· U C ·

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

DEPARTAMENTO DE
ENGENHARIA MECÂNICA

DD3TRIM: Revised and Augmented Version

Submitted in Partial Fulfilment of the Requirements for the Degree of Master in Mechanical Engineering in the speciality of Production and Project

Author

Carlos Miguel Afonso Diogo

Advisors

Marta Cristina Cardoso de Oliveira
Diogo Mariano Simões Neto

Jury

President	**Professor Doutor Luís Filipe Martins Menezes** **Professor Catedrático da Universidade de Coimbra**
Vowel	**Professor Doutor José Luís de Carvalho Martins Alves** **Professor Auxiliar da Universidade de Minho**
Advisor	**Professor Doutor Diogo Mariano Simões Neto** **Professor Convidado da Universidade de Coimbra**

Coimbra, July, 2015

I do not think there is any thrill that can go through the human heart like that felt by the inventor as he sees some creation of the brain unfolding to success... such emotions make a man forget food, sleep, friends, love, everything.

[Nikola Tesla]

ACKNOWLEDGEMENTS

I would like to express my profound and sincere gratitude to:

Professor Marta Cristina Cardoso de Oliveira,
for giving me the possibility to experiment ideas and everything I could devise,
for the support and discussions during the execution of this work,
for the motivation and for asking the right questions.

Professor Diogo Mariano Simões Neto,
for his support and discussion during the execution of this work,
for the motivation and welcomed meticulousness.

To my family, in particular to my mother,
that provided me with the conditions to focus on what was really important,
for the understanding during my long absences.

To my friends,
who directly or indirectly provided support and companionship during the long work days and even longer work nights.

This research work was sponsored by national funds from the Portuguese Foundation for Science and Technology (FCT) via the projects PTDC/EMS-TEC/1805/2012 and PEst-C/EME/UI0285/2013 and by FEDER funds through the program COMPETE – Programa Operacional Factores de Competitividade, under the project CENTRO -07-0224 -FEDER -002001 (MT4MOBI).

Abstract

The sheet metal forming processes are widely used in several production areas, namely in the automotive industry. The growing geometrical complexity of the components and the shortening of development cycles leads to new challenges in the design of sheet metal forming processes. The multi-step manufacturing of formed parts, including trimming operations, has been the solution to obtain complex geometries without defects. In this context, the virtual production using the finite element numerical simulation has been successfully applied to address these challenges. Specifically in multi-step forming operations, the numerical simulation comprises both trimming and remapping. The DD3TRIM code was developed, in CEMUC, to perform trimming operations on solid finite element meshes. Nevertheless, the continuous updating of the in-house finite element solver DD3IMP disabled its interaction with DD3TRIM.

The main objective of this work is the reprogramming of the DD3TRIM program enabling a permanent connection with DD3IMP. Several improvements were developed and implemented in the global algorithm, mainly automation and new functionalities. The idea behind the remapping algorithm comprises the transfer of the variables between two different finite element meshes, in order to provide similar results between both. The selection of the remapping method, its parameters and zone of influence are critical issues in the algorithm. In this work, the Dual Kriging interpolation method was implemented as an alternative to the previously implemented Incremental Volumetric Remapping (IVR). Several variants were developed, differing in the number of points used for the interpolation procedure. The comparison between the remapping methods is based on the error value and computational efficiency. Based on the examples performed, the IVR and implemented forms of Dual Kriging are equivalent in terms of interpolated value, but Dual Kriging is faster. Moreover, the new version of DD3TRIM enables the continuous development of multi-stage forming processes, involving trimming operations.

Keywords Finite Element Method, Dual Kriging, Remapping, Trimming, Sheet Metal Forming, DD3TRIM.

Resumo

Os processos tecnológicos de conformação plástica de chapas são amplamente utilizados em diversas áreas de produção, nomeadamente na indústria automóvel. A complexidade crescente dos produtos e a constante redução dos ciclos de desenvolvimento colocaram novos desafios na concepção de processos de conformação plástica de chapas. A solução adoptada para produzir componentes de geometria complexa, sem defeitos, recorre a processos de fabrico multi-etapa, incluindo operações de corte. Neste contexto, o conceito de produção virtual, com recurso à simulação numérica com o método dos elementos finitos, tem sido aplicado com sucesso para enfrentar esses desafios. No caso de processos de conformação multi-etapa, a simulação numérica inclui o corte geométrico e o remapeamento. O programa DD3TRIM foi desenvolvido, no CEMUC, para fazer operações de corte em malhas de elementos finitos sólidos. No entanto, o desenvolvimento contínuo do programa DD3ImMP tornou impossível a interacção entre os dois programas.

O objectivo principal deste trabalho é a reprogramação do programa DD3TRIM, a fim de estabelecer uma interligação permanente com o programa DD3IMP. Neste contexto, foram desenvolvidas e implementadas várias melhorias no algoritmo global, principalmente de automatização e novas funcionalidades. O conceito inerente a um algoritmo de remapeamento é a transferência de variáveis entre duas malhas de elementos finitos, com o objectivo de obter resultados semelhantes entre ambas. A selecção do método de remapeamento, seus parâmetros e zona de influência são questões essenciais do algoritmo. Neste trabalho, o método de interpolação Dual Kriging foi implementado como alternativa ao método de remapeamento incremental volúmico (IVR), implementado na versão original do DD3TRIM. Foram desenvolvidas diversas variantes que diferem no número de pontos utilizados para o procedimento de interpolação. A comparação entre os métodos de remapeamento é baseada na precisão e na eficiência computacional. Os exemplos realizados mostram que o método IVR e implementações do Dual Kriging são equivalentes em termos de valor interpolado, mas o Dual Kriging é mais rápido. Além disso, a nova versão do DD3TRIM permite o desenvolvimento contínuo de processos multi-etapa de conformação envolvendo operações de corte.

Palavras-chave: Método dos Elementos Finitos, Dual Kriging, Remapeamento, Corte Geométrico, Estampagem, DD3TRIM.

Contents

LIST OF FIGURES

LIST OF TABLES

MATHEMATICAL NOTATION AND ACRONYMS

Mathematical Notation

$\mathbb{R}$	Set of real numbers
$\mathbf{0}$	Null tensor
$a, \dots, z; A, \dots, Z; \alpha, \dots, \omega$	Scalars in $\mathbb{R}$
$\boldsymbol{a}, \dots, \boldsymbol{z}; \boldsymbol{A}, \dots, \boldsymbol{Z}; \boldsymbol{\alpha}, \dots, \boldsymbol{\omega}$	Vectors or tensors in $\mathbb{R}^n$
θ	Angle in the XY plane, measured from a GP, in the scalar testing mesh
λ	Weighting factor
a	Radius of the mesh in the scalar testing mesh
$\mathrm{C}_0, \mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3$	Constants for Dual Kriging Interpolation
E	Error function in the scalar testing benchmark
F	Kriging interpolation function
I	Interpolated value function
K	Covariance function
n	Total number of interpolation points considered Dual Kriging.
r	Radius of a GP in the scalar testing mesh
T	Analytical function in the scalar testing benchmark
X_0, Y_0, Z_0	Coordinates of the destination Gauss Point
$\mathbf{f}$	Kriging method’s right-hand vector
$\mathbf{h}_{0\mathrm{j}}$	Vector with Euclidean distance between point 0 and *j*
$\mathbf{K}$	Kriging Matrix
$\mathbf{u}$	Coefficients vector

Acronyms

DD3IMP	Finite element solver. Contraction of Deep Drawing 3D Implicit Code
DD3TRIM	Contraction of Deep Drawing 3D Trimming Code
DEM	Departamento de Engenharia Mecânica
DK	Dual Kriging
DK-*1*	Dual Kriging Method: Only inside the element
DK-MSx	Dual Kriging Method: Master-Slave Variant X, in which X is an integer
DK-MSxH	Dual Kriging Method Master Slave Hybrid Variant X
DK-SmGP	Dual Kriging Method: Smart GP
FCTUC	Faculdade de Ciências e Tecnologia da Universidade de Coimbra
GiD	*"The universal, adaptative and user friendly pre and postprocessing system for computer analysis in science and engineering" – CIMNE – International Center for Numerical Methods in Engineering*
GP(s)	Gauss Point(s)
IGES	Initial Graphics Exchange Specification
IVR	Incremental Volumetric Remapping
NURBS	Non Uniform Rational Basis Spline

1. INTRODUCTION

The human mind is capable of amazing creations, unfortunately, it is unable to grasp the behaviour of their own creations and their complex surroundings in only one operation. To overcome this difficulty, a typical procedure consists in subdividing a system into individual components, whose behaviour is readily understood, and then recombining them into the original system to study its behaviour.

The division can reach a sufficient state in a finite or unlimited number of components. For the second, the problem is defined as infinitesimal and therefore is associated with differential equations. These differential equations exist in many, if not all, aspects of reality. Alone, or combined, are able to create mathematical models of physical situations. But to solve these models, several conditions need to be applied, either boundary and/or initial conditions. By solving the differential equations analytically, it is possible to describe meticulously the behaviours of the entirety or any part of the system. Any analytical solution comprises two parts: a homogeneous and a particular. The homogeneous part describes the natural behaviour of a system, for example: the Young's modulus, conductivity, viscosity, area, inertia and density. This can be the same for several systems, the difference lies in the particular part of the solution, which defines the perturbations or instabilities, applied to the system, for instance: forces, moments, differences in temperature, in pressure, in electrical potential and so forth. By combining both, it is possible to obtain the solution for a specific system, in the specific conditions defined in its formulation.

However, reality is more complex. Frequently it is not possible to obtain an exact solution, either due to the complex nature of the differential equations, or to problems that arise when considering the boundary and/or initial conditions. To overcome this limitation, engineers, scientists and mathematicians have proposed methods to approximate the analytical solutions. A very common method consists in discretising the system, but instead of having exact solutions for the complete domain, only approximate ones are available in the discrete points. The methods used to discretize the system belong the numerical analysis branch in mathematics and correspond to the study of algorithms that use numerical approximation for solving mathematical analysis problems. The key idea behind numerical analysis is not seek for the exact answer, because these are often impossible to obtain. Instead, numerical analysis is concerned with obtaining approximate solutions while maintaining reasonable bounds on errors. By allowing the error bound control, numerical analysis has become the fundamental basis for solving many engineering problems, particularly after the advent of the computer.

In engineering, when solving continuum solid mechanics problems, the usual approach applied to obtain an approximate solution is the finite element method (FEM). In this, the differential strong formulation is converted into a weak integral formulation, in order to create a system of linear, or nonlinear, algebraic equations. A set of nodes can define an element, and a set of elements can define a mesh. To represent the solution inside each element, a continuous approximate function is used. After this, the global solution is generated by connecting the individual solutions, allowing for interelement continuity of the results.

The finite element method was originated, in its rawest form, in the beginning of the 19th century, when researchers approximated and modelled an elastic domain by using equivalent discrete bars. In 1943, Courant was credited as the first person to actually develop the method. In a published paper, he used piecewise polynomial interpolation in triangular subdomains to evaluate torsion problems. During the 1950s, Boeing used triangular elements to evaluate the stress state while developing aircraft wings. Over the 60s, Clough coined the term “finite element” and sparked the interest of many people. In the following years the method spread to several fields of engineering and in 1967 the first book regarding this matter was published, written by Zienkwicz and Cheung.

Whatever the mathematical formulation or program used, all finite element method applications can be divided into three distinct phases: Pre-processing, solving and post processing.

Pre-processing:

- Discretization of the domain in finite elements, i.e. subdivision of the problem in nodes and elements;
- Assumption of a shape function to represent the physical behaviour of an element;
- Creation of the physical equations for each element;

- Assembling the global system, i.e. global stiffness matrix, by combining the equations that describe each element;
- Associating the boundary, initial conditions and active loads.

Solving

- The linear or nonlinear algebraic equation system is solved, in order to obtain the nodal results, i.e. displacement, temperature.

Post-processing

- In this last phase, important information regarding the model are obtained, such as deformation, stress, heat flux, and so forth.

The increase of available computational power, including the more *recent* multithreading possibilities, also played a vital part in spreading the finite element method. The previously simplistic simulations kept being constantly replaced by newer and more complex, in an attempt to represent reality more accurately.

One specific area is the application of this method to forming processes (such as the stamping). Now more than ever, with high pressure to provide products in a fast, economic and precise way, the finite element method is an integral part of any renowned manufacturing company. Typically, the sheet metal forming process can have three stages: bending, drawing and trimming. It is mostly used in large scale manufacturing that require good superficial finish and high rates of production. The use of FEM enables engineers and designers to evaluate each step of the process and the effectiveness of the process parameters before committing to any purchase of equipment or signing contracts with impracticable conditions. Nevertheless, it is still a challenge due to the complex physical phenomena that need to be considered, for instance: large deformations, anisotropic material behaviour, contact (including friction) and other factors. Most details from the description above were extracted from Zienkiewicz et al. (2013) and Moaveni (1999). For more details, the reader is recommended to read them.

Although currently the scientific community is exploring alternatives to the conventional application of the FEM method, such as performing the calculations using GPUs (Graphics Processing Units) and replacing finite element meshes by complex surfaces to model the bodies shape, FEM continues to dominate the numerical simulation of forming processes. At the moment, several finite element programs are available, both open-source and commercial. Multiple companies exist that dedicate themselves only to the development of this type of programs. This include general purpose codes as well as specifically developed for stamping problems, such as: Abaqus, ADINA, ANSYS, AutoForm, COMSOL, FEFLOW, LS-DYNA, MSC Marc and Nastram, to name a few. Several Educational Centres, Universities, also have their in-house finite element solver.

1.1. DD3IMP

The in-house finite element solver DD3IMP has been continuously developed at the Mechanical Engineering Centre of University of Coimbra over the course of several years. Although it was originally developed to perform numerical simulations of sheet metal forming processes (Menezes & Teodosiu 2000), its range of applications has increased significantly (Oliveira et al. 2008), including the, soon to be officialised, incorporation of thermal and thermomechanical analysis.

This implicit finite element code was based on the framework of quasi-static nonlinear analysis. The mechanical model considers large elastoplastic strains and deformations, assuming that the elastic strains are negligibly small. The elastic behaviour is assumed to be isotropic, while the plastic behaviour can be described by several anisotropic yield criteria. Numerous isotropic and kinematic work hardening laws are available to describe the evolution of the yield surface with plastic work. The evolution of the deformation process is described by an updated Langrangian scheme, that is, the reference configuration is the last converged one.

The initial solution required for the Newton-Raphson Method is calculated using an explicit approach, providing the approximate first solution of the nodal displacements, the stress states and frictional contact forces. During the implicit part, an r_{min} strategy is applied to control the size of the increment and improve convergence. The principal feature of DD3IMP is the adoption of a single iterative loop to solve simultaneously the nonlinearities related with the mechanical behaviour (large deformations and elastoplastic material behaviour) and the contact with friction (Menezes & Teodosiu 2000).

In 2009, DD3IMP's suffer a huge reformulation of the structure used to allocate the variables, in order to enable the possibility of incorporating shared memory multiprocessing programming (OpenMP). Several high performance computing techniques were integrated to take advantage of multi-core processors, including OpenMP directives, with significantly improvements in performance. More information about this topic can be found in Menezes et al. (2011).

The DD3 finite element program family contains the main program DD3IMP and several (external or internal) modules dedicated, for example: to the analysis of thin coatings (HAFILM); to the trimming (DD3TRIM); to the springback calculation (DD3OSS); to the identification of parameters of constitutive laws (DD3MAT); and the educational version of DD3IMP (DD3LT). This family has been continuously develop over the last years. Special consideration will be given to the main program, DD3IMP, and to DD3TRIM which is the object of this dissertation.

1.2. DD3TRIM

DD3TRIM is a numerical tool developed to trim solid finite element meshes based on a generic trimming surface. It was developed between 2002 and 2006, by Doctorate António Baptista, in the context of analysing tailor-welded blanks (TWBs). Until 2009 it was possible to import the results file from DD3IMP to be read, in order to eliminate a portion of the original mesh (trimming), to rectify the elements modified by the elimination, to transfer the results between the meshes and generate a file capable of being read by DD3IMP and, with it, enables a new forming stage to take place.

DD3TRIM is a geometric trimming algorithm, which modifies the geometry and attempts to provide numerical results as similar as possible to the input. The main process involves altering the topology/morphology of the mesh, while considering the distortion, in shape and size, of the elements, which should be minimized. In this context, the Trimming algorithm of DD3TRIM can be divided into three phases: Pre-Processing, Correction and Post-Processing. At first all the information required for the trimming process is processed and a table containing information is filled, whether a node is saved, modified or eliminated. During the second phase, the mesh topology is modified according to one of three possible methods and by considering the abovementioned table. Finally, in the last phase, the elements' nodes can be adjusted by changing the position of the nodes in the trimmed surface, for collinear faces, degeneration into a triangular prism or into five hexahedral elements. In fact, it should be also mentioned that it was designed to operate with eight node hexahedral finite element meshes.

The trimming of a mesh is only associated with the geometrical part and therefore the output of this operation is a blank mesh with no state variable results. An important remark is that this operation is performed without considering tools nor the damage mechanisms involved in the process. To fill the results in the trimmed mesh an operation called remapping is used. In the work developed by Baptista (2006), this operation could be done with one of three remapping methods: simple interpolation/extrapolation; moving least square interpolants; incremental volumetric remapping (IVR), which was developed within the context of the work.

Due to the reformulation of DD3IMP, in 2009, it became impossible for the two programs to interact. This was the starting point for this work.

1.3. Objectives

The main objective of this work is to reprogram the finite element code DD3TRIM allowing for a permanent connection with DD3IMP, by creating a common interface. More specifically:

- Allow DD3IMP results to be processed and exported from DD3TRIM;
- Perform any necessary corrections due to compiler issues or external files formats changes;
- Provide possibility for automatic integration of changes on the DD3TRIM results;
- Execute any changes allowing for the intercalation of a trimming stage after a forming stage.

With the fulfilment of this main objective, it was decided, also, to explore the application of the Dual Kriging method for the remapping operation. This involved the development of algorithms in which particular thought must be taken regarding the quality of the results and the computational time.

1.4. Reading Guide

This dissertation is divided into five main chapters, which are briefly described in the following text.

Chapter 1: The general context of this dissertation is presented, starting with the finite element method, a brief explanation of both programs of interest, DD3IMP and DD3TRIM, the framework of this dissertation, including the objectives.

Chapter 2: In this chapter, the revisions made to DD3TRIM code, in order to fulfil the main objective of this dissertation are presented. In addition, some details concerning the additions included are described.

Chapter 3: The Dual Kriging method is presented, including the mathematical description and the selection algorithms implemented, starting from the simpler to the more complex ones.

Chapter 4: The Dual Kriging algorithms presented in the previous chapter are tested in various examples, ranging from practical to theoretical cases. The remapping results are compared with the ones obtained with the IVR method, both in terms of accuracy and computational time.

Chapter 5: The main conclusions of this dissertation are clarified, and several proposals for future work are presented.

As per definition, in the annex important information not developed by the author is presented. These include.

Annex A: The list and a small description of the modules required by DD3IMP and which were imported from DD3TRIM is presented. Its content is important for the developers of DD3IMP and not aimed at the general public.

The appendices include documents created by the author, which are not reasonable to include in the main text:

Appendix A: An example of the modified input file, Trim2.dat, necessary to run DD3TRIM is given.

Appendix B: An example of the Status Report Implemented is shown.

Appendix C: An example of the application of the Dual Kriging interpolation methods is shown.

Appendix D: The list of the mechanical properties of the materials used in the trimming tests is presented.

Appendix E: Due to its relative importance, the execution times for the tensile are presented.

Appendix F: A figure with a zoomed view of the combined Shear-tensile trimming test is presented.

2. REVISIONS AND AUGMENTATIONS

This chapter will begin by providing more information about the requirements and functionalities of DD3TRIM, namely the input files required to perform the trimming and remapping operations. The lists of modifications performed in several sub-routines of DD3TRIM are presented. After this, some improvements implemented are described in detail, most of them related with the automatization of the trimming and remapping procedure.

Because DD3TRIM was developed to work in collaboration with the finite element solver DD3IMP, the first step comprises the study of each one, as well as the common interfaces. Thus, in order to make them compatible, the understanding of the workflow and the inputs/outputs is mandatory. The four input files required for DD3TRIM are:

- Trim.dat – Configuration file, in the ASCII format, which specifies the operations that take place during the execution (trimming and remapping). The revised version can be found in APPENDIX A: Trim2.Dat.
- [name].ufo – Contains the information regarding the forming process at a specific instant: mesh (original and deformed), results (state variables) and steps. This is in fact a DD3IMP's output file.
- [name].msh – Generic finite element mesh, generated in the GiD pre-processor, which can be used for some operations, including direct remapping
- [name].igs – Initial Graphics Exchange Specification (IGES) file used in the trimming operation. Contains the geometrical description of a complex surface, defined by NURBS (Non Uniform Rational Basis Spline).

Regarding the DD3TRIM functionalities, the program comprised several functions, which can be summarized in three categories:

Trimming: performs the geometrical trimming of a finite element mesh (8-node hexahedral elements) and posterior correction near the trimmed zone. The trimming surface can be defined by a plane, a cylinder or a complex surface (NURBS). The posterior correction could be performed using three different methods, as shown in Figure 2.1, 50% in the zone to be eliminated. The first method simply eliminates those elements (see Method I in Figure 2.1), it has the downside of leaving isolated nodes and an irregular mesh contour. The second and third correction methods relocate the nodes of the trimmed elements to the surface. The nodes are moved perpendicularly to the surface in Method II and according to the edges of the elements in Method III (see Figure 2.1).

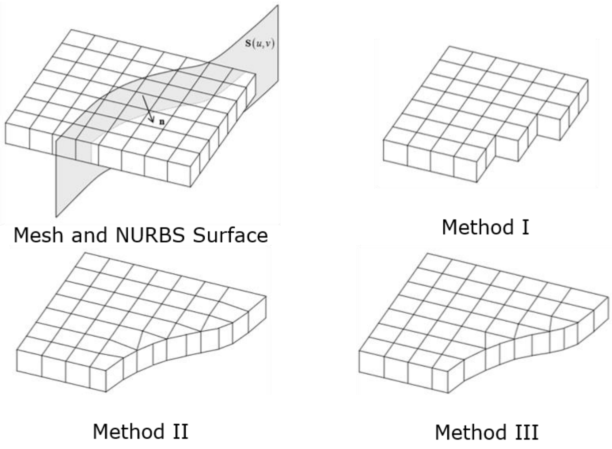


Figure 2.1: Correction Methods available in the trimming procedure (adapted from Baptista, 2006)

Splitting: allows splitting a finite element mesh (e.g. opening a ring), leaving the global geometry intact. This function creates new nodes and elements to replace the original elements that were divided by the trimming surface, i.e. in this case there is no zone to be eliminated.

Remapping: allows transferring of the results from an original mesh to a secondary mesh, which can be obtained from the trimming or splitting process, or provided externally. An important remark is that the values of the state variables (except contact forces and position) are stored in Gauss Points (GPs)[1]. This transfer is made according to the operator's selected option. Three options were originally available:

[1] These points are not coincident with the nodes and are not shared between elements.

Extrapolation-Interpolation: the results are extrapolated from the Gauss points of the elements to the nodes, of the original mesh, and then interpolated to the nodes and, subsequently, to the Gauss points of the new mesh. Nevertheless, due to the extrapolation procedure there are plenty of possibilities for minor errors to stack and contribute to a poor final result.

Moving Least Squares: a sphere is used to define a local domain of influence of the Gauss points of the new mesh. In order to obtain the value of the variable in the desired Gauss point, several weighting factors are required, these are calculated by minimizing the quadratic differences for a given variable, and weighting functions.

Incremental Volumetric Remapping: The procedure starts with the division of a region associated to each gauss point. This division is performed for both the original and the new mesh using the procedure presented in Figure 2.2. In each single Gauss volume part the state variables are considered constant. However, in the case of the new mesh this division happens twice to assure that a proper weighting is made between the old and the new volumes. This weighting depends on the percentage of volume of the original regions contained in the destination volume, as shown in Figure 2.3. According to Baptista (2006), this method allows for a minimization of the remapping error, by performing the operation in one step.

Whatever the remapping method selected, it is possible to perform any of the following operations:

- Correction of the hydrostatic pressure: which represents the stress component of the Cauchy stress tensor, which generates only elastic strain. When using the Full Integration option in DD3IMP unrealistic values for the hydrostatic pressure may occur. The correction is performed by calculating the first invariant of the Cauchy stress tensor and subtracting it from the original tensor. Since the usual integration option applied is Selective Reduced Integration, this option will rarely be used.
- Ensure consistency between the equivalent stress, flow stress and the equivalent plastic strain (dependent variables): The flow stress is evaluated with the hardening law using the equivalent plastic strain and then compared to the equivalent stress, defined according with the Cauchy stress tensor and

the yield criteria. This correction is fundamental for enabling the analysis of subsequent forming operations.

- Translate and/or rotate the final mesh in order to place it in a specific position or orientation.

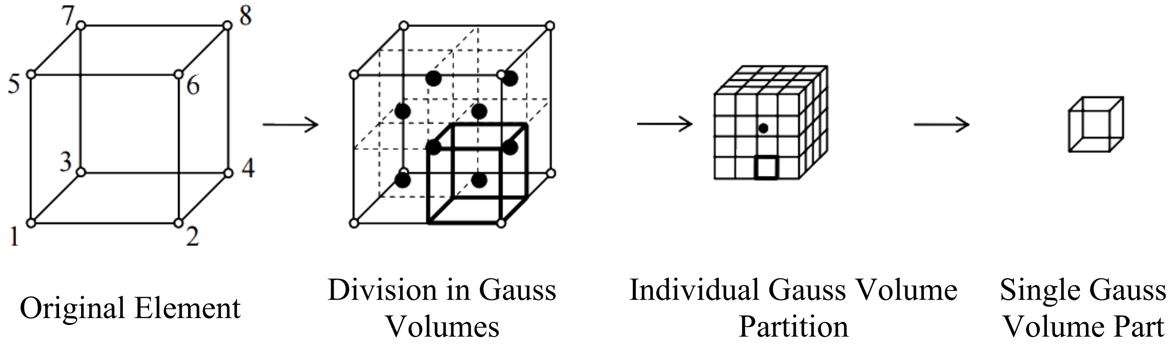


Figure 2.2 IVR's Brick Element Decomposition (adapted from Baptista, 2006, and Baptista, 2005)

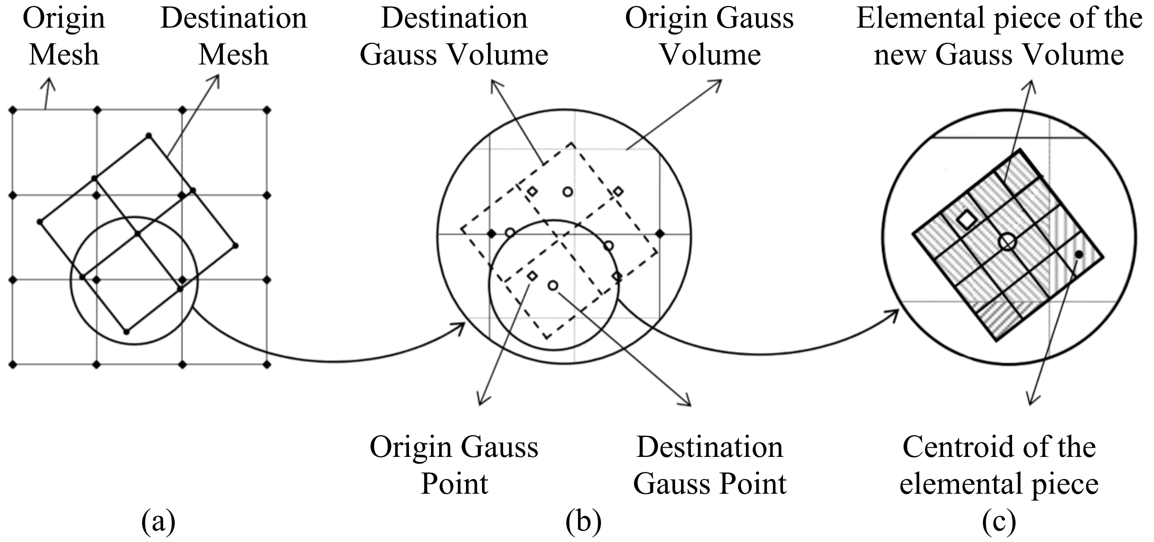


Figure 2.3 Schematic 2D IVR representation (a) Meshes of interest (b) Gauss volumes intersections (c) Division and intersection of the destination and source Gauss volumes (adapted from Baptista, 2006, and Baptista, 2005)

2.1. Modifications

In order to accomplish the objectives of this dissertation, some sub-routines were modified, and others created. Due to the length of this list, a tabular approach will be applied to categorize the modifications and additions. Table 2.1 displays the routines that were modified, corrected or condensed.

Table 2.2 contains the new subroutines created during this work, with the source files names identified in italic and the subroutines are in the bulleted lists. Some sections of the code had to be reprogrammed, as shown in Table 2.3. Also, due to the changes in DD3IMP, some routines became useless with this revision. Table 2.4 resumes the routines that were removed.

Table 2.1 Replaced and Changed Subroutines

Original Subroutine	**New Subroutine**	**Change list/Reason**
Elisa3Sup		Solved issues when reading surfaces.
Input_Ufo	*InputUfo_New* *InputUfo_Ligar*	Unification of the old Ufo reading method and the new filing system.
Inputd	*Inputd2015*	Mesh reading modifications regarding format changes.
Mapp1 *Mapp2* *Mapp3*	*Mapp0*	Due to similarities between these subroutines they were compacted and combined allowing for simpler integration of new remapping methods. The reading of the Ufo file was also, reworked.
Mapping	*Mapping0*	The interactions with the Ufo file were revised.
Nurbs_Integr		Added a secondary initial solution for the Newton-Raphson iterative process to solve convergence issues that appeared when processing some NURBS that were more complex.
Split		Integration of the new trim2.dat.
TrimCheck	*TrimCheck_Modded*	Computes the corrected equivalent stress using the material parameters from shared DD3IMP routines.

Table 2.2 New subroutines created in DD3TRIM

New Subroutines	Purpose
AA_TrimCreate • aa_TrimCreate	Automatically creates Trim2.dat if not found.
AA_TrimRead • aa_TrimRead	Reads Trim2.dat and saves the results to a module, also displays information about the selected options to the user.
AA_TrimUpdate • aa_TrimUpdate	Used while debugging to overwrite the Trim2.dat file.
Create_GP_Lists • Create_GP_Lists	Creates the necessary lists of GPs of both meshes.
Dual_Kriging • Dual_Krigging_SF • Krigging_1 • MATMULT • inv3(K) • InversionBig • inv(K) • inv2(K)	Encapsulates both Subroutines that perform Dual Kriging: updating of the state variables (stored in Elm matrix) and all the mathematical operations involved.
Mapp04 • Mapp04 • Mapp04_Elm_Refill	Comprises all Dual Kriging (trimming) remapping routines.
Remapp • ReMapp	Comprises all Dual Kriging external selection/remapping routines.
Sorting_Methods • Prepare_Coordinates • QuickSort • InsertionSort • InsertionSortDual	Contains the two main sorting algorithms, to organize arrays and matrices ascendingly; Prepares arrays with the coordinates of each node and others with the coordinates of the GPs.
Test_Mode_Fill • Test_Mode_Fill	Supplies testing functionalities to the scalar interpolation.

New Subroutines	Purpose
- ElmClear - Fill_Elm_Scalar	
Compare Scalar - Compare_Scalar - Fill_Elm_Scalar_R	Performs the mathematical comparison of the interpolated results with the analytical ones, for scalar state variables evaluation.
mod_DD3TRIM - DadosLidos - _Lists - _Stats - _Stats_TrimCheck - _Ligar - _NPE - _Mapp - _Mapp_Backup - _Mapp_Resize - _Elm_Rel - _C5_NodeShareInfo - _Symmetry	Contains all the modules necessary: the program parameters, output of execution times, output of submatrices of *Elm*, perform renaming, reshaping and cloning of matrices necessary in the mapping section, definition of parameters for Dual Kriging's result matrices and information regarding master-slave nodal relationships (more information on 3.3, DK-MSx: Master-Slave); All routines' and modules' names within the mod_DD3TRIM source file start with m_DD3TRIM which is not displayed in the table in order to facilitate the presentation.

Table 2.3 Routines Duplicated and Modified

Duplicated and Modified	Change list
Get_Canon_Modded	Originally, it acquired the coordinates of a given *point* inside an element in natural coordinates. Modified to check if a Gauss point (GP) is inside or not a given element.
DK_Inside_GP6Polyhedron	Returns information regarding the location of the GP: inside the polyhedron (six faced) defined by the original GPs or not.

Table 2.4 Routines excluded from compilation

Depreciated	Change list
Address	Provides the pointers to the K and A matrices, which used to contain all the information of the Ufo file.
Infcort	Obtained information regarding the selected trimming method and correction type.
InfCortS	Obtained the correction type inside the Split function.
InfMap	Obtained information regarding the selected remapping method.

In order to create a common interface between DD3IMP and DD3TRIM, routines were imported directly from DD3IMP, which are presented in Table 2.5. Also, some routines were simply replaced by the newer version, without requiring any changes in the DD3TRIM code, Table 2.6. DD3IMP's reformulation resulted in the creation of modules, which had to be imported into DD3TRIM and are shown in appendix ANNEX A: Modules Imported from DD3IMP. It is important to keep this information in mind to guarantee the compatibility between both codes.

Table 2.5 Routines imported from DD3IMP

Imported from DD3IMP	Objective/Motivation
Filing	Provides Ufo file reading and writing functionality.
FshapeXYZ	Returns the natural coordinates of a GP.
Functions	Replaced the old version to make use of modules.
GeInit	Initializes the shape functions for several element types.
GeMat	Reads the material data from DD3MaterX.dat.
Geoconstants	Provides several constants required by the dependent variables correction functionality.
GetDataNew	Gets and stores the constitutive model and material parameter and yield criterion of associated to a given finite element.

Imported from DD3IMP	Objective/Motivation
GeUnit	Used to define the DD3IMP mater files.
Te6ve5	Converts a symmetric second order tensor, in the three-dimensional space, into a vector in the five-dimensional space.
Valpr5	Computes the eigenvalues and eigenvectors of the Hill's yield criterion parameters matrix.

Table 2.6 Routines replaced by newer versions from DD3IMP[2]

Replaced	Change list
Te6ve5	Converts a symmetric second order tensor, in the three-dimensional space, into a vector in the five-dimensional space.
Valpr5	Computes the eigenvalues and eigenvectors of the Hill's yield criterion parameters matrix.

Some variables are defined in the nodes, and not in the Gauss Points. After the execution of DD3TRIM a new forming stage can be executed, meaning that some of this information needs to be updated. Therefore, it was decided that DD3TRIM will initialize some of this variables to zero. Table 2.7 resumes the information about these variables.

Table 2.7 Variables initialize to zero in the Ufo file

Variable	Change list
NOCB	Table that governs the contact status of the nodes of the deformable body.
NCR	Table that contains information regarding the boundary conditions for the node, displacement or force.
MState	Table that defines if a certain element is in elastic or plastic state.

[2] Note: Some routines appear in both tables since they are appropriate for both categories.

2.2. Improvements

In order to improve the functionality of the program DD3TRIM, namely the ability to automatically perform a trimming operation, some algorithms were developed and implemented.

2.2.1. Automatic mode

The original version of DD3TRIM required the manual insertion of the input files' names and additional information regarding whether or not to correct: (i) the hydrostatic pressure and (ii) the dependent variables. Although this method is useful it has some disadvantages, namely it is prone to user errors. Moreover, the so-called, batch of simulations in optimization procedures becomes impossible to execute, since user intervention is always required. The creation of a batch (.bat) file allows, among other possibilities, for the sequential execution of simulations. Therefore, this enables the user to start a multi-stage forming with several intermediate trimming operations from a single interaction.

A new version of the trim.dat file was created, which allows the files' names to be read without more intervention from the user. Similarly to DD3IMP, if this file does not exist in the directory, it will be automatically created and the user will be asked to fill it before executing the program again. This also has the advantage of avoiding error associated with the use of older version of the input file. An example of this file is presented in APPENDIX A: Trim2.Dat.

A mode flag in the top left corner defines in which mode the program is running on: 00 – **standard user**, when important information is displayed to the user, for instance which options were chosen, demanding the user intervention to proceed; 01 – **DD3IMP mode**, the program runs and exits without requiring any interaction.

2.2.2. Status report

As with any other project, the continuous improvement requires the identification of bottlenecks in performance. One of the most used ways to find them is to obtain a report with execution times, for specific parts of the code. The options required to enable this analysis were implemented in DD3TRIM, such that the time spent during each task and the configuration of the execution is saved to a file. An example can be seen in

APPENDIX B: Status Report, highlighting the provided information that can also help identifying possible issues that occur during the execution.

2.2.3. Other features

Several options were also added as shown in the example file in APPENDIX A: Trim2.Dat. The **Autoname** appends the extension .Trim to the name of the output Ufo file, which is the one that will be required by DD3IMP to perform any posterior operation. The **AutoAppend Options** allows for an automatic naming of the output Ufo file based on the selected remapping options and the original file name. **Test Mode**, **Inversion Test Mode** and **Compare Scalar** refer to options available when testing remapping algorithms with scalar variables (see section 4.4.1, for more information). In addition, new remapping options are available, for both in internal and external meshes, using Dual Kriging interpolation method. The next chapters of this dissertation present the algorithms implemented and the validation tests performed.

3. DUAL KRIGING

In this chapter, a description of the Dual Kriging interpolation method is provided, and the algorithms implemented to enable remapping operations are explained. The importance associated with the implementation of this method, besides improving the possibilities available in DD3TRIM, is the possible transposition of this method to DD3IMP: (i) to enable a better output to GiD post-processor and (ii) to be used in remeshing algorithms.

3.1. Mathematical Description

Kriging is a geostatistical method based on the work of Krige (1975), a mining engineer, and a French mathematician Matheron (1975). This method, and the ones based on it, have two main characteristics: they are the best linear unbiased estimators of a variable and, also, they provide an exact interpolator. The exact interpolator property ensures that, when the original coordinates are provided, the interpolation method will return the original values. This property arises due to the mathematical formulation of the method.

In this work, the Dual Kriging version was applied since it is the parametric form of the classical Kriging method. The interpolation function assumes the following general form:

$$F(X_0,Y_0,Z_0)=\mathrm{C}_0+\mathrm{C}_1\mathrm{X}_0+\mathrm{C}_2\mathrm{Y}_0+\mathrm{C}_3\mathrm{Z}_0+\sum\lambda_\mathrm{j}K(\mathbf{h}_{0\mathrm{j}}). \tag{3.1}$$

In the context of the work, X_0, Y_0 and Z_0 are the coordinates of the new Gauss point (GP). C_0, $\mathrm{C}_1, \mathrm{C}_2$ and C_3 are scalar quantities. In the summation, $\mathbf{h}_{0\mathrm{j}}$ is vector with the Euclidean norm between the calculation point and point j. The index 0 denotes the calculation point, while j varies between 1 and the total number of points considered, n. The variable K, in this context, is the generalized covariance function and can assume several forms, it is affected by a weighting factor lambda, λ. As in the finite element method (FEM), the problem involves a system of linear equations that can be defined as:

$$\mathbf{Ku}=\mathbf{f},\Rightarrow\mathbf{u}=\mathbf{K}^{-1}\mathbf{f}, \tag{3.2}$$

where **K** is called the Kriging matrix, **u** is the coefficients vector, and **f** contains the values of the variables in the original points. As shown in (3.2), one option to solve the system of linear equations is through the calculus of the inverse of **K**, for each new GP. Three algorithms from LAPACK were implemented to perform the matrix inversion. This option was adopted trying to combine the best performance while guaranteeing that a solution is always obtained. Thus, the three methods are used sequentially, in case the inversion fails. The sequence starts with the fastest method, which considers a full matrix, and ends with the slowest method, which considers a symmetrical matrix. It should be mentioned that it was always possible to find a solution for the problem. Even though the inversion is a fast process, less than 0.01 seconds up to a dimension of about 200×200 (i7-4860HQ @2.4-3.6GHz), the same inverse matrix can be used for several variables and executions as long as the original points remain unchanged.

The dense matrix **K** can be divided into four submatrices, which present different dimensions.

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}. \tag{3.3}$$

$\mathbf{K}_{11}$ is the bigger submatrix, with dimension $n \times n$, containing the covariance values between each original point. These values can be calculated using different mathematical functions. $\mathbf{K}_{21}=\mathbf{K}_{12}{}^{\mathrm{T}}$, contains the coordinates of the points used in the interpolation and values equal to one, with a dimension equal to the number of the space dimension ((three in R^3) + 1)×n. $\mathbf{K}_{22}$, has a dimension 4×4 and contains only zero values.

The objective is to solve (3.2) in order to find the values of the coefficients vector **u**. As previously mentioned, these will be computed using a predefined covariance function resulting in a series of coefficients that will be used with the point of interest's coordinates to obtain the required results. The first n positions of this vector will be the λ coefficients and the last four will be the C_{x} (x=0,1,2,3) constants present in equation (3.1).

The implemented algorithm allows for an easy change of the covariance functions. Nevertheless, the use of non-continuous functions can lead to serious errors in the interpolation. Three functions were suggested by McLean et al. (2006), as shown in Table 3.1. Only the first of them is available and applied during this work. The other two were rejected, because both overvalued the variable in the test case used during this work. Also, in the work of McLean et al. (2006) shows that the first function renders the more accurate results. In this context, the reader can see an application example of this function in APPENDIX C: Dual Kriging Example.

Table 3.1 Suggested Covariance Functions[3]

Function 1 (selected)	Function 2 (overvalued)	Function 3 (overvalued)
$K(h) = h$	$K(h) = h^3$	$K(h) = h^2 \times \ln(h)$

The implementation of Dual Kriging as a remapping method, applied to one GP, can be divided in two stages: (i) selection of a set of neighbour old GPs; and (ii) evaluation of the variable's value in the new GP. During the selection, the Cartesian coordinates of the Gauss points (set of old and new positions) are evaluated. Subsequently, the Kriging matrix **K** is assembled, inverted and used to calculate the desired variable on the new GP.

The priority in the implementation of the Dual Kriging method was given to internal mesh remapping. Note that in this case the information regarding the modified nodes (position) and, consequently, coordinates of new GP, is provided by the trimming algorithm (see Chapter 2). Thus, in the following section the selection algorithms developed specifically for internal meshes are present, followed by their extensions to external meshes remapping. The main idea behind this organization is to allow the reader to follow up the sequence of improvements.

3.2. DK-1: Only inside the Element

The simplest adoption of the Dual Kriging method involves only the GPs of a single old finite element to obtain the variable's value in the new GP. The trimming procedure modifies the nodes position of some finite elements and, consequently the GP, which need to be remapped using the old variables' values. Thus, the Dual Kriging is performed for each GP of the new finite element, using only the information contained in the eight GPs of the untrimmed element. However, the use of this method can lead to significant errors in the remapping procedure. Consider the two dimensional representation of the hexahedral element in Figure 3.1, which presents the original GPs in blue and the new GPs in red. Since the two nodes of the right edge were translated (during trimming), the

[3] Note: h is the Euclidian distance, as referred above.

Cartesian coordinates of the new GPs have changed accordingly. Since this Dual Kriging implementation considers only the coordinates of the old GPs to evaluate the value in the new GP, everything outside the space defined by them will have an extrapolation error associated. Due to the trimming algorithm, it is easy to conclude that the extrapolation can occur in 50% of the trimmed elements. In order to verify the impact of this approach in the remapping results, the reader is referred to section 4.2.1.

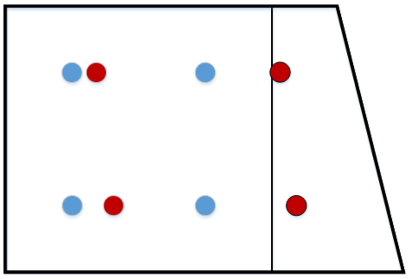


Figure 3.1: Old GPs in blue and new GPs in red. Contoured GPs will have an extrapolation resulting from the adoption of DK1.

3.3. DK-MSx: Master-Slave

In order to overcome the extrapolation issues of the approach presented in Section 3.2, a new approach was developed, trying to keep the simplicity and efficiency in the selection stage. The nomenclature adopted in this method is first presented, where the main concepts are defined, in Table 3.2.

Table 3.2 Terminology adopted in DK-MSx: Master Element Variants

Name	**Definition**
Master Node	Closest node to the new GP
Master Element	Element partially defined by the Master Node, containing the new GP
Master GPs	GPs belonging to the Master Element
Slave Elements	Elements that share the Master Node, but do not contain the new GP
Slave GPs	GPs that belong to any Slave Element, which will be considered in the Dual Kriging process.

This algorithm is based on the relative positions of the new GP inside an element (Master Element) by considering the nearest node (Master Node) and the elements that share that specific node (Slave Elements). The old GPs are then selected according to the chosen variant. In order to try to explain the idea, Figure 3.2 presents an example for the new GP, represented with the pentagon. The Master Element's GPs are represented in blue with a red circle in the background, each four-sided symbol will have a colour depending on which Slave Element it belongs to.

All GPs from the Master Element will be used to perform Dual Kriging interpolation. Besides, the seven remaining GPs, that are associated with the Master Node (proximity) and contained in the Slave Elements, will also be considered. Depending on the relative positions of each element that shares the Master Node, in the first variant, up to three additional GPs, per element, will be considered (see Figure 3.2). The number of additional GPs is a function of the relative position of the Master and the Slave Elements under analysis.

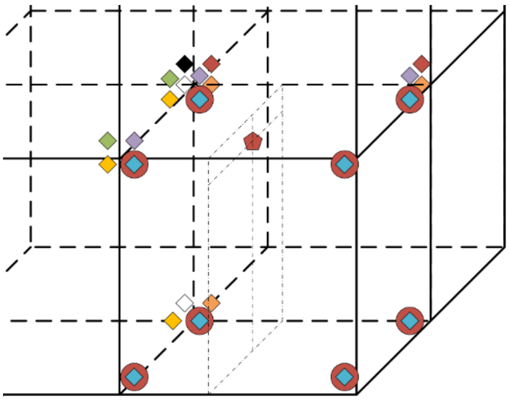


Figure 3.2: Example of the DK-MSx: Master Element's GPs selection method. The coordinates of the GPs are not exactly to scale in order to facilitate perception.

Three variants of this method were developed and implemented. The initial formulation was previously described and can also be seen in Figure 3.3, where the GPs considered are represented in yellow. However, this figure tries to highlight the GPs from the slave elements. The second variant adds GPs located in the green position leading to the possibility of considering three additional GPs, besides the ones from the first variant. The third variant was implemented considering all the GPs from the slave elements that surround the Master Node, up to a total of 64 GPs. The selection of the variant must be carefully performed, regarding the impact of the neighbourhood. In fact, the third variant will add more information that may not provide more accurate results, as it will be discussed in the following chapter.

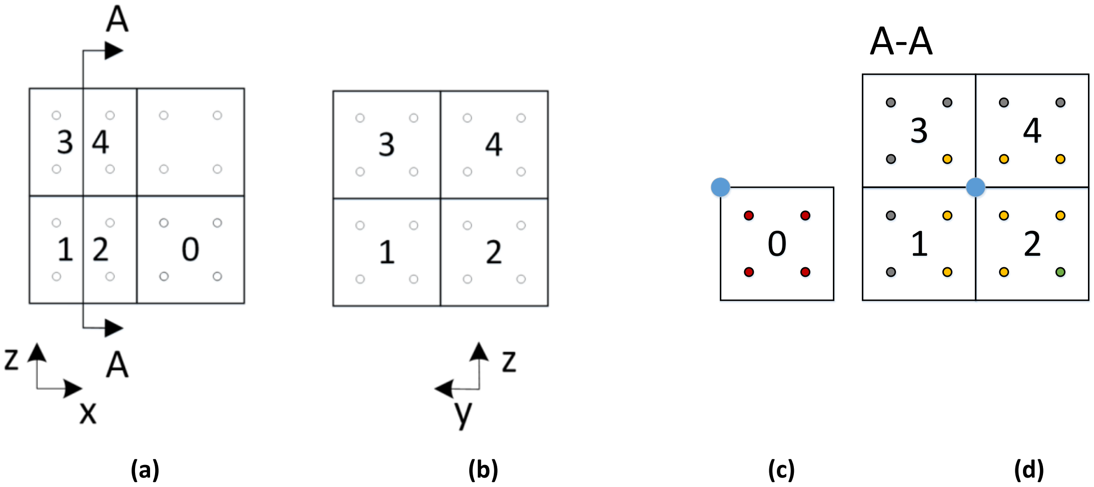


Figure 3.3 Master-Slave Variants: (a) finite element mesh in a XY plane, indicating the cut plane; (b) finite element mesh in the YZ plane; (c) selected Master node and Master element; and (d) location of the old GPs that can be used in the interpolation

The numerical simulation of sheet metal forming process resorts to symmetry conditions, in order to simplify the model and hasten the simulation. This means that the variable values are extended virtually by applying a mirror function. This affects the distribution of the variables, since the derivatives normal to the symmetry section, must be zero. The possibility of considering symmetry conditions in the Dual Kriging method was also implemented, which is expected to improve the quality of the results near these areas. Unfortunately, only the fixed coordinated direction is stored in the Ufo file, NOBC matrix. Since there is no information regarding the original mathematical condition, so it is necessary, for the user, to activate this functionality manually in the trim2.dat file.

In order to improve the computational performance, several indexed lists are created to execute this algorithm, with the massive benefit of forcing an interpolation by adding information about the neighbourhood. Two inverse connectivity table are created, one that contains the elements and the other that contains the associated GP. A third table was also created, which is critical to this algorithm efficiency, because it allows for the fast selection of the slave nodes based on the Master Node position in both the Master Element and in the Slave Elements. This table can also be modified to select specific GPs in elements. The global algorithm of Dual Kriging with the Master-Slave Mode is presented in Table 3.3.

Table 3.3 Outline of the algorithm adopted in the DK-MSx: Master Element Variants

```
Start
Fill equivalence tables for Nodes, Elements and Gauss Points

Do:     All changed nodes
        Do:     All changed GPs
                Select Master Node
                Select Master Element
                Select Master GPs, store information and position
                If Slave Elements are available:
                        Select Slave Elements
                        Select Slave Nodes
                        Select Slave GPs, store information and position
                End if
                If Analyse Symmetry is selected:
                        Select Symmetric Slave Elements
                        Select Symmetric Slave Nodes
                        Select Symmetric Slave GPs, store information and position
                End If
                Perform Dual Kriging
        End Do
End Do
```

The implementation of this algorithm considers the possibility of performing the correction of the GPs, in the modified element, but not belonging to the trimmed surface. It could be assumed that, due to the small change in the positions of the GPs, associated to an unmoved node, the results would not change. This argument is partially valid for a small change in the position. In fact, most of the times, the changes in the state variable results are minimal. However, the increase in computational time can be considered negligible. As previously mentioned, the inversion of the **K** matrix is performed very fast. The rest of the algorithm is also performed in an extremely small amount of time, as all the information required is indexed and prepared for fast access.

It should mention that, because the algorithm was developed and implemented based on the node connectivity, the integration of other variants will be effortless and expeditious.

3.3.1. DK-MSxH: Master Element Mode Hybrid

When the DK-MSx method is used for a new GP located near the centre of an element, several GPs outside the element will be considered. This can leads to an overestimation of the values in the new GP. In order to overcome this issue, DK-1 and DK-MSx were combined in what is called the hybrid mode. This is based on the definition of a polyhedron (six faces) whose corners are the GPs from the Master Element, known as GP6Polyhedron. After defining the Master Element, this new algorithm, will check if the new GP is inside or outside the GP6Polyhedron, If inside, it will perform Dual Kriging only with the values of the element. If outside, it will perform Dual Kriging using the variant selected for the method described in the previous section. The global algorithm of the DK-MSxH: Master Element Mode Hybrid is presented in Table 3.4.

Table 3.4 Outline of the algorithm adopted in the DK-MSxH: Master Element Mode Hybrid

```
Start
Fill equivalence tables for Nodes, Elements and Gauss Points

Do:     All changed nodes
        Do:     All changed GPs
                Select Master Node
                Select Master Element
                Select Master GPs, store information and position
                If New GP is outside GP6Polyhedron:
                        If Slave Elements are available:
                                Select Slave Elements
                                Select Slave Nodes
                                Select Slave GPs, store information and position
                        End if
                        If Analyse Symmetry is selected:
                                Select Symmetric Slave Elements
                                Select Symmetric Slave Nodes
                                Select Symmetric Slave GPs, store information and position
                        End If
                End If
                Perform Dual Kriging
        End Do
End Do
```

3.4. External Mesh Remapping

In order to use external finite element meshes during the multi-stage forming processes, the Dual Kriging method was incorporated as a new external mesh remapping method. Three selection algorithms were developed, which are described in the following sections.

3.4.1. Brute Force

This is the simplest GP selection algorithm one can think of. In the early tests, using small meshes, it was a fact that Dual Kriging interpolation was performed quite fast. Consequently, Brute force, which simply considers all the GPs in the mesh and performs Dual Kriging considering all of them, seemed a promising approach. However, two logistical problems may arise from this algorithm: (i) the impossibility to run due to memory allocation issues, mainly when using the 32 bits version, or (ii) the time it would take to completely solve the problem.

The first problem can be overcome by using the 64 bits version. Nevertheless, the second one is almost impossible to solve with the available hardware. Therefore, it is considered that for real finite element problems it cannot be applied due to the computational time issue. Although not tested, the usage of GPUs could potentially overcome this problem. However, the potential impact of considering GPs far from the GP of interest should be kept in mind.

3.4.2. DK-SmGP: Smart GP

In order to reduce the overhead while selecting the input information for Dual Kriging interpolation, and trying to avoid the calculation of all the Euclidian distances between the new GPs and the old GPs, a Cubic Distance Method is applied, which consists in the sequence described in the following text.

The first step consists in organizing the information concerning all GPs coordinates, of both meshes, in auxiliary arrays. Afterwards, for each new GP, a cube (3D) centred on the coordinates of the new GP is created. The length of the cube side increases progressively until a given number of GPs is found within the cube. This iterative method begins with a null dimension for the side and increases one unit at a time. It would have been possible to calculate the distance of all the points, or even to evaluate the average element size, but that was considered ineffective in terms of computation efficiency. The third step involve in storing the GPs located inside the cube into an auxiliary array. Subsequently, their distances, to the destination coordinates, are evaluated. Figure 3.4 presents a more tangible two-dimensional representation of this method, where the blue pentagon represents the new GP. The candidate GPs are illustrated with circles. For a certain GP, represented in Figure 3.4, to be considered for remapping, it must be completely inside the square. It is possible to see that the mesh density will have an important role when collecting the minimum number of GPs, in this case arbitrated as 38. In the example presented in Figure 3.4, on the regular mesh, on the left side, 40 GPs were found, while on the irregular mesh, on the right, 50 GPs are collected. These GPs are then used to perform the Dual Kriging interpolation. This method has the particularity of allowing for any given number of GPs, selected by the user, to be used. Nevertheless, the selection of this number must be done wisely. It should be enough to ensure that an interpolation is performed but not so high that it will damage the method results or expend an excessive amount of computational time.

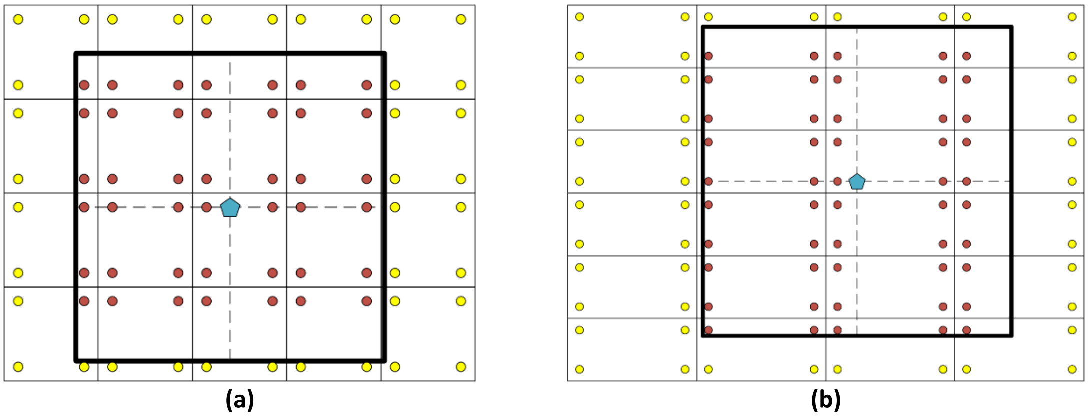

(a) (b)

Figure 3.4: 2D representation of the Cubic Distance Method. Influence of the mesh topology in the selection of the minimum amount of old GPs, considering a cube with the same size: (a) regular and (b) irregular mesh.

The use of the cube in the selection method could have a drawback, when compared to a sphere: the GPs on the corners will be farther than the ones near the middle sides of the square. However, the direct use of a sphere would require the calculus of the relative distance, to each new GP, during the selection of old GPs. In addition, it would be necessary to sort these distances. Both of these operations are extremely CPU inefficient. To partially compensate the corners' effect of the cube, the user can define how many GPs will be considered for the calculation. Thus, it is possible to define a subset by minimizing the distance to the new GP. In fact, in sheet metal forming the dimension of the mesh along the thickness is smaller than the in-plane dimensions. Therefore, this can lead to considering an excessive amount GPs along the thickness and, depending on the relation between the thickness and planar gradients, lead to errors.

After the development of this method for external mesh remapping, it was decided to include this option in the standard remapping, i.e. after trimming or splitting operation.

3.4.3. Smart GP Master-Slave

This method was developed based on the previous one. Therefore, they share several algorithmic structures. However, similarly to the Master-Slave algorithm (see section 3.3), the nearest node to any GP is assessed. All elements that share this node are found and the one that contains the new GP is defined as the Master Element. From this point forward, the algorithm remains the same as the one described in section 3.3.

4. REMAPPING EXAMPLES

This section presents the application of the Dual Kriging interpolation method to different examples, including remapping of internal and external meshes. In the first case, the accuracy will be evaluated, comparing the results before and after the trimming operation. In this context, some comments regarding the trimming correction methods, available in DD3TRIM, and their influence on the remapping results will also be made. For the second case, a more generic comparison between external mesh remapping methods is carried out using a mathematical function. The influence of the remapping algorithm adopted in the correction of the dependent variables will also be discussed in the last subsection. The results are also compared with the ones obtained with the IVR method, which was the one, from the originally implemented remapping methods, that presented better global results (Baptista 2006).

4.1. Initial Remarks

The visualization of the results was performed with the aid of GiD postprocessor. Originally, it was only possible to export results in the nodes to GiD. Although nowadays the option of exporting the results directly to GPs is available, DD3IMP output files to GiD keep only the information of the nodes, in order to minimize the size of the ASCII files. The function used in DD3IMP to transfer to information from GPs to the nodes corresponds to a simple average, taking into account just the number of neighbour GPs. In the results analysis performed in the following sections, it is assumed that the original and remapped results are affected by the same type of error, induced by this transfer function. Anyhow, regarding DD3IMP's results, in most cases under analysis, it is possible to postulate that the deformation will tend to spread throughout all connected nodes and, therefore it is reasonable to assume that most GP will have an almost equal distance from the associated nodes thus, an unweighted average could be used to interpolate the value on the nodes without considerable error. However, this assumption is not valid when an element was trimmed using DD3TRIM, since the criteria for element elimination is that the volume to eliminate is above 50%. Thus, in the worst scenario, and assuming elements with an equal three-dimensional geometry, the GP could be farther from the associated node than the original GP (up to a 50% increase) or closer (up to half of the original distance). In the analysis of the results obtained after trimming operations, the value displayed in GID for a node located in the trimming section corresponds to the average of the GPs of the elements that share the same node in the new mesh. However, in the old mesh there might not be any element there. Thus, the analysis of these results must be carefully done in order to avoid misinterpretations, since for both meshes they are interpolated results. In this context, the comparison with IVR's results seems to be the ideal benchmark, for results comparison.

In the following sections different examples are analysed, trying to highlight the advantages and potential pitfalls of the remapping methods analysed. All geometrical dimensions are given in mm and stresses in MPa.

4.2. Trimming Examples

4.2.1. Tensile Test

The first example corresponds to the trimming of a tensile test specimen using an inclined plane surface. The specimen presents symmetry conditions on both planes XY and YZ. The variable being evaluated is the stress tensor component $\sigma_{XX} = \sigma_{11}$, which was chosen due to its small gradient along the X direction. It can also be interpreted as a scalar variable even though it is a component of the stress tensor. The selected trimming surface originates two finite elements having their size reduced and two having it increased. The main information about this test can be found in Table 4.1. The material properties used to the tensile specimen can be found in APPENDIX D: Material Data.

Table 4.1 Tensile Test Information

Mesh Data			**Phase Data**		**Surface Data**			
	Length:	**Number of Elements:**	**Displacement:**		**Defined by 3 points:**			
X	20	20	ΔX	6	X	6,5	6,5	8
Y	5	5	ΔY	0	Y	3,92	3,92	-0,28
Z	1	2	ΔZ	0	Z	3	-3	0

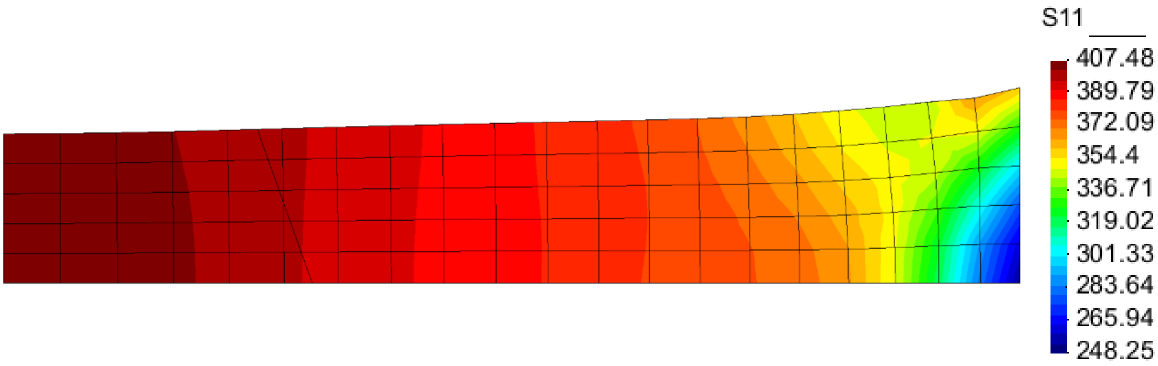


Figure 4.1 Original mesh and results (stress tensor component). The line represents the cut plane.

4.2.1.1. Correction Methods

Figure 4.2 presents the results obtained with the two trimming correction methods (Method II and Method III in Figure 2.1) applied to the trimmed section. Both methods provided an accurate representation of the trimming, according to what was expected from them. The element in the middle of the trimming section was modified and corresponds to the transition between expanded and contracted elements. Method II (Projection) was selected for the following comparisons.

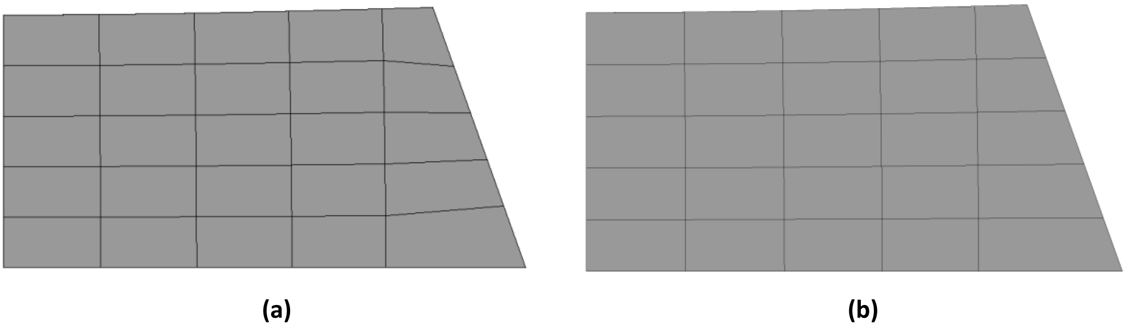


Figure 4.2 Correction method comparisons (a) Projection (2nd) (b) Intersection (3rd)

4.2.1.2. Remapping

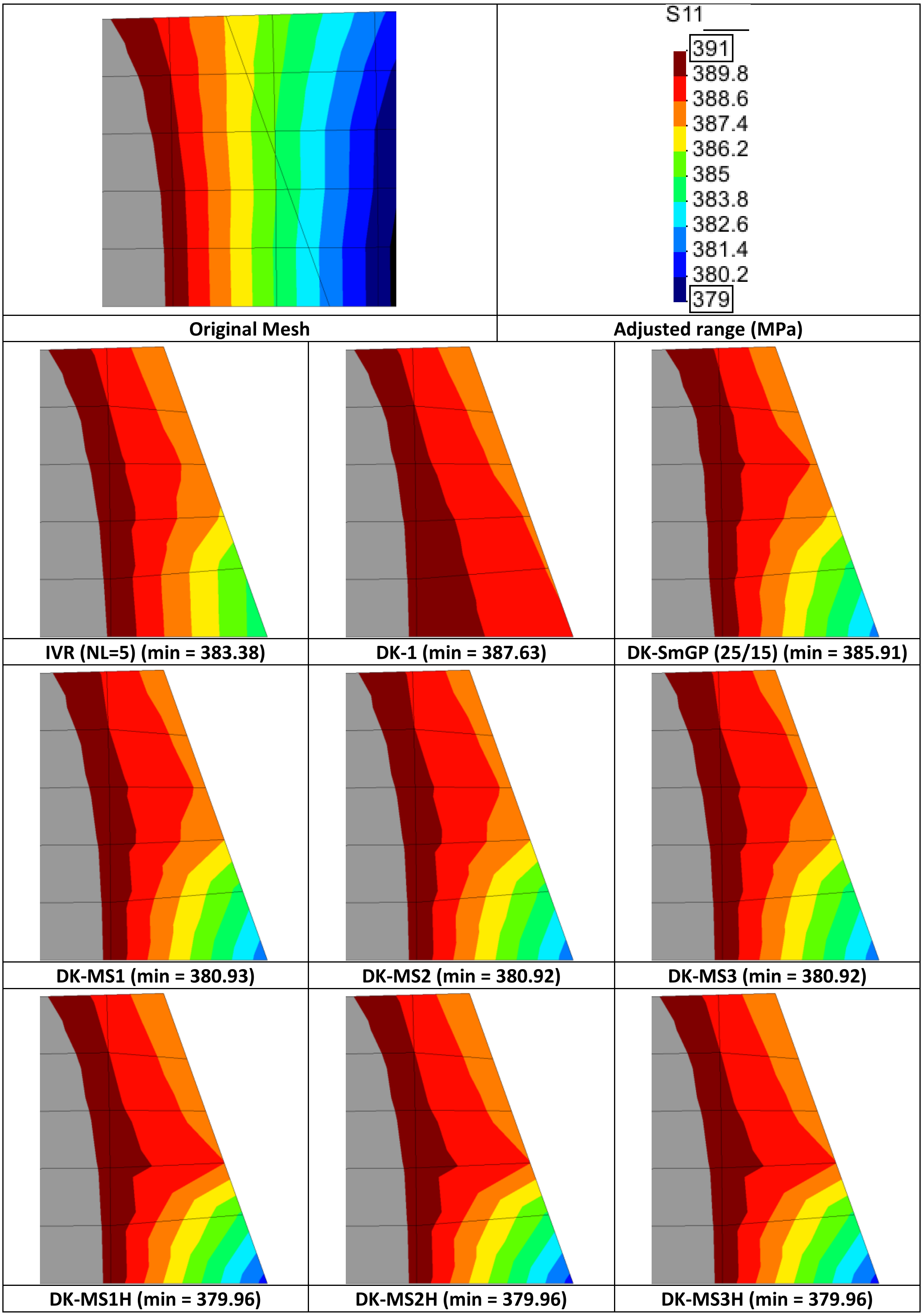
In this example, the analysis of the results is performed considering the distributions obtained for σ_{11} in the trimmed zone. Figure 4.3 presents the results for the different remapping strategies, using always the same scale. The results were obtained without taking into account the symmetry conditions, as the differences were negligible.

What is being evaluated here is the general behaviour of the algorithm. The values indicated in Figure 4.3 between the parentheses, correspond to the minimum value of σ_{11}, in MPa. For IVR, the additional information (NL=5) is the number of divisions applied by the Method. For DK-SmGP, the first number (25) was the minimum user-defined number of GPs, and the second (15) was the user-defined number of GPs to be used in the calculation.

As seen in Figure 4.3, independently of the remapping method applied there are always problems near the transition between expanded and contracted elements. In fact, all methods smoothed the gradient in this position even though the variable changed smoothly across the surface in the original mesh, and along the trimming surface. All methods also had issues in the top section, even though the GP near the top corner had the correct value (as only one GP contributes to the node's value).

DK-1 exhibits the worst values and worst distribution, with a clear overestimation of the values for the GPs located near the bottom. DK-MSx and DK-MSxH show almost no increase in the quality of results between the variants. DK-SmGP was able to provide results similar to IVR, near the top of the trimming section, and similar to DK-MSx on the three elements in the bottom section.

The computational time comparison for this test can be seen in APPENDIX E Tensile Test Computational Times. A more relevant comparison will be performed in the following section and, therefore, it was decided to comment the results only in the appendix.



Original Mesh

Adjusted range (MPa)

IVR (NL=5) (min = 383.38)

DK-1 (min = 387.63)

DK-SmGP (25/15) (min = 385.91)

DK-MS1 (min = 380.93)

DK-MS2 (min = 380.92)

DK-MS3 (min = 380.92)

DK-MS1H (min = 379.96)

DK-MS2H (min = 379.96)

DK-MS3H (min = 379.96)

Figure 4.3 Comparison of σ_{11} distributions

4.2.2. Bending Test

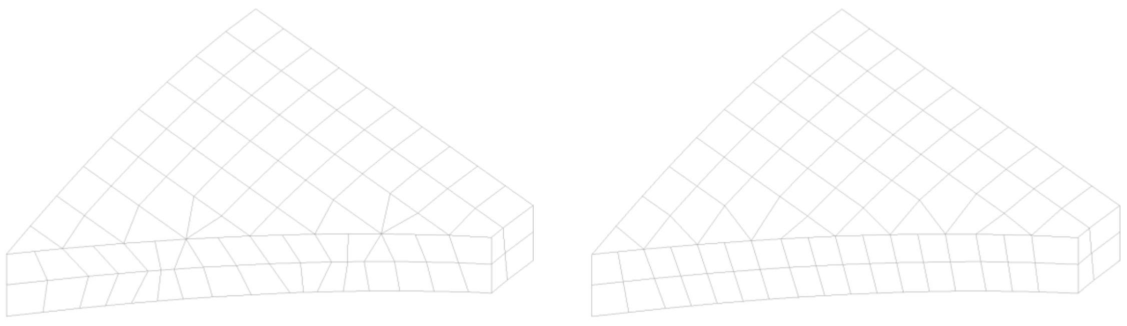
In order to consider an example with significant gradients in the state variables, along the thickness direction, the bending test was analysed. The test is performed under plane strain conditions and the specimen is fixed in $y = 0,\ y = 5$ and $x = 0$. The information about this test can be found in the Table 4.2, while the material properties can be found in APPENDIX D: Material Data.

Table 4.2 Bending Test Information

Mesh Data			Phase Data		Surface Data			
	Length:	**Number of Elements:**	**Displacement:**		**Defined by 3 points:**			
X	20	40	ΔX	0	X	6	6	0
Y	5	10	ΔY	0	Y	-2	-2	6
Z	1	2	ΔZ	10	Z	3	-9	6

4.2.2.1. Correction Methods

The comparison of the trimming correction methods is presented in Figure 4.4. The correction method II, based on the nodes' projection, leads to a less acceptable result in the through-thickness direction. This can be explained by the angle between the vertical trimming surface and the vertical edges of the mesh. On the other hand, correction method III provides an approximately uniform mesh through the thickness because the nodes are simply moved along the original edge. Therefore, in the following section the remapping results analysis will be focused on the correction method III.



(a) (b)

Figure 4.4 Correction Comparison (a) Projection (2nd) (b) Intersection (3rd)

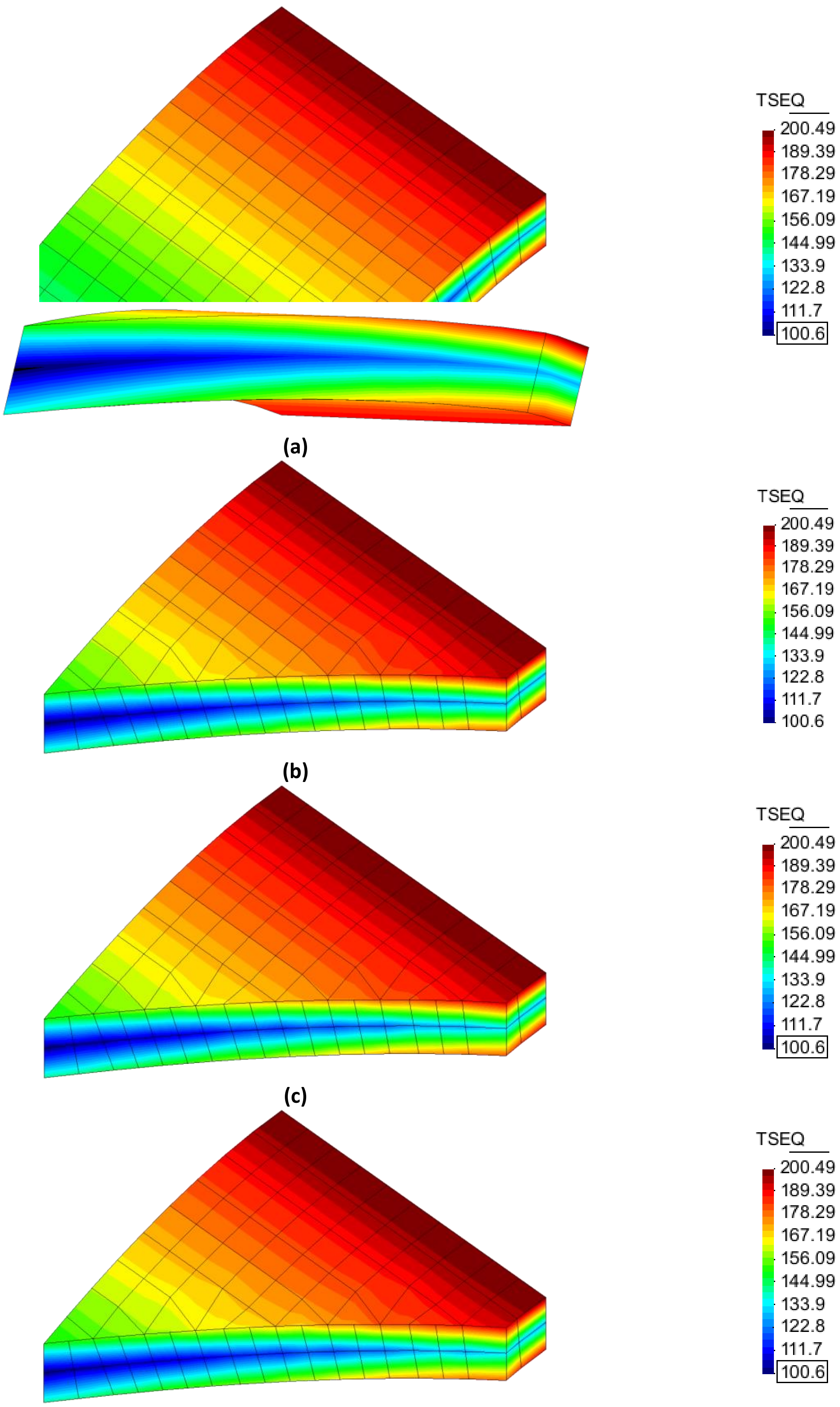
4.2.2.2. Remapping

The equivalent stress distribution in the sheet after bending is presented in Figure 4.5, near the constrained section with the fixed boundary conditions, highlighting the cross section where the trimming operation is performed. In Figure 4.5 (a) the results are shown for the original mesh. A detailed view is included for the distribution along the thickness in the original mesh, visually trimmed using GiD post-processor. Thus, it is impossible to accurately display the mesh.

Considering the top surface of the sheet, IVR and SmGP algorithms present some issues near the transition to the trimming section, namely near the maximum value. Master-Slave provided the most accurate representation in the top surface, although it also had some issues in the light orange coloured section. Comparing the distribution along the thickness, IVR and Master Slave have very similar distributions. Nevertheless, near the right corner the situation changes for the four nodes in the mid plane. The variable's value in the corner node increases from 127.94 MPa to 132.67 MPa, comparing IVR and Master Slave respectively. The difference between variant 2 and 3 of Master Slave was 2.13 MPa, while variant 3 had 130.4 MPa. The value in the unmoved node had increased from 129.34 MPa to 130.6 MPa for both Dual Kriging selection algorithms. For this node, the IVR kept the same value of 129.34 MPa. The DK Master Slave and the DK SmGP had lower bounds for the equivalent stress of 101.42 MPa and 101.5 MPa, respectively, while IVR had the lowest minimum value of 100.6 MPa.

In order to compare the difference between IVR and DK-Master Slave Variant 3, the relative difference between the values in each GP was evaluated using the following expression:

$$\frac{DK - IVR}{IVR} \times 100\%. \tag{4.1}$$



(a)

(b)

(c)

(d)

Figure 4.5 Bending Remapping Comparison (a) Original including through thickness detail (b) IVR (c) DK-MS2 (d) DK-SmGP

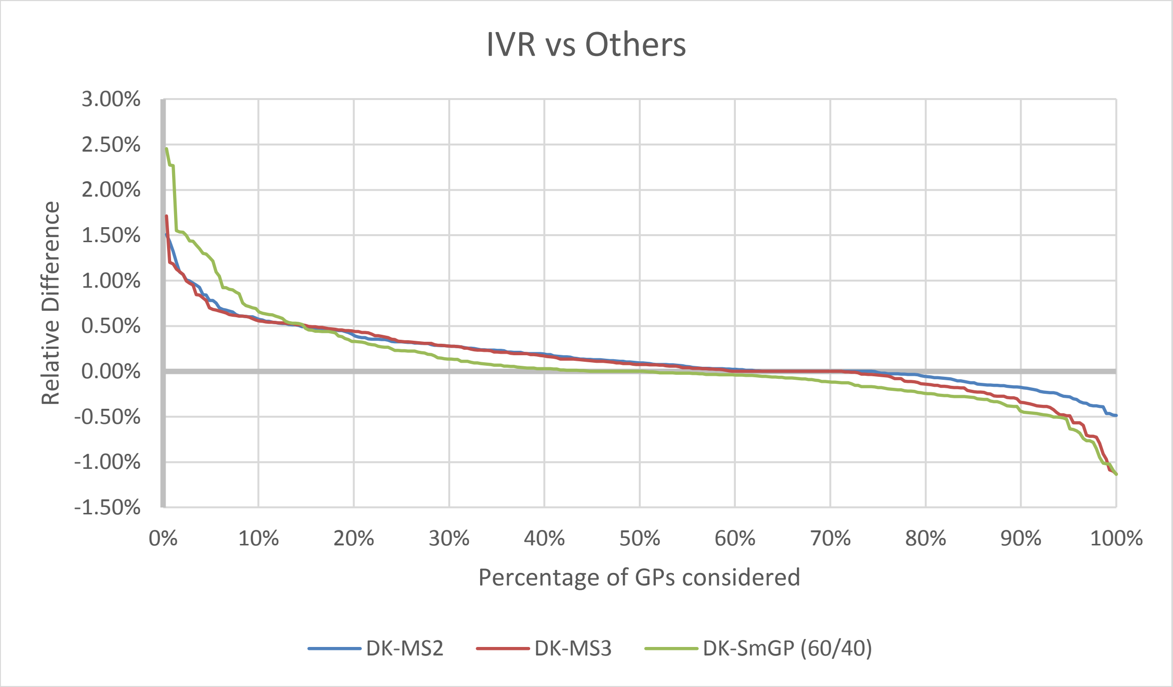


Figure 4.6 Comparison IVR vs Dual Kriging

Figure 4.6 presents the relative difference between IVR and several Dual Kriging selection methods, evaluated for each GP (288 total). For DK-MS2, only 14.2% of the GPs have more than 0.5% of difference to IVR; 24.8% are between 0.25% and 0.50%; 61% of the GPs, have values that are within 0.25% of the IVR. DK-MS3 presents 19.8% of the GPs with a value bigger than 0.5%, 25.7% within 0.25 and 0.50% of the IVR's values; 54.5% are below the 0.25% threshold. For SmGP, 21.4% of the GPs have more than 0.5% of difference, 12.2% between 0.25% and 0.50% of IVR's value, while the remaining 56.4% are below 0.25%. Therefore, it is possible to confirm that the increase in the number of GPs considered had, generally, a slightly damaging effect on the results.

4.2.3. Combined Shear-Tensile Test

In order to create a stress tensor field with significant gradients in the x and y components, it was decided to do a combined test consisting on a shear path followed by a tensile one. The data required for the finite element modelling can be found in the Table 4.3, while the material properties are presented in APPENDIX D: Material Data.

The distribution of the equivalent stress in the deformed configuration of the specimen is shown in Figure 4.7. The black line represents the intersection with the NURBS used in the trimming operation.

Table 4.3 Mesh and process Information

Mesh Data			Phase Data (Displacement):			
	Length:	Number of Elements:	Phase 1:		Phase 2:	
X	20	40	ΔX	3	ΔX	0
Y	10	20	ΔY	0	ΔY	1
Z	1	2	ΔZ	0	ΔZ	0

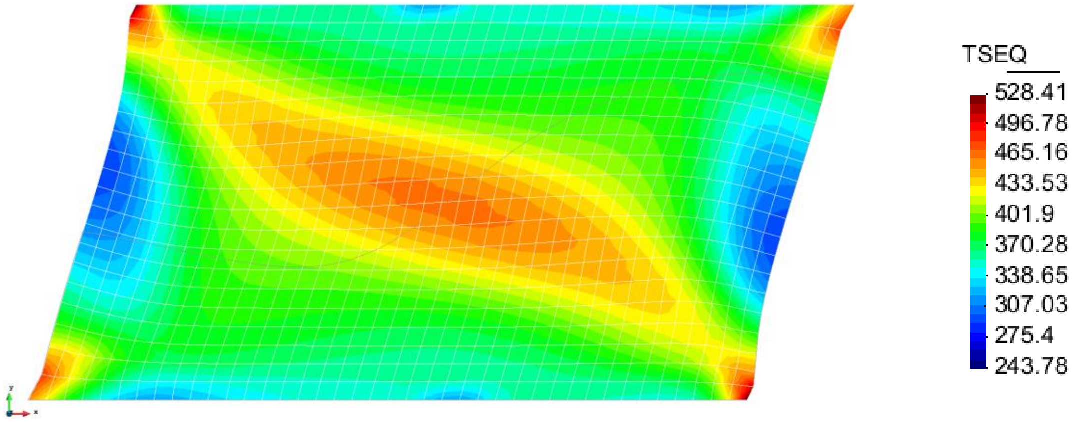
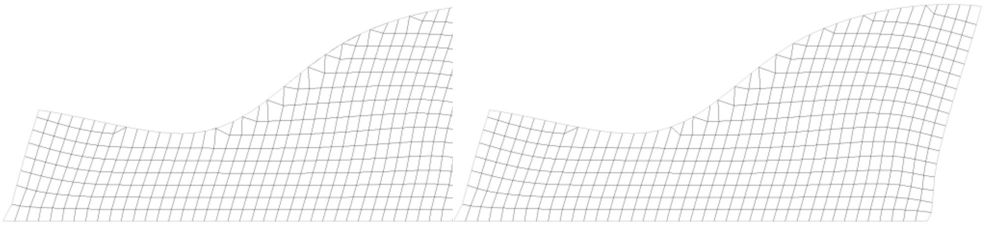


Figure 4.7 Equivalent stress distribution in the specimen

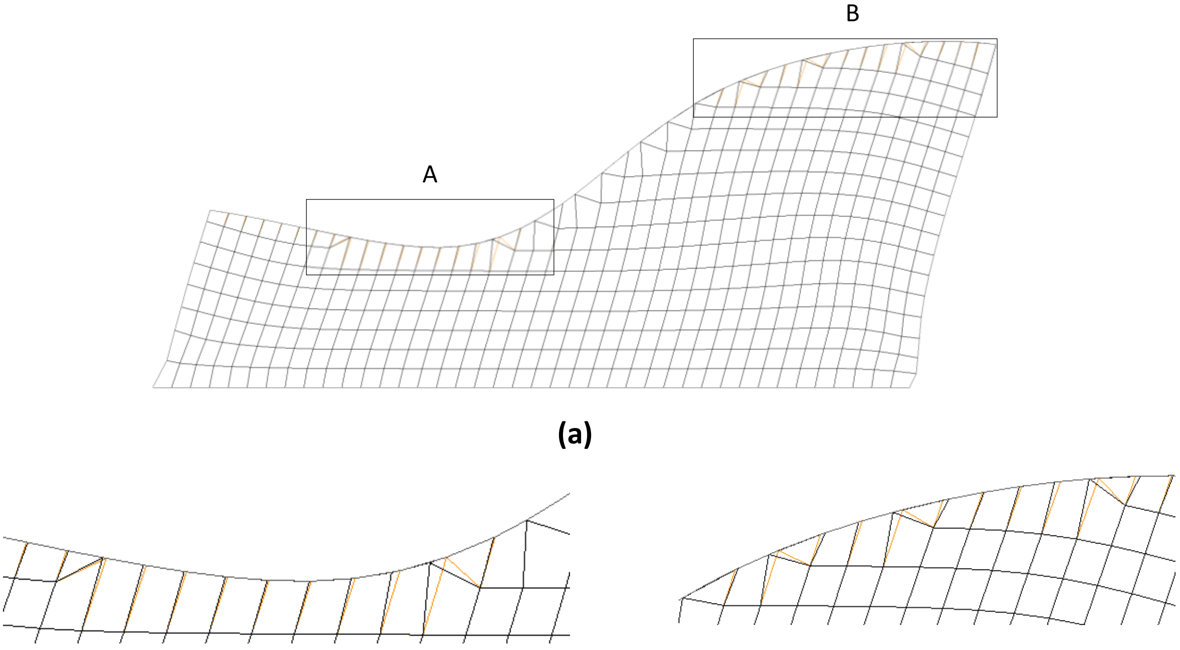
4.2.3.1. Correction Methods

The comparison of the two correction methods used in the trimming stage are presented in Figure 4.8. For this example, the differences are more subtle. Therefore, in order to highlight the dissimilarities between the two finite element discretizations, the two meshes are overlapped, as shown in Figure 4.9 (a), with zoomed details in the same figure. The correction method II is shown in black and the correction method III in yellow. The difference in the nodes positions is not significant, consequently the same is true for the remapping results. Thus, in the subsequent analysis the correction method II will be adopted for the trimming operation.



(a) **(b)**

Figure 4.8 Correction method comparisons (a) Projection (2nd) (b) Intersection (3rd)



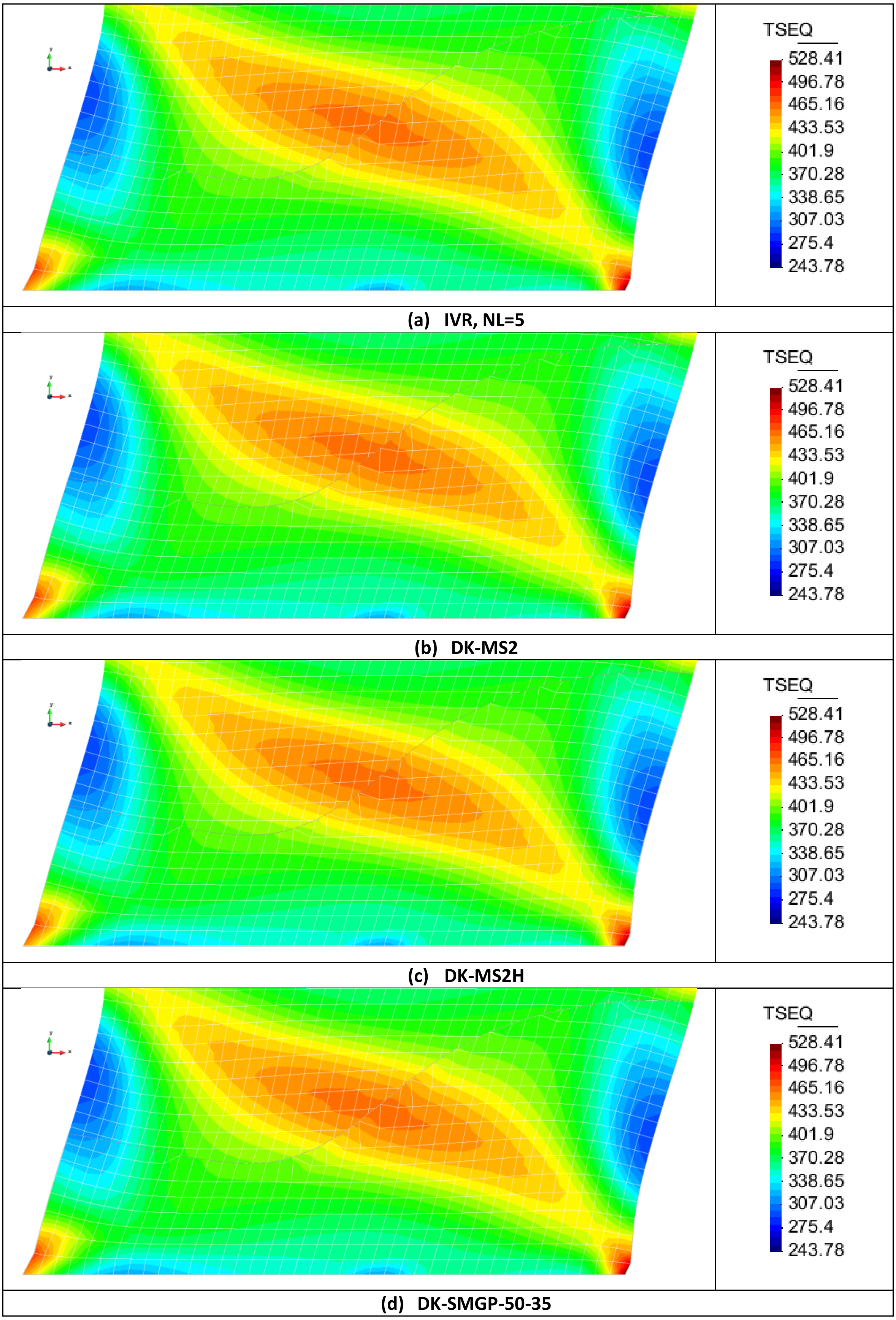
(a)

(b) **(c)**

Figure 4.9 Overlay of the meshes generated with both correction methods: (a) identification of the zones to be zoomed in (b) zone A and (c) zone B

4.2.3.2. Remapping

Figure 4.10 presents the original distribution of the equivalent stress in the topside of the trimming line, while the bottom results correspond to the ones obtained with the different remapping methods. The differences between methods are more perceptible near the central area of the specimen. Near the maximum values, IVR provides a slightly better result, since the transition between the original and remapped results is smoother. This is correlated with the mesh's geometry. DK-Master Slave, most likely, has defined the Master Element as one with more GPs in the "maximum" colour range, which resulted on a slight increase in value of the new GP. SmGP was able to obtain the best gradient near the dark blue area on the left side, IVR and DK-MS2 performed equivalent results in that area (see APPENDIX F: Zoomed Views Combined Shear-Tensile).



(a) IVR, NL=5

(b) DK-MS2

(c) DK-MS2H

(d) DK-SMGP-50-35

Figure 4.10 Remapping Comparison: Combined Shear-Tensile

The difference, between MS2 and MS2H, is small, which arises mostly in the contracted elements. Nevertheless, DK-MSxH can do something quite different than DK-1. In the special case that the new Master Element is in the removed side of the mesh, and the GP is inside the polyhedron defined by its GPs, the method will only consider those. Conversely, DK-1 will only consider the GPs of the original, independently of their location. Globally, the best results are the ones for IVR and DK-Master Slave, both MS2 and MS2H. The main difference lies in the partially adjusted elements, which had only one or two GPs moved due to the trimming correction. The MS3H had extremely similar results to MS2H.

The computational time spent on the trimming and the remapping itself is presented in Table 4.4, for both correction methods and all remapping algorithms. Considering the remapping performed with the IVR method, the trimming operation takes almost the same time as the remapping, while the consensus was that remapping was the time consuming part. Although, it should be noted that the trimming operation involves the correction of their position which is quite sensitive to the complexity of the trimming surface. Regarding the remapping algorithms, as expected, DK-1 is the fastest of all the methods, taking about 26% of the time necessary by IVR. Master-Slave's algorithm has an additional source of computational time: acquiring the additional GPs from the Slave Elements. Intuitively, the standard set of Master-Slave (DK-MS2) algorithms should be slower than DK-1 and DK-MS1H because it is forced to consider more points from the neighbourhood. However, this intuitive analysis is not fully corroborated by the small difference in computational time. By comparing the different variants (1, 2, 3), there is some consistency between the increase of computational time and the number of gauss points used in the evaluations. Between the correction methods (II and III), the differences are not significant. By comparing the same variant of DK-MSx and DK-MSxH, it can be postulated that the increase in computational time is due to the occasionally unhelpful verification if the GP is inside the GP6Polyhedron or not. Due to the formulation of method II, it is easier for a GP to end up inside another GP6Polyhedron, as the distorted elements helps reduce the distance between GPs. For method III, this verification may yields unfavourable results more often, depending on the orientation of the trimming surface. DK-SmGP was only marginally affected by the change in correction method.

Table 4.4 Execution times for each remapping option[4]

	Method II	**(% IVR)**		**Method III**	**(% IVR)**
Trimming	8,898	98%		9,475	103%
Only inside the element					
DK-1	2,438	27%		2,333	25%
Master-Slave					
DK-MS1	2,578	28%		2,776	30%
DK-MS2	2,776	31%		2,714	29%
DK-MS3	2,896	32%		2,958	32%
Master-Slave Hybrid					
DK-MS1H	2,589	29%		2,854	31%
DK-MS2H	2,786	31%		2,740	30%
DK-MS3H	2,844	31%		2,891	31%
Smart GP					
DK-SmGP (50-35)	3,240	36%		3,271	35%
IVR					
IVR (NL-5)	9,073	100%		9,224	100%

4.3. Dependent variables correction analysis

The correction of the dependent variables was implemented to ensure consistency between the stress tensor and the flow stress (Baptista 2006). This ensures that, when a new forming process begins, the stress tensor corresponds to a state that is either inside or on the yield surface, as schematically shown in Figure 4.11. Therefore when performing the subsequent stage, an increase in the applied stress, will either make the GP have an increase in elastic and/or plastic deformation, depending of its initial state, which will also assure the consistency conditions. Otherwise, if the initial stress state is not consistent with the flow stress, convergence issues may arise in the subsequent forming process, leading to incorrect results (e.g. unrealistic increase in the hydrostatic component) or divergence.

[4] Average time for a set of 5 executions on a i7-4980HQ (2.40GHz-3.60GHz)

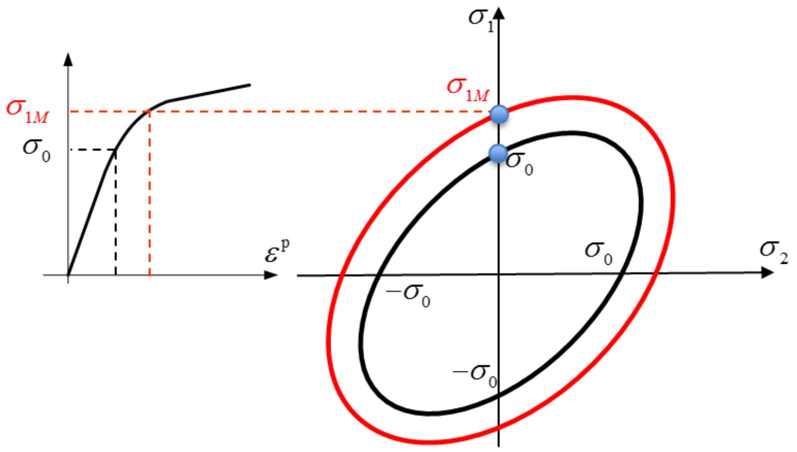


Figure 4.11: Von Mises Yield Surface with example of work hardening

In order to check the consistency conditions it is necessary to know the constitutive model adopted as well as its parameters. However, the Ufo file does not store the information regarding the material parameters, because the mesh can be composed by elements with different constitutive models. Therefore, in order to make the consistency check it is necessary that the file(s), which contains all material properties, exist in the same directory as DD3TRIM. All files are named DD3_materX.dat, where X is an integer associated with the material number. Instead of using these files, an alternative could be to have them incorporated into the Ufo file, but that would requiring a revision of DD3IMP's *filing* routines.

The correction of the flow stress is performed by calculating three values:

- Yield stress for the material, i.e. the stress for which the material will enter in the plastic deformation;
- Yield stress given by the hardening law of the material, based on its parameters and the interpolated values of the equivalent plastic deformation,
- The current equivalent stress, calculated using the yield criterion and the effective component of the interpolated values of the Cauchy stress tensor (deviatoric component of the Cauchy stress tensor minus the back stress tensor).

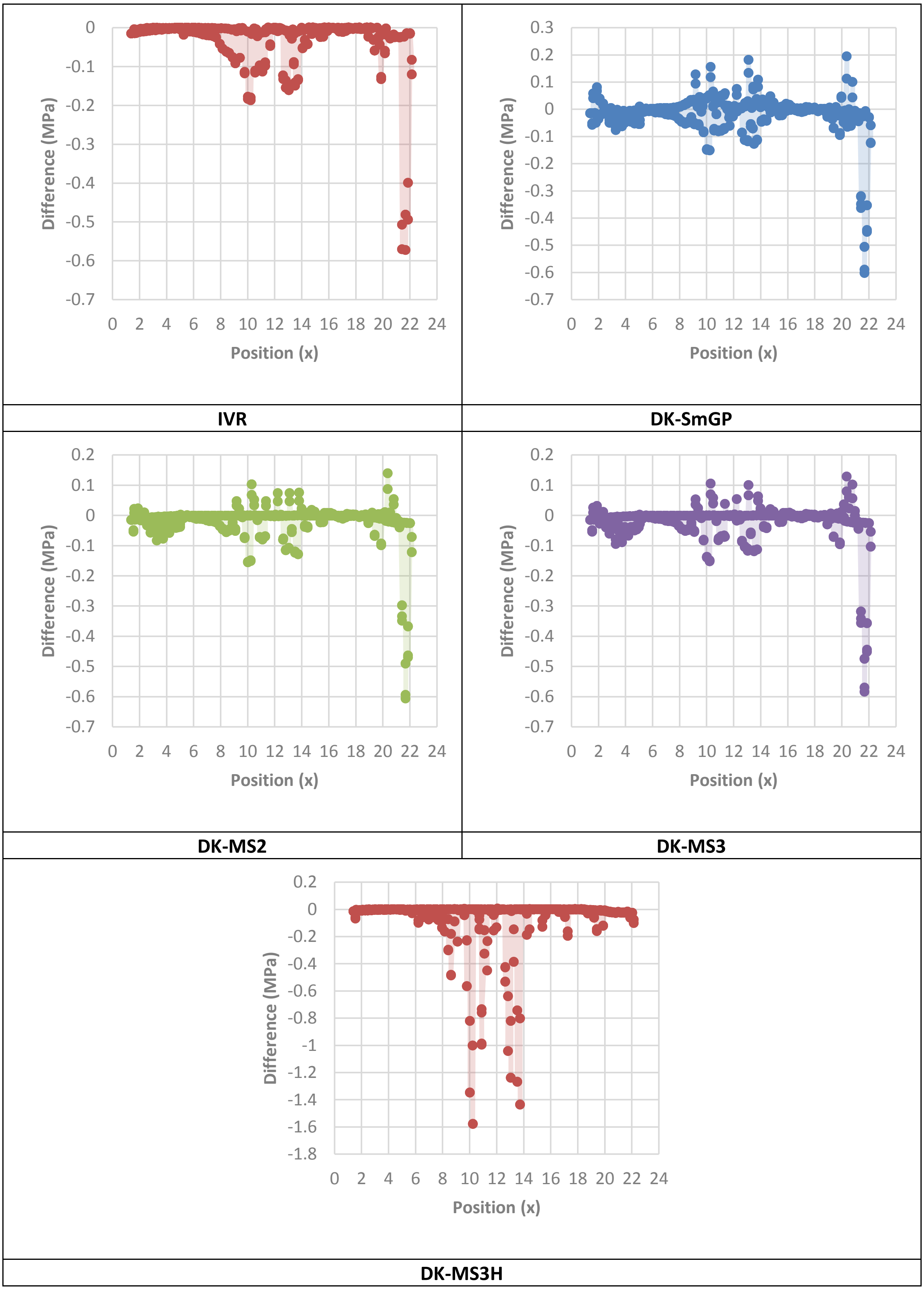
The maximum of the first two values is defined as the new flow stress. The interpolated equivalent stress is also replaced by the one calculated from the interpolated values of the Cauchy stress tensor and backstress tensor. For the GPs that remain unchanged by the trimming operation the consistency condition is assured. For the new GPs it is considered that the necessity of performing this operation is an indirect measure of the error associated to the remapping procedure. Thus, the example presented in section 4.2.3 is recovered to evaluate the impact of the remapping algorithm in the arising of problems in consistency.

4.3.1. Results Analysis

An output file was generated containing all the original and remapped values. By taking these, it was possibly to evaluate numerically the difference between the interpolated value of the flow stress and the corrected one. Consider the trimming section shown in Figure 4.7, the relation between the horizontal coordinate and the correction required is shown in the Figure 4.12. The interpolation of the flow and equivalent stress was performed accurately by the methods, i.e. the interpolated values of equivalent and flow stress are equal, which was expected since the distribution is also equal. The problem may be the equivalent plastic strain, since the small changes in this value lead to small variations when calculating the flow stress, when using the hardening law.

These graphics were made by extracting the information regarding the correction and the position of the Gauss Points. The difference in the Y-Axis was calculated by subtracting, from the interpolated, the new value.

Due to the mathematical nature of the methods, IVR is unable to obtain values above the calculated flow stress. In fact, IVR is similar to a weighted average, therefore it can only produce values between the original ones. On the other hand, DK may overestimate or underestimate the value. Comparing the DK methods, DK Master Slave yields better results than DK SmGP, near the zone with the maximum value. Near the right side, some variations are common to almost all the methods, DK-MS3H was the only method that had all values near zero or negative, as IVR. This can be partly explained because this particular method focuses on the closest GP. As expected, all the methods originate some variation and its intensity and location is approximately the same, except for DK-MS3H, which had bigger variations in the central zone.



IVR

DK-SmGP

DK-MS2

DK-MS3

DK-MS3H

Figure 4.12 Overall comparison of the location (coordinate Ox) of the difference between the interpolated and the corrected values of Flow Stress

Another important factor is the frequency of the correction. The ideal is that most corrections are made to small values. The histogram in Figure 4.13 presents the relation between the correction value and the frequency of occurrence. The logarithmical scale was applied to the vertical axis since there is a huge discrepancy between the maximum and minimum values.

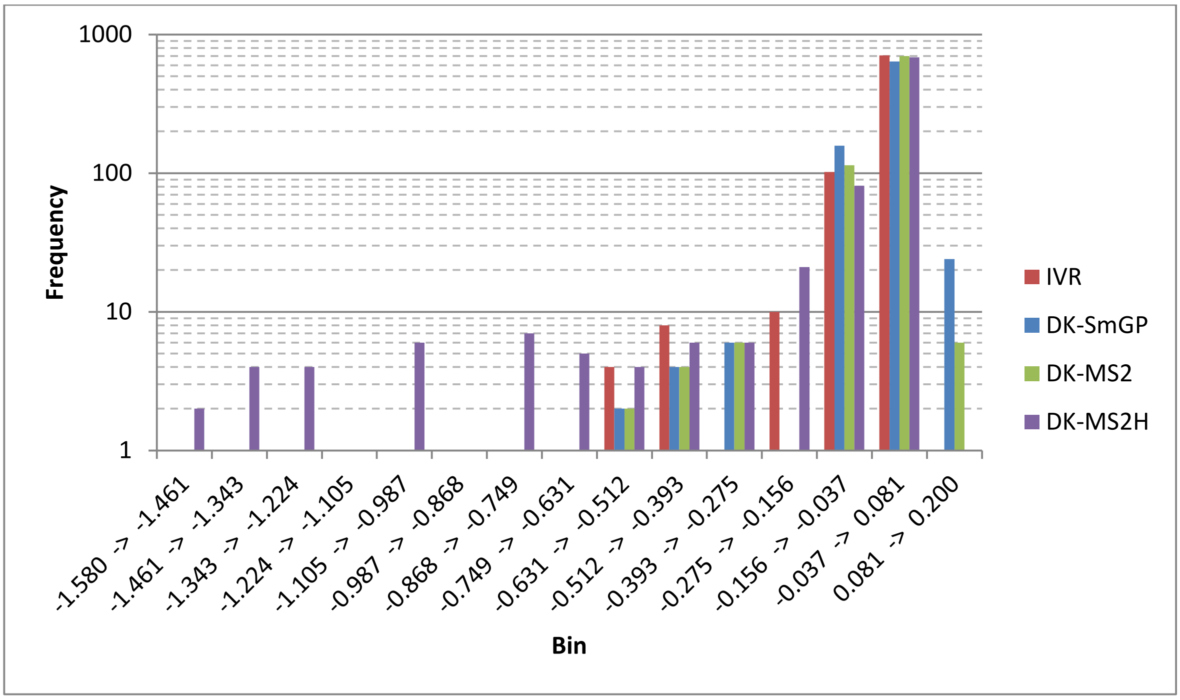


Figure 4.13 Histogram of the values of the correction

All the methods had most values between -0.037 and 0.081, which is globally a very positive result for all. The distribution of the frequency is very similar between MS2 and SmGP, since the number of GPs considered will not change much between them. DK-MS2H conversely, displayed a different behaviour, with more intense corrections than the ones from DK-MS2. This leads to the conclusion that MS2H in its current state is not as effective as the alternatives. Nevertheless, the value of the corrected flow stress is also important, as it is less than 1% of the minimum interpolated value. It is clear that DK-MS2H was affected by the maximum value of the corrections, although it only occurred for a small number of GPs.

A comment should be made regarding on why the interpolated equivalent strain is defined as the correct value. In the finite element method, the calculation of many state variables is performed after the calculation of the deformation. Moreover, the equivalent plastic strain is, by definition, a cumulative variable. Therefore, taking into account DD3IMP’s source of variables (displacement → deformation → stress), considering the equivalent plastic strain is ideal.

It is also important to consider how DD3IMP will react to this, it is imperative that the equivalent plastic strain, equivalent stress and flow stress are coherent. Because the first operation performed by DD3IMP will always be the one-step-springback, i.e. the calculation of the new equilibrium state.

4.4. External remapping

4.4.1. Validation with Scalar Interpolation

In the work Baptista (2006), an evaluation benchmark was used to compare the results of the different remapping algorithms, when using external meshes. In this section this benchmark is applied to the Dual Kriging algorithms developed.

The benchmark involves the remapping of a known mathematical function from a globally structured mesh (Mesh 1) to a semi-structured (Mesh 2), called Stage 1. Afterwards, the remapping procedure is repeated, from the semi-structured mesh to the original (globally) structured mesh, called Stage 2. The schematic of this procedure is indicated in Figure 4.14, as well as both discretizations adopted (structured, semi-structured). The information and main features of each mesh are presented in Table 4.5.

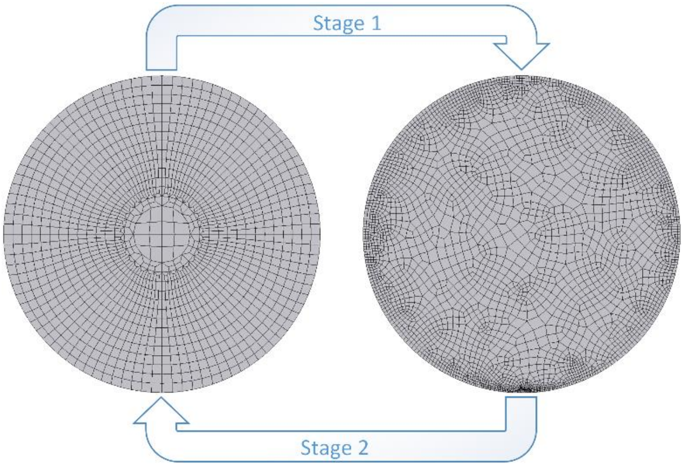
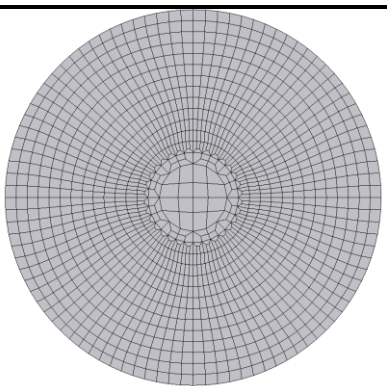
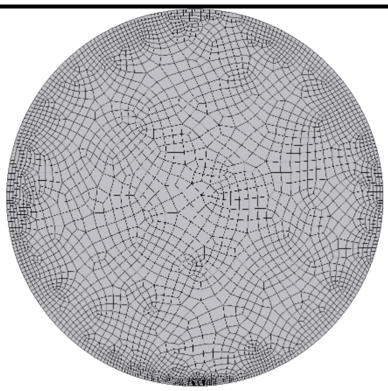


Figure 4.14 External Remapping Stages

Table 4.5 Mesh Information for the external remapping example

Mesh:	Mesh 1:	Mesh 2:
Circle with 50mm as radius Centre: (0,0) 1mm in thickness Structured along OZ Two layers		
Number of Elements:	2 688	6 694
Number of Nodes:	4 179	10 575

The mathematical function is given by:

$$T(r,\theta) = 20r^2(r-1)^2\cos(2\theta)\,, r = \sqrt{\frac{x^2+y^2}{a}}, \tag{4.2}$$

where x and y are the coordinates in the plane, a is the mesh maximum radius, r and θ are the cylindrical coordinates associated to the point. The main advantage of using this benchmark is that the error induced by the remapping procedure, in the analysed state, can be evaluated. This involves the comparison between the initial and final stage, subtracting the interpolated value, from the analytical one. Thus, the error is calculated by:

$$E(r,\theta) = T(r,\theta) - I(r,\theta), \tag{4.3}$$

where T is the value of the mathematical function in the GP and I is the interpolated value in the same GP. The values were calculated for each GP in the starting phase.

Two comparisons will be presented in the following sections. The first one aims to compare the effectiveness of the remapping method when there is no gradient through-thickness. The second considers a fictitious gradient and aims to evaluate the impact of a severe gradient through-thickness and the impact of considering old GPs far from the one being calculated. This gradient is created by changing the value of each of the four GPs along the thickness according to the configuration: $T; 0{,}1 \times T; -0{,}1 \times T; -T$, which corresponds to a cubic through-thickness distribution.

To enable a fair comparison between the SmGP method and the IVR one, the reference established was an equal computational time. That is, as long as it took approximately the same time then they are comparable. Nevertheless, it should be mentioned that theoretically the IVR is a convergent method while DK-SmGP can be a divergent one, in the sense that IVR will, perform a more accurate comparison with the increase of NL, meanwhile, DK-SmGP can perform worse interpolation, as the useful information may get outnumbered by useless. The execution parameters adopted for each remapping method are presented in Table 4.6. Note that the SmGP Master-Slave took about 60% of the comparison time, which is about 3 minutes and 20 seconds. . For the finite element meshes used, 60 GPs corresponds to approximately seven elements. For DK-SmGP, the most time consuming part of the algorithm is the iterative process of finding the GPs, i.e. the algorithm for the cubic distance method.

Table 4.6 Parameters for each method

IVR	**DK Hybrid**	
Number of divisions (NL)	**Minimum number of GPs to find**	**Number of GPs to consider**
5	60	50

4.4.1.1. Without through-thickness gradient

The first stage comprises the remapping from mesh one to mesh two. The function value distributions after remapping are presented in Figure 4.15 for different remapping methods. Globally, all distributions match the original one, as can be seen by the use of the same scale for the interpolated values. However, the error (see (4.3)) gives quite different values, justifying the use of different scales for each method. The transition zone of the original mesh, near its centre (Figure 4.15), originates some error independently of the method adopted. The zone where an extrapolation occurs, i.e. the boundary of the mesh, also has a considerable error associated, as shown in Figure 4.15.

The error in the remapping performed with the IVR method tends to spread more uniformly across the mesh. Conversely, DK tends to focus the error in the regions surrounding the extremes and the boundary area. Between the DK methods, the SmGP does not spread the error as much as SmGP Master-Slave. The dimension of the mesh along the thickness is extremely small, Consequently SmGP considers all four GPs in the thickness direction.

The maximum and minimum values of the state variable, after applying each remapping method, are displayed in Table 4.7. IVR underestimates the maximum value while returning a minimum value close to the analytical. Conversely, DK overestimates the extreme value but, for the maximum values, the accuracy of the result is comparable to the one of the IVR method.

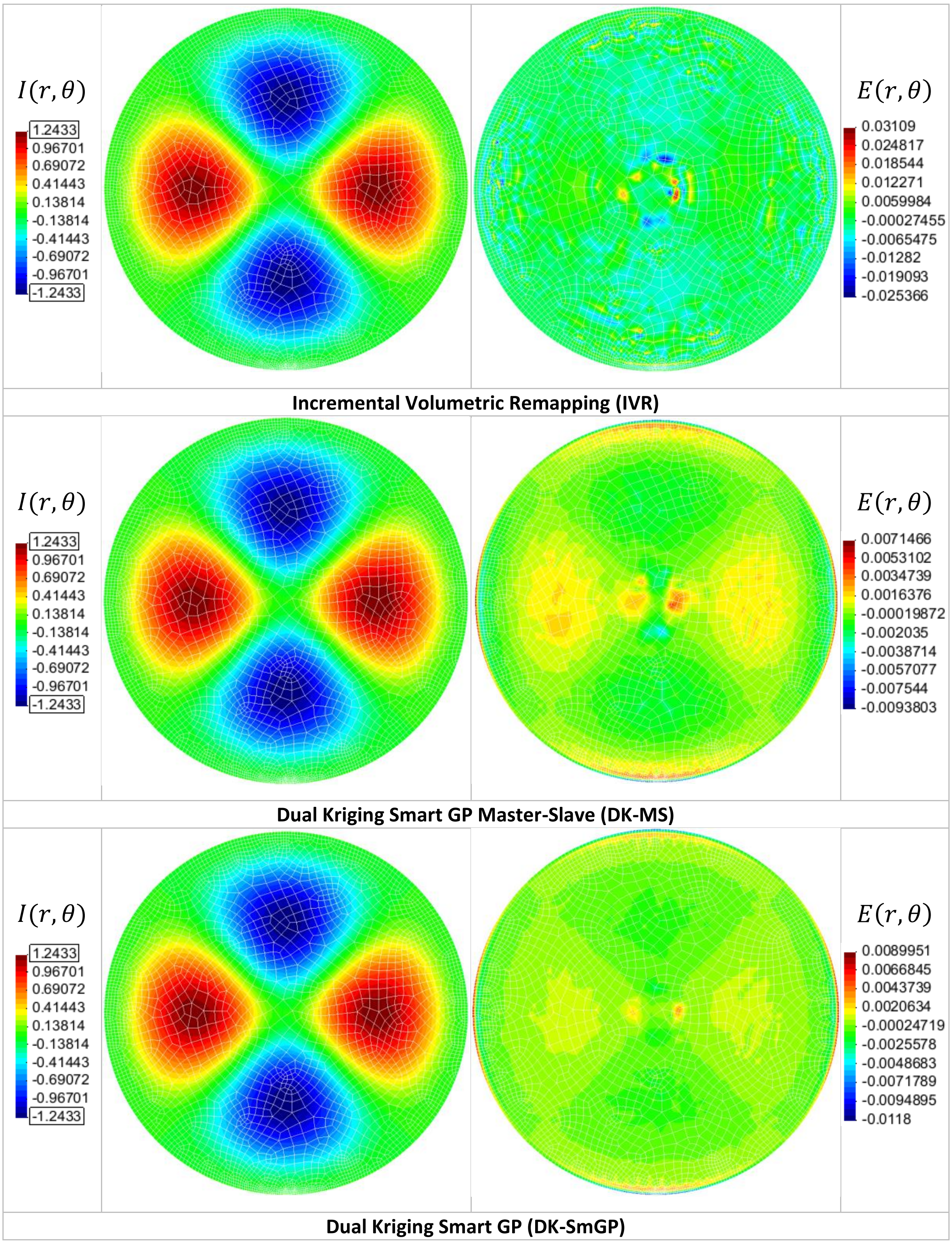
Table 4.7 Comparison of Extreme Values - No Gradient on Thickness (Stage 1)

Mesh 2	Value of the Variable					
Origin		Difference			Difference	
	Maximum	Absolute	Relative	Minimum	Absolute	Relative
Analytical	1.2418	0.0000		-1.2418	0.0000	
IVR	1.2402	-0.0016	-0.129%	-1.2417	0.0001	-0.008%
DK Smart GP Master Slave	1.2429	0.0011	0.089%	-1.2428	-0.0010	0.081%
DK Smart GP Mode	1.2433	0.0015	0.121%	-1.2433	-0.0015	0.121%

Table 4.8 presents the maximum and minimum error when compared to the analytical function, i.e. the difference between the interpolated value and the analytical one. Unlike the previous comparison, in the maximum and minimum errors, DK performed substantially better.

Table 4.8 Comparison of Extreme Errors - No Gradient on Thickness (Stage 1)

Mesh 2	Error of the Variable					
Origin		Difference			Difference	
	Maximum	Absolute	Relative	Minimum	Absolute	Relative
IVR	0.0311	0.0311	2.504%	-0.0254	-0.0254	2.045%
DK Smart GP Master Slave	0.0071	0.0071	0.575%	-0.0094	-0.0094	0.755%
DK Smart GP Mode	0.0090	0.0090	0.725%	-0.0118	-0.0118	0.950%



Incremental Volumetric Remapping (IVR)

Dual Kriging Smart GP Master-Slave (DK-MS)

Dual Kriging Smart GP (DK-SmGP)

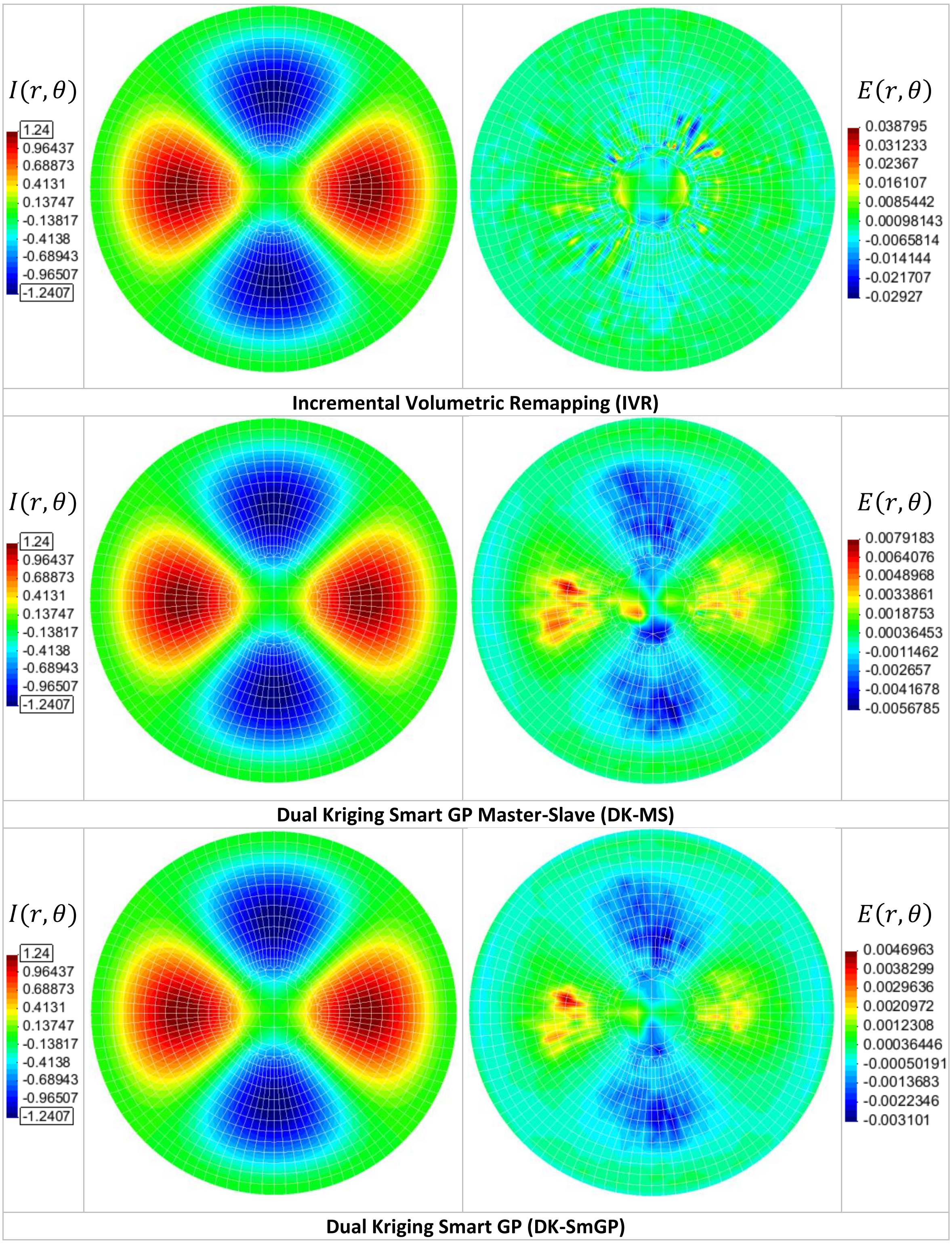
Figure 4.15 Distribution of the interpolated variable and the error Method 1 (Stage 1)

The results concerning the remapping from the semi-structured to the original structured mesh (Stage 2) are presented in Figure 4.16. The IVR method presents an uneven distribution of the error. Conversely, for the DK method the higher absolute error values focus in the regions surrounding the extreme values of the state variables. Comparing SmGP and Master-Slave, it is clear that considering the nodes along the thickness had a positive effect on the overall quality, because the through-thickness values are the same. During this sage, the boundary area of the mesh presents no severe issues.

Table 4.9 Comparison of Extreme Values - No Gradient on Thickness Stage 2

Mesh 1	Value of the Variable					
Origin		Difference			Difference	
	Maximum	Absolute	Relative	Minimum	Absolute	Relative
Analytical	1.2418	0.0000		-1.2418	0.0000	
IVR	1.2382	-0.0036	-0.290%	-1.2385	0.0033	-0.266%
DK Smart GP Master Slave	1.2387	-0.0031	-0.250%	-1.2399	0.0019	-0.153%
DK Smart GP Mode	1.2407	-0.0011	-0.089%	-1.2407	0.0011	-0.089%

The comparison of the extreme values after Stage 2, i.e. the transfer from Mesh Two to Mesh One, are presented in Table 4.9. Upon returning to the original mesh, the error for the extremes diminishes. Unlike for Stage 1, the Master Slave Method provides the best results. The extreme values of the errors are compared in Table 4.10. As expected, the difference is smaller in the Master Slave Method. Similarly to stage 1, DK provides a smaller overall error. Considering Table 4.7 and Table 4.9, it is possible to conclude that DK is more sensitive to maximum and minimum values. Concerning the remaining remapping methods, the error is always smaller than the one predicted with the IVR (see Table 4.8 and Table 4.10). Globally, the results of both stages show that all remapping methods provided good results, in the absence of gradient through-thickness.



Incremental Volumetric Remapping (IVR)

Dual Kriging Smart GP Master-Slave (DK-MS)

Dual Kriging Smart GP (DK-SmGP)

Figure 4.16 Distribution of the interpolated variable and the error Method 1 (Stage 2)

Table 4.10 Comparison of Extreme Errors - No Gradient on Thickness Stage 2

Mesh 1	Error of the Variable					
Origin		Difference			Difference	
	Maximum	Absolute	Relative	Minimum	Absolute	Relative
IVR	0.0388	0.0388	3.124%	-0.0293	-0.0293	2.359%
DK Smart GP Master Slave	0.0078	0.0078	0.626%	-0.0055	-0.0055	0.445%
DK Smart GP Mode	0.0047	0.0047	0.378%	-0.0031	-0.0031	0.250%

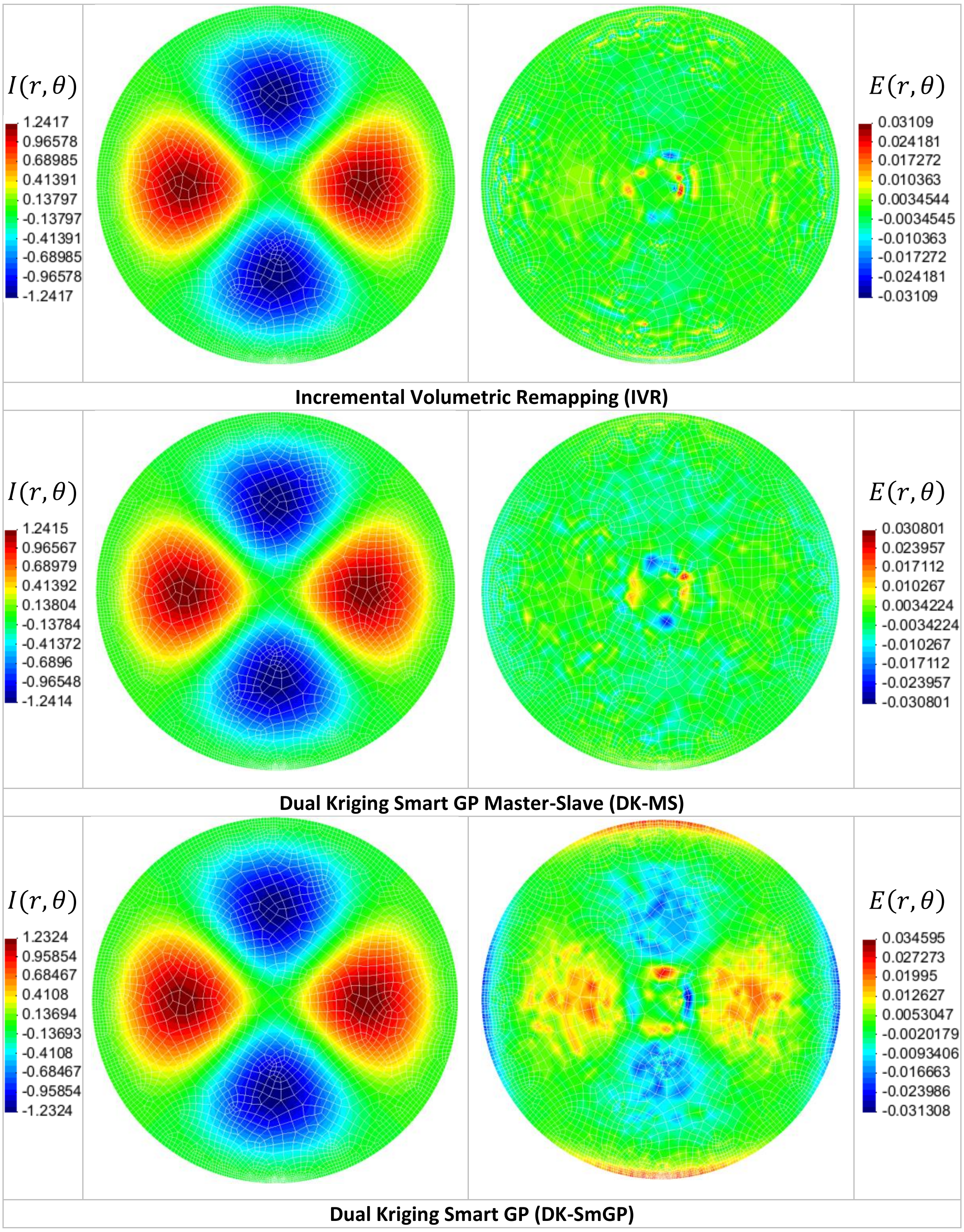
4.4.1.2. With through thickness gradient

This section presents the results for the same benchmark, but considering a gradient of the state variable along the thickness. Therefore, the structure and organization of this section is similar to the previous one. The distribution of the state variable and error, after remapping are presented in Figure 4.17 for the different remapping methods. The analysis of the error distributions indicates that the DK MS and IVR methods provide similar distributions. Conversely, DK-SmGP tends to spread the error more uniformly, since more points are considered in the interpolation of each GP. For this method, the error is more evident in the boundary of the mesh, unlike the remaining. The analysis of the extreme values is presented in

. The DK-SmGP Master-Slave provides values very similar to IVR. The DK-SmGP method obtained the worst results. When considering the extreme errors (see Table 4.12), it is clear that all methods lead to quite similar values. The existence of a gradient along the thickness affected all methods, bringing them to an equivalent plateau.

Table 4.11 Comparison of Extreme Values - Gradient along the Thickness (Stage 1)

Mesh 2	Value of the Variable					
Origin		Difference			Difference	
	Maximum	Absolute	Relative	Minimum	Absolute	Relative
Analytical	1.2418	0.0000		-1.2418	0.0000	
IVR	1.2417	-0.0001	-0.008%	-1.2417	0.0001	-0.008%
DK Smart GP Master Slave	1.2415	-0.0003	-0.021%	-1.2414	0.0004	-0.035%
DK Smart GP Mode	1.2324	-0.0094	-0.757%	-1.2324	0.0094	-0.757%



Incremental Volumetric Remapping (IVR)

Dual Kriging Smart GP Master-Slave (DK-MS)

Dual Kriging Smart GP (DK-SmGP)

Figure 4.17 Distribution of the interpolated variable and the error Method 2 (Stage 1)

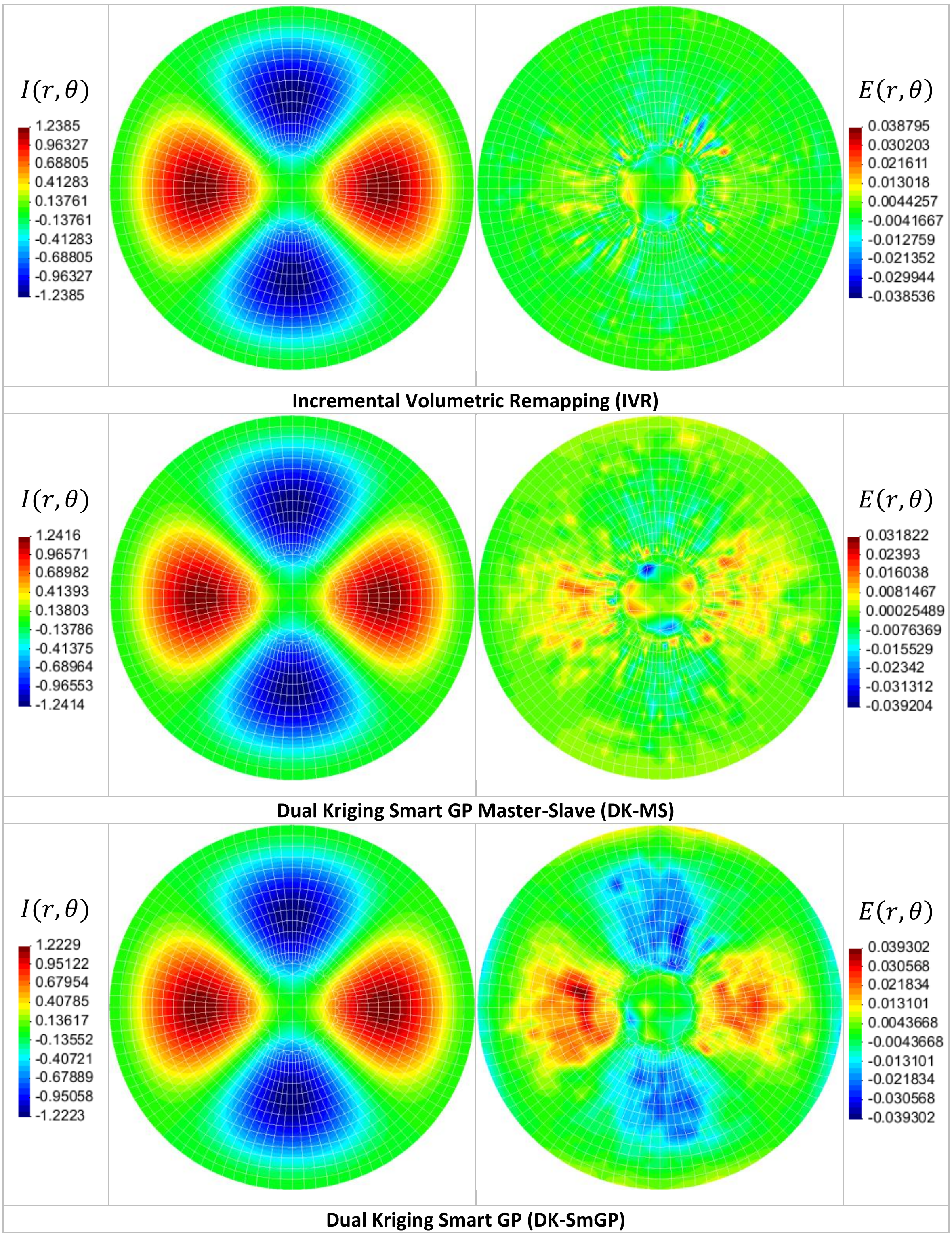
Table 4.12 Comparison of Extreme Errors – Gradient along the Thickness (Stage 1)

Mesh 2	Error of the Variable					
Origin		Difference			Difference	
	Maximum	Absolute	Relative	Minimum	Absolute	Relative
IVR	0.0311	0.0311	2.504%	-0.0311	-0.0311	2.504%
DK Smart GP Master Slave	0.0308	0.0308	2.480%	-0.0308	-0.0308	2.480%
DK Smart GP Mode	0.0346	0.0346	2.785%	-0.0313	-0.0313	2.521%

The obtained distributions for the second stage, when the function values are returning to the original mesh, are displayed in Figure 4.18. In this case, the error distribution obtained with DK-MS method shows a wider spread than the one obtained with IVR. In addition, this occurs not only near the internal transition, but also near the extremes. On the other hand, the DK-SmGP originated large zones with higher error values, when compared to the other methods. Nevertheless, it is also important to mention the error stacking from the first to the second stage. Note that the distributions on the inferior side of the mesh are similar to their correspondent on the superior side, just rotated 90º. Upon returning to the original mesh, DK MS improved the representation of the extreme values, obtaining the best values for Stage 2. SmGP performed poorly when compared to the other methods. Globally, all methods have similar extreme error values, as shown in Table 4.14. Nonetheless, DK-SmGP Master Slave was able to perform better controlling the maximum error by 0.6%.

Table 4.13 Comparison of Extreme Values - Gradient along the Thickness (Stage 2)

Mesh 1	Value of the Variable					
Origin		Difference			Difference	
	Maximum	Absolute	Relative	Minimum	Absolute	Relative
Analytical	1.2418	0.0000		-1.2418	0.0000	
IVR	1.2385	-0.0033	-0.267%	-1.2385	0.0033	-0.267%
DK Smart GP Master Slave	1.2416	-0.0002	-0.016%	-1.2414	0.0004	-0.032%
DK Smart GP Mode	1.2229	-0.0189	-1.522%	-1.2223	0.0195	-1.570%



Incremental Volumetric Remapping (IVR)

Dual Kriging Smart GP Master-Slave (DK-MS)

Dual Kriging Smart GP (DK-SmGP)

Figure 4.18 Distribution of the interpolated variable and the error Method 2 (Stage 2)

Table 4.14 Comparison of Extreme Errors - Gradient along the Thickness (Stage 2)

Mesh 1	Error of the Variable					
Origin		Difference			Difference	
	Maximum	Absolute	Relative	Minimum	Absolute	Relative
IVR	0.0388	0.0388	3.124%	-0.0385	-0.0385	3.103%
DK Smart GP Master Slave	0.0318	0.0318	2.563%	-0.0392	-0.0392	3.157%
DK Smart GP Mode	0.0393	0.0393	3.165%	-0.0393	-0.0393	3.165%

4.4.1.3. Overall Comment

An important remark concerning the remapping of the through-thickness state variable gradients, in the example of section 4.4.1.2, should be mentioned. Although the remapping results for both surfaces are quite accurate, for all the developed methods, the same is not valid for the through-thickness values. In fact, IVR method clearly provided the best results along the thickness, when compared to the implemented forms of DK. The option of not showing the through-thickness remapped values resulted from the difficulties experienced in their discussion. In fact, as shown in section 4.2.2, for the bending example the remapped results were quite similar for all analysed methods. The fact that in the benchmark test there is a more severe gradient along the thickness, combined with the reduced size of the elements may be a source of error. The impact of the gradient along the plane may also contribute to this situation. In fact, the IVR method performs a weighted average to obtain the values for each GP. On the other hand, with the DK method there is always the risk of over or underestimating the values, even if the GP is located within the interpolation domain, depending on the generalized covariance function adopted.

Several factors also influenced the distributions of results in the mesh, independently of the trial:

Boundary of the mesh: For both DK methods, this is an area where errors are likely to occur. Due to the disparity in element size (GPs will be closer to the boundary itself), it is clear that DK MS may perform a slight extrapolation.

Too many points: For DK-Hybrid and this test, the use of too many points can have a damaging effect on the results. When trying to select the GPs to consider, the algorithm will usually pick all the GPs in the thickness before reaching another GP in the plane. This is clear in the second example of the benchmark, as the values are symmetric along the mid thickness plane.

4.4.2. Brute Force Method

During the course of this work, the brute force method was applied successfully for a small data set (McLean et al. 2006) for testing purposes. The time necessary to invert the Kriging Matrix, which includes all the GPs in the mesh, see (3.2), was evaluated for an increasing number of GPs and is presented in Figure 4.19. The reader is reminded that the dimension of each matrix is the total number of points to consider, n, plus four,

$$Total\ Inversion\ Time = n \times Inversion\ Time_{n+4}, \tag{4.4}$$

Below 230 points, most values were null and therefore impossible to represent in a logarithmic scale. It is possible to see that, for 6400 GPs (only 800 elements), it would take at least about 15h02min (54 105 seconds) which is completely unfeasible. Another issue is the questionable impact of GPs at extreme distances but with relative extreme values.

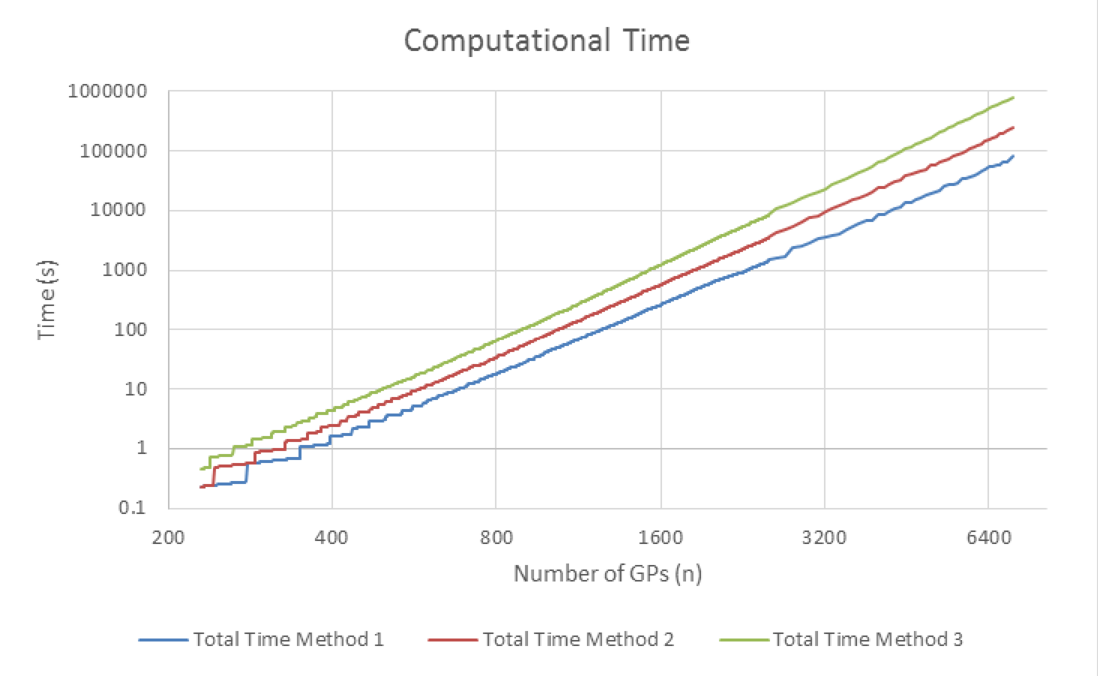


Figure 4.19. Increase in computational time as function of the number of Gauss points used: Influence of the three LAPACK methods implemented

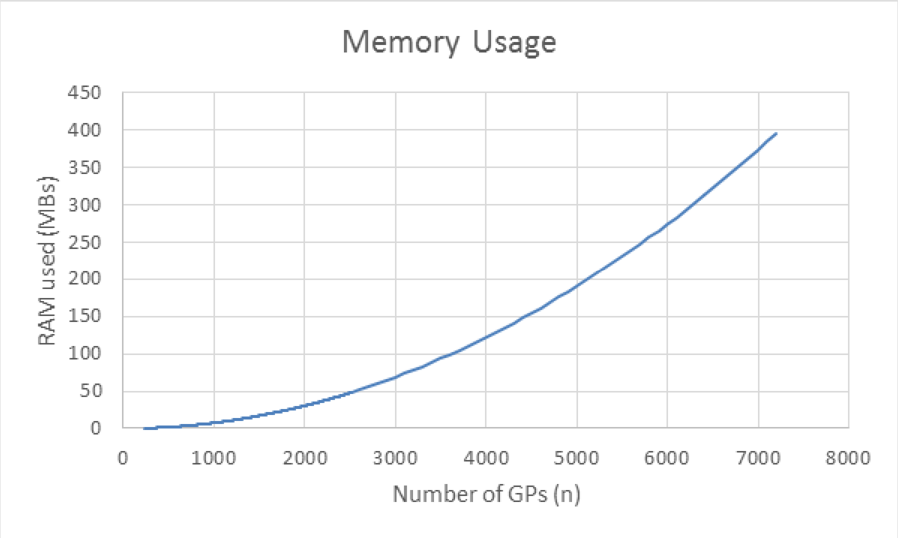


Figure 4.20. Increase in memory usage

In Figure 4.20, it is possible to see the increase in minimum RAM usage, for the same example. Theoretically, the RAM usage should be one-half of the displayed value, but the algorithm is forced, by LAPACK, to have the same matrix allocated twice. The implementation was performed with LAPACK public libraries and their performance was equated, to Matlab matrix inversion algorithm, which is based on the same libraries.

4.5. Summary

Table 4.15 presents the main advantages and potential pitfalls of the Dual Kriging interpolation algorithms developed within this work.

Table 4.15 Dual Kriging interpolation algorithm comparison

Method	Advantages	(Potential) Pitfalls
DK-1	Fastest method (1);	Domain may not contain the new GP.
DK-MSx	Fast method (3); Domain will forcefully include the new GP.	Can consider too much information from the neighbourhood (particularly variant third 3).
DK-MSxH	Fast method (2); Domain will forcefully include the new GP.	Can consider too much information from the neighbourhood (particularly third variant); May generate inconsistencies when switching between methods.
DK-SMGP	Medium speed method (4); Domain will forcefully include the new GP.	Can lead to poor results if the wrong parameters are applied; Depending on the mesh can lead to increased computational time.
Brute Force	Slow method (5); Domain will forcefully include the new GP.	Too much information will be used; Extreme execution time for bigger meshes; May not be able to run due to hardware issues.

5. CONCLUSIONS

The main focus of the work was the DD3TRIM code, which was originally developed to enable the trim and split of solid finite elements meshes, resulting from a forming stage of a sheet metal forming operation. This program performs the trimming and remapping operations, associated with the geometrical cutting operations typically involved in the intermediate steps of multi-stage stamping processes. The main objective of this work was the reprogramming of DD3TRIM, in order to enable its interaction with the more recent version of DD3IMP solver. Therefore, the understanding of the algorithms implemented in DD3TRIM and DD3IMP, coded in FORTRAN language, was the first step to achieve this objective. In addition, the variables involved in both programs (DD3TRIM and DD3IMP) needed to be carefully analysed and shared.

The solutions implemented will allow CEMUC to perform multi-stage forming processes, involving trimming operations. The modifications and improvements implemented in DD3TRIM were tested thoroughly and its ability to provide good results for the original implemented methods verified.

After accomplishing the initial objective, a new remapping method, Dual Kriging, was also implemented in DD3TRIM. The main purpose of this integration was to evaluate if the results could be comparable to the ones of the Incremental Volumetric Remapping (IVR) method. The implementation of this remapping method was completed by integrating high performance computing subroutines (LAPACK), mainly for the inversion of the Kriging matrix. Several methods were developed to select the known data (old GPs) and compared. By taking into account the organization of the mesh related variables, a fast, accurate and intuitive set of methods were developed, Master-Slave, and a new iterative method for categorization of GPs was implemented, Cubic Distance Method. In fact, the use of any of the remapping methods belonging to the Master-Slave set enables performing this operation at around 50% faster than using the IVR method.

Dual Kriging was also compared with IVR in several examples, including trimming examples and a benchmark test applicable for external remapping. Also, the frequency of necessity to correct the dependent variables, in order to assure the consistency condition, was evaluated. Globally, the results show that the Dual Kriging interpolation method is an alternative to IVR, by performing comparably in the trimming examples. In fact, even for the same trimming section, it was not clear which method provided the more coherent results. Depending on the zone, trimming correction and surface itself, one method or another could have slightly better results. Meanwhile, in the benchmark test without a severe through-thickness gradient, the remapping of the scalar variable yielded quite positive results for the Dual Kriging interpolation method. However, when considering the severe gradient in the thickness direction, the results along this direction were not as positive.

Several difficulties appeared while implementing Dual Kriging. Initially all concepts are mentally built, intuitive games of choice and consequence, of trying to assemble a puzzle without any picture on the box, and where all the pieces are random lines that could fit "anywhere", all, while searching for the best possible selection method. Finally, the extraction, analysis and preparation of results is a time-consuming process, but each time it is performed new pieces appear and new pictures can be formed.

Not all that was thought was implemented, that is why some suggestions for Future Work will be presented.

5.1. Future work

Due to the limited amount of time available for this dissertation and the infinite possibilities that the human mind can come up with, several points that were not possible to implement are suggested as future work:

Augmentation of the trimming correction methods: The methods applied were not always the *best*. Sometimes for the same control conditions, the correction methods had a contrasting effect, even for the same trimming section. While some areas were better defined by the second method, others were by the third. A suggestion, although computational expensive, is to perform an iterative process and evaluate the shape of the element. Since the shape of the element is important (Barros et al. 2013), this augmentation would be welcome.

Good coding practices: As said previously, the revision of all the routines by rewriting the code in the same fashion as DD3IMP, for example: incorporation of the description of each subroutine and substituting all variables to the explicit form. The overall usage of modules, although attempted, would not be advisable as a dissertation in Mechanical Engineer per se.

New Dual Kriging Correlation functions: Develop a study on alternative generic correlation functions for the Dual Kriging remapping algorithm. Possibly, include a self-correction option designed to soften the values by taking into consideration the derivatives and original values in the subdomain.

Dual Kriging Expansion:

Planar Dual Kriging: Consider only the nodes in the same plane (GP or element) by resorting to the node connectivity system defined in DK-MSx.[5]

Adaptive MSxH (Upgrade of the current version): Automatic selection of a subset of Gauss Points, in order to create a small variation of the hexahedral domain that contains the new Gauss Point, but is as small as possible.

Indirect Approach: Duplicate the standard Interpolation-Extrapolation by resorting to Dual Kriging to perform the transfer from the old GPs to the nodes and then to the new nodes and GPs. Although this is a tri-phased approach, it can be interesting to quantify the difference between a direct and an indirect approach.

Iterative Dual Kriging Correction: For each new element, calculate the results for each GP in the face opposite to the trimmed face. Several options could be tested: iterative process by adding the previously calculated GPs to the calculation of the new GP, until the changes in values are either null or negligible.

[5] The selection table and main algorithm are already implemented. It is only necessary to modify what to select from the Master Element and the analysis of results.

BIBLIOGRAPHY

Baptista, A.J., 2005. Application of the Incremental Volumetric Remapping Method in the Simulation of Multi-Step Deep Drawing Processes. In *AIP Conference Proceedings*. AIP, pp. 173–178.

Baptista, A.J.C., 2006. *Mechanical Modelling And Numerical Simulation Of The Multi-Step Sheet Metal Forming Process*, PhD Dissertation. University of Coimbra.

Barros, P.D. et al., 2013. Trimming of 3D solid finite element meshes: sheet metal forming tests and applications. *Engineering with Computers*, 31(2), pp.237–257.

Krige, D.G., 1975. A review of the development of geostatistics in South Africa. In *Advanced Geostatistics in the Mining Industry*. pp. 279–293.

Matheron, G., 1975. Les concepts de base et l'evolution de la géostatistique minière. In *Advanced Geostatistics in the Mining Industry*.

McLean, P., Léger, P. & Tinawi, R., 2006. Post-processing of finite element stress fields using dual kriging based methods for structural analysis of concrete dams. *Finite Elements in Analysis and Design*, 42(6), pp.532–546.

Menezes, L.F. et al., 2011. Improving computational performance through HPC techniques: Case study using DD3IMP in-house code. *AIP Conference Proceedings*, 1353(i), pp.1220–1225.

Menezes, L.F. & Teodosiu, C., 2000. Three-dimensional numerical simulation of the deep-drawing process using solid finite elements. *Journal of Materials Processing Technology*, 97(1-3), pp.100–106.

Moaveni, S., 2003. *Finite element analysis: theory and application with ANSYS*, Prentice Hall.

Oliveira, M., Alves, J. & Menezes, L., 2008. Algorithms and Strategies for Treatment of Large Deformation Frictional Contact in the Numerical Simulation of Deep Drawing Process.

Zienkiewicz, O.C., Taylor, R.L. & Zhu, J.Z., 2013. *The Finite Element Method: its Basis and Fundamentals* 6th ed., Elsevier.

ANNEX A: MODULES IMPORTED FROM DD3IMP

The objective of this annex is to list the modules necessary DD3TRIM, will need to be handled consciously in DD3IMP. Because unilateral changes may cause the inability to run DD3TRIM.

Table A. 1 Modules imported from DD3IMP

Module	Contents
ALLOC_FOR_MODULES	Calls the allocation routines responsible for the allocation of the matrices related to the information in the Ufo file.
ALLOC_FOR_READ	Allocates the following arrays (XInit, X, NNode, InTabl, NGauss, NType, MCon, MState, NCR, BCID, NOCB, IRef, ID, ID2, NCGV, XTor, XRef, ELM, BC, TF, TotalNodalForces, PF, XNEDef, DU,CGV).
ALLOC_FOR_saveNST	Allocates arrays related to saveNST, which save the last convergence state.
M_ALLOC_INOUT	Routine responsible for reading and saving some of the information.
M_FILEIO	Stores the typical names for each DD3IMP file.
M_GLOBALS_INOUT	Reads the information required for the allocation of the matrices, for example: NN – Number of nodes, NNDef, Number of deformable nodes, and so forth.
mod_alloc	Contains M_Node, M_BC, M_Element, M_Contact, M_Equations, M_Therm_Alloc, M_Alloc_InOut.
mod_include	Contains M_Globals, M_RMin, M_Run, M_Solver, M_Friction, M_Cyclic, M_Time, M_Constants, M_Integr_Solid, M_Integr_Membrane, M_Therm.
mod_materials	Contains M_Materials, M_Materials_Thermo.
mod_parameters	Contains M_Parameteres.
mod_phase	Contains M_Phase.

Module	Contents
mod_saveNST	Contains M_saveNST, saveNSTdata, readNSTdata.
mod_SurfaceMesh	Contains M_SurfaceMesth.
mod_tools	Contains M_Tool, M_Tool_Bezier, M_Tool_Nagata, M_Tool_Iges.

APPENDIX A: TRIM2.DAT

```
01**************       < DD3TRIM V01.x >   File 'trim2.dat' XX.05.2015      *****************
****************          Generic file to input Trimming parameters         *****************
===================+=========================|===============================================
Name Input Mesh    |
Name Input Ufo     |
-------------------+-------------------------------------------------------------------------
Name Output Mesh   |
Name Output Ufo    |
-------------------+-------------------------------------------------------------------------
Name Surface       |
===================+=========================|===============================================
SUBPROGRAM TO USE: |        1         Info:
-------------------+-------------------------------------------------------------------------
REMAPPING --> 0    |                  Remapp of a given mesh by an UFO file (Academic Use*
TRIMMING  --> 1    |                  Cuts parts from a given mesh
SPLITTING --> 2    |                  Splits rings or other closed meshes
===================+=========================================================================
DATA SOURCE:       |        1         Info:
-------------------+-------------------------------------------------------------------------
DD3IMP FILE --> 1  |                  From data file of type *.UFO
GID FILE    --> 2  |                  From mesh file of type *.MSH
===================+=========================================================================
TRIMMING TYPE:     |        3         Info: For Trimming and Splitting
-------------------|-------------------------------------------------------------------------
By plan --> 1      |
                   |
Equation:          | on1/off0
Ax+By+Cz+D=0       |        0            A         B         C         D
                   |                 1.000     1.000     0.000     1.000
Three points:      | on1/off0
                   |        1        Info: Note that points mustn't be collinear
                   |                     x         y         z
Point 1:           |                  25.0      -2.0       3.0
Point 2:           |                  25.0      -2.0      -3.0
Point 3:           |                  22.0       6.0       0.0
-------------------+-------------------------------------------------------------------------
By Cylinder --> 2  |                   Info: the cylinder is // to zz axis
                   |
                   |                     x         y         z
Centre point:      |                   7.0       6.0       0.0
                   |                     R
Radius:            |                   3.0
-------------------+-------------------------------------------------------------------------
By generic surface |                   Info: File *.igs must be present
  --> 3            |                   NURBS normal oriented towards the eliminating zone
===================+=========================================================================
TRIMMED ZONE       |
-------------------|
Trimming:          |
                   |                     x         y         z
Point of the zone  |                   6.2       6.2       0.0
-------------------+-------------------------------------------------------------------------
Splitting (rings): |
                   |                     x         y         z
Inside Point:      |                   2.0       3.0       0.0
Outside Point:     |                   3.0       3.0       0.0
===================+=========================================================================
CORRECTION TYPE:   |        2          Info: For Trimming and Splitting
-------------------+-------------------------------------------------------------------------
Type I    --> 1    |                   Element elimination
Type II   --> 2    |                   Element elimination + adjustment by projection
Type III  --> 3    |                   Element elimination + adjustment on edge direction
For Type III:      |                     x         y         z
Oriented Vector V  |                   0.0       0.0       0.0   If v(0,0,0) OPTION OFF
```

```
===================+========================================================================
DEGENERATED        | on1/off0
ELEMENTS           |        0          Info: Degenerates elements with pentahedric form
===================+========================================================================
PROCEDD REMAPPING  |
YES -> 1  NO -> 0  |        1          Info: For Trimming and Splitting
-------------------+-----------------------------------------------------------------------
                   |
REMAPPING TYPE:    |        4          Info:
-------------------+-----------------------------------------------------------------------
Type I    --> 1    |                   Base Method: Extrapolation -> Interpolation
-------------------+-----------------------------------------------------------------------
Type II   --> 2    |                   Minimize Functional: Moving Least Squares Method
-------------------+-----------------------------------------------------------------------
Type III  --> 3    |NLdiv   5          Discrete Calculation: Intersecting Volumes IG
-------------------+-----------------------------------------------------------------------
Type IV   --> 4    |Internal Remapping Dual Kriging
                   |Method  5          Method 1: Only Inside the Original Element
                   |                   Method 2: Master-Slave
                   |                   Method 3: Master-Slave Hybrid
                   |                   Method 4: Smart GP
                   |External Remapping
                   |Method  2          Method 1: Brute Force
                   |                   Method 2: Smart GP Master-Slave
                   |                   Method 3: Smart GP
-------------------+-----------------------------------------------------------------------
Master-Slave       |
Expand Kriging     |        2          Variant 1, Variant 2, Variant 3
Symmetry           |        1
Smart GP           |
Minimum GP         |       30
Calculation GP     |       10
-------------------+-----------------------------------------------------------------------
EXTRAS             |
Hydrostatic        |        0         Apply Hydrostatic Pressure Correction, Type III Only
Dependent Vars     |        0         Apply Dependent Variables Correction Yf, TENSEQ
                   |                                             Type I and Type III Only
===================+=======================================================================
PROCEED ROTATION   | on1/off0
(ONLY UFOS)        |        0          Info: New coordinate system axis (Normalized)
                   |                     x         y         z
Vector OX New:     |                  54.0       0.0      32.0
Vector OY New:     |                  53.0      11.0      12.0
Vector OZ New:     |                  60.0      40.0      1.00
===================+=======================================================================
PROCEED TRANSLATION| on1/off0
(ONLY UFOS)        |        0          Info: Translation vector
                   |                     x         y         z
Vector Direction:  |                  54.0       0.0      32.0
===================+=======================================================================
Extra Options      |
Autoname           |        0          Auto append .Trim, required for DD3IMP
AutoAppend Options |        1          Auto append the options to the end of output
===================+=======================================================================
Test Mode          |        0
Inversion Test Mode|        1
Compare Scalar     |        0
===================+=======================================================================

Remarks:
When Splitting:
--> Assure that the inner point is close to the centre ring, but better not over the splitting plan.
--> Assure that the outer point is close to the split plan, but better not over the plan.

--> Remapping of external generic meshes is performed according with the Remapping Method chosen
                        recommended usage of Dual Kriging if the computer resources are available.
--> Starting a simulation on DD3IMP using a generic remmaped mesh is currently unavailable.
```

APPENDIX B: STATUS REPORT

The general appearance of the status report can be found here, due to limitations in the presentation requirements, several spaces between the operation and the date (and time) were removed. The filled data are only placeholders.

```
              ___________________________________________________
             |                                                   |
             |       _____  _____  ____  _______   _             |
             |      |  __ \|  __ \(___ \(__   __) (_)            |
             |      | |  | | |  | | __) ) | |_ __ _ _ __ ___     |
             |      | |  | | |  | |(__ <  | | '__| | '_ ` _ \    |
             |      | |__| | |__| |___) ) | | |  | | | | | | |   |
             |      |_____/|_____/(____/  |_|_|  |_|_| |_| |_|   |
             |     Code to trim, divide and remapp 3D meshes     |
             |                                                   |
             | Version:VXX beta (XX/XX/15) | UFO Version: vXX    |
             |___________________________________________________|
             |  __  __            __                             |
             | |   |__ |\/| |  | |                               |
             | |__ |__ |  | |__| |__  2015                       |
             |                      CENTRO DE ENGENHARIA MECANICA|
             |                         DA UNIVERSIDADE DE COIMBRA|
             |___________________________________________________|
               Wed Jul 01 00:00:00 2015
===============================================================================
  Results of reading Trim2:
-------------------------------------------------------------------------------
    Files to read/write:
-------------------------------------------------------------------------------
        Name_In_Mesh           : InputMeshName
        Name_In_Ufo            : InputUfoName
        Name_Out_Mesh          : OutputMeshName
        Name_Out_Ufo           : OutputUfoName
        Name_Surface           : IGSSurfaceName
-------------------------------------------------------------------------------
    Selected Options Trim 2.0:
        Subprogram             : Trimming
-------------------------------------------------------------------------------
        Source                 : DD3IMP File (Ufo)
-------------------------------------------------------------------------------
        Trim Type              : Nurbs Surface (igs)
        Correction Type        : Type III: Element elimination + adjustment on edge
direction
        Degeneration           : No
-------------------------------------------------------------------------------
        Remapping              : Yes
        Remapping Type         : 4: Dual Kriging
        Dual Kriging Type      : 2: Master-Slave Variant 2
-------------------------------------------------------------------------------
        Correct Dependent Vars : Yes
-------------------------------------------------------------------------------
        Rotation               : No
        Translation            : No
======================================================================================
  Execution Report
======================================================================================
  Started New Operation at                      Wed Jul 01 00:00:00 2015

  Finished Reading/Writing Trim.dat at          Wed Jul 01 00:00:00 2015 Time Elapsed:

  Starting trimming at                          Wed Jul 01 00:00:00 2015 Time Elapsed:
                                                              Time Elapsed Cumulative:
  Finished trimming at                          Wed Jul 01 00:00:00 2015 Time Elapsed:
                                                              Time Elapsed Cumulative:

  Starting remapping at                         Wed Jul 01 00:00:00 2015 Time Elapsed:
                                                              Time Elapsed Cumulative:
  Remapping finished at                         Wed Jul 01 00:00:00 2015 Time Elapsed:
                                                              Time Elapsed Cumulative:

  Whole Process finished at                     Wed Jul 01 00:00:00 2015 Time Elapsed:
                                                              Time Elapsed Cumulative:
```

APPENDIX C: DUAL KRIGING EXAMPLE

The table below contains the coordinates and values from a numerical example:

Gauss Points	*x*	*y*	*z*	f (variable)
GP1	5,74E+00	7,89E-01	-3,76E+00	237,674
GP2	5,42E+00	7,89E-01	-3,85E+00	3004,939
GP3	5,74E+00	2,11E-01	-3,76E+00	-412,444
GP4	5,18E+00	2,11E-01	-3,91E+00	-159,243
GP5	5,66E+00	7,89E-01	-3,49E+00	-1017,314
GP6	5,11E+00	7,89E-01	-3,63E+00	-605,148
GP7	5,66E+00	2,11E-01	-3,49E+00	-1166,087
GP8	5,11E+00	2,11E-01	-3,63E+00	1783,036
Target Point	5,42E+00	7,89E-01	-3,86E+00	

The objective rests on solving the equation (3.1), which requires building the Kriging matrix. As generalized on the Dual Kriging section, **K** will be a symmetric dense square matrix, dimension 12x12. Calculating the Euclidean norm for each combination and assuming a variance function equal to the distance, the resulting $\mathbf{K}_{11}$ matrix, with dimension 8x8 is:

0.0000	0.3229	0.5774	0.8161	0.2884	0.6449	0.6454	0.8656
0.3229	0.0000	0.6615	0.6307	0.4329	0.3843	0.7216	0.6935
0.5774	0.6615	0.0000	0.5768	0.6454	0.8656	0.2884	0.6449
0.8161	0.6307	0.5768	0.0000	0.8656	0.6454	0.6449	0.2884
0.2884	0.4329	0.6454	0.8656	0.0000	0.5768	0.5774	0.8161
0.6449	0.3843	0.8656	0.6454	0.5768	0.0000	0.8161	0.5774
0.6454	0.7216	0.2884	0.6449	0.5774	0.8161	0.0000	0.5768
0.8656	0.6935	0.6449	0.2884	0.8161	0.5774	0.5768	0.0000

The assembled **K** matrix is now shown, the colour code matches the one in 3.1 Mathematical Description.

0	0.3229	0.5774	0.8161	0.2884	0.6449	0.6454	0.8656	1.0000	5.7356	0.7887	-3.7650
0.3229	0	0.6615	0.6307	0.4329	0.3843	0.7216	0.6935	1.0000	5.4228	0.7887	-3.8452
0.5774	0.6615	0	0.5768	0.6454	0.8656	0.2884	0.6449	1.0000	5.7356	0.2113	-3.7650
0.8161	0.6307	0.5768	0	0.8656	0.6454	0.6449	0.2884	1.0000	5.1769	0.2113	-3.9084
0.2884	0.4329	0.6454	0.8656	0	0.5768	0.5774	0.8161	1.0000	5.6639	0.7887	-3.4856
0.6449	0.3843	0.8656	0.6454	0.5768	0	0.8161	0.5774	1.0000	5.1052	0.7887	-3.6290
0.6454	0.7216	0.2884	0.6449	0.5774	0.8161	0	0.5768	1.0000	5.6639	0.2113	-3.4856
0.8656	0.6935	0.6449	0.2884	0.8161	0.5774	0.5768	0	1.0000	5.1052	0.2113	-3.6290
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0	0	0	0
5.7356	5.4228	5.7356	5.1769	5.6639	5.1052	5.6639	5.1052	0	0	0	0
0.7887	0.7887	0.2113	0.2113	0.7887	0.7887	0.2113	0.2113	0	0	0	0
-3.7650	-3.8452	-3.7650	-3.9084	-3.4856	-3.6290	-3.4856	-3.6290	0	0	0	0

The vector of the linear system of equations, vector containing the Euclidean norm between the target point and each interpolation point, and the solution vector are:

$$\mathbf{f} = \left\{ \begin{matrix} 237{,}674 \\ 3004{,}939 \\ -412{,}444 \\ -159{,}243 \\ -1017{,}314 \\ -605{,}148 \\ -1166{,}087 \\ 1783{,}036 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} \right\};\ \mathbf{h} = \left\{ \begin{matrix} 0.326 \\ 0.010 \\ 0.663 \\ 0.630 \\ 0.441 \\ 0.390 \\ 0.727 \\ 0.697 \end{matrix} \right\} \text{ and } \mathbf{u} = \left\{ \begin{matrix} 1789{,}725 \\ -7918{,}050 \\ 174{,}506 \\ 5953{,}820 \\ 1065{,}520 \\ 5062{,}805 \\ 456{,}145 \\ -6584{,}470 \\ -5322{,}761 \\ -1428{,}017 \\ 458{,}136 \\ -3404{,}832 \end{matrix} \right\}.$$

Applying equation (3.1), using the information above, the result for the target point is:

$$F(X_0, Y_0, Z_0) = 2978{,}409 \text{ MPa}$$

APPENDIX D: MATERIAL DATA

The following tables present the material parameters used in trimming examples. The nomenclature in DD3_materX.dat is adopted and the units are MPa.

Table D. 1 Material parameters for the tensile test. Constitutive model: ITYMAT = 4 - Swift Law + Non-Linear Kinematic Hardening; YldCRIT = 1 - Hill48

EM	PR	Y_o	CC	AN	CX	X_{sat}	F	G	H	L	M	N
210 000	0.3	122.200	435.00	0.2190	1.45	116.70	0.2635	0.28329	0.71671	1.5	1.5	1.27947

Table D. 2 Material parameters for the combined shear-tensile test. Constitutive model: ITYMAT = 1 - Swift Law + Linear Kinematic Hardening; YldCRIT = 1 - Hill48

EM	PR	Y_o	CC	AN	F	G	H	L	M	N
206 000	0.3	157.122	565.32	0.2589	0.5	0.5	0.5	1.5	1.5	1.5

Table D. 3 Material parameters for the bending test. Constitutive model: ITYMAT = 1 - Swift Law + Linear Kinematic Hardening; YldCRIT = 1 - Hill48

EM	PR	Y_o	CC	AN	F	G	H	L	M	N
210 000	0.3	123.600	529.5	0.2680	0.251	0.297	0.703	1.5	1.5	1.29

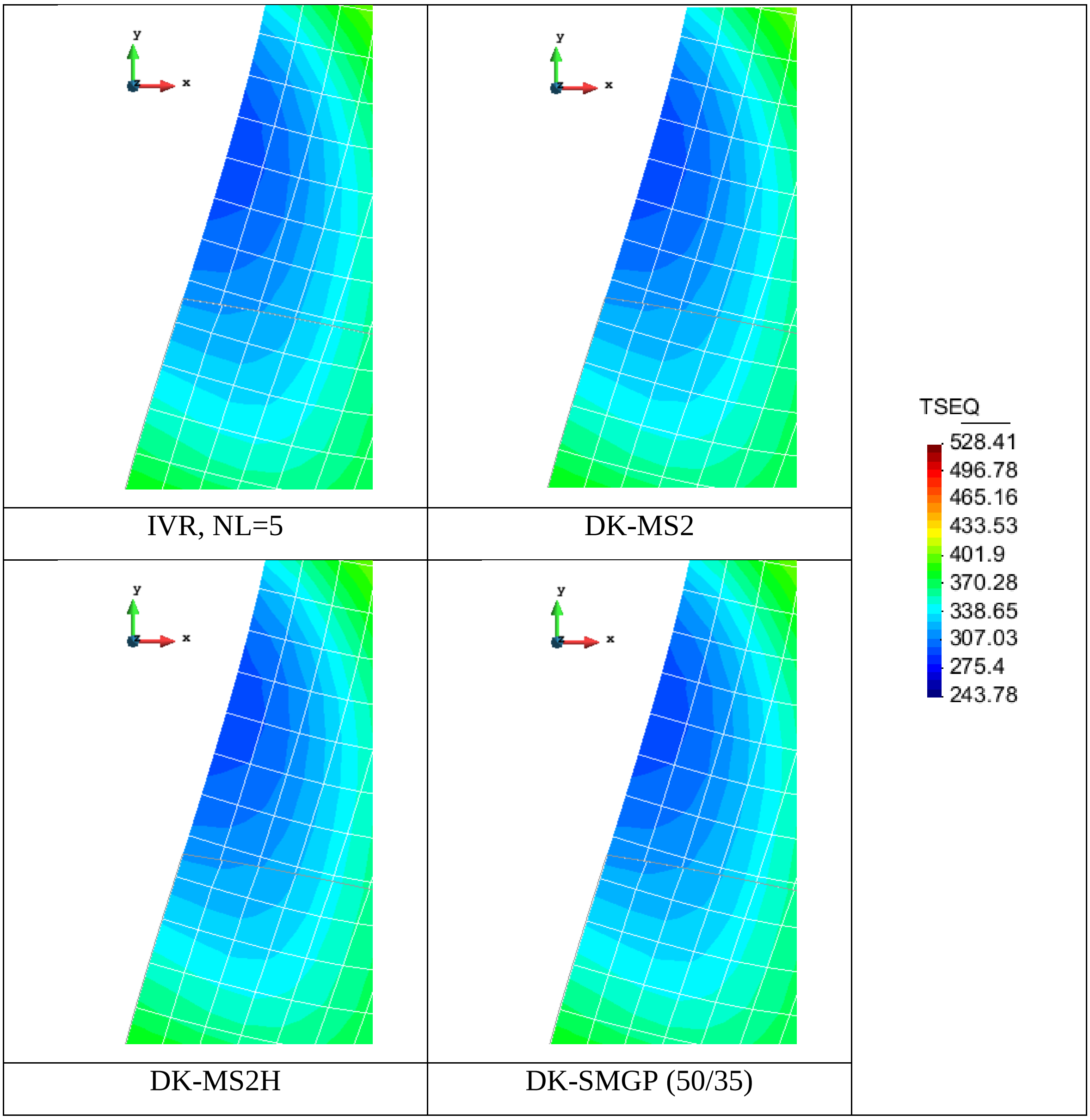
APPENDIX E TENSILE TEST COMPUTATIONAL TIMES

Table E. 1 Tensile Test Computational Time Comparison

	Method II	**% (IVR)**		**Method III**	**% (IVR)**
Trimming (average)	0,108			0,113	
DK-1	0,625	26,5%		0,594	20%
DK-MS1	0,703	29,8%		0,828	28%
DK-MS2	0,672	28,5%		0,828	28%
DK-MS2S	0,719	30,5%		1,063	**37%**
DK-MS3	1,094	**46,4%**		1,078	**37%**
DK-MS1H	0,609	25,8%		0,859	30%
DK-MS1SH	0,703	29,8%		0,750	26%
DK-MS2H	0,609	25,8%		0,765	26%
DK-MS2SH	0,625	26,5%		0,719	25%
DK-MS3H	0,594	25,2%		0,718	25%
DK-SmGP (25/15)	0,984	**41,7%**		1,094	**38%**
IVR NL=5	2,359	100%		2,906	100%

Firstly, it is clear that DK, independently of the algorithm chosen, was able to remap the results faster than IVR. The increase of the complexity of the Master Slave Algorithm, variants and/or symmetry, leads to an increase in computational time. Master-Slave Hybrid is able to provide execution times between DK-1 and Master-Slave.

APPENDIX F: ZOOMED VIEWS COMBINED SHEAR-TENSILE



IVR, NL=5

DK-MS2

DK-MS2H

DK-SMGP (50/35)

Figure F.1 Detail of the remapped zone, near the dark blue area on the left side of the combined shear-tensile specimen