



ARTICLE

Geometric Multigrid Method for Isogeometric Analysis

Houlin Yang¹, Bingquan Zuo^{1,2,*}, Zhipeng Wei^{1,2}, Huixin Luo^{1,2} and Jianguo Fei^{1,2}

¹Key Laboratory of Metallurgical Equipment and Control Technology Ministry of Education, Wuhan University of Science and Technology, Wuhan, 430081, China

²Hubei Key Laboratory of Mechanical Transmission and Manufacturing Engineering, Wuhan University of Science and Technology, Wuhan, 430081, China

*Corresponding Author: Bingquan Zuo. Email: zuobingquan@gmail.com

Received: 02 October 2020 Accepted: 10 December 2020

ABSTRACT

The isogeometric analysis method (IGA) is a new type of numerical method solving partial differential equations. Compared with the traditional finite element method, IGA based on geometric spline can keep the model consistency between geometry and analysis, and provide higher precision with less freedom. However, huge stiffness matrix from the subdivision progress still leads to the solution efficiency problems. This paper presents a multigrid method based on geometric multigrid (GMG) to solve the matrix system of IGA. This method extracts the required computational data for multigrid method from the IGA process, which also can be used to improve the traditional algebraic multigrid method (AGM). Based on this, a full multigrid method (FMG) based on GMG is proposed. In order to verify the validity and reliability of these methods, this paper did some test on Poisson's equation and Reynolds' equation and compared the methods on different subdivision methods, different grid degrees of freedom, different cyclic structure degrees, and studied the convergence rate under different subdivision strategies. The results show that the proposed method is superior to the conventional algebraic multigrid method, and for the standard relaxed V-cycle iteration, the method still has a convergence speed independent of the grid size at the same degrees.

KEYWORDS

Isogeometric method; geometric multigrid method; reflecting matrix; subdivision strategy

1 Introduction

In recent years, as a numerical method for solving partial differential equations (PDEs), isogeometric analysis [1] (IGA) has been widely concerned and studied. The basic idea of IGA: Based on the isoparametric element theory, geometric splines can not only describe the exact geometric modeling but also the solution space of numerical methods, which can unify the geometric model and the analytical model. Therefore, the spline information in CAD model is directly used as the initial mesh in IGA. Since it was proposed, most of the research on IGA has focused on parametric modeling and physical spatial discretization. However, when dealing with large-scale



problems, it is still a problem to solve the system of linear equations which are obtained from the geometric discretization. The reason is that the direct solution is inefficient and usually requires targeted development of faster and more efficient iterative solver.

The multigrid method [2,3] (MG) is often used as an efficient iterative algorithm to solve large-scale linear equations. The main idea is to eliminate high-frequency errors through fine mesh smoothing, coarse grid correction to eliminate low frequency errors, and an iterative strategy for fine grid and coarse grid cycles. According to the different mapping matrix generation methods, the MG can be divided into two types: geometric multigrid method (GMG) and algebraic multigrid method (AMG). Usually the AMG is more widely used than the other because it obtains different level meshes in an algebraic way instead of constructing them, which may be complicated in some cases [4]. However, the IGA directly generates the hierarchical mesh through the refinement method (p -refinement, h -refinement, k -refinement), which encourages us to implement the GMG in IGA.

It seems natural to extend these methods to IGA, and several remarkable works have been made in recent years. MG for IGA based on classical concepts have been considered in [5,6], and a classical multilevel method in [7]. An approach to local refinement in IGA based on a full multigrid method (FMG) can be found in [8]. The related studies mainly include Gahalaut et al. [9] applied multigrid to IGA, and did not propose the concept of mapping matrix; Hofreither et al. [10] proposed a robust error approximation. These studies above mainly focus on the implement of MG in IGA. And the efficiency in the adaptation process has been studied. Other research mainly includes Guo [11] to improve efficiency through parallel computing and multigrid method, Liu et al. [12] combined conjugate gradient method with multi-grid method, and Chen et al. [13] based on geometric analysis such as least squares coupling multigrid method. Recently, in [14] some progress has been made by using a Richardson method preconditioned with the mass matrix as a smoother (mass-Richardson smoother). Similar techniques have been used to construct symbol-based multigrid methods for IGA, see [15,16]. However, all the above research uses the coordinate mapping relationship in the process of geometric mesh construction to modify the traditional algebraic multigrid iterative process but ignore the stiffness information discretely obtained during isogeometric subdivision. In addition, the extension matrix transposition is directly used in the construction of restriction matrix, which will lead to the confusion and imbalance of mapping relationship.

Based on previous generations, this paper uses the geometric information discretized by isogeometric subdivision to couple the MG with IGA and its application. Especially for the numerical algorithms such as IGA, we use geometric method to optimize them in the shortcomings of AMG, and it has integrated into the process of FMG.

2 Isogeometric Analysis

2.1 *B-Splines and NURBS Basis Functions*

Like FEM, IGA also have shape functions, not the Lagrange interpolation functions but the spline basis functions. As the two common spline functions used in CAD, B-spline and NURBS

are briefly introduced below. With a given node vector, the B-spline basis function [17] can be recursively defined as:

$$N_{i,0}(\xi) = \begin{cases} 1 & \xi_i \leq \xi \leq \xi_{i+1} \\ 0 & \text{others} \end{cases} \tag{1}$$

$$N_{i,k}(\xi) = \frac{\xi - \xi_i}{\xi_{i+k} - \xi_i} N_{i,k-1}(\xi) + \frac{\xi_{i+k+1} - \xi}{\xi_{i+k+1} - \xi_{i+1}} N_{i+1,k-1}(\xi)$$

where the one-dimensional knot vector $\xi = [\xi_0, \xi_1, \dots, \xi_{n+k+1}]$ is a set of non-decreasing sequences between 0 and 1. k is the degree of the basis function ($p = k + 1$ is the order of the basis function). n is the number of basic functions, which is numerically equal to the number of control points.

If the basis function of B-spline is multiplied by an appropriate weight w , the non-uniform rational B-spline (NURBS) can be introduced. Its expression is:

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi)N_{j,q}(\eta)w_{i,j}}{\sum_{k=1}^n \sum_{l=1}^m N_{k,p}(\xi)N_{l,q}(\eta)w_{k,l}} \tag{2}$$

2.2 The Framework of Multigrid Isogeometric Analysis

FEA and IGA have the same theoretical basis (i.e., the weak form of partial differential equation). Take the common elliptic partial differential equation as an example:

$$-\nabla(k\nabla u) = f \quad \text{in } \Omega$$

$$u = g \quad \text{on } \Gamma_D \tag{3}$$

$$\frac{\partial u}{\partial n} = h \quad \text{on } \Gamma_N$$

where $\Omega \subset R^2$ is the open and bounded Lipschitz domain and the boundary is $\partial\Omega$, $\Gamma_D \cup \Gamma_N = \Gamma \equiv \partial\Omega$, $\Gamma_D \cap \Gamma_N = \emptyset$; Given source term $f: \Omega \subset R^2$; Symmetric positive definite matrix $k: \Omega \subset R^{2 \times 2}$; Dirichlet boundary value $g: \partial\Omega \subset \Gamma_D$; Neumann boundary value $h: \partial\Omega \subset \Gamma_N$.

It is assumed that the physical domain Ω is represented by a parameter domain Ω_0 mapping $F: \Omega_0 \rightarrow \Omega$, $F(\xi, \eta) = \begin{pmatrix} x \\ y \end{pmatrix}$. Using the control point c_{ij} , F can be defined by NURBS as:

$$F(\xi, \eta) = \sum_{i=1}^m \sum_{j=1}^n R_{ij}(\xi, \eta)c_{ij} \tag{4}$$

The weak form of the differential equation is obtained by multiplying with the trial function and integrating in the domain Ω . Let $a(u, v) = \iint_{\Omega} (\nabla(k\nabla u)) \cdot v \, dx \, dy$ and $\langle f, v \rangle = \iint_{\Omega} f \cdot v \, dx \, dy$, then the equation becomes $a(p, v) = \langle f, v \rangle$. According to Green's method, the linear equations are obtained by discrete assembly:

$$Au = f \tag{5}$$

The detailed process can be seen in [18,19].

2.3 Isogeometric Subdivision

Based on the requirements of light weight, modern CAD systems follow the principle of sufficient enough when characterizing models [20]. So, the spline order in CAD is low usually, just meeting the basic characterization needs, and the node of the parameter domain is relatively sparse. But the order of the shape function (spline basis function) and the grid size (knot vector interpolation space scale) have a great impact on the analysis accuracy. As a result, the first step in IGA is subdivision, which works like the refinement in FEM, but with a different method. Therefore, it is necessary to introduce the subdivision process before constructing the isogeometric multiple meshes.

There are mainly three subdivision methods [21] in IGA, they are p -refinement, h -refinement and k -refinement. All these subdivision methods can keep the original shape the same. The changes are mainly at the spline order and knot vector. In particular, the p -refinement retains the original parameter domain grid and only increases the order of the basis function; the h -refinement does not change the basis function order but reduces the grid size to construct a finer parameter domain grid. As a combination of p -refinement and h -refinement, k -refinement increase the order of basis function firstly, and then refines the knot vector.

Taken the one-dimensional h -refinement as an example for introduction. Inserting the knot $\mathbf{t} = [t_1, t_2, \dots, t_m] \subset [\tau_k, \tau_{n+1}]$ into the knot vector $T = [\tau_0, \tau_1, \dots, \tau_{n+k+1}]$ at one time can get the new knot vector $\bar{T} = [\bar{\tau}_0, \bar{\tau}_1, \dots, \bar{\tau}_{n+k+1+m}]$, then the corresponding coordinate calculation formula of the new spline control point can be obtained [22]:

$$\bar{D}_j = D_{i,j}^{[r+1]} = \frac{(\bar{\tau}_{j+k-r} - \tau_i)}{\tau_{i+k-r} - \tau_i} D_{i,j}^{[r]} + \frac{(\tau_{i+k-r} - \bar{\tau}_{j+k-r})}{\tau_{i+k-r} - \tau_i} D_{i-1,j}^{[r]} \begin{cases} \tau_i \leq \bar{\tau}_j < \tau_{i+1} \\ r = 1, 2, \dots, k \end{cases} \quad (6)$$

where $D_{i,j}^{[1]} = D_i$ is the coordinate of the original spline control point.

If $k = 3$ (the cubic B-spline), Formula (6) can be written as follow:

$$\bar{D}_j = D_{i,j}^{[4]} = a_i D_i + a_{i-1} D_{i-1} + a_{i-2} D_{i-2} + a_{i-3} D_{i-3} \tau_i \leq \bar{\tau}_j < \tau_{i+1} \quad (7)$$

where, the coefficients can be calculated as follows:

$$a_i = \frac{(\bar{\tau}_{j+1} - \tau_i)(\bar{\tau}_{j+2} - \tau_i)(\bar{\tau}_{j+3} - \tau_i)}{(\tau_{i+1} - \tau_i)(\tau_{i+2} - \tau_i)(\tau_{i+3} - \tau_i)}$$

$$a_{i-1} = \frac{(\bar{\tau}_{j+1} - \tau_i)(\bar{\tau}_{j+2} - \tau_i)(\tau_{i+3} - \tau_{j+3})}{(\tau_{i+1} - \tau_i)(\tau_{i+2} - \tau_i)(\tau_{i+3} - \tau_i)} + \frac{(\bar{\tau}_{j+1} - \tau_i)(\tau_{i+2} - \bar{\tau}_{j+2})(\bar{\tau}_{j+3} - \tau_{i-1})}{(\tau_{i+1} - \tau_i)(\tau_{i+2} - \tau_i)(\tau_{i+2} - \tau_{i-1})}$$

$$+ \frac{(\tau_{i+1} - \bar{\tau}_{j+1})(\bar{\tau}_{j+2} - \tau_{i-1})(\bar{\tau}_{j+3} - \tau_{i-1})}{(\tau_{i+1} - \tau_i)(\tau_{i+1} - \tau_{i-1})(\tau_{i+2} - \tau_{i-1})}$$

$$a_{i-2} = \frac{(\bar{\tau}_{j+1} - \tau_i)(\tau_{i+2} - \bar{\tau}_{j+2})(\tau_{i+2} - \bar{\tau}_{j+3})}{(\tau_{i+1} - \tau_i)(\tau_{i+2} - \tau_i)(\tau_{i+2} - \tau_{i-1})} + \frac{(\tau_{i+1} - \bar{\tau}_{j+1})(\bar{\tau}_{j+2} - \tau_{i-1})(\tau_{i+2} - \bar{\tau}_{j+3})}{(\tau_{i+1} - \tau_i)(\tau_{i+1} - \tau_{i-1})(\tau_{i+2} - \tau_{i-1})}$$

$$+ \frac{(\tau_{i+1} - \bar{\tau}_{j+1})(\tau_{i+1} - \bar{\tau}_{j+2})(\bar{\tau}_{j+3} - \tau_{i-2})}{(\tau_{i+1} - \tau_i)(\tau_{i+1} - \tau_{i-1})(\tau_{i+1} - \tau_{i-2})}$$

$$a_{i-3} = \frac{(\bar{\tau}_{i+1} - t_{j+1})(\bar{\tau}_{i+1} - t_{j+2})(\bar{\tau}_{i+1} - t_{j+3})}{(\tau_{i+1} - \tau_i)(\tau_{i+1} - \tau_{i-1})(\tau_{i+1} - \tau_{i-2})}$$

During p -refinement process, the degree of the spline curve is raised once, and the shape and continuity of the curve remain unchanged. Cohen [23] offered a solution to calculate the new control points:

$$\bar{D}_j = \frac{1}{k} \sum_{i=0}^n D_i a_i^k(j) \quad j = 0, 1, \dots, n+l \tag{8}$$

where $a_i^k(j)$ recursively as follows:

$$\alpha_i^1 = a_i^1 = \begin{cases} 1, & \tau_i \leq \bar{\tau}_j < \tau_{i+1} \\ 0, & \text{others} \end{cases}$$

$$a_i^k(j) = \frac{\bar{\tau}_{j+k+1} - \tau_i}{\tau_{i+k} - \tau_i} a_i^{k-1}(j) + \frac{\tau_{i+k+1} - \bar{\tau}_{j+k+1}}{\tau_{i+k+1} - \tau_{i+1}} a_{i+1}^{k-1}(j) + \alpha_i^k(j)$$

$$\alpha_i^k(j) = \frac{\bar{\tau}_{j+k} - \tau_i}{\tau_{i+k} - \tau_i} \alpha_i^{k-1}(j) + \frac{\tau_{i+k+1} - \bar{\tau}_{j+k}}{\tau_{i+k+1} - \tau_{i+1}} \alpha_{i+1}^{k-1}(j)$$

K -refinement is a combination of h -refinement and p -refinement. In order to ensure the continuity of elements, k -refinement is often defined as “first upgrade, then insert nodes” in IGA. This means, for the B-spline curve of order p , the order is elevated to q by p -refinement firstly, and then knots inserting is taken by h -refinement.

3 Multigrid Method

For FEM, most of MG method uses the algebraic multigrid method to construct interpolation operator and coarse the initial grid to get different hierarchical grid. In IGA, the geometric model is constructed from coarse to fine. There are different levels of grid naturally, and the mapping matrix between each level can be obtained from the subdivision process. In addition, each level has its own stiffness matrix and righthand vector. If this advantage is used to combine the multigrid method with IGA, the efficiency of solution would be improved.

3.1 Multigrid Mapping Matrix

3.1.1 Classical Algebraic Multigrid Mapping Matrix

In AMG, the main work is focused on constructing the interpolation matrix by coarsen the fine mesh, coarsening refined net Ω_m to get rough grid Ω_{m-1} ($m = 2, \dots, n$). And the coarse grid Ω_{m-1} is a subset of the fine grid defined with C_m , using F_m to define the remaining set $\Omega_m - C_m$, when $|a_m^{i,j}| \geq \theta \cdot \max_{k \neq i} |a_m^{i,k}|$, $0 < \theta \leq 1$ the point i is strongly connected to point j . Let S_m^i define all the strong connection points of point i , then the strong connection rough set is $C_m^i = S_m^i \cap C_m$ and the interpolation matrix I_{m-1}^m refers to the interpolation formula given by Chang et al. [24].

For MG, the mapping matrix is important. And its generation can affect the computational efficiency of MG directly. So, it is necessary to compare the generation rate of mapping matrix under different degrees of freedom. the time-consuming of mapping matrix generation (using Chang’s method) in Tab. 1 shows that the time of generating the mapping matrix of the algebra multigrid rises sharply with the increase of degrees of freedom. Obviously, a faster constructing method for mapping matrix will be helpful for the implement of MG.

Table 1: The generation rate of mapping matrix

n	100	324	1156	4356	16900
Algebraic multigrid method I_{m-1}^m (s)	0.169837	0.888387	4.669562	37.063700	569.107235

3.1.2 Suitable for Isogeometric Analysis of Multigrid Mapping Matrix

As discussed in Chapter 2.3, in the subdivision of IGA can generate the mapping relationship between new and old control points D_m and D_n . It is assumed that the relationship between \bar{u} (on fine grid) and u (coarse grid) also satisfies this relationship. According to the formula (7), the extension matrix from coarse to fine corresponding to h -refinement can be expressed as:

$$\bar{D}_m = R_{m \times n}^h D_n \quad (9)$$

in the formula, $R_{m \times n}^h = [r_0 \ \dots \ r_n]_{m \times n}^T$

The corresponding linear relation is that the non-zero element in row vector r is numbered as the old control point number related to the new control point, which is usually a continuous interval segment.

P -refinement extended matrix can be obtained according to the calculation of Formula (8):

$$R^p = \frac{1}{k+1} \begin{bmatrix} a_0^k(0) & a_1^k(0) & \dots & a_n^k(0) \\ a_0^k(1) & a_1^k(1) & \dots & a_n^k(1) \\ \vdots & \vdots & \dots & \vdots \\ a_0^k(n+l) & a_1^k(n+l) & \dots & a_n^k(n+l) \end{bmatrix} \quad (10)$$

Therefore, the extended matrix of k -refinement is:

$$R^k = R^h R^p \quad (11)$$

In theory, the restriction matrix P from fine grid to coarse grid is the generalized inverse matrix of extension matrix R , but it is difficult to calculate the inverse matrix of extension matrix R , and there are stability problems and limitations in inversion calculation. Therefore, in this paper, transposed matrix is chosen as generalized inverse, $P = R^T$. But in fact, the mapping matrix P obtained by transpose cannot accurately represent the mapping relationship from coarse grid to fine grid, so the mapping matrix α_i is modified by parameter P here:

$$\alpha_i P(i) D_1 = D_2(i) \quad (i = 0, 1, \dots, m) \quad (12)$$

$$\text{in the formula, } P = \begin{bmatrix} p_0 \\ \vdots \\ p_m \end{bmatrix}_{m \times n}, \quad D_1 = \begin{bmatrix} d_0^1 \\ \vdots \\ d_n^1 \end{bmatrix}_n, \quad D_2 = \begin{bmatrix} d_0^2 \\ \vdots \\ d_m^2 \end{bmatrix}_m.$$

Since the control point D is a coordinate vector, the difference between the positive and negative coordinates will affect the effect of the correction. To ensure the positive correction of the parameter α_i , the analysis model can be translated to the first quadrant to ensure that the control point coordinates are greater than zero.

In IGA, the mapping matrix obtained by different refinement strategies are shown in [Tab. 2](#). The detailed process can be referred to [\[25\]](#).

Table 2: Mapping matrix

	<i>h</i> -refinement	<i>p</i> -refinement	<i>k</i> -refinement
Continuation	R^h	R^p	R^k
Restrictions	P^h	P^p	P^k
Amended restrictions	\bar{P}^h	\bar{P}^p	\bar{P}^k

Based on one-dimensional subdivision, two-dimensional subdivision only needs to subdivide in *u*, *v* directions at the same time to get mapping matrix $R_{u,i}$ and $R_{v,j}$, and then get the mapping matrix $R_{\xi,\eta}$ by global numbering of coarse and fine meshes. *n*, *n'* is the number of control points of the grid. $R_{\xi,\eta}$ can be represented by [Eq. \(13\)](#).

$$R_{\xi,\eta} = R_{u,i}R_{v,j} \begin{cases} \xi = n'(v - 1) + u \\ \eta = n(j - 1) + i \end{cases} \tag{13}$$

A time-consuming compare of mapping matrix generation between this method and AMG is given in [Tab. 3](#). With the freedom increase, more time can be saved in pre-processing stage of MG.

Table 3: The generation rate of mapping matrix

n	100	324	1156	4356	16900
Algebraic multiple method I_{m-1}^m	0.169837	0.888387	4.669562	37.063700	569.107235
Mapping matrix $R_{\xi,\eta}$	0.003802	0.009883	0.039448	0.199938	1.603531

3.2 Multigrid Algorithm

3.2.1 Geometric Modification of Algebraic Multigrid Method

The multigrid method is divided into initial period and calculation stage. In the initial period, GMG is based on the subdivision of the solution domain to get the grid of different scales. AMG only uses the information of coefficient matrix in algebraic equations to construct multigrid, and the method is consistent in the iterative solution stage. In order to improve the implement of AMG in IGA, geometric information will be introduced to AMG, what is called Geometric modification of algebraic multigrid method (GMG).

1. Initial period

- 1). The nested grid layer $\Omega_1 \subset \Omega_2 \subset \dots \subset \Omega_m$ is constructed by subdivision, and the extension matrix R_{m-1}^m from layer *m* - 1 grid to layer *m* is stored. The mapping matrix P_m^{m-1} from layer *m* grid to layer *m* - 1 is modified according to [Eq. \(12\)](#).
- 2). Multigrid $A_{GMG}^1, A_{GMG}^2, \dots, A_{GMG}^m$ is obtained by discretizing PDEs with the method of IGA, so the Right items f^1, f^2, \dots, f^m .

2. Calculation stage

Algorithm 1: $\bar{u}^m \leftarrow GMG(A^m, f^m, u^m, d)$

1. If $m = 1$, solve directly or precisely iteratively on Ω_1 , otherwise proceed to the next step.
 2. Pre-relaxation: $\bar{u}^m \leftarrow S^{\lambda_1}(A^m, f^m, u^m)$
 3. Coarse-grid correction:
 - (a) Algebraic residuals $r^{m-1} = P_m^{m-1}(f^m - A_{AMG}^m \bar{u}^m)$;
Geometric residual $r^{m-1} = P_m^{m-1}(f^m - A_{GMG}^m \bar{u}^m)$
 - (b) Take the minimum residual r^{m-1} from (a)
 - (c) $e^{m-1} \leftarrow GMG(A^{m-1}, r^{m-1}, 0, d)$
 - (d) Correction of residuals $u^m = \bar{u}^m + R_{m-1}^m e^{m-1}$
 4. Post relaxation: $\bar{u}^m \leftarrow S^{\lambda_2}(A^m, f^m, u^m)$
-

The iterative process of multigrid method (GMG) is given above. For a given parameter space $\Omega_1 \subset (0, 1)^2$, a rough grid for IGA can be built. Though refinement, multilevel grids can be constructed., so the limit matrix of layer m to layer $m - 1$ P_m^{m-1} and the extension matrix from $m - 1$ to m R_{m-1}^m . Parameter d in GAMG determines how to access each layer, as shown in Fig. 1, $d = 1$ is a V-type cycle and $d = 2D$ is a W-type cycle. The traditional AMG, the left term of the coarse grid is calculated using the mapping method, $A_{AMG}^{m-1} = P_m^{m-1} A_{AMG}^m R_{m-1}^m$. In GMG, the corresponding A^m of each geometric level can be calculated directly. That means, GMG in this paper introduces geometric information as a more selection of residual and selects the smaller residual as the next relaxation iteration by comparing the residual between geometric grid and algebra grid.

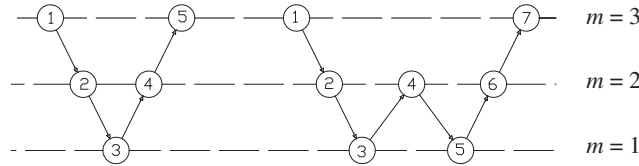


Figure 1: Multi-grid cycle

As for the iteration of relaxation operator $S^{v_1}(A^m, f^m, u^m)$, $v_1 = v_2 = 2$ represents the number of times of relaxation. In order to get the iteration structure, A^m is divided into $A^m = M^m - N^m$, and its iterative format can be written as:

$$\bar{u}^m = (I - M^{m-1} A^m) u^m + M^{m-1} f^m \quad (14)$$

M^m is selected to obtain various iterative methods for relaxation. Jacobi, Gauss-Seidel and SOR iterations are commonly used. For rough grid correction, both the residual $r^m = (f^m - A^m \bar{u}^m)$ and error $e^m = \bar{u}^m - u^m$ satisfy the residual equation $A^m e^m = r^m$, so e^{m-1} can be solved by the same relaxation operator S .

3.2.2 Full Multigrid Method

As known to all, A^m and f^m are discretized on each level grid in IGA, in order to make full use of these information, a full multigrid method (FMG) method suitable for IGA is established.

In the loop process of FMG, both AMG and GMG can be selected as solving algorithm on each layer, hence the accuracy of each layer can be guaranteed. Besides, and with the help of the extension matrix R_{m-1}^m , the results can be interpolation to the previous layer as the initial solution which can be useful to reduce the iteration steps. As shown in Fig. 2, the FMG process is as follows:

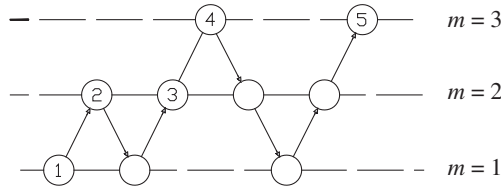


Figure 2: Full multigrid method

Algorithm 2: FMG algorithm

1. Using exact solution in the thickest grid $u^1 = A^1 / f^1$
 2. The initial solution of the second layer grid $u^2 = R_1^2 u^1$
 3. For the level m , the initial solution of the third layer $u^m = R_{m-1}^m u^{m-1}$
 4. Calculate $\bar{u}^m \leftarrow AMG(A^m, f^m, u^m)$ or $\bar{u}^m \leftarrow GMG(A^m, f^m, u^m)$
 5. If m is the final level “exit”, else go to Step 3.
-

4 Numerical Examples

In order to verify the validity of the algorithm, this paper has carried out the verification on the Poisson equation and Reynolds equation.

According to the relationship between residual norm and error norm: $\|r_k\| = \|Ae_k\| \leq \|A\| \|e_k\|$, the convergence of the residual and the error is consistent.

The relative error can be expressed as follows:

$$\|r_k\| / \|r_0\| = error \tag{15}$$

The convergence rate is defined as:

$$ration = error_{iter} / error_{iter-1} \tag{16}$$

The algorithm is written in OCTAVE, the computer processor is Core i5-4570, with 3.20 GHz frequency. We contrast the convergence and calculation time of different subdivision strategies. The unified cycle format is V cycle, the number of grid layers $m = 3$, and the subdivision mode of coarse grid to fine grid is $p_1 \ h_1 \ p_2 \ h_2 \ p_3 \ h_3$, corresponding to the 6-bit coding (such as 102026) shown in the figure below, respectively. Select the corresponding subdivision strategy from Tab. 5 (Reynolds) and Tab. 8 (Poisson), and the relaxation algorithm selects Gauss–Seidel iteration, and the number of internal iterations is controlled to be $v_1 = v_2 = 2$.

4.1 Reynolds Equation Model

The piston-cylinder liner is used as a lubrication system, and the oil film pressure is derived based on the Reynolds equation. Fig. 3a shows the piston liner system, and the area of lubricating

film is shown in Fig. 3b. Based on the Navier–Stokes equations and the mass continuity equation, the Reynolds equation [26,27] can be established as follows:

$$\frac{\partial}{\partial x} \left(\frac{\rho h^3}{12\eta} \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\rho h^3}{12\eta} \frac{\partial p}{\partial y} \right) = \frac{\partial}{\partial x} \frac{(u_0 - u_h) \rho h}{2} + \frac{\partial}{\partial y} \frac{(v_0 - v_h) \rho h}{2} + \frac{\partial (\rho h)}{\partial t} \quad (17)$$

where ρ is the lubricant density, p is the oil film pressure, η is the dynamic viscosity, u_0 and u_h are the oil film circumferential speeds on the piston and the cylinder liner surface, v_0 and v_h are the film shaft speed on the piston and the cylinder liner surface, h is the film thickness, t is time.

Since the lubrication area is symmetrical relative to the plane $\phi = 0$, only the right half needs to be considered. In general, the piston moves periodically, and the values of density and viscosity will change with the change of load value. However, in order to simplify the numerical model, the values of density and viscosity are taken as constant. The calculation parameters of oil film pressure are shown in Tab. 4.

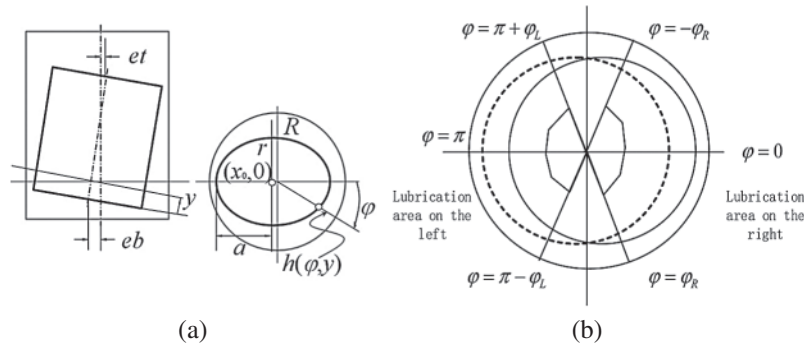


Figure 3: Piston-cylinder liner lubrication system (a) piston-cylinder system (b) oil film lubrication area

Table 4: Calculate parameters

Parameter/unit	Explanation	Value
r/m	Piston skirt radius	2.1725×10^{-2}
R/m	Cylinder radius	2.175×10^{-2}
L_{SK}/m	Piston skirt length	2.25×10^{-2}
$\eta/Pa \cdot s$	Lubricant viscosity	1.295×10^{-2}
e_t/m	Eccentricity of upper end of piston skirt	7.8565×10^{-2}
e_b/m	Eccentricity of lower end of piston skirt	-2.4924×10^{-5}
$d_{et}/m \cdot s^{-1}$	Derivative of e_t to time	-2.435×10^{-3}
$d_{eb}/m \cdot s^{-1}$	Derivative of e_b to time	-8.4461×10^{-3}
$v/m \cdot s^{-1}$	Reciprocating speed of piston	-10.6543
C_b/m	Vertical distance from piston pin to piston crown	6×10^{-3}
C_c/m	Distance from piston pin to piston centerline	0

The motion parameters of the piston-cylinder liner model have been given above, and the relevant detailed derivation process is referred to [28,29].

Table 5: Number of control points from MG based on different refinement strategies

p	h							
	0	1	2	3	4	5	6	7
0	4	9	25	81	289	1089	4225	16641
1	9	16	36	100	324	1156	4356	16900
2	16	25	49	121	361	1225	4489	17161
3	25	36	81	144	400	1296	4624	17424

When the Reynolds equation is discretized, the knot vector of the initial parameter domain is $v = u = [0 \ 0 \ 1 \ 1]$. In the initial state, given different p , h , k -refinement strategies will get different levels of grid control points.

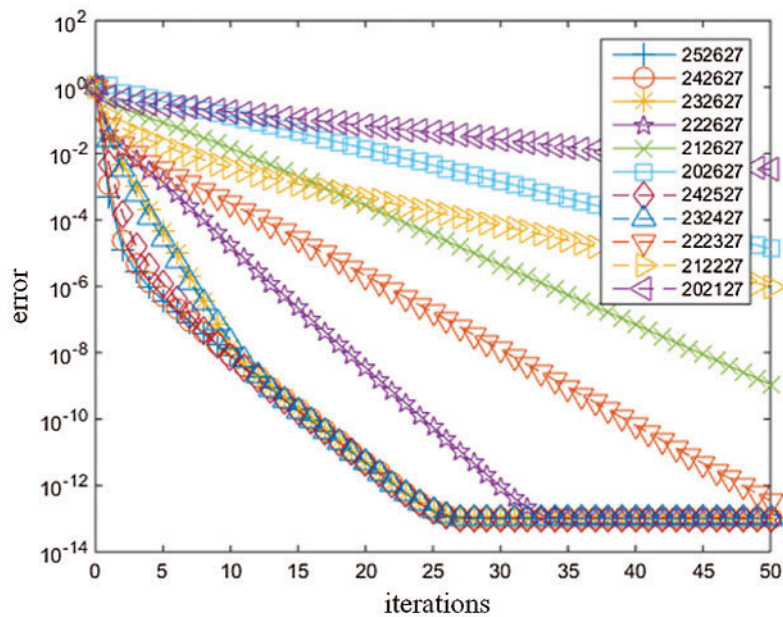
In the case of h -refinement, n is the thinnest grid, and the appropriate coarse grid is selected to make it have the fastest convergence rate. It can be seen from Tab. 6 that when k is constant, the grid size is independent of the convergence rate. Since the Reynolds equation is an elliptic equation of second order, so it has the fastest convergence rate at $k = 2$. When k increases, the convergence rate decreases sharply.

Table 6: Convergence rate

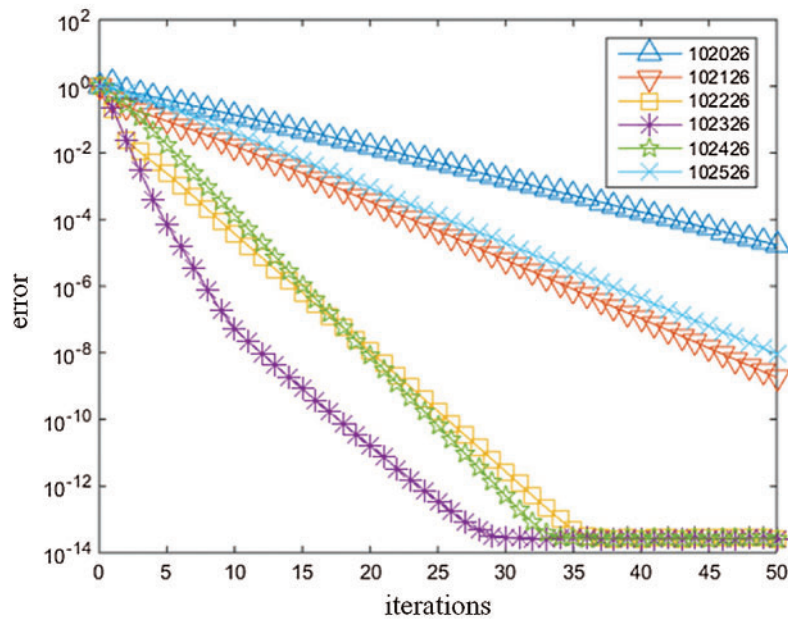
$k = 1 \text{ iter} = 8$			$k = 2 \text{ iter} = 7$			$k = 3 \text{ iter} = 20$			$k = 4 \text{ iter} = 70$		
n	L_2 -error	Ration	n	L_2 -error	Ration	n	L_2 -error	Ration	n	L_2 -error	Ration
81	$5.7682e^{-9}$	0.0893	100	$1.4610e^{-11}$	0.0420	121	$5.4724e^{-10}$	0.4866	144	$2.7144e^{-09}$	0.8618
289	$2.6945e^{-10}$	0.0595	324	$1.2098e^{-11}$	0.0438	361	$9.3986e^{-11}$	0.4863	400	$1.8839e^{-10}$	0.8599
1089	$1.6173e^{-10}$	0.0553	1156	$1.1407e^{-11}$	0.0467	1225	$2.3928e^{-11}$	0.4867	1296	$4.0223e^{-11}$	0.8405
4225	$6.3934e^{-11}$	0.0536	4356	$4.6670e^{-12}$	0.0453	4489	$1.0456e^{-11}$	0.4835	4624	$1.8401e^{-11}$	0.8400
16641	$2.9877e^{-11}$	0.0534	16900	$1.9957e^{-12}$	0.0444	17161	$4.9974e^{-12}$	0.4830	17424	$9.0104e^{-12}$	0.8402

Fig. 4 only shows the comparison between h -refinement and k -refinement in order to explore the influence of different refinement strategies on the convergence rate. The broken lines in the figure indicate the number of subdivisions of each layer of the grid. For example, 252627 represents the second-order fifth-level grids, the second-order sixth-level grids and the second-order seventh-level grids. The figure below is similar. The traditional AMG and GMG are used to compare the convergence of different degrees of freedom and spline degree k . The convergence rate of AMG and GMG iterations are shown in Fig. 5. Tab. 7 shows the CPU calculation time of the AMG algorithm and the GMG algorithm. It can be seen that: when the number k is constant, the convergence rate of AMG and GMG is almost the same under different grid sizes. Due to the different ways in which the coarse grid matrix is generated, the non-zero elements of the matrix A are different. As shown in Fig. 6, nz is the number of non-zero elements. The band length in the GMG matrix A is reduced globally and locally, and the number of non-zero elements is less than that in the AMG matrix. Therefore, when the degree of freedom is large, the

GMG of the natural discrete matrix A can save more time and computing resources. Meanwhile, as the right items of different levels of are also produced in the discrete process, FMG can be used to calculate the original right items in the current layer. It can be seen from Fig. 7 that the full multiple cycles have superior initial values in the iterative calculation process, and the desired error accuracy can be obtained in less iterations.



(a)



(b)

Figure 4: Relative error of refinement strategy (a) H -refinement relative error (b) K -refinement relative error

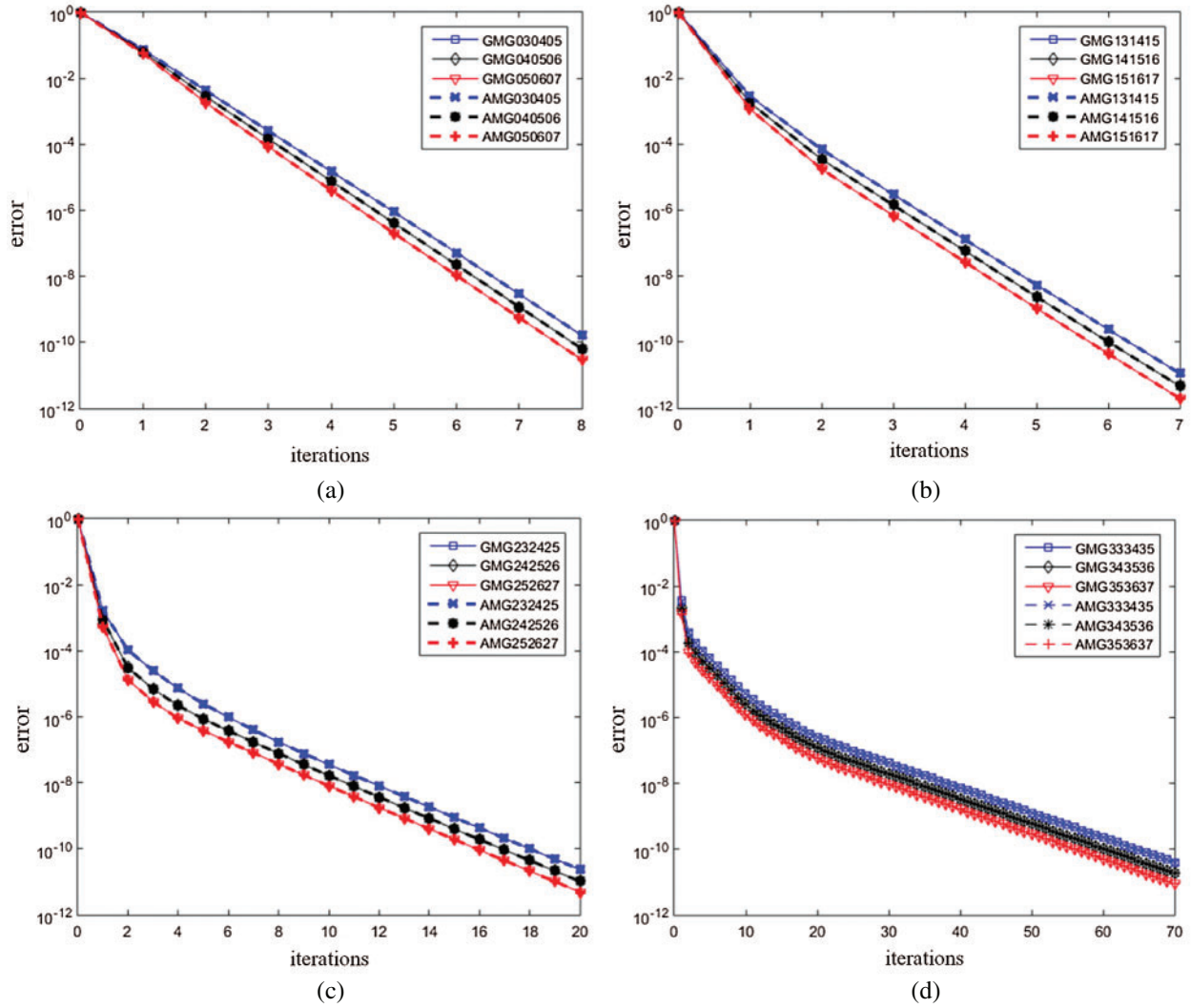


Figure 5: Relative error between AMG and GMG (a) $k = 1$ (b) $k = 2$ (c) $k = 3$ (d) $k = 4$

Table 7: CPU calculation time of AMG and GMG

n	Computation time/s		n	Computation time/s	
	AMG	GMG		AMG	GMG
1089	0.01429	0.01381	1156	0.02011	0.02132
4225	0.04896	0.04004	4356	0.09043	0.08749
16641	0.19569	0.19189	16900	0.47305	0.45915
1225	0.06512	0.07268	1296	0.54113	0.56633
4489	0.35961	0.35194	4624	3.43262	3.32487
17161	2.00979	1.89226	17424	14.15859	13.95189

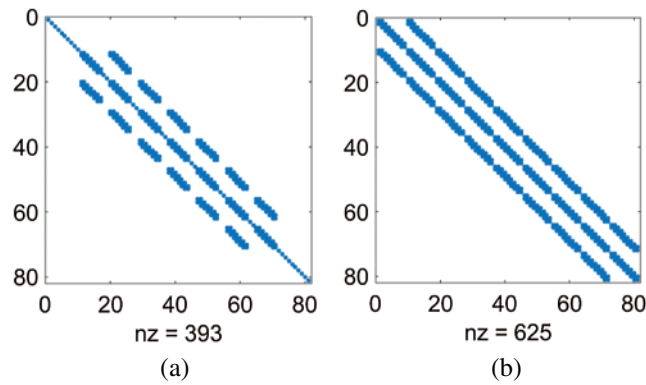


Figure 6: Coarse grid matrix A (a) GMG (b) AMG

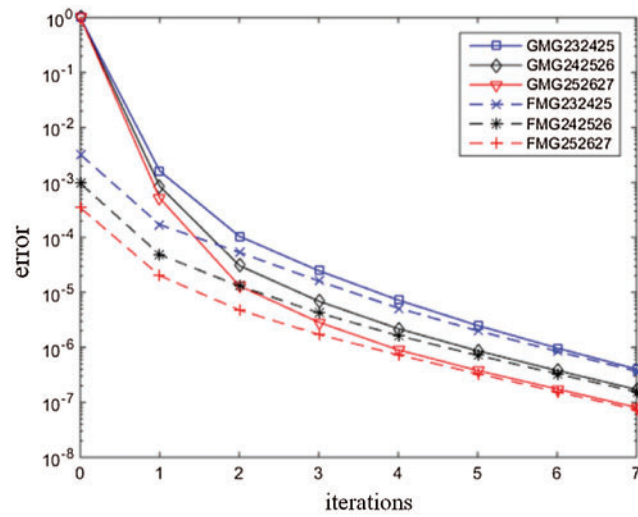


Figure 7: Relative error between FMG and GMG

4.2 Poisson Equation Model

The equation of the two-dimensional Poisson problem with Dirichlet boundary conditions in the domain Ω with the following expression:

$$\begin{cases} -\nabla(k\nabla u) = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{18}$$

The domain Ω is the quarter-ring of the inner radius $R_i = 1$ and the outer radius $R_e = 2$ is shown in Fig. 8. When the right term heat source is $f = ((8 - 9\sqrt{(x^2 + y^2)} \sin(2 \arctan(y/x)))/(x^2 + y^2))$, the analytical solution is $u = (x^2 + y^2 - 3\sqrt{(x^2 + y^2)} + 2) \sin(2 \arctan(y/x))$.

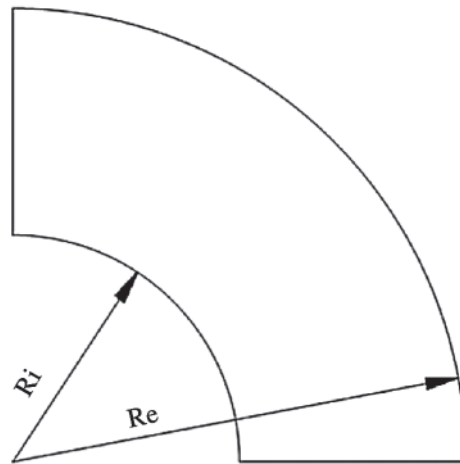


Figure 8: Two dimensional poisson heat transfer problem

For the parameter domain knot vector of two-dimensional Poisson equation is $v = u = [0 \ 0 \ 0 \ 1 \ 1 \ 1]$ in the initial state, the convergence and calculation time of subdivision strategy are compared under the working environment mentioned above. The related calculated data correspond to the Reynolds equation are listed below.

Table 8: Number of control points from MG based on different subdivision strategies

p	h							
	0	1	2	3	4	5	6	7
0	9	16	36	100	324	1156	4356	16900
1	16	25	49	121	361	1225	4489	17161
2	25	36	81	144	400	1296	4624	17424

Table 9: Convergence rate

$k = 2 \text{ iter} = 7$			$k = 3 \text{ iter} = 20$			$k = 4 \text{ iter} = 70$		
n	L_2 -error	Ration	n	L_2 -error	Ration	n	L_2 -error	Ration
100	$3.6649e^{-11}$	0.053717	121	$3.7055e^{-11}$	0.4703	144	$4.3997e^{-11}$	0.8503
324	$5.1800e^{-10}$	0.112660	361	$1.5032e^{-12}$	0.4413	400	$8.2384e^{-12}$	0.8505
1156	$3.5185e^{-10}$	0.131564	1225	$5.1833e^{-13}$	0.4587	1296	$1.8076e^{-12}$	0.8351
4356	$2.0245e^{-10}$	0.119191	4489	$1.1490e^{-13}$	0.4633	4624	$4.8159e^{-13}$	0.8362
16900	$1.5931e^{-10}$	0.121456	17161	$5.3937e^{-14}$	0.7366	17424	$1.2852e^{-13}$	0.8485

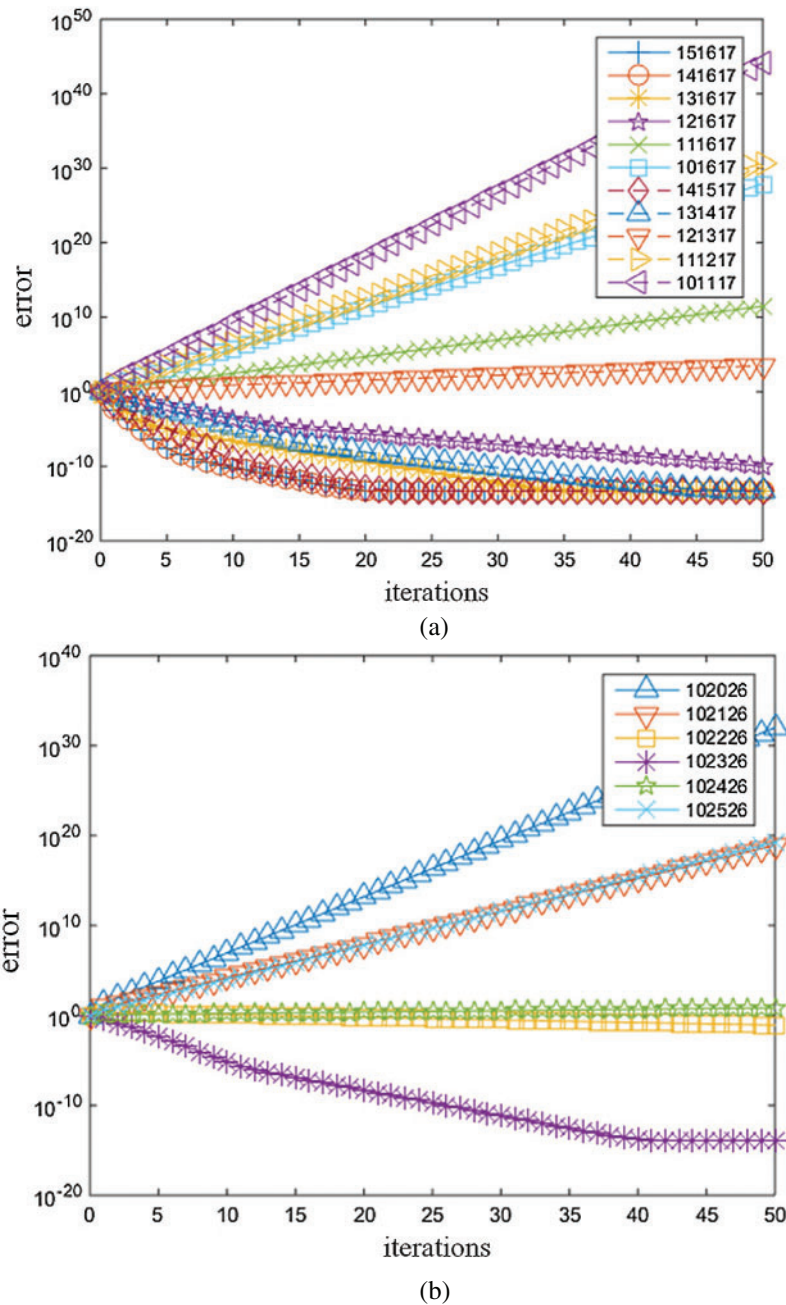


Figure 9: Relative error of refinement strategy (a) H -refinement relative error (b) K -refinement relative error

Tab. 9 discusses the influence of the choice of grid level on convergence in the AMG algorithm under different subdivision situations. Fig. 9 only shows the comparison between h -refinement and k -refinement in order to explore the influence of different refinement strategies on the convergence rate. It can be seen from the figure that the mapping matrix satisfies the mapping relationship between the control points when there is a large gap between the number

of control points of the coarse and fine grid, whether h -refinement or k -refinement. For the assumed scale series between \bar{u} and u are large, the linear distortion leads to the inaccurate mapping relationship, and the convergence rate is not converged as expected. Within a certain range of the number gap between the coarse and fine grid points, the convergence rate of h -refinement is different in the early iteration, but the convergence curve tends to coincide in the later 10 iterations, while the trend of k -refinement only convergence rate is the same. Therefore, the selection of the appropriate grid hierarchy is beneficial to the convergence of the multi-grid method. Fig. 10 shows the relative error of the AMG algorithm and the GMG algorithm. Tab. 10 shows the CPU calculation time of the AMG algorithm and the GMG algorithm. Although the convergence rate of the AMG algorithm and the GMG algorithm is almost the same in the verification of the Reynolds equation, the GMG algorithm still saves more time in the initial period calculation.

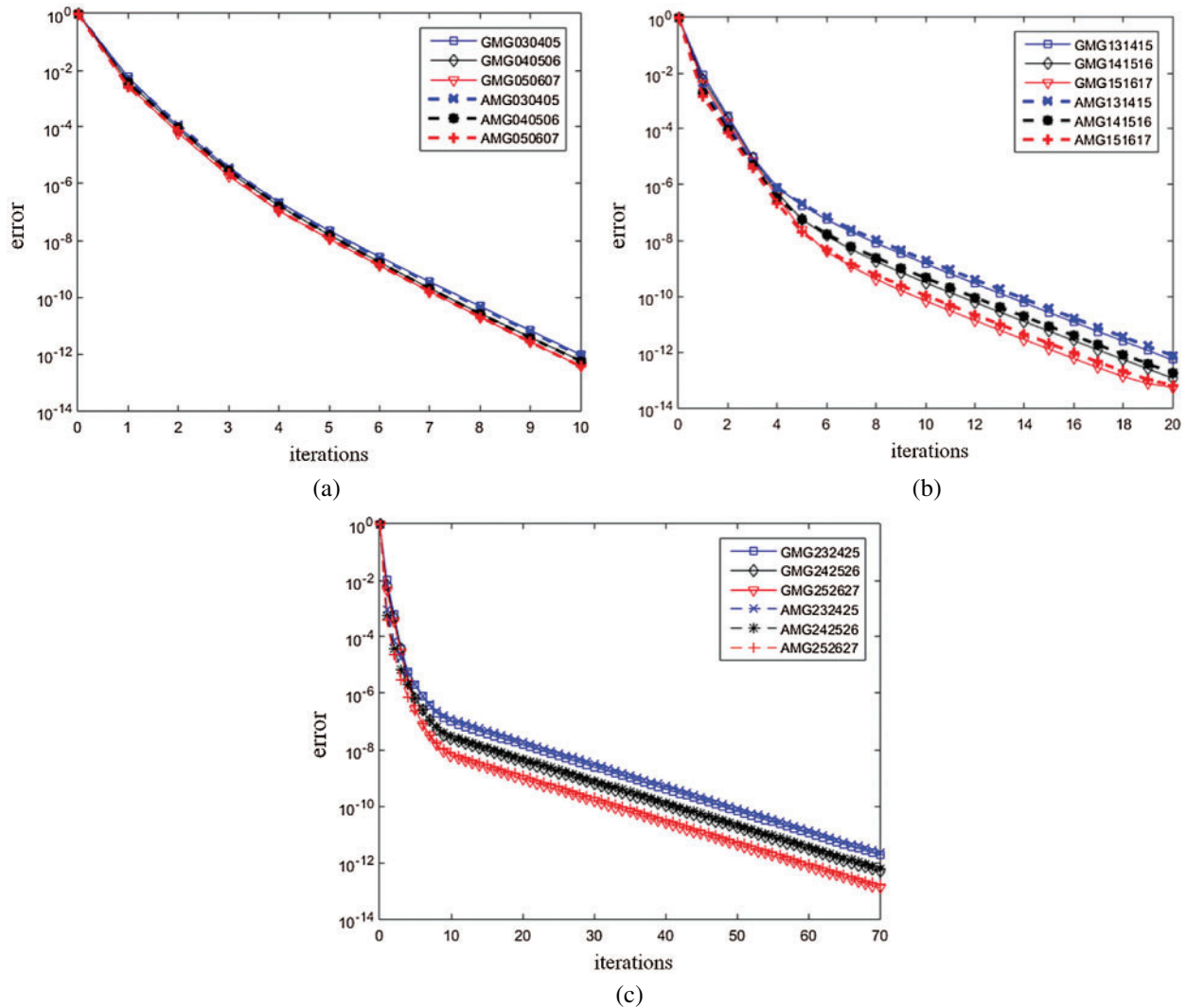
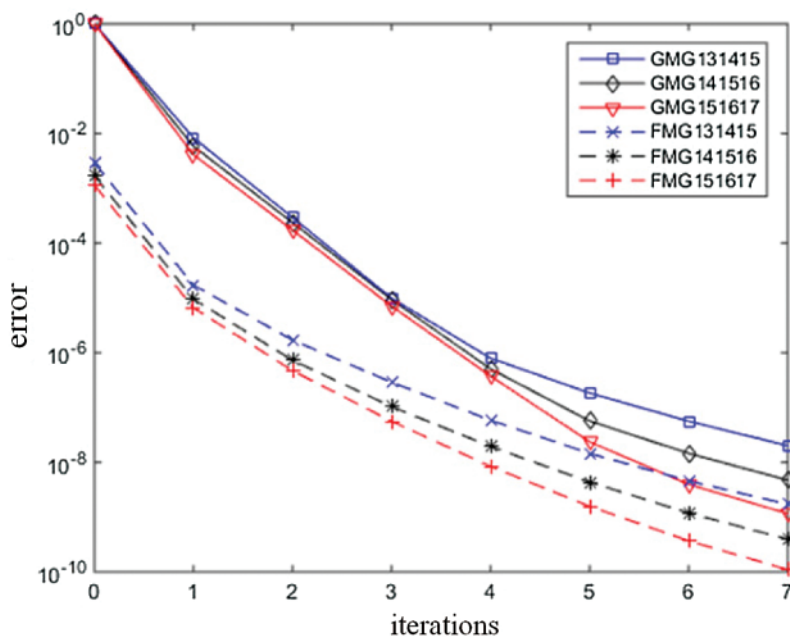


Figure 10: Relative error between AMG and GMG (a) $k=2$ (b) $k=3$ (c) $k=4$

Table 10: CPU calculation time of AMG and GMG

n	Computation time/s		n	Computation time/s		n	Computation time/s	
	AMG	GMG		AMG	GMG		AMG	GMG
1156	0.03075	0.02785	1225	0.06954	0.07617	1296	0.50252	0.52221
4356	0.10854	0.10078	4489	0.40158	0.37563	4624	3.31978	3.03466
16900	0.63829	0.58589	17161	2.18113	2.08475	17424	13.98436	12.82764

Fig. 11 shows the convergence comparison between FMG and GMG under the Reynolds equation model. It can be seen that: at the start stage, with the help of the good initial value, the FMG algorithm has a better accurate result, but the convergence tends to be consistent with others in the later stage.

**Figure 11:** Relative error between FMG and GMG

In this paper, the algebraic multigrid method and the geometric analysis of the natural discrete stiffness matrix A are used instead of the transpose generation matrix $A_{AMG}^{m-1} = P_m^{m-1} A_{AMG}^m R_{m-1}^m$, and the geometric multi-grid method and the algebraic multigrid method are compared. Numerical results show that: convergence rates of the two methods keep consistent, but GMG reduces the time cost at the initial period. For the subdivision strategies of different levels (3 levels), the convergence rate of h -refinement intermediate layer is faster when it is closer to the fine grid, and the convergence rate of k -refinement intermediate layer is faster when it is in the middle of coarse and fine grid. When FMG is applied to IGA, its convergence rate is faster than that of GMG.

5 Conclusion

Based on the framework of IGA, the analytical models of Reynolds equation and Poisson equation are established, and multigrid is applied to the IGA. According to its solving process and method, a calculation program was developed in OCTAVE (FMG based on GMG modified the traditional AMG). The convergence rate of the iteration is independent of the size of the discrete mesh, so the multigrid method has the optimal computational complexity. When the degree of freedom of fine grid is determined, the degree of freedom of coarse grid can be reduced and the boundary information can be quickly spread to the whole, which can not only accelerate the convergence rate but also save the cost of calculation time. However, the selection of grid level also affects the convergence accuracy. In particular, the selection of the middle level has a greater impact on the results, and the performance of the middle grid close to the top grid is better. As for the combination of FMG and IGA, the results show that the algorithm can make full use of IGA to discretize the geometric information of each layer. Compared with the V-cycle, it has a faster initial convergence speed, but it is consistent with conventional algorithms in the later stage. How to make better use of the existing information to accelerate calculation efficiency and improve calculation accuracy is the direction we will continue to study in the future.

Funding Statement: This research was supported by the Natural Science Foundation of Hubei Province (CN) (Grant No. 2019CFB693), the Research Foundation of the Education Department of Hubei Province (CN) (Grant No. B2019003) and the open Foundation of the Key Laboratory of Metallurgical Equipment and Control of Education Ministry (CN) (Grant No. 2015B14). The support is gratefully acknowledged.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Nguyen, V. P., Anitescu, C., Bordas, S. P., Rabczuk, T. (2015). Isogeometric analysis: An overview and computer implementation aspects. *Mathematics and Computers in Simulation*, 117, 89–116. DOI 10.1016/j.matcom.2015.05.008.
2. Briggs, W. L., Henson, V. E., McCormick, S. F. (2000). *A multigrid tutorial*. 2nd edition. Philadelphia, PA: Society for Industrial and Applied Mathematics.
3. Tai, C. H., Zhao, Y. (2003). Parallel unsteady incompressible viscous flow computations using an unstructured multigrid method. *Journal of Computational Physics*, 1(1), 277–311. DOI 10.1016/j.jcp.2003.07.005.
4. Stüben, K. (2001). A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 1–2, 281–309. DOI 10.1016/S0377-0427(00)00516-1.
5. Hofreither, C., Zulehner, W. (2015). Spectral analysis of geometric multigrid methods for isogeometric analysis. *Numerical Methods and Applications*, 8962, 123–129. DOI 10.1007/978-3-319-15585-2_14.
6. Hofreither, C., Zulehner, W. (2016). On full multigrid schemes for isogeometric analysis. *Domain Decomposition Methods in Science and Engineering*, 22, 267–274. DOI 10.1007/978-3-319-18827-0_25.
7. Buffa, A., Harbrecht, H., Kunoth, A., Sangalli, G. (2013). BPX-preconditioning for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 265, 63–70. DOI 10.1016/j.cma.2013.05.014.
8. Chemin, A., Elguedj, T., Gravouil, A. (2015). Isogeometric local h-refinement strategy based on multigrids. *Finite Elements in Analysis and Design*, 100, 77–90. DOI 10.1016/j.finel.2015.02.007.
9. Gahalaut, K. P., Johannes, K., Satyendra, T. (2013). Multigrid methods for isogeometric discretization. *Computer Methods in Applied Mechanics and Engineering*, 100, 413–425. DOI 10.1016/j.cma.2012.08.015.

10. Hofreither, C., Takacs, S., Zulehner, W. (2017). A robust multigrid method for isogeometric analysis in two dimensions using boundary correction. *Computer Methods in Applied Mechanics and Engineering*, 316, 22–42. DOI 10.1016/j.cma.2016.04.003.
11. Guo, L. C. (2013). *Improving the computational efficiency of isogeometric analysis via parallel computing and multigrid methods*. Hefei, China: University of Science and Technology of China.
12. Liu, S., Chen, D. X., Feng, Y. X., Xu, Z. L., Zheng, L. K. (2014). A multigrid preconditioned conjugate gradient method for isogeometric analysis. *Applied Mathematics & Mechanics*, 35(6), 630–639. DOI 10.3879/j.issn.1000-0887.2014.06.005.
13. Chen, D., Xu, Z. (2014). An application of multigrid in viscous flow simulations with least squares isogeometric method. *Journal of Xian Jiaotong University*, 48(11), 122–127. DOI 10.7652/xjtuxb201411021.
14. Hofreither, C., Zulehner, W. (2015). Mass smoothers in geometric multigrid for isogeometric analysis. *Curves and Surfaces*, 9213, 272–279. DOI 10.1007/978-3-319-22804-4_20.
15. Donatelli, M., Garoni, C., Manni, C., Serra-Capizzano, S., Speleers, H. (2015). Robust and optimal multi-iterative techniques for IgA Galerkin linear systems. *Computer Methods in Applied Mechanics and Engineering*, 284, 230–264. DOI 10.1016/j.cma.2014.06.001.
16. Donatelli, M., Garoni, C., Manni, C., Serra-Capizzano, S., Speleers, H. (2017). Symbol-based multigrid methods for Galerkin B-spline isogeometric analysis. *SIAM Journal on Numerical Analysis*, 55(1), 31–62. DOI 10.1137/140988590.
17. Piegl, L. A., Tiller, W. (1997). *The NURBS book*. Berlin Heidelberg: Springer.
18. Vuong, A. V., Heinrich, C., Simeon, B. (2010). ISOGAT: A 2D tutorial MATLAB code for isogeometric analysis. *Computer Aided Geometric Design*, 8(8), 644–655. DOI 10.1016/j.cagd.2010.06.006.
19. Wang, D. D., Xuan, J. C., Zhang, C. H. (2012). A three dimensional computational investigation on the influence of essential boundary condition imposition in NURBS isogeometric finite element analysis. *Chinese Journal of Computational Mechanics*, 29(1), 31–37. DOI 10.7511/jslx20121006.
20. Wang, X., Hu, P., Zhu, X., Gai, Y. (2016). Topology description function approach using nurbs interpolation for 3D structures with self-weight loads. *Lixue Xuebao/Chinese Journal of Theoretical & Applied Mechanics*, 48(6), 1437–1445. DOI 10.6052/0459-1879-16-145.
21. Da Veiga, L. B., Buffa, A., Rivas, J., Sangalli, G. (2011). Some estimates for h-p-k-refinement in isogeometric analysis. *Numerische Mathematik*, 118(2), 271–305. DOI 10.1007/s00211-010-0338-z.
22. Cohen, E., Lyche, T., Riesenfeld, R. (1980). Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Computer graphics and image processing*, 14(2), 87–111. DOI 10.1016/0146-664X(80)90040-4.
23. Cohen, E., Lyche, T., Schumaker, L. L. (1985). Algorithms for degree-raising of splines. *ACM Transactions on Graphics*, 3(3), 171–181. DOI 10.1145/282957.282962.
24. Chang, Q., Wong, Y. S., Fu, H. (1996). On the algebraic multigrid method. *Journal of Computational Physics*, 2(2), 279–292. DOI 10.1006/jcph.1996.0094.
25. Wei, Z. P., Luo, H. X., Zuo, B. Q., Fei, J. G. (2020). Research on multigrid algorithm based on isogeometric hierarchical model of K-subdivision. *Modular Machine Tool & Automatic Manufacturing Technique*, 7, 30–35. DOI 10.13462/j.cnki.mmtamt.2020.07.007.
26. Patir, N., Cheng, H. S. (1978). An average flow model for determining effects of three-dimensional roughness on partial hydrodynamic lubrication. *Journal of Lubrication Technology*, 1(1), 12–17. DOI 10.1115/1.3453103.
27. Patir, N., Cheng, H. S. (1979). Application of average flow model to lubrication between rough sliding surfaces. *Journal of Tribology*, 2, 220–229. DOI 10.1115/1.3453329.
28. Xu, H. Y., Luo, H. X., Zuo, B. Q., Li, Z. X. (2019). The application of isogeometric method in Reynolds equation. *Machinery Design & Manufacture*, 7, 184–188. DOI 10.19356/j.cnki.1001-3997.2019.07.046.
29. Zhang, M. X., Zuo, B. Q., Huang, Z. D., Xiong, T. F. (2015). A combined solution for DAE and PDE of cylinder-piston system. *Computer Applications and Software*, 12, 160–164. DOI 10.3969/j.issn.1000-386x.2015.12.038.