Isogeometric Analysis and Iterative Solvers for Shear Bands

Luc Berger-Vergiat

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

Columbia University
2015

Abstract

Isogeometric Analysis and Iterative Solvers for Shear Bands

Luc Berger-Vergiat

Numerical modeling of shear bands present several challenges, primarily due to strain softening, strong nonlinear multiphysics coupling, and steep solution gradients with fine solution features. In general it is not known a priori where a shear band will form or propagate, thus adaptive refinement is sometimes necessary to increase the resolution near the band.

In this work we first explore the use of isogeometric analysis for shear band problems by constructing and testing several combinations of NURBS elements for a mixed finite element shear band formulation. Owing to the higher order continuity of the NURBS basis, fine solution features such as shear bands can be resolved accurately and efficiently without adaptive refinement. The results are compared to a mixed element formulation with linear functions for displacement and temperature and Pian–Sumihara shape functions for stress. We find that an element based on high order NURBS functions for displacement, temperature and stress, combined with gauss point sampling of the plastic strain leads to attractive results in terms of rate of convergence, accuracy and cpu time. This element is implemented with a $\overline{\mathrm{B}}$-bar strain projection method and is shown to be nearly locking free.

Second we develop robust parallel preconditioners to GMRES in order to solve the Jacobian systems arising at each time step of the problem efficiently. The main idea is to design Schur complements tailored to the specific block structure of the system and that account for the varying stages of shear bands. We develop multipurpose preconditioners that apply to standard irreducible discretizations as well as our recent work on isogeometric discretizations of shear bands. The proposed preconditioners are tested on benchmark examples and compared to standard state of practice solvers such as GMRES/ILU and LU direct solvers.

Nonlinear and linear iterations counts as well as CPU times and computational speedups are reported and it is shown that the proposed preconditioners are robust, efficient and outperform traditional state of the art solvers.

Finally, we extend the preconditioners to further take advantage the physics of the problem. That is most of the deformation and plasticity is localized in a narrow band while out of this domain only small deformations and minor plasticity is observed. Hence, a preconditioner that decomposes the domain and concentrate more effort in the shear band domain while reusing information away from the band may lead to a significantly improved computational performance. To this end, we first propose a schur complement strategy which takes advantage of the gauss point history variables conveniently. Then, a general overlapping domain decomposition procedure is performed, partitioning the domain into so called ’shear band subdomain’ and a ’healthy subdomain’, which is used to precondition the Schur complement system. The shear band subdomain preconditioner is then solved exactly with an LU solver while the healthy subdomain preconditioner is only solved once in the elastic region and reused throughout the simulation. This localization awareness approach is shown to be very efficient and leads to an attractive solver for shear bands.

Contents

List of Tables

List of Figures

Acknowledgements

I want to thank my family for their support throughout my PhD and more generally my education and life. My parents have always pushed me do more and to achieve my best. My older sister Aurélie has been a supportive and competitive sibling, achieving herself to complete her PhD in medicine earlier this year. She was an example to follow as well as a friend to talk to all my life.

I would also like to acknowledge my advisor Pr.Waisman, he has had the partience to let me try and to fail when I could learn from it. He also guided me while always allowing me to voice my doubts and concern as well as pursuing my interest and quench my curiosity. I am grateful for Pr.Spiegelman's teaching; he has been an ever joyful source of knowledge and a challenging person to measure up to. I also have had the great opportunity to work with Ray Tuminaro at Sandia National Laboratory; he is probably the nicest collaborator one can hope for but he is also an expert in the intricate art of making your computer solve problems faster, thanks for introducing me to it! I would like to thank Pr.Fish for chairing my PhD defense committee as well as Pr.Mandli, Pr.Du amd Pr.Sun for attending my defense.

I am thankful to all the faculties in the CEEM department at Columbia University for providing me with a tremendous work environment. Thanks to all my friends and colleagues at CEEM and elsewhere: Adrian, Andrea, Audrey, Aya, Badri, Branislav, Colin, Hannah, Juan, Lauren, Luciana, Madeleine, Mahesh, Mahua, Malini, Matt, Melchor, Miguel, Mostafa, Sabrina, Suparno, Tavishi, Thalia, Uttam, Valia, Wei Wei, Yongxiang; you made my PhD an ever eventful and enjoyable adventure.

Chapter 1

Introduction

1.1 Literature review

Shear bands are highly localized zones of intense plastic deformations [105] which are formed in metals under high strain rate loading and can be classified as a ductile fracture mechanism. This localization phenomenon occurs due to significant strain hardening which leads to thermal softening that results in locally reduced stress bearing capacity of the material and eventually lead to its failure [45, 68]. Modeling shear bands accurately is very challenging due to the complicated multiphysics and difficult numerics. This is a highly nonlinear, coupled thermo-mechanical problem with localized crack-like domains that require specialized discretizations and significant computational power to resolve ~~shear bands~~ accurately in space and time.

Experimentally derived material models for these loading regimes describe plastic flow as being dependent on temperature, strain rate, and a hardening parameter [105]. While several models are available, all are similar in that increasing temperature (due to plastic work) has a softening effect, causing plastic flow to occur more readily, while increases in strain rate and the hardening parameter have a hardening effect. Following the experimental work of [72], shear bands develop in three stages, as depicted in Figure 1.1 . In Stage 1, before localization, a homogeneous distribution of plastic strain exists. Stage 2 begins when the thermal softening effect dominates the strain and strain rate hardening effects, resulting in strain softening, and thus strain localization is initiated. Stage 3 is marked by severe localization and rapid softening, a phenomena termed stress collapse, which indicates a sudden and large drop in the material's load bearing capability [106].

Multiple numerical schemes have been developed to analyze shear bands. The most common approach is to first simplify the governing equations in Eq (1.12) by assuming adiabatic conditions in which $\kappa = 0$. In other words, since shear band propagation is a fast process while thermal diffusion is a slow one, the later has commonly been neglected in computations [3, 66, 79]. While such an approach simplifies the numerics significantly, it also results in a lack of physical length scale in the system. Hence the system is ill posed and the convergence to a shear band width has been shown to be sensitive to the mesh size [17, 107, 108].

In order to remedy this problem, various approaches such as viscoplasticity [70, 103] and gradient plasticity [2] have also been proposed with the intention to regularize the shear band problem. Other methods add regularized damage to the constitutive model to cure the mesh sensitivity and introduce additional softening in the problem [29, 48].

An alternative to low order FE model for shear band is the meshfree family of methods, which has been studied by [65,67]. A meshfree approach can easily handle large deformations and was shown in [65] to suppress mesh alignment sensitivity. Another approach is the multiscale technique called the multiresolution method, proposed by McVeigh et. al. [69, 76] and has also been used to model shear bands [77,96,97]. The multiresolution method can be understood as a local refinement technique that may employ different governing equations in different local regions. However, an embedded length scale parameter which can be related to size, morphology of microvoids and the width of propagating shear bands must also be assumed. The technique has been used for adiabatic shear bands and shear bands assisted by micro-voids. Since in practice, it is generally not known a priori where a shear

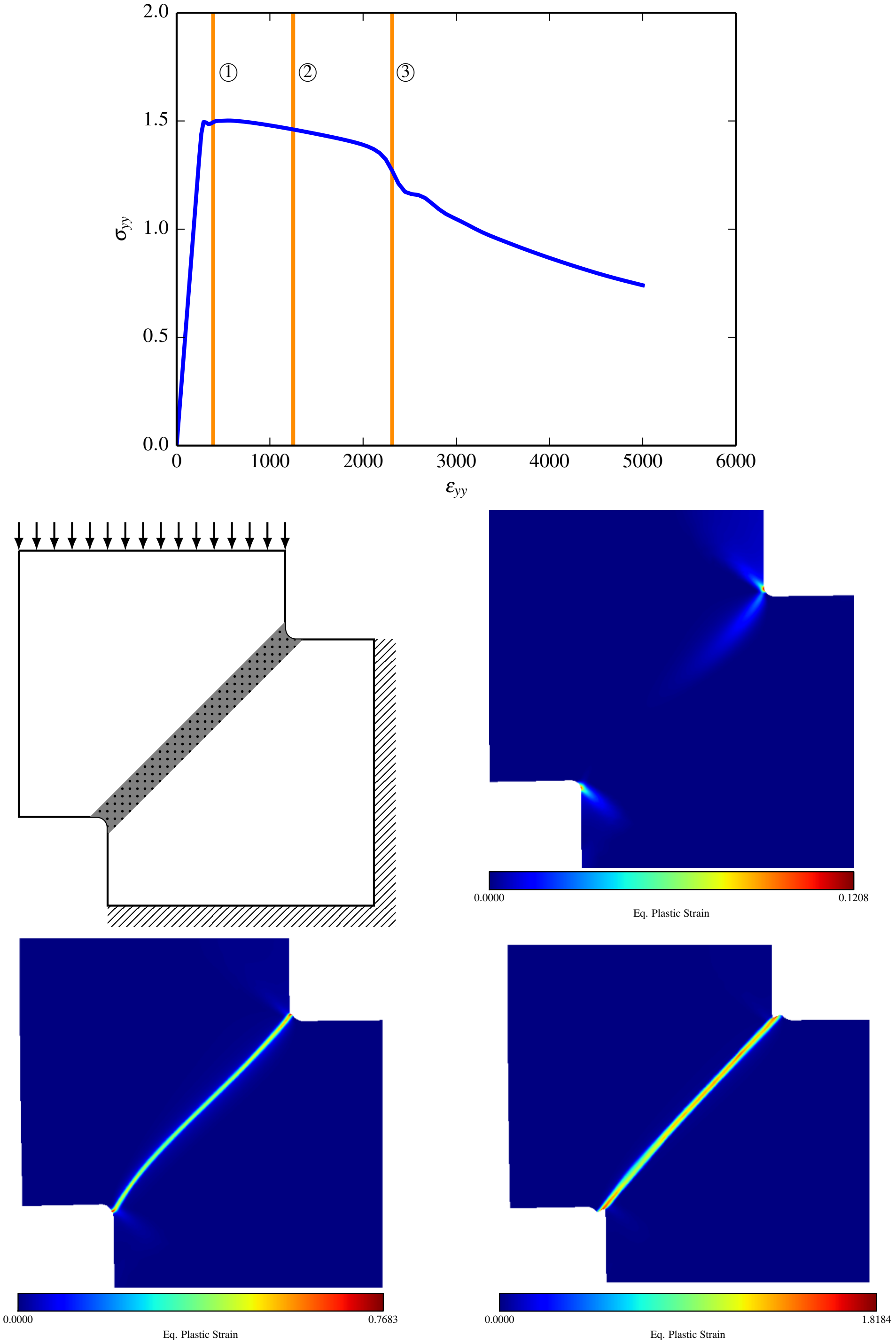


Figure 1.1: In the upper figure a normalized stress-strain curve representative of a shear band developing in a plate under compression, middle left plot, is shown. The line denoted ① shows the elstic-plastic limit (stage 1) corresponding to the middle left plot, the line denoted ② marks the region dominated by the thermal softening (stage 2) seen at the bottom left plot, finally the line denoted ③ shows the stress collapse (stage 3) seen at the bottom right plot.

band will initiate or propagate, adaptive local refinement techniques are likely unavoidable for large problems. However, for the same reason, it is also desirable to develop methods where relatively coarse meshes are capable of furnishing good levels of accuracy. Isogemetric analysis is particularly well suited for this purpose, since it employs higher order shape functions capable of efficiently resolving the small scale solution features associated with shear bands.

Previous work by McAuliffe and Waisman [73] made use of a mixed finite element formulations to simultaneously solve for the four equations of system 1.12 with the displacements, temperatures, stresses, and equivalent plastic strains each independently interpolated degrees of freedom. The Implicit Nonlinear Consistent (INC), or monolithic solver combined with thermal diffusion achieved results that were insensitive to mesh size, and capable of achieving significantly higher accuracy than a split explicit scheme for a given amount of computational effort. Mesh alignment sensitivity of the formulation was also tested in [73]. Mesh alignment sensitivity is the tendency of the mesh to be more compliant along element edges during shear band propagation (first studied by [101]). It was found that the INC solver improves, but does not completely eliminate the alignment sensitivity. The shape functions employed in that work were low order, being bilinear for velocity and temperature, the Pian-Sumihara interpolations for stress [81] and gauss point sampling was done for equivalent plastic field.

Isogeometric Analysis [32,55] (IGA) was originally developed with the intent of streamlining the simulation and design process. This process typically starts with discrete geometrical description of the part or system to be analyzed with a Computer Aided Design (CAD) package, which employ the NURBS basis functions [12, 44] or a T-splines basis function [14, 16]. Once the geometry is fully described it is passed on to the simulation group, which in turn rediscretize the geometry according to the needs of the simulation technique chosen for analysis. IGA eliminates this rediscretization step from the simulation work flow by using the same basis functions for both the CAD geometrical description and analysis phases. Another added benefit of this approach is that now the geometry used during the analysis is exact.

In this thesis we explore the use of Isogeometric Analysis for shear band problems by constructing and testing several combinations of NURBS elements for a mixed finite element shear band formulation. Owing to the higher order continuity of the NURBS basis, fine solution features can be resolved accurately and efficiently without adaptive refinement and with the implementation of a $\overline{\text{B}}$-bar strain projection method which is shown to lead to nearly locking free elements. The results are compared to a mixed element formulation with linear functions for displacement and temperature and Pian-Sumihara shape functions for stress.

Furthermore, robust parallel preconditioners to GMRES are developed in order to solve the resulting Jacobian systems efficiently. The main idea is to design schur complements tailored to the specific block structure of the system and that account for the varying stages of shear bands. We develop multipurpose preconditioners that apply to standard irreducible discretizations as well as our recent work on isogeometric discretizations of shear bands. Serial and parallel performance is studied and reported.

Finally a domain decomposition method coupled with a Schur complement is proposed to create a targeted preconditioner that concentrate computational effort on the shear band region while reusing information away from the shear band domain. This allow for an efficient algorithm that is aware of the physics and leads to fast converging solvers.

1.2 Problem statement and numerical approach

Shear bands can be modeled by a system of partial differential equations (PDEs) describing conservation of momentum, conservation of energy, elastic and inelastic constitutive relations [105]. Under small deformation assumption, the strain tensor may be written as

$$\underline{\underline{\epsilon}} = \frac{1}{2}\left[\underline{\nabla}\ \underline{u} + (\underline{\nabla}\ \underline{u})^t\right] = \underline{\nabla}^s \underline{u} \tag{1.1}$$

where $\underline{\nabla}$ and $\underline{\nabla}^s$ are the gradient and the symmetric part of the gradient operator, respectively, and $\underline{u}$ is the displacement vector. In our model both the elastic and plastic constitutive relations are expressed in rate form so we introduce the rate of deformation, or strain rate, tensor as

$$\underline{\underline{D}} = \underline{\underline{\dot{\epsilon}}} = \underline{\nabla}^s \underline{v} = \frac{1}{2}\left(\underline{\underline{L}} + \underline{\underline{L}}^t\right) \tag{1.2}$$

where $\underline{v}$ is the velocity vector and $\underline{\underline{L}}$ is the velocity gradient tensor. The constitutive laws are developed assuming that the rate of deformation can be additively decomposed in three components: the elastic rate of deformation $\underline{\underline{D}}^e$, the thermal rate of deformation $\underline{\underline{D}}^T$ and the plastic rate of deformation $\underline{\underline{D}}^p$, i.e.

$$\underline{\underline{D}} = \underline{\underline{D}}^e + \underline{\underline{D}}^T + \underline{\underline{D}}^p. \tag{1.3}$$

This leads us to the following elastic constitutive relation in rate form

$$\underline{\underline{\dot{\sigma}}} = \underline{\underline{\underline{\underline{C}}}}^{elas} : \left(\underline{\underline{D}} - \underline{\underline{D}}^T - \underline{\underline{D}}^p\right) \tag{1.4}$$

where $\underline{\underline{\underline{\underline{C}}}}^{elas}$ is the elasticity tensor and $\underline{\underline{\sigma}}$ is the stress tensor. The rate of thermal deformation is defined using the thermal expansion coefficient α

$$\underline{\underline{D}}^T = \alpha \dot{T} \underline{\underline{I}} \tag{1.5}$$

where T is the temperature and $\underline{\underline{I}}$ is the second order identity tensor.

Finally the rate of plastic deformation is defined using $J2$ plasticity as follows

$$\underline{\underline{D}}^p = \frac{3}{2}\frac{g(\bar{\sigma}, T, \bar{\gamma}_p)}{\bar{\sigma}} \underline{\underline{S}} \tag{1.6}$$

where g is a modified Litonski plastic flow law, which has been previously used by [67,107,108] and is defined as

$$g(\bar{\sigma}, T, \bar{\gamma}_p) = \dot{\bar{\gamma}}_{p,ref} \left(\frac{\bar{\sigma}}{\sigma_{ref} \left[1 + \bar{\gamma}_p/\bar{\gamma}_{p,ref}\right]^N \left[1 + \delta - \delta \exp\left(\frac{T - T_{ref}}{k}\right)\right]} \right)^m . \tag{1.7}$$

The material constants obtained from experiments in the flow law above are $\dot{\bar{\gamma}}_{p,ref}$, $\bar{\gamma}_{p,ref}$, σ_{ref} and T_{ref} are the reference plastic strain rate, yield strain, yield stress and temperature. m, N, δ and k are the rate sensitivity exponent, the strain hardening exponent and the first and second thermal softening parameters.

The deviatoric stress is defined as

$$\underline{\underline{S}} = \underline{\underline{\sigma}} - \frac{tr(\underline{\underline{\sigma}})}{3} \underline{\underline{I}}. \tag{1.8}$$

$tr()$ being the trace operator.

Remark 1: Note that (1.6) implies that the direction of the plastic strain rate is the same as the direction of the deviatoric stress $\underline{\underline{S}}$. ■

The effective, or equivalent, stress is given by

$$\bar{\sigma} = \sqrt{\frac{3}{2} \underline{\underline{S}} : \underline{\underline{S}}} \tag{1.9}$$

the equivalent plastic strain (EQPS) rate is then defined as the power conjugate of the equivalent stess which is expressed as

$$\bar{\sigma} \dot{\bar{\gamma}}_p = \underline{\underline{\sigma}} : \underline{\underline{D}}^p. \tag{1.10}$$

That leads to the following derivation of the EQPS rate

$$\begin{aligned}\dot{\bar{\gamma}}_p &= \frac{\underline{\underline{\sigma}} : \underline{\underline{D}}^p}{\bar{\sigma}} \\ &= \frac{3}{2}\frac{g(\bar{\sigma}, T, \bar{\gamma}_p)}{\bar{\sigma}^2}\underline{\underline{\sigma}} : \underline{\underline{S}} \\ &= \frac{\frac{3}{2}\underline{\underline{S}} : \underline{\underline{S}}}{\bar{\sigma}^2} g(\bar{\sigma}, T, \bar{\gamma}_p) \\ &= g(\bar{\sigma}, T, \bar{\gamma}_p).\end{aligned} \tag{1.11}$$

Assuming that no body forces are applied, the governing system of PDEs can finally be expressed as follows

$$\begin{cases} \rho\ddot{\underline{u}} &= \underline{\nabla} \cdot \underline{\underline{\sigma}} \\ \rho c\dot{T} &= \underline{\nabla} \cdot [\kappa \underline{\nabla}\ T] + \chi\bar{\sigma}\dot{\bar{\gamma}}_p \\ \dot{\underline{\underline{\sigma}}} &= \underline{\underline{\underline{C}}}^{elas} : \left(\underline{\nabla}^s \dot{\underline{u}} - \alpha\dot{T}\underline{\underline{I}} - \frac{3}{2}\frac{g(\bar{\sigma}, T, \bar{\gamma}_p)}{\bar{\sigma}}\underline{\underline{S}}\right) \\ \dot{\bar{\gamma}}_p &= g(\bar{\sigma}, T, \bar{\gamma}_p) \end{cases} \tag{1.12}$$

The right hand side of the second equation in system (1.12) assumes that part of the plastic work produces heat while the rest is stored by defects in the metal's lattice, also known as the cold work of plasticity [85]. The ratio of the rate of heat produced by plasticity $\dot{Q}^p = \chi\bar{\sigma}\dot{\bar{\gamma}}_p$ to the rate of the plastic work $\dot{W}^p$ is know as the Taylor-Quinney coefficient [84] and is denoted χ.

Remark 2: Note that here we assume that the thermoelastic heating rate $\dot{Q}^e = -\alpha ET\underline{\underline{D}}^e$ is negligible compared to the thermoplastic heating rate $\dot{Q}^p$. ■

These equations are verified in the solid domain Ω with the following boundary conditions

$$\begin{aligned} \underline{u}(\underline{x},t) &= \underline{\bar{u}}(\underline{x},t) && \text{on } \Gamma_D^u, \\ \underline{n}.\underline{\underline{\sigma}}(\underline{x},t) &= \underline{\bar{t}}(\underline{x},t) && \text{on } \Gamma_N^u, \\ T(\underline{x},t) &= \bar{T}(\underline{x},t) && \text{on } \Gamma_D^T, \\ \underline{n}.\underline{q}(\underline{x},t) &= \bar{q}(\underline{x},t) && \text{on } \Gamma_N^T. \end{aligned} \tag{1.13}$$

Where $\underline{\bar{t}}$ is the prescribed normal traction and $\bar{q}$ is the prescribed normal thermal flux. (Γ_D, Γ_N) is a partition of $\partial\Omega$ the boundary of Ω with the classic notation Γ_D as the subset of $\partial\omega$ where the Dirichlet or eseential boundary conditions are applied and Γ_N the subset of $\partial\Omega$ where the Neumann or natural boundary conditions are applied. The superscripts u and T are used to differentiate the boundary conditions applied to the mechanical equilibrium and the thermal equilibrium respectively. These notation are summarized schematically in Figure 1.2

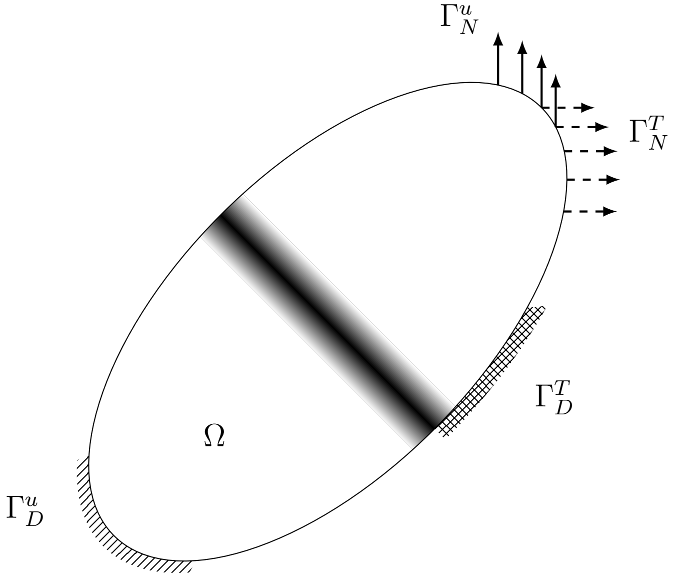


Figure 1.2: Schematic representation of the thermomechanical problem with its boundary conditions.

The weak form of the equations in system (1.12) is obtained by multiplying each equation by its respective test function: w_u, w_T, w_σ and $w_{\bar{\gamma}_p}$ and by integrating over the domain Ω.

This leads to the following residual vector

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_u \\ \mathbf{R}_T \\ \mathbf{R}_\sigma \\ \mathbf{R}_{\bar{\gamma}_p} \end{bmatrix} = \begin{bmatrix} \displaystyle\int_\Omega w_u \rho \underline{\ddot{u}} + \underline{\nabla}\ w_u \cdot \underline{\underline{\sigma}} d\Omega - \int_{\Gamma_N^u} \underline{\bar{t}} d\Gamma_N^u \\ \displaystyle\int_\Omega w_T \left[\rho c \dot{T} - \chi \bar{\sigma} g(\bar{\sigma}, T, \bar{\gamma}_p)\right] + \underline{\nabla}\ w_T \cdot \kappa \underline{\nabla}\ T d\Omega - \int_{\Gamma_N^T} \kappa \bar{q} d\Gamma_N^T \\ \displaystyle\int_\Omega w_\sigma \left[\underline{\underline{\dot{\sigma}}} - \underline{\underline{\underline{\underline{C}}}}^{elas} : \left(\underline{\nabla}^s \underline{\dot{u}} - \alpha \dot{T} \underline{\underline{I}} - \frac{3}{2} \frac{g(\bar{\sigma}, T, \bar{\gamma}_p)}{\bar{\sigma}} \underline{\underline{S}}\right)\right] d\Omega \\ \displaystyle\int_\Omega w_{\bar{\gamma}_p} \left[\dot{\bar{\gamma}}_p - g(\bar{\sigma}, T, \bar{\gamma}_p)\right] d\Omega \end{bmatrix}. \tag{1.14}$$

Remark 3: Note that the divergence theorem was applied were appropriate to reduce the order of the derivatives in the equations. ■

The weak form is integrated in time using the Newmark method [99, 110], assuming an implicit scheme. A brief derivation of the coefficients needed for Newmark integration is provided in Appendix A. Newton's method is then used to solve the resulting nonlinear system of equations at a given time step. To this end, we linearize the residual by computing the variation of $\mathbf{R}$ with respect to the unknown variables $\underline{u}$, T, $\underline{\underline{\sigma}}$ and $\bar{\gamma}_p$ as follow

$$\mathbf{J}\delta\mathbf{u} = \mathbf{R}. \tag{1.15}$$

where $\mathbf{J}$ is the jacobian (or tangent stiffness) of the problem and $\delta\mathbf{u}$ is the incremental solution computed at each Newton iteration.

The jacobian is derived using variational calculus and more precisely a Gâteaux derivative [71, 95]

$$\mathbf{J}\delta\mathbf{u} = \frac{d}{d\epsilon}\mathbf{R}(\mathbf{u} + \epsilon\delta\mathbf{u})\bigg|_{\epsilon=0}. \tag{1.16}$$

We start with the derivatives of $\mathbf{R}_u$ in the directions of $\delta\underline{u}$, δT, $\delta\underline{\underline{\sigma}}$ and $\delta\bar{\gamma}_p$.

$$\begin{aligned}
\mathbf{J}_{uu}\delta\underline{u} &= \left.\frac{d\mathbf{R}_u(\underline{u}+\epsilon\delta\underline{u})}{d\epsilon}\right|_{\epsilon=0} \\
&= \frac{d}{d\epsilon}\left.\left(\int_\Omega w_u\rho\frac{\partial^2}{\partial t^2}(\underline{u}+\epsilon\delta\underline{u}) + \underline{\nabla}\ w_u\cdot\underline{\underline{\sigma}}d\Omega - \int_{\Gamma_t} w_u\underline{\bar{t}}d\Gamma_t\right)\right|_{\epsilon=0} \\
&= \int_\Omega w_u\rho\frac{\partial^2\delta\underline{u}}{\partial t^2}d\Omega.
\end{aligned} \tag{1.17}$$

$$\mathbf{J}_{uT}\delta T = \left.\frac{d\mathbf{R}_u(T+\epsilon\delta T)}{d\epsilon}\right|_{\epsilon=0} = 0. \tag{1.18}$$

$$\begin{aligned}
\mathbf{J}_{u\sigma}\delta\underline{\underline{\sigma}} &= \left.\frac{d\mathbf{R}_u(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\right|_{\epsilon=0} \\
&= \frac{d}{d\epsilon}\left.\left(\int_\Omega w_u\rho\frac{\partial^2\underline{u}}{\partial t^2} + \underline{\nabla}\ w_u\cdot(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})d\Omega - \int_{\Gamma_t} w_u\underline{\bar{t}}d\Gamma_t\right)\right|_{\epsilon=0} \\
&= \int_\Omega \underline{\nabla}\ w_u\cdot\delta\underline{\underline{\sigma}}d\Omega.
\end{aligned} \tag{1.19}$$

$$\mathbf{J}_{u\bar{\gamma}_p}\delta\bar{\gamma}_p = \left.\frac{d\mathbf{R}_u(\bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)}{d\epsilon}\right|_{\epsilon=0} = 0. \tag{1.20}$$

We now move on to the derivatives of $\mathbf{R}_T$ in the directions of $\delta\underline{u}$, δT, $\delta\underline{\underline{\sigma}}$ and $\delta\bar{\gamma}_p$.

$$\mathbf{J}_{Tu}\delta\underline{u} = \left.\frac{d\mathbf{R}_T(\underline{u}+\epsilon\delta\underline{u})}{\epsilon}\right|_{\epsilon=0} = 0. \tag{1.21}$$

$$\begin{aligned}
\mathbf{J}_{TT}\delta T &= \left.\frac{d\mathbf{R}_T(T+\epsilon\delta T)}{\epsilon}\right|_{\epsilon=0}, \\
&= \frac{d}{d\epsilon}\left(\int_\Omega w_T\left[\rho c\frac{\partial T+\epsilon\delta T}{\partial t} - \chi\bar{\sigma}g(\bar{\sigma},T+\epsilon\delta T,\bar{\gamma}_p)\right] + \underline{\nabla}\ w_T\cdot\kappa\underline{\nabla}\ T+\epsilon\delta T d\Omega\right. \\
&\quad \left.\left. - \int_{\Gamma_q} w_T\kappa\bar{q}d\Gamma_q\right)\right|_{\epsilon=0}, \\
&= \int_\Omega w_T\left(\rho c\frac{\partial\delta T}{\partial t} - \chi\bar{\sigma}\left.\frac{d}{d\epsilon}g(\bar{\sigma},T+\delta T,\bar{\gamma}_p)\right|_{\epsilon=0}\right) + \underline{\nabla}\ w_T\cdot\kappa\underline{\nabla}\ \delta T d\Omega.
\end{aligned} \tag{1.22}$$

$$
\begin{aligned}
\mathbf{J}_{T\sigma}\delta\underline{\underline{\sigma}} &= \left.\frac{d\mathbf{R}_T(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\right|_{d\epsilon=0}, \\
&= \frac{d}{d\epsilon}\left(\int_\Omega w_T\left[\rho c\frac{\partial T}{\partial t}-\chi\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})g(\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}}),T,\bar{\gamma}_p)\right]+\underline{\nabla}\ w_T\cdot\kappa\underline{\nabla}\ Td\Omega\right. \\
&\quad \left.\left.-\int_{\Gamma_q} w_T\kappa\bar{q}d\Gamma_q\right)\right|_{\epsilon=0}, \\
&= \int_\Omega w_T\left(-\chi\left.\frac{d\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\right|_{\epsilon=0}g(\bar{\sigma},T,\bar{\gamma}_p)-\chi\bar{\sigma}\left.\frac{d}{d\epsilon}g\left(\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}}),T,\bar{\gamma}_p\right)\right|_{\epsilon=0}\right)d\Omega.
\end{aligned}
\tag{1.23}
$$

$$
\begin{aligned}
\mathbf{J}_{T\bar{\gamma}_p}\delta\bar{\gamma}_p &= \left.\frac{d\mathbf{R}_T(\bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)}{d\epsilon}\right|_{d\epsilon=0}, \\
&= \frac{d}{d\epsilon}\left.\left(\int_\Omega w_T\left[\rho c\frac{\partial T}{\partial t}-\chi\bar{\sigma}g(\bar{\sigma},T,\bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)\right]+\underline{\nabla}\ w_T\cdot\kappa\underline{\nabla}\ Td\Omega-\int_{\Gamma_q} w_T\kappa\bar{q}d\Gamma_q\right)\right|_{\epsilon=0}, \\
&= \int_\Omega -w_T\chi\bar{\sigma}\left.\frac{d}{d\epsilon}g(\bar{\sigma},T,\bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)\right|_{\epsilon=0}d\Omega.
\end{aligned}
\tag{1.24}
$$

Last but one comes the derivation of $\mathbf{R}_\sigma$ in the directions of $\delta\underline{u}$, δT, $\delta\underline{\underline{\sigma}}$ and $\delta\bar{\gamma}_p$.

$$
\begin{aligned}
\mathbf{J}_{\sigma u}\delta\underline{u} &= \frac{d\mathbf{R}_\sigma(\underline{u}+\epsilon\delta\underline{u})}{d\epsilon} \\
&= \frac{d}{d\epsilon}\left.\left(\int_\Omega w_\sigma\left[\dot{\underline{\underline{\sigma}}}-\underline{\underline{\underline{C}}}^{elas}:\left(\nabla^s(\underline{u}+\epsilon\delta\underline{u})-\frac{3}{2}\frac{g(\bar{\sigma},T,\bar{\gamma}_p)}{\bar{\sigma}}\underline{\underline{S}}-\alpha\frac{\partial T}{\partial t}\underline{\underline{I}}\right]\right)\right)\right|_{\epsilon=0} \\
&= \int_\Omega -w_\sigma\underline{\underline{\underline{C}}}^{elas}:\nabla^s\delta\underline{u}\ d\Omega.
\end{aligned}
\tag{1.25}
$$

$$
\begin{aligned}
\mathbf{J}_{\sigma T}\delta T &= \frac{d\mathbf{R}_\sigma(T+\epsilon\delta T)}{d\epsilon} \\
&= \frac{d}{d\epsilon}\left.\left(\int_\Omega w_\sigma\left[\dot{\underline{\underline{\sigma}}}-\underline{\underline{\underline{C}}}^{elas}\nabla^s\underline{u}-\frac{3}{2}\frac{g(\bar{\sigma},T+\epsilon\delta T,\bar{\gamma}_p)}{\bar{\sigma}}\underline{\underline{S}}-\alpha\frac{\partial T+\epsilon\delta T}{\partial t}\underline{\underline{I}}\right]\right)\right|_{\epsilon=0} \\
&= \int_\Omega w_\sigma\underline{\underline{\underline{C}}}^{elas}:\left(\frac{3}{2\bar{\sigma}}\left.\frac{d}{d\epsilon}g(\bar{\sigma},T+\epsilon\delta T,\bar{\gamma}_p)\right|_{\epsilon=0}\underline{\underline{S}}+\alpha\frac{\partial\delta T}{\partial t}\underline{\underline{I}}\right)d\Omega.
\end{aligned}
\tag{1.26}
$$

$$\begin{aligned}
\mathbf{J}_{\sigma\sigma}\delta\underline{\underline{\sigma}} &= \frac{d\mathbf{R}_\sigma(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\\
&= \frac{d}{d\epsilon}\int_\Omega w_\sigma \left[\dot{\underline{\underline{\sigma}}} + \epsilon\delta\dot{\underline{\underline{\sigma}}} - \underline{\underline{\underline{C}}}^{elas} : \left(\nabla^s \underline{u} - \frac{3}{2}\frac{g(\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}}),T,\bar{\gamma}_p)}{\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}\underline{\underline{S}}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})\right.\right.\\
&\qquad\left.\left. -\alpha\frac{\partial T}{\partial t}\underline{\underline{I}}\right)\right] d\Omega\bigg|_{\epsilon=0}\\
&= \int_\Omega w_\sigma \left[\delta\dot{\underline{\underline{\sigma}}} + \underline{\underline{\underline{C}}}^{elas} : \left(\frac{3}{2\bar{\sigma}}\,\frac{d}{d\epsilon}g(\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}}),T,\bar{\gamma}_p)\bigg|_{\epsilon=0}\underline{\underline{S}}\right.\right.\\
&\qquad\left.\left. -\frac{3}{2}\,\frac{d\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\bigg|_{\epsilon=0}\frac{1}{\bar{\sigma}^2}g(\bar{\sigma},T,\bar{\gamma}_p)\underline{\underline{S}} + \frac{3}{2\bar{\sigma}}g(\bar{\sigma},T,\bar{\gamma}_p)\,\frac{d\underline{\underline{S}}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\bigg|_{\epsilon=0}\right)\right] d\Omega
\end{aligned} \tag{1.27}$$

$$\begin{aligned}
\mathbf{J}_{\sigma\bar{\gamma}_p}\delta\bar{\gamma}_p &= \frac{d\mathbf{R}_\sigma(\bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)}{d\epsilon}\\
&= \frac{d}{d\epsilon}\int_\Omega w_\sigma \left[\dot{\underline{\underline{\sigma}}} - \underline{\underline{\underline{C}}}^{elas} : \left(\nabla^s \underline{u} - \frac{3}{2}\frac{g(\bar{\sigma},T,\bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)}{\bar{\sigma}}\underline{\underline{S}} - \alpha\frac{\partial T}{\partial t}\underline{\underline{I}}\right)\right] d\Omega\bigg|_{\epsilon=0}\\
&= \int_\Omega w_\sigma \underline{\underline{\underline{C}}}^{elas} : \left(\frac{3}{2\bar{\sigma}}\,\frac{d}{d\epsilon}g(\bar{\sigma},T,\bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)\bigg|_{\epsilon=0}\underline{\underline{S}}\right) d\Omega.
\end{aligned} \tag{1.28}$$

And finally we derive $\mathbf{R}_{\bar{\gamma}_p}$ in the directions of $\delta\underline{u}$, δT, $\delta\underline{\underline{\sigma}}$ and $\delta\bar{\gamma}_p$.

$$\mathbf{J}_{\bar{\gamma}_p u}\delta\underline{u} = \frac{d\mathbf{R}_{\bar{\gamma}_p}(\underline{u}+\epsilon\delta\underline{u})}{d\epsilon} = 0. \tag{1.29}$$

$$\begin{aligned}
\mathbf{J}_{\bar{\gamma}_p T}\delta T &= \frac{d\mathbf{R}_{\bar{\gamma}_p}(T+\epsilon\delta T)}{d\epsilon}\\
&= \frac{d}{d\epsilon}\left(\int_\Omega w_{\bar{\gamma}_p}\left[\dot{\bar{\gamma}}_p - g(\bar{\sigma},T+\epsilon\delta T,\bar{\gamma}_p)\right] d\Omega\right)\bigg|_{\epsilon=0}\\
&= \int_\Omega -w_{\bar{\gamma}_p}\,\frac{d}{d\epsilon}g(\bar{\sigma},T+\epsilon\delta T,\bar{\gamma}_p)\bigg|_{\epsilon=0} d\Omega
\end{aligned} \tag{1.30}$$

$$\begin{aligned}
\mathbf{J}_{\bar{\gamma}_p\sigma}\delta\underline{\underline{\sigma}} &= \frac{d\mathbf{R}_{\bar{\gamma}_p}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon} \\
&= \frac{d}{d\epsilon}\left(\int_\Omega w_{\bar{\gamma}_p}\left[\dot{\bar{\gamma}}_p - g(\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}}),T,\bar{\gamma}_p)\right]d\Omega\right)\Bigg|_{\epsilon=0} \\
&= \int_\Omega -w_{\bar{\gamma}_p}\frac{d}{d\epsilon}g(\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}}),T,\bar{\gamma}_p)\Big|_{\epsilon=0}d\Omega
\end{aligned} \tag{1.31}$$

$$\begin{aligned}
\mathbf{J}_{\bar{\gamma}_p\bar{\gamma}_p}\delta\bar{\gamma}_p &= \frac{d\mathbf{R}_{\bar{\gamma}_p}(\bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)}{d\epsilon} \\
&= \frac{d}{d\epsilon}\left(\int_\Omega w_{\bar{\gamma}_p}\left[\dot{\bar{\gamma}}_p+\epsilon\delta\dot{\bar{\gamma}}_p - g(\bar{\sigma},T,\bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)\right]d\Omega\right)\Bigg|_{\epsilon=0} \\
&= \int_\Omega w_{\bar{\gamma}_p}\left[\delta\dot{\bar{\gamma}}_p - \frac{d}{d\epsilon}g(\bar{\sigma},T,\bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)\Big|_{\epsilon=0}\right]d\Omega
\end{aligned} \tag{1.32}$$

with

$$\begin{aligned}
\frac{d\underline{\underline{S}}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\Bigg|_{\epsilon=0} &= \frac{d}{d\epsilon}\left(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}} - \frac{\mathrm{tr}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{3}\underline{\underline{I}}\right)\Bigg|_{\epsilon=0} \\
&= \left(\underline{\underline{\underline{I}}} - \frac{1}{3}\underline{\underline{I}}\otimes\underline{\underline{I}}\right):\delta\underline{\underline{\sigma}} \\
&= \underline{\underline{\underline{P}}}:\delta\underline{\underline{\sigma}}
\end{aligned} \tag{1.33}$$

where $\underline{\underline{\underline{I}}}$ and $\underline{\underline{\underline{P}}}$ are the fourth order identity tensor and the deviatoric projection tensor respectively.

For the above derivations no assumptions where made for the choice of equivalent stress and plastic flow law making the above general enough for any metal model.

With our definition for the equivalent stress 1.9 we obtain the following

$$\frac{d\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\Bigg|_{\epsilon=0} = \frac{d\bar{\sigma}}{d\underline{\underline{S}}}:\frac{d\underline{\underline{S}}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\Bigg|_{\epsilon=0} \tag{1.34}$$

using the chain rule. We recognize the deviatoric projection tensor in the second term of the chain rule and thus only need to carry on the calculation of the first term of the chain rule.

$$\frac{d\bar{\sigma}}{d\underline{\underline{S}}} = \frac{d}{d\underline{\underline{S}}}\left(\frac{3}{2}\underline{\underline{S}}:\underline{\underline{S}}\right)^{1/2} = \frac{1}{2}\left(\frac{3}{2}\underline{\underline{S}}:\underline{\underline{S}}\right)^{-1/2}\frac{d}{d\underline{\underline{S}}}\left(\frac{3}{2}\underline{\underline{S}}:\underline{\underline{S}}\right) = \frac{3}{2\bar{\sigma}}\frac{1}{2}\frac{d}{d\underline{\underline{S}}}\left(\underline{\underline{S}}:\underline{\underline{S}}\right) = \frac{3}{2\bar{\sigma}}\underline{\underline{S}} \tag{1.35}$$

which yield the final expression

$$\left.\frac{d\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\right|_{\epsilon=0} = \frac{3}{2\bar{\sigma}}\underline{\underline{S}}:\underline{\underline{\underline{\underline{P}}}}:\delta\underline{\underline{\sigma}}. \qquad (1.36)$$

Finally the derivatives of the modified Litonski flow law with respect to $\bar{\sigma}$, T and $\bar{\gamma}_p$ is carried out as follow

$$\begin{aligned}
\left.\frac{dg(\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}}),T,\bar{\gamma}_p)}{d\epsilon}\right|_{\epsilon=0} &= \frac{\partial g(\bar{\sigma},T,\bar{\gamma}_p)}{\partial\bar{\sigma}} \times \left.\frac{d\bar{\sigma}(\underline{\underline{\sigma}}+\epsilon\delta\underline{\underline{\sigma}})}{d\epsilon}\right|_{\epsilon=0} \\
&= m\dot{\gamma}_0\left[\frac{\frac{\bar{\sigma}}{\sigma_0}}{\left[1+\frac{\bar{\gamma}_p}{\gamma_0}\right]^N\left\{1-\delta\left[\exp\left(\frac{T-T_0}{k}\right)-1\right]\right\}}\right]^{m-1} \\
&\quad\times\frac{\frac{1}{\sigma_0}}{\left[1+\frac{\bar{\gamma}_p}{\gamma_0}\right]^N\left\{1-\delta\left[\exp\left(\frac{T-T_0}{k}\right)-1\right]\right\}} \\
&\quad\times\frac{3}{2\bar{\sigma}}\underline{\underline{S}}:\underline{\underline{\underline{\underline{P}}}}:\delta\underline{\underline{\sigma}} \qquad (1.37) \\
&= m\dot{\gamma}_0\left[\frac{\frac{\bar{\sigma}}{\sigma_0}}{\left[1+\frac{\bar{\gamma}_p}{\gamma_0}\right]^N\left\{1-\delta\left[\exp\left(\frac{T-T_0}{k}\right)-1\right]\right\}}\right]^{m} \\
&\quad\times\frac{1}{\bar{\sigma}}\frac{3}{2\bar{\sigma}}\underline{\underline{S}}:\underline{\underline{\underline{\underline{P}}}}:\delta\underline{\underline{\sigma}} \\
&= m\frac{g(\bar{\sigma},T,\bar{\gamma}_p)}{\bar{\sigma}}\times\frac{3}{2\bar{\sigma}}\underline{\underline{S}}:\underline{\underline{\underline{\underline{P}}}}:\delta\underline{\underline{\sigma}}.
\end{aligned}$$

$$
\begin{aligned}
\left.\frac{dg(\bar{\sigma}, T+\epsilon\delta T, \bar{\gamma}_p)}{d\epsilon}\right|_{\epsilon=0} &= m\dot{\gamma}_0 \left[\frac{\frac{\bar{\sigma}}{\sigma_0}}{\left[1+\frac{\bar{\gamma}_p}{\gamma_0}\right]^N \left\{1-\delta\left[\exp\left(\frac{T-T_0}{k}\right)-1\right]\right\}}\right]^{m-1} \\
&\quad \times \frac{\frac{\bar{\sigma}}{\sigma_0}}{\left[1+\frac{\bar{\gamma}_p}{\gamma_0}\right]^N} \times -\frac{1}{\left\{1-\delta\left[\exp\left(\frac{T-T_0}{k}\right)-1\right]\right\}^2} \\
&\quad \times -\delta \exp\left(\frac{T-T_0}{k}\right)\frac{\delta T}{k} \\
&= mg(\bar{\sigma}, T, \bar{\gamma}_p)\frac{\delta\exp\left(\frac{T-T_0}{k}\right)}{\left\{1-\delta\left[\exp\left(\frac{T-T_0}{k}\right)-1\right]\right\}}\frac{\delta T}{k}
\end{aligned}
\tag{1.38}
$$

$$
\begin{aligned}
\left.\frac{dg(\bar{\sigma}, T, \bar{\gamma}_p+\epsilon\delta\bar{\gamma}_p)}{d\epsilon}\right|_{\epsilon=0} &= m\dot{\gamma}_0 \left[\frac{\frac{\bar{\sigma}}{\sigma_0}}{\left[1+\frac{\bar{\gamma}_p}{\gamma_0}\right]^N \left\{1-\delta\left[\exp\left(\frac{T-T_0}{k}\right)-1\right]\right\}}\right]^{m-1} \\
&\quad \times -\frac{\frac{\bar{\sigma}}{\sigma_0}}{\left\{1-\delta\left[\exp\left(\frac{T-T_0}{k}\right)-1\right]\right\}} \times -N\frac{\frac{\delta\bar{\gamma}_p}{\gamma_0}}{\left[1+\frac{\bar{\gamma}_p}{\gamma_0}\right]^{N+1}} \\
&= mg(\bar{\sigma}, T, \bar{\gamma}_p)\frac{N}{\gamma_0+\bar{\gamma}_p}\delta\bar{\gamma}_p
\end{aligned}
\tag{1.39}
$$

The block structure of the final Jacobian matrix is given by

$$
\mathbf{J} = \begin{bmatrix}
\mathbf{J}_{uu} & 0 & \mathbf{J}_{u\sigma} & 0 \\
0 & \mathbf{J}_{TT} & \mathbf{J}_{T\sigma} & \mathbf{J}_{T\bar{\gamma}_p} \\
\mathbf{J}_{\sigma u} & \mathbf{J}_{\sigma T} & \mathbf{J}_{\sigma\sigma} & \mathbf{J}_{\sigma\bar{\gamma}_p} \\
0 & \mathbf{J}_{\bar{\gamma}_p T} & \mathbf{J}_{\bar{\gamma}_p\sigma} & \mathbf{J}_{\bar{\gamma}_p\bar{\gamma}_p}
\end{bmatrix}
\tag{1.40}
$$

Remark 4: Note that this final expression of the Jacobian is non symmetric which will restrict the choice of available solvers later on. ■

Employing a finite element discretization in space, the fields $\underline{u}$, T, $\underline{\underline{\sigma}}$ and $\bar{\gamma}_p$ are represented on a mesh as

$$\begin{aligned}
\underline{u}(\mathbf{x}) &= \sum_i N_i^u(\mathbf{x}) u_i(t) \\
T(\mathbf{x}) &= \sum_i N_i^T(\mathbf{x}) T_i(t) \\
\underline{\underline{\sigma}}(\mathbf{x}) &= \sum_i N_i^\sigma(\mathbf{x}) \sigma_i(t) \\
\bar{\gamma}_p(\mathbf{x}) &= \sum_i N_i^{\bar{\gamma}_p}(\mathbf{x}) \bar{\gamma}_{pi}(t)
\end{aligned} \tag{1.41}$$

where u_i, T_i, σ_i and $\bar{\gamma}_{pi}$ are nodal values associated to the i$^{\text{th}}$ node of the mesh and N_i^u, N_i^T, N_i^σ and $N_i^{\bar{\gamma}_p}$ are their respective shape functions. Assuming Galerkin finite element formulation, the admissible test functions w_u, w_T, w_σ and w_γ are chosen similarly to the unknow fields but their respective shape functions vanish on the Dirichlet boundary.

This finite element formulation leads the following discretized Jacobian matrix for system (1.12)

$$\mathbf{J} = \begin{bmatrix} \mathbf{M}_u & 0 & \mathbf{K}_u & 0 \\ 0 & \mathbf{M}_T + \mathbf{K}_T + \mathbf{G}_{TT} & \mathbf{G}_{T\sigma} & \mathbf{G}_{T\bar{\gamma}_p} \\ \mathbf{K}_\sigma & \mathbf{G}_{\sigma T} & \mathbf{M}_\sigma + \mathbf{G}_{\sigma\sigma} & \mathbf{G}_{\sigma\bar{\gamma}_p} \\ 0 & \mathbf{G}_{\bar{\gamma}_p T} & \mathbf{G}_{\bar{\gamma}_p \sigma} & \mathbf{M}_{\bar{\gamma}_p} + \mathbf{G}_{\bar{\gamma}_p\bar{\gamma}_p} \end{bmatrix} \tag{1.42}$$

where the terms called $\mathbf{M}_*$ can be interpreted as mass matrices, the terms $\mathbf{K}_*$ can be interpreted as stiffness terms and the terms $\mathbf{G}_{**}$ are terms arising from the derivation of the plastic flow rule g.

In the linear a regime before plastic deformation starts, all $\mathbf{G}_{**}$ terms drop and the Jacobian reduces to the following simplified expression

$$\mathbf{J} = \mathbf{J}^{lin} = \begin{bmatrix} \mathbf{J}_{uu} & \mathbf{0} & \mathbf{J}_{u\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{TT} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{\sigma u} & \mathbf{J}_{\sigma T} & \mathbf{J}_{\sigma\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{\bar{\gamma}_p\bar{\gamma}_p} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_u & \mathbf{0} & \mathbf{K}_u & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_T + \mathbf{K}_T & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_\sigma & \mathbf{M}_{\sigma T} & \mathbf{M}_\sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{\bar{\gamma}_p} \end{bmatrix}. \tag{1.43}$$

In chapter 2 the difficulties associated with the spatial discretization of system (1.12) are presented and the appropriate choice of shape functions for all fields is investigated and various element configurations are studied, considering the Babuška-Brezzi (BB) condition for the stability. The performance of higher order shape functions upon mesh refinement is investigated and report.

In chapter 3 a general preconditioning algorithm for any discretization of system (1.12) is proposed. These preconditioners are based on the Schur complement approach accounting for the the multiphysics nature of the problem. The performance of the proposed preconditioners are assessed in serial for higher order shape functions as well as in parallel for low order shape functions on a supercomputer machine at Araggone National Laboratory.

In chapter 4 a domain decomposition technique is developed to account for the localized feature of shearbands. This key idea is to concentrate most of the preconditioning work in the shear band domain where most of the deformation takes place, while reusing cheaper preconditioners in the region outside the shearband domain. The performance of this preconditioner is investigated on low and high-order mesh descretization and it is shown that such approach leads to a fast and efficient algorithm.

Concluding remarks and future work directions are presented in chapter 5.

Chapter 2

Isogeometric Analysis of Shearbands

A major issue with the simulation of shear bands is that the numerical approach may lead to mesh size and alignment sensitivity. These problems are not specific to shear bands but also observed in most problems where strain localization occur.

Despite the original intent, perhaps the most interesting feature of Isogeometric Analysis (IGA) is the property of the Non-uniform rational B-spline (NURBS) shape functions when used for simulation. Basis functions constructed with NURBS are globally continuous across elements, not only within them. This means that a solution computed with IGA is higher order and not piecewise higher order as would be the case with the *p-version* of the finite elements method. Moreover, it is also easy to raise the polynomial order of these NURBS basis [36] which lead to more accurate simulations [15] with roughly as many nodes as for a simulation with a low order basis. This last property is due to the fact that IGA allows for a new type of refinement named $k\text{-}refinement$, an attractive feature which raises the order of the shape functions without excessively increasing the size of the linear system to be solved. Nonetheless, it should be noted that these matrices are also much denser than standard FEM matrices since the NURBS basis is not compactly supported.

In addition, the IGA framework retains the ability to employ the B-bar method [22, 38] which is used to remediate the volumetric locking problems that appears during the simulation of nearly incompressible deformation.

Here we reduce the computational cost of shear band simulations by finding shape functions which furnish the greatest accuracy for the lowest cost. To this end we tested multiple elements, with a focus on IGA based elements. The main evaluation method of these discretizations is study of the convergence rates, volumetric locking and memory usage. In the following sections of this chapter we begin by presenting the IGA formulation for the mixed finite elements, then we study two test cases and assess the convergence properties of the proposed elements. The formulation is shown to be free of both mesh size and mesh alignment sensitivity, and leads to fast rates of convergence.

2.1 NURBS-based isogeometric analysis for mixed formulations

In this section we review the basics of NURBS functions and how they are used in the paradigm of isogeometric analysis (IGA) [32, 55]. More specifically, we will focus on how mixed formulations are discretized in IGA.

NURBS are functions designed to exactly represent curves and were originally created for computational geometry purposes. Detailed description of these functions, their properties, and how to compute them can be found in Piegl and Tiller [83] and Rogers [88].

NURBS are parametrized curves in d-dimensional spaces ($\mathbb{R}^d$ for example), constructed as a tensorial product of d 1D-NURBS curves. 1D-NURBS are themselves constructed as a rational combination of B-splines. In order to further describe B-splines functions we need to introduce the parametric space used to construct them.
The parametric space associated with a B-spline function is called a knot vector and is composed of non-decreasing real numbers (the knots) $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$. The multiplicity of knots cannot exceed the degree of the curve $p + 1$ where p is also the degree of the poly-

nomial basis used to construct the B-spline. n is the number of basis functions defined on the knot vector, it is also the number of control points used to describe the geometry of the B-spline. A knot vector is said to be open if its first and last knots have multiplicity $p+1$. Open knot vectors are commonly used since they produce bases that are interpolatory at the endpoints of the interval.

B-spline basis of order p are constructed recursively from an underlying piecewise constant basis

$$N_i(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi \leq \xi_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

Bases of order p are then constructed using the following recursion formula

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{2.2}$$

first introduced by Cox and de Boor [36].

The associated B-spline curve is obtained using a linear combination of the basis function as follows

$$\mathcal{C}(\xi) = \sum_{i=1}^{n} \mathbf{P}_i N_i(\xi), \tag{2.3}$$

where $\mathbf{P}_i \in \mathbb{R}^d$ is the $i-th$ control point. An example of a 1D basis of NURBS functions and its associated curve are presented in Figure 2.1. Note how the shape functions are non zero over several elements instead of two as is the case for linear Lagrange or Hermite polynomials. It can also be observed that the curve is not interpolatory at all points (i.e. the curve does not pass through the point), it generally is interpolatory only at the boundary.

The B-splines can then be extended to multiple dimensions using a tensorial product, in the case of $d=2$, we get the following expression for a B-spline surface

$$\mathcal{S}(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_i(\xi) M_j(\eta) \mathbf{P}_{ij}, \tag{2.4}$$

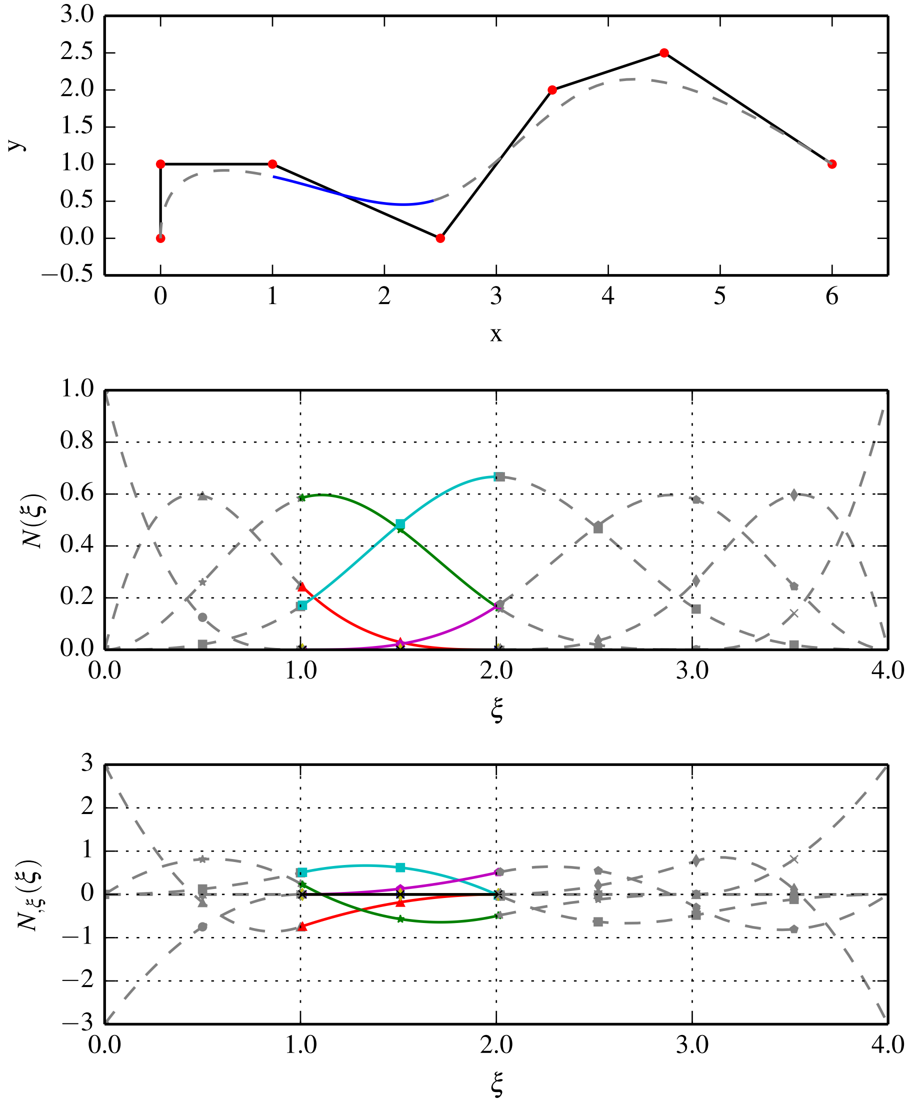


Figure 2.1: The top plot represents the curve $\mathcal{C}(\xi)$ in the physical space, the second and third plots represent the shape functions corresponding to the discretization of $\mathcal{C}(\xi)$ and their derivatives in the parametric space. The curve, the shape functions and their derivatives are ploted in color for $\xi \in [1.0, 2.0]$ and dashed gray for $\xi \in [0.0, 1.0[\cup]2.0, 4.0]$, ● represent the control points.

where $M_j(\eta)$ are basis functions defined by the following knot vector $\mathcal{H} = \{\eta_1, \ldots, \eta_{m+q+1}\}$. Finally NURBS basis functions are created using a rational combination of 1D B-spline basis functions, here again for $d = 2$, the result is

$$R_{ij}(\xi, \eta) = \frac{N_i(\xi)M_j(\eta)w_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} N_i(\xi)M_j(\eta)w_{ij}}. \tag{2.5}$$

These shape functions are combined with the control points to construct NURBS surfaces as follows

$$\mathcal{S}(\xi, \eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{ij}B_{ij}. \tag{2.6}$$

An example of 2D NURBS shape function used to discretize a quarter of a plate with a central circular hole are presented in Figure 2.2. The NURBS basis forms a partition of unity which makes it easy to use as a shape function basis for the finite element method (FEM). NURBS *k-refinement* preserves the number and geometry of the elements on the mesh for any order of shape functions, which allows us to use NURBS of different order in a mixed formulation. These properties will also be used to compute different basis of shape functions for the $\overline{\text{B}}$-bar projection technique [38, 39, 99].

2.1.1 Spatial discretization of shear band

The shear band problem in System (1.12) is discretized in space by choosing appropriate shape functions for the solution fields. In a general framework, different shape functions are used for each field so that

$$\begin{aligned}
\mathbf{u} &= \mathbf{N_u}\mathbf{u} & w_u &= \mathbf{N_u}\mathbf{w_u}, \\
T &= \mathbf{N_T}\mathbf{T} & w_T &= \mathbf{N}_T\mathbf{w_T}, \\
\sigma &= \mathbf{N}_\sigma\sigma & w_\sigma &= \mathbf{N}_\sigma\mathbf{w}_\sigma, \\
\bar{\gamma}_p &= \mathbf{N}_{\bar{\gamma}_p}\bar{\gamma}_p & w_{\bar{\gamma}_p} &= \mathbf{N}_{\bar{\gamma}_p}\mathbf{w}_{\bar{\gamma}_p},
\end{aligned} \tag{2.7}$$

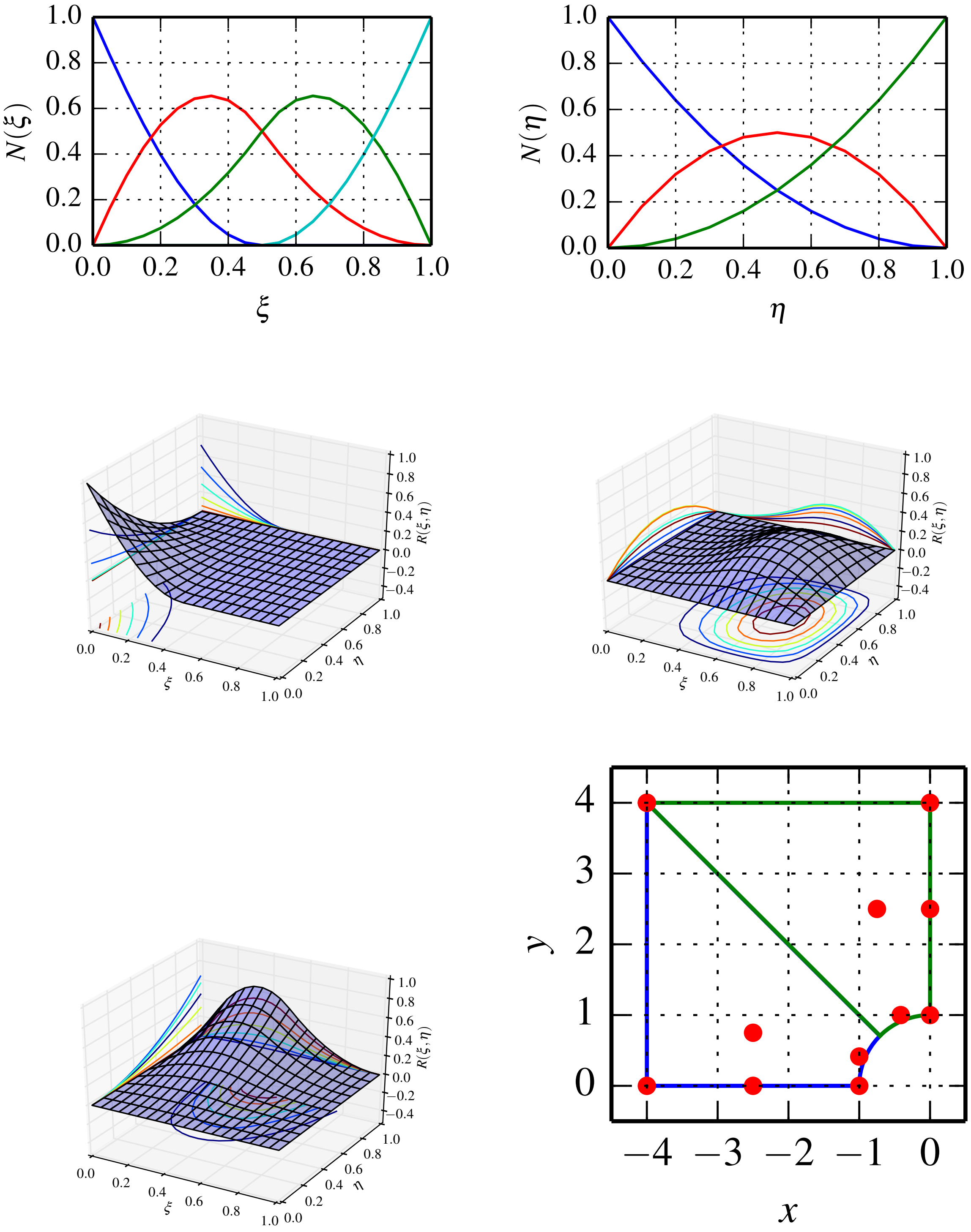


Figure 2.2: The first row shows the 1D shape functions in the ξ and η parametric spaces, the three following figures show 2D shape functions resulting from the rational combination of the 1D shape functions. Finally the last figure shows the control point and the contour of the quarter plate.

where $\mathbf{N}_*$ is in practice stored as 1D arrays of shape functions using the Voigt notation and the assembly process of the finite element method is applied to get a global vector. At this point, at least two different type of limitations are going to guide our choice of shape functions.

We note that spatial derivatives of the displacement and temperature fields appear in our formulation, therefore the shape functions $\mathbf{N_u}$ and $\mathbf{N_T}$ must be at least linear. In addition, no derivatives of stress and equivalent plastic strain (EQPS) appear in the weak form, hence $\mathbf{N}_\sigma$ and $\mathbf{N}_{\bar{\gamma}_p}$ are only required to be piecewise constants.

Second, to obtain a stable element, we need to verify that our mixed formulation based on the Hellinger-Reissner variational formulation (displacement/stress formulation) [49, 86] satisfies the Babuška-Brezzi (BB) condition [9, 10, 23]. The stress/displacement formulation (in the case of linear elasticity for the sake of clarity) is composed of the momentum equilibrium and the constitutive relation. This formulation can be compared to the displacement/pressure formulation typically used for incompressibility problems [52], which comprises the momentum equation and the definition of pressure in a solid. Both of these formulations lead to a saddle point problem, as presented in Table 2.1

	stress / displcement	displacement / pressure
Mixed Form	$\begin{cases} \underline{\underline{\sigma}} - \underline{\underline{\underline{\underline{C}}}}^{elas} : \underline{\nabla}^s \underline{u} = 0 \\ -\underline{\nabla} \cdot \underline{\underline{\sigma}} = \underline{b} \end{cases}$	$\begin{cases} -\underline{\nabla} \cdot \left[-p\underline{\underline{I}} + 2\mu\underline{\nabla}\ \underline{u}\right] = \underline{f} \\ \underline{\nabla} \cdot \underline{u} - \dfrac{p}{\lambda} = 0 \end{cases}$
Algebraic Form	$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \sigma \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{g} \end{bmatrix}$	$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}$

Table 2.1: comparison of mixed formulations leading to saddle point problems with λ and μ the Lamé constants and p the pressure in the solid. Appropriate boundary conditions are assumed for both formulations.

We see that the stress in the Hellinger-Reissner formulation plays a similar role as the displacement in the incompressible formulation. More examples and details about these classic mixed formulations for elasticity as well as detailed explanation of the Babuška-

Brezzi condition can be found in [8].

To develop an element for shear bands that will be stable and will not lock under certain loading conditions, it should pass the inf-sup criteria [24]. Various methods exist in the literature to assess the inf-sup condition, for example, one can analyze numerically the eigenvalues of the assembled Jacobian, as proposed by Bathe *et al.* [13, 35] and determine if an element will satisfy the inf-sup condition. While such tests have been shown to work well on some simple problems such as Stokes flow, the acoustic fluids and the bending of Reissner-Mindlin plates, it is not trivial how these techniques can be extended to complex, time dependent, multiphysics problems, as the current shear band problem.

One engineering way to determine if an element might pass the inf-sup criteria is via a constraints count approach, discussed in Hughes [99] and Zienkiewicz and Taylor [110]. We emphasize that the constraints count method is a necessary but not sufficient approach to pass the inf-sup condition, hence even if an element passes the constraint count method, it doesn't guarantee it will be stable and locking free. Nonetheless, it is an easy test to construct and allows us to quickly filter out elements that would not be stable in any case.

The idea is to construct a ratio $r = \frac{n_{eq}}{n_c}$ with n_{eq} the number of equilibrium equations and n_c the number of constraints due to the constitutive relation used, such that $r \approx 1$, for a stable, non locking element. If the ratio is approaching from bellow, $r \to 1^-$, then the element may be subject to stresses oscillations, which can lead to an unstable element [110]. While if it is approaching from above, $r \to 1^+$, then the element may experience severe locking.

For 2D mixed formulations, given in the left column of Table 2.1, we have three independent stresses σ_{xx}, σ_{yy} and σ_{xy} as well as two independent displacements u_x and u_y where at least three displacements degrees of freedom should be fixed to avoid rigid body motion. These considerations lead us to inequality (2.8) which has to be satisfied in order to have a stable element

$$3n_\sigma - 2n_u + 3 \geq 0, \tag{2.8}$$

where n_σ and n_u are the number of stress and displacement degrees of freedom in the problem. Now we can express the number of degrees of freedom of each field as a function of the polynomial orders N_σ and N_u used for the shape functions $\mathbf{N}_\sigma$ and $\mathbf{N}_u$ to obtain the final inequality that relates the shape function order with the constraint count method

$$3N_\sigma^2 - 2N_u^2 + 3 \geq 0. \tag{2.9}$$

The different choices of P_u/P_σ couples and the optimal order of suggested shape functions, based on inequality (2.9), are summarized in Table 2.2. It appears that the most reasonable choice for a mixed formulation would be a P^n/P^n (e.g. P^1/P^1 or P^2/P^2) element or a P^4/P^3 element, that have good r-ratio values but also relatively low shape functions order. Other elements with higher order shape functions are also a possible but might lead to large bands in the jacobian matrix and not necessarily greater computational efficiency.

Order of P_u/P_σ	r-ratio	associated polynomial	roots	smallest acceptable value for n
n/n	$\frac{3(n+1)^2}{2(n+1)^2-3}$	n^2+2n+4	*imaginary*	0
n/(n-1)	$\frac{3n^2}{2(n+1)^2-3}$	n^2-4n+1	$2\pm\sqrt{3}$	4
n/(n-2)	$\frac{3(n-1)^2}{2(n+1)^2-3}$	$n^2-10n+4$	$3\pm\sqrt{21}$	10
n/(n-3)	$\frac{3(n-2)^2}{2(n+1)^2-3}$	$n^2-16n+13$	$8\pm\sqrt{51}$	16

Table 2.2: Choices of shape functions using the constraints count method

An illustration of the P^4/P^3 NURBS element is shown in Figure 2.3. Note that even though the number of nodes in this element is high, on larger meshes most nodes are shared with adjacents elements, thus in practice the total number of unknowns is comparable to that of a linear element.

The constraint count technique as described in this section will be used to assess the suitability of more complex elements, specifically derived for shear bands, that involve additional solution fields: temperature and plastic strains. Since shear banding involves large, near incompressible deformations, a suitable element for shear bands must be free of locking.

Otherwise, the model will be artificially too stiff, and the shear band too diffuse. While a similar artificial stiffening may occur due to under resolved solution gradients, this issue can always be resolved by mesh refinement, whereas volumetric locking cannot.

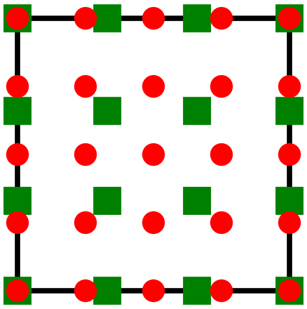


Figure 2.3: Schematic representation of the P^4P^3 element with displacement and temperature nodes represented by ● and stress and equivalent plastic strain nodes represented by 🟩.

2.1.2 Near incompressibility and the B-bar method

Verifying that elements pass the BB condition is a necessity in order to have a stable element, but this does not mean that the element has any good convergence properties. For shear bands, volumetric locking is a severe risk in elements since the shear banding process in metallic materials is modeled with J_2 plasticity which leads to nearly isochoric deformations in the solid. Modeling this type of deformation is not simple if one wants to avoid volumetric locking, therefore herein we implemented the volumetric strain projection technique called B-bar [22, 38, 99]. The idea of the B-bar method is to reduce the amount of strain energy in the solid by projecting the volumetric strain onto a lower order basis function. This technique can also be interpreted as a selective reduced order integration [99].

The projected strain is discretized as a field depending on $\bar{\mathbf{N}}_u$, where $\bar{\mathbf{N}}_u$ is a shape function used for $\underline{u}$ and T but with one order lower continuity than the "full order" shape function basis $\mathbf{N}_u$, so that

$$\bar{\epsilon}(\mathbf{x}) = \bar{\mathbf{N}}_u(\mathbf{x})\alpha, \tag{2.10}$$

where α is the vector of lower order stains, in practice this vector is not computed. At the same time, the "full order" strain is expressed as a function of the nodal displacements $\mathbf{u}$ and the "full order" shape function basis $\mathbf{N}_u$

$$\epsilon(\mathbf{x}) = \underline{\nabla}^s \mathbf{N}_u(\mathbf{x})\mathbf{u}. \tag{2.11}$$

Thus, the relation between α and u can be obtained by enforcing the equality of the strain and lower order strain in a weak sense using the lower order shape function basis. This is done by solving the following least square problem

$$\int_\Omega \bar{\mathbf{N}}_u(\mathbf{x})\bar{\epsilon}(\mathbf{x})d\Omega = \int_\Omega \bar{\mathbf{N}}_u(\mathbf{x})\epsilon(\mathbf{x})d\Omega, \tag{2.12}$$

which leads, after replacing $\bar{\epsilon}$ and ϵ by their respective expressions, to the following expression for α

$$\alpha = \bar{\mathbf{M}}^{-1} \int_\Omega \bar{\mathbf{N}}_u(\mathbf{x}) \ \underline{\nabla}^s \mathbf{N}_u(\mathbf{x})\mathbf{u} \ d\Omega, \tag{2.13}$$

Note that a lower order mass matrix is employed in this process, which is given by

$$\bar{\mathbf{M}} = \int_\Omega \bar{\mathbf{N}}_u(\mathbf{x})\bar{\mathbf{N}}_u(\mathbf{x}) \ d\Omega, \tag{2.14}$$

and the final projected strain is

$$\bar{\epsilon}(\mathbf{x}) = \bar{\mathbf{N}}_u(\mathbf{x}) \left(\bar{\mathbf{M}}^{-1} \int_\Omega \bar{\mathbf{N}}_u(\mathbf{x}) \ \underline{\nabla}^s \mathbf{N}_u(\mathbf{x})\mathbf{u} \ d\Omega \right). \tag{2.15}$$

Next, the volumetric part of the strain is replaced by the projected volumetric strain, which is simply done by redefining the strain as follow [38]

$$\epsilon(x) = \epsilon(x) - \frac{tr(\epsilon(x)) - tr(\bar{\epsilon}(x))}{3}\mathbf{I} \tag{2.16}$$

In our formulation only two blocks of the jacobian matrix contain strain terms: $\mathbf{K}_u$ and $\mathbf{K}_\sigma$, hence we introduce the projected strain in these terms by the following update

$$\mathbf{K}_u = \int_\Omega \underline{\nabla}\ \mathbf{w_u} \delta\sigma d\Omega \quad \rightarrow \quad \bar{\mathbf{K}}_u = \int_\Omega \underline{\nabla}\ \bar{\mathbf{w}}_\mathbf{u} \delta\sigma d\Omega, \tag{2.17}$$

$$\mathbf{K}_\sigma = -\int_\Omega \mathbf{w}_\sigma \left(\mathbf{C}^{elas} \underline{\nabla}^s \delta\dot{\mathbf{u}}\right) d\Omega \quad \rightarrow \quad \bar{\mathbf{K}}_\sigma = -\int_\Omega \mathbf{w}_\sigma \left(\mathbf{C}^{elas} \underline{\nabla}^s \delta\dot{\mathbf{u}}\right) d\Omega. \tag{2.18}$$

2.1.3 Mixed elements investigated

Taking into account the difficulties explained in the previous sections:

1. balancing the number of degrees of freedom for stresses and displacements in order to get a good r-ratio,
2. alleviating volumetric locking,
3. avoiding spurious oscillations of the equivalent plastic strain field,

we implement and investigate the behavior of multiple elements, grouped in three main categories:

- Pian and Sumihara type elements
- Mixed formulation based elements
- Irreducible elements

The characteristics of each element in their respective family are presented in the following paragraphs, where Table 2.3 summarizes the nomenclature and Table 2.4 the principal features of each elements. In the following subsections we describe each family of methods in more details.

Name	Abbreviation
Pian-Sumihara Shear band Quad	PSSQ
Mixed formulation NURBS Shear band Quad	MNSQ
Hybrid NURBS Shear band Quad 1	HNSQ1
Hybrid NURBS Shear band Quad 2	HNSQ2
Irreducible NURBS Shear band Quad	INSQ

Table 2.3: Nomenclature of elements implemented in this study. These names may be followed by a number in parentheses which indicate the order of the shape functions used (i.e. HNSQ2(3) would indicate that the HNSQ2 element is employed with cubic NURBS shape functions).

2.1.3.1 Pian and Sumiharas elements

The Pian-Sumihara (PSSQ) based element is an element designed to improve the convergence rate of the finite element method when the Hellinger-Reissner variational form is used. For the 2D case, σ_{xy} is constant and σ_{xx} and σ_{yy} are modeled with 2 degrees of freedom accounting for the constant and linear part of these stresses. For more details, see [59, 81, 82, 109]. This formulation ensures that the element has a perfectly balanced ratio of constraints: r-ratio=1. This element has also no apparent problem of volumetric locking, which makes it an excellent element for this shear band problem [73]. However, one significant disadvantage of the PSSQ element is that it is solely based on a low order element (bilinear element). Hence extensions to higher order elements, and in particular NURBS functions, are not trivial.

2.1.3.2 Mixed formulation NURBS elements

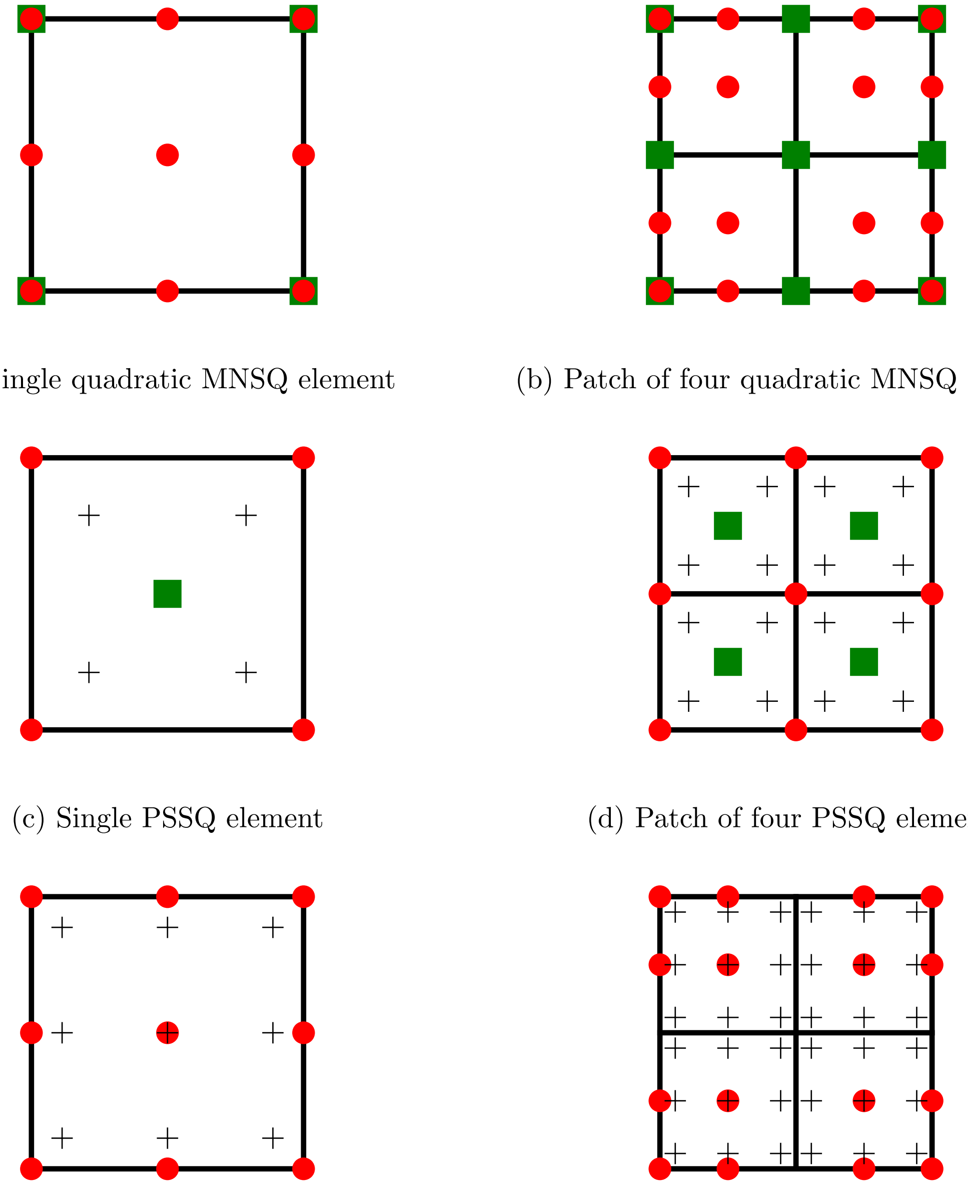
This family of elements is based on a standard mixed formulation element (MNSQ) [21] devised by applying the constraint count technique to determine which orders of shape

functions for each field is the most likely to yield a stable element. This element type takes full advantage of the NURBS shape functions for each field and so high order convergence is expected. Preliminary research in 1D confirmed this good performance, but with 2D elements, the outlook becomes more complex. Volumetric locking is a concern, and even after implementation of the B-bar method, using low order shape functions leads to slow convergence. However higher order elements does indeed alleviate some of these issues and was shown to give good results on the examples presented in section 2.2.1.
However, a clear disadvantage is the fact that using linear and higher order shape functions for the equivalent plastic strain field leads to severe oscillations when solving the problem presented in section 2.2.2. These oscillations are so severe that the Newton solver diverges when the gradient of the EQPS field becomes large. Due to this instability, the mixed formulation elements are thus only used for the more simple tension loading cases.

2.1.3.3 Irreducible elements

In order to obtain an acceptable r-ratio, an irreducible element (INSQ), using the approach described by Comi *et al.* [31], was implemented for shear bands by McAuliffe *et al.* [74]. The basic idea is to treat the stresses and the equivalent plastic strain as history variables computed at the gauss points and then to assemble the contribution of each gauss point to the global system in order to maintain a fully monolithic solver. This approach converges on all the test cases and also simplifies the implementation since it only requires to compute shape functions for the displacement and temperature fields.

Nonetheless, introducing history variables in the discretization also has its drawbacks. One probelm is the memory usage required for the sampling at the gauss points as explained in section 2.1.4.2. Another problem is that history variables do not provide the same rate of convergence as the high order NURBS fields. This is due to the fact that the interpolation for history variables does not have NURBS properties of continuity and regularity. To this



(a) Single quadratic MNSQ element

(b) Patch of four quadratic MNSQ elements

(c) Single PSSQ element

(d) Patch of four PSSQ elements

(e) Single quadratic INSQ element

(f) Patch of four quadratic INSQ elements

Figure 2.4: Schematic representation of the studied elements. ● represents displacement and temperature degrees of freedom. 🟩 represents stresses and EQPS for the MNSQ element and stress for the PSSQ element. + represents the EQPS degrees of freedoms for PSSQ and stress and EQPS for INSQ.

end, we attempted to implement a hybrid NURBS-irreducible element (HNSQ1), that uses NURBS shape functions to discretize the EQPS. However, this element is prone to oscillations of the EQPS field that are severe enough to prevent it from converging on example 2 from section 2.2.2. On the other hand, reformulation of HNSQ1 using the NURBS shape functions to discretize the stress field instead of the EQPS leads to a stable element (HNSQ2) that is not subject to oscillations of the EQPS field. Thus, in this paper we will present convergence results focusing on this element. One future direction to further improve the HNSQ2 element could be based on a discontinuous Galerkin type formulation. Such formulation might be able to benefit from higher order shape functions for all fields including the EQPS field and remain stable.

Name	Shape functions				r-ratio	$\overline{\text{B}}$-bar	Oscillations in the EQPS field	Convergence
	$\mathbf{N_u}$	$\mathbf{N_T}$	$\mathbf{N}_\sigma$	$\mathbf{N}_{\gamma_p}$				
PSSQ	P^1	P^1	PS	I	1	no	no	yes
MNSQ	P^n	P^n	$\text{P}^{(n-1)}$	$\text{P}^{(n-1)}$	1.021	yes	yes	no
HNSQ1	P^n	P^n	I	P^n	1.596	yes	yes	no
HNSQ2	P^n	P^n	P^n	I	1.596	yes	no	yes
INSQ	P^n	P^n	I	I	1.596	yes	no	yes

Table 2.4: Summary of the tested elements, the r-ratio is calculated for the shape function order that yields the most stable element.

2.1.4 Implementation of IGA in a finite element code

In this section the main modifications needed to implement IGA in a finite element code are presented. The modifications presented apply for single patch simulations. In the case of multiple patches a connectivity and assembly of the patches is required. More details on the computational cost and the memory usage for the different elements studied are presented in section 2.1.3.

2.1.4.1 IGA based finite element algorithm

Most FE codes follow the layout presented in Algorithm 1. The first step in an IGA implementation is to generate/read input parameters used in simulations, which specifically to IGA include: control points coordinates, knot vectors and other relevant information related to the mesh. Then, memory is allocated for the necessary arrays as presented in Algorithm 1. This task may become more complex when $\overline{\text{B}}$-bar is used since it requires a lower order mesh to be generated as well. Book keeping also needs to be planed carefully to keep track of the nodes introduced at the Gauss points in the case of elements HNSQ1, HNSQ2 and INSQ.

Algorithm 1 IGA-FE code layout

```
subroutine main()
call preprocessing (read input material parameters, control points, knot vectors, $\overline{\text{B}}$-bar
  flag, elements type flag)
for t = 0, T do                                                Adaptive time stepping loop
    while $\| \mathbf{R} \| > tol \cdot \| \mathbf{R}_0 \|$ do                        Newton solver loop
        for e = 1, number of elements do                                 Assembly loop
            get NURBS shape functions $\mathbf{N}$                                  Eq. (2.5)
            compute element residual $\mathbf{R}$                                        ??
            if $\overline{\text{B}}$-bar then
                compute $\overline{\text{B}}$-bar terms for Jacobian                Eqs. (2.17)-(2.18)
            end if
            compute element Jacobian $\mathbf{J}$                                        ??
        end for
        solve linearized system: $\delta\mathbf{u} = \mathbf{J}^{-1}\mathbf{R}$
    end while
    update solution: $\mathbf{u} = \mathbf{u} + \delta\mathbf{u}$
    call postprocessing (Algorithm 2 and Figure 2.5), to post process values from con-
  trol points to physical domain.
end for
```

The shape functions used in the residual and jacobian calculations are computed using special procedures in order to obtain the NURBS shape functions and their derivatives. First, B-spline functions are computed using the algorithm in [83] and then NURBS shape functions are obtained from these B-spline functions via the methods outlined in [32]. It is important to note that the solution of the system of equations is obtained in terms of quantities associated with control points, Gauss points. Thus, first an L_2-projection is employed to transfer all the quantities to control points. This process is illustrated in Figure 2.5 by going from the mesh shown in Figure 2.5(a) to the mesh with only control points in Figure 2.5(b). Then, an interpolation step is required to transfer all quantities from control points, illustrated by Figure 2.5(b), to nodes of the actual physical mesh, illustrated by Figure 2.5(c)-(d), for results output. Note that for plotting purposes it is common to resolve solution features (e.g. shearbands) on the physical mesh by subdividing the original mesh into a locally refined mesh. Algorithm 2 describes this post processing step.

For example, consider a quantity α^{gp} computed at the Gauss points that we want to express as

$$\alpha^{gp} = \mathbf{N}_u \alpha, \tag{2.19}$$

where $\mathbf{N}$ are the NURBS shape functions and α is the vector of values we wish to obtain at the control points. The following least square problem is then solved to obtain α

$$\mathbf{M}\alpha = \int_\Omega \mathbf{N}_u \alpha^{gp} d\Omega, \tag{2.20}$$

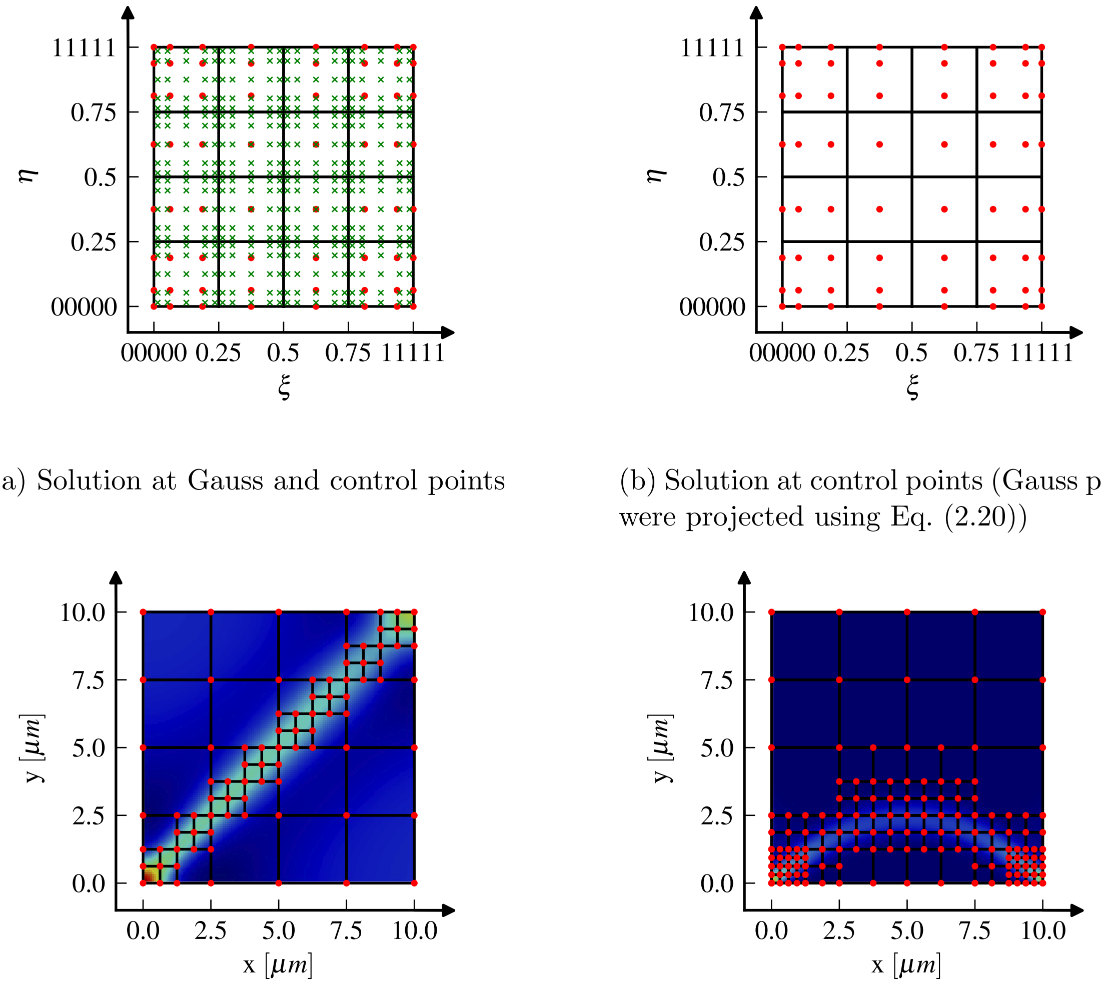
where $\mathbf{M} = \displaystyle\int_\Omega \mathbf{N}_u^T \mathbf{N}_u d\Omega$ is the mass matrix associated with the specific NURBS shape functions (note that $\mathbf{M}$ is different from $\bar{\mathbf{M}}$ used for B-bar method).

Finally, the solutions (displacement, temperature, stresses and equivalent plastic strain) at the control points are interpolated to the refined physical mesh by a standard interpolation employing NURBS functions at any point x_j

$$\underline{u}(x_j) = \mathbf{N}_u(x_j)\mathbf{u}. \tag{2.21}$$

2.1.4.2 Memory and computational cost considerations

The computational and memory costs associated with the Isogeometric elements are mainly influenced by the number of variables introduced in the linear system of equations solved



(a) Solution at Gauss and control points

(b) Solution at control points (Gauss point values were projected using Eq. (2.20))

(c) Solution on the physical mesh for the top traction example

(d) Solution on the physical mesh for the top shear example

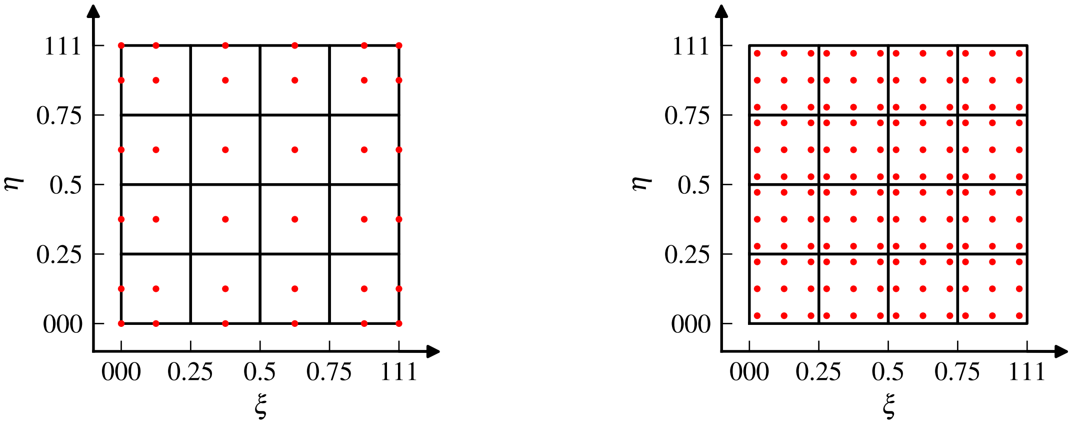
Figure 2.5: Projection step from gauss points (figure a) to control points (figure b) and evaluation of the solution on a locally refined physical mesh (figure c and d). ● represents the values solved at the control points (figure a and b) and plotted values on the physical domain (figure c and d). × represents the values solved at the Gauss points. ξ and η are paremetric variables spanning the knot vectors.

Algorithm 2 IGA Solution Output

```
subroutine postprocessing
Project variables from Gauss points to control points                    Eq. (2.20)
for iel = 1, number of elements do
    for j = 1, number of plotted values per element do
        get the solution ($u$, $T$, $\sigma$ and $\gamma_p$) in the physical domain    Eq. (2.21)
    end for
    Get the physical mesh connectivity for the plotted values in this element
end for
write plotted solution and associated connectivity to a file (.vtu for ParaView)
Return
```

at each Newton iteration. The coupling between different nodes are also leading to larger bandwidth of the jacobian that results in increasing computations and memory usage. In addition, some post-processing operations also require additional computational work, however, it is usually negligible compared to the cost of the linear solver.

We compare the relative cost of *k-refinement* used for the stresses in element HNSQ2 and gauss point sampling in element INSQ. In the example given in Figure 2.6, for a 4×4 mesh, the number of nodes in the case of NURBS elements is 49. However, in the case of gauss point sampling the number of nodes in the mesh is 144. Denoting n as the number of



(a) HNSQ2 element

(b) INSQ element

Figure 2.6: Stress discretization in HNSQ2 and INSQ elements. ● shows the location of stress degrees of freedom. ξ and η are paremetric variables spanning the knot vectors.

element per mesh direction and p as the order of the element, the total number of nodes in the mesh (N_k for element HNSQ2 and N_gp for element INSQ) is

$$N_k = (n + p + 1)^2 = n^2 + 2(p + 1)n + (p + 1)^2 \qquad N_{gp} = n^2 \times (p + 1)^2. \tag{2.22}$$

This clearly shows that the number of nodes used by HNSQ2 elements grows much slower compared to the number of nodes used by INSQ element upon mesh refinement (when n increases). Moreover, the sparsity pattern for the HNSQ2 element is worse than that of the INSQ element as can be seen in Figure 2.7. Also the size of the jacobian for HNSQ2 is $1,872 \times 1872$ and the number of non-zero entries in it is $66,547$ whereas the size of the jacobian for INSQ is $4,896 \times 4,896$ for $58,608$ non-zero entries in it. In other words, even though the linear system size due to HNSQ2 is smaller than INSQ, it also much denser and hence more memory is needed to store the jacobian in case of the HNSQ2 mesh. Furthermore due to the larger bandwidth of HNSQ2 jacobian the linear solves at every Newton iteration require more memory and more CPU time. Detailed studies on cpu time and memory usage for the HNSQ2 element can be found in Figures 2.13e and 2.13f.

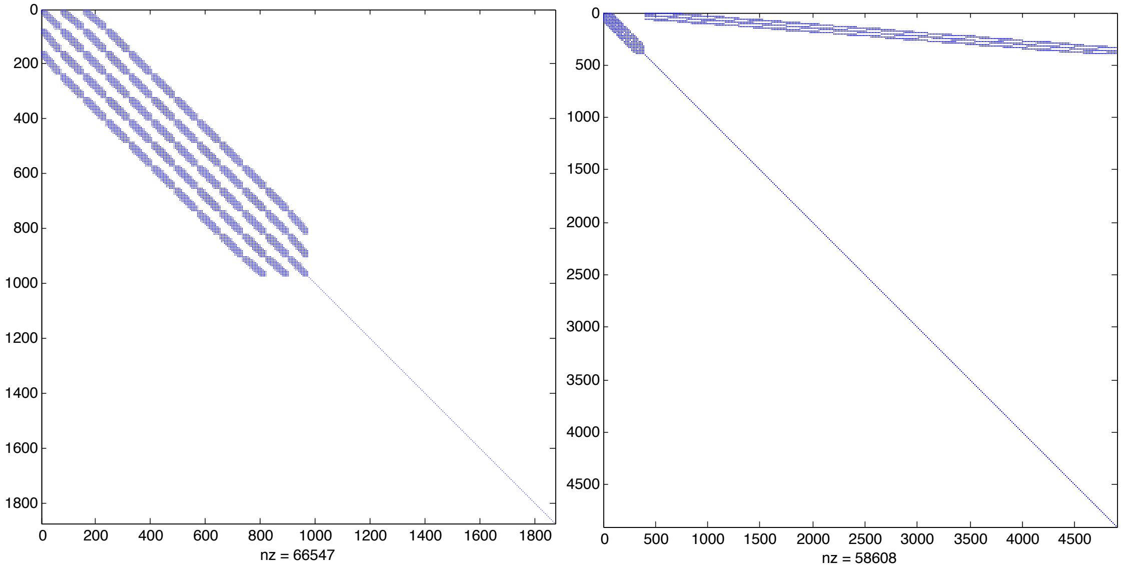


Figure 2.7: Sparsity pattern for the HNSQ2 (left) and INSQ (right) elements for a ten by ten mesh with quadratic elements.

2.2 Numerical examples

In this section we study the performance of the elements on two benchmark examples:

- A pure tension test on a quarter of a plate with a central imperfection
- A shear test on a regular plate (no imperfection added)

While all the elements presented in Table 2.4 have been tested, only the the HNSQ2 element that uses NURBS shape functions for all the fields except the EQPS field, is shown as it was found to have the best performances. This element provides good convergence results upon k-refinement but is also not subject to spurious oscillations in the EQPS field, thus providing a good compromise between performance and robustness.

The physical parameters used for all benchmark examples are the same and are summarized in Table 2.5.

Table 2.5: Material parameters used in the example problems

Physical quantity name	Name	Value	SI base unit
Young's modulus	E	200E9	$kg \cdot m^{-1} \cdot s^{-2}$
Poisson's ratio	ν	0.3	–
Mass density	ρ	7830	$kg \cdot m^{-3}$
Specific heat	c_p	448	$m^2 \cdot s^{-2} \cdot K^{-1}$
Thermal conductivity	κ	803.5	$kg \cdot m \cdot s^{-3} \cdot K$
Thermal softening parameter	δ	0.8	–
Thermal softening parameter	k	500	K
Reference temperature	T_0	293	K
Taylor-Quinney Coefficient	χ	0.9	–
Yield stress	σ_0	2E9	$kg \cdot m^{-1} \cdot s^{-2}$
Yield strain	γ_0	0.01	–
Reference strain rate	$\dot{\gamma}_0$	0.001	s^{-1}
Strain hardening exponent	n	0.01	–
Rate sensitivity parameter	m	70	–

On all benchmark simulations we also adopt an average acceleration, Newmark scheme (with parameters $\gamma = 0.25$ and $\beta = 0.5$, see Appendix A) and adaptive time stepping is used when the nonlinear solution is slow to converge. The NURBS meshes used for the simulations employ a single patch, and *k-refinement* is used in order to maintain full continuity throughout the meshes. As a point of comparison, we use the PSSQ element to test the performance of the proposed NURBS based element. Details on the implementation and characteristics of the PS element can be found in [59, 81, 82, 109], details on using the PSSQ element for the simulation of shear bands can be found in [73, 74].

In the following convergence analysis, we define the error norm as the relative distance in an Euclidian space between some reference solution and the current computed solution, which is expressed as follow:

$$e = \frac{||u_{ref} - u||_2}{||u_{ref}||_2} = \frac{\displaystyle\sum_{i=1}^{n}(u_{ref,i} - u_i)^2}{\displaystyle\sum_{i=1}^{n} u_{ref,i}^2}, \tag{2.23}$$

where u_{ref} is the solution obtained on the finest mesh (40 × 40) with the highest order element (P^4) and n is the number of nodes on our coarsest mesh (10 × 10). The error terms for a specific solution field are computed at nodes which are shared by coarse and fine meshes, as illustrated by the red circles in Figure 2.8. This also ensures that all the values used to compute the error on each mesh are actually values which have been solved for and not interpolated. This means that the error presented is least biased by the post-processing operations, and the number of terms in the error norm are determined by the coarsest grid.

On all the examples studied, the same structured discretization is employed, which is generated by a tensorial product type of construction. We use meshes with 10 by 10, 20 by 20, 30 by 30 and 40 by 40 elements to study the convergence with *h-refinement*. Convergence of the error is also investigated by *k-refinement* capabilities of the NURBS shape functions.

All the simulations in this paper are obtained using an implementation of the NURBS shape functions in FEAP [98] using PETSc [11] as a solver for the linearized Newton system. The results are then post-processed using ParaView [50] and particularly its Python scripting

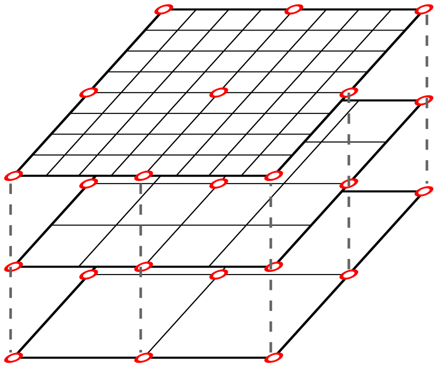


Figure 2.8: The error terms contributing to the error norm are computed at the nodes denoted by ●, which are shared on all hierarchy of grids.

capabilities (pvpython). Finally the graphs and plots are generated with the python library Matplotlib [57].

2.2.1 Plate with imperfection under pure tension

The first benchmark example is that of a plate with a central imperfection subjected to a uniform tension test. Since horizontal and vertical symmetry exists, only the top left quarter of the plate is considered as shown in Figure 2.9. A uniform velocity ramp is applied at the top of the plate: the velocity linearly increases from $0m/s$ to $5m/s$ in $1\mu s$ and remains constant at $5m/s$ after the ramp. With this first example two studies are done:

1. the sensitivity of the formulation to mesh density and mesh alignment is tested,

2. the convergence of the HNSQ2 is compared to the Pian and Sumihara element.

The total duration of the simulation is $2\mu s$ and the error is evaluated at time $t = 1.2\mu s$. In Figure 2.13e the convergence of the error on the displacement as a function of the CPU time is presented and Figure 2.13f shows the relation between error decrease and memory needed to solve the problem. The simulation time for this example ranged from 16.23 seconds for PSSQ on a 10 by 10 mesh to 6 hours 54 minutes and 1 second for the P^4 HNSQ2 element on a 40 by 40 mesh. The imperfection is induced by reducing the value of the yield stress

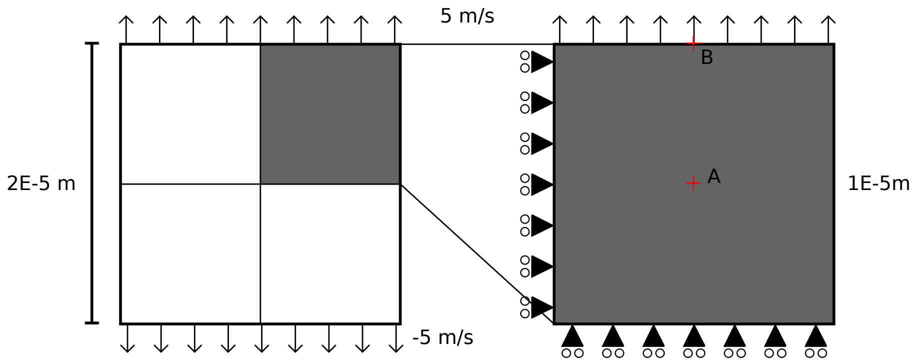


Figure 2.9: Representation of the geometry and boundary conditions used for example 1. Symmetry is assumed and only a quarter of the plate is modeled.

and yield strain using a 2D beta function given as

$$\beta_{2D}(X_1, X_2) = 1 - 0.04 \left[\operatorname{sech} \left(\frac{\sqrt{X_1^2 + X_2^2}}{5 \cdot 10^{-6}} \right) \right]^2, \tag{2.24}$$

$$\sigma_{yield} = \sigma_0 \beta_{2D}(X_1, X_2), \tag{2.25}$$

$$\gamma_{yield} = \gamma_0 \beta_{2D}(X_1, X_2). \tag{2.26}$$

This corresponds to a smooth imperfection centered on $x = 0$ and $y = 0$ with a maximal reduction of the reference yield parameters of 4% see Figure 2.10.

2.2.1.1 Mesh sensitivity study

We first study the sensitivity of the HNSQ2 element to the mesh. We consider the following mesh densities: 20x20, 30x30 and 40x40 and also three different element aspect ratios: 2, 1 and 0.5 using 40x20, 20x20 and 20x40 meshes. Figure 2.11 shows the 20x20 and 20x30 meshes as well as the line along which the EQPS field is plotted. The resulting EQPS curves are shown in Figure 2.12.

It can be seen that the EQPS converges well to the same solution on all meshes. This indicates that

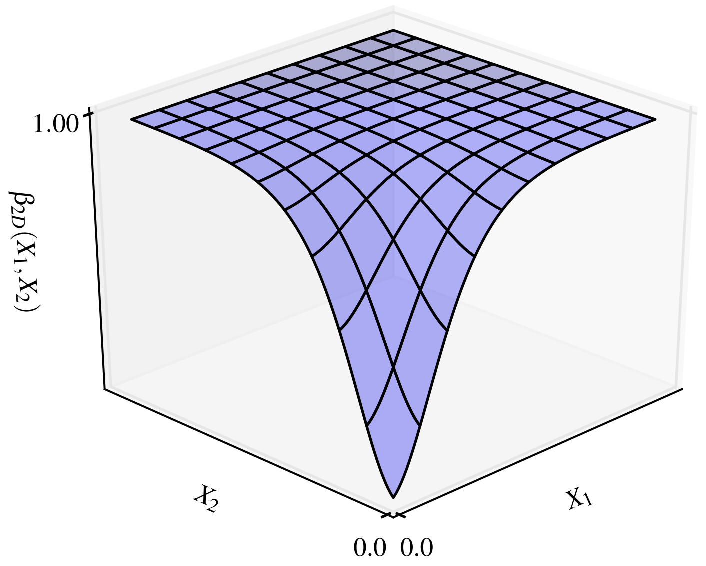


Figure 2.10: β_{2D} function defined by equation (2.24)

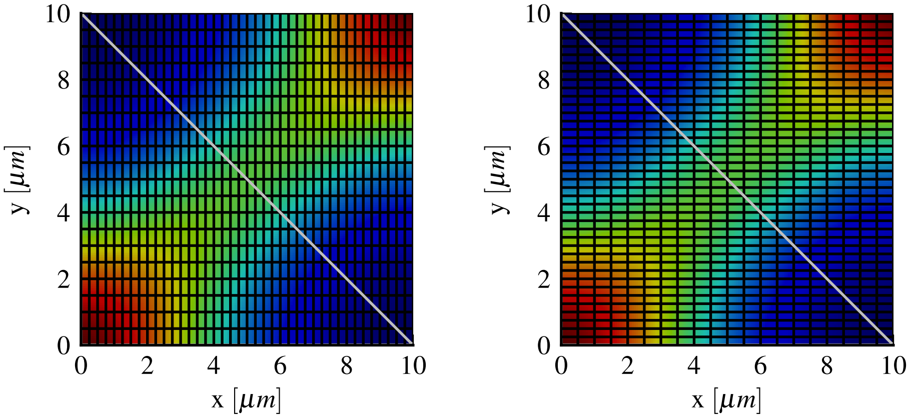


Figure 2.11: The 40x20 and 20x40 meshes used for this study are presented above, the grey line represents the arc along which the EQPS is represented in Figure 2.12. The meshes are plotted on top of the EQPS field computed at time $t = 1.25\mu s$

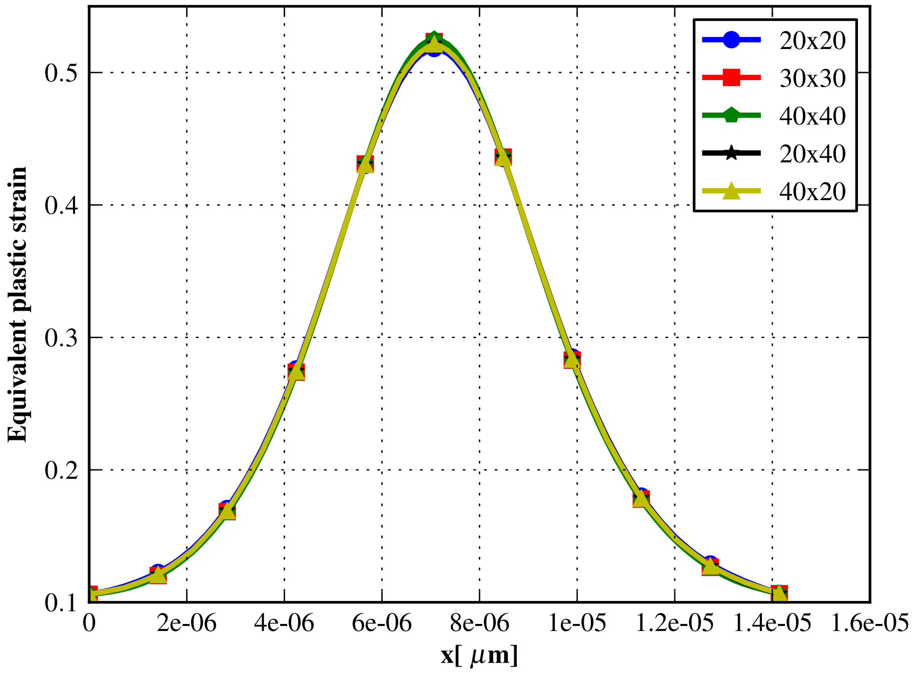
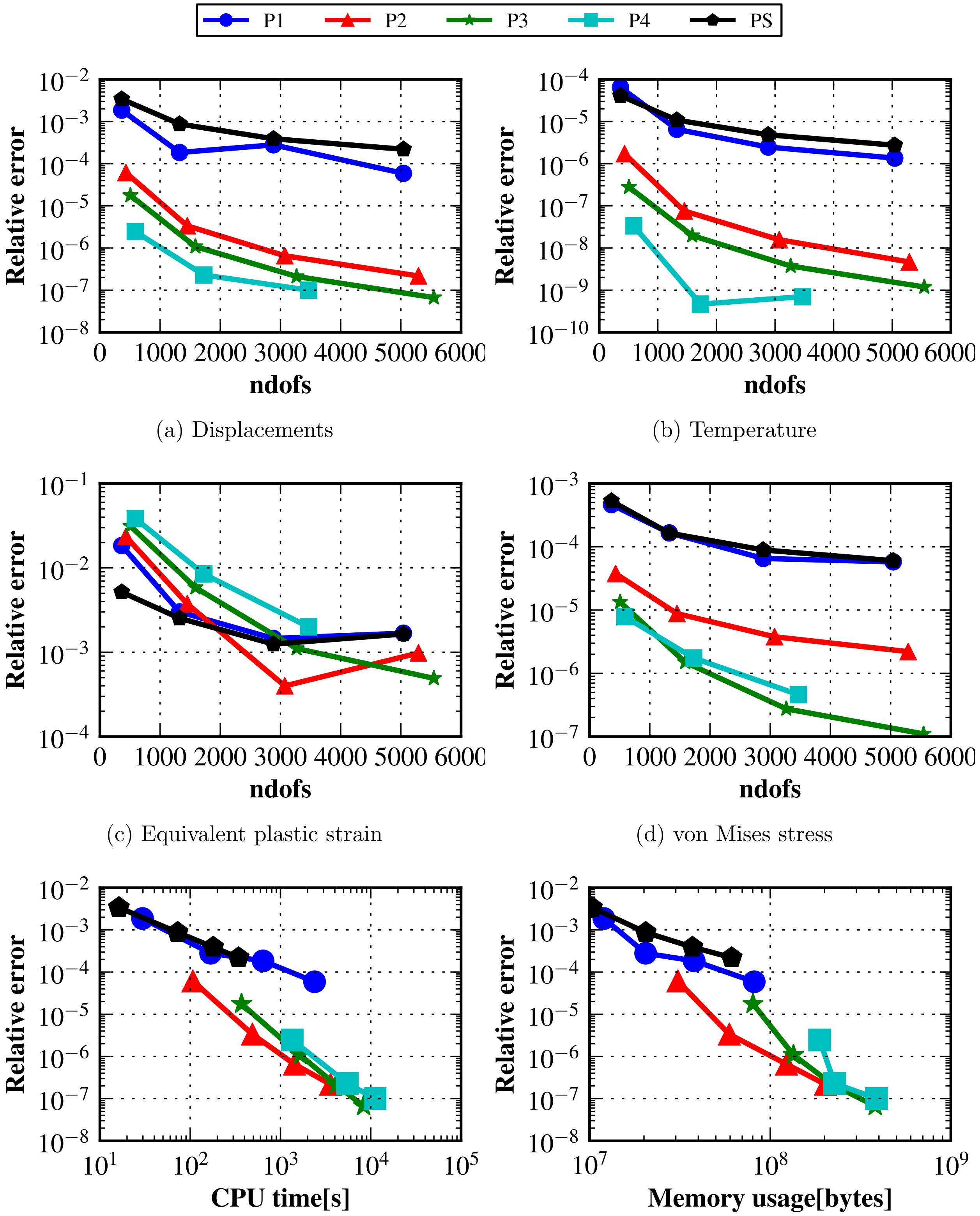


Figure 2.12: Plot of the EQPS for three different mesh densities as well as three different aspect ratios.

1. the size of the shear band does not depends on the size of the element used for the simulation,
2. the aspect ratio of the elements in the mesh does not affect the results, suggesting that shear band formation is not affected by mesh alignment.

These results are in agreement with those reported in [73] and show that inclusion of thermal conductivity together with a monolithic solver guarantee the accurate representation of the localization observed during the formation of shear bands. In addition, use of IGA suppresses mesh alignment sensitivity because of the nonlocality of the NURBS basis function. Nonlocality has also been found to suppress mesh alignment in mesh free methods [65, 67].



(a) Displacements

(b) Temperature

(c) Equivalent plastic strain

(d) von Mises stress

(e) Error against CPU usage

(f) Error against memory usage

Figure 2.13: Convergence of various solution fields the pure tension case in example 1, modeled by the HNSQ2 element with different order of NURBS basis function. CPU time and memory usage are also reported.

2.2.1.2 Convergence study

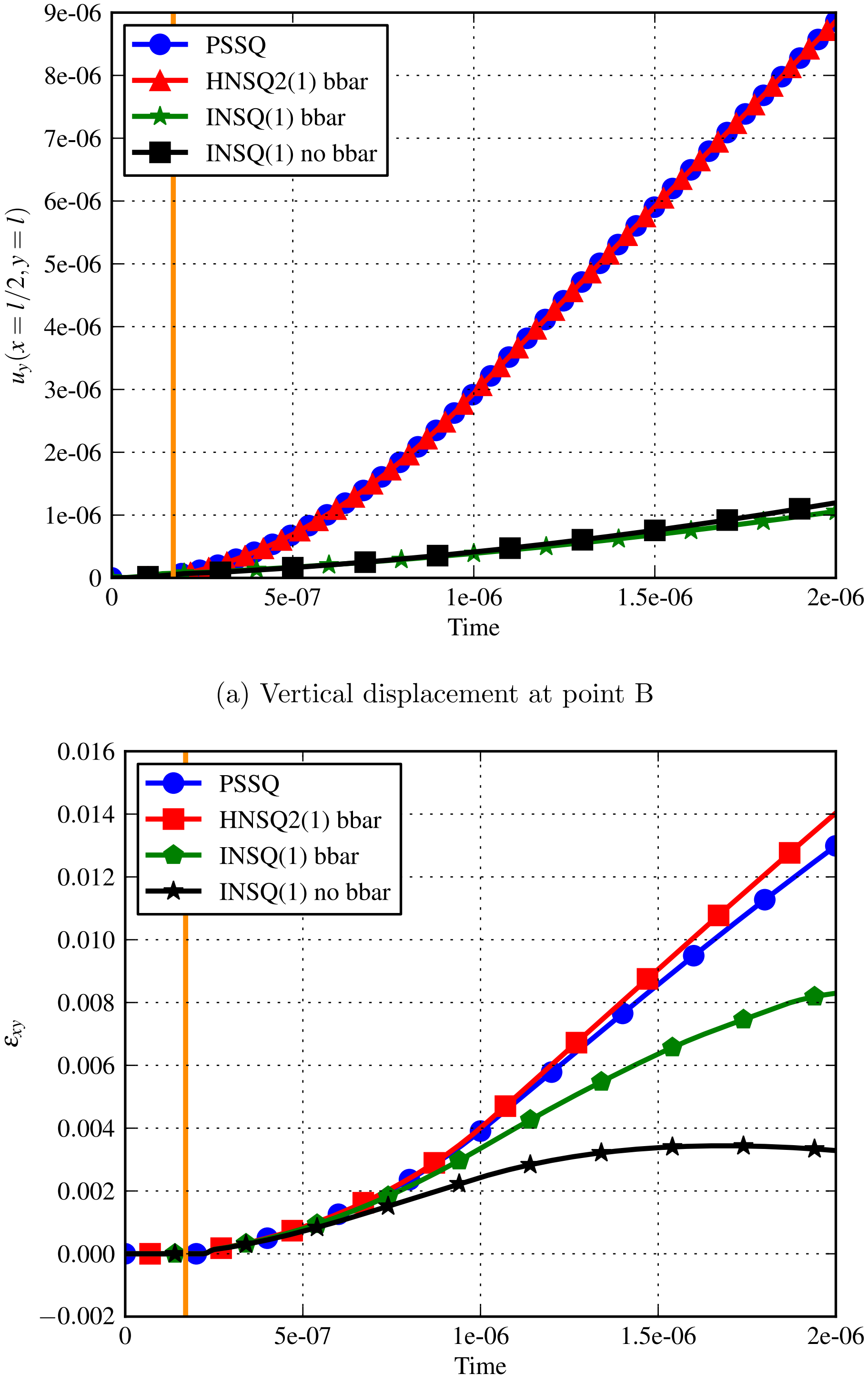
As expected, the convergence of the fields discretized with NURBS shape functions gives fast rates and smaller initial errors on coarse meshes see Figure 2.13a-2.13d as compared with the PSSQ element. Nonetheless, the convergence rate on the EQPS field for both elements is similar.

It can also be observed that the rate of increase of the computational time (cpu time) with respect to the size of the mesh is of the same order for all the elements. Thus to achieve similar accuracy the HNSQ2 element will converge in less time than the PSSQ element. Similarly Figure 2.13f shows that the HNSQ2 element requires less memory for a given accuracy than the memory required by the PSSQ element, especially if high order NURBS functions are used.

Finally, Figure 2.14 shows a comparison of the vertical displacement (u_y) and shear strain during the deformation to verify that the elements are not subject to volumetric locking. It can be observed that the PSSQ and HNSQ2 (with $\overline{\text{B}}$-bar) allow shear deformations to occur during the simulation whereas the INSQ element with $\overline{\text{B}}$-bar only allows for limited amount of shear deformation and vertical displacement. The INSQ element without $\overline{\text{B}}$-bar is severely resisting shear deformation. This confirms the need for the $\overline{\text{B}}$-bar method to overcome volumetric locking for coarse mesh with low order element.

2.2.2 Plate under shear loading

The second example examines a more complex experiment. A shear load applied at the top of the plate triggers the formation of a shear band at the left and right bottom corners of the plate, see Figure 2.15. In this configuration, the plastic strain develops in a much more localized region which develops into an arc type shear band. The equivalent plastic strain sustained by the plate at each corner is substantially higher than in the previous example and very steep gradients are observed around this shear band. No imperfection is modeled



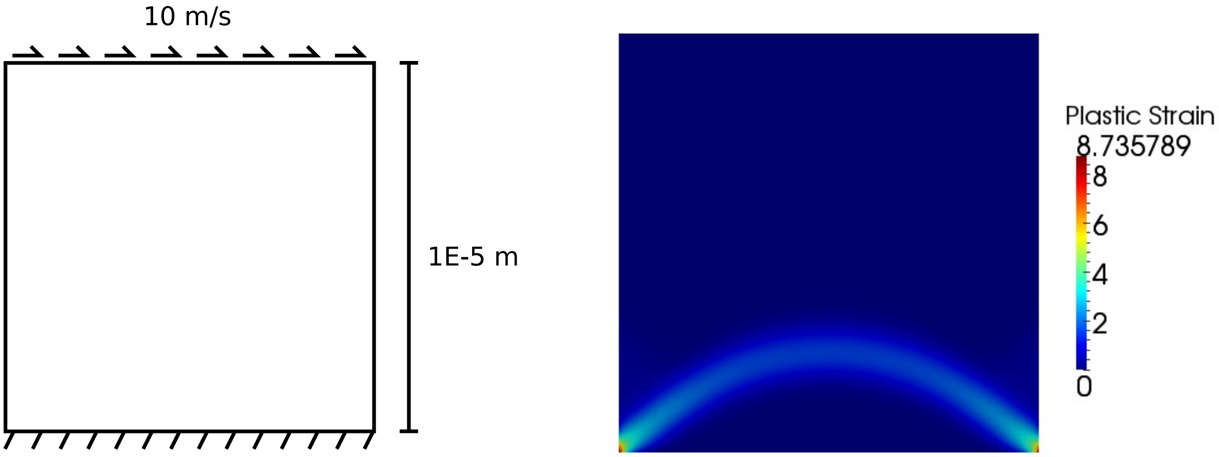
(a) Vertical displacement at point B

(b) shear strain at point A

Figure 2.14: Illustration of volumetric locking behaviour for different elements due to J2 plasticity in pure tension example 1. The orange line marks the onset of plasticity.

in this plate since the intense stress concentration observed at the bottom of the plate is sufficient to trigger the shear band.

The convergence rate of the HNSQ2 element is reported in Figure 2.16. The error norms



(a) Geometry and boundary conditions

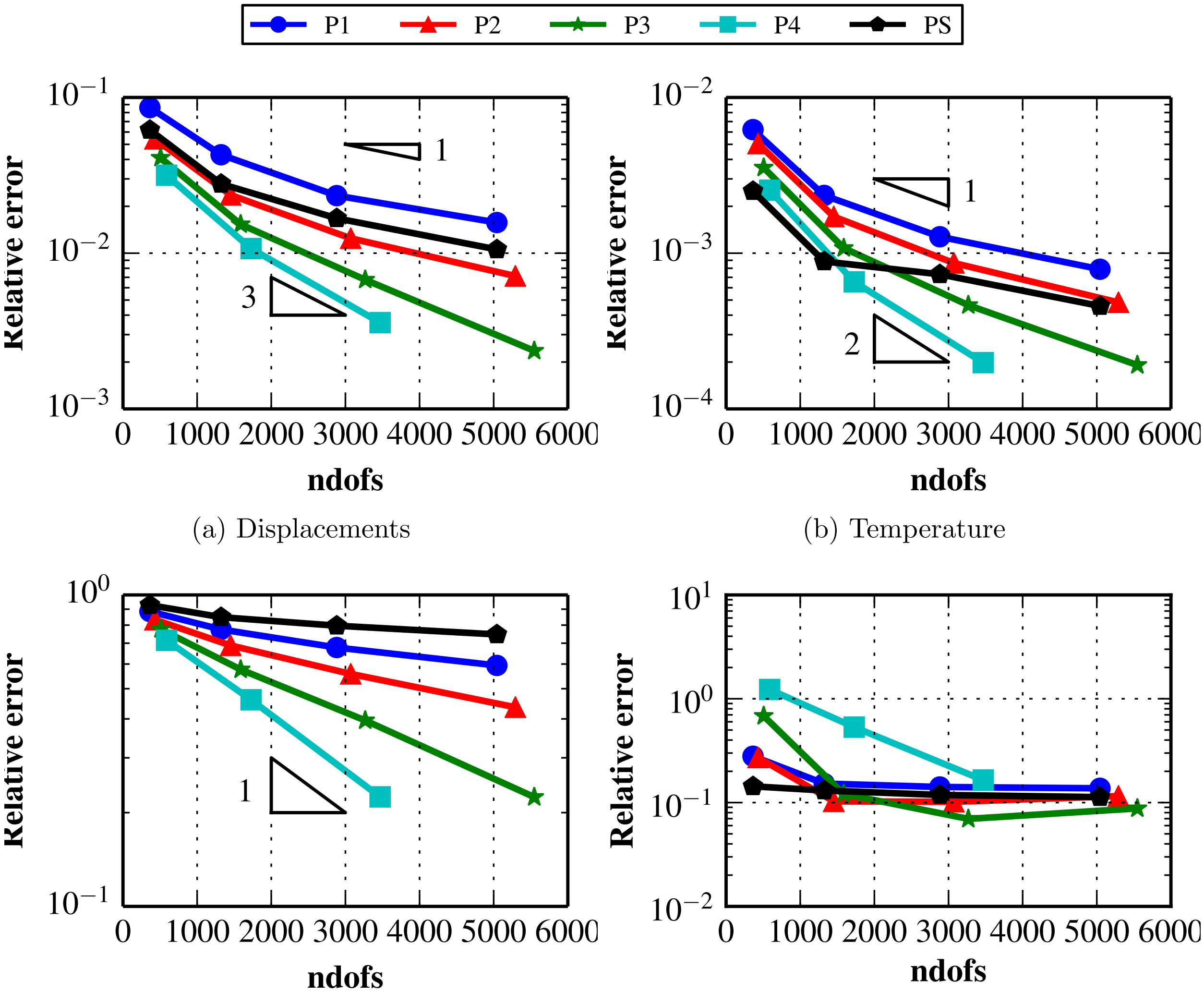
(b) Plastic strain at time $t = 2\mu s$

Figure 2.15: Plate under shear loading leading to the formation of an arc shaped shear band.

are computed at time $t = 0.8\mu s$ when the shear band is fully formed.

Figure 2.16 shows that the HNSQ2 element and the PSSQ element have similar rates of convergence for the displacement and temperature fields. However, the equivalent plastic strain computed with the HNSQ2 element are much more accurate and are providing interestingly faster rate of convergence than the PSSQ element. Thus the PSSQ element underestimates the plastic work which is occurring in the plate and this could lead to errors propagating to other solution fields. This especially affects the temperature field since the heat source in our formulation is the plastic work.

All elements listed in Tables 3 and 4 have been tested on this example problem, which presented convergence difficulties for some of the elements. For instance, the HNSQ1 element which employs NURBS shape functions for the equivalent plastic strain field did not converge on this problem due to oscillations in the equivalent plastic strain field as depicted in Figure 2.17. This element type, provides interestingly fast convergence rates for all fields



(a) Displacements

(b) Temperature

(c) Equivalent plastic strain

(d) von Mises stress

Figure 2.16: Convergence rate of the HNSQ2 element for the shearing example.

when used to model a plate under pure tension. In fact the rate of convergence for each field was found to increase with the order of the NURBS shape functions similarly to what can be observed when NURBS are used for linear problems (see [32]). Such an element would greatly improve our ability to resolve shear bands on relatively coarse mesh and therefore reduce computational time and memory needs, compared to elements requiring more refined meshes.

The MNSQ element has also proven to be unstable for a plate under shear loading for the same reasons of spurious oscillations in the equivalent plastic strain field. These oscillations might be prevented by the implementation of a Streamline Upwind/Petrov-Galerkin (SUPG) type of method [25, 56] which we intend to explore in future work.

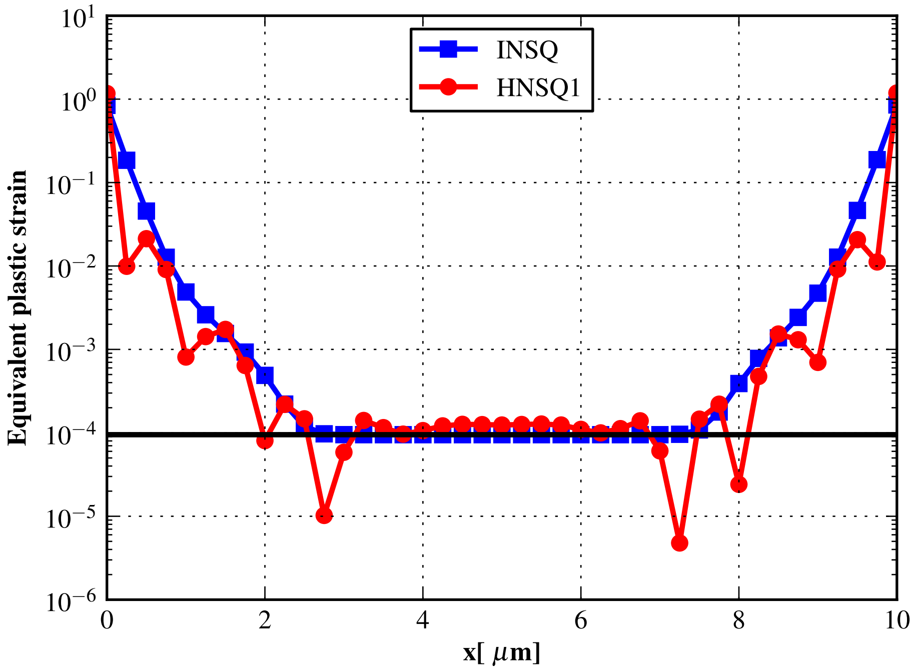


Figure 2.17: Oscillations observed in the equivalent plastic strain field along the bottom edge of the plate. The black line indicate the value of the zero base line for the equivalent plastic strain. This value is non zero since the oscillations are creating negative values which cannot be plotted on a semi-log scale.

2.3 Summary

IGA with high order NURBS elements is employed to discretize system (1.12) and the solution at every time step is obtained with a monolithic Newton type solver. This approach is shown to be insensitive to mesh refinement and mesh alignment and is therefore an attractive scheme for shear band modeling. Moreover, the results demonstrate that the use of higher order NURBS shape functions for the four field shear band problem significantly improve the convergence rate over the low order Pian–Sumihara element used in previous work on shear bands. The advantage of higher order NURBS becomes particularly apparent when CPU time and memory usage are accounted for. Several combinations of shape functions were used for the four fields, with varying degrees of success. Certain types of elements were found to be prone to oscillation instabilities in the equivalent plastic strain field (MNSQ and HNSQ1), which is unphysical and fatal to the simulation.

The most successful element was the HNSQ2 element, which was tested on two examples with different types of loading conditions. This element was found to behave efficiently in all cases compared to the reference Pian–Sumihara element, indicating that it is a well suited alternative for the simulation of shear bands. This becomes especially pronounced when the advantages of k-refinement is used to model shear bands on coarse meshes.

Chapter 3

Schur based fieldsplit preconditioners

In order to solve efficiently monolithic schemes, efficient and robust preconditioners for iterative Krylov-type solvers such as GMRES [90] or BICGStab [102] are required. The typical black box algebraic preconditioners such as Jacobi, ILU or algebraic multigrid [46, 100] have difficulties accelerating the solution of (1.12) due to the highly nonlinear multiphysics present, and the strong coupling of the equations.

Many preconditioners for multiphysics-type problems have been proposed. For example Ipsen [58] proposed a general method based on a Schur complement to precondition nonsymmetric matrices and his work led to the widely used block Schur preconditioners for coupled physics problems [54]. Other block and non-block preconditioners have been successfully applied to many physical problems such as fluids flow [33, 37, 41], solid mechanics [1, 87], magnetohydrodynamic [34] and mechanics [28].

In solid mechanics problems involving plasticity, the Jacobian varies greatly with the state of the system, and thus even if a preconditioner is effective at one time step, it may not be so at another time step. Therefore a good preconditioner for such a system needs to take advantage of the properties of the physics in order to offer a significant speed up to the Krylov solver.

In this chapter two parallel preconditioners to GMRES that take advantage of the specific discretization and physics of the problem are investigated. Both methods are based on a Conservation laws-Constitutive laws partition of the Jacobian and an outer Schur complement method. Then each method approximates the Schur complement term differently. The first approach is based off a multiplicative Schwarz method and is denoted by a Schur-Schwarz preconditioner. The second employs another inner Schur complement and is denoted as Schur-Schur preconditioner. Both preconditioners converge well in serial and parallel and are found to be robust for solving the shear band problem throughout all three stages of the deformation. In contrast, we show that directly preconditioning the Jacobian system or the outer Schur complement by off the shelf methods, such as an ILU(0) method would either diverge or be significantly suboptimal compared to the two proposed preconditioners. The strong and weak scaling of both proposed preconditioners is shown to be better than that of parallel state of practice LU direct solvers. The Schur-Schwarz preconditioner having the best strong scalability and the Schur-Schur the best weak scalability as well as better behavior on higher order isogeometric discretizations.

3.1 Schur complement based preconditioners

To maintain the strong coupling between the different physical fields involved in system (1.12) and at the same time use efficient solvers for the different field blocks, a Schur complement method is used to precondition the GMRES solver. In the following sections we first briefly review the Schur complement and multiplicative Schwarz methods. Next we present a special partition of the Jacobian in (1.40), which is used together with an outer Schur complement to derive the proposed preconditioners. The two proposed preconditioners are: (i) a Schwarz type scheme applied to the outer approximated Schur complement matrix and (ii) another inner Schur complement applied to a similar approximate of an outer Schur. For comparison, we also discuss an ILU(0) applied to the Schur complement and used as preconditioner to

GMRES.

3.1.1 A brief review of Schur complement method

In the following two sections, the Schur complement method [91] is briefly introduced. Let us consider the following linear system

$$\mathbf{J}\delta\mathbf{x} = \mathbf{R}, \tag{3.1}$$

assuming that there is a partition of the unknowns $\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{x}_0 \\ \delta\mathbf{x}_1 \end{bmatrix}$ and equations $\mathbf{R} = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \end{bmatrix}$, it is convenient to rewrite (3.1) in the following form

$$\begin{bmatrix} \mathbf{J}_{00} & \mathbf{J}_{01} \\ \mathbf{J}_{10} & \mathbf{J}_{11} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_0 \\ \delta\mathbf{x}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \end{bmatrix}. \tag{3.2}$$

Furthermore, let us assume that the linear system is solved by first preconditioning the system with a left preconditioner. Hence, multiplying both sides by a preconditioner (matrix or linear operator) $\mathbf{P}^{-1}$, yields

$$\mathbf{J}\delta\mathbf{x} = \mathbf{R} \quad \Rightarrow \quad \mathbf{P}^{-1}\mathbf{J}\delta\mathbf{x} = \mathbf{P}^{-1}\mathbf{R}. \tag{3.3}$$

Two main approaches to obtain a reliable preconditioner are as follows

1. *Split Preconditioners:* some blocks of $\mathbf{J}$ are simplified or assumed to be zero in the computation of $\mathbf{P}$ [18]

2. *Approximate Commuter Preconditioners:* some blocks of $\mathbf{J}$ are inverted using inexpensive and fast inexact algorithms [40].

The two approaches are often combined in an effort to achieve the best balance between fast, reliable and accurate preconditioners. Here we use both approaches at different stages of the proposed method. Moreover, it has been illustrated in the literature [89] that a robust strategy is to employ a GMRES solver as the iterative solver and apply these preconditioners to the Krylov space of GMRES to accelerate its convergence.

Assuming that the $\mathbf{J}_{00}$ block matrix in the partitioned system (3.2) is invertible, it is possible to write $\mathbf{J}$ as the product of three matrices, two of which are triangular with identity blocks on their diagonal and a block diagonal matrix, as follows

$$\begin{bmatrix} \mathbf{J}_{00} & \mathbf{J}_{01} \\ \mathbf{J}_{10} & \mathbf{J}_{11} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{J}_{10}\mathbf{J}_{00}^{-1} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{00} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{J}_{00}^{-1}\mathbf{J}_{01} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}, \tag{3.4}$$

where

$$\mathbf{S} = \mathbf{J}_{11} - \mathbf{J}_{10}\mathbf{J}_{00}^{-1}\mathbf{J}_{01} \tag{3.5}$$

is the lower Schur complement of (3.4). The alternate expression of $\mathbf{J}$ given in equation (3.4) is particularly useful to compute its inverse based on the inverses of $\mathbf{J}_{00}$ and $\mathbf{S}$:

$$\mathbf{J}^{-1} = \begin{bmatrix} \mathbf{1} & -\mathbf{J}_{00}^{-1}\mathbf{J}_{01} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{00}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ -\mathbf{J}_{10}\mathbf{J}_{00}^{-1} & \mathbf{1} \end{bmatrix}. \tag{3.6}$$

This expression of the inverse of $\mathbf{J}$ shows that if $\mathbf{J}_{00}$ can be easily inverted then the computational effort can be concentrated on computing the inverse of $\mathbf{S}$. Since the cost of forming and inverting $\mathbf{S}$ is large, in practice we only compute an approximate inverse of $\mathbf{S}$ that we denote by $\mathbf{Q} \approx \mathbf{S}^{-1}$ and use it to form a preconditioner for $\mathbf{J}$. That is,

$$\begin{aligned} \mathbf{J}^{-1} \approx \mathbf{P}^{-1} &= \begin{bmatrix} \mathbf{1} & -\mathbf{J}_{00}^{-1}\mathbf{J}_{01} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{00}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ -\mathbf{J}_{10}\mathbf{J}_{00}^{-1} & \mathbf{1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{J}_{00}^{-1} + \mathbf{J}_{00}^{-1}\mathbf{J}_{01}\mathbf{Q}\mathbf{J}_{10}\mathbf{J}_{00}^{-1} & -\mathbf{J}_{00}^{-1}\mathbf{J}_{01}\mathbf{Q} \\ -\mathbf{Q}\mathbf{J}_{10}\mathbf{J}_{00}^{-1} & \mathbf{Q} \end{bmatrix}. \end{aligned} \tag{3.7}$$

Two ways to obtain reliable $\mathbf{Q}$ operators (Schur-Schwarz and Schur-Schur methods) are presented in the following sections.

3.1.2 Conservation laws-Constitutive laws split

To invert the Jacobian matrix $\mathbf{J}$ efficiently it is crucial that the block structure of $\mathbf{J}_{\eta\eta}$ is preserved. To this end we introduce the following split: the constitutive laws, grouped in the field denoted by η, are assigned to the partition indicated by subscript 0 and the conservations laws in field u and T are assigned to the partition denoted by subscript 1 in Eq. (3.2). This split is presented in a matrix form as follow

$$\mathbf{J} = \left[\begin{array}{c|c} \mathbf{J}_{00} & \mathbf{J}_{01} \\ \hline \mathbf{J}_{10} & \mathbf{J}_{11} \end{array}\right] = \left[\begin{array}{c|cc} \mathbf{J}_{\eta\eta} & \mathbf{J}_{\eta u} & \mathbf{J}_{\eta T} \\ \hline \mathbf{J}_{u\eta} & \mathbf{J}_{uu} & \mathbf{0} \\ \mathbf{J}_{T\eta} & \mathbf{0} & \mathbf{J}_{TT} \end{array}\right]. \tag{3.8}$$

The sparsity pattern of the corresponding matrix is shown in Figure 3.1 for a single quadratic INSQ element. Note that reordering of degrees of freedom has also been applied to matrix $\mathbf{J}$ to obtain Figure 3.1. It is clear from the figure that the sparse structure of $\mathbf{J}_{\eta\eta}$ is preserved by this split and can then be exploited within the Schur decomposition.

With this partition of the system, the schur complement, given in Eq.(3.5), that needs to be inverted to compute $\mathbf{J}^{-1}$ is

$$\begin{aligned} \mathbf{S} &= \mathbf{J}_{11} - \mathbf{J}_{10}\mathbf{J}_{00}^{-1}\mathbf{J}_{01} \\ &= \begin{bmatrix} \mathbf{J}_{uu} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{TT} \end{bmatrix} - \begin{bmatrix} \mathbf{J}_{u\eta} \\ \mathbf{J}_{T\eta} \end{bmatrix} \mathbf{J}_{\eta\eta}^{-1} \begin{bmatrix} \mathbf{J}_{\eta u} & \mathbf{J}_{\eta T} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{J}_{uu} - \mathbf{J}_{u\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{J}_{\eta u} & -\mathbf{J}_{u\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{J}_{\eta T} \\ -\mathbf{J}_{T\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{J}_{\eta u} & \mathbf{J}_{TT} - \mathbf{J}_{T\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{J}_{\eta T} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_{uu} & \mathbf{S}_{uT} \\ \mathbf{S}_{Tu} & \mathbf{S}_{TT} \end{bmatrix}. \end{aligned} \tag{3.9}$$

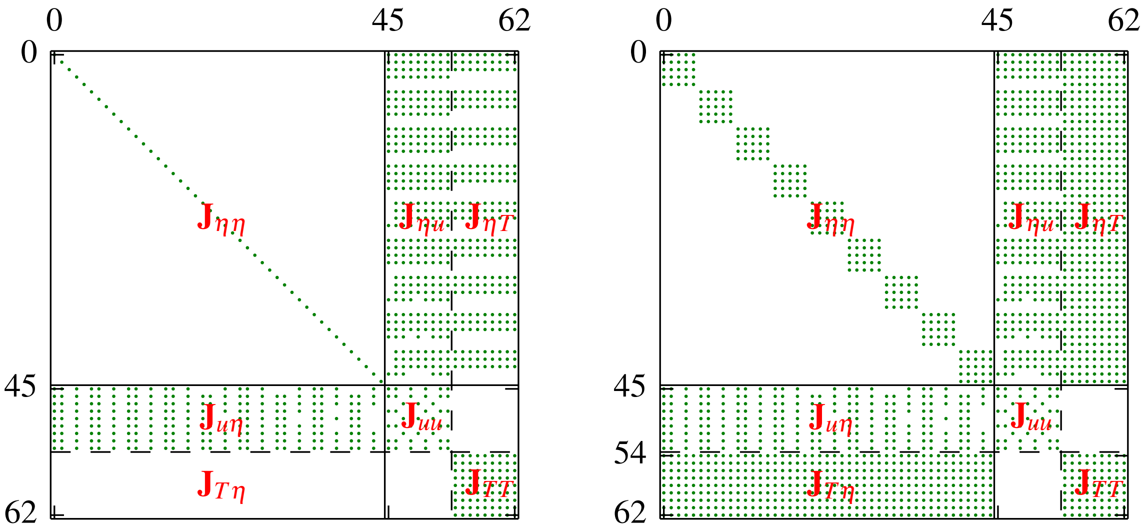


Figure 3.1: Sparsity pattern of the Jacobian matrix in the linear-elastic (left) and nonlinear plastic (right) regime for a single quadratic order INSQ element with 9 gauss points. The green dots indicate nonzero terms in the matrix. The solid black lines show the split of the displacement-temperature and stresses-EQPS nested blocks in the matrix, the dashed lines show the subpartition into displacement and temperature blocks.

Notice that the last term is used to simplify the notation of the schur complement. Based on equations (3.15) and (3.8) we obtain the following exact expression for the inverse of $\mathbf{J}$

$$\mathbf{J}^{-1} = \begin{bmatrix} \mathbf{1} & -\mathbf{J}_{\eta\eta}^{-1}\begin{bmatrix}\mathbf{J}_{\eta u} & \mathbf{J}_{\eta T}\end{bmatrix} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{\eta\eta}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ -\begin{bmatrix}\mathbf{J}_{u\eta} \\ \mathbf{J}_{T\eta}\end{bmatrix}\mathbf{J}_{\eta\eta}^{-1} & \mathbf{1} \end{bmatrix} \tag{3.10}$$

At this point the only work left is to propose an efficient and reliable strategy to compute $\mathbf{J}_{\eta\eta}^{-1}$ and $\mathbf{S}^{-1}$. Also, note that if a preconditioner is applied to a Krylov-type solver (e.g. GMRES), in order to accelerate its convergence one has to solve approximately an outer Schur complement system as follows:

$$\mathbf{S}\mathbf{y} = \mathbf{z} \tag{3.11}$$

where $\mathbf{z}$ is an incoming component of a Krylov vector and $\mathbf{y}$ is the preconditioned out-going vector. Furthermore, solving approximately Eq. (3.11) can also be accelerated by a preconditioning scheme on its own, as follows

$$\bar{\mathbf{P}}^{-1}\mathbf{S}\mathbf{y} = \bar{\mathbf{P}}^{-1}\mathbf{z} \tag{3.12}$$

where $\bar{\mathbf{P}}^{-1}$ is a preconditioner used for the schur complement sub problem. However, two difficulties arise when attempting to solve system (3.11) or (3.12): (i) $\mathbf{S}$ itself is computationally expensive to be formed explicitly and (ii) since $\mathbf{S}$ cannot be formed exactly, $\bar{\mathbf{P}}$ cannot be directly obtained from $\mathbf{S}$ which limits the choice of available preconditioners for the Schur complement problem.

To this end, in order to resolve difficulty (ii) and form a preconditioner, we approximate $\mathbf{S}$ by $\mathbf{S}^*$ that has the following expression

$$\begin{aligned}\mathbf{S} \approx \mathbf{S}^* &= \mathbf{J}_{11} - \mathbf{J}_{10}\left[\mathrm{diag}(\mathbf{J}_{00})\right]^{-1}\mathbf{J}_{01} \\ &= \begin{bmatrix} \mathbf{J}_{uu} & \mathbf{J}_{uT} \\ \mathbf{J}_{Tu} & \mathbf{J}_{TT} \end{bmatrix} - \begin{bmatrix} \mathbf{J}_{u\eta} \\ \mathbf{J}_{T\eta} \end{bmatrix} \left[\mathrm{diag}(\mathbf{J}_{\eta\eta})\right]^{-1} \begin{bmatrix} \mathbf{J}_{\eta u} & \mathbf{J}_{\eta T} \end{bmatrix}\end{aligned} \tag{3.13}$$

where the diagonal part of $\mathbf{J}_{\eta\eta}$ is employed and $\mathbf{S}^*$ can be easily formed. Now the preconditioner $\bar{\mathbf{P}}^{-1}$ is formed based on $\mathbf{S}^*$ and the preconditioned Schur complement system is written as

$$\left[\bar{\mathbf{P}}\left(\mathbf{S}^*\right)\right]^{-1}\mathbf{S}\mathbf{y} = \left[\bar{\mathbf{P}}\left(\mathbf{S}^*\right)\right]^{-1}\mathbf{z}. \tag{3.14}$$

We emphasize that Eq. (3.14) need only be solved approximately, for example, a few inner iterations using Richardson's method may suffice. Hence, an optimal performance can be achieved by considering the optimal number of inner and outer iterations which would lead to a minimal solution wall time.

With this notation we introduce a general class of preconditioners for system (1.12), that is written for an abstract type preconditioning operator $\mathbf{Q}$ used to approximate the Schur complement

$$\begin{aligned}
\mathbf{J}^{-1} \approx \mathbf{P}^{-1} &= \begin{bmatrix} \mathbf{1} & -\mathbf{J}_{\eta\eta}^{-1}\begin{bmatrix}\mathbf{J}_{\eta u} & \mathbf{J}_{\eta T}\end{bmatrix} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{\eta\eta}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ -\begin{bmatrix}\mathbf{J}_{u\eta} \\ \mathbf{J}_{T\eta}\end{bmatrix}\mathbf{J}_{\eta\eta}^{-1} & \mathbf{1} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{J}_{\eta\eta}^{-1} + \mathbf{J}_{\eta\eta}^{-1}\begin{bmatrix}\mathbf{J}_{\eta u} & \mathbf{J}_{\eta T}\end{bmatrix}\mathbf{Q}\begin{bmatrix}\mathbf{J}_{u\eta} \\ \mathbf{J}_{T\eta}\end{bmatrix}\mathbf{J}_{\eta\eta}^{-1} & -\mathbf{J}_{\eta\eta}^{-1}\begin{bmatrix}\mathbf{J}_{\eta u} & \mathbf{J}_{\eta T}\end{bmatrix}\mathbf{Q} \\ -\mathbf{Q}\begin{bmatrix}\mathbf{J}_{u\eta} \\ \mathbf{J}_{T\eta}\end{bmatrix}\mathbf{J}_{\eta\eta}^{-1} & \mathbf{Q} \end{bmatrix}.
\end{aligned} \tag{3.15}$$

Note that in this case, $\mathbf{Q}$ is an abstract operator corresponding to the approximate solution of Eq. (**??**) which is an approximation of $\mathbf{S}^{-1}$.

3.1.3 The Schur-Schwarz and Schur-Schur preconditioners

The first straightforward option to obtain $\mathbf{Q}$ is by performing a few iterations of a Krylov-type method such as GMRES with an ILU(0) preconditioner applied to Eq. (**??**). In this case $\bar{\mathbf{P}}_{ILU}$ is the ILU(0) preconditioner obtained from $\mathbf{S}^*$. While this is an obvious choice, it is shown later that it is not a good one. The systems arising in the shear band problem have strong coupling between components, in particular when plasticity initiates and the shear band starts to form, and hence such preconditioners yields poor performance.

A better choice for $\mathbf{Q}$ in Eq. (3.15) is by a multiplicative Schwarz method [92]. See **??** for a detailed description of the method. In this approach, one defines first the following block diagonal matrices

$$\mathbf{B}_0 = \begin{bmatrix} \mathbf{S}_{uu}^{*-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \quad \text{and} \quad \mathbf{B}_1 = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{TT}^{*-1} \end{bmatrix}, \tag{3.16}$$

then the preconditioner is obtained by setting

$$\begin{aligned}\bar{\mathbf{P}}_{sz}^{-1} &= \mathbf{B}_0 + \mathbf{B}_1 - \mathbf{B}_1\mathbf{S}^*\mathbf{B}_0 \\ &= \begin{bmatrix} \mathbf{S}_{uu}^{*-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} + \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{TT}^{*-1} \end{bmatrix} - \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{TT}^{*-1} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{uu}^{*} & \mathbf{S}_{uT}^{*} \\ \mathbf{S}_{Tu}^{*} & \mathbf{S}_{TT}^{*} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{uu}^{*-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_{uu}^{*-1} & -\mathbf{S}_{uT}^{*} \\ -\mathbf{S}_{TT}^{*-1}\mathbf{S}_{Tu}^{*}\mathbf{S}_{uu}^{*-1} & \mathbf{S}_{TT}^{*-1} \end{bmatrix}\end{aligned} \tag{3.17}$$

In this case we apply only one iteration of a Richardson method to Eq. (**??**), hence the overall action of $\mathbf{P}_{sz}^{-1}$ (in this case $\mathbf{Q}_{sz} = \bar{\mathbf{P}}_{sz}^{-1}$) can be written as

$$\mathbf{P}_{sz}^{-1} = \begin{bmatrix} \mathbf{1} & -\mathbf{J}_{\eta\eta}^{-1} \begin{bmatrix} \mathbf{J}_{\eta u} & \mathbf{J}_{\eta T} \end{bmatrix} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{\eta\eta}^{-1} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{P}}_{sz}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ -\begin{bmatrix} \mathbf{J}_{u\eta} \\ \mathbf{J}_{T\eta} \end{bmatrix} \mathbf{J}_{\eta\eta}^{-1} & \mathbf{1} \end{bmatrix}. \tag{3.18}$$

Finally the block $\mathbf{S}_{uu}^{*-1}$ and $\mathbf{S}_{TT}^{*-1}$ in $\bar{\mathbf{P}}_{sz}^{-1}$, are inverted using an LU direct solver and a fully converged GMRES solver preconditioned by ILU, respectively.

The second proposed preconditioner is the so called Schur-Schur preconditioner, denoted $\mathbf{P}_{sr}$. The key idea is to decompose the outer Schur sub problem, given in Eq. (3.11) with a second Schur complement method, written as

$$\mathbf{S}^* = \begin{bmatrix} \mathbf{S}_{uu}^{*} & \mathbf{S}_{uT}^{*} \\ \mathbf{S}_{Tu}^{*} & \mathbf{S}_{TT}^{*} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{S}_{Tu}^{*}\mathbf{S}_{uu}^{*-1} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{uu}^{*} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{inner} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{S}_{uu}^{*-1}\mathbf{S}_{uT}^{*} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}, \tag{3.19}$$

where $\mathbf{S}_{inner}$ is

$$\mathbf{S}_{inner} = \mathbf{S}_{TT}^{*} - \mathbf{S}_{Tu}^{*}\mathbf{S}_{uu}^{*-1}\mathbf{S}_{uT}^{*}. \tag{3.20}$$

Following Eq. (3.6) the exact inverse for the outer Schur complement reads

$$\mathbf{S}^{*-1} = \begin{bmatrix} \mathbf{1} & -\mathbf{S}_{uu}^{*-1}\mathbf{S}_{uT}^{*} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{uu}^{*-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{inner}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ -\mathbf{S}_{Tu}^{*}\mathbf{S}_{uu}^{*-1} & \mathbf{1} \end{bmatrix} \tag{3.21}$$

Hence, to form a Schur-Schur type preconditioner to $\mathbf{S}^{-1}$, we first need to obtain an approximation for the inner Schur block. To this end, the same approximation as the one in Eq. (3.13) is proposed

$$\mathbf{S}_{inner} \approx \mathbf{S}_{inner}^{*} = \mathbf{S}_{TT}^{*} - \mathbf{S}_{Tu}^{*} \left[\mathrm{diag}(\mathbf{S}_{uu}^{*})\right]^{-1} \mathbf{S}_{uT}^{*}. \tag{3.22}$$

The inner block $\mathbf{S}_{inner}^{*}$ is approximately inverted with an ILU(0) solver. This approximate solve is accurate enough due to the fact the $\mathbf{S}_{inner}^{*}$ is the result of two nested Schur complement and hence is much denser than a system arising from a usual finite element discretization. The approximate inverse of $\mathbf{S}_{inner}^{*}$ is denoted $\mathbf{S}_{inner}^{-1}$ in equation (**??**) eventhough it is not an exact inverse. In summary, this preconditioner includes two levels of Schur decomposition with a few approximations along the way. Hence we obtain the following abstract expression for the Schur-Schur preconditioner

$$
\begin{aligned}
\mathbf{P}_{sr}^{-1} &= \begin{bmatrix} \mathbf{1} & -\mathbf{J}_{\eta\eta}^{-1}\begin{bmatrix}\mathbf{J}_{\eta u} & \mathbf{J}_{\eta T}\end{bmatrix} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{\eta\eta}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{sr} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ -\begin{bmatrix}\mathbf{J}_{u\eta} \\ \mathbf{J}_{T\eta}\end{bmatrix}\mathbf{J}_{\eta\eta}^{-1} & \mathbf{1} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{1} & -\mathbf{J}_{\eta\eta}^{-1}\begin{bmatrix}\mathbf{J}_{\eta u} & \mathbf{J}_{\eta T}\end{bmatrix} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \\
&\times \begin{bmatrix} \mathbf{J}_{\eta\eta}^{-1} & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} \mathbf{S}_{uu}^{*-1} + \mathbf{S}_{uu}^{*-1}\mathbf{S}_{uT}^{*}\mathbf{S}_{inner}^{-1}\mathbf{S}_{Tu}^{*}\mathbf{S}_{uu}^{*-1} & -\mathbf{S}_{uu}^{*-1}\mathbf{S}_{uT}^{*}\mathbf{S}_{inner}^{-1} \\ -\mathbf{S}_{inner}^{-1}\mathbf{S}_{Tu}^{*-1}\mathbf{S}_{uu}^{*-1} & \mathbf{S}_{inner}^{-1} \end{bmatrix} \end{bmatrix} \\
&\times \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ -\begin{bmatrix}\mathbf{J}_{u\eta} \\ \mathbf{J}_{T\eta}\end{bmatrix}\mathbf{J}_{\eta\eta}^{-1} & \mathbf{1} \end{bmatrix}
\end{aligned} \tag{3.23}
$$

where again, $\mathbf{Q}$ denotes the two level preconditioner that is acting on the outer global system. As with the Schur-Schwarz preconditioner, the $\mathbf{S}_{uu}^{*-1}$ block is inverted using an LU solver.

To observe the effectiveness of the Schur-Schwarz and Schur-Schur preconditioners, the iteration matrix defined in equation (3.24), is computed for each of the three preconditioners and its eigenvalues are plotted and illustrated in Figure 3.2. Note that we have considered one INSQ element for this illustration.

$$
\mathbf{M}_{iter} = \mathbf{1} - \mathbf{P}^{-1}\mathbf{J}. \tag{3.24}
$$

As can be observed, most eigenvalues fall within the unit circle which ensures a rapid reduction of most components of the error. The eigenvalues that do not fall in the unit circle represent modes that are amplified by the application of $\mathbf{P}^{-1}$. As the simulation progresses and plasticity develops, the off diagonal terms of $\mathbf{J}_{00}$ become more prominent, and several modes are found to lie outside the unit circle starting at around $0.2\mu s$ of the simulation. The magnitude of these divergent modes increases later in the simulation as the plastic straining becomes more intense.

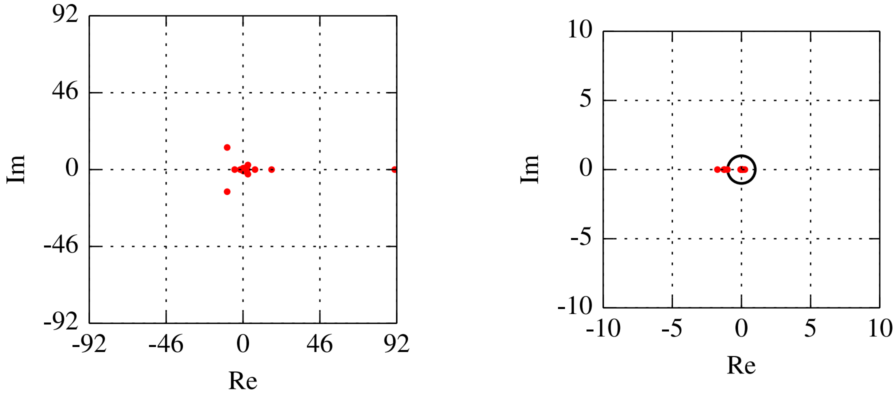


Figure 3.2: Eigenvalues of $\mathbf{P}_{sz}$ (left) and $\mathbf{P}_{sr}$ (right). Note that the scale of the axis is changing due to the large eigen values of $\mathbf{P}_{sz}$. $\mathbf{P}_{sz}$ has 14 eigen values larger than 1 and $\mathbf{P}_{sr}$ has 4 eigen values larger than 1. Note that the effectiveness of the preconditioner varies as the system evolves with time and the shear band is formed.

Complete descriptions of the Schur-Schwarz and Schur-Schur preconditioners with all block solvers and approximations specified are provided in the flow chart presented in Figure 3.3.

3.2 Numerical results

In this section we demonstrate the capabilities and performance of the proposed preconditioners on a 45^o shear band benchmark example. In the first section the parameters of the problem are presented, in the second and third sections two sets of numerical studies are presented:

- *Serial studies:* using IsoGeometric Analysis (NURBS type basis functions) [19] to observe the behavior of the preconditioners as the shear band initiates and propagates and the effect of k- and h-refinement on the convergence of the preconditioners,

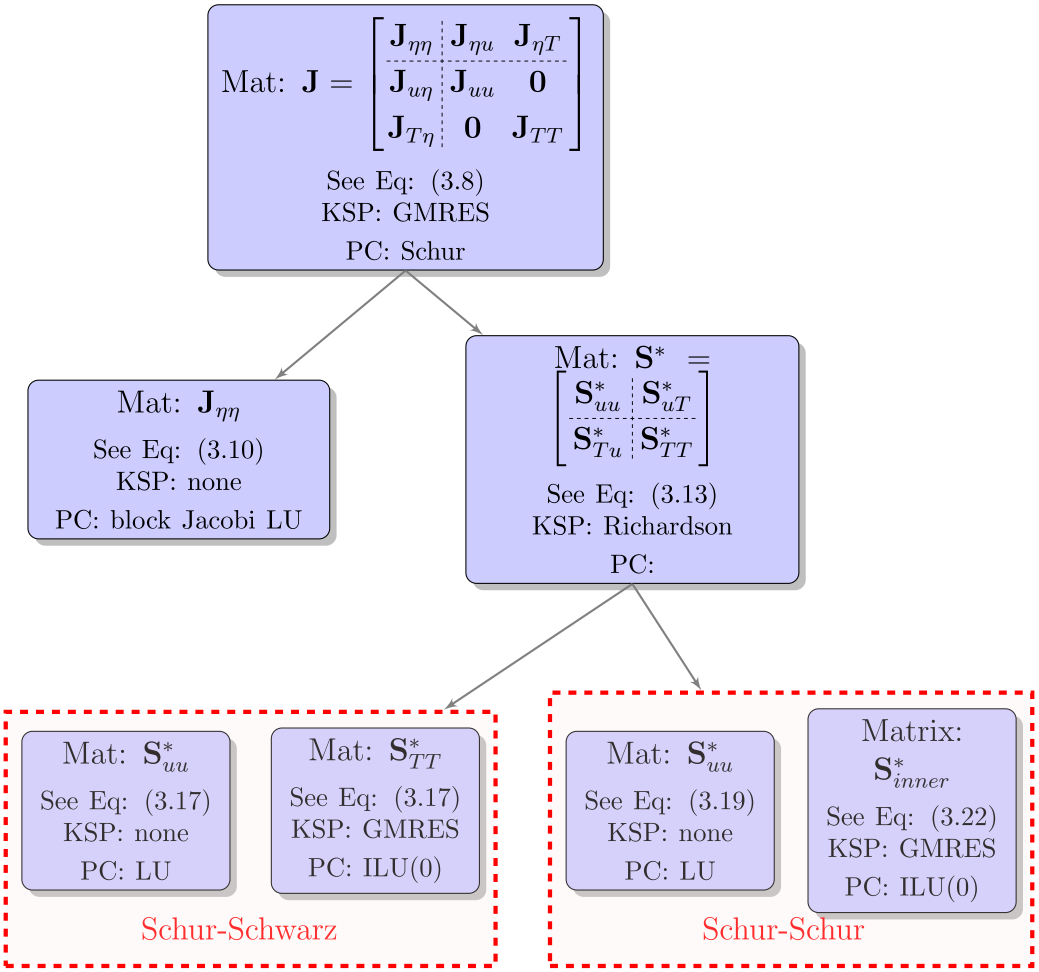

Mat: $\mathbf{J} = \left[\begin{array}{c:cc} \mathbf{J}_{\eta\eta} & \mathbf{J}_{\eta u} & \mathbf{J}_{\eta T} \\ \hdashline \mathbf{J}_{u\eta} & \mathbf{J}_{uu} & \mathbf{0} \\ \mathbf{J}_{T\eta} & \mathbf{0} & \mathbf{J}_{TT} \end{array}\right]$
See Eq: (3.8)
KSP: GMRES
PC: Schur
Mat: $\mathbf{J}_{\eta\eta}$
See Eq: (3.10)
KSP: none
PC: block Jacobi LU
Mat: $\mathbf{S}^* = \left[\begin{array}{c:c} \mathbf{S}^*_{uu} & \mathbf{S}^*_{uT} \\ \hdashline \mathbf{S}^*_{Tu} & \mathbf{S}^*_{TT} \end{array}\right]$
See Eq: (3.13)
KSP: Richardson
PC:
Mat: $\mathbf{S}^*_{uu}$
See Eq: (3.17)
KSP: none
PC: LU
Mat: $\mathbf{S}^*_{TT}$
See Eq: (3.17)
KSP: GMRES
PC: ILU(0)
Schur-Schwarz
Mat: $\mathbf{S}^*_{uu}$
See Eq: (3.19)
KSP: none
PC: LU
Matrix: $\mathbf{S}^*_{inner}$
See Eq: (3.22)
KSP: GMRES
PC: ILU(0)
Schur-Schur

Figure 3.3: Flow chart of the Schur-Schwarz and Schur-Schur preconditioners, Mat indicates the matrix being solved, KSP indicates which iterative solver is being used (if any) and PC indicates which preconditioner accelerates the solver (if direct solver then no KSP is used).

- *Parallel studies:* using standard irreducible bilinear elements [73,74,75] to observe the parallel performance of the preconditioners.

The first series of numerical examples, involve smaller meshes and high order discretizations, are run on a DELL optiplex 755 with 4 cores at 2.83GHz and 4GB of RAM. The second series of numerical examples are run using the MIRA cluster [42,78], which is part of the Argonne Leadership Computing Facility (ALCF), at Argonne National Labs. The problem is partitioned into processors using the library METIS/PARMETIS [61,62] and the PDE model is implemented using the Finite Element Analysis Program (FEAP) [98]. The linear solves are done by calling the FieldSplit option in PETSc [11,26,94] and using the external packages MUMPS [5] and Hypre Euclid [43]. The posprocessing and graphics are generated using ParaView [50] and Matplotlib [57].

3.2.1 The 45^o shear band problem

Consider a square plate under pure tension applied at the top and bottom edges. The plate has a circular imperfection at its center that triggers the localization of the deformation into two shear bands forming an X-shape passing to the four corners of the plate through its center. The imperfection is induced by reducing the value of the yield stress and yield strain using a 2D beta function given as

$$\sigma_{yield} = \sigma_0 \beta_{2D}(x,y), \qquad \gamma_{yield} = \gamma_0 \beta_{2D}(x,y) \tag{3.25}$$

with

$$\beta_{2D}(x,y) = 1 - 0.04 \left[\operatorname{sech}\left(\frac{\sqrt{x^2+y^2}}{5 \cdot 10^{-}6} \right) \right]^2 . \tag{3.26}$$

This corresponds to a smooth imperfection centered on $x = 0$ and $y = 0$ with a maximal reduction of the reference yield parameters of 4%, this function is plotted over a unit square in Figure 2.10.

Due to the symmetries of this problem only a quarter of the plate needs to be modeled, herein the top right quarter of the plate. The width of the quarter plate is $10\mu m$ and a uniform velocity v_{BC} is applied at its top edge. v_{BC} increases linearly from $v_{BC} = 0\ m/s$ to $v_{BC} = 5\ m/s$ from time $t = 0$ to time $t = 1\mu s$, v_{BC} remains constant afterward. A null thermal flux condition is applied to all the edges of the plate. The initial values of displacement, stresses and plastic strains are taken as zero and the initial temperature is set to $T = 293.0K$ (room temperature). The values of the material parameters used to run the simulations are presented in Table 2.5 in Appendix C. The profile of v_{BC} and a schematic representation of the plate are shown in Figure 3.4.

The stress-strain and stress-time curves, averaged over the plate, are recorded and shown in Figure 3.5. The vertical lines in the figure indicate the three stages of the deformation (linear elastic, onset of plasticity and stress collapse) for which the Jacobian matrix will change. The corresponding plastic strain field and the specific sparsity pattern of the Jacobian are shown in Figure 3.6. Note that the sparsity pattern changes with the plasticity in the system and the values in the matrix increase significantly, which indicate a high degree of nonlinearity and coupling between the physics.

In general the number of equations in the problem is given by

$$n_{eq} = 3 \times n_n + 5 \times n_{el} \times n_{gp} - n_{bc} \tag{3.27}$$

where n_n is the number of nodes in the problem, n_{el} is the number of elements in the problem, $n_{gp} = (p+1)^2$ is the number of gauss points per elements, p is the polynomial order of the shape functions and n_{bc} is the number of Dirichlet boundary conditions applied to the problem.

In the simulations presented in the results section we only consider uniform structured meshes with equal number of nodes in both x and y directions and Dirichlet boundary conditions are applied along three edges of the plate. This simplifies expression (3.27) to

$$n_{eq} = (5(p+1)^2 + 3)n^2 + 3n \tag{3.28}$$

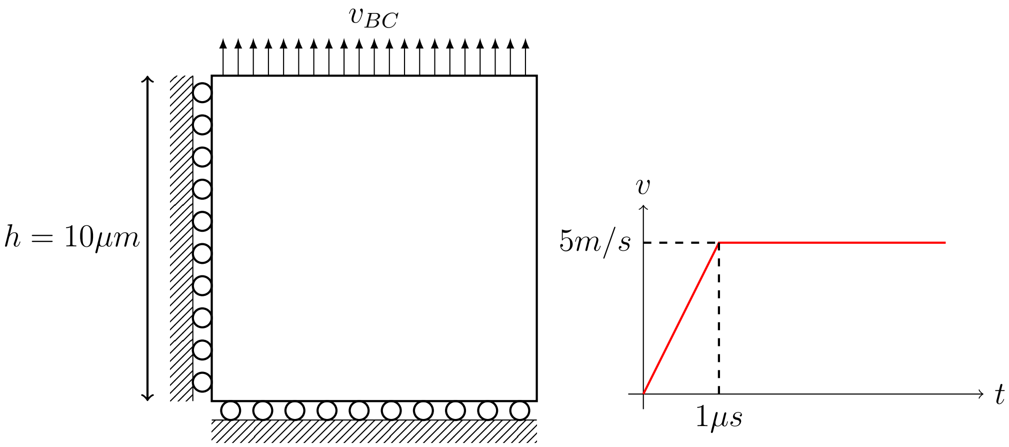


Figure 3.4: The modeled quarter plate with it boundary conditions and the evolution of the velocity v_{BC}.

where n is the number of elements per spatial direction.

The PDE system is integrated in time, where at every time step, a monolithic solution is obtained using a Newton method. Newton iterations are terminated under the following conditional norms:

- *Relative residual:* $\frac{||\mathbf{R}^{\alpha}_{\beta,n+1}||}{||\mathbf{R}_{\beta,n}||} \leq 10^{-4}$
- *Absolute residual:* $||\mathbf{R}^{\alpha}_{\beta,n+1}|| \leq 10^{-12}$
- *Relative solution increment:* $\frac{||\delta x^{\alpha}_{\beta,n+1}||}{||\delta x_{\beta,n}||} \leq 10^{-16}$

where β is the set of all multiphysics degrees of freedom, corresponding to $\forall\beta \in \{u, T, \sigma, \bar{\gamma}_p\}$. $n+1$ denotes the current time step and α denotes the current Newton iteration. These criterions ensure that a correct solution for each field is obtained at every time step.

The convergence criteria for the GMRES linear solver applied to the global system in Eq. (3.1) is

$$\frac{||\mathbf{R}_j||}{||\mathbf{R}_0||} \leq 10^{-8} \tag{3.29}$$

where j denotes the current linear solver iteration.

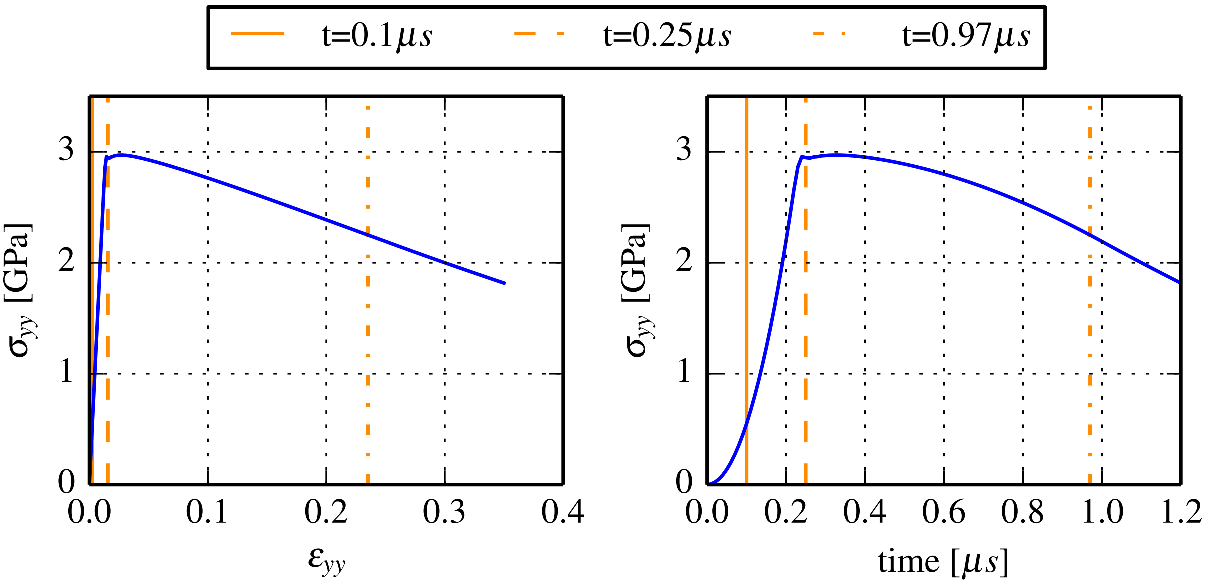
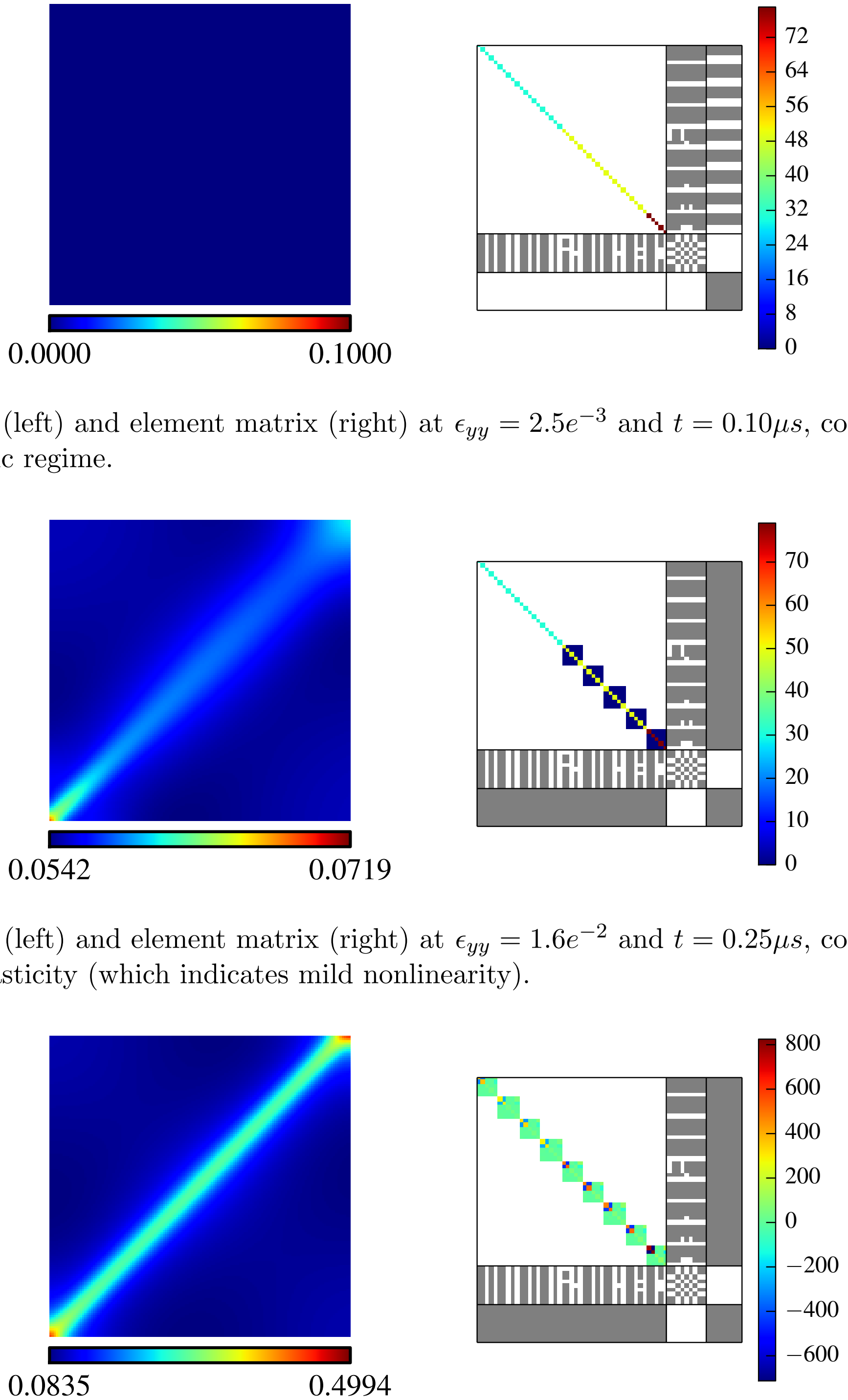


Figure 3.5: Stress-strain and stress-time curves for the 45^o shear band problem.

3.2.2 Serial performance of the preconditioners applied to Isogeometric discretizations

In this section the problem presented in Figure 3.4 is descretized using the INSQ element [19] that uses NURBS based shape functions for displacements and temperature fields, while the stresses and plastic strain are gauss point history variables. This element has been shown to have good convergence properties upon h-refinement and k-refinement but the latest leads to increased computational times with a classic direct LU solver. With NURBS shape functions of degree $p \geq 1$ the preconditioners that are proposed are facing a more complex task since the coupling in $\mathbf{J}_{uu}$ and in $\mathbf{J}_{TT}$ are occurring between larger numbers of degrees of freedom than with regular bilinear shape functions.

One noticeable aspect with the implementation of the proposed preconditioners is the memory allocations requirements. Since LU direct solvers are used for comparison as well as to solve inner subblocks in each of the preconditioners, as shown in Figure 3.3, it is



(a) EQPS field (left) and element matrix (right) at $\epsilon_{yy} = 2.5e^{-3}$ and $t = 0.10\mu s$, corresponding to the linear elastic regime.

(b) EQPS field (left) and element matrix (right) at $\epsilon_{yy} = 1.6e^{-2}$ and $t = 0.25\mu s$, corresponding to the onset of plasticity (which indicates mild nonlinearity).

(c) EQPS field (left) and element matrix (right) at $\epsilon_{yy} = 2.4e^{-1}$ and $t = 0.97\mu s$, corresponding to the the stress collapse regime (which indicates high nonlinearity).

Figure 3.6: EQPS and Jacobian element sparsity pattern for quadratic (P2) INSQ element, corresponding to the three deformation regimes marked in the curves of Figure 3.5. We purposely highlight only the $\mathbf{J}_{\eta\eta}$ terms where the color bar indicate the values in this block. Notice that the sparsity pattern of the Jacobian is evolving with the plasticity.

important to make optimal choices for LU solvers. Accurate identification of the sparsity pattern of each element leads to more efficient memory allocations and therefore significant cpu time savings. Table 3.1 illustrates the gains in memory usage and computational time of sparse element memory allocation in comparison with dense element memory allocation. These results are reported for on a 30 by 30 mesh with quadratic (P2-NURBS) elements for a simulation running from $t = 0$ to $t = 1.2\mu s$. It can be observed that about 20% savings in memory allocation directly leads to 20% savings in cpu time for the proposed preconditioners. However, describing accurately the sparsity pattern of the element matrix, may also lead to highly degraded performance of LU solvers. This degradation can be minimized if a proper reordering of equations algorithm is employed. Optimal LU results are obtained with the Nested Dissection algorithm [51,80], which is the default in PETSc for dense memory allocations. However, for sparse element allocations the results in the Table 3.1 have been obtained using a Minimum Degree ordering (AMD or QMD) [4,47] which gives better timings. The results in the Table 3.1 have been obtained using the optimal ordering for each element allocation type.

	Dense element allocation		Sparse element allocation	
	cpu [s]	Memory [MB]	cpu [s]	Memory [MB]
LU solver	323.82	150.83	448.67	128.11
Schur-Schwarz	427.94	195.94	413.32	170.18
Schur-Schur	568.24	194.98	382.93	170.13

Table 3.1: Comparison of cpu time [s] (wall time of the simulation) for dense element allocation and sparse element allocation of the Jacobian matrix on a 30 by 30 mesh with quadratic elements from $t = 0$ to $t = 1.2\mu s$.

Figure 3.7a shows the stress-strain curve similar to Figure 3.5 and in addition the number of linear iterations of GMRES solver at the first Newton iteration of each time step. Due to the choice of a relatively small time step, $\Delta t = 40ns$, the number of Newton iterations per time step does not exceed 6. Nonetheless the maximum number of Newton iterations occur at the onset of plasticity (ϵ_{yy}=0.0025) and rapidly reduces back to three or two iterations per time step. On the other hand, the linear iterations required to converge at each Newton iteration increase steadily with the increase of the deformation, as indicated in Figure 3.7b. This is due to the increasing spectral radius of the iteration matrix as shown in Figure 3.2 and the evolving sparsity pattern of the Jacobian matrix, as illustrated in Figure 3.6.

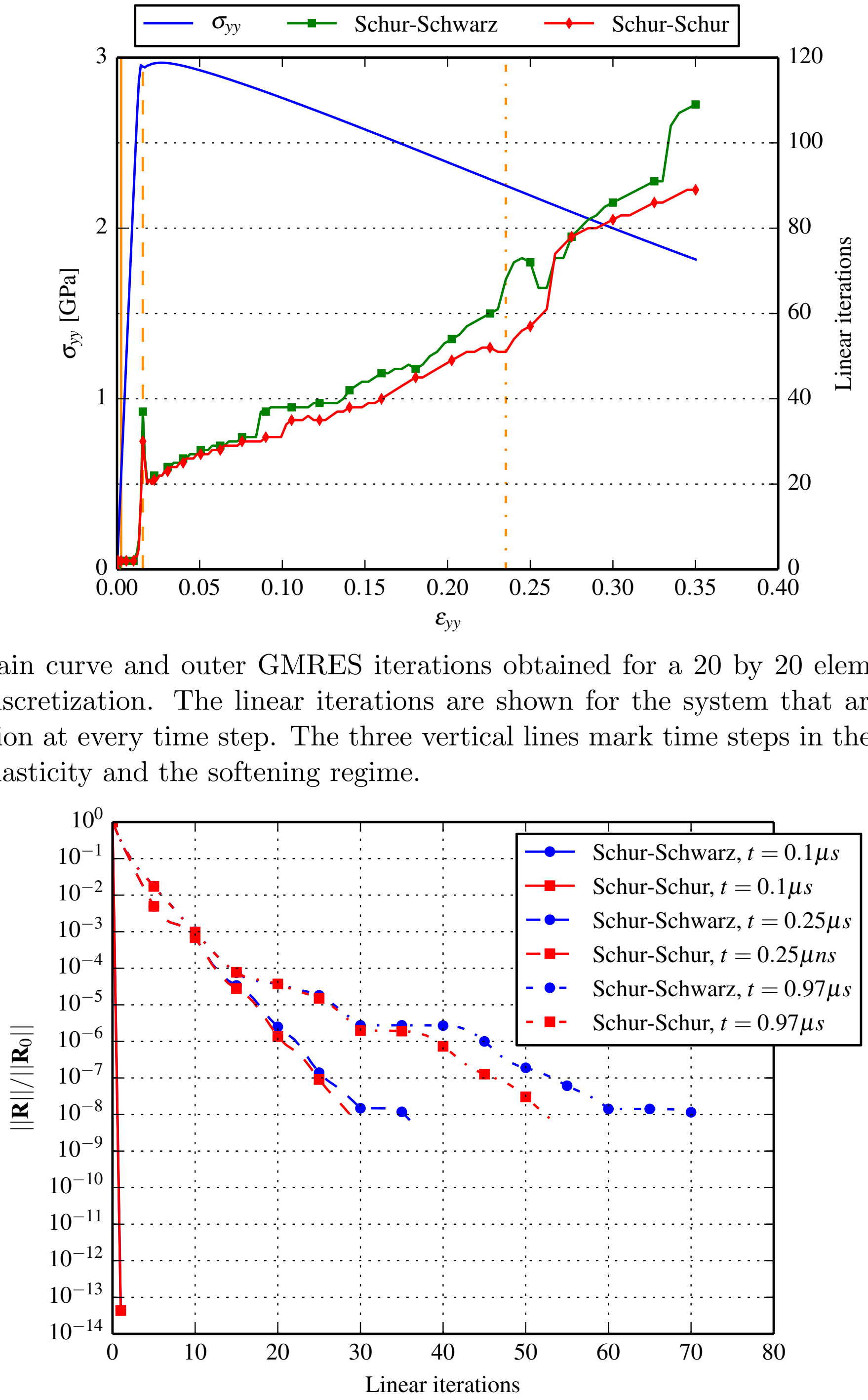
Table 3.2 reports a comprehensive convergence study of the preconditioners with respect to h- and k-refinements using the NURBS shape functions. The problem described in Figure 3.4 is solved using four different meshes: 10×10, 20×20, 30×30 and 40×40 elements, with four different NURBS shape functions degree: $p = 1$, $p = 2$, $p = 3$ and $p = 4$.

		LU solver				Schur-ILU			
		10^2	20^2	30^2	40^2	10^2	20^2	30^2	40^2
NURBS order	P1	10.49	41.36	94.48	168.78	14.33	69.20	254.48	600.08
	P2	33.71	138.28	318.88	580.70	45.06	268.21	792.97	1741.09
	P3	101.62	415.23	960.68	1734.03	147.30	703.06	1909.84	4081.25
	P4	260.49	1088.46	2425.82	4446.31	365.17	1654.68	4328.16	8833.92

		Schur-Schwarz				Schur-Schur			
		10^2	20^2	30^2	40^2	10^2	20^2	30^2	40^2
NURBS order	P1	12.68	49.56	122.44	231.05	13.99	51.74	122.99	226.02
	P2	39.96	172.52	409.88	766.84	41.44	170.15	382.93	761.39
	P3	122.42	509.96	1177.34	2189.86	118.43	484.65	1153.38	2103.72
	P4	305.42	1241.48	2876.39	5273.26	291.62	1194.01	2765.66	5047.98

Table 3.2: Overall computational time (in seconds) to simulate the shearband benchmark example as function of h- and k-refinement. Note that LU direct solver performs better than all three preconditioners.

As can be expected the proposed preconditioners have larger costs than LU solvers on small systems because of the setup time required for the Schur-Schwarz and Schur-Schur preconditioners. We also observe that the Schur-Schwarz preconditioner is more suited for low order discretizations outperforming the Schur-Schur preconditioner on almost all P1 meshes. However, as the polynomial order is increased the Schur-Schur preconditioner is clearly more efficient than the Schur-Schwarz perconditioner. Note also that an off-the-shelf Schur-ILU



(a) Stress-strain curve and outer GMRES iterations obtained for a 20 by 20 element mesh with P3 NURBS discretization. The linear iterations are shown for the system that arise at the first Newton iteration at every time step. The three vertical lines mark time steps in the linear regime, the onset of plasticity and the softening regime.

(b) Convergence behavior of the outer GMRES solver in the linear elastic regime, at the onset of plasticity and in the softening regime at times marked on Figure 3.7a by orange vertical lines.

Figure 3.7

number of processors	number of equations per partition		number of ghost elements per partition	
	AVG	STD	AVG	STD
16	12,525.0	96.1	98.8	62.3
32	6,262.5	93.9	79.0	53.3
64	3,131.2	48.3	59.3	32.2
128	1,565.6	26.3	42.2	23.4

Table 3.3: Meshes and partitions statistics for a 200 by 200 mesh, means are denoted AVG and standard deviations by STD.

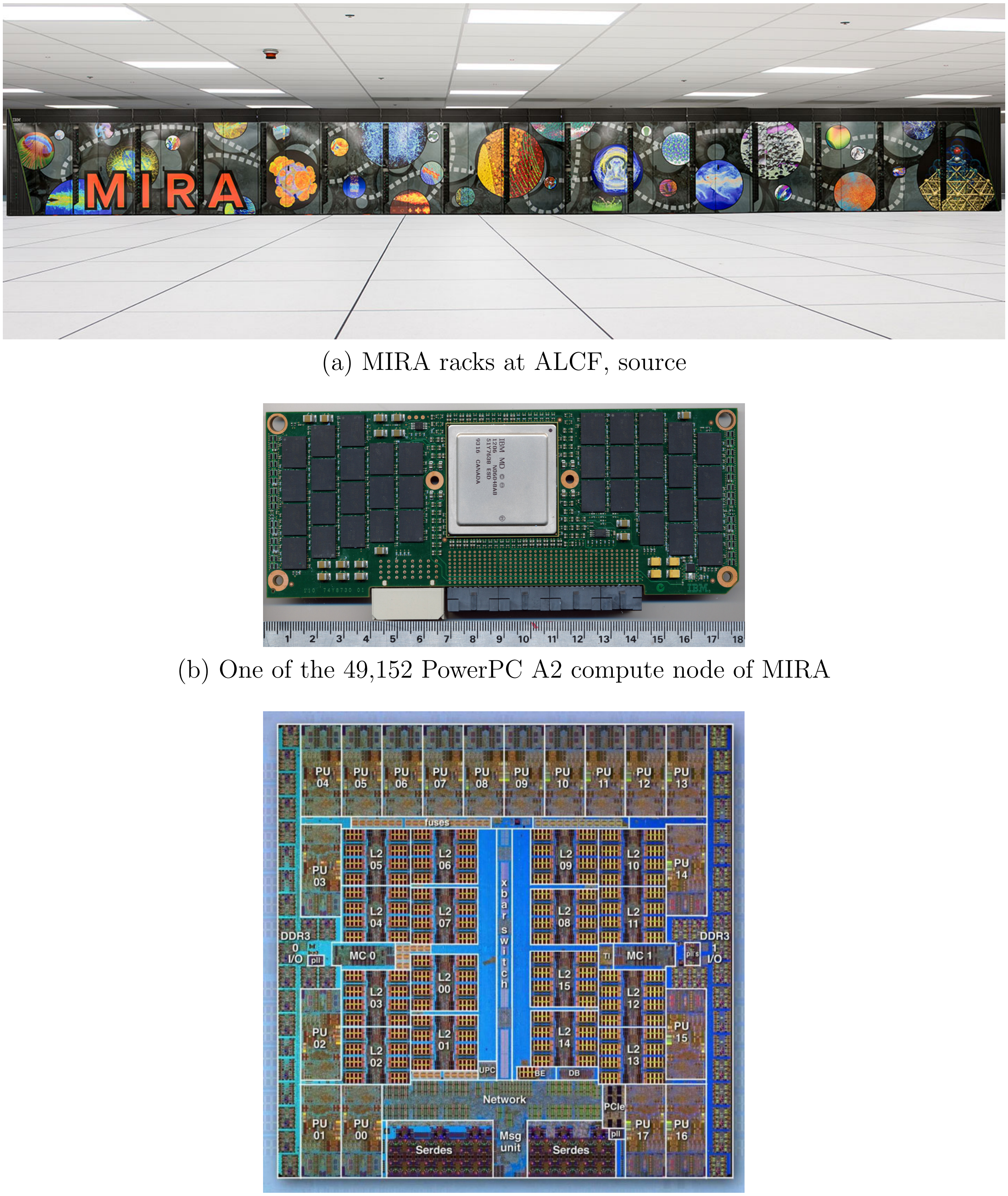
preconditioner is clearly the worst preconditioner with the poorest performance.

3.2.3 Parallel performance of the preconditioners

In this section we study the convergence and scalability of the preconditioners implemented on a parallel machine (MIRA) at Argonne National Lab. The supercomputer MIRA used for these simulations is composed of $49,512$ compute nodes, each node has 16GB of dedicated shared RAM and 16 Power PC A2 cores at 1600MHz.

First, strong scaling studies on 100 by 100 and 200 by 200 meshes are conducted. The problem is decomposed in increasingly smaller partitions, as shown in Figure 3.9, but the number of cores per partition is kept the same thus increasing the total number of cores used to solve the problem. An example of two partitionings, obtained using METIS [61], are shown in Figure 3.9 and statistics of the partitions used for the 200 by 200 simulations are presented in Table 3.3. These particular partitions are based on optimized graph algorithms in METIS for our specific descretization.

In Table 3.4 the results for the 100 by 100 mesh for time $t = 0$ to $t = 1.2\mu s$ are summarized. It shows that the LU solver is initially faster when a small number of cores are used to solve the problem but as the number of cores and the number of compute nodes used for the simulation increases, the proposed preconditioners are proving more efficient in utilizing the resources available. In particular it can be seen that a GMRES solver preconditioned by



(a) MIRA racks at ALCF, source

(b) One of the 49,152 PowerPC A2 compute node of MIRA

(c) Die of the application-specific integrated circuit of the PowerPC A2

Figure 3.8: MIRA operates at multiple scales from rack to core, understanding this architecture allows to write code that scales well on the supercomputer. scource: Argonne National Laboratory

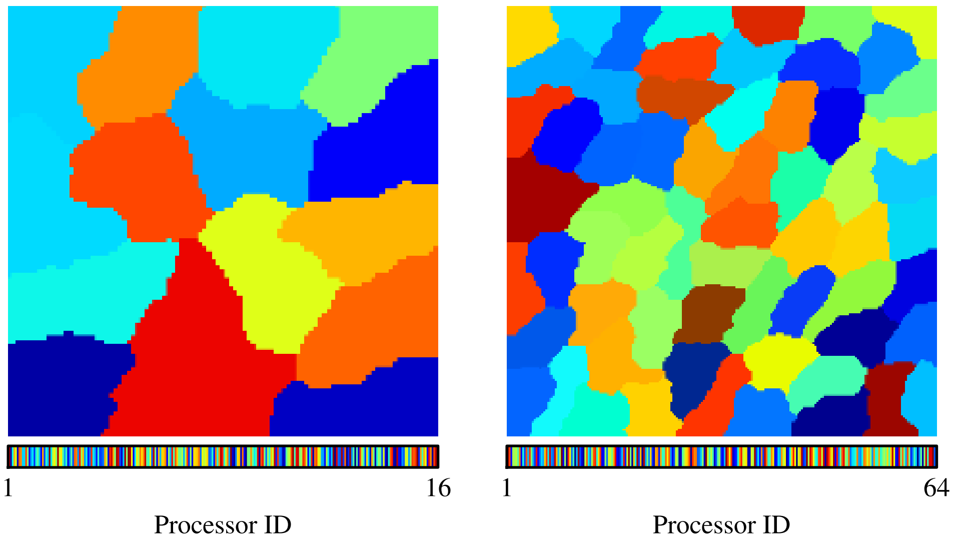


Figure 3.9: Illustration of the partition of the problem domain into 16 (left) and 64 (right) processors using the METIS algorithm [61]. Note that the partitions don't correspond to the structured mesh of the discretized physical problem.

the Schur-Schwarz preconditioner, gives similar performance as the LU direct solver on 32 cores, however on 64 cores it outperforms the other solvers and converges significantly faster. Note that the Schur-Schur preconditioner also shows good parallel scalability, although on this relatively small problem converges slower than the other methods.

The CPU times and computational speedup [64] for these simulations are presented in Figure 3.10. It is observed that the computation time of the LU solver is always better than that of the GMRES solver preconditioned by the Schur-Schur preconditioner, the computational speedup achieved with the later is better as the number of processors increase.

The same trends are also observed on the 200 by 200 problem that was run for 15 time steps due to limitations of the allocated compute time on MIRA. The CPU time required to solve the problem with the LU solver is almost always longer than that required by the GMRES solver preconditioned with the Schur-Schwarz and Schur-Schur preconditioners, as reported in Table 3.5. The speedups shown in Figure 3.11, are also greater with the proposed preconditioners, especially for the Schur-Schwarz preconditioner that requires fewer

Processors			LU	Schur-Schwarz		Schur-Schur	
-n	-p	-np	CPU [s]	CPU [s]	Iterations	CPU [s]	Iterations
1	4	4	2670.87	3600+	3772	3600+	2706
1	8	8	1809.87	2929.00	5110	2892.17	3504
1	16	16	1378.88	1861.63	5268	2250.42	3591
2	16	32	1159.95	1176.04	5385	1757.05	3699
4	16	64	1058.16	890.48	5621	1361.88	3749

Table 3.4: 100 by 100 mesh, -n is the number of compute nodes, -p is the number of cores per compute node and -np is the total number of cores used for these simulation. Final time step is the total number of times steps required for this simulation and final time is the final simulated time

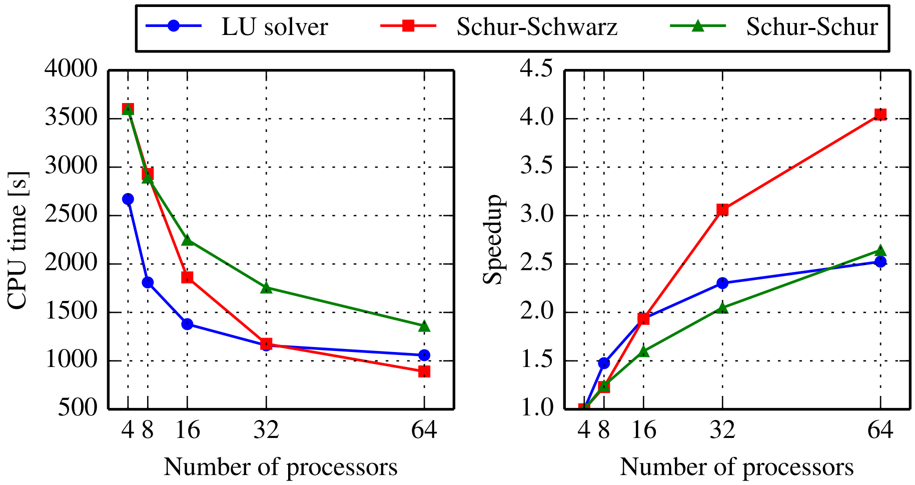


Figure 3.10: Strong scaling of the preconditioners compared to the LU solver for a 100 by 100 mesh. The Schur-Schwarz preconditioner outperforms the other methods as the number of processors increase.

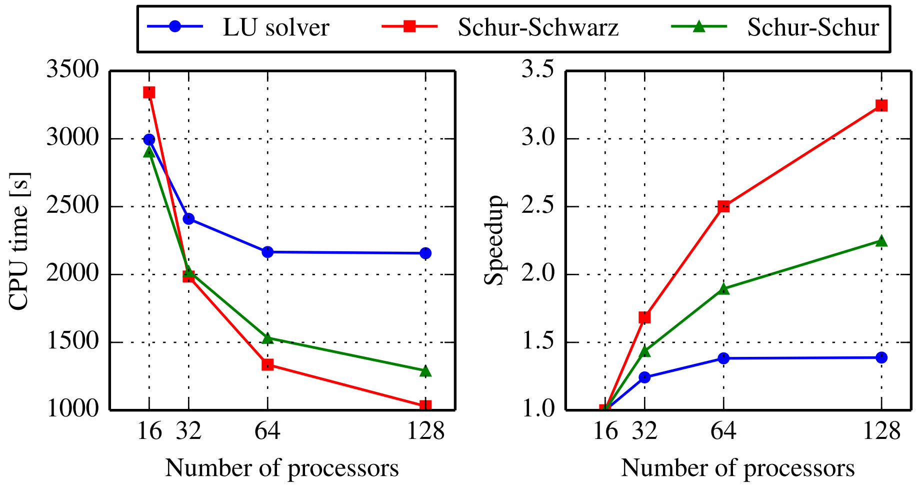


Figure 3.11: Strong scaling of the preconditioners compared to the LU solver for a 200 by 200 mesh.

matrix multiplications and hence less message passings during the simulations. While the iteration matrix of the Schur-Schwarz preconditioner has a worse spectral radius, which means more GMRES iterations per linear solve as presented in Figure 3.7a, the computational savings achieved by the reduced communications are sufficient for an overall faster solver/preconditioner pair.

Processors			LU	Schur-Schwarz		Schur-Schur	
-n	-p	-np	CPU [s]	CPU [s]	iterations	CPU [s]	iterations
2	8	16	2994.00	3342.02	2373	2906.31	1410
4	8	32	2410.32	1985.20	2450	2024.08	1453
8	8	64	2165.67	1335.78	2500	1533.18	1477
16	8	128	2156.96	1029.80	2595	1291.75	1517

Table 3.5: 200 by 200 mesh, -n is the number of compute nodes, -p is the number of cores per compute node and -np is the total number of cores used for these simulation. CPU[s] representes the simulation wall time in seconds and iterations gives the total number of linear iterations for the simulation.

Finally the weak scaling of the GMRES solver preconditioned with the Schur-Schwarz and Schur-Schur preconditioners is investigated using increasingly larger meshes but keeping the size of each partition close to constant, as reported in Table 3.6 and compared to the weak

Mesh	Equations	Processors	Equations per processor
50	57,650	8	7206
100	230,300	32	7196
200	920,600	128	7192
400	3,681,200	512	7189

Table 3.6: Problem setting used for weak scaling studies. Choice of processors and mesh size that give approximately similar work per processor.

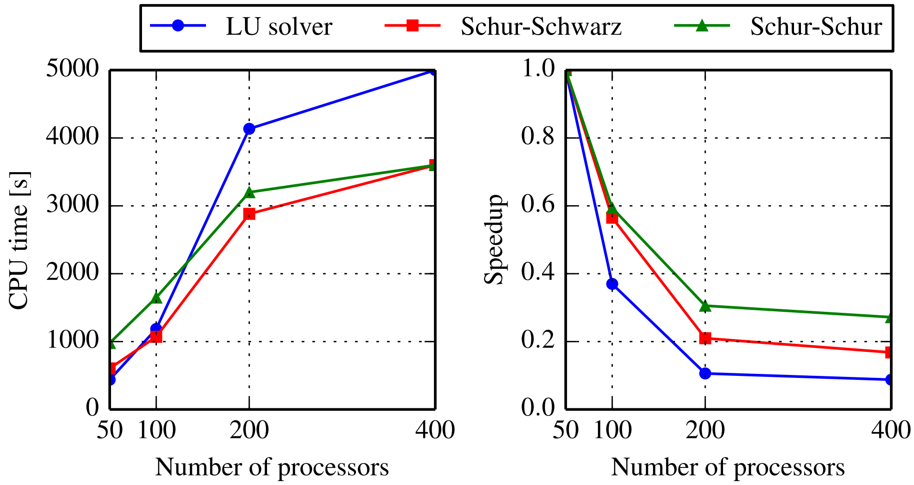


Figure 3.12: Weak scaling of the preconditioners compared to the LU solver

scaling of the LU solver. Figure 3.12 summarizes the simulations CPU times and speedups. On small meshes the LU solver has comparable performances with the preconditioned GMRES methods but as the mesh size grows its computational time rapidly increasing whereas those of the GMRES preconditioned with the Schur-Schwarz and Schur-Schur preconditioners grows at lower rates. In contrast to the results obtained for the strong scaling, the weak scalability of the Schur-Schwarz preconditioner is not as good as that of the Schur-Schur preconditioner. For large scale problems the CPU time of the Schur-Schur is eventually catching up with that of the Schur-Schwarz precondtioner as can be observed for the simulation ran on a 400 by 400 mesh.

The scalability studies show that the Schur-Schwarz preconditioner has better strong scaling than the Schur-Schur preconditioner making it powerful on hardware architecture using low amount of memory per processor but the Schur-Schur preconditioner exhibit better weak scaling abilities that makes it more efficient for very large systems of equations on massively parallel systems.

3.3 Summary

In order to resolve shear bands accurately very fine meshes, specialized basis functions (we consider assumed stress, high order Isogemetric based discretization) and parallel implementations are needed. This fine resolution of shear bands entails implicit schemes with very large Jacobian systems and a sparsity pattern that evolves with the deformation of shear bands, that needs to be solved at every time step.

To this end robust parallel preconditioners to GMRES are develop in order to alleviate the linear solve bottleneck of the analysis. The key idea is to partition the Jacobian into conservation laws and constitutive laws to exploit the localized block structure of constitutive laws and obtain elegantly a Schur complement of the system. The Schur complement is then approximated by a simpler expression for which a schwarz type method, so called Schur-Schwarz preconditioner or a second inner Schur splitting, so called Schur-Schur preconditioner, are derived. A detailed presentation of the proposed preconditioners is provided with emphasis on the mathematical derivation, the block solvers employed and an eigenvalue analysis of the iteration matrix for each preconditioner.

Numerical experiments on a shear band benchmark problem are carried out in order to study the performance of preconditioners in comparison with a direct LU solver and an off-the-shelf GMRES preconditioned by ILU solver. First, we consider a serial implementation and study the behaviour of the solvers with the increase of the order of NURBS shape functions, so called k-refinement, as well as the standard h-refinement. As expected for all solvers, the solution time increases with both k- and h-refinements. On these relatively small systems, we find the direct LU solver to be the most efficient solver and converge the fastest while the GMRES preconditioned by ILU is clearly the worse. Nonetheless, the proposed two preconditioners perform also reasonably well and are closer to the LU solver. Finally, the solvers are implemented in parallel on a supercomputer at the Argonne National Laboratory. On these larger systems, a cross over is observed where both of the proposed preconditioners converge faster than the LU-direct solve. We conclude that the Schur-Schwarz preconditioner has better strong scaling that makes it more suitable for low memory per core architecture whereas the Schur-Schur preconditioner is more performant on isogeometric discretizations of higher degree and has better weak scaling making it suitable for very large simulation on massively parallel systems.

Chapter 4

Schur based field- and domain-decomposition preconditioners

Fracture mechanics is an established field of engineering science that has enormously benefited from the development of advanced computational techniques. However, many challenges still exist, for both ductile and brittle fracture, that prevent easy and automatic modeling of fracture. One major difficulty is the scale at which the physical phenomena driving the fracture is described. For ductile fracture of metals such as steel, one of the governing mechanism of failure is the formation and propagation of shear bands. These bands are usually 1 to $10\mu m$ [30] wide and appear when the material undergoes high strain rates (of the order of $10m \cdot s^{-1}$) or higher.

In order to accurately represent shear bands using the finite element method (FEM), several elements across the band width are required. This means that a local refinement, so called h-refinement, is needed in order obtain small enough elements in the shear band region [63]. Another way to obtain accuracy in the shear band region is by employing p-refinement, which means higher polynomial basis functions. In the context of Isogeometric analysis and the NURBS functions, high order discretization may be achieved by the so called k-refinement, as proposed for shear bands by the author [19] (see also Chapter 2). It should also be noted that a combination of h- and p- refinement or h- and k- refinement is also possible for shear bands. In other words, resolving accurately localization problems in the framework of monolithic schemes is computationally costly and the bottleneck whether h- p- or k- refinement remains the linear solution of the Jacobian system.

As shown in chapter 3, efficient preconditioners can be devised in order to reduce the amount of time spent solving these linearized system, however, further improvements can be made.

To this end, we present a preconditioner that takes advantage of the physics. In the shear band problem, most of the deformation and plasticity is localized in a narrow band while out of this domain only small deformations and minor plasticity is observed. Hence, a preconditioner that decomposes the domain and concentrate more effort in the shear band domain while reusing information away from the band could lead to significantly improved computational performance.

Domain decomposition strategies for linear elasticity and brittle fracture modeled by the eXtended Finite Element Method (XFEM) have previously been proven successful [20, 53, 104].

In the work developed in this chapter, we proceed along similar conceptual lines, however due to the special features of the Jacobian arising from the shear band problem a different approach is taken. First, we employ a schur complement strategy as developed in Chapter 3, which takes advantage of the gauss point history variables conveniently. Then, a general overlapping domain decomposition procedure is performed, partitioning the domain into so called 'shear band subdomain' and a 'healthy subdomain' and are used to precondition the Schur complement system. Finally, the shear band subdomain preconditioner is solved exactly with an LU solver while the healthy subdomain preconditioner is only solved once in the elastic region and reused throughout the simulation. This approach is shown to be very efficient and leads to an attractive solver for shear bands.

The remaining of this chapter is divided in two sections, in the first section we introduce the mathematical background of the proposed preconditioner and present its algorithmic implementation. The second section provides numerical studies to evaluate the performance of the proposed algorithm on a benchmark problem.

4.1 Linearized system solver

The proposed solver is based on three steps. First, a schur complement is performed as done in Section 3.1.1 . Second the Schur system is decomposed following a domain decomposition by defining a shear band subdomain and a healthy subdomain. Finally, the shear band domain is solved by an LU direct solve every step while the healthy domain is only solved exactly at the first elastic step and is then reused throughout the simulation.

4.1.1 Step I: fieldsplit Schur complement approach

In the first step a fieldsplit into constitutive laws - conservation laws is performed using a Schur complement approach as presented in chapter 3. Since the η variables for the irreducible family of elements are localized at gauss points (history variables), inverting $\mathbf{J}_{\eta\eta}$ can be done very effectively. This allows us to form the Schur complement defined in (3.4) as indicated below

$$\begin{aligned}
\mathbf{S} &= \begin{bmatrix} \mathbf{S}_{uu} & \mathbf{S}_{uT} \\ \mathbf{S}_{Tu} & \mathbf{S}_{TT} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{J}_{uu} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{TT} \end{bmatrix} - \begin{bmatrix} \mathbf{J}_{u\eta} \\ \mathbf{J}_{T\eta} \end{bmatrix} \mathbf{J}_{\eta\eta}^{-1} \begin{bmatrix} \mathbf{J}_{\eta u} & \mathbf{J}_{\eta T} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{J}_{uu} - \mathbf{J}_{u\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{J}_{\eta u} & -\mathbf{J}_{u\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{J}_{\eta T} \\ -\mathbf{J}_{T\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{J}_{\eta u} & \mathbf{J}_{TT} - \mathbf{J}_{T\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{J}_{\eta T} \end{bmatrix}
\end{aligned} \tag{4.1}$$

that we use to solve (1.15) as

$$\begin{cases} \mathbf{S}\delta\mathbf{x}_\alpha = \mathbf{R}_\alpha - \mathbf{J}_{\alpha\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{R}_\eta & \text{(4.2a)} \\ \delta\mathbf{x}_\eta = \mathbf{J}_{\eta\eta}^{-1}(\mathbf{R}_\eta - \mathbf{J}_{\eta\alpha}\delta\mathbf{x}_\alpha) & \text{(4.2b)} \end{cases}$$

where α regroups the u and T variables and the associated vectors are defined as

$$\delta\mathbf{x}_\alpha = \begin{bmatrix} \delta\mathbf{u} \\ \delta\mathbf{T} \end{bmatrix} \tag{4.3a}$$

$$\mathbf{R}_\alpha = \begin{bmatrix} \mathbf{R}_u \\ \mathbf{R}_T \end{bmatrix} \tag{4.3b}$$

The first equation in system (4.2) requires the solution of a linear system of equations whereas the second equation only requires matrix vector multiplications. To draw a comparison with the method proposed in Chapter 3, herein the approach is improved as the exact Schur complement is formed. The ability to compute the exact Schur complement $\mathbf{S}$ compared to the approximate Schur complement $\mathbf{S}^*$ allows us to apply the GMRES solver directly at the level of $\mathbf{S}$ instead of applying it to $\mathbf{J}$. These differences in the solution strategy are summarized in Table 4.1 for clarity.

	Chapter 3	Chapter 4
System of equations to solve	$\mathbf{P}^{-1}\mathbf{J}\delta\mathbf{x} = \mathbf{P}^{-1}\mathbf{R}$	$\mathbf{P}^{-1}\mathbf{S}\mathbf{x}_\alpha = \mathbf{P}^{-1}\left(\mathbf{R}_\alpha - \mathbf{J}_{\alpha\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{R}_\eta\right)$
Preconditioned iterative solver	GMRES$(\mathbf{J}, \mathbf{R}, \delta\mathbf{x}, \mathbf{P})$	GMRES$(\mathbf{S}, \mathbf{R}_\alpha - \mathbf{J}_{\alpha\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{R}_\eta, \delta\mathbf{x}_\alpha, \mathbf{P})$

Table 4.1: Comparison of the solution strategies applied in Chapter 3 and Chapter 4

In the rest of this section we will focus on how equation (4.2a) can be solved using a preconditioned GMRES solver.

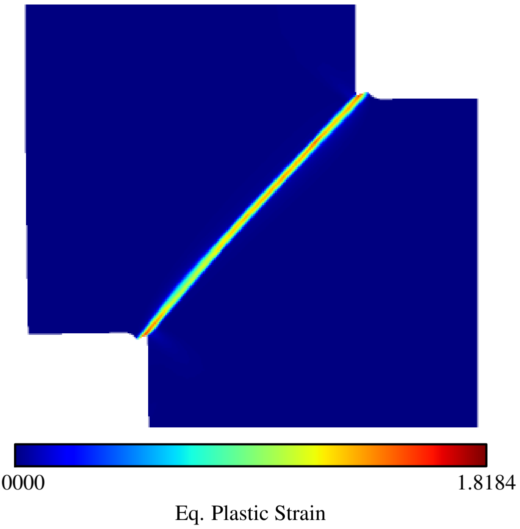


Figure 4.1: Illustration of the shear band formed under compressive impact on the upper edge.

4.1.2 Step II: domain-decomposition of the Schur complement system

Since most of the significant deformation and plasticity takes place in the narrow shear band while minor plasticity happens out of this region, a preconditioner that takes advantage of this physics would be quite efficient. Hence, a preconditioner that decomposes the domain and concentrate more effort in the shear band domain while reusing information away from the band could lead to significantly improved computational performance. This leads us to propose a domain decomposition approach to precondition the GMRES solver used in (4.2a) as represented in Figure 4.1.

Such domain decomposition approach allows us to use different operators on the shear band subdomain Ω_{sb}, and the healthy subdomain Ω_h, which is defined as the rest of the domain. Moreover, an overlap may be associated with each domain, these overlaps are parameterized by a length parameter, δ_h and δ_{sb} for the healthy and shear band subdomains respectively, see Figure 4.2.

The domain decomposition method used in work is a restricted additive Schwarz method. We define a restriction operator $\mathbf{R}_i^\delta$ as a subidentity matrix with diagonal entries set to 1

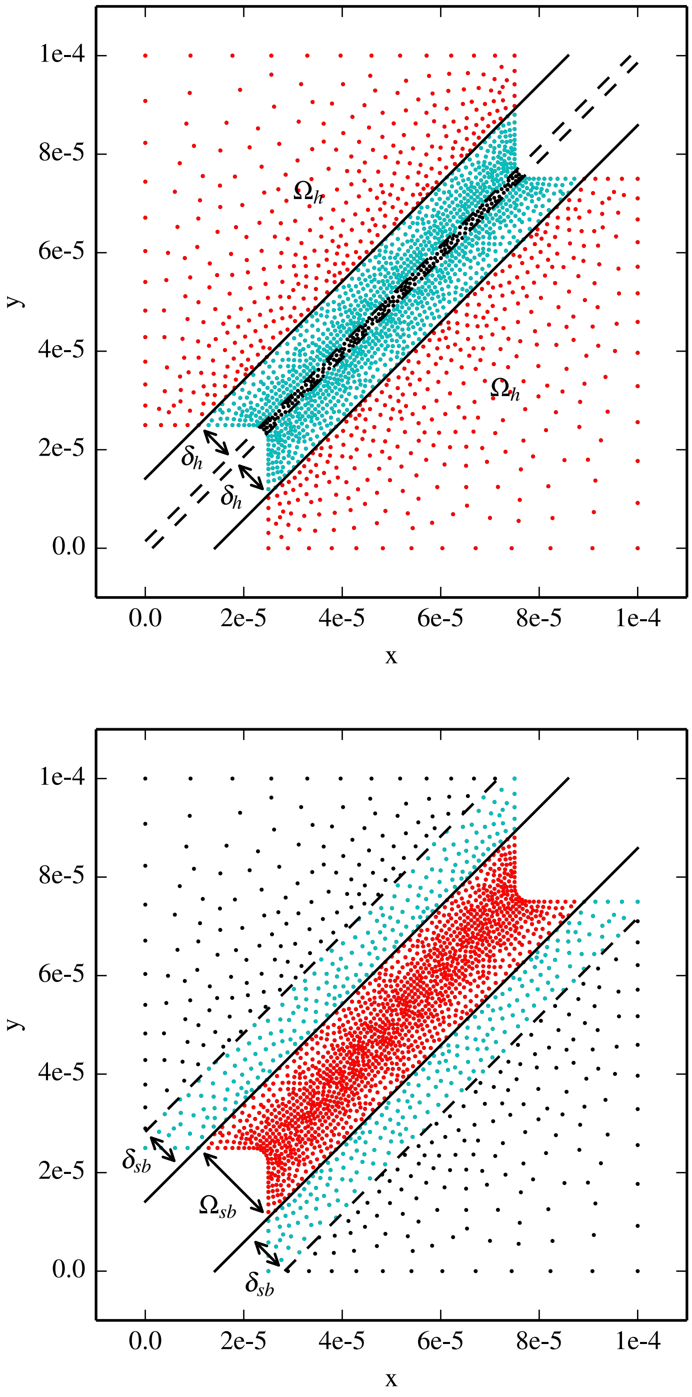


Figure 4.2: Healthy and shear band subdomains (red dots), with overlap nodes (cyan dots) for the coarse mesh.

if the corresponding node is in subdomain i or within the overlap region δ_i, and set to 0 otherwise as presented in equation 4.4.

$$\mathbf{R}_i^{\delta_i} = \begin{bmatrix} 1 & 0 & \dots & & & \\ 0 & 1 & 0 & \dots & & \\ & 0 & 0 & 0 & \dots & \\ & & 0 & \ddots & 0 & \dots \end{bmatrix}. \tag{4.4}$$

With this expression for the restriction operator we can define the matrix

$$\mathbf{S}_i = \mathbf{R}_i^{\delta_i} \mathbf{S} \mathbf{R}_i^{\delta_i}. \tag{4.5}$$

Remark 5: Note that even though $\mathbf{S}_i$ is not invertible due to the expression of $\mathbf{R}_i^{\delta_i}$, we can invert its restriction to the subspace of vectors spanned by Ω_i in $\mathbb{R}^n$. Therefore we adopt the following notation

$$\mathbf{S}_i^{-1} \equiv ((\mathbf{S}_i)_{|\Omega_i})^{-1}. \tag{4.6}$$

■

A matrix form of this preconditioner is as follows

$$\mathbf{P}_{ASM}^{-1} = \mathbf{S}_h^{-1} + \mathbf{S}_{sb}^{-1}. \tag{4.7}$$

More details on the additive Schwarz method can be found in [92, 93] and details about the restricted additive Schwarz variant that we use can be found in [27, 28]. Additionally, more detailed derivations of the additive Schwarz method are also provided in section 3.1.1 and in Appendix C.

With these notations the preconditioned form of equation (4.2a) to be solved with a GMRES algorithm is

$$\mathbf{P}_{ASM}^{-1}\mathbf{S}\mathbf{x}_\alpha = \mathbf{P}^{-1}(\mathbf{R}_\alpha - \mathbf{J}_{\alpha\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{R}_\eta) \tag{4.8}$$

$$\left(\mathbf{S}_h^{-1} + \mathbf{S}_{sb}^{-1}\right)\mathbf{S}\mathbf{x}_\alpha = \left(\mathbf{S}_h^{-1} + \mathbf{S}_{sb}^{-1}\right)(\mathbf{R}_\alpha - \mathbf{J}_{\alpha\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{R}_\eta) \tag{4.9}$$

4.1.3 Step III: Subdomains solution strategy

To lower the cost of computing $\mathbf{P}_{ASM}^{-1}$ we propose to use the elastic part of $\mathbf{S}_h$, denoted $\mathbf{S}_h^{lin}$, since the healthy subdomain does not sustain significant amount of plasticity. Hence,

$$\begin{aligned}
\mathbf{S}_h^{lin} &= \begin{bmatrix} \mathbf{M}_u & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_T + \mathbf{K}_T \end{bmatrix} - \begin{bmatrix} \mathbf{K}_u & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{M}_\sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\bar{\gamma}_p} \end{bmatrix} \begin{bmatrix} \mathbf{K}_\sigma & \mathbf{M}_{\sigma T} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{M}_u - \mathbf{K}_u\mathbf{M}_\sigma\mathbf{K}_\sigma & -\mathbf{K}_u\mathbf{M}_\sigma\mathbf{M}_{\sigma T} \\ \mathbf{0} & \mathbf{M}_T + \mathbf{K}_T \end{bmatrix}
\end{aligned} \tag{4.10}$$

Remark 6: The off diagonal term in $\mathbf{S}_h^{lin}$ is due to the thermal expansion of the material. This term can easily be dropped since in the elastic regime no heat is produced, hence no thermal expansion occures. The remaining diagonal terms are the elastic stiffness matrix and the linear thermal conduction matrix. ■

This approach allows us to compute $(\mathbf{S}_h^{lin})^{-1}$ at the first Newton iteration of the first time step of the simulation and to reuse it throughout the rest of the simulation. Moreover, this approach can also be applied adaptively by updating $\mathbf{S}_h$ according to some criterion, for instance if the number of GMRES iterations between two linearized systems solves is greater than a given maximum number of iterations.

Reformulating the additive Schwarz preconditioner in (4.7), reads

$$\mathbf{P}_{ASM_0}^{-1} = (\mathbf{S}_h^{lin})^{-1} + \mathbf{S}_{sb}^{-1}. \tag{4.11}$$

In summary, the solution strategy for the system of equation (4.2) is as follows

$$\begin{cases} \text{GMRES}(\mathbf{S},\ \mathbf{R}_\alpha - \mathbf{J}_{\alpha\eta}\mathbf{J}_{\eta\eta}^{-1}\mathbf{R}_\eta,\ \mathbf{P}_{ASM_0}^{-1}) & \rightarrow \mathbf{x}_\alpha \\ \mathbf{J}_{\eta\eta}^{-1}\left(\mathbf{R}_\eta - \mathbf{J}_{\eta\alpha}\mathbf{x}_\alpha\right) & \rightarrow \mathbf{x}_\eta \end{cases}. \tag{4.12}$$

We first form Schur complement of equation 4.1 after which we solve equation 4.11 with a GMRES solver predonditioned by $\mathbf{P}_{AMS_0}$. We now need to specify how the matrices $\mathbf{J}_{\eta\eta}$, $\mathbf{S}_h^{lin}$ and $\mathbf{S}_{sb}$ are solved for.

First in order to fully take advantage of the fact that $\mathbf{J}_{\eta\eta}$ is block diagonal, we employ a customized LU solver that has minimal setup cost and that requires no MPI communications after $\mathbf{J}_{\eta\eta}^{-1}$ has been allocated. To do so it suffice to duplicate in memory $\mathbf{J}_{\eta\eta}$, then on each MPI rank a routine sweeps through the elements locally owned and inverts individual 5 by 5 blocks at each gauss points in the element. The resulting inverted block is then stored in $\mathbf{J}_{\eta\eta}^{-1}$. The inversion of the gauss point blocks is parallelized on the local rank using OpenMP without the need for locks since the individual blocks are retrieved and stored in separate memory locations. With this implementation the time spent inverting $\mathbf{J}_{\eta\eta}$ at each Newton iteration is less than 1% of the total simulation time. With an off-the-shelf banded LU solver the time spent inverting $\mathbf{J}_{\eta\eta}$ was between 7 and 8% of the total simulation time.

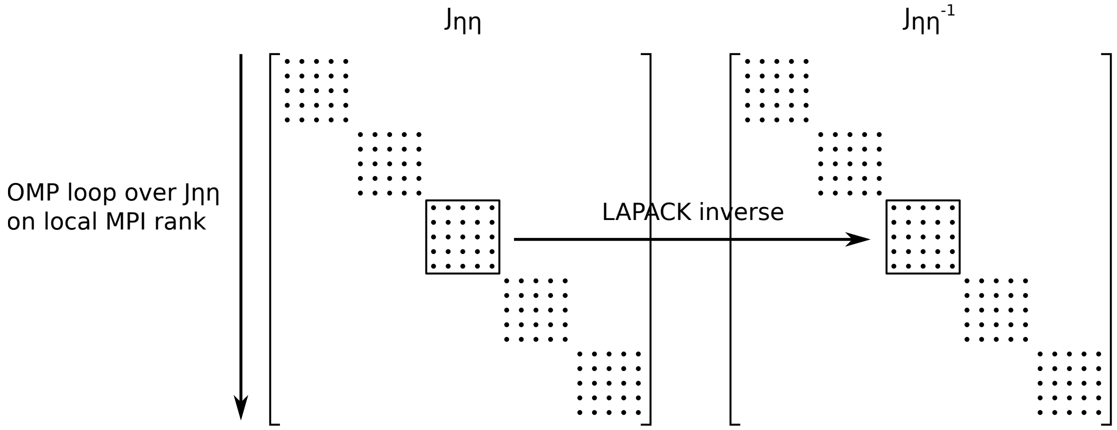


Figure 4.3: Inversion of $\mathbf{J}_{\eta\eta}$ with custom algorithm

Second $\mathbf{S}_h^{lin}$ is factored at the first time step using PETSC's LU solver with a nested dissection reordering. The factors are stored to be reused at each linear iteration during the simulation. $\mathbf{S}_{sb}$ is inverted with the same PETSC LU solver at each Newton iteration. The cost associated with the factorization of this matrix is reasonably low considering that the shear band is highly localized.

The complete preconditioner is schematically summarized in Figure 4.4.

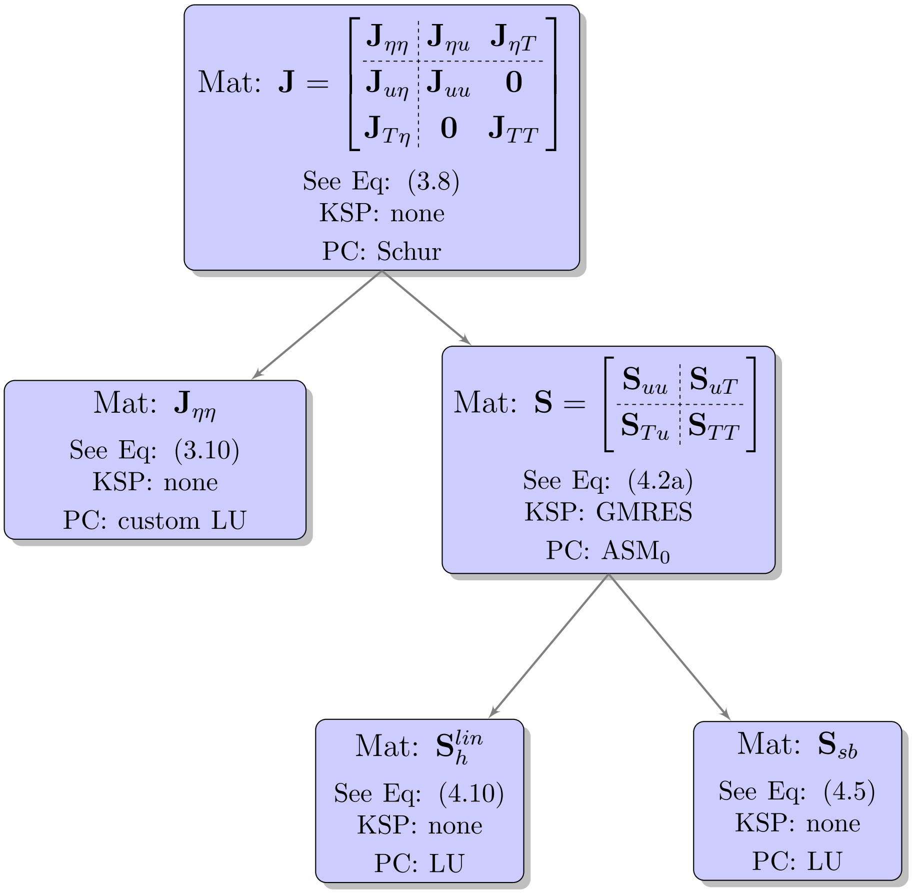

Mat: $\mathbf{J} = \left[\begin{array}{c:cc} \mathbf{J}_{\eta\eta} & \mathbf{J}_{\eta u} & \mathbf{J}_{\eta T} \\ \hdashline \mathbf{J}_{u\eta} & \mathbf{J}_{uu} & \mathbf{0} \\ \mathbf{J}_{T\eta} & \mathbf{0} & \mathbf{J}_{TT} \end{array}\right]$
See Eq: (3.8)
KSP: none
PC: Schur
Mat: $\mathbf{J}_{\eta\eta}$
See Eq: (3.10)
KSP: none
PC: custom LU
Mat: $\mathbf{S} = \left[\begin{array}{c:c} \mathbf{S}_{uu} & \mathbf{S}_{uT} \\ \hdashline \mathbf{S}_{Tu} & \mathbf{S}_{TT} \end{array}\right]$
See Eq: (4.2a)
KSP: GMRES
PC: ASM_0
Mat: $\mathbf{S}_h^{lin}$
See Eq: (4.10)
KSP: none
PC: LU
Mat: $\mathbf{S}_{sb}$
See Eq: (4.5)
KSP: none
PC: LU

Figure 4.4: Schematic representation of the Schur-domain decomposition preconditioner

4.2 Numerical results

In this section we present the convergence of the proposed preconditioner (denoted ASM_0) in terms of linear iterations and wall-clock time. Two experiments are conducted:

1. the proposed preconditioner is compared to state-of-practice solver/preconditioner pairs with different mesh refinements
2. a study of the influence of overlap regions, controlled by the length parameter δ, is presented

The state-of-practice solvers used for comparison are PETSC's LU solver applied to $\mathbf{J}$ (denoted LU($\mathbf{J}$)), PETSC's LU solver and a GMRES solver preconditioned with an ILU(0) algorithm (denoted LU($\mathbf{S}$) and GMRES($\mathbf{S}$,ILU(0)) respectively), both applied to solve equation (4.2a) while the inverse of $\mathbf{J}_{\eta\eta}$ is computed using the custom algorithm proposed in section 4.1.3. The LU solvers are the fastest algorithm on small meshes and hence constitute a good comparison point here. The GMRES-ILU(0) preconditioner provied a good compromise of robustness, speed and ease use.

Finally the preconditioner is also compared to an ASM_0 variant denoted ASM, that uses $\mathbf{S}_h$ instead of $\mathbf{S}_h^{lin}$ to evaluate the appropriateness of the elastic assumption outside of the shear band zone.

The numerical examples are run on a DELL optiplex 755 desktop with 4 cores at 2.83GHz and 4GB of RAM. The PDE model is implemented using the Finite Element Analysis Program (FEAP) [98], the linear solves are done with PETSc [11, 26, 94]. The postprocessing and graphics are generated using ParaView [50] and Matplotlib [57].

4.2.1 Shear band in a metal plate formed under impact

The numerical simulation are conducted on the impact problem presented in Figure 4.5. The problem consists of a square plate of size $1E-4m$ with square $2.5E-5m$ cutouts at the top-right and bottom-left corners. The corner of the cutouts are rounded with concave fillets of radii $R = 3E-6m$ to avoid too large stress concentration in these areas. The bottom and right edges of the plate are fixed and the top edge of the plate is subjected to an inward velocity v_{BC} that results in large compressive stresses in the plate. The velocity profile applied is shown in Figure 4.6, at time $t = 0\mu s$, $v_{BC} = 0m/s$, it then increases linearly to $v_{BC} = 25m/s$ at time $t = 2\mu s$ after which it remains constant.

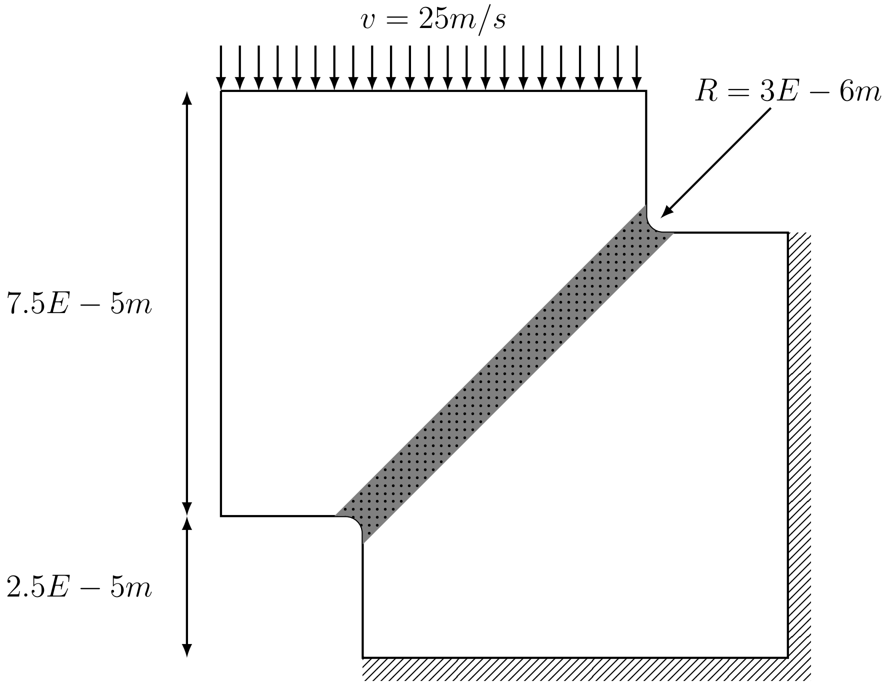


Figure 4.5: Impact onto a metallic plate with rounded cutouts. The gray dotted subdomain indicate the region where the shear band forms.

The initial applied temperature on the plate is uniform and set to $T = 293.0K$ and the normal thermal flux at all edges of the plate is null i.e. insulated conditions. Under this assumption all the heat generated by the plasticity remains in the plate during the simulation.

The material parameters used for this example are similar to those reported in Table 2.5

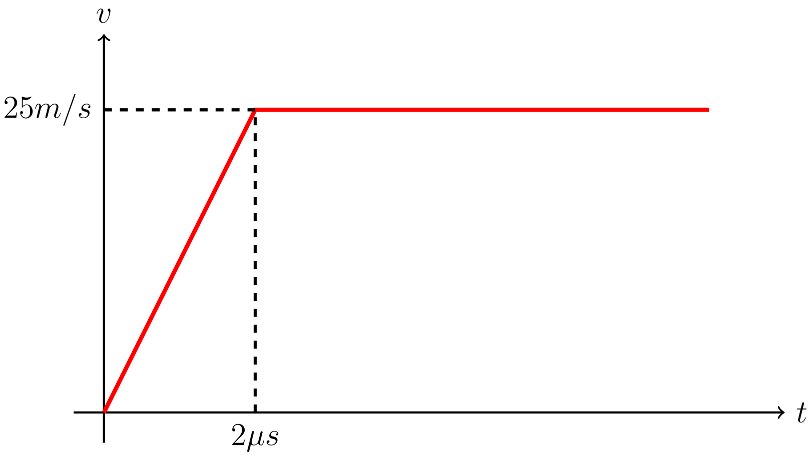


Figure 4.6: The velocity profile applied to the top edge of the plate under compression.

with the noticeable exception of the thermal conductivity κ that is set to 50.0 compared to 803.5 in the previous examples. This leads to much sharper shear bands since the heat does not travel as much and hence the thermal softening effect is stronger than observed previously. The dimensions of the plate are also an order of magnitude larger than those previously modeled in Chapter 2 and Chapter 3, which also leads to a narrower shear band. In this example the geometry leads to a natural concentration of stresses at the two opposite fillets and hence no imperfection is required to trigger the shear band. The equivalent plastic strain field on the deformed configuration at time $t = 1.5\mu s$ is shown in Figure 4.1. We can see that the shear band extends from one fillet to the other and that a significant displacement jump occurs along the band.

Finally, the mesh for the simulation is generated with GMSH [61, 62] using the input file 'msh.geo' presented in Appendix D and can be seen in Figure 4.7. The mesh is unstructured with quadrilateral elements only. A local refinement along the axis of the shear band is used to ensure that the shear band features are resolved accurately.

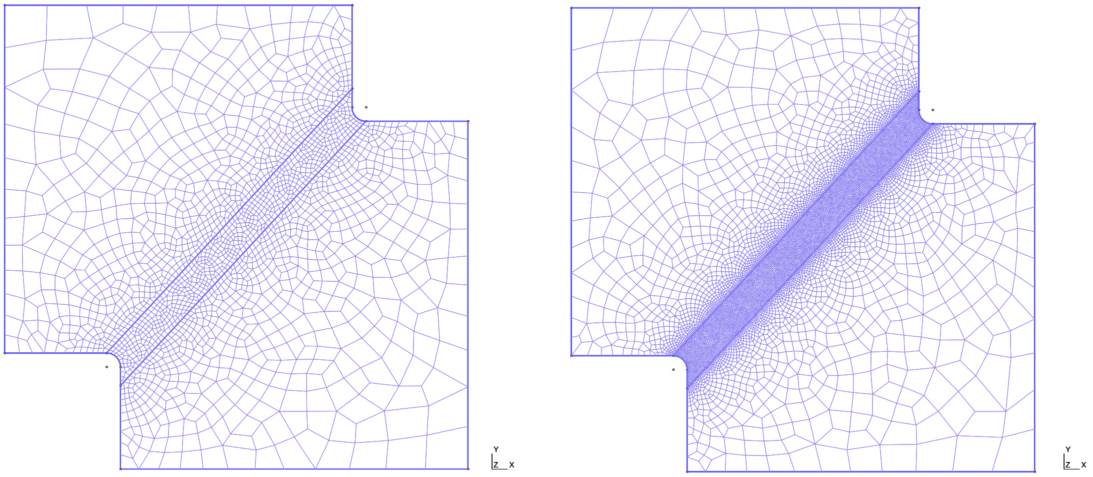


Figure 4.7: Meshes for the compressive shear band problem with local refinement along the shear band. The left mesh present a coarse level of refinement whereas the right mesh present a fine level of refinement.

4.2.2 Preconditioner performance upon mesh refinement

In this example, three different level of mesh refinement in the central region of the plate where the shear band forms, are considered. The size of the overlap between the two subdomains is set to $\delta = 1\mu m$ for all the meshes.

The mesh, subdomains and overlap statistics for all studies are presented in Table 4.2.

Remark 7: As the mesh is refined the number of nodes in the overlap region increases which usually leads to faster convergence of the GMRES algorithm. ■

Mesh	Nodes	Healthy	Healthy overlap	Shear band	Shear band overlap
Coarse	2173	1257	165	916	132
Medium	5463	1969	505	3494	367
Fine	10084	2500	1022	7584	584

Table 4.2: Summary of the mesh, domains and overlap statistics for the three meshes used.

The simulation wall times associated with each solver-preconditioner pair on all the meshes are summarized in Table 4.3. On all the meshes the proposed preconditioner is performing best in terms of simulation time. Its preformance is closely followed by that of the ASM preconditioner which provides almost as good a strong scaling as the ASM$_0$ preconditioner does.

The ILU(0) preconditioner performs very well on the coarsest mesh but its scaling is very poor. Furthermore, as expected the LU($\mathbf{J}$) and LU($\mathbf{S}$) preconditioners have the worst scalings, LU($\mathbf{J}$) performing slightly better than LU($\mathbf{S}$) on the coarsest mesh because of the time associated with the computation of the Schur complement, but forming the Schur complement proves to be a good option as the mesh is refined.

Mesh	LU($\mathbf{J}$)	LU($\mathbf{S}$)	GMRES($\mathbf{S}$,ILU(0))	ASM	ASM$_0$
Coarse	373.36	401.91	399.93	401.31	332.81
Medium	1272.19	1260.48	1256.41	1242.89	1034.71
Fine	2909.45	2715.76	5015.37	1947.29	1828.12

Table 4.3: CPU times [s] for the simulation of the shear band under compression problem for three different meshes

The total number of nonlinear iterations required for the ASM and ASM$_0$ preconditioners to perform the simulation on all the meshes (presented in Figure 4.8a) is very similar. This justify the use of the reuse of the preconditioner in the healthy domain without any updates throughout the analysis. We also notice that the LU solver requires a number of iterations very close to those of the proposed preconditioners.

In contrast to that, the GMRES-ILU(0) preconditioner performance is not reported in Figure 4.8a due to the very large number of Newton iterations it requires. The GMRES-ILU(0) is often reaching the maximum number of iterations allowed for the solution of a given linearized system which leads to time step size reduction and hence more Newton iterations compared to the other proposed solvers.

Figure 4.8b presents the number of iterations required by the AMS(0) and ASM preconditioners to complete the simulation on the three different meshes. The first trend noticed is that the ASM preconditioner always outperforms the ASM$_0$ preconditioner which is ex-

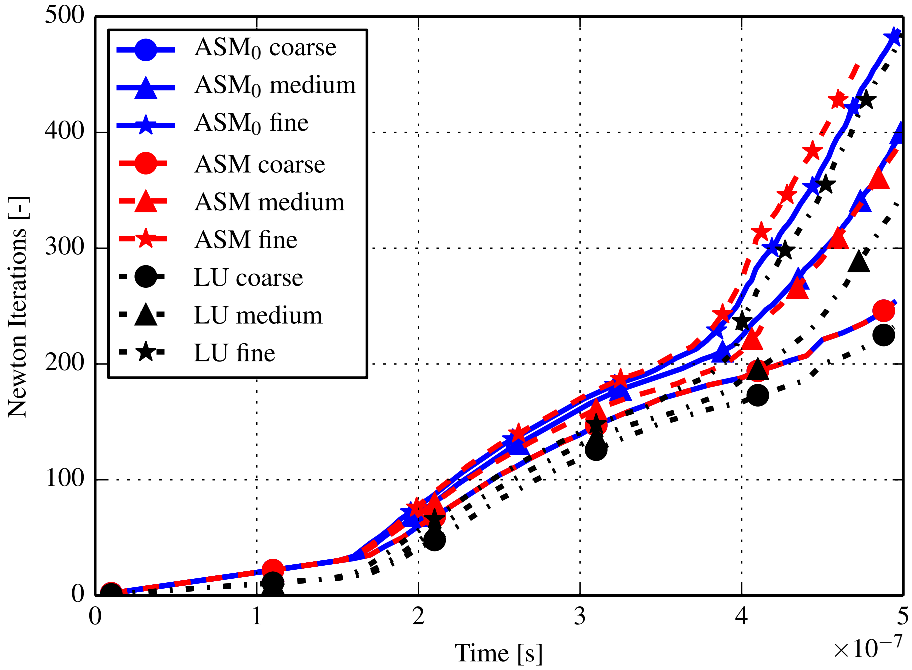
pected since ASM updates the preconditioner of the healthy domain every step. Surprisingly though the ASM_0 has a better convergence on the fine mesh until time $t = 4.2 \times 10^{-7}$ after which the ASM catches up. After investigation it appears that this is due to the fact that when the plasticity starts to develop in the plate the ASM preconditioner is slightly slower than ASM_0 which leads to a time step reduction and hence more linear iterations. This phenomenon is not indicative of the convergence of ASM compared to ASM_0 but more an artifact due the specific value of the convergence criteria for this experiment.
The second trend is that despite the fact that the ASM_0 preconditioner requires more linear iterations than the ASM preconditioner, the two are still close despite the much less expansive approach of ASM_0 .

Finally in Figure 4.9 we show the convergence of the GMRES(**S**,ILU(0)) preconditioner compared to the ASM_0 and AMS preconditioners. As it was the case for the nonlinear iterations, the number of linear iterations for the GMRES(**S**,ILU(0)) preconditioner are greatly larger than those required by the other preconditioners. This illustrate well the fact that the shear band problem is hard to capture properly during the entire simulation especially as we see that the number iterations blows up in the softening regime when while being reasonable at the onset of plasticity.

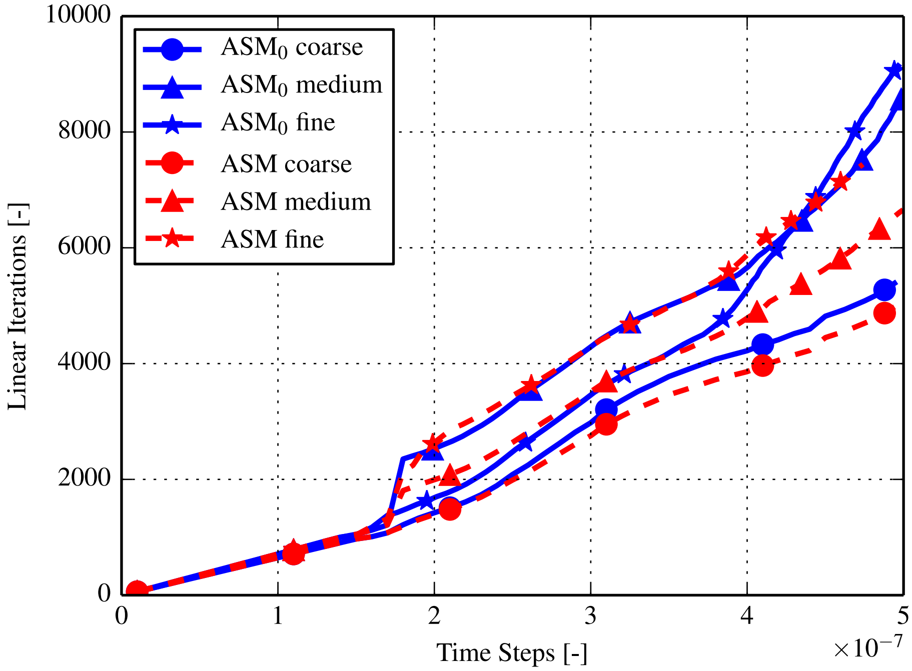
Remark 8: In Figure 4.8a we see clearly a change of slope after $t = 1.8 \times 10^{-7} s$ which marks the begining of the plastic regim in the solid. At time $t = 3.9 \times 10^{-7} s$ we see a second change of slope, this time due to the begining of the stess collapse in the solid. ■

4.2.3 Influence of the overlap length

In this section we observe the effect of the overlapping domain on the convergence of both ASM_0 and ASM preconditioners. In Figure 4.10 we present the approach used to set the overlap for the domain decomposition. The four nodes on the left and the right belong to Ω_h while the 5th and 8th nodes are in the overlap of Ω_h. The four central nodes belong to



(a) The top plot shows the Newton iterations count for the proposed preconditioner ASM_0 , its ASM variant and LU solvers.



(b) The bottom plot illustrates the cumulative number of linear iterations for the ASM_0 and ASM preconditioner for different mesh sizes.

Figure 4.8: Strong scaling study of the proposed preconditioner and comparison to state-of-the-practive solvers.

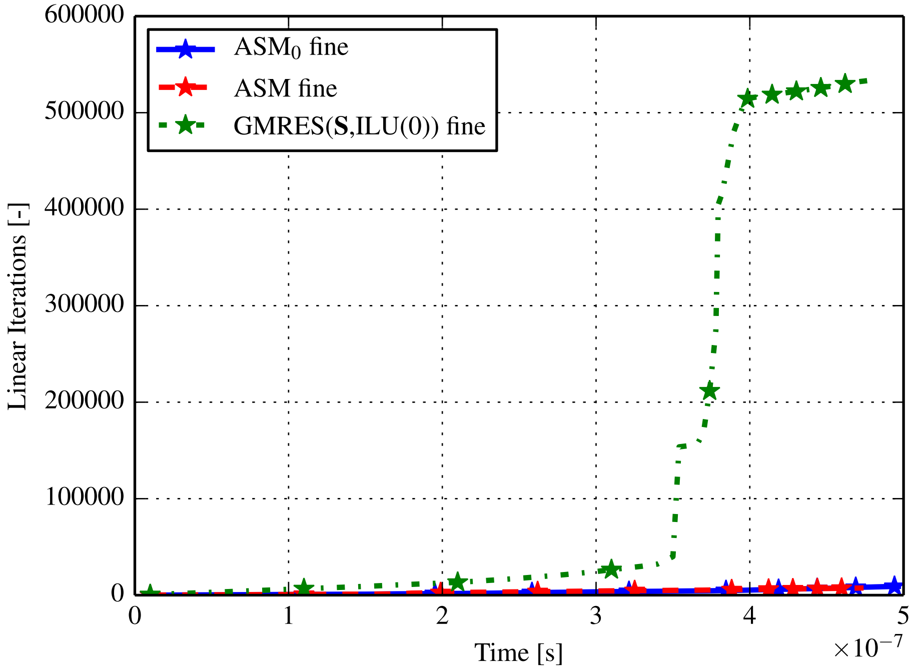


Figure 4.9: Comparison of the number of linear iterations required for the three GMRES based solvers on the fine mesh.

Ω_{sb} while the 3rd and 9th nodes are in the overlap of Ω_{sb}. The distance δ indicates how far the overlap extends from the boundary between Ω_h and Ω_{sb}.

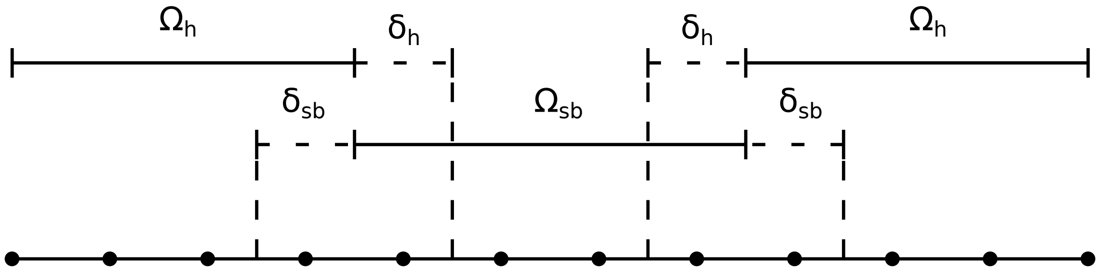


Figure 4.10: Schematic representation of both the healthy Ω_h and shear band Ω_{sb} subdomains with overlap regions of length δ.

In Table 4.4 the statistics of the mesh are summarized, we report the number of nodes in the two domains and their overlaps for varying values of δ. The number of nodes in the overlap region of Ω_h increases rapidly as the overlaps extend in the most refined region of the mesh whereas the number of nodes in the overlap region of Ω_{sb} increases more steadily. As observed in Figure 4.11, the convergence of both preconditioners is significantly improved

δ	Nodes in Ω	Nodes in Ω_h	Nodes in δ_h	Nodes in Ω_{sb}	Nodes in δ_{sb}
$1\mu m$	5463	1969	505	3494	367
$2\mu m$	5463	1969	1164	3494	622
$4\mu m$	5463	1969	2704	3494	976

Table 4.4: Mesh and domains statistics for varying overlap sizes δ and a shear band subdomain of width $10\mu m$

when the size of the overlap region grows. The reduction in iteration for the ASM$_0$ preconditioner is well correlated to the length of the overlap while that of the ASM preconditioner is more uncertain as can be seen by the fact that the overlap $\delta = 1\mu m$ and $\delta = 2\mu m$ are yielding almost the same number of linear iterations.

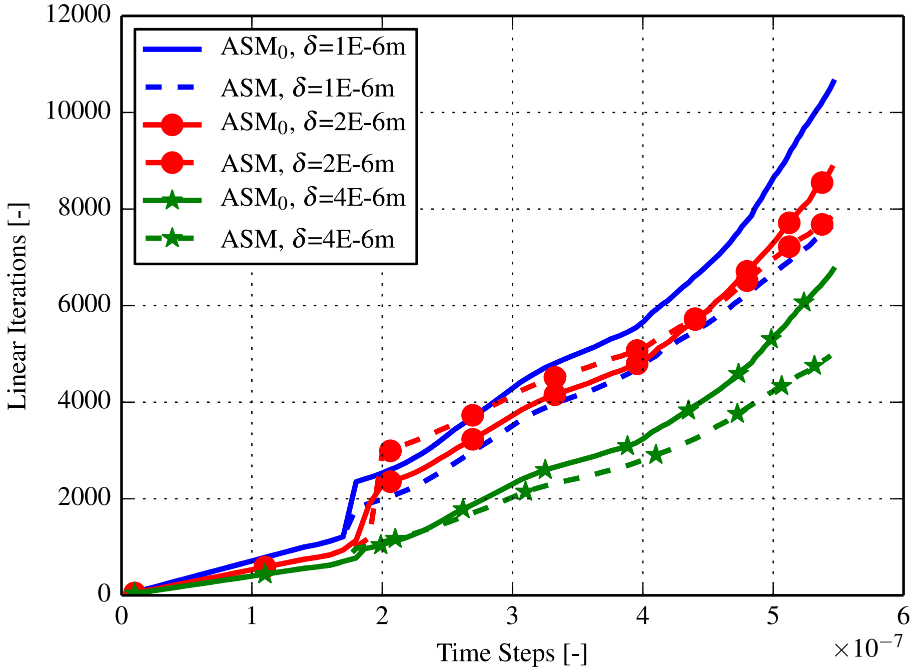


Figure 4.11: Cumulated number of linear iterations for the ASM$_0$ and ASM preconditioner to converge during the simulation.

Another overlapping possibility is to allow only one of the subdomains to have an overlap. This leads to a lighter preconditioner since one of the subdomain matrix will be smaller.However, it may also lead to a slower convergence rate [93]. Using the same mesh that was used to assess the importance of the length of the overlap we are setting first $(\delta_h, \delta_{sb}) = (\delta, \delta)$, then $(\delta_h, \delta_{sb}) = (0, \delta)$ and finally $(\delta_h, \delta_{sb}) = (\delta, 0)$ for $\delta = 1\mu m$, $2\mu m$ and $4\mu m$. In this experiment, only the ASM$_0$ preconditioner is used. The results are presented in Table 4.5 where the CPU time associated with $5\mu s$ of simulation time are reported and in Figure 4.12 that shows the total number of linear iterations required each simulation.

In Table 4.5 we observe as expected that the CPU time is strongly correlated with the number of nodes in the overlap regions. Note that in the case of $(\delta_h, \delta_{sb}) = (0, \delta)$, an overlap length of $\delta = 4\mu m$ reduces the computational time compared to an overlap length of $\delta = 2\mu m$. This is due to the large amount of iterations required at the onset of plasticity for $\delta = 2\mu m$.

δ $[\mu m]$	$(\delta_h, \delta_{sb}) = (\delta, \delta)$	$(\delta_h, \delta_{sb}) = (\delta, 0)$	$(\delta_h, \delta_{sb}) = (0, \delta)$
1	812.59	798.28	773.77
2	899.79	839.89	786.15
4	987.42	847.35	778.97

Table 4.5: CPU time [s] required to complete $5\mu s$ of simulation for three overlap strategies.

From Figure 4.12 it can be observed that despite the better performance in term of CPU time of the smaller overlaps, these are yielding significantly larger number of iterations. This is especially true when $(\delta_h, \delta_{sb}) = (\delta, 0)$, that result is interesting since δ_h contains more nodes than δ_{sb} so one would expect to observe less iterations for this strategy. However the nodes in δ_h are solved using the assumption that they behave elastically which is not the case since the nodes in δ_h also belong to Ω_{sb} where large plastic deformations occur. This approximation combined with the fact that no overlap is provided to the domain Ω_{sb} since δ_{sb} explains the counter intuitive convergence observed.

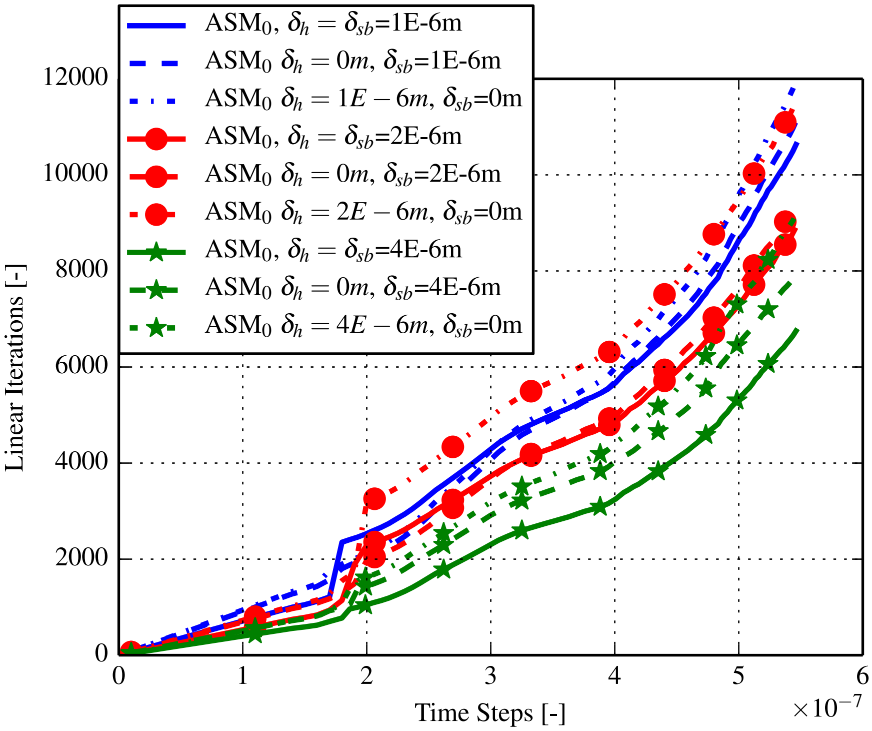


Figure 4.12: Total number of linear iterations required to compute $5\mu s$ of simulation.

4.3 Summary

In this chapter we develop a field- and domain- decomposition type preconditioner that takes into account the physics and the path of shear band localization. The key idea is to reduce the computational effort required for the simulations by considering that most of the deformation and plasticity is localized in a narrow band while only minor plasticity happens out of this band. That assumption allows us to develop preconditioners that concentrate effort only in the shear band domain while reusing information away from the band.
This concept is implemented by employing a restricted additive Schwarz algorithm applied as a preconditioner to a GMRES solver for solving the Schur complement of the global Jacobian.

The proposed preconditioner is applied to a benchmark problem and significant improvements in CPU are reported. The proposed preconditioners requires significantly less computational time than LU solvers and GMRES solvers with off-the-shelf preconditioners. That makes this strategy a promising candidate for large scale simulations. Finally, the approach is extended to an overlapping domain decomposition method and several overlapping strategies are studied.

It is demonstrated that the conceptual field- and domain- decomposition scheme works well where the overlapping and non overlapping strategies lead to attractive solvers for the shear band problem.

Chapter 5

Contribution and future directions

In this thesis three main contributions to computational modeling of shear bands have been made and are summarized as follows.

- **IGA for Shear bands:** NURBS based higher order elements for shear bands have been designed and implemented in a mixed formulation monolithic framework. These elements have several attractive features. They are insensitive to mesh size and alignment, have higher rates of convergence and can resolve the localization features accurately, alleviating some local mesh refinement requirements. The best performing element was found to be the Hybrid NURBS Shearband Quad 2 (HNSQ2) element which employ NURBS basis for displacement, temperature and stress fields while plastic strains are sampled at gauss points. The element was tested on various examples and outperformed all other elements in terms of rates of convergence and CPU time.

- **Parallel solvers for shear bands:** Robust parallel preconditioners that alleviate the linear solve bottleneck of the monolithic solution were developed. The key idea is to use a Schur complement technique and partition the Jacobian into constitutive laws-conservation law in order to leverage the properties of the four equations used to model the shear bands. Then multipurpose preconditioners such as of a Schur-Schur and Schur-Schwarz types have been developed and applied to standard irreducible discretizations as well as Isogeometric discretizations. The proposed preconditioners were tested in serial and parallel on benchmark examples and compared to standard state of practice solvers. They were found to be efficient, robust and scale better than traditional direct and iterative off-the-shelf solvers.

- **Domain decomposition preconditioners:** The parallel solvers developed are further extended to account for the physics of shear bands. To this end an overlapping domain decomposition in the form of a restricted additive Schwarz method has been proposed. This preconditioner relies on the fact that shear bands are concentrated in narrow bands with intense plastic deformations while out of these bands only minor deformations are observed. Hence, reuse of information out of the shear band is proposed within such domain decomposition strategy. The preconditioner is examined on an impact benchmark problem and is shown to reduce computational cost by about 30% on large meshes when compared to the state of practice solvers.

Several future directions are possible to continue and enhance the work presented in this document. Two main directions pertain to the physics of the problem and a third direction is related with the numerics.

- **Adaptive domain decomposition:** the ASM_0 preconditioner relies on the fact that the shear band domain is defined easily in our example since the shear band is rectiligne and forms at once. In practice shear bands can curve and propagate over time [60] this requires us to develop algorithm to detect and track the shear band while the simulation is progressing. This can be achieved using criterias based on instability points and eigenvalue analysis [6,7] coupled with closest neighbor algorithm to find the overlap regions.

- **Iterative Solvers for phase field and shear bands:** we will pursue is the extension of the work done on the preconditioners to include the phase field description of brittle fracture proposed by McAuliffe [75]. The equation describing the phase field is a diffusion equation and it is possible to include it the framework of the Schur complement method to operate on it with specialized solvers. The phase field equation has an embeded length scale that allows it to be more loosely coupled to the rest of the system hence split preconditioners can also be attempted on that field.

- **Multigrid methods for Isogeometric system:** we want to explore is the developpment of specialized multigrid algorithm that can take into account the properties of isogeometric discretization such as the larger support of the shape functions that it leads to as well as the multiple options for the refinement (h-, p- and k-refinement).

Chapter 6

Bibliography

[1] B. Aagaard, M. Knepley, and C. Williams. A domain decomposition approach to implementing fault slip in finite-element models of quasi-static and dynamic crustal deformation. *Journal of Geophysical Research: Solid Earth*, 118(6):3059–3079, 2013.

[2] E. Aifantis. On the role of gradients in the localization of deformation and fracture. *International Journal of Engineering Science*, 30(10):1279–1299, 1992.

[3] R. A. Al-Rub and G. Voyiadjis. A finite strain plastic-damage model for high velocity impact using combined viscosity and gradient localization limiters: Part i - theoretical formulation. *International Journal of Damage Mechanics*, 15(4):293–334, 2006.

[4] P. Amestoy, T. Davis, and I. Duff. Algorithm 837: Amd, an approximate minimum degree ordering algorithm. *ACM Trans. Math. Softw.*, 30(3):381–388, Sept. 2004.

[5] P. Amestoy, I. Duff, and J.-Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering*, 184(2-4):501 – 520, 2000.

[6] M. Arriaga, C. McAuliffe, and H. Waisman. Instability analysis of shear bands using the instantaneous growth-rate method. *International Journal of Impact Engineering*, (0):–, 2015.

[7] M. Arriaga, C. McAuliffe, and H. Waisman. Onset of shear band localization by a local generalized eigenvalue analysis. *Computer Methods in Applied Mechanics and Engineering*, 289(0):179 – 208, 2015.

[8] F. Auricchio, F. Brezzi, and C. Lovadina. *Mixed Finite Element Methods*, chapter 9, pages 237–278. John Wiley & Sons, Ltd, 2004.

[9] I. Babuška. Error bounds for finite element method. *NumerischeMathematik*, 16:322–333, 1971.

[10] I. Babuška and R. Narasimhan. The babuška-brezzi condition and the patch test: an example. *Computer Methods in Applied Mechanics and Engineering*, 140:183–199, 1997.

[11] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.

[12] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for Use in Computer Graphics And Geometric Modeling.* Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann, San Francisco, 1987.

[13] K.-J. Bathe. The inf-sup condition and its evaluation for mixed finite element methods. *Computers & Structures*, 79(2):243 – 252, 2001.

[14] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, and T. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):229 – 263, 2010. Computational Geometry and Analysis.

[15] Y. Bazilevs, L. B. da Veiga, J. Cottrell, T. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(07):1031–1090, 2006.

[16] Y. Bazilevs, M.-C. Hsu, and M. Scott. Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249-252(0):28 – 41, 2012. Higher Order Finite Element and Isogeometric Methods.

[17] T. Belytschko, H.-Y. Chiang, and E. Plaskacz. High resolution two-dimensional shear band computations: imperfections and mesh dependence. *Computer Methods in Applied Mechanics and Engineering*, 119(1-2):1–15, 1994.

[18] M. Benzi and D. Szyld. Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods. *Numerische Mathematik*, 76(3):309–321, 1997.

[19] L. Berger-Vergiat, C. McAuliffe, and H. Waisman. Isogeometric analysis of shear bands. *Computational Mechanics*, pages 1–19, 2014.

[20] L. Berger-Vergiat, H. Waisman, B. Hiriyur, R. Tuminaro, and D. Keyes. Inexact schwarz-algebraic multigrid preconditioners for crack problems modeled by extended finite element methods. *International Journal for Numerical Methods in Engineering*, 90(3):311–328, 2012.

[21] D. Boffi, F. Brezzi, and M. Fortin. *Mixed finite element methods and applications.* Number v. 44 in Springer series in Computational Mathematics. Springer, 2013.

[22] R. Bouclier, T. Elguedj, and A. Combescure. Locking free isogeometric formulations of curved thick beams. *Computer Methods in Applied Mechanics and Engineering*, 245-246(0):144 – 162, 2012.

[23] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising form lagrange multipliers. *RAIRO*, 8(R2):129–151, 1974.

[24] F. Brezzi and K.-J. Bathe. A discourse on the stability conditions for mixed finite elment formulations. *Computer Methods in Applied Mechanics and Engineering*, 82(1-3):27–57, 1990.

[25] F. Brezzi, M.-O. Bristeau, L. Franca, M. Mallet, and G. Rogé. A relationship between stabilized finite element methods and the galerkin method with bubble functions. *Computer Methods in Applied Mechanics and Engineering*, 96(1):117 – 129, 1992.

[26] J. Brown, M. Knepley, D. May, L. McInnes, and B. Smith. Composable linear solvers for multiphysics. In *Parallel and Distributed Computing (ISPDC), 2012 11th International Symposium on*, pages 55–62, June 2012.

[27] X.-C. Cai, M. Dryja, and M. Sarkis. Restricted additive schwarz preconditioners with harmonic overlap for symmetric positive definite linear systems. *SIAM Journal on Numerical Analysis*, 41(4):1209–1231, 2003.

[28] X.-C. Cai and M. Sarkis. A restricted additive schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21(2):792–797, 1999.

[29] M. Cervera, M. Chiumenti, and C. A. de Saracibar. Shear band localization via local {J2} continuum damage mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(9–11):849 – 880, 2004.

[30] P. Cizek, F. Bai, W. Rainforth, and J. Beynon. Fine structure of shear bands formed during hot deformation of two austenitic steels. *Materials Transactions*, 45(7):2157–2164, 2004.

[31] C. Comi and U. Perego. A unified approach for variationally consistent finite elements in elastoplasticity. *Computer Methods in Applied Mechanics and Engineering*, 121(1-4):323 – 344, 1995.

[32] J. Cottrell, T. Hughes, and Y. Bazilevs. *Isogeometric Analysis: toward integration of CAD and FEA*. John Wiley & Sons, Ltd, 2009.

[33] E. Cyr, J. Shadid, and R. Tuminaro. Stabilization and scalable block preconditioning for the Navier-Stokes equations. *Journal of Computational Physics*, 231(2):345 – 363, 2012.

[34] E. Cyr, J. Shadid, R. Tuminaro, R. Pawlowski, and L. Chacón. A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive mhd. *SIAM Journal on Scientific Computing*, 35(3):701–730, 2013.

[35] S. De and K.-J. Bathe. Displacement/pressure mixed interpolation in the method of finite spheres. *International Journal for Numerical Methods in Engineering*, 51(3):275–292, 2001.

[36] C. de Boor. *A Practical Guide to Splines.* Number v. 27 in Applied Mathematical Sciences. Springer-Verlag, New York, 2001.

[37] S. Eisenstat and H. Walker. Choosing the forcing terms in an inexact newton method. *SIAM Journal on Scientific Computing*, 17(1):16–32, 1996.

[38] T. Elguedj, Y. Bazilevs, V. Calo, and T. Hughes. B-bar and F-bar projection methods for nearly incompressible linear and nonlinear elasticity and plasticity using higher-order nurbs elements. *Computer Methods in Applied Mechanics and Engineering*, 197(33-40):2732 – 2762, 2008.

[39] T. Elguedj and T. Hughes. Isogeometric analysis of nearly incompressible large strain plasticity. *Computer Methods in Applied Mechanics and Engineering*, 268(0):388 – 416, 2013.

[40] H. Elman, V. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. Block preconditioners based on approximate commutators. *SIAM Journal on Scientific Computing*, 27(5):1651–1668, 2005.

[41] H. Elman, V. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 227(3):1790 – 1808, 2008.

[42] A. L. C. Facility. Argonne leadership computing facility, Blue Gene/Q systems and supporting resources. Technical Report ALCF-BGQ-Resources-07152014, Argonne National Laboratory, 2014.

[43] R. Falgout, J. Jones, and U. Yang. The design and implementation of hypre, a library of parallel high performance preconditioners. In A. Bruaset and A. Tveito, editors, *Numerical Solution of Partial Differential Equations on Parallel Computers*, volume 51 of *Lecture Notes in Computational Science and Engineering*, pages 267–294. Springer Berlin Heidelberg, 2006.

[44] G. Farin. *NURBS Curves and Surfaces: from Projective Geometry to Practical Use.* A.K. Peters Ltd., Natick, MA, 1995.

[45] C. Fressengeas. Adiabatic shear morphology at very high strain rates. *International Journal of Impact Engineering*, 8:141–157, 1989.

[46] M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala. ML 5.0 smoothed aggregation user's guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

[47] A. George and J. Liu. A fast implementation of the minimum degree algorithm using quotient graphs. *ACM Transactions on Mathematical Software*, 6(3):337–358, Sept. 1980.

[48] C. Giry, F. Dufour, and J. Mazars. Stress-based nonlocal damage model. *International Journal of Solids and Structures*, 48(25–26):3431 – 3443, 2011.

[49] E. Hellinger. *Der allgemeine Ansatz der Mechanik der Kontinua*, volume 4, chapter 4, page 602. Springer Verlag, 1914.

[50] A. Henderson. *ParaView Guide, A Parallel Visualization Application.* Kitware Inc., 2008.

[51] B. Hendrickson and E. Rothberg. Improving the run time and quality of nested dissection ordering. *SIAM Journal on Scientific Computing*, 20(2):468–489, 1998.

[52] L. Hermann. Elasticity equations for incompressible and nearly incompressible materials by a variational theorem. *AIAA journal*, 3(10):1896–1900, 1965.

[53] B. Hiriyur, R. Tuminaro, H. Waisman, E. Boman, and D. Keyes. A quasi-algebraic multigrid approach to fracture problems based on extended finite elements. *SIAM Journal on Scientific Computing*, 34(2):A603–A626, 2012.

[54] V. Howle, R. Kirby, and G. Dillon. Block preconditioners for coupled physics problems. *SIAM Journal on Scientific Computing*, 35(5):S368–S385, 2013.

[55] T. Hughes, J. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135 – 4195, 2005.

[56] T. Hughes, L. Franca, and G. Hulbert. A new finite element formulation for computational fluid dynamics: viii. the galerkin/least-squares method for advective-diffusive

equations. *Computer Methods in Applied Mechanics and Engineering*, 73(2):173 – 189, 1989.

[57] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[58] I. Ipsen. A note on preconditioning nonsymmetric matrices. *SIAM Journal on Scientific Computing*, 23(3):1050–1051, 2001.

[59] Z. Jiao, T. Pian, and S. Yong. A new formulation of the isoparametric finite elements and the relationship between hybrid stress element and incompatible element. *International Journal of Numerical Methods in Engineering*, 40:15–27, 1997.

[60] J. Kalthoff and S. Winkler. Failure mode transition at high rates of shear loading. *DGM Informationsgesellschaft mbH, Impact Loading and Dynamic Behavior of Materials*, 1:185–195, 1988.

[61] G. Karypis and V. Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, Jan. 1998.

[62] G. Karypis and V. Kumar. Parallel Multilevel series k-Way Partitioning Scheme for Irregular Graphs. *SIAM Review*, 41(2):278–300, Jan. 1999.

[63] A. Khoei, A. Tabarraie, and S. Gharehbaghi. H-adaptive mesh refinement for shear band localization in elasto-plasticity cosserat continuum. *Communications in Nonlinear Science and Numerical Simulation*, 10(3):253 – 286, 2005.

[64] D. Kuck, Y. Muraoka, and S.-C. Chen. On the number of operations simultaneously executable in fortran-like programs and their resulting speedup. *Computers, IEEE Transactions on*, C-21(12):1293–1310, Dec 1972.

[65] S. Li and W. Liu. Numerical simulations of strain localization in inelastic solids using mesh-free methods. *International Journal for Numerical Methods in Engineering*, 48(9):1285–1309, 2000.

[66] S. Li, W.-K. Liu, D. Qian, P. Guduru, and A. Rosakis. Dynamic shear band propagation and micro-structure of adiabatic shear band. *Computer Methods in Applied Mechanics and Engineering*, 191(1-2):73–92, 2001. Micromechanics of Brittle Materials and Stochastic Analysis of Mechanical Systems.

[67] S. Li, W.-K. Liu, A. Rosakis, T. Belytschko, and W. Hao. Mesh-free galerkin simulations of dynamic shear band propagation and failure mode transition. *International Journal of Solids and Structures*, 39(5):1213 – 1240, 2002.

[68] X. Ling and T. Belytschko. Thermal softening induced plastic instability in rate-dependent materials. *Journal of the Mechanics and Physics of Solids*, 57:788–802, April 2009.

[69] W. Liu and C. McVeigh. Predictive multiscale theory for design of heterogeneous materials. *Computational Mechanics*, 42(2):147–170, 2008.

[70] B. Loret and J. Prevost. Dynamic strain localization in elasto-(visco-)plastic solids, part 1. general formulation and one-dimensional examples. *Computer Methods in Applied Mechanics and Engineering*, 83(3):247–273, 1990.

[71] D. Lovelock and H. Rund. *Tensors, Differential Forms, and Variational Principles*. Dover Books on Mathematics Series. Dover Publ., 1975.

[72] A. Marchand and J. Duffy. An experimental study of the formation process of adiabatic shear bands. *Journal of the Mechanics and Physics of Solids*, 38:251–238, 1988.

[73] C. McAuliffe and H. Waisman. Mesh insensitive formulation for initiation and growth of shear bands using mixed finite elements. *Computational Mechanics*, 51(5):807–823, 2013.

[74] C. McAuliffe and H. Waisman. A Pian-Sumihara type element for modeling shear bands at finite deformation. *Computational Mechanics*, pages 1–16, 2013.

[75] C. McAuliffe and H. Waisman. A unified model for metal failure capturing shear banding and fracture. *International Journal of Plasticity*, 65(0):131–151, 2015.

[76] C. McVeigh and W. Liu. Linking microstructure and properties through a predictive multiresolution continuum. *Computer Methods in Applied Mechanics and Engineering*, 197(41-42):3268 – 3290, 2008.

[77] C. McVeigh and W. Liu. Multiresolution continuum modeling of micro-void assisted dynamic adiabatic shear band propagation. *Journal of the Mechanics and Physics of Solids*, 58(2):187 – 205, 2010.

[78] J. Milano and P. Lembke. *IBM System Blue Gene Solution: Blue Gene/Q Hardware Overview and Installation Planning*. IBM Redbooks publication, 2013.

[79] D. Peirce, C. Shih, and A. Needleman. A tangent modulus method for rate dependent solids. *Computers and Structures*, 18(5):875–887, 1984.

[80] F. Pellegrini, J. Roman, and P. Amestoy. Hybridizing nested dissection and halo approximate minimum degree for efficient sparse matrix ordering. *Concurrency: Practice and Experience*, 12(2-3):69–84, 2000.

[81] T. Pian and K. Sumihara. Rational approach for assumed stress finite elements. *International Journal of Numerical Methods in Engineering*, 20:1685–1638, 1984.

[82] T. Pian and P. Tong. Relations between incompatible displacement model and hybrid stress model. *International Journal of Numerical Methods in Engineering*, 22:173–181, 1986.

[83] L. Piegl and W. Tiller. *The NURBS book.* Springer-Verlag, New York, 2nd edition, 1997.

[84] H. Quinney and G. Taylor. The emission of the latent energy due to previous cold working when a metal is heated. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 163(913):157–181, 1937.

[85] G. Ravichandran, A. Rosakis, J. Hodowany, and P. Rosakis. On the conversion of plastic work into heat during high strain rate deformation. *AIP Conference Proceedings*, 620(1):557–562, 2002.

[86] E. Reissner. On a variational theorem in elasticity. *Journal of Mathematical Physics*, 29(2):90–95, 1950.

[87] S. Rhebergen, G. Wells, R. Katz, and A. Wathen. Analysis of block preconditioners for models of coupled magma/mantle dynamics. *SIAM Journal on Scientific Computing*, 36(4):A1960–A1977, 2014.

[88] D. Rogers. *An Introduction to {NURBS} With Historical Perspective.* Morgan Kaufmann, San Francisco, 2001.

[89] Y. Saad. A flexible inner-outer preconditioned gmres algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[90] Y. Saad and M. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[91] J. Schur. Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind. *Journal für die reine und angewandte Mathematik*, 147:205–232, 1917.

[92] H. Schwarz. Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, 1870.

[93] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, New York, NY, USA, 1996.

[94] B. Smith, L. C. McInnes, E. Constantinescu, M. Adams, S. Balay, J. Brown, M. Knepley, and H. Zhang. PETSc's software strategy for the design space of composable extreme-scale solvers. Preprint ANL/MCS-P2059-0312, Argonne National Laboratory, 2012. DOE Exascale Research Conference, April 16-18, 2012, Portland, OR.

[95] I. Sokolnikoff. *Tensor analysis: theory and applications*. Applied mathematics series. Wiley, 1951.

[96] S. Tang, A. Kopacz, S. C. O'Keeffe, G. Olson, and W. Liu. Concurrent multiresolution finite element: formulation and algorithmic aspects. *Computational Mechanics*, 52(6):1265–1279, 2013.

[97] S. Tang, A. Kopacz, S. C. O'Keeffe, G. Olson, and W. Liu. Three-dimensional ductile fracture analysis with a hybrid multiresolution approach and microtomography. *Journal of the Mechanics and Physics of Solids*, 61(11):2108 – 2124, 2013.

[98] R. Taylor. *FEAP - - A Finite Element Analysis Program*, April 2011.

[99] T.J.R.Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, 1987.

[100] R. Tuminaro and C. Tong. Parallel smoothed aggregation multigrid: Aggregation strategies on massively parallel machines. In *Proceedings of the 2000 ACM/IEEE*

Conference on Supercomputing, Supercomputing '00, Washington, DC, USA, 2000. IEEE Computer Society.

[101] V. Tvergaard, A. Needleman, and K. Lo. Flow localization in the plane strain tensile test. *Journal of the Mechanics and Physics of Solids*, 29:115–142, 1981.

[102] H. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[103] G. Voyiadjis and F. Abed. A coupled temperature and strain rate dependent yield function for dynamic deformations of bcc metals. *International Journal of Plasticity*, 22(8):1398 – 1431, 2006. Special issue in honour of Dr. Kirk Valanis Valanis Issue.

[104] H. Waisman and L. Berger-Vergiat. An adaptive domain decomposition preconditioner for crack propagation problems modeled by XFEM. *International Journal for Multiscale Computational Engineering*, 11(6):633–654, 2013.

[105] T. Wright. *The Physics and Mathematics of Adiabatic Shear Bands*. Cambridge University Press, 2002.

[106] T. Wright and J. Walter. On stress collapse in adiabatic shear bands. *Journal of the Mechanics and Physics of Solids*, 35(6):701 – 720, 1987.

[107] M. Zhou, G. Ravichandran, and A. Rosakis. Dynamically propagating shear bands in impact-loaded prenotched plates—ii. numerical simulations. *Journal of the Mechanics and Physics of Solids*, 44(6):1007–1032, 1996.

[108] M. Zhou, A. J. Rosakis, and G. Ravichandran. Dynamically propagating shear bands in impact-loaded prenotched plates- i. experimental investigations of temperature signatures and propagation speed. *Journal of the Mechanics and Physics of Solids*, 44(6):981–1006, 1996.

[109] T. Zhou and X. Xie. A unified analysis for stress/strain hybrid methods of high performance. *Computer Methods in Applied Mechanics and Engineering*, 191:4619–4640, 2002.

[110] O. Zienkiewicz and R. Taylor. *The Finite Element Method*, volume Volume 1. Butterworth Heinemann, 2000.

Appendices

Appendix A

Newmark method for time integration

Assuming that the solution vector and its first and second time derivatives are know at time step n ($\mathbf{x}_n$, $\dot{\mathbf{x}}_n$, $\ddot{\mathbf{x}}_n$) and that these variables are the solution of a second order ODE of the form

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{x} = \mathbf{F}, \tag{A.1}$$

then using a weighted mean $\mathbf{x}$ and $\dot{\mathbf{x}}$ can be expressed at time $n+1$ as

$$\begin{array}{l} \dot{\mathbf{x}}_{n+1} = \dot{\mathbf{x}}_n + \Delta t\left[(1-\gamma)\,\ddot{\mathbf{x}}_n + \gamma\ddot{\mathbf{x}}_{n+1}\right] \\ \mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\dot{\mathbf{x}}_n + \frac{\Delta t^2}{2}\left[(1-2\beta)\,\ddot{\mathbf{x}}_n + \beta\ddot{\mathbf{x}}_{n+1}\right] \end{array}. \tag{A.2}$$

To facilitate some implementation aspects of the finite element method (A.2) can be recast to express $\dot{\mathbf{x}}_{n+1}$ and $\ddot{\mathbf{x}}_{n+1}$ as a function of $\mathbf{x}_{n+1}$, $\mathbf{x}_n$, $\dot{\mathbf{x}}_n$ and $\ddot{\mathbf{x}}_n$ as follows

$$\begin{array}{l} \ddot{\mathbf{x}}_{n+1} = \dfrac{1}{\beta\Delta t^2}(\mathbf{x}_{n+1} - \mathbf{x}_n) - \dfrac{1}{\beta\Delta t}\dot{\mathbf{x}}_n - \left(\dfrac{1}{2\beta} - 1\right)\ddot{\mathbf{x}}_n \\ \dot{\mathbf{x}}_{n+1} = \dfrac{\gamma}{\beta\Delta t}(\mathbf{x}_{n+1} - \mathbf{x}_n) - \left(\dfrac{\gamma}{\beta} - 1\right)\dot{\mathbf{x}}_n - \left(\dfrac{\gamma}{2\beta} - 1\right)\Delta t\ddot{\mathbf{x}}_n \end{array}. \tag{A.3}$$

Using a predictor-corrector approach $\dot{\mathbf{x}}_{n+1}$ and $\ddot{\mathbf{x}}_{n+1}$ can be predicted as

$$\begin{aligned}\ddot{\tilde{\mathbf{x}}}_{n+1} &= -\frac{1}{\beta\Delta t^2}\mathbf{x}_n - \frac{1}{\beta\Delta t}\dot{\mathbf{x}}_n - \left(\frac{1}{2\beta} - 1\right)\ddot{\mathbf{x}}_n \\ \dot{\tilde{\mathbf{x}}}_{n+1} &= -\frac{\gamma}{\beta\Delta t}\mathbf{x}_n - \left(\frac{\gamma}{\beta} - 1\right)\dot{\mathbf{x}}_n - \left(\frac{\gamma}{2\beta} - 1\right)\Delta t\ddot{\mathbf{x}}_n\end{aligned}. \tag{A.4}$$

$\mathbf{x}_{n+1}$ is then obtained by solving the following equation

$$\left(\frac{1}{\beta\Delta t^2}\mathbf{M} + \frac{\gamma}{\beta\Delta t}\mathbf{C} + \mathbf{K}\right)\mathbf{x}_{n+1} = \mathbf{F}_{n+1} - \mathbf{M}\ddot{\tilde{\mathbf{x}}}_{n+1} - \mathbf{C}\dot{\tilde{\mathbf{x}}}_{n+1}. \tag{A.5}$$

Finally $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$ are updated during the correction phase

$$\begin{aligned}\ddot{\mathbf{x}}_{n+1} &= \ddot{\tilde{\mathbf{x}}}_{n+1} + \frac{1}{\beta\Delta t^2}\mathbf{x}_{n+1} \\ \dot{\mathbf{x}}_{n+1} &= \dot{\tilde{\mathbf{x}}}_{n+1} + \frac{\gamma}{\beta\Delta t}\mathbf{x}_{n+1}\end{aligned} \tag{A.6}$$

Appendix B

Explicit derivation of a NURBS function basis

$\mathcal{C}^0$ basis functions for knot vector $\Xi = 0, 0, 0, 0.5, 1, 1, 1$

$$N_{0,0} = N_{1,0} = N_{4,0} = 0 \tag{B.1}$$

$$N_{2,0} = \begin{cases} 1 & \xi \in [0.0; 0.5[\\ 0 & \xi \in [0.5; 1.0[\end{cases} \tag{B.2}$$

$$N_{3,0} = \begin{cases} 0 & \xi \in [0.0; 0.5[\\ 1 & \xi \in [0.5; 1.0[\end{cases} \tag{B.3}$$

The Cox-de Boor recursion formula allows to compute higher order Bernstein polynomials that can be used as function basis too.

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{B.4}$$

Using the $\mathcal{C}^0$ function basis and the Cox-de Boor formula the $\mathcal{C}^1$ function basis is calculated as follows:

$$\begin{aligned}
N_{1,1}(\xi) &= \tfrac{\xi-\xi_1}{\xi_2-\xi_1}N_{1,0}(\xi)+\tfrac{\xi_3-\xi}{\xi_3-\xi_2}N_{2,0}(\xi)\\
&= \begin{cases} \frac{0.5-\xi}{0.5-0.0}\times 1 & \xi\in[0.0,0.5[\\ 0 & \xi\in[0.5,1.0[\end{cases}\\
&= \begin{cases} (1-2\xi) & \xi\in[0.0,0.5[\\ 0 & \xi\in[0.5,1.0[\end{cases}
\end{aligned} \tag{B.5}$$

$$\begin{aligned}
N_{2,1}(\xi) &= \tfrac{\xi-\xi_2}{\xi_3-\xi_2}N_{2,0}(\xi)+\tfrac{\xi_4-\xi}{\xi_4-\xi_3}N_{3,0}(\xi)\\
&= \begin{cases} \frac{\xi-0.0}{0.5-0.0}\times 1 & \xi\in[0.0,0.5[\\ \frac{1.0-\xi}{1.0-0.5}\times 1 & \xi\in[0.5,1.0[\end{cases}\\
&= \begin{cases} 2\xi & \xi\in[0.0,0.5[\\ 2(1-\xi) & \xi\in[0.5,1.0[\end{cases}
\end{aligned} \tag{B.6}$$

$$\begin{aligned}
N_{3,1}(\xi) &= \tfrac{\xi-\xi_3}{\xi_4-\xi_3}N_{3,0}(\xi)+\tfrac{\xi_5-\xi}{\xi_5-\xi_4}N_{4,0}(\xi)\\
&= \begin{cases} 0 & \xi\in[0.0,0.5[\\ \frac{\xi-0.5}{1.0-0.5}\times 1 & \xi\in[0.5,1.0[\end{cases}\\
&= \begin{cases} 0 & \xi\in[0.0,0.5[\\ 2\xi-1 & \xi\in[0.5,1.0[\end{cases}
\end{aligned} \tag{B.7}$$

Reapplying the Cox-de Boor a second time we obtain the quadratic function basis desired

$$\begin{aligned}
N_{0,2} &= \tfrac{\xi-\xi_0}{\xi_2-\xi_0}N_{0,1}(\xi)+\tfrac{\xi_3-\xi}{\xi_3-\xi_1}N_{1,1}(\xi)\\
&= \tfrac{0.5-\xi}{0.5-0.0}\begin{cases} 1-2\xi & \xi\in[0.0;0.5[\\ 0 & \xi\in[0.5;1.0[\end{cases}\\
&= \begin{cases} (1-2\xi)^2 & \xi\in[0.0,0.5[\\ 0 & \xi\in[0.5,1.0[\end{cases}
\end{aligned} \tag{B.8}$$

$$\begin{aligned}
N_{1,2} &= \tfrac{\xi-\xi_1}{\xi_3-\xi_1} N_{1,1}(\xi) + \tfrac{\xi_4-\xi}{\xi_4-\xi_2} N_{2,1}(\xi) \\
&= \tfrac{\xi-0.0}{0.5-0.0} \begin{cases} 1-2\xi & \xi \in [0.0; 0.5[\\ 0 & \xi \in [0.5; 1.0[\end{cases} + \tfrac{1.0-\xi}{1.0-0.0} \begin{cases} 2\xi & \xi \in [0.0; 0.5[\\ 2(1-\xi) & \xi \in [0.5; 1.0[\end{cases} \\
&= \begin{cases} 2\xi(2-3\xi) & \xi \in [0.0, 0.5[\\ 2(1-\xi)^2 & \xi \in [0.5, 1.0[\end{cases}
\end{aligned} \tag{B.9}$$

$$\begin{aligned}
N_{2,2} &= \tfrac{\xi-\xi_2}{\xi_4-\xi_2} N_{2,1}(\xi) + \tfrac{\xi_5-\xi}{\xi_5-\xi_3} N_{3,1}(\xi) \\
&= \tfrac{\xi-0.0}{1.0-0.0} \begin{cases} 2\xi & \xi \in [0.0; 0.5[\\ 2(1-\xi) & \xi \in [0.5; 1.0[\end{cases} + \tfrac{1.0-\xi}{1.0-0.5} \begin{cases} 0 & \xi \in [0.0; 0.5[\\ 2\xi-1 & \xi \in [0.5; 1.0[\end{cases} \\
&= \begin{cases} 2\xi^2 & \xi \in [0.0, 0.5[\\ 2(1-\xi)(3\xi-1) & \xi \in [0.5, 1.0[\end{cases}
\end{aligned} \tag{B.10}$$

$$\begin{aligned}
N_{3,2} &= \tfrac{\xi-\xi_3}{\xi_5-\xi_3} N_{3,1}(\xi) + \tfrac{\xi_6-\xi}{\xi_6-\xi_4} N_{4,1}(\xi) \\
&= \tfrac{\xi-0.5}{1.0-0.5} \begin{cases} 0 & \xi \in [0.0; 0.5[\\ 2\xi-1 & \xi \in [0.5; 1.0[\end{cases} \\
&= \begin{cases} 0 & \xi \in [0.0, 0.5[\\ (2\xi-1)^2 & \xi \in [0.5, 1.0[\end{cases}
\end{aligned} \tag{B.11}$$

The derivatives of the function basis can be computed using a recusrive formula somewhat similar to the Cox-de Boor recursion

$$\frac{dN_{i,p}}{d\xi}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_i + 1} N_{i+1,p-1}(\xi) \tag{B.12}$$

This formula can be easily verified using the quadratic function basis ($N_{0,2}, N_{1,2}, N_{2,2}, N_{3,2}$)

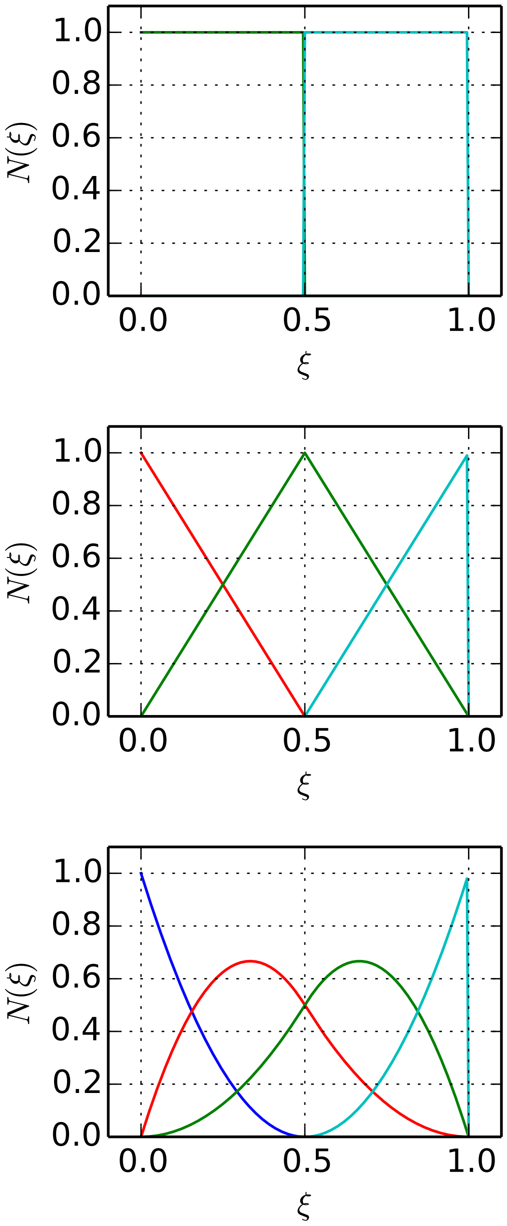


Figure B.1: Function basis associated to knot vector $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ of order $\mathcal{C}^0$, $\mathcal{C}^1$ and $\mathcal{C}^2$

that we just computed. First let us compute the derivatives of this function basis using the usual derivation rules

$$\begin{aligned}
\frac{dN_{0,2}}{d\xi}(\xi) &= \begin{cases} 4(2\xi-1) & \xi \in [0.0, 0.5] \\ 0 & \xi \in [0.5, 1.0] \end{cases} \\
\frac{dN_{1,2}}{d\xi}(\xi) &= \begin{cases} 4(1-3\xi) & \xi \in [0.0, 0.5] \\ 4(\xi-1) & \xi \in [0.5, 1.0] \end{cases} \\
\frac{dN_{2,2}}{d\xi}(\xi) &= \begin{cases} 4\xi & \xi \in [0.0, 0.5] \\ 4(2-3\xi) & \xi \in [0.5, 1.0] \end{cases} \\
\frac{dN_{3,2}}{d\xi}(\xi) &= \begin{cases} 0 & \xi \in [0.0, 0.5] \\ 4(2\xi-1) & \xi \in [0.5, 1.0] \end{cases}
\end{aligned} \tag{B.13}$$

Using equation (B.12) we obtain the following

$$\begin{aligned}
\frac{dN_{0,2}}{d\xi}(\xi) &= \frac{2}{\xi_2-\xi_0}N_{0,1}(\xi) - \frac{2}{\xi_3-\xi_1}N_{1,1}(\xi) \\
&= \frac{-2}{0.5-0.0}\begin{cases} 1-2\xi & \xi \in [0.0, 0.5[\\ 0 & \xi \in [0.5, 1.0[\end{cases} \\
&= \begin{cases} 4(2\xi-1) & \xi \in [0.0, 0.5[\\ 0 & \xi \in [0.5, 1.0[\end{cases}
\end{aligned} \tag{B.14}$$

$$\begin{aligned}
\frac{dN_{1,2}}{d\xi}(\xi) &= \frac{2}{\xi_3-\xi_1}N_{1,1}(\xi) - \frac{2}{\xi_4-\xi_2}N_{2,1}(\xi) \\
&= \frac{2}{0.5-0.0}\begin{cases} 1-2\xi & \xi \in [0.0, 0.5[\\ 0 & \xi \in [0.5, 1.0[\end{cases} - \frac{2}{1.0-0.0}\begin{cases} 2\xi & \xi \in [0.0, 0.5[\\ 2(1-\xi) & \xi \in [0.5, 1.0[\end{cases} \\
&= \begin{cases} 4(1-3\xi) & \xi \in [0.0, 0.5[\\ 4(\xi-1) & \xi \in [0.5, 1.0[\end{cases}
\end{aligned} \tag{B.15}$$

$$\begin{aligned}
\frac{dN_{2,2}}{d\xi}(\xi) &= \frac{2}{\xi_4-\xi_2}N_{2,1}(\xi) - \frac{2}{\xi_5-\xi_3}N_{3,1}(\xi) \\
&= \frac{2}{1.0-0.0}\begin{cases} 2\xi & \xi \in [0.0, 0.5[\\ 2(1-\xi) & \xi \in [0.5, 1.0[\end{cases} - \frac{2}{1.0-0.5}\begin{cases} 0 & \xi \in [0.0, 0.5[\\ 2\xi - 1 & \xi \in [0.5, 1.0[\end{cases} \\
&= \begin{cases} 4\xi & \xi \in [0.0, 0.5[\\ 4(2-3\xi) & \xi \in [0.5, 1.0[\end{cases}
\end{aligned} \tag{B.16}$$

$$\begin{aligned}
\frac{dN_{3,2}}{d\xi}(\xi) &= \frac{2}{\xi_5-\xi_3}N_{3,1}(\xi) - \frac{2}{\xi_6-\xi_4}N_{4,1}(\xi) \\
&= \frac{2}{1.0-0.5}\begin{cases} 0 & \xi \in [0.0, 0.5[\\ 2\xi - 1 & \xi \in [0.5, 1.0[\end{cases} \\
&= \begin{cases} 0 & \xi \in [0.0, 0.5[\\ 4(2\xi-1) & \xi \in [0.5, 1.0[\end{cases}
\end{aligned} \tag{B.17}$$

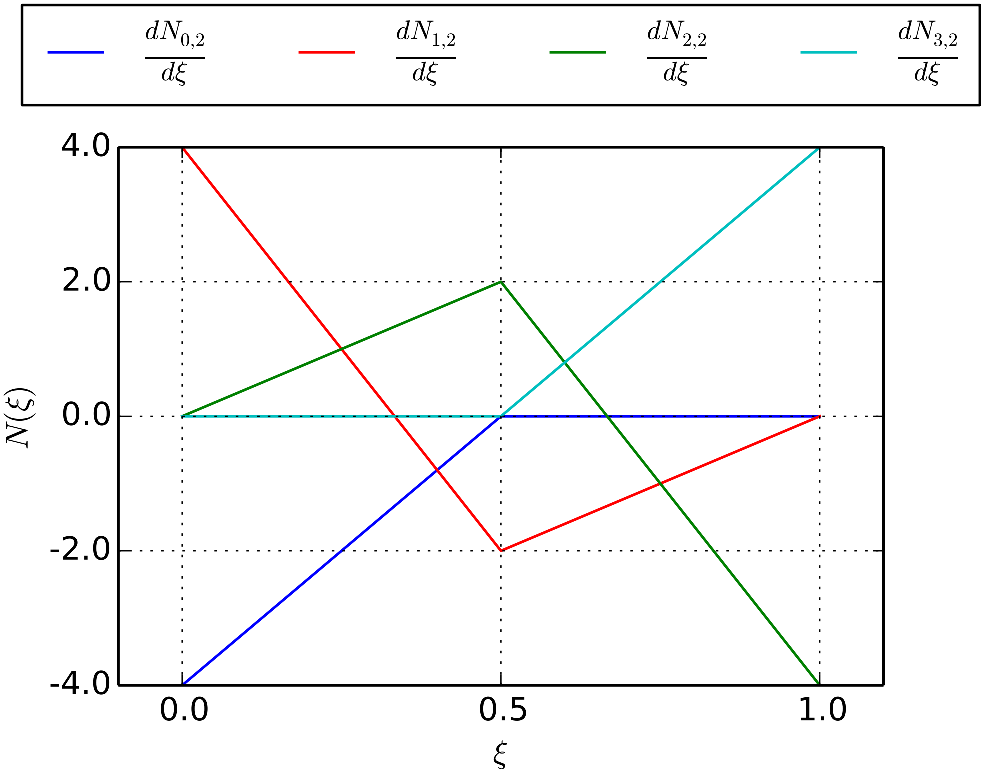


Figure B.2: Derivatives of the function basis associated to knot vector $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ of order $\mathcal{C}^2$

Appendix C

Multiplicative Schwarz method

The multiplicative Schwarz method computes an approximate inverse of $\mathbf{A}$ that we will write $\mathbf{A}^{\#}$, using the successive inverse of $\mathbf{A}_{00}$ and $\mathbf{A}_{11}$. Hence this method requires both $\mathbf{A}_{00}$ and $\mathbf{A}_{11}$ to be invertible matrices. Using the following notations

$$\mathbf{B}_0 = \begin{bmatrix} \mathbf{A}_{00}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \quad \text{and} \quad \mathbf{B}_1 = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{11}^{-1} \end{bmatrix}, \tag{C.1}$$

one iteration of this method between steps n and $n+1$ is derived as follows

$$\mathbf{r}^n = \mathbf{b} - \mathbf{A}\mathbf{u}^n, \tag{C.2}$$

$$\mathbf{u}^{n+1/2} = \mathbf{u}^n + \mathbf{B}_0\mathbf{r}^n, \tag{C.3}$$

$$\mathbf{r}^{n+1/2} = \mathbf{b} - \mathbf{A}\mathbf{u}^{n+1/2}, \tag{C.4}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{n+1/2} + \mathbf{B}_1\mathbf{r}^{n+1/2}. \tag{C.5}$$

an approximate solution to (3.1) can be computed as

$$\mathbf{u} = [\mathbf{B}_0 + \mathbf{B}_1 - \mathbf{B}_1\mathbf{A}\mathbf{B}_0]\,\mathbf{b} \tag{C.6}$$

and the multiplicative Schwarz operator can be defined as

$$\mathbf{A}^{\#} = \mathbf{B}_0 + \mathbf{B}_1 - \mathbf{B}_1\mathbf{A}\mathbf{B}_0 \tag{C.7}$$

which can be expanded to obtain the final expression

$$\begin{aligned}\mathbf{A}^{\#} &= \begin{bmatrix}\mathbf{A}_{00}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{1}\end{bmatrix} + \begin{bmatrix}\mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_{11}^{-1}\end{bmatrix} - \begin{bmatrix}\mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_{11}^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{A}_{00} & \mathbf{A}_{01}\\ \mathbf{A}_{10} & \mathbf{A}_{11}\end{bmatrix}\begin{bmatrix}\mathbf{A}_{00}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{1}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{A}_{00}^{-1} & -\mathbf{A}_{01}\\ -\mathbf{A}_{11}^{-1}\mathbf{A}_{10}\mathbf{A}_{00}^{-1} & \mathbf{A}_{11}^{-1}\end{bmatrix}\end{aligned} \tag{C.8}$$

Remark 9: Usualy the Schwarz method is used in Richardson iterations to obtain a converged value for $\mathbf{u}$ and equation (C.6) is applied to a residual ($\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{u}$) to compute an update $\delta\mathbf{u}$. ■

Remark 10: As mentioned above, $\mathbf{u}_0$ and $\mathbf{u}_1$ are complement of each other in $\mathbf{u}$, hence $\mathbf{A}_{00}$ and $\mathbf{A}_{11}$ are not overlapping, this allows us to compute the final expression of $\tilde{\mathbf{A}}$. In the case of overlapping matrices $\mathbf{A}_{00}$ and $\mathbf{A}_{11}$ the block matrix multiplications do not hold. ■

Appendix D

Compressive shear band GMSH input

```
Mesh.RecombineAll=1;


//Mesh size control
h = 2.0e-6;
h0 = h/1; // divide by 1, 2 or 3 for coa, med or fin global refinement
h1 = 5*h0;
h2 = 2*h0;
h3 = h0/2; // divide by 2, 4 or 6 for coa, med or fin local refinement


//Points
Point(1) = {2.5e-5, 0, 0, h2};
Point(2) = {1.0e-4, 0, 0, h1};
Point(3) = {1.0e-4, 7.5e-5, 0, h2};
Point(4) = {7.5e-5, 1.0e-4, 0, h2};
Point(5) = {0.0, 1.0e-4, 0, h1};
Point(6) = {0.0, 2.5e-5, 0, h2};
Point(7) = {7.8e-5, 7.8e-5, 0, h3};
Point(8) = {2.2e-5, 2.2e-5, 0, h3};
Point(9) = {2.5e-5, 1.8e-5, 0, h3};
Point(10) = {2.5e-5, 2.2e-5, 0, h3};
Point(11) = {2.2e-5, 2.5e-5, 0, h3};
Point(12) = {7.5e-5, 8.2e-5, 0, h3};
Point(13) = {7.5e-5, 7.8e-5, 0, h3};
Point(14) = {7.8e-5, 7.5e-5, 0, h3};

//Lines
Line(1) = {10, 9};
Line(2) = {9, 1};
Line(3) = {1, 2};
Line(4) = {2, 3};
Line(5) = {3, 14};
Line(6) = {14, 9};
Line(7) = {13, 12};
Line(8) = {12, 4};
Line(9) = {4, 5};
Line(10) = {5, 6};
Line(11) = {6, 11};
Line(12) = {11, 12};
Circle(13) = {13, 7, 14};
Circle(14) = {11, 8, 10};

//Surfaces
Line Loop(21) = {2, 3, 4, 5, 6};
Plane Surface(31) = {21};
Line Loop(22) = {8, 9, 10, 11, 12};
```

```
Plane Surface(32) = {22};

Line Loop(23) = {-13, 7, -12, 14, 1, -6};

Plane Surface(33) = {23};

Physical Surface(50) = {31, 32, 33};
```