A Parallel High-Order Solver for Linear Elasticity Problems Using A Weak Galerkin Finite Element Method on Unstructured Quadrilateral Meshes

by Yunze Li

B.S. in Mechanical Engineering, July 2016, Xi'an Jiaotong University

A Thesis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Master of Science

May 19, 2019

Thesis directed by

Chunlei Liang
Associate Professor of Mechanical and Aerospace Engineering

© Copyright 2019 by Yunze Li
All rights reserved

Acknowledgments

First and foremost, I want to extend my heartfelt gratitude to my supervisor, Dr Chunlei Liang, whose patient guidance and valuable suggestions help me significantly to complete this thesis. He has been a great example inspiring me to be a future scholar in the field of scientific computation.

Moreover, I would like to thank Dr Lin Mu. She gave me tremendous encouragement and technical instruction in learning weak Galerkin finite element method. I am also greatly indebted to my lab mates and best friends: Mao Li, Xiaoliang Zhang and Bin Zhang, who have shared with me a lot of great time together in the past two years.

Last but not least, I am deeply indebted to my parents, who always support me without any condition.

Abstract

A Parallel High-Order Solver for Linear Elasticity Problems Using A Weak Galerkin Finite Element Method on Unstructured Quadrilateral Meshes

Weak Galerkin Finite Element Method (WGFEM) [15] is a high-order discontinuous method for unstructured grids. Recently, The WGFEM has been successfully applied to solve lineanr elasticity problems. In this work, we integrate a domain decomposition method with a WGFEM solver for parallel solutions of linear elasticity equations on unstructured quadrilateral grids. In particular, we employ a Balancing Domain Decomposition by Constraints (BDDC) [4] to effectively reduce the computational cost of the coarse problem for the interface unknowns. The WG-BDDC method shares some similarity with the well-known FETI-DP. However, the standard continuous Galerkin Finite Element Method solves unknowns on nodal points. These nodal points on subdomain interfaces may connect to multiple elements in the FETI-DP method after domain decomposition. The WG-BDDC method does not need to involve solutions on the nodal points. Therefore, the communication between two adjacent subdomains only requires to collect information over their common faces.

A unique feature of the WGFEM method is the use of weak gradient operator to differentiate basis functions. For 2D problems, an integration by parts was used to change surface integrals to line integrals. WGFEM employs two kinds of basis functions. The first group is on the edges of each element and the second group is inside each element. Two groups of bsas function can be chosen independently. In order to have continuous solution in each element given that two different spaces are used, a stabilizer is introduced at element interfaces. In the BDDC method, the unknowns are grouped to interior, dual and primal spaces for each subdomain by the technique of Schur complement. Unknowns in the primal spaces will constitute a new global problem to solve. The size of this new global problem is significantly smaller than that of the global problem involving all unknowns on subdomain interfaces.

First, we tested the orders of accuracy of WG-BDDC method on structured grids. Subse-

quently, by using our university cluster, the WG-BDDC is robust for solving linear elasticity problems on parallel computers. Excellent scalability performance was obtained for the WG-BDDC method on uniform grids by testing over 144 CPUs. Moreover, the WG-BDDC method is successfully extended to fully unstructured grids of all quadrilateral elements to solve beam deformation problems. The results on unstructured grids is comparable qualitatively to that on structured grids.

Table of Contents

Acknowledgments . . . . . **iii**

Abstract . . . . . **iv**

List of Figures . . . . . **vii**

List of Tables . . . . . **viii**

1 Introduction **1**

2 Numerical Method **3**

2.1 Mathematical Models 3
2.2 Form Variation 5
2.3 Weak Galerkin Finite Element Method 6
2.3.1 Preliminary 6
2.3.2 Weak Operators 7
2.3.3 Weak Galerkin Quadrilateral Meshes and Basis Functions 8
2.3.4 Weak Galerkin Method for Linear Elasticity Equation 10

3 Parallel Computing Method **17**

3.1 Block Cholesky Elimination and Schur Complement 17
3.2 FETI Method and FETI-DP Method 18
3.3 WG-BDDC Method 22
3.3.1 Primal and Dual Spaces 23
3.3.2 Preconditioned Conjugate Gradient Method 25
3.3.3 Local Information Recovery 26
3.3.4 Local Information Recovery 27

4 Results and Discussion **28**

4.1 Accuracy Verification 28
4.1.1 2D Linear Elasticity Equation 28
4.1.2 Beam Deformation Tests 31
4.2 Scalability and Speed-up Tests 34

5 Conclusions and Future Work **36**

A Stabilizer **37**

B One-dimensional Weak Galerkin Method **39**

B.1 Transfer Global Equation into Weak Form 39
B.2 Weak Gradient Operator 39
B.3 Stabilizer 40
B.4 Solution for 1-D Example 40

Bibliography **44**

List of Figures

2.1 Stress, displacements, and body force 4
2.2 Weak Galerkin quadrilateral elements and solution points. 8
2.3 Mapping for quadrilateral elements. 9
2.4 Mapping for quadrilateral elements. 15

3.1 Domain decomposition in FETI 19
3.2 BDDC computational domain 23

4.1 WG method plotting using structured mesh 29
4.2 WG method plotting using structured mesh 30
4.3 Grid deformation after simulation using a structured mesh 31
4.4 Contour plot the x-component displacements on a structured mesh 32
4.5 Grid deformation after simulation using an unstructured mesh 33
4.6 Contour plot the x-component displacements on an unstructured mesh 33
4.7 Scalability of WGFEM method 34

B.1 Mesh for 1-D problem 41

List of Tables

4.1 L^2 errors for the second order WGFEM on structured grids 29
4.2 L^2 errors for the third order WGFEM on structured grids 30
4.3 L^2 errors for the second order WGFEM on unstructured grids 31

Chapter 1: Introduction

Finite Element (FE) methods [2] have been widely used in solving structural mechanics problems over the past 30 years. The FE method can be used on unstructured grids and thus is geometrically flexible. One of the most popular FE methods is the continuous Galerkin finite element method (CGFEM). In CGFEM, the computational domain is divided into a group pf connected elements and unknowns are placed at nodal points of each element. Usually the Galerkin method is used to satisfy the conservation law in a weak form, and the law is satisfied for the global domain rather than for each element. Therefore, CGFEM involves very large, sparse matrices in application.

There is a need of accurate parallel computing algorithms for the finite element method because practical engineering problems requires large amount of computational time. One famous parallel computing technique for the standard continuous finite element method is the Finite Element Tearing and Inter Connecting (FETI) [5] accelerated by separating interface unknowns in dual and primal spaces, namely FETI-DP method [3]. In particular numerical analysis, the FETI method is an iterative substructuring method for solving systems of linear equations from the finite element method for the solution of elliptic partial differential equations. In each iteration, FETI [6] requires the solution of a Neumann problem in each substructure and the solution of a coarse problem.

This MS thesis research aims to explore future practical applications of the WG method as a new high order discontinuous finite element method. The WG method for solving the second order elliptic equations is developed by Wang and Ye[2], and Mu[3] proved its accuracy on some other partial differential equations such as Biharmonic equations and Helmholtz equations. Usually, quadrilateral elements are favored for solving boundary-layer problems. Therefore, studies in quadrilateral grids instead of triangular grids have been done for the WG method which solves boundary unknowns on element faces. It consists

of two key components, i.e., the weak gradient operators contributing stiffness matrix and stabilizer minimizing the differences between interior unknowns and unknowns on the element interfaces through L_2 projections [11]. In this research, interior space and boundary space are the same orders of polynomial when making numerical tests, which are the second order and the third order.

The simplest version of FETI with no preconditioner in the substructure is scalable with the number of substructures but the condition number grows polynomially with the number of elements per substructure. To avoid this disadvantage, the BDDC algorithm [4], first introduced by Dohrmann, is a variant of the two-level Neumann–Neumann type preconditioner for solving the interface coarse problem. BDDC with a preconditioner consisting of the solution of a Dirichlet problem in each substructure is scalable with the number of substructures and its condition number grows only polylogarithmically with the number of elements per substructure. Throughout this thesis, we will employ the Block Cholesky Elimination [7] and our discussion can also be seen as a practice of domain decomposition methods using such a framework. We use the Cholesky factorization to decompose matrices into triangular matrices for getting their inverse matrices and LAPACK [1] for matrix multiplying. The Block Cholesky Elimination is a decomposition of a positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose. Unknowns on boundaries for each sub-domain is grouped into dual and primal spaces, and the large global matrix is replaced by several local matrices. After the Block Cholesky Elimination on each CPU, the displacements of primal space from each processor is collected by MPI process. A preconditioned conjugate gradient method [12] was used to solve the interface problems of the BDDC algorithms, and the iterations were stopped when the relative primal residual is small enough. The last step is recovering the displacements of interior and dual space by the results of primal space from PCG iteration. The calculations in each sub-domain are independent, so we can use different CPU to do the Block Cholesky Elimination and information recovery at the same time.

Chapter 2: Numerical Method

2.1 Mathematical Models

The FE method has enjoyed many successes in computational solid mechanics. One of the most notable areas is elastic mechanics, an important branch of solid mechanics, for studying the deformation and internal forces generated by elastic and deformable objects under external forces. When an external force does not exceed a certain limit, the object is restored to its original state after the external force is removed. Linear and small displacement theory has been widely used to model elasticity. Such a model is very suitable for the FE method to solve. The basic starting points for modeling elastic mechanics are

- Balance of momentum
- Strain displacement
- Strain-stress temperature

Following Newton's second law, the dynamic equations of motion for linear elasticity problems can be described in terms of the stress tensor $\boldsymbol{\sigma}$, the strain tensor $\boldsymbol{\varepsilon}$, the displacement vector $\boldsymbol{u}$, and the body force vector $\boldsymbol{f}$. The governing elasticity equation can be written as

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \boldsymbol{f} = \rho \ddot{\boldsymbol{u}} \tag{2.1}$$

In this MS thesis research, we focus on solving two-dimensional elasticity equation with a balance of momentum. By defining the computational domain and boundary conditions, we have a complete mathematical model for static linear elasticity problems.

$$\begin{aligned} -\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\boldsymbol{u}) &= \boldsymbol{f}, \qquad in \quad \Omega \\ \boldsymbol{u} &= \hat{\boldsymbol{u}}, \qquad on \quad \partial\Omega \end{aligned} \tag{2.2}$$

where $\sigma(u)$ is a symmetric Cauchy stress tensor and ρ stands for mass density of the elastic body.

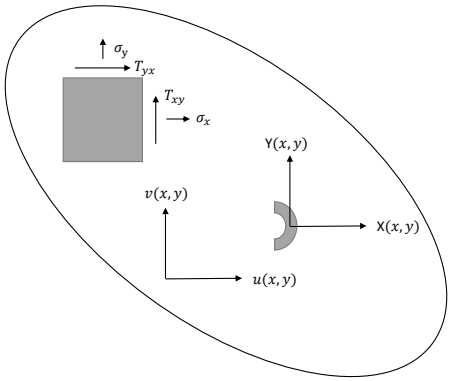


Figure 2.1: Stress, displacements, and body force

As shown in figure 2.1, x and y components of displacements are $u(x,y)$ and $v(x,y)$. $X(x,y)$ and $Y(x,y)$ are x and y components of forces. Meanwhile, normal and shear stresses σ_x, σ_y, τ_{xy}, and τ_{yx} are also defined in the figure. For linear, isotropic and homogeneous materials, the stress-strain relation is

$$\boldsymbol{\sigma}(u) = 2\mu\varepsilon(u) + \lambda(\boldsymbol{\nabla}\cdot\boldsymbol{u})\boldsymbol{I} \tag{2.3}$$

Where $\varepsilon(u)$ is evaluated through the strain-displacement relation,

$$\varepsilon(u) = \frac{1}{2}(\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}\boldsymbol{u}^T) \tag{2.4}$$

μ and λ are Lame indices which can be written as

$$\lambda = \frac{Ev}{(1+v)(1-2v)} \tag{2.5}$$

$$\mu = \frac{E}{2(1+v)} \tag{2.6}$$

where E is the elasticity modulus and v is the Poisson's ratio.

2.2 Form Variation

To facilitate our introduction of a variational form for the above mathematical model, we define some notations for tensor calculations. The divergence of a tensor field A is defined as the vector with components

$$(\nabla \cdot A)_i = \frac{\partial A_{ij}}{\partial x_j} \tag{2.7}$$

The gradient of a vector field b is the tensor with components

$$(\nabla b)_{i,j} = \frac{\partial b_i}{\partial x_j} \tag{2.8}$$

Now we introduce the virtual work ω and integrate the governing equation for static linear elasticity problems, where ω is a vector

$$-\int_\Omega [\nabla \cdot \boldsymbol{\sigma}(u)]\omega dX = \int_\Omega f\omega dX \tag{2.9}$$

After an integration by parts to the left hand side, we have

$$\int_\Omega [\nabla \cdot \boldsymbol{\sigma}(u)]\omega dX = \int_{\partial\Omega} \nabla \cdot [\boldsymbol{\sigma}(u)\omega] ds - \int_\Omega \boldsymbol{\sigma}(u) : \varepsilon(\omega) dX \tag{2.10}$$

The weak form of equation(2.2) can be rewritten as

$$\int_{\Omega} \boldsymbol{\sigma}(u) : \varepsilon(\omega) dX = \int_{\Omega} f\omega dX + \int_{\partial\Omega} [\boldsymbol{\sigma}(u)\omega] \cdot n ds \tag{2.11}$$

Taking into account of the stress-strain relation, the left hand side can be further derived into

$$\begin{aligned} \int_{\Omega} \boldsymbol{\sigma}(u) : \varepsilon(\omega) dX &= \int_{\Omega} [\lambda tr\varepsilon(u) I + 2\mu\varepsilon(u)] : [\frac{1}{2}(\nabla\omega + \nabla\omega^T)] dX \\ &= 2\mu \int_{\Omega} \varepsilon(u) : \varepsilon(\omega) dX + \lambda \int_{\Omega} tr\varepsilon(u) I : \varepsilon(\omega) dX \end{aligned} \tag{2.12}$$

2.3 Weak Galerkin Finite Element Method

2.3.1 Preliminary

In order to formulate the Weak Galerkin method, it is important to introduce the notion of weak derivatives and define related polynomial spaces[8]. We are considering functions for which all the derivatives, which belong to L^2 space so that most of the classical derivation rules can be applied. In this MS thesis research, all the computation for a Dirichlet problem in the form of linear elasticity equations over a bounded domain $\Omega \subset \mathbb{R}^n$ is in Sobolev space. The norm is defined as

$$||v||_{s,\Omega} = \left(\sum_{|\alpha|=s} \int_{\Omega} |\partial^2 v|^2 d\Omega \right)^{1/2} \tag{2.13}$$

where

$$\alpha = (\alpha_1, \alpha_2, ..., \alpha_n) \tag{2.14}$$

$$|\alpha| = \alpha_1 + \alpha_2 + ... + \alpha_n \tag{2.15}$$

The divergence is defined as

$$H(div;\Omega) = \{v : v \in [L^2(\Omega)]^n, \nabla \cdot v \in L^2(\Omega)\} \tag{2.16}$$

whose norm is defined as

$$||v||_{H(div;\Omega)} = (||v||_{\Omega}^2 + ||\nabla \cdot v||_{\Omega}^2)^{1/2} \tag{2.17}$$

The curl of $H(curl;\Omega)$ in $L^2(\Omega)$ is defined as

$$H(curl;\Omega) = \{v : v \in [L^2(\Omega)]^n, \nabla \times v \in L^2(\Omega)\} \tag{2.18}$$

whose norm is defined as

$$||v||_{H(curl;\Omega)} = (||v||_{\Omega}^2 + ||\nabla \times v||_{\Omega}^2)^{1/2} \tag{2.19}$$

2.3.2 Weak Operators

The WG method consists of two key components, i.e., the weak gradient for contributing stiffness matrix and stabilizer for contact interior unknowns and unknowns on the boundary. Unknowns are divided by on boundaries (u_b) and interior (u_0) for each element independently. Let's assume that $T \subset \mathbb{R}^d$ is an arbitrary polygon domain, the boundary is ∂T. The weak function in the domain is $v = \{v_0, v_b\}$ with $v_0 \in L^2(T)$ and $v_b \in L^2(\partial T)$. The weak space for all T is Sobolev space, which is defined as $S(T)$

$$S(T) = \{v = \{v_0, v_b\} : v_0 \in L^2(T), v_b \in L^2(\partial T)\} \tag{2.20}$$

Now we define weak operators we used in WGFEM. For any $v \in S(T)$, the discrete weak gradient of v is $\nabla_w v$

$$(\nabla_w v, q)_{\Omega} = -(v_0, \nabla \cdot q)_{\Omega} + \langle v_b, q \cdot n \rangle_{\partial\Omega} \tag{2.21}$$

For any $v \in S(T)$, the weak divergence operator is $\nabla_w \cdot v$

$$(\nabla_w \cdot v, q)_{\Omega} = -(v_0, \nabla q)_{\Omega} + \langle v_b \cdot n, q \rangle_{\partial\Omega} \tag{2.22}$$

where q is a test function whose basis function is in L^2 space.

2.3.3 Weak Galerkin Quadrilateral Meshes and Basis Functions

Two sets of points are defined in the reference element: interior points and boundary points. Unlike the continuos finite element methods, the WG method does not assign points on element corner points. In Figure 2.2, we present two quadrilateral WG elements as T_1 and T_2. Solid points represent solution points for u_0 and hollow points is solution points for u_b. In the case of the first-order WG scheme, there is one solution point for u_0 and three interior basis functions representing 1, x and y respectively. Meanwhile, Gauss-Legendre quadrature point locations are used for points on boundaries in the present study. We set linear basis functions and two unknown variables on each edge. For the second-order WG scheme, the solution point for u_0 is the same, but there are three points on each boundary.

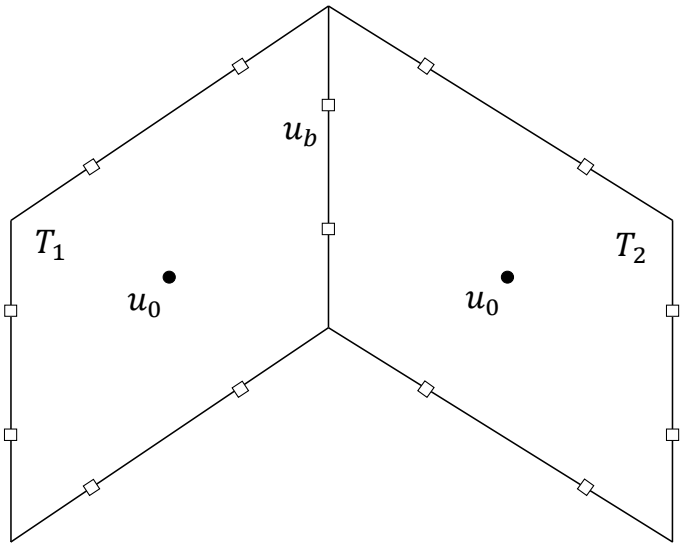


Figure 2.2: Weak Galerkin quadrilateral elements and solution points.

In two dimensional cases, displacement u for each solution point has the following form

$$u = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{2.23}$$

Considering on the second order cases, we have $N_0 = 3$ and $N_b = 8$ so that in total 11 unknown variable in each direction. The basis function has the form that $\psi = \{\psi_0, \psi_b\}$. We have $N_0 = 3$ and $N_b = 8$, so that in total 11 unknown variable in each direction.

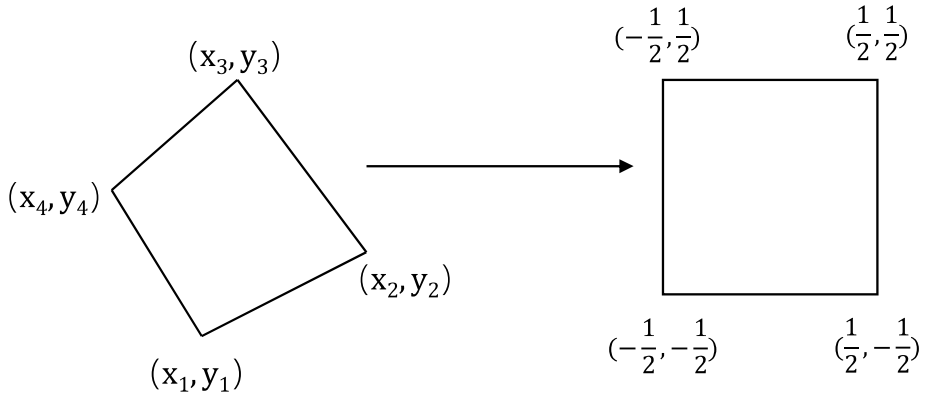


Figure 2.3: Mapping for quadrilateral elements.

The basis function of u_0 is in reference element as shown in Figure 2.3.

$$\psi_1 = [1, 1] \tag{2.24}$$

$$\psi_2 = [\xi, \xi] \tag{2.25}$$

$$\psi_3 = [\eta, \eta] \tag{2.26}$$

The basis function of u_b is based on each edge and they are arc functions.

$$\psi_4 = \begin{cases} \hat{s} & on \quad e_1 \\ 0 & else \end{cases} \qquad \psi_5 = \begin{cases} 1 - \hat{s} & on \quad e_1 \\ 0 & else \end{cases} \tag{2.27}$$

$$\psi_6 = \begin{cases} \hat{s} & on \quad e_2 \\ 0 & else \end{cases} \qquad \psi_7 = \begin{cases} 1-\hat{s} & on \quad e_2 \\ 0 & else \end{cases} \tag{2.28}$$

$$\psi_8 = \begin{cases} \hat{s} & on \quad e_3 \\ 0 & else \end{cases} \qquad \psi_9 = \begin{cases} 1-\hat{s} & on \quad e_3 \\ 0 & else \end{cases} \tag{2.29}$$

$$\psi_{10} = \begin{cases} \hat{s} & on \quad e_4 \\ 0 & else \end{cases} \qquad \psi_{11} = \begin{cases} 1-\hat{s} & on \quad e_4 \\ 0 & else \end{cases} \tag{2.30}$$

2.3.4 Weak Galerkin Method for Linear Elasticity Equation

By defining the computational domain, we have a mathematical model for static linear elasticity problems

$$-\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \boldsymbol{f}, \qquad in \quad \Omega \tag{2.31}$$

Considering the Dirichlet boundary condition, we have

$$\boldsymbol{u} = \hat{\boldsymbol{u}}, \qquad on \quad \partial\Omega \tag{2.32}$$

After the processes of form variation in 2.3.1, we get WGFEM weak form equation.

$$\begin{aligned} \sum_e 2\mu \int_\Omega \varepsilon(u) : \varepsilon(\omega) dX + \sum_e \lambda \int_\Omega tr\varepsilon(u) I : \varepsilon(\omega) dX + s(u,\omega) = \\ \sum_e \{ \int_\Omega f\omega dX - \int_{\partial\Omega} [\boldsymbol{\sigma}(u)\omega] \cdot n ds \} \end{aligned} \tag{2.33}$$

In this section, we use the weak operators to evaluate these terms which are grouped into three parts: tensor production, stabilizer and right hand side.

2.3.4.1 Tensor Production

We apply the constant test function q to the weak gradient equation. Then the weak gradient equation can be rewritten as

$$\int_{\Omega_e} (\nabla_w u_h) q dX = -\int_{\Omega_e} u_0 \nabla \cdot q dv + \int_{\partial\Omega_e} u_b q \cdot \mathbf{n} ds \tag{2.34}$$

Define weak Galerkin finite element space $u_0 \in P_1(\Omega_e^0)$, $u_b \in P_1(\partial\Omega_e^0)$, $\nabla_w u \in [P_0(\Omega_e^0)]^2$, $q \in [P_0(\Omega_e^0)]^2$. In this research, the bases of q and $\nabla_w u_h$ are $[1 \quad 0]^T$ and $[0 \quad 1]^T$ for each degree of freedom, thus

$$\begin{bmatrix} \underline{q_x} \\ \underline{q_y} \end{bmatrix} = \begin{bmatrix} 1 \cdot \underline{q_x} \\ 1 \cdot \underline{q_y} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{q_x} \\ \underline{q_y} \end{bmatrix} \tag{2.35}$$

So the term containing ∇q in equation (2.34) is 0. On each edge $u_b = \psi_{b1} u_{b1} + \psi_{b2} u_{b2}$, while on each degree of freedom, $u_{b,i}$ is only related to its own shape function $\psi_{b,i}$. As a result, we can get the following equation on each edge DOF $u_{b,i}$. In two dimensional cases, there are now two equations involving two dependent variables.The term $\frac{ds}{d\hat{s}}$ is the length of this edge.

$$\underline{q_x} \int_{\Omega_e} \psi_{q_x} \psi_{\nabla_w u_h} dX \underline{\nabla_w u_{h_{x,i}}} = \underline{q_x} \int_{\partial\Omega_e} \psi_{q_x} \cdot n_x \psi_{b,i} d\hat{s} \frac{ds}{d\hat{s}} \underline{u_b} \tag{2.36}$$

$$\underline{q_y} \int_{\Omega_e} \psi_{q_y} \psi_{\nabla_w u_h} dX \underline{\nabla_w u_{h_{y,i}}} = \underline{q_y} \int_{\partial\Omega_e} \psi_{q_y} \cdot n_y \psi_{b,i} d\hat{s} \frac{ds}{d\hat{s}} \underline{u_b} \tag{2.37}$$

From this equation, we can get the weak gradient vector $\begin{bmatrix} \underline{\nabla_w u_{h_\xi}} \\ \underline{\nabla_w u_{h_\eta}} \end{bmatrix}$ of u_0 in the reference element as

$$\frac{\partial \psi_1}{\partial \xi} = 0 \quad , \quad \frac{\partial \psi_1}{\partial \eta} = 0 \tag{2.38}$$

$$\frac{\partial \psi_2}{\partial \xi} = 0 \quad , \quad \frac{\partial \psi_2}{\partial \eta} = 0 \tag{2.39}$$

$$\frac{\partial \psi_3}{\partial \xi} = 0 \quad , \quad \frac{\partial \psi_3}{\partial \eta} = 0 \tag{2.40}$$

The weak gradient vectors of u_b in physical element are

$$\frac{\partial \psi_4}{\partial x} = \frac{y_2 - y_1}{2|A|} \quad , \quad \frac{\partial \psi_4}{\partial y} = -\frac{x_2 - x_1}{2|A|} \tag{2.41}$$

$$\frac{\partial \psi_5}{\partial x} = \frac{y_2 - y_1}{2|A|} \quad , \quad \frac{\partial \psi_5}{\partial y} = -\frac{x_2 - x_1}{2|A|} \tag{2.42}$$

$$\frac{\partial \psi_6}{\partial x} = \frac{y_3 - y_2}{2|A|} \quad , \quad \frac{\partial \psi_6}{\partial y} = -\frac{x_3 - x_2}{2|A|} \tag{2.43}$$

$$\frac{\partial \psi_7}{\partial x} = \frac{y_3 - y_2}{2|A|} \quad , \quad \frac{\partial \psi_7}{\partial y} = -\frac{x_3 - x_2}{2|A|} \tag{2.44}$$

$$\frac{\partial \psi_8}{\partial x} = \frac{y_4 - y_3}{2|A|} \quad , \quad \frac{\partial \psi_8}{\partial y} = -\frac{x_4 - x_3}{2|A|} \tag{2.45}$$

$$\frac{\partial \psi_9}{\partial x} = \frac{y_4 - y_3}{2|A|} \quad , \quad \frac{\partial \psi_9}{\partial y} = -\frac{x_4 - x_3}{2|A|} \tag{2.46}$$

$$\frac{\partial \psi_{10}}{\partial x} = \frac{y_1 - y_4}{2|A|} \quad , \quad \frac{\partial \psi_{10}}{\partial y} = -\frac{x_1 - x_4}{2|A|} \tag{2.47}$$

$$\frac{\partial \psi_{11}}{\partial x} = \frac{y_1 - y_4}{2|A|} \quad , \quad \frac{\partial \psi_{11}}{\partial y} = -\frac{x_1 - x_4}{2|A|} \tag{2.48}$$

where $|A|$ is the area of element. x_i and y_i $(i = 1,2,3,4)$ are vertex coordinates of the element.

Kinematics or strain displacement refers to the geometry of deformation. Measures of changed in lengths and in angles, generally called strains, are important quantities that are used to assist in describing the deformation. An extensional strain is a ratio of a change in length relative to corresponding original length as a result of a general deformation. Shear strains refer to distortion as a result of changed in angles usually between two originally perpendicular lines. For the two-dimensional linear theory, the appropriate linear strains are

$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x} \qquad \varepsilon_{yy} = \frac{\partial u_y}{\partial y} \qquad \gamma_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \tag{2.49}$$

In order to calculate in discrete form, the expressions for the strains must be developed. Recall that the strain vector ε is defined by

$$\varepsilon = \begin{bmatrix} \varepsilon_{xx} \\ \gamma_{xy} \\ \gamma_{yx} \\ \varepsilon_{yy} \end{bmatrix} \tag{2.50}$$

Using the representation for the displacement within the element, this can be expresses that

$$\varepsilon = \begin{bmatrix} \frac{\partial \psi}{\partial x} & 0 \\ \frac{\partial \psi}{\partial y} & 0 \\ 0 & \frac{\partial \psi}{\partial x} \\ 0 & \frac{\partial \psi}{\partial y} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{2.51}$$

This can be written as

$$\varepsilon = Lu \tag{2.52}$$

where

$$u = [u_x \quad u_y]^T \tag{2.53}$$

In this case, γ is split into two terms as presented.

$$\sum_e 2\mu \int_\Omega \varepsilon(u) : \varepsilon(\omega) dX = \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}^T \begin{bmatrix} \frac{\partial \psi}{\partial x} & 0 \\ \frac{\partial \psi}{\partial y} & 0 \\ 0 & \frac{\partial \psi}{\partial x} \\ 0 & \frac{\partial \psi}{\partial y} \end{bmatrix}^T \begin{bmatrix} 2\mu & 0 & 0 & 0 \\ 0 & \mu & \mu & 0 \\ 0 & \mu & \mu & 0 \\ 0 & 0 & 0 & 2\mu \end{bmatrix} \begin{bmatrix} \frac{\partial \psi}{\partial x} & 0 \\ \frac{\partial \psi}{\partial y} & 0 \\ 0 & \frac{\partial \psi}{\partial x} \\ 0 & \frac{\partial \psi}{\partial y} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{2.54}$$

and

$$\sum \lambda \int_{\Omega} tr\varepsilon(u)I : \varepsilon(\omega)dX = \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}^T \begin{bmatrix} \frac{\partial \psi}{\partial x} & 0 \\ \frac{\partial \psi}{\partial y} & 0 \\ 0 & \frac{\partial \psi}{\partial x} \\ 0 & \frac{\partial \psi}{\partial y} \end{bmatrix}^T \begin{bmatrix} \lambda & 0 & 0 & \lambda \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & \lambda \end{bmatrix} \begin{bmatrix} \frac{\partial \psi}{\partial x} & 0 \\ \frac{\partial \psi}{\partial y} & 0 \\ 0 & \frac{\partial \psi}{\partial x} \\ 0 & \frac{\partial \psi}{\partial y} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{2.55}$$

2.3.4.2 Stabilizer

Let's define the weak function on each element space

$$\mathbf{S}(k,T) = \{u = \{u_0, u_b\} : u_0 \mid \in P_k(\Omega_T^0), u_b \in P_k(E), E \subset \partial\Omega_T\} \tag{2.56}$$

where k is the order of polynomial for each function, and E is the boundary edge of each element. For each element T, we use Q_0 to represent the L^2 projection from L^T to $Pk(T)$ and Q_b is the L^2 projection from $L^2(T)$ to $Pk(\partial T)$ [13]. The local weak discrete space Q_h is

$$Q_h v = \{Q_0 u_0, Q_b u_b\} \tag{2.57}$$

A stabilizer is introduced to connect the two independent spaces, the interior and the boundary. As shown in figure 2.4, the stabilizer describes the difference of boundary values and the projected interior value to the boundary. When we project the value from interior basis function to boundary basis function, the two values are different even though they share the same location. The stabilizer is a boundary integral which measure the difference and connect the spaces.

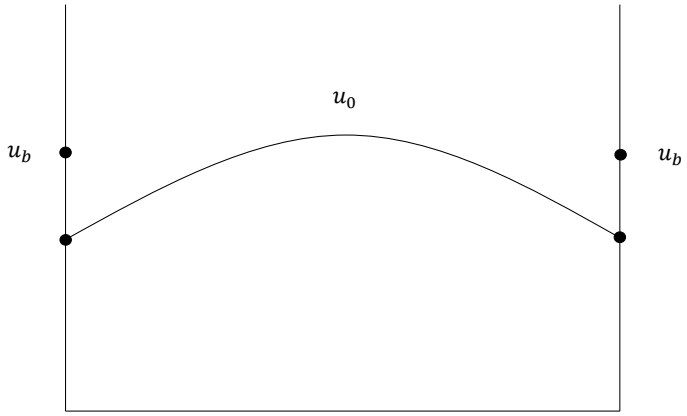


Figure 2.4: Mapping for quadrilateral elements.

The following equation is the stabilizer to measure the difference and connect the interior values to the boundary value

$$\mathbf{S}(u,v) = \sum_T h_t^{-1} \langle Q_b u_0 - u_b, Q_b \omega_0 - \omega_b \rangle_{\partial T} \tag{2.58}$$

where h_T is the characteristic length of each element.

Then we introduce the way to evaluate S. The basis function of u_0 is in reference element. At first we get the function $\xi(x,y)$ and $\eta(x,y)$ for each element.

$$\xi(x,y) = A_1 + B_1 x + C_1 y + D_1 xy \tag{2.59}$$

$$\eta(x,y) = A_2 + B_2 x + C_2 y + D_2 xy \tag{2.60}$$

Replace x and y to the polynomial of s for each edge of an element. For example, the replacement on the first edge is

$$x = (x_2 - x_1)\hat{s} + x_1 \quad on \quad e_1 \tag{2.61}$$

$$y = (y_2 - y_1)\hat{s} + y_1 \quad on \quad e_1 \tag{2.62}$$

Where (x_1, y_1) is the the coordinate of the first point and (x_2, y_2) is the the coordinate of the second point.

The stabilizer S is a matrix about the integral of $\hat{s}$

$$\mathbf{S} = h^{-1} \begin{bmatrix} \langle u_0, v_0 \rangle & -\langle u_0, v_b \rangle \\ -\langle u_b, v_0 \rangle & \langle u_b, v_b \rangle \end{bmatrix} \tag{2.63}$$

More details about calculating stabilizer matrix is shown in Appendix A.

2.3.4.3 Right Hand Side

The calculation for right hand side is the same as continuous finite element with an extra term including $\int_{\partial\Omega}[\sigma(u)\omega] \cdot nds$. Multiply e by a coefficient matrix c, we can get the stress tensor σ we need as

$$\sigma = ce = \begin{bmatrix} \sigma_{xx} \\ \tau_{xy} \\ \tau_{yx} \\ \sigma_{yy} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial \psi}{\partial x} & 0 \\ \frac{\partial \psi}{\partial y} & 0 \\ 0 & \frac{\partial \psi}{\partial x} \\ 0 & \frac{\partial \psi}{\partial y} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{2.64}$$

Chapter 3: Parallel Computing Method

In this research, we use a new version of the domain decomposition (DD) method [14] called Balancing Domain Decomposition by Constraints (BDDC) to solve large global equations. The basic idea of DD method is to divide the computational space into a number of smaller spaces of a set of non-overlapping sub-domains. Each sub-domain contains its own set of grid elements. For finite element methods, after domain decomposition, the remaining challenging task is to correctly represent the interfaces that connect these sub-domains by satisfying the continuity constraints. The DD method only considers two levels of mesh and the multiplicative coarse domain is used to correct the local grid sub-field. The finite element tearing and interconnecting (FETI) methods are a family of domain decomposition (DD) algorithms with Lagrange multipliers that have been developed during the last decade for the fast iterative solution of large-scale systems of equations arising from the finite element method of partial differential equations. A significant difference between BDDC and FETI is applying additive routines instead of multiplicative operations aforementioned. Moreover, the flexibility of for choosing constraining points on common interfaces will help significantly reduce communication cost. In this section we will discuss the difference between BDDC and FETI with more details.

3.1 Block Cholesky Elimination and Schur Complement

The Block Cholesky Elimination [10] is a decomposition of a positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose. First, we consider that M is symmetric and positive definite block matrix

$$M = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix} \tag{3.1}$$

where M is the global stiffness matrix, A is the square matrix represents the interior values of each subdomain, C is the values on the interface and B represents the connectivity between A and C. We employ the block Cholesky elimination and obtain the following equation

$$\begin{bmatrix} A & B^T \\ B & C \end{bmatrix} = \begin{bmatrix} I_A & 0 \\ BA^{-1} & I_C \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & C - BA^{-1}B^T \end{bmatrix} \begin{bmatrix} I_A & A^{-1}B^T \\ 0 & I_C \end{bmatrix} \tag{3.2}$$

Both I_A and I_C are identity matrices. The Schur complement is represented as

$$S = C - BA^{-1}B^T \tag{3.3}$$

The inverse of a symmetric, positive definite block matrix can be represented as

$$\begin{bmatrix} A & B^T \\ B & C \end{bmatrix}^{-1} = \begin{bmatrix} I_A & -A^{-1}B^T \\ 0 & I_C \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I_A & 0 \\ -BA^{-1} & I_C \end{bmatrix} \tag{3.4}$$

3.2 FETI Method and FETI-DP Method

The FETI method is a numerically scalable domain decomposition methods with Lagrange multipliers for the iterative solution of second-order linear elasticity system which was first proposed by Farhat. FETI method partitioned the entire computational domain into two level sub-domains, the coarse grid and fine grid. A Lagrangian multiplier is employed to resolve the singularity for the matrix of each sub-domain after the process of domain decomposition. In this section, we use an example of a two-dimensional partitioned sub-domain case with the Dirichlet boundary condition on the boundary. The original geometry and computational domain is shown in Figure 3.1.

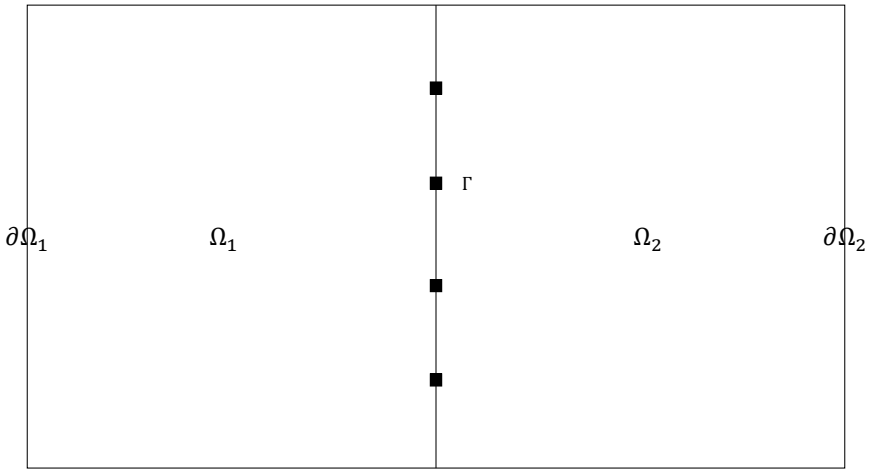


Figure 3.1: Domain decomposition in FETI

The original governing equation for each sub-domain in matrix form is

$$\begin{bmatrix} A_{II}^{(j)} & A_{I\Gamma}^{(j)} \\ A_{\Gamma I}^{(j)} & A_{\Gamma\Gamma}^{(j)} \end{bmatrix} \begin{bmatrix} u_I^{(j)} \\ u_\Gamma^{(j)} \end{bmatrix} = \begin{bmatrix} f_I^{(j)} \\ f_\Gamma^{(j)} \end{bmatrix} \tag{3.5}$$

where j = 1,2.

The boundary condition along $\partial\Omega$ is Dirichlet condition, a homogeneous Neumann condition is applied on Γ. The global problem after assemblage is

$$\begin{bmatrix} A_{II}^{(1)} & 0 & A_{I\Gamma}^{(1)} \\ 0 & A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(1)} & A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_\Gamma \end{bmatrix} = \begin{bmatrix} f_I^{(1)} \\ f_I^{(2)} \\ f_\Gamma \end{bmatrix} \tag{3.6}$$

where $A_{\Gamma\Gamma} = A_{\Gamma\Gamma}^{(1)} + A_{\Gamma\Gamma}^{(2)}$ and $f_\Gamma = f_\Gamma^{(1)} + f_\Gamma^{(2)}$. Unknown variables are decomposed into two computational sub-domains which are represented by $u_I^{(1)}$, $u_I^{(2)}$ and u_Γ corresponding to the domains Ω_1, Ω_2 and Γ respectively. Since the interior matrices are positive definite, we can eliminate the interior unknown variables by Schur complement operators. The interface

unknowns have the new form as

$$S^{(j)} = A_{\Gamma\Gamma}^{(j)} - A_{\Gamma I}^{(j)} A_{II}^{(j)-1} A_{I\Gamma}^{(j)} \tag{3.7}$$

$$g_{\Gamma}^{(j)} = f_{\Gamma}^{(j)} - A_{\Gamma I}^{(j)} A_{II}^{(j)-1} f_{I}^{(j)} \tag{3.8}$$

Based on the given equation $Au = f$, we can reduce the global equation into system which only include interface unknowns

$$(S^{(1)} + S^{(2)})u_{\Gamma} = g_{\Gamma}^{(1)} + g_{\Gamma}^{(2)} \tag{3.9}$$

For FETI method, we introduce the Lagrange multiplier to enforce the continuity along interfaces. The equation for each sub-domain is

$$\begin{bmatrix} A_{II}^{(j)} & A_{I\Gamma}^{(j)} \\ A_{\Gamma I}^{(j)} & A_{\Gamma\Gamma}^{(j)} \end{bmatrix} \begin{bmatrix} u_{I}^{(j)} \\ u_{\Gamma}^{(j)} \end{bmatrix} = \begin{bmatrix} f_{I}^{(j)} \\ f_{\Gamma}^{(j)} + \lambda_{\Gamma}^{(j)} \end{bmatrix} \tag{3.10}$$

where $\lambda_{\Gamma}^{(j)}$ is a virtual vector called Lagrange multiplier and $\lambda_{\Gamma}^{(1)} = -\lambda_{\Gamma}^{(2)}$. Then we can solve the Equation 3.3 and Equation 3.5.

$$g_{\Gamma}^{(j)} = f_{\Gamma}^{(j)} - A_{\Gamma I}^{(j)} A_{II}^{(j)-1} f_{I}^{(j)} \tag{3.11}$$

$$u_{\Gamma}^{(j)} = S^{(j)(-1)}(g_{\Gamma}^{(j)} + \lambda_{\Gamma}^{(j)}) \tag{3.12}$$

FETI-DP is an improved version of FETI with further dividing the interface to primal spaces and dual spaces. In FETI-DP method, Schur complement is calculated twice. We consume

that we have a linear system which can be written as

$$\begin{bmatrix} A_{II}^{(j)} & A_{1I}^{(j)} & \cdots & A_{mI}^{(j)} & \cdots & A_{lI}^{(j)} \\ A_{1I}^{(j)} & A_{11}^{(j)} & \cdots & A_{1m}^{(j)} & \cdots & A_{ll}^{(j)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{mI}^{(j)} & A_{m1}^{(j)} & \cdots & A_{mm}^{(j)} & \cdots & A_{ml}^{(j)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{lI}^{(j)} & A_{l1}^{(j)} & \cdots & A_{lm}^{(j)} & \cdots & A_{ll}^{(j)} \end{bmatrix} \begin{bmatrix} u_I^{(j)} \\ u_1^{(j)} \\ \vdots \\ u_m^{(j)} \\ \vdots \\ u_l^{(j)} \end{bmatrix} = \begin{bmatrix} f_I^{(j)} \\ f_1^{(j)} \\ \vdots \\ f_m^{(j)} \\ \vdots \\ f_l^{(j)} \end{bmatrix} \tag{3.13}$$

The variables along the interface between the two non-overlapping sub-domians can be written as

$$\begin{bmatrix} u_1^j \\ \vdots \\ u_m^{(j)} \\ \vdots \\ u_l^{(j)} \end{bmatrix} = T_E \begin{bmatrix} \hat{u}_1^j \\ \vdots \\ \hat{u}_m^{(j)} \\ \vdots \\ \hat{u}_l^{(j)} \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots \\ \vdots & \ddots & \vdots & 0 & \cdots \\ -1 & \cdots & 1 & \cdots & -1 \\ \cdots & 0 & \vdots & \ddots & \vdots \\ \cdots & 0 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \hat{u}_1^j \\ \vdots \\ \hat{u}_m^{(j)} \\ \vdots \\ \hat{u}_l^{(j)} \end{bmatrix} \tag{3.14}$$

T_E is a square matrix and each columns represents the new space of interface unknown variables. The original interface is divided into two spaces: primal space and dual space. $\hat{u}_m$ is primal space having all of the boundary information. At the same time, the boundary is continue because $u_i^{(j)} = \hat{u}_m^{(j)} + \hat{u}_i^{(j)}$. The the problem can be written as

$$T^T \begin{bmatrix} A_{II}^{(j)} & A_{1I}^{(j)} & \cdots & A_{mI}^{(j)} & \cdots & A_{lI}^{(j)} \\ A_{1I}^{(j)} & A_{11}^{(j)} & \cdots & A_{1m}^{(j)} & \cdots & A_{ll}^{(j)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{mI}^{(j)} & A_{m1}^{(j)} & \cdots & A_{mm}^{(j)} & \cdots & A_{ml}^{(j)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{lI}^{(j)} & A_{l1}^{(j)} & \cdots & A_{lm}^{(j)} & \cdots & A_{ll}^{(j)} \end{bmatrix} T \begin{bmatrix} \hat{u}_I^{(j)} \\ \hat{u}_1^{(j)} \\ \vdots \\ \hat{u}_m^{(j)} \\ \vdots \\ \hat{u}_l^{(j)} \end{bmatrix} = T^T \begin{bmatrix} f_I^{(j)} \\ f_1^{(j)} \\ \vdots \\ f_m^{(j)} \\ \vdots \\ f_l^{(j)} \end{bmatrix} \tag{3.15}$$

where

$$T = \begin{bmatrix} I & 0 \\ 0 & T_E \end{bmatrix} \tag{3.16}$$

Unknown variables on each sub-domain have been changed into primal space which only include connection information along interfaces. The other unknowns are changed into dual space including interior unknowns and the rest interface unknowns. The global equation is rewritten as

$$\begin{bmatrix} A_{II}^{(1)} & A_{I\Delta}^{(1)} & 0 & 0 & A_{I\Pi}^{(1)} & 0 \\ A_{\Delta I}^{(1)} & A_{\Delta\Delta}^{(1)} & 0 & 0 & A_{\Delta\Pi}^{(1)} & B_{\Delta}^{(1)} \\ 0 & 0 & A_{II}^{(2)} & A_{I\Delta}^{(2)} & A_{I\Pi}^{(1)} & 0 \\ 0 & 0 & A_{\Delta I}^{(2)} & A_{\Delta\Delta}^{(2)} & A_{\Delta\Pi}^{(2)} & B_{\Delta}^{(2)} \\ A_{\Pi I}^{(1)} & A_{\Pi\Delta}^{(1)} & A_{\Pi I}^{(2)} & A_{\Pi\Delta}^{(2)} & A_{\Pi\Pi}^{(1)} + A_{\Pi\Pi}^{(2)} & 0 \\ 0 & B_{\Delta}^{(1)} & 0 & B_{\Delta}^{(2)} & 0 & 0 \end{bmatrix} = \begin{bmatrix} u_I^{(1)} \\ u_{\Delta}^{(1)} \\ u_I^{(2)} \\ u_{\Delta}^{(2)} \\ u_{\Pi} \\ \lambda \end{bmatrix} \begin{bmatrix} f_I^{(1)} \\ f_{\Delta}^{(1)} \\ f_I^{(2)} \\ f_{\Delta}^{(2)} \\ f_{\Pi}^{(1)} + f_{\Pi}^{(2)} \\ 0 \end{bmatrix} \tag{3.17}$$

where λ is Lagrange multiplier which is an external unknown vector. After tearing, the unknown variables have new format in the dual space that $u_{\Delta}^{(1)} \neq u_{\Delta}^{(2)}$. To maintain the continuity an assembled residual of resulting vector is needed. The residue is mapped into the appropriate space of enforced vector on the right-hand side of the equation. Then we can use the residue vector to correct the solution and gradually obtain the final results.

3.3 WG-BDDC Method

In this section, we discuss WG-BDDC method in details. Local matrices after domain decomposition are singular and the pseudo-inverses is needed in FETI method, but the WG-BDDC method has an advantage to cover this difficulty. The preconditioned conjugate gradient (PCG) method is employed to solve a global coarse problem. WG-BDDC method is processed by the following steps:

- Divide the unknowns of the interface into primal and dual sub-spaces and calculate

the preconditioner S_{BDDC}.

- Solve the linear system for unknowns in primal space by PCG method.
- Calculate the other unknowns by a recovery process.

3.3.1 Primal and Dual Spaces

WG method is a discontinuous finite element method, so in WG-BDDC method, primal constraints are introduced over edges/faces. As shown in Figure 3.2 , we partition the interface space Γ into two sub-spaces, i.e, primal space Π and dual space Δ. Unknowns in primal space are represented by solid squares which are the constraints between each sub-domain. Unknowns in dual space are represented by hollow squares which are variables local to each subspace. The information from hollow squares in dual space to those in primal

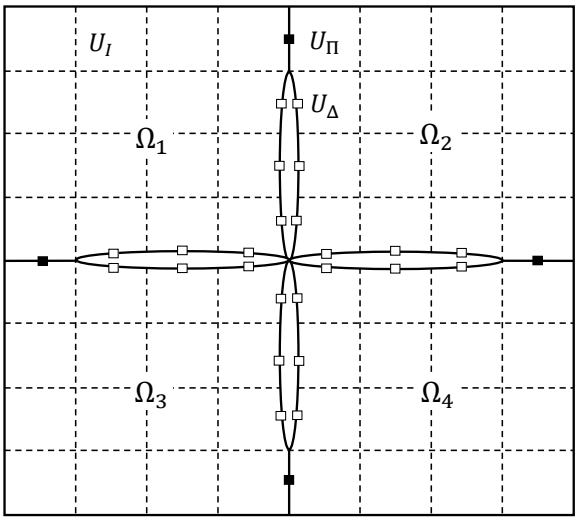


Figure 3.2: BDDC computational domain

space is connected by preconditioner. After using weak Galerkin finite element method, we

have gotten the equations in matrix form for each subdomain which is

$$\begin{bmatrix} A_{II}^{(i)} & A_{I\Delta}^{(i)} & A_{I\Pi}^{(i)} \\ A_{\Delta I}^{(i)} & A_{\Delta\Delta}^{(i)} & A_{\Delta\Pi}^{(i)} \\ A_{\Pi I}^{(i)} & A_{\Pi\Delta}^{(i)} & A_{\Pi\Pi}^{(i)} \end{bmatrix} \begin{bmatrix} u_I^{(i)} \\ u_\Delta^{(i)} \\ u_\Pi^{(i)} \end{bmatrix} = \begin{bmatrix} f_I^{(i)} \\ f_\Delta^{(i)} \\ f_\Pi^{(i)} \end{bmatrix} \tag{3.18}$$

We define the subscript c represents the unknown variables in primal space after assemble and the subscript r denotes the remaining unknown variable, where

$$A_{cc} = \sum_{i=1}^{N} A_{\Pi\Pi}^{(j)} \tag{3.19}$$

$$A_{rr}^{(j)} = \begin{bmatrix} A_{II}^{(j)} & A_{I\Delta}^{(j)} \\ A_{\Delta I}^{(j)} & A_{\Delta\Delta}^{(j)} \end{bmatrix} \tag{3.20}$$

The global equation is rewritten as

$$\begin{bmatrix} A_{rr}^{(1)} & 0 & 0 & \cdots & 0 & A_{rc}^{(1)} \\ 0 & A_{rr}^{(2)} & 0 & \cdots & 0 & A_{rc}^{(2)} \\ 0 & 0 & A_{rr}^{(3)} & \cdots & 0 & A_{rc}^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_{rr}^{(N)} & A_{rc}^{(N)} \\ A_{cr}^{(1)} & A_{cr}^{(2)} & A_{cr}^{(3)} & \cdots & A_{cr}^{(N)} & A_{cc} \end{bmatrix} \begin{bmatrix} u_r^{(1)} \\ u_r^{(2)} \\ u_r^{(3)} \\ \vdots \\ u_r^{(N)} \\ u_c \end{bmatrix} = \begin{bmatrix} f_r^{(1)} \\ f_r^{(2)} \\ f_r^{(3)} \\ \vdots \\ f_r^{(N)} \\ f_c \end{bmatrix} \tag{3.21}$$

The implementation of the WG-BDDC algorithm is presented in the following manner

$$S_{BDDC}^{-1} = \hat{R}_{D,\Gamma}^T \{ R_{\Gamma,\Delta}^T (\sum_{j=1}^{N} \begin{bmatrix} 0 & R_\Delta^{(j)^T} \end{bmatrix} \begin{bmatrix} A_{II}^{(j)} & A_{I\Delta}^{(j)} \\ A_{\Delta I}^{(j)} & A_{\Delta\Delta}^{(j)} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ R_\Delta^{(j)} \end{bmatrix}) R_{\Gamma,\Delta} + \Phi S_\Pi^{-1} \Phi^T \} \hat{R}_{D,\Gamma} \tag{3.22}$$

The matrix Φ is the connection between Schur complement S and global stiffness matrix. We assemble the global matrix from multiple non-overlapped sub-domains and the mapping

matrix R is applied on each sub-domain.

$$\Phi = R_{\Gamma\Pi}^T - R_{\Gamma\Delta}^T \sum_{j=1}^{N} \begin{bmatrix} 0 & R_{\Delta}^{(j)^T} \end{bmatrix} \begin{bmatrix} A_{II}^{(j)} & A_{I\Delta}^{(j)} \\ A_{\Delta I}^{(j)} & A_{\Delta\Delta}^{(j)} \end{bmatrix}^{-1} \begin{bmatrix} A_{\Pi I}^{(j)^T} \\ A_{\Pi\Delta}^{(j)^T} \end{bmatrix} R_{\Pi}^{(j)} \tag{3.23}$$

The matrix S_Π is the global coarse system matrix

$$S_\Pi = \sum_{j=1}^{N} R_{\Pi}^{(j)^T} \{A_{\Pi\Pi} - \begin{bmatrix} A_{\Pi I}^{(j)} & A_{\Pi\Delta}^{(j)} \end{bmatrix} \begin{bmatrix} A_{II}^{(j)} & A_{I\Delta}^{(j)} \\ A_{\Delta I}^{(j)} & A_{\Delta\Delta}^{(j)} \end{bmatrix}^{-1} \begin{bmatrix} A_{\Pi I}^{(j)^T} \\ A_{\Pi\Delta}^{(j)^T} \end{bmatrix} \} R_{\Pi}^{(j)} \tag{3.24}$$

The major computational cost is on $\begin{bmatrix} A_{II}^{(j)} & A_{I\Delta}^{(j)} \\ A_{\Delta I}^{(j)} & A_{\Delta\Delta}^{(j)} \end{bmatrix}^{-1}$, which is calculated in each processor(j). Then we use conjugate gradient method to solve this linear equation.

The BDDC preconditioner has the foloowing form

$$S_{BDDC}^{-1} = R_{D,\Gamma}^T S_{\Gamma\Gamma}^{-1} R_{D,\Gamma} \tag{3.25}$$

The coarse problem after preconditioning can be written as

$$S_{\Gamma BDDC}^{-1} S_{\Gamma\Gamma} u_{\Gamma\Gamma} = S_{BDDC}^{-1} f_\Gamma \tag{3.26}$$

3.3.2 Preconditioned Conjugate Gradient Method

Conjugate gradient method (CG) is an algorithm for the numerical solution of symmetric and positive-definite systems of linear equations. Unfortunately, large systems often arise when numerically solving partial differential equations or optimization problems. The CG method is often implemented as an iterative algorithm effective for solving sparse systems. The preconditioned conjugate gradient method (PCG) is a improved version of CG which has been reported by Bramble and Pasciak [9] that helps to calculate the convergence of

CG iterations to solve symmetric saddle point problems. In this research, we employed a preconditioner which is an approximation to the inverse of stiffness matrix using MPI communications. The algorithm is

Algorithm 1: Preconditioned Conjugate Gradient Method

Input:

$r_0 = b - Mu_0$

$z_0 = S_{BDDC}^{-1} r_0$

$p_0 = z_0$

k=0

Repeat;

$\alpha_k = \frac{r_k^T z_k}{p_k^T A p_k}$

$u_{k+1} = u_k + \alpha_k p_k$

$r_{k+1} = r_k - \alpha_k A p_k$

if r_{k+1} *is sufficiently small* **then**

    return u_{k+1}

else

    $z_{k+1} = S_{BDDC}^{-1} r_{k+1}$

    $\beta_k = \frac{z_{k+1}^T r_{k+1}}{z_k^T r_k}$

    $p_{k+1} = z_{k+1} + \beta_k p_k$

    k=k+1

end

where r is the residue vector, z is a operator with preconditioner, p is the orthogonal direction vector and k is the iteration number.

3.3.3 Local Information Recovery

The solution on the interface, u_Π, is obtained from the PCG solver. The local interior unknowns u_r is calculated by each processor, and the relationship between interior unknowns

and interface unknowns is

$$A_{rr}^{(j)} u_r^{(j)} + (A_{\Pi r}^{(j)})^T u_\Pi^{(j)} = f_r^{(j)} \tag{3.27}$$

We have got the solutions of u_c and $f_r^{(j)}$, then the local information u_r is calculated by

$$u_r^{(j)} = (A_{rr}^{(j)})^{-1} (f_r^{(j)} - (A_{\Pi r}^{(j)})^T u_\Pi^{(j)}) \tag{3.28}$$

Chapter 4: Results and Discussion

The performance of the WG-BDDC method with triangular elements is validated. However, it has not been tested on all quadrilateral elements. Therefore, a WG-BDDC solver which use all quadrilateral element meshes is developed. Several tests were made to verify the order of accuracy and the results are revealed in Section 4.1. Finally, the WG-BDDC solvers is tested on multiple processors from 4 to 144 cores in order to demonstrate the scalability of such a parallel algorithm by solving the same problem in Section 2.1.

4.1 Accuracy Verification

4.1.1 2D Linear Elasticity Equation

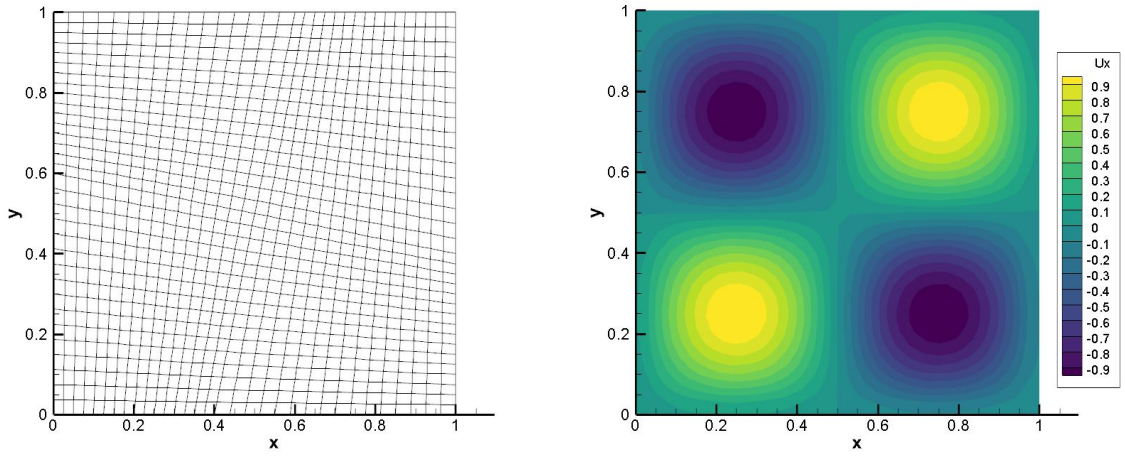
The 2D linear elsticity equation is tested using the following governing equation

$$\begin{aligned} -\boldsymbol{\nabla}\cdot\boldsymbol{\sigma} &= \boldsymbol{f}, \qquad in \quad \Omega \\ \boldsymbol{u} &= 0, \qquad on \quad \partial\Omega \end{aligned} \tag{4.1}$$

The computational domain is $[0,0]\times[1,1]$. To verify the order of accuracy of the WG method on all quadrilateral meshes, tests were conducted on structured grids. The exact solution is given by

$$u = \begin{bmatrix} sin(2\pi x)sin(2\pi y) \\ 1 \end{bmatrix} \tag{4.2}$$

Samples of meshes and contour plots are presented in Figure 4.1.



(a) Structured mesh elements

(b) Contour plot the x-component displacements

Figure 4.1: WG method plotting using structured mesh

To verify the second-order accuracy, we use five different grids as shown in Table 4.1. By measuring L^2 errors on each grid, the orders of numerical accuracy are reported four times. As grid is refined, the order of accuracy approaches the optional order, which is 2. However, the order of accuracy obtained based the coarsest two grids is only 1.42. The reason is the basis functions of interior unknowns are linear rather than bi-linear, which cause the stabilizer term has not enough effect on connecting unknowns on boundaries and interior unknowns.

$number\ of\ elements$	$L^2-error$	$order$
16	7.998e-01	-
64	2.993e-01	1.418
256	8.567e-02	1.804
1024	2.217e-02	1.949
4096	5.575e-03	1.991

Table 4.1: L^2 errors for the second order WGFEM on structured grids

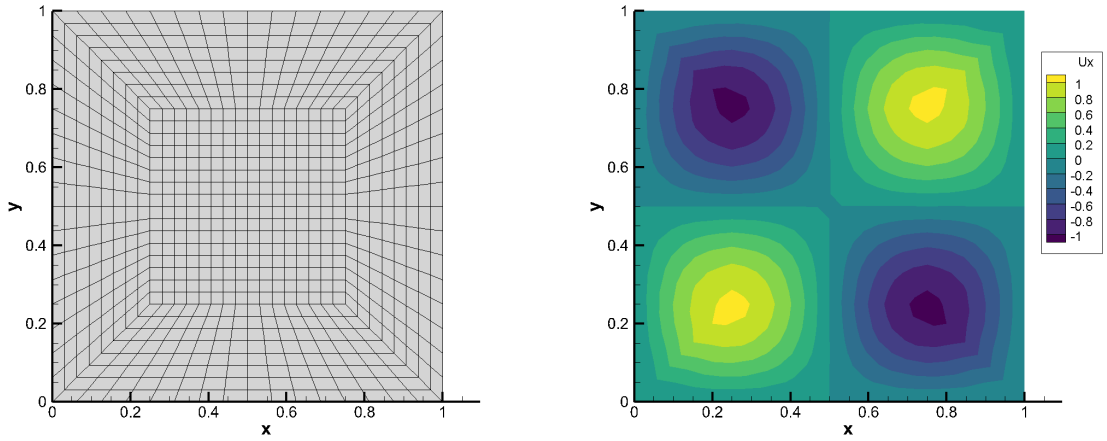
As to the third order tests, the results are shown in Table 4.2, which are similar to the traditional continuous finite element method. Six basis functions are employed for interior

unknowns which gives a high accuracy contacting with unknowns on boundaries and interior unknowns.

$number\ of\ elements$	$L^2-error$	$order$
16	8.217e-02	-
64	6.743e-03	3.606
256	7.176e-04	3.232
1024	8.613e-05	3.058
4096	1.065e-05	3.015

Table 4.2: L^2 errors for the third order WGFEM on structured grids

To verify the second-order accuracy on unstructured meshes, we use three different grids as shown in Table 4.3. Sample of meshes and counter plot of x-component displacement are presented in Figure 4.2.



(a) Unstructured mesh elements

(b) Contour plot the x-component displacements

Figure 4.2: WG method plotting using structured mesh

$number\ of\ elements$	$L^2 - error$	$order$
48	2.523e-01	-
192	8.914e-02	1.500
768	3.067e-02	1.648

Table 4.3: L^2 errors for the second order WGFEM on unstructured grids

4.1.2 Beam Deformation Tests

We also make beam deformation tests for linear elasticity equation using both structured and unstructured grids. The solver is used to simulate a beam deformation test problem on a square box. In this problem, a uniform load is applied on the right boundary $(x = 1)$. Essential boundary conditions $(u = 0)$ are imposed on bottom and left sides. On the top side, a natural boundary condition is specified. Bickford[7] solved the same problem by continuous finite element method as an example in his book. The grid obtained on a structured mesh is shown in Figure 4.3.

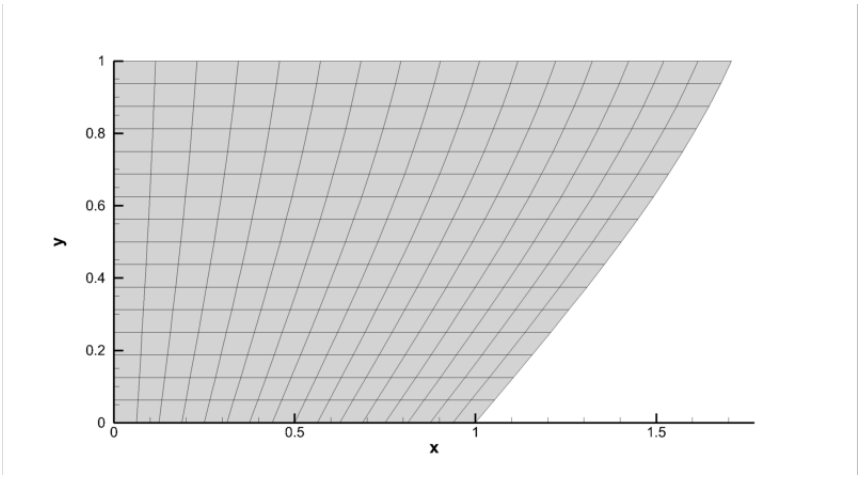


Figure 4.3: Grid deformation after simulation using a structured mesh

The contour plot of beam deformation on such a structured mesh after post processing is shown in Figure 4.4.

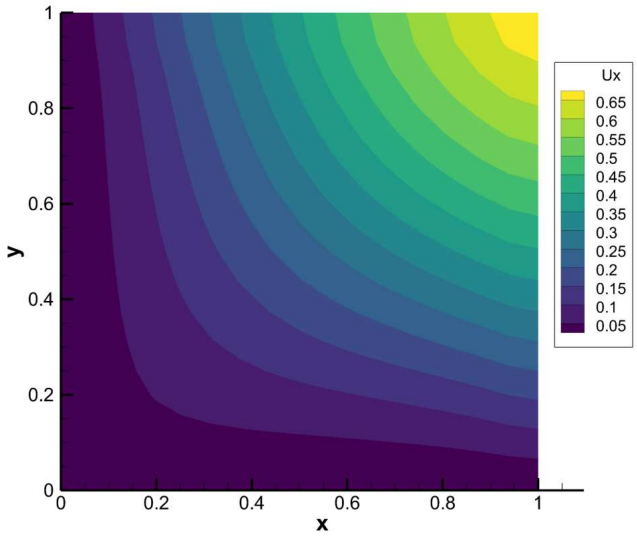


Figure 4.4: Contour plot the x-component displacements on a structured mesh

We solve the same problem in another grid to test the performance of WGFEM solver in unstructured grids. The grid deformation using the displacements obtained from the simulations on the unstructured mesh is shown in Figure 4.5.

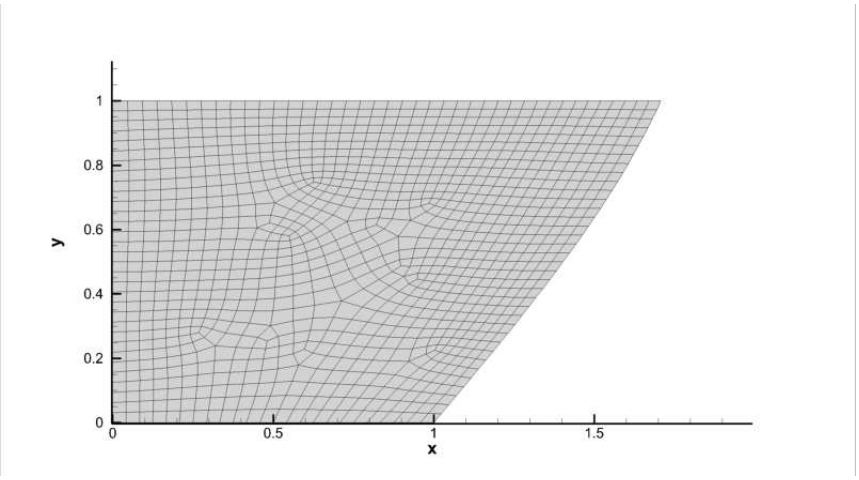


Figure 4.5: Grid deformation after simulation using an unstructured mesh

The contour plot of the x-component displacements on the unstructured mesh is shown in Figure 4.6.

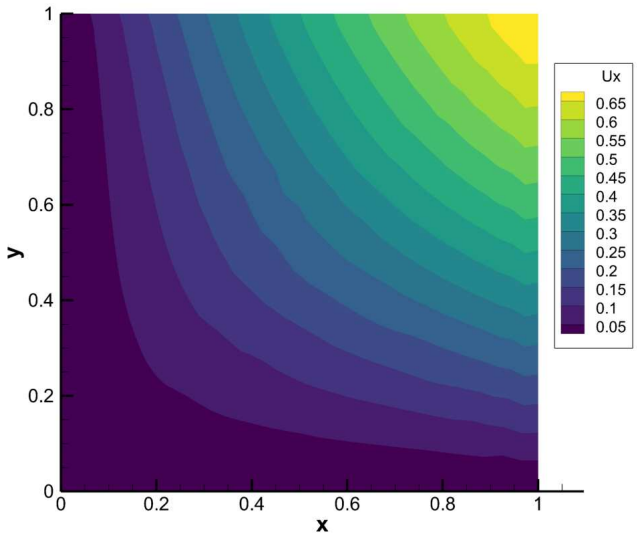
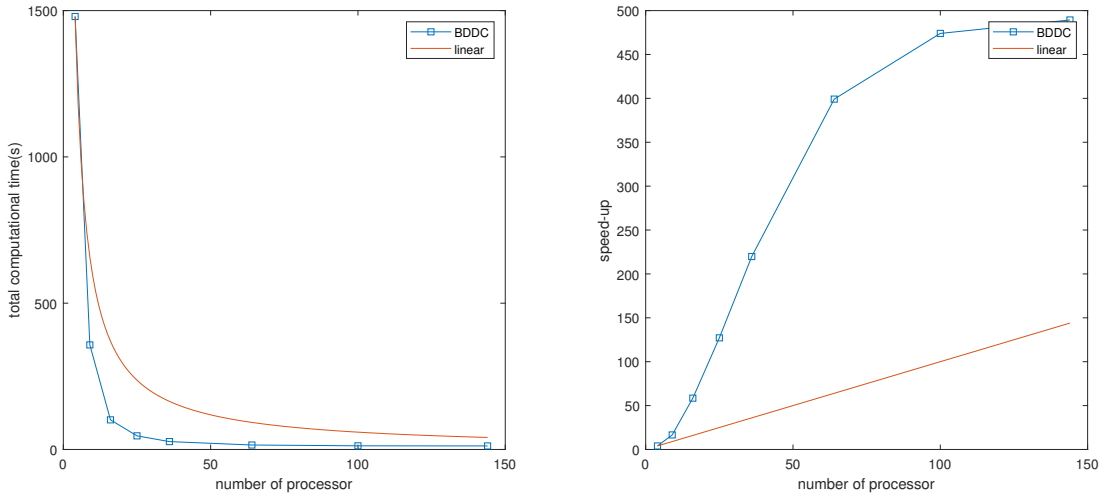


Figure 4.6: Contour plot the x-component displacements on an unstructured mesh

By comparing with the solution in Bickford's book, our results calculated by WGFEM solver are correct.

4.2 Scalability and Speed-up Tests

Finally, the WG-BDDC solvers is tested on multiple processors from 4 to 64 cores in order to demonstrate the scalability of such a parallel algorithm by solving the same problem in section 4.1. In our research, all sub-domains considered here are rectangular. The total number of elements for a square-box computational domain is 20,736. Figure 4.7 shows the total computational time on different numbers of processors.



(a) Total time

(b) Speed-up

Figure 4.7: Scalability of WGFEM method

Comparing to the linear scability, the line of WG-BDDC clearly shows a super-linear speed up and the slope of the WG-BDDC speed-up curve decreases as the number of processors increases from 36 to 64. As we discussed before, the most cost is on the Cholesky elimination of matrix contribution and displacement recovery. If we do not consider the cost on MPI and PCG iteration, speed-up is expected to be n^3 which is much more than linear speed-up. The more sub-domains are used, the higher the cost of PCG iterations is. Let us assume the number of all unknowns is N, and the number of unknowns for each sub-domain is n. For each sub-domain, the cost on Cholesky elimination is $O(n^3)$ and the cost of PCG iteration is $O(N^2/n)$. Therefore, when the ratio of N and n is large, $O(n^3)$ is much bigger

than $O(N^2/n)$ and the speed-up is super-linear. For the above problem with total 20,736 elements, when the number of total elements for each sub-domain is 144, it is expected that the cost PCG iterations for the global problem will be as expensive as the cost of the Cholesky Elimination for the local problem being solved on each processor.

Chapter 5: Conclusions and Future Work

In this research, a WGFEM solver has been successfully implemented on both structured and unstructured grids involving all quadrilateral cells. Subsequently, we successfully created a new parallel algorithm, WG-BDDC. Moreover, a parallel high-order solver using weak Galerkin finite element method (WG-BDDC) is successfully extended to simulate beam deformation using unstructured grids. We successfully verified its solution accuracy by using Fortran and MPI libraries on parallel computers with up to 144 cores on both structured grids and unstructured grids. In the future, this parallel WG-BDDC solver can be extended to handle three-dimensional geometries for large-scale engineering structural mechanics problems.

Appendix A: Stabilizer

The basis function of u_0 is in reference element. At first we get the function $\xi(x,y)$ and $\eta(x,y)$ for each element.

$$\xi(x,y) = A_1 + B_1 x + C_1 y + D_1 xy \tag{A.1}$$

$$\eta(x,y) = A_2 + B_2 x + C_2 y + D_2 xy \tag{A.2}$$

The coefficients are calculated by

$$\begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \\ 1 & x_3 & y_3 & x_3 y_3 \\ 1 & x_4 & y_4 & x_4 y_4 \end{bmatrix} \begin{bmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \end{bmatrix} \tag{A.3}$$

$$\begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \\ 1 & x_3 & y_3 & x_3 y_3 \\ 1 & x_4 & y_4 & x_4 y_4 \end{bmatrix} \begin{bmatrix} A_2 \\ B_2 \\ C_2 \\ D_2 \end{bmatrix} \tag{A.4}$$

Then replace x and y to the polynomial of s

$$x = (x_2 - x_1)\hat{s} + x_1 \quad on \quad e_1 \tag{A.5}$$

$$x = (x_3 - x_2)\hat{s} + x_2 \quad on \quad e_2 \tag{A.6}$$

$$x = (x_4 - x_3)\hat{s} + x_3 \quad on \quad e_3 \tag{A.7}$$

$$x = (x_1 - x_4)\hat{s} + x_4 \quad on \quad e_4 \tag{A.8}$$

$$y = (y_2 - y_1)\hat{s} + y_1 \quad on \quad e_1 \tag{A.9}$$

$$y = (y_3 - y_2)\hat{s} + y_2 \quad on \quad e_2 \tag{A.10}$$

$$y = (y_4 - y_3)\hat{s} + y_3 \quad on \quad e_3 \tag{A.11}$$

$$y = (y_1 - y_4)\hat{s} + y_4 \quad on \quad e_4 \tag{A.12}$$

The stabilizer is a matrix about the integral of $\hat{s}$

$$\mathbf{S} = h^{-1} \begin{bmatrix} \langle u_0, v_0 \rangle_{\partial\Omega} & -\langle u_0, v_b \rangle_{\partial\Omega} \\ -\langle u_b, v_0 \rangle_{\partial\Omega} & \langle u_b, v_b \rangle_{\partial\Omega} \end{bmatrix} \tag{A.13}$$

Appendix B: One-dimensional Weak Galerkin Method

B.1 Transfer Global Equation into Weak Form

For a one-dimensional elasticity problem, we have a global equation as

$$\frac{d^2u}{dx^2} = f \tag{B.1}$$

In the finite element method, we construct a weak formulation, by multiplying both sides of the differential equation by a test function $v(x)$ and integrating from 0 to L.

$$\int_0^L \frac{d^2u}{dx^2} v dx = \int_0^L f v dx \tag{B.2}$$

Through integration by parts and applying the nature boundary condition, we rewrite the global equation(B.1) into weak form

$$\int_0^L \frac{du}{dx}\frac{dv}{dx} dx = -\int_0^L \hat{f} v dx \tag{B.3}$$

B.2 Weak Gradient Operator

Similar to standard Galerkin method, we multiply one arbitrary test function q on it and integrate over the element. The weak gradient operator is $\frac{d_w u}{dx}$.

$$\int_{\Omega_e} \frac{d_w u}{dx} q dx = -\int_{\Omega_e} u_0 \frac{d_w q}{dx} dx + \int_{\partial\Omega_e} u_b (q \cdot n) ds \tag{B.4}$$

The basis function for u_0 is

$$\psi_0^1 = 1 \tag{B.5}$$

The basis function for u_b is

$$\psi_b^1 = \begin{cases} 1, & x = x_{n-1} \\ 0, & else \end{cases} \qquad \psi_b^2 = \begin{cases} 1, & x = x_n \\ 0, & else \end{cases} \tag{B.6}$$

The basis function for weak gradient operator $\frac{d_w u}{dx}$ is

$$\psi_w = 1 \tag{B.7}$$

First we calculate the weak gradient of u_0 and u_b by equation(B.4).

$$\frac{d_w u_0}{dx} = 0 \tag{B.8}$$

$$\frac{d_w u_b^1}{dx} = -1 \tag{B.9}$$

$$\frac{d_w u_b^2}{dx} = -1 \tag{B.10}$$

B.3 Stabilizer

The stabilizer is calculated in the same way as that of 2-D problem, which is

$$S = h^{-1} \begin{bmatrix} \langle u_0, v_0 \rangle & -\langle u_0, v_b \rangle \\ -\langle u_b, v_0 \rangle & \langle u_b, v_b \rangle \end{bmatrix} \tag{B.11}$$

B.4 Solution for 1-D Example

A bar which has a cross sectional area A and a Young's modulus E will be considered. The bar is under tension which is caused by the force F. We apply the conditions which are $E = 1, A = 1, F = 10, L = 1$ as figure B.1. A nature boundary condition is applied on the

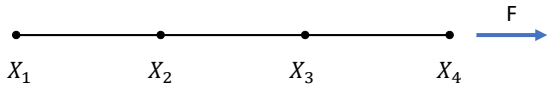


Figure B.1: Mesh for 1-D problem

left side. The tensor product of weak gradient of each element is

$$K = \frac{EA}{h}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \tag{B.12}$$

where h is the length of each element. After assembling and applying the boundary condition, the tensor product of weak gradient in matrix form is

$$K = \frac{EA}{h}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{B.13}$$

The stabilizer for each element is

$$S = h^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{B.14}$$

After assembling, the stabilizer in matrix form is

$$S = h^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.15}$$

Then we solve the global equation in matrix form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_0^1 \\ u_0^2 \\ u_0^3 \\ 0 \\ u_b^2 \\ u_b^3 \\ u_b^4 \end{bmatrix} = \begin{bmatrix} 1.667 \\ 5.0 \\ 8.333 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix} \tag{B.16}$$

The results are

$$u = \begin{bmatrix} u_0^1 \\ u_0^2 \\ u_0^3 \\ u_b^1 \\ u_b^2 \\ u_b^3 \\ u_b^4 \end{bmatrix} \begin{bmatrix} 1.667 \\ 5.0 \\ 8.333 \\ 0.0 \\ 3.333 \\ 6.667 \\ 10.0 \end{bmatrix} \tag{B.17}$$

Bibliography

[1] LAPACK 3.8.0 Release Note. `http://www.netlib.org/lapack/lapack-3.8.0.html`. 2017.

[2] William B BickFord. *A First Course In The finite element method*. Richard d Irwin, 1994.

[3] Farhat Charbel, Lesoinne Michel, LeTallec Patrick, Pierson Kendall, and Rixen Daniel. FETI-DP: a dual–primal unified FETI method—part I: A faster alternative to the two-level FETI method. *International Journal for Numerical Method in Engineering*, 50(7):1523–1544, 2001.

[4] Clark R. Dohrmann. A Preconditioner for Substructuring Based on Constrained Energy Minimization. *SIAM J. Sci. Comput.*, 25(1):246–258, 2003.

[5] C. Farhat. A lagrange multiplier based divide and conquer finite element algorithm. *Computing Systems in Engineering*, 2(2):149 – 156, 1991.

[6] Charbel Farhat and François-Xavier Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32:1205 – 1227, 1991.

[7] Alan George, Michael Heath, Joseph Liu, and Esmond Ng. Sparse Cholesky Factorization on a Local-Memory Multiprocessor. *Siam Journal on Scientific and Statistical Computing*, 9, 1988.

[8] Toshio Horiuchi. The imbedding theorems for weighted Sobolev spaces. *J. Math. Kyoto Univ.*, 29(3):365–403, 1989.

[9] A. Knyazev. Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method. *SIAM Journal on Scientific Computing*, 23(2):517–541, 2001.

[10] Jing Li and Olof B. Widlund. FETI-DP, BDDC, and block Cholesky methods. *International Journal for Numerical Method in Engineering*, 66(2):250–271, 2006.

[11] Yujie Liu and Junping Wang. A Simplified Weak Galerkin Finite Element Method: Algorithm and Error Estimates. *Numerical Analysis*, 2018.

[12] Jonathan R Shewchuk. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Technical report, Pittsburgh, PA, USA, 1994.

[13] Lunji Song, Kaifang Liu, and Shan Zhao. A Weak Galerkin Method with an Over-Relaxed Stabilization for Low Regularity Elliptic Problems. *Journal of Scientific Computing*, 71(1):195–218, 2017.

[14] Dolean Victorita, Jolivet Pierre, and Nataf Frédéric. *An Introduction to Domain Decomposition Methods, Chapter 6: Neumann–Neumann and FETI algorithms*, pages 131–159, 2016.

[15] Junping Wang and Xiu Ye. A Weak Galerkin Finite Element Method for Second-Order Elliptic Problems. *Journal of Computational and Applied Mathematics*, 241(2):103–115, 2013.