique for interpolations using very high- order polynomials, and its applications to some ill-posed linear problems, Computer Modeling in Engineering & Sciences, 43 (1), 253-276(2009)
11. C. S. Liu, C. L. Kuo, A multiple-scale Pascal polynomial triangle solving elliptic equations and inverse Cauchy problems, Engineering Analysis with Boundary Elements, 62, 35-43(2016)
12. A. Nachaoui, E. S. Al-Rawi, A. F. Qasim, Solving Three Dimensional and Time Depend- ing PDEs by Haar Wavelets Method, Open Access Library Journal, 5(5), 1-18(2018)

13. A. Nachaoui, M. Nachaoui, B. Gasimov, Parallel numerical computation of an analytical method for solving an inverse Problem, Advanced Mathematical Models & Applications, 6(2), 162-173(2021)
14. A. Nachaoui, Cauchy's problem for the modified bi-harmonic equation: ill-posedness and iterative regularizing methods, The First International Conference on New Trends in Applied Mathematics, 19-21(2022).
15. C. C. Paige, M. A. Saunders, LSQR: Sparse linear equations and least squares problems, ACM Transactions on Mathematical Software, 8, 195-209(1982)
16. S. M. Rasheed, A. Nachaoui, M. F. Hama, A. K. Jabbar, Regularized and preconditioned conjugate gradient like-methods for polynomial approximation of an inverse Cauchy problem, Advanced Mathematical Models & Applications, 6 (2), 89-105(2021)
17. Q. Zhuang, L. Chen, L. Galerkin, Spectral-element method for the bi-harmonic equations and its applications, Comput Math Appl, 74(12),2958–2968(2017)