

重调和方程的 Argyris 元法

岳 运

贵州大学数学与统计学院, 贵州 贵阳

收稿日期: 2023 年 10 月 9 日; 录用日期: 2023 年 11 月 10 日; 发布日期: 2023 年 11 月 22 日

摘 要

本文考虑求解四阶椭圆边值问题的协调有限元方法: Argyris 元。重调和方程有着非常重要的物理背景和广泛的应用。例如, 在薄板弯曲问题中, 用来描述其弯曲程度, 是一个经典的问题。Argyris 元是四阶问题的协调有限元, 也即是 Argyris 元基函数构成的有限元空间是解空间的子空间。而 Argyris 元的基函数是 5 次多项式, 推导而言比较麻烦。本文利用每个单元上的单元信息求解出来每个单元上的基函数, 再利用单元上的基函数构造多项式逼近原函数, 这样便不必要推出其基函数的一般表达式, 减少了麻烦的计算。

关键词

重调和, 有限元, Argyris 元

The Argyris Element Method of Biharmonic Equation

Yun Yue

School of Mathematics and Statistics, Guizhou University, Guiyang Guizhou

Received: Oct. 9th, 2023; accepted: Nov. 10th, 2023; published: Nov. 22nd, 2023

Abstract

In this paper, we consider the coordination finite element method for the fourth order elliptic boundary value problem: Argyris element. Reharmonic equations have a very important physical background and wide applications. For example, in the problem of sheet bending, used to describe the degree of bending, is a classic problem. The Argyris element is the coordinated finite element of the fourth order problem, that is, the finite element space formed by the Argyris element basis function is a subspace of the solution space. But the basis function of Argyris element is a polynomial of 5 times, which is more troublesome to derive. In this paper, the unit information on each cell is used to solve the basis function on each cell, and the basis function on the unit is used to construct polynomial approximating the original function, which is not necessary to introduce the general expression of its basis function, which reduces the troublesome calculation.

Keywords

Biharmonic, Finite Element, Argyris Element

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

重调和方程是一个非常重要的概念. 它用于描述非线性系统中的振动, 主要以动力学参量表示, 并将参量扩展到不同类型的动态系统. 这种描述可以用于扩大动态系统的参数空间, 从而使用有关动态系统的数据, 采集方法来构建出更复杂的动力学模型. 此外, 重调和方程还可以使用来解决系统的不确定性问题, 如可靠度、稳定性以及发展性等问题, 并能够提供有用的信息来帮助研究者对某些系统的行为有更好的理解和分析. 重调和方程也可以结合之前研究中使用的经典方法来解决一些实际应用中的棘手问题, 比如利用有限元或非线性参数拟合的方法来得到提高系统性能的反馈控制算法, 也可以基于统计学模型来构建可靠性安全性模型, 以及使用状态对系统的行为进行模拟和估计. 总而言之, 重调和方程是一种有用的概念, 它在科学、工程、物理、化学等

多个领域中都有应用. 它主要用于描述多种振动情况, 可以用于分析系统行为, 并据此提出可靠的解决方案.

有限元起源于 1943 年, 纽约大学的 R. Courant 尝试在一系列三角形区域上定义分片连续函数, 并采用最小势能原理求解 St. Venant 扭转问题. 1954 年, 斯图加特大学的 J. H. Argyris 了结构分析的能量原理, 并出版了相关书籍, 为后续有限单元法的研究奠定了重要基础, 其提出的 Argyris 元是解决重调和方程的重要方法, 更多的内容看 [1]. 本文基于前人研究四阶问题的方法基础上, 使用 Argyris 有限元数值离散重调和方程, 并推导了单元上 Argyris 元的基函数, 最后得出数值解, 并分析了误差的收敛阶.

重调和方程起源于弹性力学中周边固定的薄板弯曲问题, 这一问题如下描述:

$$\Delta^2 u = f \quad \text{in } \Omega \quad (1.1)$$

边界条件:

$$u = \frac{\partial u}{\partial n} = 0 \quad \text{on } \partial\Omega \quad (1.2)$$

这里 $\Omega \subset \mathbb{R}^2$ 是有界矩形区域, 边界为: $\partial\Omega$, \mathbf{n} 是边界 $\partial\Omega$ 的外法向量. 其中 $f \in L^2(\Omega)$, Δ 是 Laplace 算子.

问题 (1.1)-(1.2) 的弱形式为:

$$a(u, \varphi) = F(\varphi) \quad (1.3)$$

其中 $\varphi \in H_0^2(\Omega)$, 且 $H_0^2(\Omega) = \{\varphi \in H^2(\Omega) : \varphi = \frac{\partial \varphi}{\partial n} = 0\}$, 这里

$$a(u, \varphi) = \langle \Delta u, \Delta \varphi \rangle \quad (1.4)$$

$$F(\varphi) = \langle f, \varphi \rangle \quad (1.5)$$

2. 弱解的存在性和唯一性

Lemma 1 (Lax-Milgram) [2] 设 $B : H \times H \rightarrow R$ 是双线性型, 且存在正数 α, β 使得

$$(1) |B[u, v]| \leq \alpha \|u\| \|v\|$$

$$(2) B[u, v] \geq \beta \|u\|^2$$

且 $u, v \in H$, 设 $f \in H^*$ 是有界线性泛函, 则存在唯一的 $u \in H$ 使得 $B[u, v] = (f, v), \forall v \in H$

Theorem 1 (Existence and Uniqueness) [2] 设 $V = H_0^2(\Omega)$, 内积是 $\langle v, w \rangle = \iint_{\Omega} \Delta v \cdot \Delta w dx dy, \forall v, w \in V$. 由内积诱导的范数为 $\|v\|_{2, \Omega} = \left(\iint_{\Omega} |\Delta v|^2 dx dy \right)^{\frac{1}{2}}$. 则 (1.3) 有唯一解.

Proof. 因为 $V = H_0^2(\Omega)$, 则 V 是 Hilbert 空间. 显然 a 是双线性型的, 且

$$\begin{aligned} a(u, v) &= \iint_{\Omega} \Delta u \cdot \Delta v dx dy \\ &\leq \left(\iint_{\Omega} |\Delta u|^2 dx dy \right)^{\frac{1}{2}} \left(\iint_{\Omega} |\Delta v|^2 dx dy \right)^{\frac{1}{2}} \\ &= \|v\|_{2, \Omega} \|u\|_{2, \Omega} \end{aligned}$$

另外, $\forall v \in V, a(v, v) = \|\Delta v\|_{2,\Omega}^2$, 由于 $v|_{\partial\Omega} = 0$ and *Poincaré* 不等式, 令 $\Delta v = \nabla w$, 我们得到

$$a(v, v) = \|\nabla w\|_{2,\Omega}^2 \geq C\|w\|_{2,\Omega}^2$$

最后, 因为 $f \in L^2(\Omega)$, 我们可以得到

$$F(v) = \langle f, v \rangle = \iint_{\Omega} f(x, y)v(x, y)dxdy \leq \left(\iint_{\Omega} |\Delta f|^2 dx dy \right)^{\frac{1}{2}} \left(\iint_{\Omega} |\Delta v|^2 dx dy \right)^{\frac{1}{2}} = C\|v\|_{2,\Omega}$$

而且因为 f 是一个在 V 上的连续函数. 由 *Lax - Milgram* 定理, 知道 (1.3) 存在唯一解.

3. Argyris 元

3.1. 有限元空间

在本节中, 令 τ_h 表示区域 $\bar{\Omega}$ 上的三角形单元 [1] [2], 且 $h = \max_{K \in \tau_h} h_K$, 这里 $h_K = \text{diam}(K)$. 我们以 $\tau_h, \varepsilon_h, \mathcal{N}_h$ 分别表示三角形单元、边及其顶点.

现考虑 (1.3) 的有限元空间, 区域 Ω 是矩形区域, 划分其三角形单元, 设其单元 K 直径为 $h_K \leq h$.

令 V_h 为方程 (1.3) 的有限元空间, 则在空间 V_h 寻找基函数 φ_h , 且有:

$$a_h(u_h, \varphi_h) = (f, \varphi_h), \forall \varphi_h \in V_h \quad (3.1)$$

Argyris 有限元有 21 个自由度, 如图 1. 包含了函数节点值、一阶偏导数值, 二阶偏导数值以及边上的中点的法向导数值 $u_i, \left(\frac{\partial u}{\partial x}\right)_i, \left(\frac{\partial u}{\partial y}\right)_i, \left(\frac{\partial^2 u}{\partial x \partial x}\right)_i, \left(\frac{\partial^2 u}{\partial y \partial y}\right)_i, \left(\frac{\partial^2 u}{\partial x \partial y}\right)_i, \left(\frac{\partial u}{\partial n}\right)_i, i = 1, 2, 3$, 它的基函数是 5 次多项式, 如图 1. 更多的 Argyris 有限元空间看 [3], 定义如下:

$$S_a = \{ \varphi_h \in \mathbb{P}_5(\tau_h) : \varphi_h \text{ 是连续的对任意 } p \in \mathcal{N}_h, \\ \varphi_h \text{ 为 } 0 \text{ 对任意的 } p \in \mathcal{N}_h \cap \partial\Omega, \\ \text{且 } \int_e \frac{\partial \varphi_h}{\partial n} = 0, \forall e \in \varepsilon_h \}$$

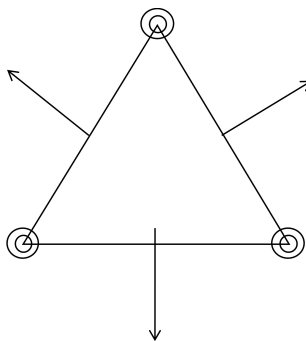


Figure 1. Argyris finite element

图 1. Argyris 有限元

3.2. Argyris 元基函数

下面介绍 argyris 元基函数的推导过程 [4]: $\forall \tau_h \in \Omega$, 其节点值设为 u_i, u_j, u_k , 同理, 其一阶偏导, 二阶偏导为: $u_{x_i}, u_{x_j}, u_{x_k}, u_{y_i}, u_{y_j}, u_{y_k}, u_{x_i x_i}, u_{x_j x_j}, u_{x_k x_k}, u_{y_i y_i}, u_{y_j y_j}, u_{y_k y_k}, u_{x_i y_i}, u_{x_j y_j}, u_{x_k y_k}$ 其法向导数为 $(\frac{\partial u}{\partial n})_i, (\frac{\partial u}{\partial n})_j, (\frac{\partial u}{\partial n})_k$. 共计 21 个自由度, 便有 21 个基函数, 其中 i, j, k 是三角形单元的三个顶点的序号, 按逆时针方向旋转. 不妨设 $\varphi_k, k = 1, \dots, 21$ 是每个单元上的基函数, 则有:

$$\text{令 } c^T = (c_1^k \ c_2^k \ c_3^k \ c_4^k \ c_5^k \ c_6^k \ c_7^k \ c_8^k \ c_9^k \ c_{10}^k \ c_{11}^k \ c_{12}^k \ c_{13}^k \ c_{14}^k \ c_{15}^k \ c_{16}^k \ c_{17}^k \ c_{18}^k \ c_{19}^k \ c_{20}^k \ c_{21}^k)$$

$$A(x, y) = (x^5 \ x^4 \ x^3 \ x^2 \ x \ y^5 \ y^4 \ y^3 \ y^2 \ y \ x^4 y \ x^3 y^2 \ x^3 y \ x^2 y^3 \ x^2 y^2 \ x^2 y \ x y^4 \ x y^3 \ x y^2 \ x y \ 1)$$

故 $\varphi_k = c^T * A(x, y)$.

再令

$$A_x(x, y) = (5x^4 \ 4x^3 \ 3x^2 \ 2x \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 4x^3 y \ 3x^2 y^2 \ 3x^2 y \ 2xy^3 \ 2xy^2 \ 2xy \ y^4 \ y^3 \ y^2 \ y \ 0)$$

$$A_y(x, y) = (0 \ 0 \ 0 \ 0 \ 0 \ 5y^4 \ 4y^3 \ 3y^2 \ 2y \ 1 \ x^4 \ 2x^3 y \ x^3 \ 3x^2 y^2 \ 2x^2 y \ x^2 \ 4xy^3 \ 3xy^2 \ 2xy \ x \ 0)$$

$$A_{xx}(x, y) = (20x^3 \ 12x^2 \ 6x \ 2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 12x^2 y \ 6xy^2 \ 6xy \ 2y^3 \ 2y^2 \ 2y \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$$

$$A_{yy}(x, y) = (0 \ 0 \ 0 \ 0 \ 0 \ 20y^3 \ 12y^2 \ 6y \ 2 \ 0 \ 0 \ 2x^3 \ 0 \ 6x^2 y \ 2x^2 \ 0 \ 12xy^2 \ 6xy \ 2x \ 0 \ 0)$$

$$A_{xy}(x, y) = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 4x^3 \ 6x^2 y \ 3x^2 \ 6xy^2 \ 4xy \ 2x \ 4y^3 \ 3y^2 \ 2y \ 1 \ 0)$$

再设其在边上的外法向导数为 n_1^k, n_2^k, n_3^k , 故 $n_1^k|_x = y_k - y_j, n_1^k|_y = -(x_k - x_j), n_2^k|_x = y_i - y_k, n_2^k|_y = -(x_i - x_k), n_3^k|_x = y_j - y_i, n_3^k|_y = -(x_j - x_i)$. 再把 $n_1^k|_x, n_1^k|_y$ 标准化, 为了不引起记号的混淆, 仍记标准化之后的法向导数为: $n_1^k|_x, n_1^k|_y$. 故有:

$$n_1^k|_x = \frac{n_1^k|_x}{\sqrt{(n_1^k|_x)^2 + (n_1^k|_y)^2}}$$

$$n_1^k|_y = \frac{n_1^k|_y}{\sqrt{(n_1^k|_x)^2 + (n_1^k|_y)^2}}$$

同理有:

$$n_2^k|_x = \frac{n_2^k|_x}{\sqrt{(n_2^k|_x)^2 + (n_2^k|_y)^2}}$$

$$n_2^k|_y = \frac{n_2^k|_y}{\sqrt{(n_2^k|_x)^2 + (n_2^k|_y)^2}}$$

$$n_3^k|_x = \frac{n_3^k|_x}{\sqrt{(n_3^k|_x)^2 + (n_3^k|_y)^2}}$$

$$n_3^k|_y = \frac{n_3^k|_y}{\sqrt{(n_3^k|_x)^2 + (n_3^k|_y)^2}}$$

则法向导数的大小为:

$$n_1^k(x, y) = n_1^k|_x * A_x(x, y) + n_1^k|_y * A_y(x, y)$$

$$n_2^k(x, y) = n_2^k|_x * A_x(x, y) + n_2^k|_y * A_y(x, y)$$

$$n_3^k(x, y) = n_3^k|_x * A_x(x, y) + n_3^k|_y * A_y(x, y)$$

再随机定义每条边上法向导数的方向, 值得注意的是, 每条边上的法向导数只有一个方向, 这里提供一种做法: 利用每个单元上的节点的全局坐标的大小以及符号函数去定义每个边上的方向. 至此, Argyris 元基函数在每个单元上的信息就用完了, 下一步解线性方程组即可把基函数的系数解出来得到每个基函数. 最终获得的线性方程组的系数矩阵如下:

$$M = \begin{pmatrix} A(x_i, y_i) \\ A(x_j, y_j) \\ A(x_k, y_k) \\ A_x(x_i, y_i) \\ A_x(x_j, y_j) \\ A_x(x_k, y_k) \\ A_y(x_i, y_i) \\ A_y(x_j, y_j) \\ A_y(x_k, y_k) \\ A_{xx}(x_i, y_i) \\ A_{xx}(x_j, y_j) \\ A_{xx}(x_k, y_k) \\ A_{yy}(x_i, y_i) \\ A_{yy}(x_j, y_j) \\ A_{yy}(x_k, y_k) \\ A_{xy}(x_i, y_i) \\ A_{xy}(x_j, y_j) \\ A_{xy}(x_k, y_k) \\ n_1^k\left(\frac{x_j+x_k}{2}, \frac{y_j+y_k}{2}\right) \\ n_2^k\left(\frac{x_i+x_k}{2}, \frac{y_i+y_k}{2}\right) \\ n_3^k\left(\frac{x_i+x_j}{2}, \frac{y_i+y_j}{2}\right) \end{pmatrix}_{21 \times 21}$$

其右端项为 $N = I_{21 \times 21}$, 通过解 21 个线性方程组求出每个基函数的系数, 由此便得到了单元上的

基函数 φ_k .

4. 数值结果

在本节, 使用 Argyris 元计算了三个数值算例并得出了它们的误差收敛阶以及 L^∞ 误差. 误差收敛阶由 ChenLong 软件包里面的 showrate 函数给出, 而所谓的 L^∞ 误差即是每个划分下的最大误差. 下面给出数值算例, 且考虑的区域都是 $[0,1] \times [0,1]$ 的矩形区域. 第一个例子的真解为:

$$u(x, y) = 256x^2(1 - x)^2y^2(1 - y)^2 \tag{4.1}$$

数值解的图像 (如图 2) 及其 L^∞ 误差图像 (如表 1), 误差收敛阶为 5.9231939。

Table 1. L^∞ error

表 1. L^∞ 误差

划分次数	1	2	3	4
argyris	0.015930107186014	2.290408560303714e-04	3.980622519030030e-06	7.003945135366021e-08

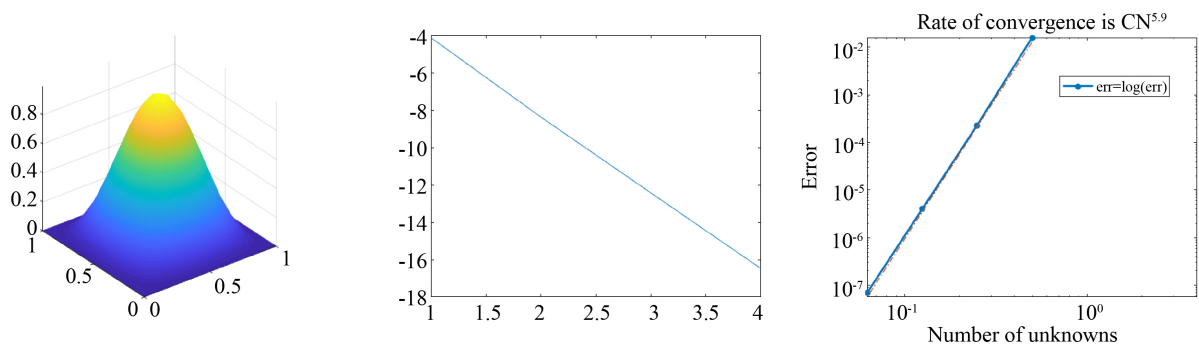


Figure 2. Argyris finite element numerical solutions and error logarithmic graph and error convergence order

图 2. Argyris 有限元的数值解以及误差对数图以及误差收敛阶

第二个数值算例的真解为:

$$u(x, y) = \sin(3\pi x^2)\sin(3\pi y^2)(x - 1)^2(y - 1)y \tag{4.2}$$

数值解的图像 (如图 3) 及其 L^∞ 误差图像 (如表 2), 误差收敛率分别为: 4.5915051 .

Table 2. L^∞ error

表 2. L^∞ 误差

划分次数	1	2	3	4
argyris	0.014590149744577	6.924157812136233e-04	3.572775205996220e-05	9.680799148634178e-07

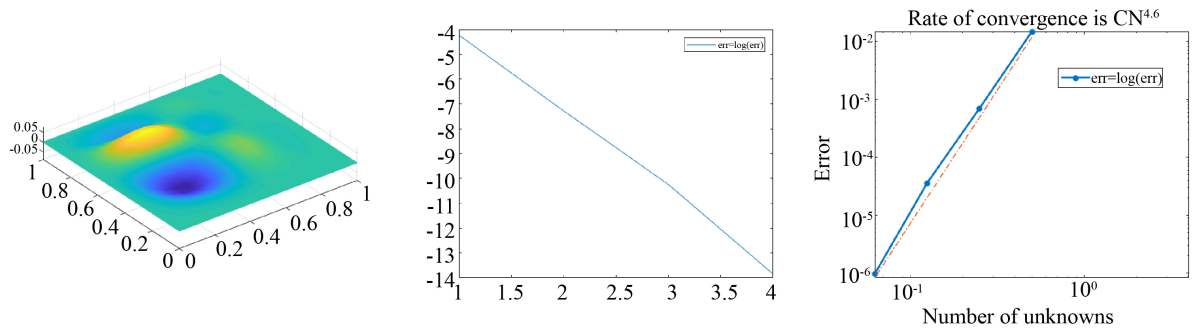


Figure 3. Argyris finite element numerical solutions and error logarithmic graph and error convergence order
图 3. Argyris 有限元的数值解以及误差对数图以及误差收敛阶

第三个数值算例的真解为:

$$u(x, y) = x^2(1 - x)^2y^2(1 - y)^2e^{x^2}e^{y^2} \tag{4.3}$$

数值解的图像 (如图 4) 及其 L^∞ 误差图像 (如表 3), 误差收敛率为: 5.2439747 .

Table 3. L^∞ error

表 3. L^∞ 误差

划分次数	1	2	3	4
<i>argyris</i>	$7.657202532339747e - 05$	$2.190043267198422e-06$	$5.608861774130617e-08$	$1.421075362514012e-09$

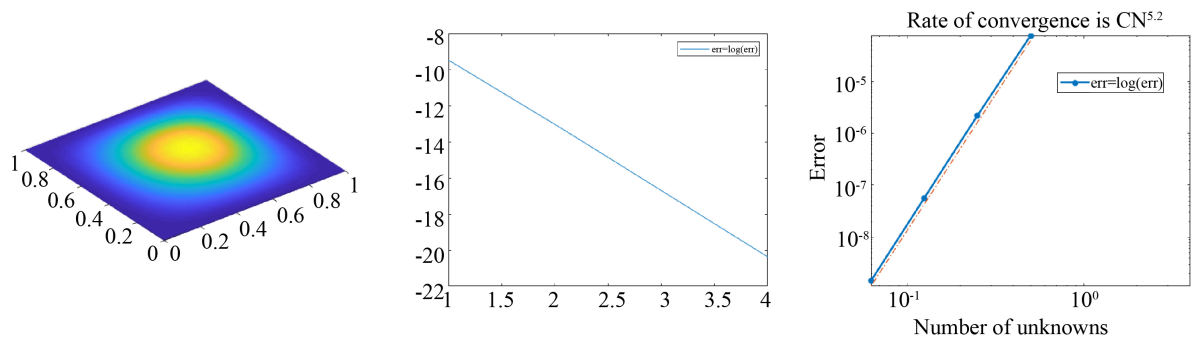


Figure 4. Argyris finite element numerical solutions and error logarithmic graph and error convergence order
图 4. Argyris 有限元的数值解以及误差对数图以及误差收敛阶

一般来说, 对于重调和方程来说, Argyris 元的收敛速度快于 Morley 元与 Bell 元, 从上面的几个例子可以看出, Argyris 元的收敛速度确实是较为快速的.

5. 总结

重调和方程在生活中应用十分广泛，本文阐述了方程的弱解并给出了重调和方程的弱形式，用 Argyris 有限元空间近似四阶问题的解空间，效果比较好，这与理论保持一致。其中较为重要的便是 Argyris 元基函数的推导，使用面积坐标的话推导出基函数一般表达式是一个非常麻烦的事情，本文舍弃这种想法，利用计算机在每个单元上求解出对应的基函数，用基函数逼近原函数，最终求解出重调和方程的解。在下一步的工作中我将会具体考虑重调和方程的有限元解法在离散液晶张量静力学模型的能量中的应用，这是一个值得探究的问题。

参考文献

- [1] 石钟慈, 王鸣. 有限元方法 [M]. 北京: 科学出版社, 2010.
- [2] 张鸿庆, 王鸣. 有限元的数学理论 [M]. 北京: 科学出版社, 1991.
- [3] 王烈衡, 许学军. 有限元方法的数学基础 [M]. 北京: 科学出版社, 2004.
- [4] Domi'Niguez, V. and Sayas, F.J. (2009) Algorithm 884: A Simple Matlab Implementation of the Argyris Element. *ACM Transactions on Mathematical Software*, **35**, 137-147.
<https://doi.org/10.1145/1377612.1377620>