

# Multiscale methods for random composite materials

submitted by

Anhadjeet Singh Sandhu

for the degree of Doctor of Philosophy in Engineering

of the

University of Exeter

College of Engineering, Mathematics and Physical Sciences

September 2019

## **COPYRIGHT**

Attention is drawn to the fact that copyright of this thesis rests with its author. This copy of the thesis has been supplied on the condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Signature of Author .....

Anhadjeet Singh Sandhu



---

# ABSTRACT

Simulation of material behaviour is not only a vital tool in accelerating product development and increasing design efficiency but also in advancing our fundamental understanding of materials. While homogeneous, isotropic materials are often simple to simulate, advanced, anisotropic materials pose a more sizeable challenge. In simulating entire composite components such as a 25m aircraft wing made by stacking several 0.25mm thick plies, finite element models typically exceed millions or even a billion unknowns. This problem is exacerbated by the inclusion of sub-millimeter manufacturing defects for two reasons. Firstly, a finer resolution is required which makes the problem larger. Secondly, defects introduce randomness. Traditionally, this randomness or uncertainty has been quantified heuristically since commercial codes are largely unsuccessful in solving problems of this size. This thesis develops a rigorous uncertainty quantification (UQ) framework permitted by a state of the art finite element package **dune-composites**, also developed here, designed for but not limited to composite applications. A key feature of this open-source package is a robust, parallel and scalable preconditioner **GenEO**, that guarantees constant iteration counts independent of problem size. It boasts near perfect scaling properties in both, a strong and a weak sense on over 15,000 cores. It is numerically verified by solving industrially motivated problems containing upwards of 200 million unknowns.

Equipped with the capability of solving expensive models, a novel stochastic framework is developed to quantify variability in part performance arising from localized out-of-plane defects. Theoretical part strength is determined for independent samples drawn from a distribution inferred from B-scans of wrinkles. Supported by literature, the results indicate a strong dependence between maximum misalignment angle and strength knockdown based on which an engineering model is presented to allow rapid estimation of residual strength bypassing expensive simulations. The engineering model itself is built from a large set of simulations of residual strength, each of which is computed using the following two step approach. First, a novel parametric representation

of wrinkles is developed where the spread of parameters defines the wrinkle distribution. Second, expensive forward models are only solved for independent wrinkles using `dune-composites`.

Besides scalability the other key feature of `dune-composites`, the `GenEO` coarse space, doubles as an excellent multiscale basis which is exploited to build high quality reduced order models that are orders of magnitude smaller. This is important because it enables multiple coarse solves for the cost of one fine solve. In an MCMC framework, where many solves are wasted in arriving at the next independent sample, this is a sought after quality because it greatly increases effective sample size for a fixed computational budget thus providing a route to high-fidelity UQ.

This thesis exploits both, new solvers and multiscale methods developed here to design an efficient Bayesian framework to carry out previously intractable (large scale) simulations calibrated by experimental data. These new capabilities provide the basis for future work on modelling random heterogeneous materials while also offering the scope for building virtual test programs including nonlinear analyses, all of which can be implemented within a probabilistic setting.



---

# ACKNOWLEDGEMENTS

I would like to thank Dr. Timothy Dodwell for taking on the task of supervising my PhD and doing so in a cheerful and patient manner. I have learnt much from his wealth of knowledge about all things mathematical but I am most grateful for his above and beyond approach to supervising me.

Special mention must be given to a number of people. Firstly, Dr. Anne Reinarz, who helped me with running simulations on supercomputers. Secondly, Dr. Samuel Erland for making any downtime productive and entertaining. I would also like to thank the X-AT group for their support.

Finally, I would like to thank my parents, my sister Meher, and N. Each have supported me through my PhD in their own ways without which this would not have been possible. Thank you for listening even when you didn't have to. I owe this to you.

---

# CONTENTS

List of Figures . . . . .	8
List of Tables . . . . .	17
Nomenclature . . . . .	18
<b>1 Introduction</b>	<b>24</b>
1.1 Outline of this thesis . . . . .	26
<b>2 Literature review</b>	<b>31</b>
2.1 Wrinkles and non destructive testing . . . . .	36
2.2 Uncertainty quantification . . . . .	39
2.2.1 Markov chain Monte Carlo . . . . .	41
2.3 Solvers . . . . .	51
2.3.1 Iterative solvers . . . . .	52
2.4 Preconditioning . . . . .	54
2.4.1 Multilevel preconditioners . . . . .	56
2.4.2 Domain decomposition preconditioners . . . . .	58
2.5 Modern multiscale methods . . . . .	64
2.6 Concluding remarks . . . . .	66
<b>3 dune-composites - an open source, high performance package for solving large-scale anisotropic elasticity problems</b>	<b>69</b>
3.1 Introduction - Why do we need dune-composites? . . . . .	69
3.2 Preliminaries : Anisotropic elasticity equations and their finite element discretisation . . . . .	71
3.3 A robust, scalable, parallel iterative solver for composite structures . . . . .	73

---

3.3.1	Krylov subspace methods preconditioned with two-level additive Schwarz methods . . . . .	73
3.3.2	A robust coarse space via Generalised Eigenproblems in the Overlaps (GenEO) . . . . .	75
3.3.3	Implementation of GenEO on a High Performance Computer . . . .	78
3.4	Using and extending dune-composites . . . . .	81
3.4.1	Defining a Model . . . . .	81
3.4.2	Internals of dune-composites . . . . .	82
3.5	Examples . . . . .	83
3.5.1	Example 1: A flat composite plate . . . . .	84
3.5.2	Example 2 : Corner unfolding – validation & performance comparison with ABAQUS (up to 32 cores) . . . . .	88
3.5.3	Example 3 : Large composite structure – parallel efficiency of dune-composites (up to 15,360 cores) . . . . .	90
3.5.4	Subsurface flow applications: Strong scaling for the SPE10 benchmark	94
3.6	Concluding remarks . . . . .	97
<b>4</b>	<b>A Bayesian framework for assessing the strength distribution of composite structures with random defects</b>	<b>99</b>
4.1	Introduction . . . . .	99
4.2	Bayesian approach to construct defect distributions from measured data . .	100
4.2.1	Parameterizing a wrinkle defect . . . . .	100
4.2.2	Posterior Sampling using a Metropolis-Hastings algorithm . . . . .	101
4.2.3	Monte Carlo Simulations . . . . .	105
4.3	Industrially Motivated Case Study . . . . .	106
4.3.1	Model Problem and its industrial application . . . . .	107
4.3.2	Extracting wrinkle data from B-Scans using Multiple Field Image Analysis (MFIA) . . . . .	108
4.3.3	Defining a Wrinkle, Prior and likelihood definition . . . . .	110
4.3.4	Finite Element Modelling . . . . .	113
4.4	Results . . . . .	114
4.4.1	Bayesian Sampling of wrinkles . . . . .	114
4.4.2	Monte Carlo simulations . . . . .	116
4.4.3	An ‘engineering model’ for Corner Bend Strength . . . . .	117
4.5	Concluding remarks . . . . .	119

---

---

<b>5</b>	<b>Multiscale methods for composites</b>	<b>122</b>
5.1	What is a good multiscale model? . . . . .	122
5.2	Preliminaries . . . . .	126
5.2.1	Customized coarse space - GenEO . . . . .	128
5.2.2	Partition of Unity . . . . .	129
5.3	Numerical experiments . . . . .	131
5.4	Concluding remarks . . . . .	144
<b>6</b>	<b>Data-driven multiscale models for high dimensional MCMC</b>	<b>148</b>
6.1	Introduction . . . . .	148
6.2	Standard Markov Chain Monte Carlo . . . . .	150
6.3	Multiscale Markov Chain Monte Carlo . . . . .	151
6.3.1	Multiscale model . . . . .	154
6.3.2	The algorithm . . . . .	155
6.4	Numerical experiments . . . . .	155
6.4.1	Parallel implementation . . . . .	166
6.5	Concluding remarks . . . . .	168
<b>7</b>	<b>Concluding remarks and future work</b>	<b>170</b>
7.1	Future work . . . . .	171
7.1.1	Nonlinear mechanics . . . . .	171
7.1.2	Grids, subdomains and PoUs . . . . .	177
<b>A</b>	<b>Code blocks for chapter 3</b>	<b>180</b>
	<b>References</b>	<b>183</b>

---

# LIST OF FIGURES

2-1	The traditional approach to new aircraft design where each level of the test pyramid signifies the number of physical tests conducted which reduces as we move upwards with only one or two tests being conducted at the component scale [13]. . . . .	32
2-2	Micrograph of polycrystalline metal showing individual grains composed of a crystal lattice each [14]. . . . .	32
2-3	(a) and (b) show two types of unit cell that are representative of perfectly packed fibres (after [26]). (Below) shows an SEM picture of a unidirectional composite ply. Unit cells (a) and (b) are poor representations of reality. . . . .	34
2-4	Scale separation distinguishing micro from macro scale. The range of scales introduces significant complexity in the analysis of aerospace composites. From Left to Right: fibre/resin scale ( $5\mu\text{m}$ ), ply scale (0.25mm), laminate scale (5-30mm) to the structure ( $> 1\text{m}$ )(Composite Fan Blade). . . . .	35
2-5	Ultrasonic B-scans showing out of plane fibre waviness or wrinkles. . . .	37
2-6	Sequential plots of a sample Markov chain for a known posterior. (From left-right) 1. The first 10 samples of the chain showing clearly where some proposals have been rejected and previous samples carried forward; 2. 100 samples of the same chain. At this point we can say burn-in is complete; 3. 1000 samples drawn from the posterior show a well mixed chain. 4. 10000 samples showing all the correlated samples drawn. We note this example does not involve any model solves and it simply demonstrating the Markov chain process, in a conceptual way. . . . .	45

---

2-7	Sequential plots with 4 Markov chains exploring a more complicated yet known posterior given in Eq. (2.6). The acceptance ratio of all chains is near the theoretical optimum of 0.23 [82] for a step size of $\beta = 0.27$ . . .	47
2-8	Demonstration of the coarsening or aggregation process in AMG. Left: Isotropic material where nodes within a radius are strongly connected and have similar solutions so neighbouring nodes can be collapsed onto one node. Right: A stiffening inclusion (in blue) makes it difficult to design an effective aggregation scheme. The nodes in green are strongly and non-locally connected. The dashed red line shows a previously existing strong connection broken by the inclusion. If we see this inclusion as a single carbon fibre in a matrix, the challenge of aggregating degrees of freedom within even a single ply becomes obvious. . . . .	58
2-9	(Left) Non overlapping partitions on a unit cube. (Right) A subdomain with overlaps shared by neighbours. . . . .	59
2-10	Two subdomain decomposition of a 1-D example showing partition of unity for the minimal overlap case. The Dirichlet subdomain boundaries have zero contribution to the solution and since there are no other shared nodes, the additive Schwarz reduces to block Jacobi algorithm. . . . .	61
2-11	Left: Dependency of additive Schwarz on overlap for a fixed number of subdomains (32 in this test); Right: Dependency of restrictive additive Schwarz on overlap size. The only difference in comparison to the left plot is the inclusion of partition of unity in the construction of the preconditioner. The overlap is given in number of elements that a subdomain is extended by. . . . .	61
2-12	Left: In the additive Schwarz test, the overlap is fixed in number of elements while subdomain count increased. From the plot, at least 64 subdomains are required for convergence because the overlap, as a percentage of subdomain size, is insufficient for larger subdomains. Therefore, none of the solutions converge to the required tolerance; Right: The RAS test shows slight improvement in convergence with the addition of partition of unity. . . . .	62
2-13	Additive Schwarz example with overlap fixed in terms of percentage of subdomain size which illustrates the paradoxical nature of two competing requirements i.e. overlap and number of subdomains. As the subdomain count increases, global communication becomes more necessary, so convergence deteriorates. Once again, AS on the left is compared with RAS on the right. . . . .	62

---

---

2-14	One dimensional composite problem with resin rich (red) and carbon fibre (blue) dominated zones. $E_{resin} = 10\text{GPa}$ and $E_{composite} = 162\text{GPa}$ .	63
2-15	The two-norm of the residual against CG iterations for a 1D composite as shown in Fig. 2-14 using no coarse space (blue line) and Nicolaides coarse space (red line). One-level AS does not converge for this composite problem but with the Nicolaides coarse correction, the solution converges in 22 iterations. . . . .	63
3-1	(Left) Domain $\Omega$ partitioned into non-overlapping subdomains $\Omega'_j$ where colouring differentiates independent subdomains. (Middle) Shows overlapping subdomain $\Omega_j$ with a single layer of overlap ( $O = 1$ ). Overlap region $\Omega_j^\circ$ is shown in white. Transparent red regions show cells of the grid which belong to 'nearest neighbour' processors. (Right) Shows partition of unity (PoU) operator $\Xi_j$ on a single processor, defined as in (3.12). . . . .	74
3-2	Code structure. . . . .	81
3-3	Visualisation of results for <b>Example01a</b> using PARAVIEW (left) Visual output of laminate and stacking sequence using <code>plotProperties()</code> function (right) Visualisation of solution, in deformed coordinates (scalar factor of displacement is 4). . . . .	85
3-4	(Left) The eigenvectors corresponding to the first nine non-zero eigenvalues on a subdomain with no global Dirichlet boundary. (Right) The reduction of the residual against CG iterations for <b>Example01b</b> using no coarse space, only zero energy modes (ZEM) and the full <b>GenEO</b> coarse space. . . . .	87
3-5	(Left) Diagram of the corner bend specimen with resin edge treatment. (Right) Cross section of the corner showing the loading conditions. . . .	88
3-6	(Left & Middle) Stresses (in MPa) as functions of the distance $r$ from the outer radius at the apex of the curve, at 2.156mm from the edge of the resin-edge-treated laminate ( <b>dune-composites</b> , solid blue; ABAQUS, dotted red). The background colours indicate the stacking sequence: $+45^\circ$ =red, $-45^\circ$ =blue, $90^\circ$ =green, $0^\circ$ =yellow. (Right) Cost comparison between the sparse direct solver implemented in ABAQUS and the iterative preconditioned CG solver in <b>dune-composites</b> . . . . .	90

---

---

3-7	(Left) Geometry of the wingbox with dimensions; the colouring shows the number of eigenmodes used in <b>GenEO</b> in each of the subdomains of Setup 6 in Table 3.2. (Right) Close-up plot of the corner of the wingbox using <code>plotProperties()</code> , which shows the wrinkle and the inter-lacing of the different stacking sequences in the corner, cover and spar regions.	91
3-8	FE solution for Example 3: (Left) Overall deformation of the wingbox with colours showing the magnitude of the displacements in cm. (Right) Camanho failure criterion (3.22) in a close-up of the corner containing the wrinkle defect. . . . .	92
3-9	Parallel performance of <b>dune-composites</b> on ARCHER: (Left) A weak scaling test, as summarised in Table 3.2. (Right) A strong scaling test using Setup 5 in Table 3.2, with the dashed line showing perfect scaling, as summarised in Table 3.3. . . . .	94
3-10	(Top) Logarithm of the permeability field $\mathbf{K}$ for the SPE10 benchmark, from bottom to top: $K_x$ , $K_y$ and $K_z$ . (Bottom) A strong scaling test using the SPE10 dataset, with the dashed line showing perfect scaling. .	96
4-1	Illustration of Eq. (4.1) showing the transformation from pristine to defective state for a 39 ply composite with a representative stacking sequence . . . . .	102
4-2	Representation of Bayesian approach in a simplified 2D parameter space for a known likelihood function (a variant of the Rosenbrock function). (Left) Sampling from isotropic Gaussian prior $\pi_0(\boldsymbol{\xi})$ . (Right) MCMC sampling from posterior $\pi(\boldsymbol{\xi} \mathcal{D}_{obs})$ . . . . .	103
4-3	Schematic of a wing spar highlighting the region of showing a B-scan at a defect location. . . . .	107
4-4	Estimating alignment at a point by minimizing the integral of the gray scale over the trial fibre using the MFIA algorithm [164]. Randomly sampled points are used to reconstruct an alignment over the domain. .	109
4-5	A hierarchical multilevel sampling scheme that is biased towards regions of high misalignment is used draw sampling locations for the trial fibre. The number of new samples per cell on every level are proportional to the relative average misalignment observed in that cell on the previous level. The figure shows level 1 through 3 from left to right and each picture only shows the samples drawn on that level. . . . .	110

---



---

4-6	FE model showing the true geometry of the part with a sample wrinkle amplified for visual clarity. Note that it is a fully internal wrinkle with no trace at the surfaces. . . . .	114
4-7	The ACF showing the longest autocorrelation length across all dimensions of MCMC is illustrated here. Monte Carlo samples of wrinkles are obtained by subsampling every $\Lambda = 100$ samples. . . . .	115
4-8	(Left) Two-dimensional posterior distributions of the first five coefficients $a_i$ in Eq. (4.1), note 2-D plot axes are plotted on a scale of $\pm 0.25$ to visualize dependencies. For example, the $a_3$ plots suggest that a relatively constant amount of the $3^{rd}$ KL mode compared to others is present in all wrinkles studied here. (Right) Posterior distribution of covariance length scale parameter $\lambda$ plotted separately. . . . .	115
4-9	Top row shows B-Scan data, bottom two rows show 8 independent posterior samples of wrinkles in B-Scan coordinates. . . . .	116
4-10	(Left) CDF of critical or failure moment $M_c$ per unit width of a part, where wrinkle distributions are using the Bayesian framework introduced within this chapter. (Right) $M_c$ of samples obtained by assuming a Gaussian prior with mean and variance derived from data. . . . .	118
4-11	Left to right showing $\sigma_{33}$ , $\tau_{13}$ and $\tau_{23}$ respectively at $M_a = 1$ kNmm/mm.	118
4-12	Approximating the relation between $M_c$ and maximum wrinkle slope with Eq. (4.24). $q = 2.867$ and $\lambda_q = 4.212$ for the fitted curve. $q = 2.587$ and $\lambda_q = 3.834$ for lower 99% confidence bound. . . . .	119
5-1	A Gaussian random field showing a more continuous distribution of scales such that features are not readily distinguishable using discrete lengths scales. . . . .	123
5-2	Eigenvalue spectrum of (left) an isotropic material, (right) a composite and (bottom) a Gaussian random field. . . . .	124
5-3	One dimensional example to clarify definitions in Section 5.2. (Left) Non-overlapping partitions of $\Omega$ . (Right) Partitions of $\Omega$ overlapped by one element giving an overlap 2 elements wide. . . . .	127

---

---

5-4	(Left) The domain $\Omega$ divided into 64 non-overlapping subdomains $\Omega'$ . (Middle) The solid coloured region shows overlapping subdomain $\Omega_j$ where the red subregion is denoted by $\Omega^\circ$ in our notation and the white subregion is $\Omega_j \setminus \Omega_j^\circ$ . (Right) Close up of $\Omega_j$ where the set $\text{dof}(\Omega_j)$ is marked by square black grid points while the spherical green grid points on the boundaries denote the set $\overline{\text{dof}}(\Omega_j) \setminus \text{dof}(\Omega_j)$ . Note: we assume that all surfaces are Dirichlet boundaries. . . . .	127
5-5	Visual comparison of the two partition of unity operators in one dimension.	130
5-6	The red nodes show the interior boundary $\partial\Omega_{j,\text{int}}^\circ$ while the black nodes show the exterior boundary $\partial\Omega_{j,\text{ext}}^\circ$ of an arbitrary 2D subdomain. . . .	132
5-7	(Top) 2D composite elastic field showing the layered material with applied constraints. The domain is discretized into $4 \times 4$ subdomains and $\Omega_6$ is overlaid in a white mesh. Deformations for $\Omega_6$ are shown in Fig. 5-9. (Bottom) Gaussian random field for the same problem with a high contrast $\approx 100$ . . . . .	133
5-8	Sensitivity of <b>GenEO</b> multiscale method to 3 PoUs for the problem shown in Fig. 5-7(top). (a) Non-smooth PoU (b) Smooth PoU and (c) uPoU. In (d) we present the Von Mises stress error in $L_2$ norm relative to fine scale solution, assumed to be the truth. The stress error curves correspond to 50% overlap for all three PoUs and are identified by their markers. . . .	135
5-9	First 6 eigenmodes in 2D elasticity excluding the zero energy rigid body modes. The deformations are shown for a subdomain that does not share any nodes with the domain boundary i.e. floating subdomain marked by a white mesh overlaid on Fig. 5-7(top). Red cells represent softer interply regions of a laminate and blue cells show carbon fibre layers. . .	136
5-10	Von Mises stress plots for the layered composite beam shown in Fig. 5-7. (a) Fine scale solution with 80,802 dofs (b) Non-smooth PoU, (c) Smooth PoU and (d) uPoU. (b), (c) and (d) were computed with 480 dofs. The solution in (a) is treated as the truth relative to which the error $\varepsilon_\sigma$ is computed. . . . .	138
5-11	Stresses after patch recovery at 50% overlap for all three PoUs. . . . .	139
5-12	Von Mises stress error relative to the fine scale solution pictured in Fig. 5-10(top). Both plots shown here are solved using uPoU. (Top) Before stress recovery, $L_2$ norm error $\approx 2.6\%$ . (Bottom) Post stress recovery, $L_2$ norm error = $2.1\%$ . . . . .	139

---

---

5-13	(a) Fine scale solution for heterogeneous material shown in Fig. 5-7. (b) Multiscale (640 dofs) solution projected on the fine grid. (c) Difference between fine and multiscale solutions. . . . .	141
5-14	Visualizing uPoU on $\Omega_7$ of the heterogeneous beam with 50% overlap. .	142
5-15	Stress plots for 3D composite beam using the smooth PoU. . . . .	143
5-16	Wrinkle defined in Eq. (5.18) embedded in a laminate. . . . .	144
5-17	Stress plots for 3D composite beam containing a wrinkle solved using the smooth PoU. . . . .	145
5-18	Stress plots for 3D composite beam containing a wrinkle calculated with uPoU. . . . .	146
6-1	Representation of Bayesian approach in a simplified 2D parameter space for some likelihood function. (Left) Sampling from isotropic Gaussian prior $\pi_0(\xi)$ . (Right) MCMC sampling from posterior $\pi(\xi \mathcal{D}_{obs})$ . . . . .	150
6-2	The solid contours denote some (true) posterior distribution $\pi$ and the dashed contours mark a coarse approximation of it $\pi_H$ . Generally these are unknown quantities but we draw them here to demonstrate the delayed acceptance MCMC scheme. The stars denote two consecutive samples in the $\xi$ -chain (red arrow) pulled from $\pi$ while the black dots show the subchain or $\eta$ -chain (dashed black arrows) sampling from $\pi_H$ between $\xi^{i-1}$ and $\xi^i$ . . . . .	152
6-3	Basis functions used to create a Gaussian random field. All plots range between -1 and 1 on the domain $[0, 1]^2$ . . . . .	157
6-4	The reference elastic field created using the basis functions shown in Fig. 6-3. This is treated as the truth to be determined by solving the inverse problem. . . . .	158
6-5	Mean field for computed parameters . . . . .	159
6-6	Probability density of the error between fine scale and multiscale models $t_H$ . The variance in error $\mathbb{V}(t_H) = 2.3 \times 10^{-4}$ . . . . .	161
6-7	Top: Actual scaled error $t_H$ for a chain can be seen converging to the user provided threshold $\epsilon = 10^{-1}$ . Bottom: Error reduces as the multiscale model learns the posterior distribution. The error plotted here, $\hat{t}_H$ , is the mean of the scaled error $t_H$ of a chain between two consecutive updates. The plots are shown for 4 chains. . . . .	162

---

---

6-8	Black dots highlight the samples where the updated condition was triggered and the solution added to the coarse space with respect to the first two parameters $p_1$ and $p_2$ . The red dots mark all other samples drawn by the MCMC process. Note the burn-in where almost every sample requires an update. . . . .	163
6-9	The x-axis shows sample of a chain and the vertical lines mark the samples where coarse space updates occurred. As the chain propagates, the frequency of updates reduces visualized here by the vertical lines becoming sparse as the coarse space learns about the posterior distribution. The vertical black line marks the burn-in length of the chain. . . . .	164
6-10	Posterior densities for the first 9 parameters of the random field. . . . .	165
7-1	An arbitrary body showing a particle and the relevant features in a peridynamic framework. . . . .	173
7-2	Left: Two particle PMB model where $c$ is the micromodulus or stiffness of the spring. Right: Constitutive law for the spring connection. . . . .	173
7-3	Left: A peridynamic subdomain embedded within a finite element domain. Right: More detailed drawing of the peridynamic subdomain showing the handshake region in red and the unconstrained peridynamic particles in green. . . . .	174
7-4	Solution strategy for multiscale modelling of failure using peridynamics on the microscale and FEM on the macroscale. . . . .	176
7-5	Coupled multiscale example . . . . .	177



---

# LIST OF TABLES

3.1	Demonstration of performance of different preconditioners for <b>Example01b</b> for fixed problem size (30,000 DOFs) but increasing the number of subdomains: Number of pCG iterations (it), coarse space dimension ( $\dim(V_h)$ ), an estimate of the condition number $\kappa$ . . . . .	88
3.2	Details of the six setups and results used in the weak scaling test. In all of the tests, we used two layers of 20-node serendipity elements per fibrous layer and only one layer of elements in each of the interface layers. The number of elements per core was fixed at 2808. . . . .	93
3.3	Strong scaling test with Setup 5 in Table 3.2, demonstrating near optimal strong scaling up to at least 11,320 cores. Note: EpC is Elements per Core and TCT is Total Core Time i.e. $TCT = \frac{T_{total} * N_{cores}}{60 \times 60 \times 24}$ . . . . .	94
3.4	A strong scaling test using the SPE10 dataset. . . . .	97
4.1	Assumed mechanical properties for CFRP material (M21/IMA), where 1 is the fibre direction in-plane, 2 is perpendicular to the fibre direction in-plane and 3 is out-of-plane. $s_{33}$ is the tensile through-thickness strength and $s_{13}$ is the transverse shear strength. . . . .	114
5.1	Assumed mechanical properties for CFRP material (M21/IMA), where 1 is the fibre direction in-plane, 2 is perpendicular to the fibre direction in-plane. . . . .	133
6.1	Convergence properties of all three MCMC algorithms tested for the first parameter $p_1$ . . . . .	166

---

# NOMENCLATURE

## Finite elements

### Preliminaries

$\Omega$	Finite element domain
$\Gamma$	Boundary of $\Omega$ the type of which is denoted by subscript
$\mathcal{T}$	Uniform mesh on $\Omega$ with element size denoted by subscript
$\sigma_{ij}$	Cauchy stress tensor
$\mathbf{u}(\mathbf{x})$	Displacement vector
$\mathbf{f}(\mathbf{x})$	Body force per unit volume
$\epsilon_{ij}(\mathbf{u})$	Infinitesimal strain tensor
$C_{ijkl}$	Elasticity tensor
$h_i$	Dirichlet boundary conditions prescribed on $\Gamma_D$
$g_i$	Neumann boundary conditions defined on $\Gamma_N$
$\phi^{(i)}(\mathbf{x})$	Vector valued shape functions at $i$ -th node
$\Omega'_j$	Non-overlapping $j$ -th partition of $\Omega$
$\delta$	Overlap size
$\Omega_i^\delta$	$i$ -th partition overlapped by $\delta$
$\Omega_i^\circ$	Set of overlapped degrees of freedom on the $i$ -th subdomain
$\chi$	Partition of unity operator

### Matrices

Upper case bold notation

$\mathbf{A}$	Coefficient matrix, stiffness matrix
$\mathbf{A}_{\Omega_i^\circ}$	Coefficient matrix of overlapped dofs of $i$ -th subdomain
$\mathbf{D}$	Matrix containing only the diagonal elements of $\mathbf{A}$
$\Sigma$	Covariance matrix

---

$\mathbf{R}$	Matrix containing only the off-diagonal elements of $\mathbf{A}$
$\mathbf{M}^{-1}$	Preconditioner
$\mathbf{R}_i$	Restriction operator for $i$ -th subdomain
$\mathcal{S}$	Sparsity pattern of a matrix
$\mathbf{X}_j$	Partition of unity operator on $j$ -th subdomain

### Vectors

Lower case bold notation

$\tilde{\mathbf{u}}$	Solution vector $[\mathbf{u}_{h,1}, \dots, \mathbf{u}_{h,n}]$
$\mathbf{u}_{h,k}$	Solution vector at the $k$ -th node for mesh size $h$
$\mathbf{b}$	Load vector
$\mathbf{x}$	Spatial coordinates
$\boldsymbol{\theta}$	Hyperparameters
$\boldsymbol{\xi}$	Random vector drawn from some distribution
$\boldsymbol{\mu}$	Vector of mean values
$\mathbf{r}$	Residual
$\mathbf{v}$	Trial vector
$\boldsymbol{\lambda}$	Eigenvalues
$\mathbf{n}$	Outward normal vector

### Scalars

$h$	Mesh parameter
$n$	Number of degrees of freedom in a mesh
$E$	Young's modulus specified by subscript
$\nu$	Poisson's ratio in the plane denoted by subscript
$\varepsilon$	Experimental uncertainty
$\mathcal{P}$	Probability
$\beta$	Step size of a Markov chain
$N$	Number of subdomains
$m^*$	Number of distinct eigenvalues of $\mathbf{A}$
$m$	Number of eigenmodes chosen for coarse space
$\kappa$	Condition number of a matrix



---

## Spaces

$\mathbb{R}^n$	n-dimensional space of real numbers
$\mathcal{K}_k$	Krylov subspace
$\mathbb{M}$	Space of all matrices with sparsity pattern $\mathcal{S}$
$H^1(\Omega)$	Sobolev space (piecewise linear) over $\Omega$
$V$	Function space for each component of $\mathbf{u}$ , subset of $H^1(\Omega)$
$V_h$	$V$ restricted to mesh $\mathcal{T}_h$
$V_{h,0}$	Subset of $V_h$ with complete support on $\Omega_j$
$\mathbb{B}$	$\mathbb{R}^3$
$V_H$	Function space for each component of $\tilde{u}_H$

## Subscripts

$h$	Fine mesh size
$H$	Coarse mesh size
$AS$	Additive Schwarz method
$RAS$	Restrictive additive Schwarz method
$JSM$	Jacobi Schwarz method

## Superscripts

$k$	$k$ -th index in a vector
$(k)$	$k$ -th instance of a vector
$e$	Pertaining to experiments
$m$	Pertaining to model
$T$	Transpose

Subscript  $i$  refers to the coordinate, dimensions or angles for the  $i^{th}$  layer of a multilayered stack.

## Miscellaneous

$\mathcal{F}$	Model or experimental output depending on superscript
$\varepsilon$	Error of type denoted by superscript
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

---

## Wrinkles

$n$	Number of samples with wrinkles observed
$\xi$	Random vector of coefficients for parameterizing wrinkles
$W(\mathbf{x}, \xi)$	Deformation field introduced by wrinkles
$\psi(\mathbf{x})$	Orthonormal basis defining wrinkles
$N_w$	Number of basis vectors parameterizing wrinkles
$\phi_j(\mathbf{x}_j, \xi)$	Misalignment at $\mathbf{x}_j$
$\mathbb{X}$	Subset of $\mathbb{R}^{N_w}$
$\mathbb{R}^{N_w}$	$N_w$ dimensional space of real numbers from where $\xi$ is drawn
$N_\phi$	Number of misalignment measurements per sample
$\mathcal{D}_{obs}$	Set of observed data $N_\phi \times n$ in size
$\mathbf{d}_{obs}^{(i)}$	$N_\phi$ long vector of misalignment observations for $i$ -th sample
$\mathbb{R}^{N_\phi}$	$N_\phi$ dimensional space of real numbers
$\mathbb{D}$	subset of $\mathbb{R}^{N_\phi}$
$\mathcal{J}(\theta)$	Variance of gray scale along the pixel array orientated at $\theta$
$\bar{\mathcal{G}}(\theta)$	Mean gray scale over pixel array at $\theta$
$\mathcal{G}(h, \theta)$	Gray scale at pixel array coordinate $h$ at $\theta$
$H$	Length of pixel array
$(h, \theta)$	1 dimensional local coordinate of pixel array at $\theta$
$m_j$	Number of cells at level $j$
$\Omega_j$	$j$ -th cell in $\Omega$
$\bar{\phi}_i^{(j)}$	Mean absolute misalignment in $i$ -th cell at level $j$
$\mathcal{X}_j^{(j)}$	Set of points sampled in $\Omega_j$
$\gamma_i^{(j)}$	% contribution of a cell to total absolute misalignment
$g_i(x_i)$	Envelope functions in the $i$ -th direction
$C(\cdot, \cdot)$	Covariance operator
$\mathcal{F}(\sigma)$	Failure criterion
$M_c^*$	Critical moment of failure for pristine sample
$M_d$	Failure moments for observed wrinkles
$M_c$	Failure moment for samples with wrinkles
$M_w$	Weibull modulus
$M_s$	Scale parameter for Weibull distribution

## Bayesian

$\pi_0(\cdot)$	Prior distribution
----------------	--------------------

---

---

$\pi(\cdot)$	Posterior distribution
$\mathcal{L}(\cdot \cdot)$	Likelihood of a proposal candidate
$\Delta$	Misfit function
$\xi'$	Proposal candidate
$\alpha(\cdot, \cdot)$	Acceptance probability
$N_{MC}$	Number of Monte Carlo samples
$\Lambda$	Autocorrelation time of a Markov chain
$\Xi$	Set of independent wrinkles
$\hat{Q}_M$	Monte Carlo estimate of $Q_M$

### Acronyms

AMG	Algebraic Multigrid
AR	Acceptance Ratio
AS	Additive Schwarz
(Bi-)CG(Stab)	(Bi-) Conjugate Gradient (Stabilised)
DOF(s)	Degree(s) Of Freedom
DUNE	Distributed and Unified Numerics Environment
FE(M)	Finite Element (Method)
FETI	Finite Element Tearing and Interconnecting
GenEO	GENeralized Eigenvalue in Overlaps
GMRES	Generalized Minimal RESidual
GMsFEM	Generalized Multiscale Finite Element Method
GPR	Gaussian Process Regression
IC(T)	Incomplete Cholesky (Thresholded)
ILU(T)	Incomplete Lower Upper (Thresholded)
JSM	Jacobi Schwarz Method
MALA	Metropolis Adjusted LAngevin
MCMC	Markov Chain Monte Carlo
MINRES	MINimal RESidual
MPI	Message Passing Interface
NDT	Non-Destructive Testing
PCN	Preconditioned Crank Nicholson
PCNL	Preconditioned Crank Nicholson Langevin
PDE(s)	Partial Differential Equation(s)
PDF	Probability Density Function
POD	Proper Orthogonal Decomposition

---

PoU	Partition of Unity
QMR	Quasi-Minimal Residual
RAS	Restrictive Additive Schwarz
RVE	Representative Volume Element
SPD	Symmetric, Positive, Definite
(S)SOR	(Symmetric) Successive Over-Relaxation
SYMMLQ	SYMMetric LQ
uPoU	Displacement based Partition of Unity
UQ	Uncertainty Quantification
XRCT	X-Ray Computed Tomography

---

# CHAPTER 1

---

## INTRODUCTION

Scientific advances in aerospace composite design and materials offer exciting engineering opportunities, making them the material of choice for many modern aircraft (e.g Airbus A350, Boeing 787). However, composite manufacturers face huge challenges in designing and making complex components quickly enough to remain commercially competitive. There is a growing realization in both academia and industry that to meet ambitious global growth targets, composites manufacturers should ‘*reduce time, cost and risk to market through the use of validated simulation tools*’ [1]. Currently simulation capabilities allowing high-fidelity full-scale analysis of a composite structure are limited and not openly available. Furthermore, analyses have mostly been confined to a deterministic approach that does little to account for the uncertainty introduced by defects, for instance. But, why is this? What makes this analysis of large scale composite structures so challenging?

When we apply classical finite element (FE) analysis to a composite structure, the problem reduces to finding a vector of displacements  $\mathbf{u}^{(i)} \in \mathbb{R}^3$  at each of the  $n$  nodes within a FE mesh. This leads to the sparse system of FE equations [2]:

$$\mathbf{A}\tilde{\mathbf{u}} = \mathbf{b}, \quad \text{where} \quad \tilde{\mathbf{u}} = [\mathbf{u}_h^{(1)}, \dots, \mathbf{u}_h^{(n)}]^T, \quad (1.1)$$

$\mathbf{A}$  is the global stiffness matrix and  $\mathbf{b}$  is the load vector arising from the applied boundary conditions or loading. In solving the linear system Eq. (1.1) and quantifying uncertainty therein, we face three significant challenges that are defined below:

---

## Scale of calculations

Composite materials are manufactured from thin fibrous layers, less than 1mm thick, separated by even thinner resin interfaces of thickness less than 0.05mm, yet entire component parts are generally several metres long, see Fig. 2-4. To resolve stresses and accurately predict failure, several elements need to be placed through each layer [3]. Naturally this means that the number of nodes  $n$  is very large. As an example, we model a 1m section of a wing box given in Section 3.5.3, while resolving the resin interfaces, giving in total 200 million degrees of freedom. Solving linear systems of this size requires specialized, parallel solvers. Current industry standard tools, such as ABAQUS [4], are not able to deal with these problem sizes, largely due to limitations of the parallel solvers employed.

## Material anisotropy

Central to the benefits of composite structures is the inclusion of directional fibres, which gives them an excellent weight to stiffness ratio under a particular loading. This means, there is a large contrast ( $\sim 1 : 40$ ) in mechanical properties within a single layer of composite, related to the fibre direction(s) and those directions dominated by the stiffness of the matrix material (typically a toughened epoxy resin). In the FE discretization, this leads to a stronger coupling between degrees of freedom in the fibre direction, as opposed to those in the orthogonal directions.

The fibrous layers are stacked with different fibre orientations to form a laminate, adding an additional level of complexity. The fibre directions act as stiff constraints on the deformation, whilst the weak connections give rise to low-energy mechanics within the structure. Mathematically, this causes significant numerical challenges in solving Eq. (1.1) via iterative solvers, since the system is very ill-conditioned. For such cases, classical iterative solvers (required to address Challenge 1) converge very slowly [5].

## Variability in materials and models

Manufactured components are guaranteed to have defects that compromise their performance measured as a *knockdown* in strength. In the aerospace industry where great emphasis is placed on safety, large heuristically derived factors have to be applied and redundancies built in. The final products are thus over-engineered, preventing the industry from exploiting the full benefits offered by composites. The primary driver for this is the uncertainty in part performance which is difficult to model. Quantifying the uncertainty becomes particularly challenging and even impossible in certain cases. For example, the resolution required to resolve small defects generates a very high

---

dimensional problem, a stumbling block for Bayesian methods [6]. For large, exotic and expensive components, limited availability of experimental data lowers confidence in Bayesian model outputs. Furthermore, the interest often lies in rare events which means many simulations can be expected prior to a single occurrence of a rare event. It is for these reasons uncertainty quantification is difficult in composite applications. Combining these with the first two challenges that arise in solving large, ill-conditioned systems, stochastic simulations using classical Bayesian inversion become intractable.

## 1.1 Outline of this thesis

### Chapter 2: Literature review

Chapter 2 highlights the demand for stochastic modelling of large, ill-conditioned systems arising in composite structures. From this point of view, we first review some existing methods of Bayesian inversion for model parameter estimation. The frontier of Bayesian applications is dominated by expensive models that motivate the development of effective solvers which at this scale necessarily require preconditioning. Therefore, we review some popular solvers followed by preconditioners used by them. A good preconditioner can double as a good coarse approximation; an idea that has led to the development of modern multiscale methods.

This chapter aims to provide a general overview of the current state of affairs from the perspective of the three foundational pillars of this thesis - solvers and preconditioners for large systems, multiscale methods for composites and Bayesian inversion using state of the art methods.

### Chapter 3: `dune-composites` - an open source, high performance package for solving large-scale anisotropic elasticity problems

In Chapter 3 we address two of the three motivational challenges in this thesis. As part of a larger international collaborative effort, we provide an open source package called `dune-composites` for modelling large composite structures. This chapter demonstrates some key features of the package, most important of which is the implementation of a novel, robust preconditioner that performs exceptionally well in parallel, boasting near perfect scalability. It is capable of handling a number of typical pitfalls for composite solvers such as very large problems, geometrically varying stacking sequences and defect inclusions, to name a few.

Furthermore, it interfaces well with the latest solvers including those in `dune-istl` [7] and `Hypre` [8] and provides room for users to extend it in any direction or use it

---

directly for engineering applications. This chapter documents the structure of the code and highlights the novelty aspects of the code base.

We illustrate its use through smaller examples that can be run on a desktop computer building up to a large scale simulation on the UK national supercomputer **ARCHER** using 15,360 of its cores with the aim of providing the user with sufficient understanding of how to implement the code for their own applications.

The work presented in this chapter is under review by the Journal of Computer Physics Communications.

R. BUTLER, T. DODWELL, A. REINARZ, **A. SANDHU**, R. SCHEICHL AND L. SEELINGER, **dune-composites** – *High-performance **dune** modules for solving large-scale, strongly anisotropic elliptic problems with applications to aerospace composites*, Computer Physics Communications 249 (2020).

#### **Chapter 4: A Bayesian framework for assessing the strength distribution of composite structures with random defects**

Large deformations during consolidation make composites prone to wrinkle defects [9, 10] which creates difficulties in quantifying part strength for two primary reasons; measuring these internal defects and their effect on strength. Chapter 4 presents a novel characterization method for wrinkles which is coupled with standard Markov chain Monte Carlo (MCMC) algorithm to find a possibly true distribution of wrinkles based on observed defects. Independent samples from this distribution are included in finite element simulations to predict residual strength or knockdown. These simulations are made possible by **dune-composites** package in Chapter 3.

An engineering result is derived from stochastic simulation that relates failure load to maximum gradient of the wrinkle for an industrially motivated case study of wrinkles in corner bends. The novelty lies in the method rather than this particular application since choices like failure criteria for composites are not universal. These choices are left to the user and can be trivially substituted depending on requirements.

The work presented in this chapter was published in the Journal of Composite Structures.

**A. SANDHU**, A. REINARZ, AND T. DODWELL, *A Bayesian framework for assessing the strength distribution of composite structures with random defects*, Composite Structures 205 (2018), pp 58-68.



---

## Chapter 5: Multiscale methods for composites

This chapter draws heavily from Chapter 3 with the key observation being the adaptability of the **GenEO** coarse space as a good multiscale basis. In Chapter 3 this coarse space has been shown to provide an excellent preconditioner for two-level additive Schwarz methods but, in this chapter, we focus on its use as a multiscale method. We describe a type of generalized multiscale method whereby a bespoke macroscale model is constructed from a handful of eigenmodes computed on overlapping subdomains by solving a generalized eigenvalue problem. The eigenmodes per subdomain are stitched together using a partition of unity (PoU) operator. Importantly, the choice of PoU has a major bearing on the quality of the multiscale model. So, we explore a range of PoU choices. We demonstrate the approach to constructing customized coarse spaces in a step by step manner with the help of a few toy examples in both, two and three dimensions.

The foundational work for this chapter appeared in the a special issue on the bifurcation and degradation of geomaterials with engineering applications.

T. DODWELL, A. SANDHU AND R. SCHEICHL, *Customized Coarse Models for Highly Heterogeneous Materials*. In: Bifurcation and Degradation of Geomaterials with Engineering Applications. IWBDG 2017. Springer Series in Geomechanics and Geoengineering.

## Chapter 6: Data-driven multiscale models for high dimensional MCMC

Chapter 6 links Chapter 3 and Chapter 5 in a delayed acceptance MCMC setting to address the third motivational challenge of this thesis - variability in materials and models. Here, we see the wrinkles introduced in Chapter 4 as localized variations in an elastic field that can be parameterized in some way. The parameters form the unknowns to be inferred in a Bayesian sense. This chapter addresses the problem of running stochastic simulations when a single forward solve is too costly. The idea is to combine a good reduced order model with an MCMC framework that permits efficient exploration of the posterior parameter space. This is achieved with the **GenEO** coarse space working as a surrogate model that explores an approximate posterior. In a way, the surrogate model which is cheap to solve, acts a filter for samples drawn from the prior that screens bad samples at a much lower cost thus improving overall efficiency. The effective sample size is therefore, greatly increased for a fixed computational budget.

As the Markov chain explores the posterior, the coarse space is adaptively enriched every time its indicator of quality exceeds a prescribed threshold. This is how a reduced order model tailored to a particular posterior is built which is much more effective than

---

constructing a bespoke space over the prior. Importantly, the coarse adaptation is an online step and does not need a wasteful offline calculation.

This work is in preparation for the International Journal of Probabilistic Engineering Mechanics

T. DODWELL AND **A. SANDHU**, *Data-driven multiscale models for high dimensional MCMC*



---

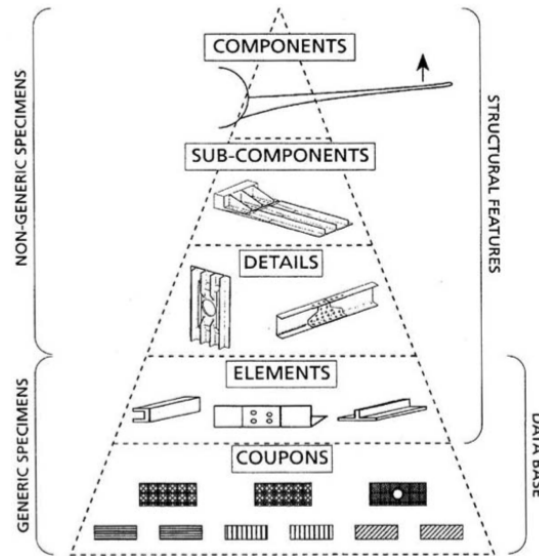
## CHAPTER 2

---

# LITERATURE REVIEW

Composite materials are widely used in several industries, an incomplete list of which includes aerospace, automotive, marine, sports and many more high precision applications [11]. Particularly in the aerospace industry, where safety is paramount, for a component to be certified airworthy and/or crashworthy, an extensive test pyramid outlines the testing protocols for newly developed components, see Fig. 2-1. The number of tests conducted decreases as development moves up through the levels of the pyramid with no more than one test being conducted on the aircraft scale. It is evident that this approach is expensive and cannot effectively quantify various sources of uncertainty. To overcome this limitation, large safety factors are imposed at every design stage producing over-engineered parts in the interest of safety. A modelling based initiative [12] allows numerical simulation and stochastic methods to be used in the certification process with the ultimate aim of lowering costs whilst challenging conservatism to obtain more optimized designs. Advances in mathematical modelling have virtualized parts of the test pyramid in the sense that mechanical testing supplemented with extensive models can reduce cost of development. The primary driving force behind these modelling protocols is the underlying industrial desire to reduce time, cost and risk to market through the use of validated simulation tools.

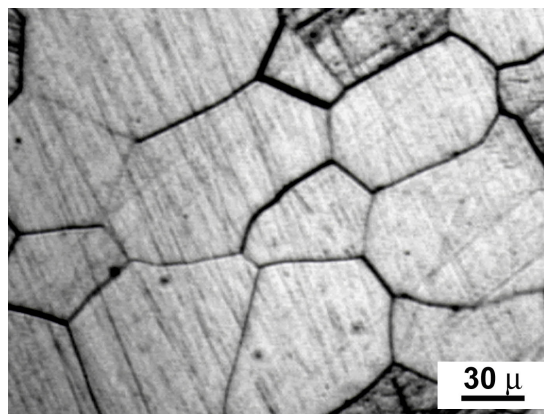
So why has this not been done yet? Particularly for composite materials, as we move up the pyramid, existing models lose accuracy and gain uncertainty since the effects of finer details such as defects are washed out because they cannot be resolved at the component level with existing computation capabilities. However, at the coupon level, most relevant heterogeneities can be captured. The multiscale paradigm capitalizes on



**Figure 2-1:** *The traditional approach to new aircraft design where each level of the test pyramid signifies the number of physical tests conducted which reduces as we move upwards with only one or two tests being conducted at the component scale [13].*

this phenomenon by enabling models at various levels to communicate with each other. The question then becomes, what information must be communicated across levels to achieve the objective of increasing accuracy and lowering uncertainty. In order to understand that, we will take a bottom-up approach starting at the coupon level.

All engineering materials, considered on a scale sufficiently small, reveal some degree of heterogeneity. In metals, the heterogeneity is evident at granular level where each grain is one continuous crystal lattice that does not tessellate well with its neighbours, see Fig. 2-2. Grains typically occur on the micrometre scale so an individual grain



**Figure 2-2:** *Micrograph of polycrystalline metal showing individual grains composed of a crystal lattice each [14].*

---

has little to do with the mechanical behaviour at the macroscale (say, metre scale) at which a metal is considered homogeneous. Somewhere between the macroscale and the granular scale, the influence of a single grain disappears. Beyond that length scale, the material is effectively homogeneous. Consider a small sample of a metal at this scale. Rodney Hill defines this sample as a representative volume that

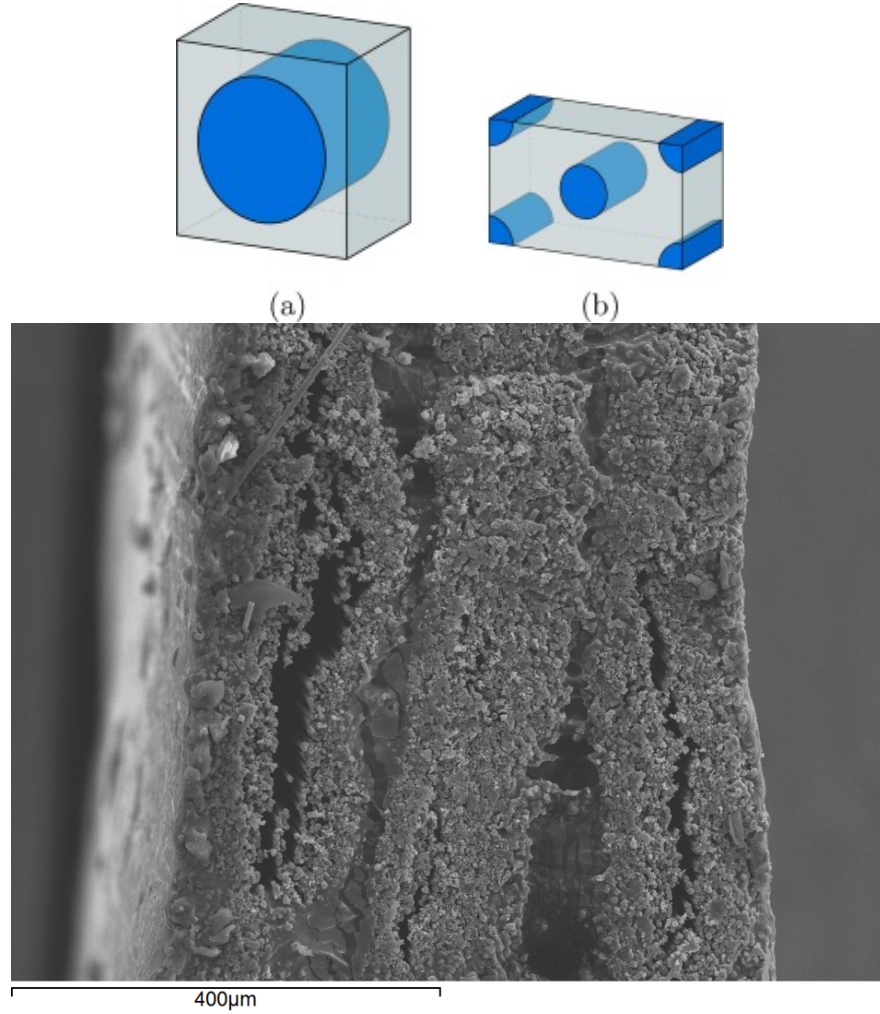
“(a) is structurally entirely typical of the whole mixture on average, and (b) contains a sufficient number of inclusions for the apparent overall moduli to be effectively independent of the surface values of traction and displacement, as long as these values are ‘macroscopically uniform’.” [15]

By macroscopically uniform, he meant the values fluctuate about a mean by  $\varepsilon$ , a value much smaller than the dimensions of the representative volume. If  $\varepsilon$  is not negligible, the representative sample is not large enough. On the other hand, as the size of a representative volume element (RVE) is increased, the boundary value problem that must be solved to obtain its stress state increases in complexity. A paper by Heinrich *et al.* [16] provides a detailed analysis of this balancing act.

The problem considered by Hill in [15] was the elastic behaviour of two solid phases bonded together under the assumption that the phases themselves are isotropic and uniform. He was then able to calculate bounds on the moduli for an arbitrary volume fraction as long as the ratio between the two isotropic moduli did not differ by a factor greater than two, approximately. The principle behind an RVE is to derive effective properties of the heterogeneous mixture contained within. This is known as homogenization, a detailed review of which can be found in [17] and particularly in the context of fibrous composites in [18–21]. As Hill showed in his work, calculating even simple Reuss [22] and Voigt [23] bounds on a representative volume applies a constraint on the contrast between constituent properties, although he was able to compute them for arbitrary mixtures. Prior to his work, homogenization was limited to very low volume fractions where, inclusions were mere impurities or dilute dispersions rather than stiffening inclusions [24]. However, the term arbitrary mixture is somewhat misleading here because an RVE can only be rigorously defined when one of the following statements is true; either the microstructure is periodic or the volume in consideration contains an infinite amount of microstructures so that it is statistically homogeneous and ergodic. The former statement is extremely limiting so most RVEs are based on latter. In a physical sense, the second statement suggests that the microstructure is invisible from the macroscale and can be considered homogeneous. This is formally known as scale separation which introduces the micro, meso and macroscales. Not all materials exhibit features at each of the scales. So, from a modelling perspective, more important than their existence is the separation between them.

---

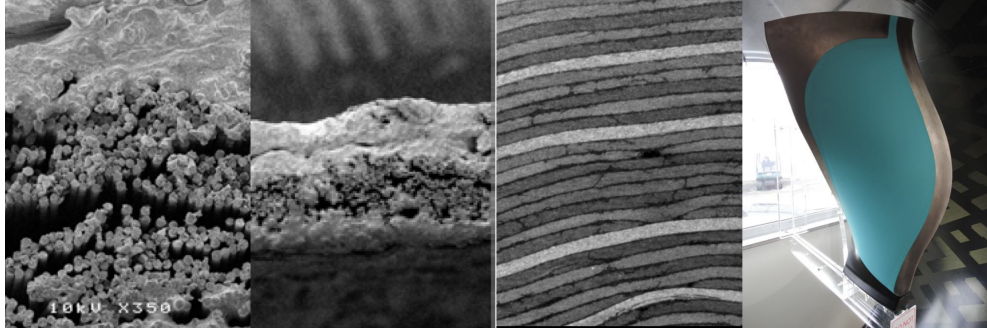
Consider an idealized microstructure of a unidirectional fibrous laminate. The fibres are assumed to be cylindrical and arranged parallel to each other. For such an arrangement, a unit cell or RVE is easily constructed that can be repeated to produce the laminate. However, as shown in Fig. 2-3, the cylindrical and parallel assumptions are an inaccurate representation of reality. Deviations from the idealized structure such as waviness, fibres crossing over and a variety of diameters, all of which occur in carbon fibre composites, have a measurable influence on their stress state particularly in terms of failure [25].



**Figure 2-3:** (a) and (b) show two types of unit cell that are representative of perfectly packed fibres (after [26]). (Below) shows an SEM picture of a unidirectional composite ply. Unit cells (a) and (b) are poor representations of reality.

All non-fictitious materials are of the non-periodic type because any manufacturing process will produce random deviations from the idealized microstructure. For an RVE

---



**Figure 2-4:** *Scale separation distinguishing micro from macro scale. The range of scales introduces significant complexity in the analysis of aerospace composites. From Left to Right: fibre/resin scale ( $5\mu\text{m}$ ), ply scale ( $0.25\text{mm}$ ), laminate scale ( $5\text{-}30\text{mm}$ ) to the structure ( $>1\text{m}$ )(Composite Fan Blade).*

to perform effectively, a material must exhibit sufficient scale separation. For example, in metals the heterogeneities are far smaller than the typical scale of engineering structures which permits structures to be represented by periodically arranged RVEs. Owing to this RVEs have been used with great success in modelling metals or even metal matrix composites [27, 28]. This is not the case for fibrous composites. In Fig. 2-4, we can identify individual structures at multiple scales without much separation between them relative to an equivalent metallic structure.

For common applications, composites are generally modelled within a finite element framework with effective or homogenized properties instead of taking into account individual material properties and geometric arrangement at the microscale. These effective properties can be expensive or even impossible to measure post manufacture due to substantial geometric changes. For these reasons, a lot of effort has gone into the homogenization methods to derive composite properties directly from materials and microstructure. Beginning with the rule of mixtures many more analytical methods for homogenization such as concentric cylinder assemblage [18], Hashin-Shtrikman bounds [29] and the Mori-Tanaka [17] method were developed. These were complemented by numerical approaches like the generalized method of cells [20, 21] leading to the newer and more complicated homogenization by differential expansions [30] or by integral transformation [31]. For most engineering applications, these homogenized properties are sufficient on the macroscale however, these methods assume regular fibre placement in the matrix, which we can see from Fig. 2-3 is far from reality. Furthermore, deviations from the ideal structure occur not only at the microscale but also mesoscale for composites, and these methods are no longer valid. For instance, mesoscopic defects like wrinkles on the ply level introduce non-periodic localized variations in the effective properties that become difficult to homogenize in a general way. So, the FE mesh must



---

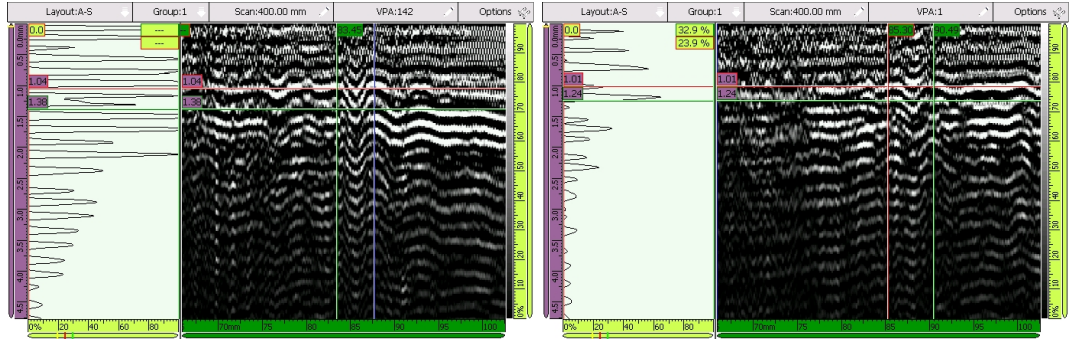
be refined till homogenization is applicable across an element. This implies an upper bound on the mesh parameter if we are to capture the influence of such defects. Consequently, the problem becomes intractably large. Thus, we are left with no choice but to employ a multiscale approach. In particular, we are especially interested in multiscale methods of the spectral kind which excludes RVEs since they provide a good way of modelling large structures with small non-periodic defects.

To that end we will now review wrinkles defects in composites and the difficulties in visualizing them in Section 2.1. Wrinkles are then cast as a localized variation in mechanical properties such that the elastic field of any component can be parameterized, the parameters of which are assumed to fit an unknown distribution. In order to sample from this distribution and quantify its uncertainty, we resort to Markov chain Monte Carlo methods, the current state of which is reviewed in Section 2.2.1. The systems of equations describing such a problem for composite structures may lie outside the scope of commercial iterative solvers. We discuss these limitations and their impact on the development of solvers in Section 2.3. Importantly, the literature makes it clear that the success of iterative solvers depends on preconditioning. In Section 2.4 we review the development of preconditioners from a historical perspective to the development of the latest domain decomposition preconditioners. Uncertainty quantification requires several forward solves however, even preconditioned iterative solvers are too expensive to permit many simulations. This constraint forced the development of multiscale methods which trade some accuracy for speed. An overview of existing multiscale methods is presented in Section 2.5 highlighting the need for customized coarse spaces.

## 2.1 Wrinkles and non destructive testing

While manufacturing large, complex composite components, small process-induced defects can form [32], for example porosity [33], in-plane fibre waviness [34], out-of-plane wrinkles [35, 36]. In practice, we observe a distribution of locations, sizes and shapes of these defects, and therefore the direct effect they have on part performance is uncertain. Within the aerospace industry, where safety is paramount, this uncertainty is mitigated by heuristic safety factors derived from extensive testing, which leads to high certification cost and over-conservativeness of design.

Wrinkle defects occur in the consolidation, forming and/or curing stages. There are number of different mechanisms that cause out-of-plane wrinkling [9, 10, 37], although in most cases they are caused by the combination of the mechanics of the laminate in its uncured state, and the geometric constraints imposed by the manufacturing tool to which the laminate must conform. Importantly, the presence of a wrinkle defect



**Figure 2-5:** *Ultrasonic B-scans showing out of plane fibre waviness or wrinkles.*

can significantly effect the structural integrity of the as-manufactured part, in some cases leading to expensive wholesale rejection. Naturally there has been a focused research effort to develop non-destructive methods (NDT) to measure and classify wrinkles in as-manufactured parts [38–42]. Amongst the variety of NDT methods available, only some are suitable for investigating geometric features at the meso-scale (sub-laminate scale). The most popular of these are X-ray computed tomography (XRCT) [38], ultrasonic techniques [39, 40] or infrared thermography [41]. XRCT can provide fibre scale resolution in a 3D volume but it is a much slower process compared to some ultrasonic methods which sacrifice accuracy for speed by limiting resolution. Faster NDT methods such as infrared thermography [42] enable scanning of larger areas but are limited to 2D information since they cannot penetrate much deeper than the surface of composite components. For these reasons, the industry prefers using ultrasonic techniques. Besides speed and accuracy, choice of NDT methods is driven by physical constraints. Aerospace components are usually too large to accommodate in a CT scanner. Ultrasonic methods require that parts be submerged in a coupling medium (typically water) which can again, be a limitation for large parts. Phased arrays [41] provide a viable alternative in such cases by enabling in-situ ultrasonic scanning. Principally, it is a combination of multiple individual ultrasonic probes programmed to work harmoniously, steering and focusing sound without source motion. The array of probes is embedded in flexible housing capable of bending along curved surfaces. The phased array functions like a medical ultrasound scanner with one key difference - different regions are explored by steering the ultrasonic beam instead of the device itself.

These wrinkle measurements have supported a growing research interest in the mechanical knockdown of wrinkle defects. The formation of wrinkles not only disrupts the even distribution of fibre and resin, but can significantly increase interply shear stresses triggering failure at significantly reduced loads [25, 36, 43, 44]. Numerical

---

studies have used wrinkle measurements, to perform parametric studies using finite elements to explore the deterministic effect of variations of wrinkle shape on structural integrity [36, 43, 45]. These studies include prediction of both, failure initiation [25, 44] and mix-mode propagation [36]. Notably, in a recent study, Xie *et al.* explores the compressive strength of flat plate coupons containing internal wrinkles [45]. Here, the authors define wrinkles by a cosine function enveloped within a 3D Gaussian exponential. They use six classifying parameters to characterize a wrinkle namely, amplitude, wavelength, maximum misalignment angle, wrinkle-center location, offset parameter for the cosine basis function and the extent of the Gaussian envelope. Based on a large number of simulations, the study recommends that maximum wrinkle angle is the strongest indicator of strength knock down. Previously, Wang *et al.* developed three methods of fabricating out-of-plane waviness which were used to quantify a wider class of wrinkles by introducing additional parameters [46]. However, those wrinkles are also constructed from cosine functions fitted to empirical misalignment measurements. Based on this study, Lemanski *et al.* illustrated numerically, a strength reduction of approximately 54% for peak misalignment of  $22^\circ$  [43]. Other studies have explored various combinations of these parameters, reporting compressive strength knockdown dependencies on other parameters such as amplitude [47] and amplitude-wavelength ratio [48].

In this review we identify and challenge two limitations in the existing numerical studies. Firstly, the parameterization of the wrinkle has mostly been limited to single sinusoidal functions engulfed by a Gaussian envelope. We are aware of just one study by Kratmann *et al.* [49] which introduced a Fourier basis. This basis has limitations since a large number of Fourier modes are required to capture localized wrinkle profiles. We are unaware of any study that explores the sensitivity of their results to the choice of wrinkle parameterization in the field of linear elasticity. However, in buckling analysis of composite sandwich panels, Wadee [50] demonstrates that localized imperfections defined by sech based functions are of greater concern than periodic imperfections. We seek a more general parameterization of wrinkles by exploiting the literature from Gaussian random fields [51] and informing the parameterization directly from measured data. Secondly, analytical studies have explored the effects of variations of out-of-plane wrinkles in a deterministic way. Current studies have not explored the stochastic effects of wrinkles, to derive a distribution in strength of components due to defects. The success of a stochastic simulation is dependent on the ability to define the probability distribution of possible wrinkles. We see this as a Bayesian question. What is the distribution of possible wrinkles given that we observe a set of wrinkles for which we have NDT measurements? To answer the question, we need to be able to sample from

---

an unknown distribution, typically done via Markov chain Monte Carlo techniques.

## 2.2 Uncertainty quantification

Uncertainty quantification problems are usually of two types; forward uncertainty propagation and inverse uncertainty quantification. Our application involves elements of both so we will review them in that order.

**Forward uncertainty propagation** In scenarios where we are uncertain about model inputs, we wish to quantify the uncertainty in model outputs propagated as a result of parametric variability. For example, an important result in Chapter 4 is the prediction of variability in part strength given a possible distribution of defects that may be present in it. The variations in defects are seen as uncertain inputs to the model and our aim is to quantify the uncertainty in terms of mean and variance in the resulting distribution of model output which is a failure load.

**Inverse uncertainty quantification** Given a mathematical model describing a physical process for which we have some experimental observations, inverse uncertainty quantification estimates two things; 1) model inadequacy or the discrepancies between experimental and model outputs called bias correction and 2) parameters in the sense of distributions. This is the more interesting of the two types of UQ as it requires considerable ingenuity due to its computationally intensive nature. As a result, analysis of this type has mostly been restricted to scalar valued problems in the fields of geophysics [52], electromagnetism [53], genetics [54] and atmospheric analysis [55] to name a few.

In order to understand some of the considerations of inverse UQ, let us imagine a forward model  $\mathcal{F}^m(\mathbf{x}, \boldsymbol{\theta})$  for which we have experimental measurements  $\mathcal{F}^e(\mathbf{x})$ . If we denote the difference between model and experimental outputs with  $\varepsilon^m(\mathbf{x})$  and  $\varepsilon^e$  denotes experimental uncertainty, then we can write a comprehensive model updating formula that encompasses all sources of uncertainty as

$$\mathcal{F}^e(\mathbf{x}) = \mathcal{F}^m(\mathbf{x}, \boldsymbol{\theta}) + \varepsilon^m(\mathbf{x}) + \varepsilon^e \quad (2.1)$$

The bias correction element of inverse UQ estimates  $\varepsilon^m(\mathbf{x})$  and the parameter estimation component searches for the distribution of  $\boldsymbol{\theta}$ .

The forward models under consideration are elliptic partial differential equations with random high contrast coefficients describing some random field and solved using a finite element approach. Assuming that distributions of these random coefficients

---

are directly accessible, independent and identically distributed (iid) samples are drawn and solved deterministically. This is the Monte Carlo method and is the default choice for PDEs with random inputs. However, in its simplest form it converges slowly. Many acceleration techniques have been proposed over the years such as Latin hypercube sampling [56], importance sampling [57], Gibb’s sampler [58] and the Markov chain Monte Carlo [59] method.

Other methods that do not rely on deterministic solves exist such as stochastic finite elements with a perturbation approach [60] or probabilistic finite elements with second moment analysis [61]. However, these methods are limited to random inputs and outputs of small magnitude, which cannot be guaranteed, particularly in nonlinear analysis where outputs can vary strongly for small perturbations to inputs. To get around the small magnitude limitation, stochastic Galerkin methods were developed as generalizations of polynomial chaos expansion [62] which was used extensively in solid mechanics [63]. Generally speaking, these methods offer better convergence when they can be applied.

A class of methods that combines the advantages of sampling and non sampling based methods, called stochastic collocation methods was first proposed in [64]. It is a sampling based method similar in many ways to the Monte Carlo method but rather than approximating the solution statistics (as in traditional sampling methods), stochastic collocation aims to approximate a solution response function in the probability space. To approximate the function, it uses deterministic solutions at collocation points as opposed to solutions for iid samples in the Monte Carlo method. Furthermore, the Monte Carlo method does not approximate a solution response function but only the solution statistics. So, in that sense, stochastic collocation may be described as a strong approximation method in comparison to the weak approximation methods (Monte Carlo) [65].

Another popular UQ tool is Gaussian process regression (GPR) that learns from the data and uses the similarities between observations to predict the value for an unseen point. It is non-parametric by contrast as it aims to find distributions over all possible models consistent with observations [66]. Naturally, optimization over all possible models requires an infinite amount of time, so we restrict the search to a subspace. To do that a constraint is imposed which qualitatively suggests that samples from the input space close to each other, will produce outputs that are near each other. Mathematically, this is encoded in the covariance matrix which is the central component of any GPR. This covariance matrix is what enables us to get conditional probabilities of model outputs  $\mathcal{F}^m(\mathbf{x}, \boldsymbol{\theta})$  given experimental observations. A major limitation of GPRs is they are tightly bound to Gaussian distributions. If the likelihoods are non-

---

Gaussian, posteriors and marginal likelihoods are not available in closed form and the process is not Gaussian anymore. Another limitation, and arguably a more pertinent one to this thesis, is the scalability of Gaussian process regression which suffers from the so called curse of dimensionality. In the context of elasticity, where the model outputs  $\mathcal{F}^m(\mathbf{x}, \boldsymbol{\theta})$  are displacement measurements for example, the regression problem can become very high dimensional, where irrespective of the choice of covariance kernel, GPR will likely fail [67]. Such cases can be frequently encountered in the real world, and commonly in this thesis, so we have to resort to approximate inference methods like MCMC.

As mentioned previously, we are interested in large scale composite problems with highly varying coefficients in the governing equation, producing very high dimensional problems. All UQ methods reviewed here lose their merits in a high dimensional context with the exception of the Monte Carlo method which is dimension independent [68]. For this reason, it is our method of choice with Markov chain sampling.

### 2.2.1 Markov chain Monte Carlo

While the aim of models is to reduce reliance on experimental tests, tests are used to reduce and quantify uncertainty in the models themselves. The typical probabilistic approach is Bayesian, more commonly known as Markov Chain Monte Carlo (MCMC). Here, a prior distribution of the model inputs is tuned to a set of observed test data to learn a posterior distribution that fits the data (probabilistically), thus reducing uncertainties in the models themselves. However, existing capabilities are notoriously computationally expensive, limited to small scale applications and simplified experimental data sets that may often be synthetically generated, as the underlying methods are designed for simplified toy mathematical benchmarks [69–71]. To that end, we now review the relevant literature on MCMC.

Say, the occurrence of a wrinkle in a part is event A and observing a wrinkle is event B. Then Bayes' theorem states the conditional probability of a wrinkle occurring, given that certain wrinkles have been observed is written as

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(B|A)\mathcal{P}(A)}{\mathcal{P}(B)} \quad (2.2)$$

where  $\mathcal{P}(A|B)$  is the conditional probability of A given B,  $\mathcal{P}(A)$  and  $\mathcal{P}(B)$  are independent likelihoods or marginal probabilities of each event. Interest usually lies in finding  $\mathcal{P}(A)$ , the posterior probability density known as the posterior distribution. When events A and B are described by models, the problem becomes one of finding posteriors for model parameters.

---

The posterior probability distribution for a set of model parameters contains all that the data has to tell us in the context of a prescribed model. It is the fundamental quantity for Bayesian parameter estimation. Assuming that samples from the posterior distribution are directly accessible, we can estimate a function (integral) over that distribution using the Monte Carlo method. However, models of reality are ever increasing in complexity and the posterior distributions become inaccessible. For example, an engineer modelling a compression test of composite plates wishes to know the underlying elastic field which is described by spatially varying elastic constants. Given material variability, the elastic constants at a point must fit some unseen distribution. How then do we obtain independent samples from it? Markov chain Monte Carlo enables this exploration which has today become a ubiquitous tool for Bayesian inference.

Bayesian statistics tools have been well developed in a broad range of application fields including groundwater hydrology [52, 72], image visualization [73] and ecology [71]. Here, we use these Bayesian tools to integrate high-fidelity finite element modelling capabilities of composites [25] with experimental observations from various tests. The main idea is that from available information a broad probability distribution (the *prior* in the Bayesian terminology) is assigned to the input parameters. If in addition we have experimental data related to real parts, it is possible to reduce the uncertainty and to get a better representation of parameters by conditioning the prior distribution on this data (leading to the *posterior*). However, direct sampling from a posterior distribution is not possible, therefore we generate samples using a Metropolis-Hastings type Markov chain Monte Carlo (MCMC) approach [74]. This approach consists of two main steps: (i) given the previous samples, a new sample is generated using a proposal distribution [6], such as a *random walk*; (ii) the likelihood of this new sample (i.e. how well the proposal matches observations) is compared to the likelihood of the previous sample. Based on this proposal and comparison steps, the proposed sample is either accepted and used for inference, or rejected and the previous sample is used again. The process leads to a Markov chain of possible input parameters, which have the probability distribution we seek, namely the distribution of parameters given experimental data.

To obtain useful MCMC results such as the expectation of a function over the posterior distribution, one must generate sufficient samples from the posterior. Each sample requires the evaluation of posterior probability at that point in the parameter space. Every posterior probability density evaluation requires a forward solve to estimate the likelihood of sampled parameters. In the context of elasticity, a forward solve refers to one solve of the finite element model Eq. (1.1). Thus, the statistical efficiency of MCMC depends on two factors

1. Rejection rate: With the exception of delayed rejection [75], all MCMC algorithms

---

reject bad samples along with any related information so the computation effort spent in deciding the quality of the sample is wasted and statistical efficiency deteriorates.

2. Cost of a single forward solve: Any MCMC algorithm must have a finite rejection rate. If the cost of obtaining a solution and making the accept/reject decision is too high, the MCMC efficiency drops as rejection rate increases.

First we shall review methods that aim to improve efficiency of MCMC algorithms from the viewpoint of sample rejection followed by computation cost of a forward solve. Every time a sample is rejected, computational effort equivalent to one forward solve is wasted. So, we must decide how to traverse the posterior efficiently in order to evaluate its probability density without unnecessary model solves. Most MCMC algorithms concern themselves with the way samples are drawn from the prior so as to minimize rejection whilst ensuring that the posterior is effectively sampled.

### Metropolis-Hastings proposals

The scientific literature is strewn with a variety of sampling techniques, each with its merits and demerits including speed and scope for parallelization. The goal has always been to find an intelligent strategy for exploring the parameter space with minimal effort. A major bottleneck in this process is the number of unused likelihood evaluations which typically involve solving an expensive model. It is in this sense we require sampling methods to be efficient. In this section we look at a few MCMC proposals within the Metropolis-Hastings framework, noting that the accept-reject step is identical independent of the way a proposal is made.

Also known as the standard random walk, the Metropolis-Hastings [74] sampling method is simple yet powerful. In fact, it is the most commonly used MCMC method that can draw samples from any probability distribution  $\mathcal{P}(\mathbf{x})$  assuming the value of the function of interest  $f(\mathbf{x})$  proportional to  $\mathcal{P}(\mathbf{x})$  can be computed. The central idea is that the  $k$ th sample depends only the  $(k-1)$ th sample producing a chain of dependent samples. A standard sample is of the form

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \beta \boldsymbol{\xi}^{(k)}, \quad \boldsymbol{\xi}^{(k)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (2.3)$$

where  $\mathbf{x}$  vector represents the spatial location of the  $k$ -th sample and  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  is a normal distribution with mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ . Let us demonstrate with the help of two examples, see Figs. 2-6 and 2-7. The first example shows an off-the-shelf



---

Metropolis-Hastings algorithm tasked with sampling from a target distribution

$$\mathcal{N}\left(\begin{bmatrix} 4 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}\right) \quad (2.4)$$

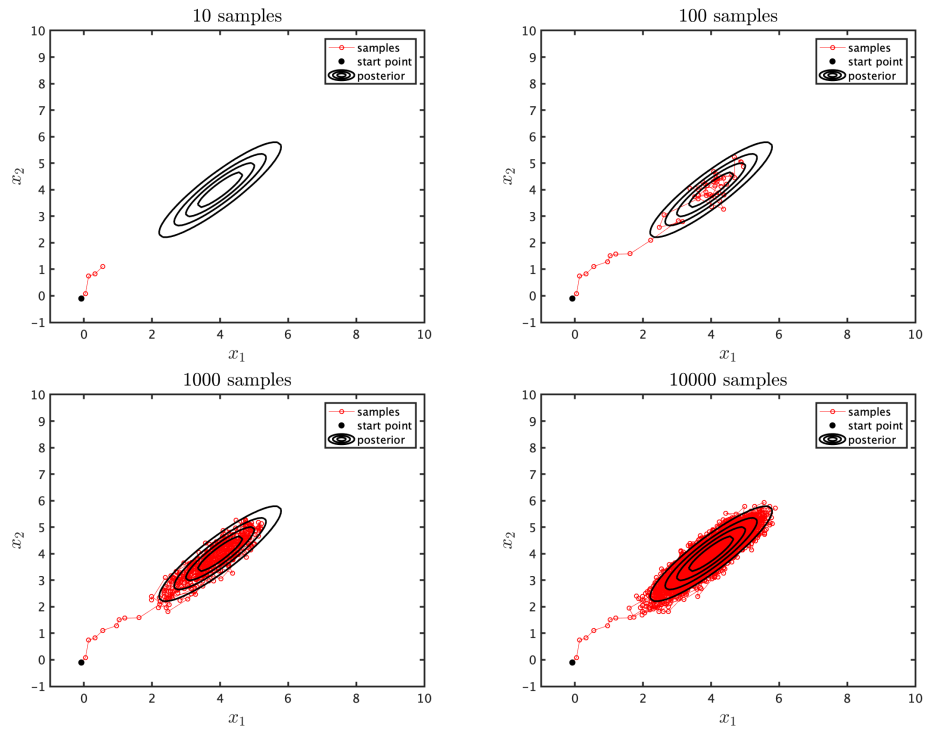
A chain is initiated from a random point (black dot in Fig. 2-6) drawn from

$$\mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}\right) \quad (2.5)$$

Metropolis-Hastings makes the samples converge to a stationary distribution. The time that passes before the samples start to represent the target distribution is called the burn-in period, after which the Markov chain is said to be mixed. A chain is said to be 'well mixed' when its length exceeds any observed correlation length scale many times over. Once the chain is sufficiently well diffused through the posterior, independent samples are pulled from the chain of correlated samples for further Monte Carlo analysis. However, this is a simple example where intuition and the 'eyeball norm' of an experienced eye provide sufficient diagnostic information regarding convergence. When posterior distributions become more complex, making decisions about length of burn-in or degree of mixing are not straightforward. Consequently, it is difficult to know when to terminate the MCMC process. It is common to ask, are we there yet? To demonstrate this difficulty, let us consider the second example that shows four Markov chains exploring a posterior given by the inverse of the Rosenbrock function

$$f(x_1, x_2) = (a - x_1)^2 + b(x_1 - x_2^2)^2 \quad (2.6)$$

where  $a = 50$  and  $b = 100$ . A step size  $\beta = 0.27$  is used when making a proposal according to Eq. (2.3). One of the simplest (and probably the earliest) diagnostic measures is the use of multiple chains. Multiple chains started with vastly different initial conditions, will give similar results if they converge. Only three decades ago, a convergence test used for scientific purposes was the so called *thick felt-tip pen test* [76]. To pass this test, the statistics for the set of independent samples from each chain should be separated (graphically) by no more than the width of thick felt-tip pen. If it failed, the chain continued running. Many theoreticians argue that such methods are fundamentally flawed because the thick felt-tip pen diagnostic measures the separation between two states of the sampled distribution rather than the how far either is from the true distribution. This statement is true for all existing diagnostic tools based on MCMC outputs. In fact, this argument is widely accepted yet, a suite of diagnostic tools plagued with this flaw continues to be used today because "a weak diagnostic is



**Figure 2-6:** Sequential plots of a sample Markov chain for a known posterior. (From left-right) 1. The first 10 samples of the chain showing clearly where some proposals have been rejected and previous samples carried forward; 2. 100 samples of the same chain. At this point we can say burn-in is complete; 3. 1000 samples drawn from the posterior show a well mixed chain. 4. 10000 samples showing all the correlated samples drawn. We note this example does not involve any model solves and it simply demonstrating the Markov chain process, in a conceptual way.

---

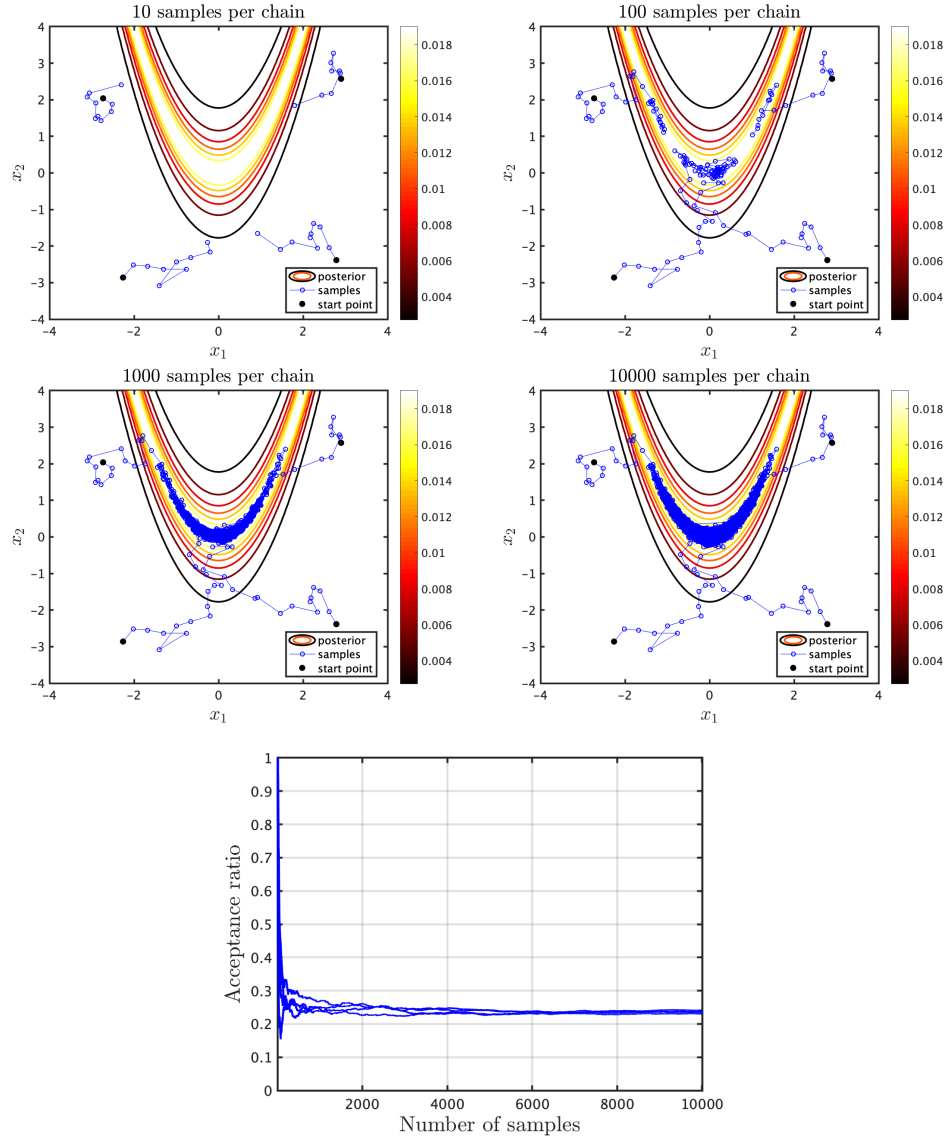
better than no diagnostic at all” [77]. Despite this logical setback, Gelfand and Smith kicked off a flurry of activity in the field of MCMC diagnostics. Within the five years that followed, a whole suite of tools was developed, of which the most popular today are the Gelman and Rubin measure [78] for multiple chains and the Raftery and Lewis [79] diagnostic for a single long chain.

The debate behind one long chain and several short chains has not yet been settled either. Advocates of the single long chain school of thought, Geyer [80] and Raftery and Lewis [81] argue that in the case of long burn-ins or strong autocorrelation within a chain, several short chains maybe essentially worthless. On the other hand, a single long chain may get trapped in a local optimum and fail to explore multimodal posteriors. With this in mind, we now look at the most popular diagnostic test for both single and multiple chains.

The Gelman and Rubin diagnostic [78] is two step process. First, a conservative estimate of the marginal posteriors is built, from which starting points are sampled for the desired number of chains. The number of chains depends on the modality of the target distribution yet no scientific basis relating number of chains required to the number of modes exists. Second, estimate the target density based on the last  $n$  proposals (where the length of chain  $\gg n$ ) as a generalized student t-distribution. The key concept is that the scale parameter for the t-distribution is based on inter-chain (means of chains) and intra-chain variances. For example, in Fig. 2-7 for 100 samples per chain, we can expect the variance of each chain to be lower than the variance of the means of chains. As chains converge, the ratio of these two quantities approaches 1. At that point, the chains can be terminated and the last  $n$  samples from all chains used for inference. Gelman and Rubin implemented this diagnostic tool in a software package called CODA [83].

The Raftery and Lewis diagnostic [79] supplements convergence detection with a way of bounding variances of any quantile for estimated functions (cdf, pdf, etc) of parameters. Based on the user’s quantity of interest or rather quantile of interest and desired accuracy, this test (also implemented in CODA [83]) recommends the length of chain, burn-in time and subsampling rate to extract uncorrelated samples. This test has however, received criticism for lack of robustness in predicting the length of chains given different initial conditions for the same problem. Furthermore, the appropriateness of the subsampling rate has also been challenged [84].

Both these diagnostic tests are applicable to all MCMC algorithms. More specialized tests also exist but they are also much more difficult to implement. A paper by Cowles and Carlin [77] reviewed 13 existing tests and reached the surprising conclusion that all of them can sometimes fail to identify convergence of the type they were



**Figure 2-7:** Sequential plots with 4 Markov chains exploring a more complicated yet known posterior given in Eq. (2.6). The acceptance ratio of all chains is near the theoretical optimum of 0.23 [82] for a step size of  $\beta = 0.27$ .

---

designed to detect even for contrived, toy problems. To their own dismay, they concluded that automated convergence diagnosis is unsafe, a state that may never change. Instead they recommend a number of alternative approaches to remedy this problem and can be found in [77].

Another important parameter is the acceptance ratio of a chain which, unsurprisingly is the ratio of number of new samples accepted to the length of the chain post burn-in. This is not so much a diagnostic tool as it is a way to tune MCMC parameters such as step size. A theoretical result shows that a ratio of 0.23 achieves optimum diffusion through a  $d$ -dimensional posterior [82]. However, this may not always be possible in practice. Acceptance ratio is strongly dependent on proposal selection. For instance, if the proposal is too narrow and the steps are small, the burn-in period is extended which may result in wasteful forward solves. On the other hand, if the proposal is too wide, the burn-in is indeed shortened but acceptance rate deteriorates. This makes sense because if the  $k$ th sample is far away from  $(k - 1)$ th sample, it is less likely to be as good.

Several specializations of the standard random walk exist that offer advantageous performance in certain situations. An incomplete list of examples is Gibb's sampler [58], Hit and run algorithm [85] and Metropolis-within-Gibbs' sampler [6].

It is well known, that off-the-shelf MCMC algorithms suffer from the curse of dimensionality [6] i.e. the mixing time tends to infinity as the dimensionality of the problem grows. The problem then is that under mesh refinement of PDEs (increasing number of unknowns), the MCMC convergence is expected to deteriorate significantly. Using standard algorithms, high dimensional inference using MCMC for any large model becomes prohibitive. Preconditioned Crank-Nicholson (PCN) proposal was developed to address this issue.

The preconditioned CN or PCN proposal is a modification of the standard random walk that differs only in the way a new sample is drawn. Most importantly, it lifts the curse of dimensionality i.e. MCMC convergence is achieved independent of mesh refinement. The proposal is of the form

$$\mathbf{x}^{(k)} = \sqrt{1 - \beta^2} \mathbf{x}^{(k-1)} + \beta \boldsymbol{\xi}^{(k)}, \quad \boldsymbol{\xi}^{(k)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (2.7)$$

which is an off-centre proposal with a scaled step size  $\beta$ . Cotter *et al.* use a Navier-Stokes example to demonstrate the robustness of this method to mesh refinement [6] and hence its superiority to standard random walk. There are other proposals of this type like Crank Nicholson Langevin (CNL) and pCNL proposals which are not reviewed here but can be found in [6].

---

The advantages of the random walk and PCN are two-fold.

1. In practice, proposal densities are chosen to be symmetric distributions which makes the acceptance probability a simple calculation requiring only the PDF be evaluated for the previous and current sample and
2. It is not necessary to know the normalizing constant which is usually an inaccessible quantity.

Both these random walk algorithms are similar in every way except proposal distribution which means both of them have an acceptance ratio (AR) between 0 and 1. Two competing requirements determine a balanced AR. As mentioned previously, there exists an optimum value of  $AR = 0.23$  that ensures mixing through the posterior. However, AR less than one means samples are rejected and computational effort is wasted. To improve AR without extending burn-in or compromising mixing properties, we can include some information about the posterior into the proposal. This brings us to a new type of proposal distribution that incorporates information about the likelihood.

Both standard walk and PCN proposals incorporate no information about the posterior density [6, 86, 87]. The idea behind Metropolis adjusted Langevin (MALA) proposals is that gradient information about the target probability density function is used in drawing a fresh proposal every time however, the accept-reject step remains identical to Metropolis-Hastings. Generally speaking, computing numerical derivatives where large models are involved is an expensive and often infeasible idea although in applications where derivatives are readily available, MALA proposals offer significant speed ups.

Instead of computing derivatives of densities for very large systems, an alternative approach may be to speed up the forward solves so that more independent samples may be drawn for a fixed budget. Various MCMC strategies have been developed to improve effective sample size for a given computational budget.

### **Delayed acceptance**

In cases where solving a forward model is a computationally intensive task, it may be considered unwise to evaluate the model to a high degree of accuracy for all samples, especially the rejected ones. It is beneficial to compute the solution approximately to screen for bad samples and then refine the solution for good ones. This is the essence of delayed acceptance MCMC. The idea is to replace the forward model with a computationally efficient and faster model that lowers the complexity of the full model by solving its projection in a subspace. Only for samples that are accepted at the

---

first stage (coarse evaluation) are full model solves carried out with a second stage accept-reject step to find the possibly true posterior or an unbiased estimate thereof [88].

The efficiency of delayed acceptance relies on the quality of the substitute model which is built from representative samples drawn from the posterior. Naturally, the choice of these representative samples is important. T.Cui *et al.* [69] propose an innovative solution by combining model reduction with adaptive MCMC. Their method does two things well;

1. The adaptive MCMC [89] process draws from the posterior in search for samples that can be used to build or enrich the surrogate model as a chain mixes through the target distribution. In this way, the accuracy of the faster model is adaptively improved.
2. The surrogate model draws from a distribution that is an approximation of the full posterior which is used to improve the efficiency of MCMC sampling. In simple terms, the approximate posterior acts as a filter to achieve better sampling from the true posterior.

Classical approaches have generally computed reduced models offline which means they are not tailored to the target distribution [90]. As a result, it may contain modes that are never observed in the true posterior. The posterior is generally not known till the data becomes available, so classical reduced models have to maintain numerical accuracy over a much larger region which unnecessarily adds to the computational burden. Enriching a model adaptively, reduces the burden by concentrating in localized regions of the posterior so that it can be of lower dimension while retaining accuracy of the offline approach. In this thesis, we couple multiscale methodology with delayed acceptance MCMC to build robust surrogates to improve statistical efficiency.

Motivation for the next section (see Section 2.3) is derived from the second factor governing statistical efficiency of any MCMC process; the need for better solvers. Conclusions of a stochastic nature are prohibitively expensive if a single forward solve requires excessive computational resources therefore, we seek a solver (and preconditioner) that accelerates convergence enabling forward solves to be computed in reasonable time. To that end, we first review the existing methods that are considered state of the art and identify avenues of improvement.

---

## 2.3 Solvers

Let us reconsider the linear system of equations

$$\mathbf{A}\tilde{\mathbf{u}} = \mathbf{b} \tag{2.8}$$

where  $\mathbf{A}$  is an  $n \times n$  coefficient matrix,  $\mathbf{b}$  is a known vector of size  $n$  and  $\tilde{\mathbf{u}} \in \mathbb{R}^n$  is the unknown vector. Such systems occur naturally when discretizing parabolic or elliptic PDEs with finite elements. In the case of elasticity,  $\mathbf{A}$  is also known as the stiffness matrix and is usually symmetric, positive and definite (SPD). With these assumptions in mind, we shall now review the existing solvers and identify gaps to be filled.

The traditional taxonomy of solvers into direct or iterative categories does not describe well the present state of affairs. Boundaries between the two have become increasingly blurred. For example, iterative solvers use plenty of ideas from the field of sparse direct solvers, mainly in the form of preconditioners, significantly improving their reliability. Direct solvers are almost always based on some flavour of Gaussian elimination but iterative solvers encompass a vast variety of techniques like the pure iterative methods such as Jacobi [91], Gauss-Seidel [92] or SOR [93] to Krylov subspace [94] methods which in theory, converge to multilevel methods. Classifying all these techniques under one heading can provide a somewhat misleading picture of the field, without even considering preconditioners. This will become evident in the following sections.

It is well known that direct or even sparse direct solvers are limited in application since they limit the size of a problem that can be solved. Elasticity being a vector valued problem in 3 dimensions grows rapidly when the mesh parameter is reduced. Imagine a unit cube discretized using a single linear hexahedral finite element with 8 nodes and 24 degrees of freedom. Its stiffness matrix is a  $24 \times 24$  dense matrix. Discretize the cube into 50 elements in each dimension and we create a system of nearly  $4 \times 10^5$  degrees of freedom (dofs). A stiffness matrix of this size (or even its factors) cannot be stored in the memory of a typical desktop computer. Therefore, given our interest in composite applications with millions of degrees of freedom, we are left no choice but to use iterative solvers. So, we begin by reviewing the present state of research on iterative solvers, with two aspects in mind - their degree of parallelism and rate of convergence.



---

### 2.3.1 Iterative solvers

Historically, initial development of iterative methods was dominated by stationary iterative methods such as successive overrelaxation and its variants that rely on matrix splittings. Examples of classical stationary iterative methods are Jacobi, Gauss-Seidel, SOR and SSOR [95]. The key idea, in contrast to direct methods, is that the system of equations is resolved iteratively where an initial solution is assumed and progressively improved. Say, we have a trial vector  $\mathbf{v}$ , we can substitute it into Eq. (2.8) and arrive at a residual vector  $\mathbf{r} = \mathbf{A}\mathbf{v} - \mathbf{b}$ . The purpose of iterative methods is to systematically alter the trial vector in a way such that components of the residual eventually disappear.

Now we briefly review each of these methods. Jacobi iteration is probably the simplest iterative method. It requires that  $\mathbf{A}$  be diagonally dominant which is split into its diagonal matrix  $\mathbf{D}$  and remainder  $\mathbf{R}$  such that  $\mathbf{A} = \mathbf{D} + \mathbf{R}$ . The  $(k + 1)$ th iterate, denoted  $\mathbf{u}^{k+1}$  is obtained by computing

$$\mathbf{u}^{k+1} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{u}^k) \quad (2.9)$$

For composite or general heterogeneous applications, Jacobi method is unsuitable as the coefficient matrix is unlikely to be diagonally dominant. Furthermore, convergence is guaranteed iff the spectral radius of the iteration matrix  $(\mathbf{D}^{-1}\mathbf{R})$  is less than 1, which is not achievable without preconditioning in composite modelling. However, this algorithm is amenable to parallelism but convergence is extremely slow relative to state of the art methods.

The desire for improved convergence led to the Gauss-Seidel method which is similar to Jacobi iteration but uses lesser memory allowing larger problems to be solved. In lowering memory requirements however, the method sacrifices its parallelism - an unimportant concern at the time of it's development. Although convergence is achieved faster than Jacobi method, it is still not fast enough. Young and Frankel, in search for better convergence introduce an extrapolation parameter  $\omega$  into the Gauss-Seidel iteration. When  $\omega = 1$ , Gauss-Seidel is recovered otherwise the method is known as successive over relaxation (SOR). With the optimal choice of  $\omega$ , convergence rate can be improved by an order of magnitude in comparison to Gauss-Seidel method but knowing that parameter in advance may not be possible. So, convergence suffers. A symmetric version of SOR called SSOR exists but provides no advantage over SOR as a stand-alone iterative method however, it is useful as a preconditioner for Krylov subspace methods.

Generally speaking, stationary iterative methods lack sufficient generality and de-

---

pend heavily on convergence parameters which may not be known *a priori*. These methods were developed with the aim of improving convergence properties to unlock the ability to solve larger problems however, convergence can often stagnate for reasons such as high contrast coefficients in the governing PDE or small length scales over which material variation occurs. Both of these challenges are encountered in the modelling of layered composites. For many practical problems, these methods do not converge at all. This led to the next wave of development which produced Krylov subspace methods [96–98].

### Krylov subspace methods

A Krylov method as defined by Gutknecht [94] is an iterative method starting from some initial approximation and corresponding residual and generating for all, or at least most  $k$ , until it possibly finds the exact solution, iterates  $\mathbf{u}^k$  such that

$$\mathbf{u}^k - \mathbf{u}_0 = q_{k-1}(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) = \text{span}\{\mathbf{r}, \mathbf{A}\mathbf{r}, \mathbf{A}^2\mathbf{r}, \dots, \mathbf{A}^{k-1}\mathbf{r}\} \quad (2.10)$$

with a polynomial  $q_{k-1}$  of exact degree  $k-1$ . For some  $k$ ,  $\mathbf{u}^k$  may not exist or  $q_{k-1}$  may have lower degree, for example in Bi-CG. Krylov methods, at least in theory, converge in a finite number of steps in exact arithmetic. One of the most popular methods from this class is the conjugate gradient or CG for short.

Hestenes and Steifel [99], and Lanczos [100] independently proposed the conjugate gradient (CG) method for SPD matrices which showed unprecedented convergence rates with well defined bounds [97]. Disregarding numerical errors, CG converges in a maximum of  $m^*$  steps where  $m^*$  is the number of distinct eigenvalues of  $\mathbf{A}$ . Experimental observations revealed that ill-conditioned systems required slightly more than  $m^*$  iterations to converge, however, well-conditioned systems converged in  $m$  iterations where  $m \ll m^*$ . This made CG a very attractive method. As it gained more traction, the research community went down bifurcating paths. Whilst a subset of researchers worked on extending CG to the non-symmetric or indefinite cases, the remainder focused their efforts into improving condition number of the system. Extensions for the symmetric indefinite case led to the development of MINRES and SYMMLQ [101, 102]. The non-symmetric indefinite case resulted in some famously known methods like GMRES [103], QMR [104] and Bi-CGStab [105] to name a few. We do not review these methods in detail as they lie outside the scope of applications in this thesis.

We narrow our focus to SPD matrices that are ill-conditioned. Most composite structures are of this type. To make them amenable to strong convergence using CG, their condition number must be improved. The answer to that problem is precondi-

---

tioning. It is a strategy for converting a not so readily solvable system into one for which a fast solution method exists. Computational experience accumulated over the last couple of decades indicates that a good preconditioner holds the key to an effective iterative solver.

## 2.4 Preconditioning

“In ending this book with the subject of preconditioners, we find ourselves at the philosophical center of the scientific computing of the future.... Nothing will be more central to computational science in the next century than the art of transforming a problem that appears intractable into another whose solution can be approximated rapidly. For Krylov subspace matrix iterations, this is preconditioning.” [106]

Mathematically, a preconditioned linear system of equations is of the form

$$\mathbf{M}^{-1}\mathbf{A}\tilde{\mathbf{u}} = \mathbf{M}^{-1}\mathbf{b} \quad (2.11)$$

where  $\mathbf{M}^{-1} \approx \mathbf{A}^{-1}$  is a linear operator called a *preconditioner*. Generally speaking, preconditioning improves the spectral properties of the coefficient matrix making it more favourable to iterative solutions. A preconditioner must be able to do two things;

1. It must be cheap to construct and easily applicable. This condition ensures a reasonable expense per iteration. But, what is meant by a reasonable cost? An important consideration in the affordability of a preconditioner is its reusability. For an invariant coefficient matrix, with an evolving right hand side, it is worth investing in a powerful preconditioner that is used repeatedly so that the setup cost is eventually amortized. If it cannot be reused, in nonlinear problems for instance, it must necessarily be cheap.
2. The preconditioned system must be easy to solve. This condition means that the solution must converge rapidly and the cost of solving the preconditioned system should be much lower than solving the same unpreconditioned one.

These conditions contradict each other therefore, a balance between the two is usually sought. This paradox resulted in two general approaches to constructing preconditioners. One way is to build specialized algorithms that are near optimal for a narrow band of problems. They could perform very well but require in-depth knowledge of the problem being solved including the original PDEs, their discretization, boundary conditions and so forth. Very effective preconditioners can be built in this way but they

---

tend to be extremely sensitive to details of the problem that even subtle changes may penalize the efficacy of the preconditioner. This method is not always feasible or even desirable. For example, the information required to build a problem specific preconditioner may not be available or extremely difficult to obtain. Furthermore, problem specific preconditioners cannot be applied as a black box and the user must understand it fully. These reasons inspire an interest in purely algebraic methods that use only the information contained in the coefficient matrix and are universally applicable. This is the second approach. Preconditioners designed in this way are sub-optimal for any particular problem but perform better across of a wider class of problems.

The term preconditioning was originally employed by Turing [107] but the scope was limited to managing rounding errors in direct solvers. In the context of iterative methods, Evans [108] was the first to use the term but the concept however, predates that usage. Previously, Jacobi used planar rotations to make systems of linear equations diagonally dominant, a condition necessary for convergence of the Jacobi method reviewed in Section 2.3. Today preconditioning is mostly targeted at improving the condition number of a system. The first use of preconditioning in this context was reported by Cesari [109, 110]. Early ideas for preconditioning CG can be found in [96, 111] but major developments came later with the works of Meijirink and Van der Vorst [112] where they proposed incomplete Cholesky CG (ICCG) method. Although the concept of incomplete factorization was not new at this point, observing their suitability as a CG preconditioner is attributed to Meirijink and Van der Vorst. This was followed by a paper by Kershaw [113] that did a lot to popularize the method. Preconditioners of this type are variations on the treatments of LU factors of the coefficient matrix in order to approximate its inverse whilst maintaining sparsity structure. Sparsity is enforced either via a level-of-fill [114, 115] or a drop tolerance [116] approach, selection criteria for which is not rigorously defined. Consequently, unpredictably large amounts of memory are a required to store LU factors, a fundamental challenge of factorization based methods. As a class, ILU preconditioners are sub-optimal in the sense that as the mesh parameter  $h \rightarrow 0$ , the condition number of the preconditioned system  $\kappa \rightarrow \infty$ . This particular constraint makes ILU preconditioners unsuitable for composite application where the mesh parameter can be very small to accommodate material anisotropy or defect resolution. Since the works of Kershaw [113] preconditioning has occupied center stage and much work has gone into improvements and extensions including level-of-fill and drop tolerance based incomplete factorizations, generalization to blocked matrices, stabilized versions and most recently, efficient parallel implementations. However, parallelization of incomplete factorization can be quite challenging and scalability is a concern.

---

These two limitations of incomplete factorization have motivated developments of two distinct type of preconditioners. First, difficulty in parallelization is the primary motivator for seeking a class algebraic preconditioners known as sparse direct inverses [117, 118]. These preconditioners are based on solving a series of smaller (least-square type) problems instead of factorization which is why they can be dominated by setup times. Importantly, they can breakdown for structural applications. Although, efficient parallel implementations have become available recently due to advances in parallel computing. In spite of their parallel character, they are similar to ILU preconditioners in terms of performance which makes them unsuitable for our applications. Second, preconditioners from the multilevel paradigm are a consequence of the scalability concern. For large anisotropic elasticity problems however, we find ourselves in need of a robust, parallel and scalable preconditioner that does not entirely fit in either category. This class of preconditioners called domain decomposition methods are particularly suitable due to their inherent parallelism. To this end, we first review multilevel methods and their limitations with respect to composites thus motivating the domain decomposition paradigm, reviewed in greater detail to provide the reader with a strong basis for understanding the objectives of this thesis. Importantly, the domain decomposition framework provides a natural way to parallelize any of the preconditioners mentioned here by breaking a problem down into smaller domains which can be worked upon in parallel.

#### 2.4.1 Multilevel preconditioners

The earliest preconditioners were built on relaxations schemes due to their popularity at the time rather than suitability. Krylov subspace methods were seen as nothing more than acceleration techniques. Multigrid methods are then a different class of acceleration techniques. In their simplest form, they use one fine and one coarse discretization of the underlying problem. After iterating on the fine scale, the residual is projected onto the coarse grid. Using the residual as the right hand side in Eq. (2.8) and solving, we obtain a correction that is interpolated back onto the fine scale. Importantly, we do not have to stop at two discrete levels. This gives rise to the multilevel paradigm, the first such methods being published by Fedorenko [119] although it did not gain traction till [120] provided the necessary analysis to support the idea and nudge it into the limelight. Other early ideas came from works by Brandt [121] and Hackbusch [122]. Since then numerous methods have come up that traditionally fall into two classes; geometric multigrid and algebraic multigrid.

---

## Geometric multigrids

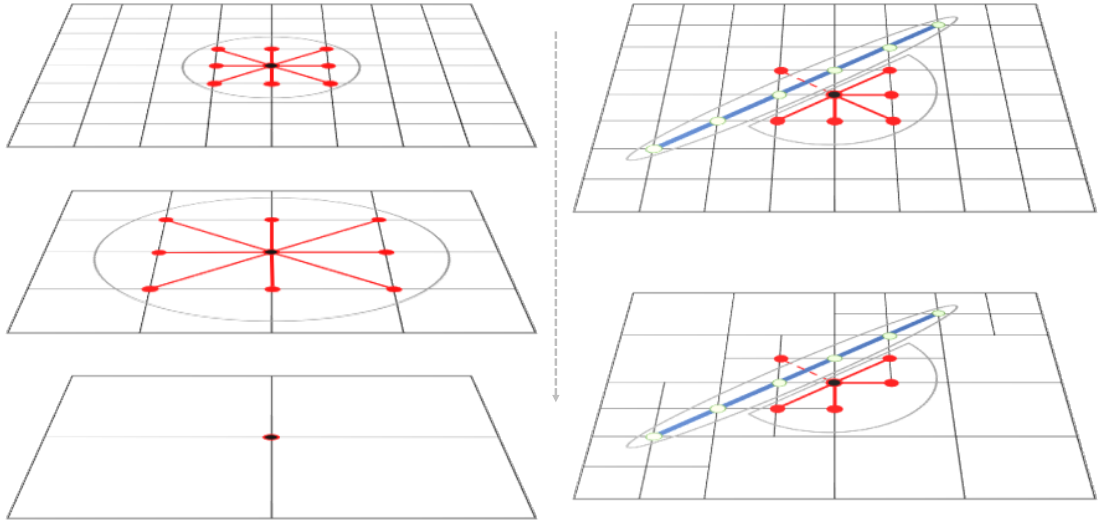
The original multigrid method, or more precisely the geometric multigrid was limited in application because most problems required a bespoke version. In principle, a finely discretized domain is coarsened till it can be solved quickly on the coarsest level. The solution is then interpolated back onto the finest grid. This way the slowly varying components of the error are removed. The rapidly varying error components are removed with a smoother which can be any iterative method described previously. It is easy to see how geometric multigrids can be sensitive to discretization (i.e. unstructured grids), geometries or material variations.

Consider a typical carbon fibre composite with distinct layers and interfaces where the layers are an order of magnitude thicker than resin rich interfaces between them. In a standard FE simulation of such a structure with high contrast materials requires a number of elements through the thin resin layers to accurately capture stresses. Therefore, every resin layer must have at least one element through thickness to provide the crudest stress estimate. Given a fixed domain, this constraint imposes a lower limit on the number of unknowns in the model which may still be too large a system to solve. Therefore, algebraic multilevel approach is preferred over the geometric one.

## Algebraic multigrids

The need for general multilevel algorithms led to one of the most famous methods, the algebraic multigrid (AMG). They are the most widely used preconditioners for iterative solvers for Eq. (2.8) in both commercial and scientific FE codes [7, 8]. They have demonstrated excellent scalability for a broad class of problems over thousands of processors, and have the advantage of working only with the information contained in the coefficient matrix, so that they can be applied ‘black-box’. As a preconditioner, AMG constructs the matrix  $\mathbf{M}$  by repeatedly coarsening the full matrix  $\mathbf{A}$  through recursive aggregation over the degrees of freedom. The aggregation process is algebraic and based on the fact that the solution at two neighboring nodes will be similar if they are ‘strongly connected’. The success of an AMG preconditioner depends on this aggregation process. Extrapolating from Fig. 2-8 (right) for multiple fibrous inclusions we can see that the connectivity of degrees of freedom within a composite laminate is very complex and even for a simple laminate, performance of all AMG preconditioners that we tested was prohibitively poor. In particular, this includes off-the-shelf AMG used in the commercial software ABAQUS [4], as well as tailored aggregation strategies in the AMG preconditioners provided through `dune-istl` [7] and `Hypre` [8].

This initial testing of AMG preconditioners highlighted the need for the develop-



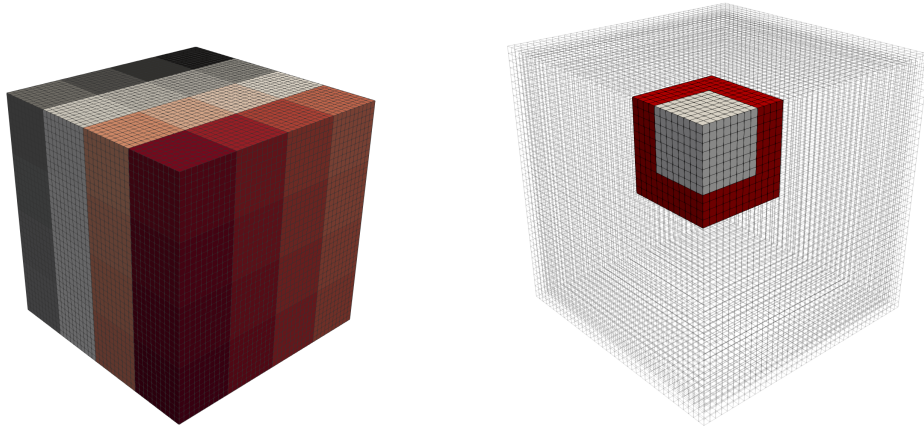
**Figure 2-8:** *Demonstration of the coarsening or aggregation process in AMG. Left: Isotropic material where nodes within a radius are strongly connected and have similar solutions so neighbouring nodes can be collapsed onto one node. Right: A stiffening inclusion (in blue) makes it difficult to design an effective aggregation scheme. The nodes in green are strongly and non-locally connected. The dashed red line shows a previously existing strong connection broken by the inclusion. If we see this inclusion as a single carbon fibre in a matrix, the challenge of aggregating degrees of freedom within even a single ply becomes obvious.*

ment and implementation of a robust preconditioner with respect to both problem size, material heterogeneity and anisotropy for large-scale composite based applications. Over the last decade there has been significant effort from the domain decomposition community to develop scalable and robust preconditioners suitable for parallel computation. One such preconditioning approach is provided by the additive Schwarz framework [123] that we classify as domain decomposition preconditioners.

#### 2.4.2 Domain decomposition preconditioners

Domain decompositions methods follow a divide and conquer philosophy and broadly fit into one of two categories; overlapping Schwarz methods and iterative substructuring methods. In this review, we focus on the former type which is more relevant to the subject at hand although both ideas serve a common goal of achieving good convergence independent on the number of subdomains.

The original Schwarz method relies on splitting the domain into smaller, overlapping subdomains using a graph partitioning tool such as METIS [124] or SCOTCH [125] so that each subdomain is handled by an individual MPI process. Good graph partitioners are aimed at balancing computational burden per processor by ensuring that subdomains are roughly equal in number of unknowns while minimizing communica-



**Figure 2-9:** (Left) Non overlapping partitions on a unit cube. (Right) A subdomain with overlaps shared by neighbours.

tion through intelligent domain decomposition. This basic parallel version, known as the Jacobi Schwarz method, has poor convergence properties and necessarily requires overlapped regions.

There are two key ingredients required by any overlapping domain decomposition method to have a chance at convergence;

1. A way to extract the subproblem and be able to project it back onto the complete domain. This is done via restriction and extension operators respectively.
2. A technique of stitching together local solutions which is achieved through partition of unity operators, shown as  $\chi$  in Fig. 2-10.

If the assembly of local solves to give a global solution is done without the partition of unity to average out the solution in the overlapped region in some sense, the resulting algorithm, known as additive Schwarz method (AS) converges poorly or not at all due to competing solutions in the overlaps. Therefore, a partition of unity must be applied giving the restricted additive Schwarz (RAS) algorithm.

Given a restriction operator  $\mathbf{R}_i$  that maps  $\mathbf{A}$  to  $\mathbf{A}_i$ , its transpose is the prolongation operator that reverses the restriction. Then the additive Schwarz preconditioner is defined as

$$\mathbf{M}_{AS}^{-1} := \sum_{i=1}^N \mathbf{R}_i^T (\mathbf{R}_i \mathbf{A} \mathbf{R}_i^T)^{-1} \mathbf{R}_i \quad (2.12)$$

Similarly, the RAS preconditioner can then be written as a sum of local inverses over all domains

$$\mathbf{M}_{RAS}^{-1} := \sum_{i=1}^N \mathbf{R}_i^T \chi_i (\mathbf{R}_i \mathbf{A} \mathbf{R}_i^T)^{-1} \mathbf{R}_i \quad (2.13)$$



---

where  $\chi_i$  is the partition of unity operator for the  $i$ -th subdomain.

Recall Eq. (2.8) for which the Jacobi iteration may be written as

$$\mathbf{D}\mathbf{u}^{k+1} = \mathbf{D}\mathbf{u}^k + (\mathbf{b} - \mathbf{A}\mathbf{u}^k) \quad (2.14)$$

In its original form, the Jacobi method is an element by element iteration however the block Jacobi method reorders the matrix into block form making it more suitable for iterative solvers since operations can be carried out on blocks, independently. Block Jacobi method therefore, has a higher degree of parallelism.

Say, the set of indices of a mesh  $\{1, 2, \dots, n\}$  are partitioned into two sets  $\mathcal{S}_1 = \{1, 2, \dots, n_s\}$  and  $\mathcal{S}_2 = \{n_s + 1, n_s + 2, \dots, n\}$  and corresponding partitions applied to  $\mathbf{u}$  and  $\mathbf{b}$  such that  $\mathbf{u}_1$  and  $\mathbf{u}_2$  denote the partitioned sets of unknowns and  $\mathbf{b}_1$  and  $\mathbf{b}_2$  the partitioned right hand side. The linear system in Eq. (2.8) has the following block form

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \quad (2.15)$$

where  $A_{ij} = A_{i \times j}$  for  $1 \leq i, j \leq 2$ . The block Jacobi algorithm now becomes

$$\begin{pmatrix} \mathbf{A}_{11} & 0 \\ 0 & \mathbf{A}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1^{k+1} \\ \mathbf{u}_2^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 - \mathbf{A}_{12}\mathbf{u}_2^k \\ \mathbf{b}_2 - \mathbf{A}_{21}\mathbf{u}_1^k \end{pmatrix} \quad (2.16)$$

or simply

$$\begin{pmatrix} \mathbf{A}_{11} & 0 \\ 0 & \mathbf{A}_{22} \end{pmatrix} \mathbf{u}^{k+1} = \mathbf{b} - \begin{pmatrix} 0 & \mathbf{A}_{12} \\ \mathbf{A}_{21} & 0 \end{pmatrix} \mathbf{u}^k \quad (2.17)$$

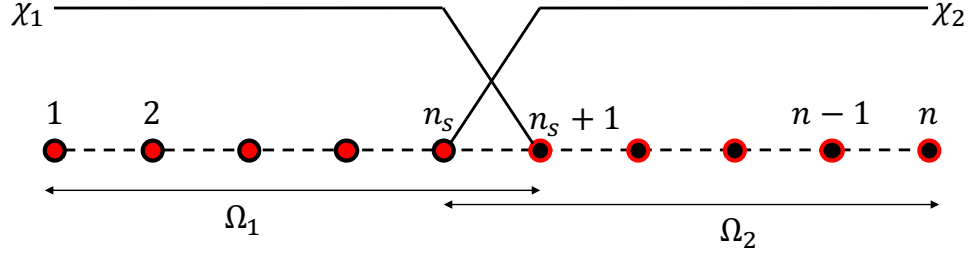
Notice, Eq. (2.17) is the blocked version of Eq. (2.9). We can rewrite this as

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \overbrace{\begin{pmatrix} \mathbf{A}_{11} & 0 \\ 0 & \mathbf{A}_{22} \end{pmatrix}^{-1}}^{\mathbf{M}_{JSM}^{-1}} \mathbf{r}^k \quad (2.18)$$

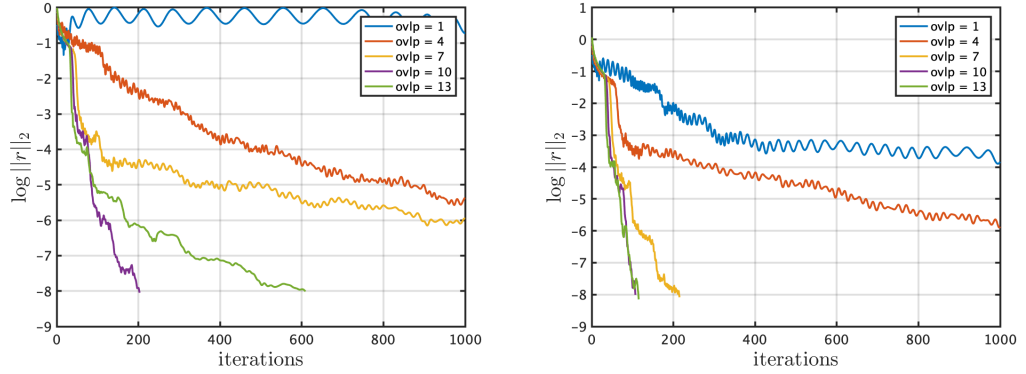
where  $\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{u}^k$  is the  $k$ -th residual vector. When the overlap between neighbours is minimized, the subdomain boundaries can be seen as Dirichlet boundaries in which case, we obtain the additive Schwarz algorithm as block Jacobi method i.e.  $\mathbf{M}_{JSM}^{-1} = \mathbf{M}_{AS}^{-1}$ .

Convergence of both AS and RAS as iterative methods is slow. However, they make good preconditioners for Krylov subspace methods. But how good are any of these domain decomposition methods? The iteration count of one-level methods plateaus out and weak scaling cannot be achieved. Stagnation is caused by few eigenmodes in the low

---



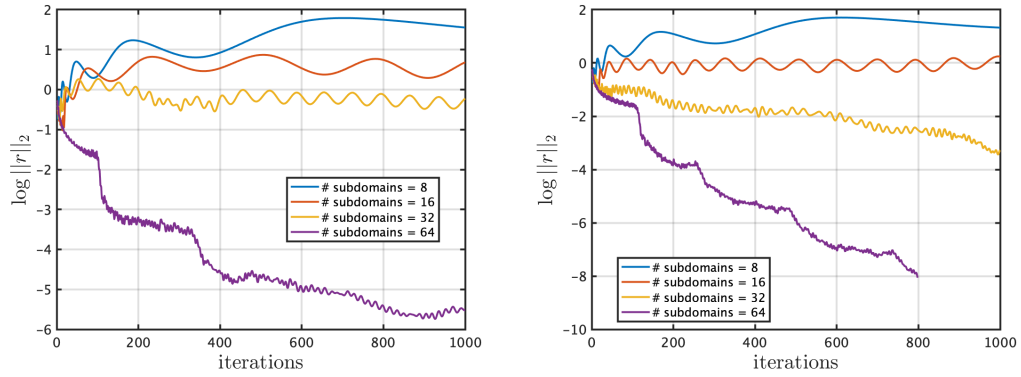
**Figure 2-10:** Two subdomain decomposition of a 1-D example showing partition of unity for the minimal overlap case. The Dirichlet subdomain boundaries have zero contribution to the solution and since there are no other shared nodes, the additive Schwarz reduces to block Jacobi algorithm.



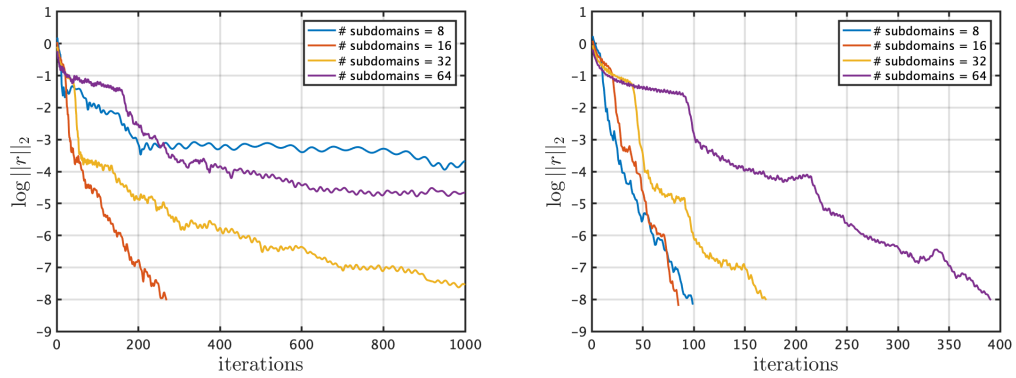
**Figure 2-11:** Left: Dependency of additive Schwarz on overlap for a fixed number of subdomains (32 in this test); Right: Dependency of restrictive additive Schwarz on overlap size. The only difference in comparison to the left plot is the inclusion of partition of unity in the construction of the preconditioner. The overlap is given in number of elements that a subdomain is extended by.

energy spectrum of the preconditioned problem i.e. the system is ill-conditioned even after preconditioning. This phenomenon occurs due to the lack of global information exchange. The solution on the  $i$ th subdomain depends on the value of  $\mathbf{b}$  across all subdomains which is not communicated in one-level methods. Therefore, a coarse grid correction must be applied which significantly improves convergence.

A classical coarse space correction was proposed by Nicolaides [126], which in a scalar valued problem is a vector of ones. In vector valued elasticity for example, the equivalent space would include all the zero energy modes of deformation. To demonstrate this with an example, we consider a contrived one dimensional elasticity problem for a composite material, shown in Fig. 2-14. A comparison of one and two level additive Schwarz methods for this problem divided into 32 subdomains is presented in Fig. 2-15. In the one-level AS case, the condition number of the system of linear equa-

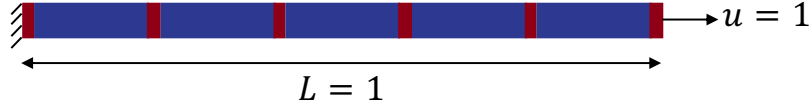


**Figure 2-12:** Left: In the additive Schwarz test, the overlap is fixed in number of elements while subdomain count increased. From the plot, at least 64 subdomains are required for convergence because the overlap, as a percentage of subdomain size, is insufficient for larger subdomains. Therefore, none of the solutions converge to the required tolerance; Right: The RAS test shows slight improvement in convergence with the addition of partition of unity.

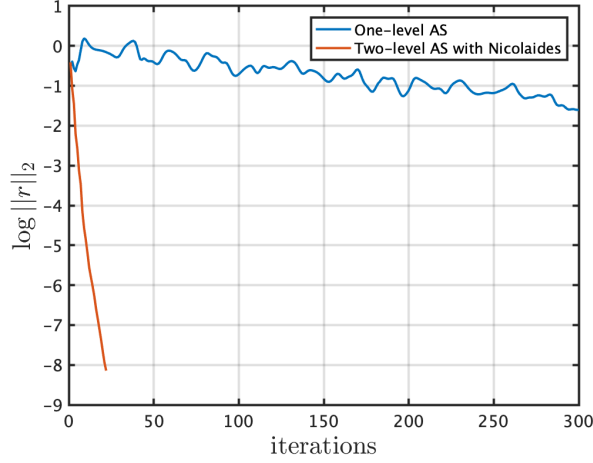


**Figure 2-13:** Additive Schwarz example with overlap fixed in terms of percentage of subdomain size which illustrates the paradoxical nature of two competing requirements i.e. overlap and number of subdomains. As the subdomain count increases, global communication becomes more necessary, so convergence deteriorates. Once again, AS on the left is compared with RAS on the right.

tions is  $\approx 5.12 \times 10^7$  which with preconditioning reduces to  $\approx 8.17 \times 10^2$  however, a converged solution is not obtained due to the Krylov solver stagnating. Nicolaides coarse space correction not only helps achieve convergence but it does so in just 22 iterations as shown by the red line in Fig. 2-15. The condition number improves from  $\approx 9.13 \times 10^8$  to  $\approx 2 \times 10^3$  with two-level AS preconditioning. In general, Nicolaides coarse space is a specific example of a wider class of spectral coarse spaces that are calculated by solving a local generalized eigenvalue problem. Two level additive Schwarz



**Figure 2-14:** One dimensional composite problem with resin rich (red) and carbon fibre (blue) dominated zones.  $E_{resin} = 10 \text{ GPa}$  and  $E_{composite} = 162 \text{ GPa}$ .



**Figure 2-15:** The two-norm of the residual against CG iterations for a 1D composite as shown in Fig. 2-14 using no coarse space (blue line) and Nicolaides coarse space (red line). One-level AS does not converge for this composite problem but with the Nicolaides coarse correction, the solution converges in 22 iterations.

preconditioners are of the form

$$\mathbf{M}_{AS,2}^{-1} = \mathbf{M}_{AS}^{-1} + \sum_{i=1}^N \mathbf{R}_H^T (\mathbf{R}_H \mathbf{A} \mathbf{R}_H^T)^{-1} \mathbf{R}_H \quad (2.19)$$

where  $\mathbf{R}_H$  is a coarse space computed from a coarse mesh with an element of size  $H$ , each representing a subdomain.  $\mathbf{M}_{AS,2}^{-1}$  is a two level additive Schwarz preconditioner that can be applied to the fine scale.

Two level preconditioners offer a major advantage that is exploited in this thesis. If chosen wisely,  $\mathbf{R}_H$  produces a good multiscale model eliminating the need to solve a large preconditioned fine scale model. Instead, one can directly solve the coarse model  $(\mathbf{R}_H \mathbf{A} \mathbf{R}_H^T) \tilde{\mathbf{u}} = \mathbf{R}_H \mathbf{b}$  and recover fine solutions in the region of interest by applying  $\mathbf{R}_H^T$  locally. Therefore, we review the choice of  $\mathbf{R}_H$  under the heading of multiscale methods in Section 2.5.

---

## 2.5 Modern multiscale methods

The need for fast forward solves has been highlighted in Section 2.2.1 where we need to solve a problem many times to quantify the uncertainty in an inferred quantity. In the case of expensive forward models, Section 2.2.1 talks about a surrogate model that is cheaper to solve and can be used as a screening step to increase effective sample size for a fixed computational budget. In this section, we concern ourselves with the development of that surrogate which encodes some amount of fine scale information into a coarser basis producing a multiscale basis. The major advantage of a multiscale basis embedded in the domain decomposition framework is that it offers a more efficient solution in a parallel setting.

One of the first multiscale methods was homogenization of heterogeneous materials, originally developed for elasticity problems. The term *homogenization* was coined by Ivo Babuska [127] to define methods based on averaging principles. It refers to an approach that explores macroscale behaviour of a medium by its microstructural properties. The idea is to replace a heterogeneous material with an equivalent homogeneous one. However, this substitution assumes a periodic arrangement of heterogeneities in the medium. Homogenization becomes difficult when the heterogeneities have a random character. Real world composites have geometric variations, changing stacking sequences and defects. This means the homogenized finite element has an upper bound in size which it cannot exceed otherwise the effective bulk properties will become invalid. This produces very large systems of equations. However, homogenized mechanical properties are still used to define the constitutive law over a finite element in a composite simulation. Each element in a composite lamina is defined by an effective (orthotropic) material matrix rather than individual fibre and matrix properties. Therefore, homogenization alone is insufficient as a multiscale method but is integral to the finite element approach. The homogenization paradigm is further extended or even exploited by the RVE concept, which we have already considered and concluded as being unsuitable for real world composite simulations. So we seek another FE based multiscale approach, particularly one suitable for parallelization.

Finite element tearing and interconnecting (FETI) methods are robust domain decomposition methods for solving large linear systems. Farhat and Roux [128] introduced them as iterative substructuring methods requiring reduced interprocessor communication in comparison to classical substructuring. Classical FETI and its variants are well established in the field of robust parallel solvers for large linear systems. They have been quite successful due to their broad scope of application and moderate complexity but most importantly, due to their excellent parallel scalability and robustness [129, 130].

---

Mandel and Tezaur [131] provided the first convergence proof for FETI methods applied to 2D problems with homogeneous coefficients by bounding the spectral condition number of the preconditioned system. Klawonn and Widlund [132] extended the implementation and analysis of the FETI method to 3D heterogeneous coefficients although the coefficients could only be piecewise constant with respect to the domain decomposition i.e. each subdomain in itself remains homogeneous but the overall structure does not. The authors proved the boundedness of its condition number is independent of discontinuous nature of the coefficients as long as they are aligned with subdomain interfaces and a special scaling is applied to the preconditioner. Pechstein and Scheichl [133] developed the analysis of FETI methods further to include highly heterogeneous multiscale problems where either the coefficient jumps are not aligned with grid partitions or local variations within a subdomain are of high contrast. However, numerical experiments have shown the lack of robustness in such cases. This is a major limitation of FETI methods in context of composite applications as the coefficient jumps are high contrast and not smooth. For a comprehensive analysis, we refer the reader to a monograph by Toselli and Widlund [134].

The latest multiscale methods in structural applications, involve some kind of spectral decomposition. It is a natural representation because eigenvectors of stiffness matrices represent various modes of deformation, ranked according to corresponding eigenvalues or energy requirement. The first two-level preconditioner based on Nicolaides coarse space mentioned in the previous section is also a spectral space where only the zero energy eigenmodes are used to formulate the coarse space or reduced order model on the second level. However, this space is insufficient as a multiscale model by itself, since this would produce deformations as combinations of rigid body modes only. So what is a good coarse space?

Some popular coarse spaces are derived by solving an eigenvalue problem. One such basis was proposed by Efendiev [90] involving a two stage computation called offline and online stages. In the offline stage, a series of local *snapshots* or solutions are computed which is usually large dimensional (dimension of the fine grid). This space of snapshots is subsequently reduced via some spectral procedure like proper orthogonal decomposition (POD). This new offline space forms the coarse basis for a given set of input parameters. Calibrating the coarse space by selecting a subset of the offline basis functions allows a balance to be sought between accuracy and efficiency. Crucially, a coarse space computed in this way to build a reduced order model is bound to given inputs, loads, boundary conditions and so on. Any changes to these would require the model to be recomputed, an expensive task particularly for large problems. Furthermore, the dimensionality of the solution of a reduced order model is affected by

---

how well it represents the multiscale character of the problem. Naturally, for a range of inputs, an evolving model is required. GMSFEM remedies this problem. The ingenuity of GMSFEM lies in systematic enrichment of the coarse space on the fly in the online stage thus tailoring reduced order models to capture the general mechanics over a wide range of possible inputs whilst ensuring it is adapted to the current problem.

In this method, the coarse space is obtained by solving a generalized eigenvalue problem of the form  $\mathbf{A}\mathbf{u} = \lambda\mathbf{B}\mathbf{u}$ . The choices for  $\mathbf{A}$  and  $\mathbf{B}$  are what distinguish different GMSFEM methods. In their paper, the authors use three different choices [90], in the third of which  $\mathbf{A}$  is the stiffness matrix and  $\mathbf{B}$  is the mass matrix of a subdomain.

Another choice for  $\mathbf{B}$  in overlapped grids is the overlap matrix extracted from  $\mathbf{A}$  such that the only non zero terms are those indexed by the overlapping degrees of freedom. In a physical sense, this Generalized Eigenvalue problem for Overlapping (**GenEO**) [135] subdomains computes eigenvectors that are compatible across subdomains due to the overlap constraint i.e. all modes computed over all subdomains can be stitched together (with a partition of unity) to produce a global multiscale basis. The **GenEO** coarse space is extensively used in this thesis both, for preconditioning large systems and as a reduced order model for stochastic simulations.

## 2.6 Concluding remarks

In this chapter, we have reviewed the literature on five distinct topics that are pulled together in this thesis for multiscale modelling of composite materials with random defects. Prior to the development of iterative solvers, modelling of large structures with tiny defects lay outside the scope of human capability but specialized iterative solvers called Krylov subspace solvers provided a computational route to solving extremely large problems. However, even these powerful solvers are prone to high iteration counts or convergence failure for many realistic problems that tend to be ill-conditioned. It then became clear that the success of iterative solvers depended strongly on preconditioning.

Domain decomposition preconditioners are best suited for large problems that can only be solved in parallel, particularly for composite materials where multilevel preconditioners are not sufficiently robust. However, a global preconditioner built from local inverses calculated on subdomains only, stagnates when the number of subdomains increases beyond a threshold i.e. does not scale well due to a lack of global communication. This is remedied by adding a second level to the preconditioner via the additive Schwarz framework. In this way, robust and scalable two level additive

---

Schwarz preconditioners can be assembled that are well suited to high dimensional, anisotropic elasticity problems.

Solving large systems of anisotropic equations is part of a larger endeavour - to quantify the uncertainty in manufactured parts that are defective to some degree in order to challenge industrial design conservatism and reduce cost. This is a Bayesian problem, the solution to which hinges on a number of successful solves of large systems. At this scale, even state of the art preconditioned iterative solvers become prohibitively expensive and the uncertainty remains unquantified. However, observing that a good coarse space (required for the second level of preconditioning) can be used as a robust multiscale method, opens a route to the solution. This permits model order reduction to a manageable dimensionality thus making repeated solves possible. This coarse model is a key ingredient of MCMC, the method used to quantify the uncertainty in the underlying random field.

There is a wide variety of MCMC methods, all competing to provide the greatest effective sample size in the most efficient way. Efficiency is governed by the number of wasteful solves because the time spent computing a bad sample is never recovered. Then the aim can be redefined as minimizing rejection and correlation between subsequent samples in search of the largest effective sample size for a fixed budget. This is most effectively achieved via the delayed acceptance framework where bad samples can be screened at a fraction of the cost by using a good multiscale model.

From the viewpoint of supplementing the test pyramid computationally with the goal of reducing uncertainty at higher levels, we have identified a need for efficient solvers that can tackle problems at the component level whilst maintains the accuracy of coupon level simulations i.e. robust and scalable solvers including preconditioning. This we address in Chapter 3. We also highlight a need for better parameterization of non-periodic features like wrinkles in composites and quantify the strength penalty they impose on components in a stochastic sense. This is studied in Chapter 4. This review also shows the lack of a good multiscale model that captures enough fine scale detail and simultaneously be cheap to solve. This is the key ingredient developed in Chapter 5 (as a byproduct from Chapter 3) which is then used in a Bayesian setting to make inferences about variations in the underlying elastic field.





---

---

## CHAPTER 3

---

# DUNE-COMPOSITES - AN OPEN SOURCE, HIGH PERFORMANCE PACKAGE FOR SOLVING LARGE-SCALE ANISOTROPIC ELASTICITY PROBLEMS

### 3.1 Introduction - Why do we need dune-composites?

In Chapter 1, we showed why there is a need to develop validated simulation tools for modelling of composites. It highlights three challenges that arise in modelling large composite structures and the limitations of existing capabilities. In this chapter, we tackle the first two challenges; scale of calculations and material anisotropy. In Section 2.3 we show how the challenge of scale restricts our choice of solver and demands a preconditioner. In Section 2.4 we show how material anisotropy produces ill conditioned systems making existing preconditioners less effective. Thus there is a need for a specialised preconditioned parallel solvers that are both robust and scalable.

The domain decomposition, or the one level additive Schwarz framework [123] whereby the domain is decomposed into a number of overlapping subdomains, which in our case corresponds to one processor, and the subdomain's local stiffness matrix is inverted on each processor directly. Reportedly, as shown in Section 2.4.2 this one level

---

approach is insufficient for very large problems and global information in the form of a coarse space must be added. Section 2.4.2 also shows the success of the Nicolaides coarse space [126]. In `dune-composites`, we use `GenEO` [135] to construct a coarse space by combining low energy eigenvectors of the local subdomain stiffness matrices using a partition of unity. The resulting preconditioner leads to an almost optimal scaling with respect to problem size and number of processors, allowing us to successfully tackle large industrially important problems with over 200 million degrees of freedom.

`dune-composites` is a high-performance composite FE package built on top of DUNE (Distributed and Unified Numerics Environment), an open source modular toolbox for solving partial differential equations (PDEs) with grid-based methods [136–138]. Based on the core DUNE philosophy, `dune-composites` is written using C++ and exploits modern inheritance and templating programming paradigms. It is open source and publicly available at <https://dune-project.org/modules/dune-composites/>. The package provides a single codebase with the following key features:

- implementation and interface to a novel, robust preconditioner called `GenEO` [3, 135] for parallel Krylov solvers, which exhibits excellent scalability over thousands of processors on Archer, the UK national HPC system. The preconditioner is now provided as part of `dune-pdelab` since release 2.6 at <https://dune-project.org/modules/dune-pdelab/> after initially developing it within the `dune-composites` module;
- interfaces to handle composite applications, including stacking sequences, complex part geometries, defects and non-standard boundary conditions, such as multi-point constraints or periodicity;
- to overcome shear locking of standard FEs, mesh stabilisation strategies to support reduced integration [139], as well as a new 20-node 3D serendipity element (with full integration) have been implemented;
- interfaces to other state-of-the-art parallel solvers (& preconditioners) in `dune-istl` [7] and `Hypre` [8];
- a code structure which supports both engineering end-users, and those requiring flexibility to extend any aspect of the code in a modular way to introduce new applications, solvers or material models.

The purpose of this chapter is to highlight the novel mathematical aspects of the code and document its structure. We illustrate its use through a range of industry motivated examples, which should enable other scientists to build on and extend

---

---

`dune-composites` for use in their own applications. We begin by outlining the mathematical formulation of the new robust preconditioner and its implementation on a distributed memory computer in Sec. 3.3. We then provide details of the structure and salient features of the code in Sec. 3.4 including the pertinent code blocks in Appendix A. Finally in Sec. 3.5, through the use of a series of example problems, we provide details of how to implement, build and run your own applications. We also use these examples as an opportunity to demonstrate the computational efficiency of `dune-composites`.

### 3.2 Preliminaries : Anisotropic elasticity equations and their finite element discretisation

A composite structure occupies the domain  $\Omega \subset \mathbb{R}^3$  with the boundary  $\Gamma$  and a unit, outward normal vector  $\mathbf{n} \in \mathbb{R}^3$ . At each point  $\mathbf{x} \in \Omega$ , we define a vector valued displacement  $\mathbf{u}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^3$ . In each of these three global directions, the boundary may contain a Dirichlet component  $\Gamma_D^{(i)}$  and a Neumann component  $\Gamma_N^{(i)}$ , such that

$$\Gamma = \Gamma_D^{(i)} \cup \Gamma_N^{(i)} \quad \text{and} \quad \Gamma_D^{(i)} \cap \Gamma_N^{(i)} = \emptyset, \quad i = x, y, z \quad (3.1)$$

Let  $\sigma_{ij}$  denote the Cauchy stress tensor and  $\mathbf{f}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^3$  the body force per unit volume. The infinitesimal strain tensor, is defined as the symmetric part of the displacement gradients

$$\epsilon_{ij}(\mathbf{u}) = \frac{1}{2} (u_{i,j} + u_{j,i}). \quad (3.2)$$

where  $u_{i,j} = \frac{\partial u_i}{\partial x_j}$ . The strain tensor is connected to Cauchy stress tensor via the generalised Hooke's law

$$\sigma_{ij}(\mathbf{u}) = C_{ijkl}(\mathbf{x}) \epsilon_{kl}(\mathbf{u}). \quad (3.3)$$

$C_{ijkl}(\mathbf{x})$  is a symmetric, positive definite fourth order tensor. A composite laminate is made up of a stack of composite layers (or plies  $\sim 0.2\text{mm}$ ), separated by a very thin layer of resin ( $15\mu\text{m}$ ). A single composite layer is modelled as a homogeneous orthotropic elastic material, characterised in general by 9 parameters and a vector of orientations  $\boldsymbol{\theta}$ . Resin interfaces are assumed isotropic, defined by just 2 scalar (Lamé) parameters. These fibres are aligned in local coordinates and can be rotated in any direction using standard tensor rotations, for more details see e.g. [140].

Given functions  $h_i : \Gamma_D^{(i)} \rightarrow \mathbb{R}$  and  $g_i : \Gamma_N^{(i)} \rightarrow \mathbb{R}$ , prescribing the Dirichlet and Neumann boundary data (for each component), we seek the unknown displacement

---

field  $\mathbf{u}(\mathbf{x})$ , which satisfies the force equilibrium equations and the boundary conditions,

$$\nabla \cdot \underline{\sigma}(\mathbf{u}) + \mathbf{f} = 0, \quad \mathbf{x} \in \Omega, \quad u_i = h_i \text{ for } \mathbf{x} \in \Gamma_D^{(i)} \quad \text{and} \quad \sigma_{ij} n_j = g_i \quad \text{for } \mathbf{x} \in \Gamma_N^{(i)}, \quad (3.4)$$

as well as eqs. (3.2) and (3.3). Then, we define the function space for each component of displacement  $u_i$  to be

$$V^{(i)} := \{v \in H^1(\Omega) : v_i(\mathbf{x}) = h_i, \quad \mathbf{x} \in \Gamma_D^{(i)}\}, \quad (3.5)$$

leading to the weak formulation of (3.4), of finding  $\mathbf{u} \in V := V^{(1)} \otimes V^{(2)} \otimes V^{(3)}$  such that

$$a(\mathbf{u}, \mathbf{v}) := \int_{\Omega} \sigma_{ij}(\mathbf{u}) \epsilon_{ij}(\mathbf{v}) \, dx = \int_{\Gamma_n} \sigma_{ij} n_j v_i \, ds - \int_{\Omega} f_i v_i \, dx := b(\mathbf{v}), \quad \forall \mathbf{v} \in V. \quad (3.6)$$

We consider the discretisation of the variational equation (3.6) with conforming FEs on a mesh  $\mathcal{T}_h$  on  $\Omega$ . Let  $V_h \subset V$  denote the restriction of  $V$  onto a FE space on  $\mathcal{T}_h$  and seek an approximation  $\mathbf{u}_h \in V_h$  such that

$$a(\mathbf{u}_h, \mathbf{v}_h) - b(\mathbf{v}_h) = 0, \quad \text{for all } \mathbf{v}_h \in V_h. \quad (3.7)$$

We block together displacements from all three space dimensions, so that  $\mathbf{u}_h^{(i)} \in \mathbb{B} = \mathbb{R}^3$  denotes the vector of displacement coefficients containing all space components associated with the  $i^{th}$  basis function. We introduce the (vector-valued) FE basis for  $V_h$  defined by the spanning set of (vector-valued) shape functions  $\{\phi^{(i)}(\mathbf{x})\}_{i=1}^n$ . These are the normal scalar shape functions, repeated for each displacement component. Therefore the vector displacement at a point is given by  $\mathbf{u}_h(\mathbf{x}) = \sum_{i=1}^n (\mathbf{u}_h^{(i)})^T \phi^{(i)}(\mathbf{x})$ . The choice of basis converts (3.7) into a symmetric positive-definite (spd) system of algebraic equations

$$\mathbf{A} \tilde{\mathbf{u}} = \mathbf{b} \quad \text{where} \quad \mathbf{A} \in \mathbb{B}^n \times \mathbb{B}^n \quad \text{and} \quad \mathbf{b} \in \mathbb{B}^n \quad (3.8)$$

where the blocks in the global stiffness matrix and in the load vector, for any  $i, j = 1, \dots, n$ , are given by  $\mathbf{A}_{ij} = a(\phi^{(i)}, \phi^{(j)})$  and  $\mathbf{b}_i = b(\phi_i)$ . The vector  $\tilde{\mathbf{u}} = [\mathbf{u}_h^{(1)}, \dots, \mathbf{u}_h^{(n)}]^T \in \mathbb{B}^n$  is the block vector of unknown FE coefficients. System (3.8) is assembled element-wise from (3.6), using Gaussian integration.

---

### 3.3 A robust, scalable, parallel iterative solver for composite structures

The key innovation of `dune-composites`, as a software package, is the design and implementation of a highly robust, scalable parallel iterative solver for composite applications. This solver is applicable to a more general class of problems and been made available in the `dune-pdelab` module. In this section, we provide the mathematical and implementation details of the new solver. Apart from new types of FEs that had to be implemented in DUNE, the remainder of the package largely provides interfaces to handle the set-up for complex composite applications.

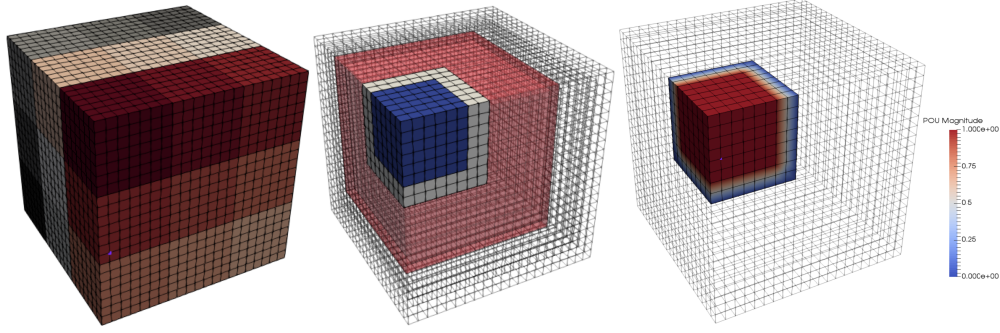
#### 3.3.1 Krylov subspace methods preconditioned with two-level additive Schwarz methods

In `dune-composites` we use *preconditioned Krylov subspace methods* both in sequential and parallel, as provided by DUNE’s “Iterative Solver Template Library” `dune-istl` [7]. Krylov subspace methods are iterative solvers which construct a sequence of approximations  $\mathbf{u}^{(k)}$  in the  $k$ -dimensional subspace:

$$\mathcal{K}_k = \text{span}\{\mathbf{r}, \mathbf{A}\mathbf{r}, \mathbf{A}^2\mathbf{r}, \dots, \mathbf{A}^{k-1}\mathbf{r}\} \subset \mathbb{R}^k \quad (3.9)$$

where  $\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{u}}^{(0)}$  is the initial residual. The simplest Krylov subspace method for a symmetric positive-definite matrix  $\mathbf{A}$  is the Conjugate Gradient method (CG), first introduced by Hestenes and Stiefel (1952). In each step, the approximate solution  $\tilde{\mathbf{u}}^{(k)}$  is updated by adding the search direction  $\mathbf{d}^{(k)}$  scaled by a factor chosen to minimise the energy norm over the space  $\mathcal{K}_k$ . The search directions are chosen to be  $\mathbf{A}$ -orthogonal to all previous directions i.e.  $\langle \mathbf{d}^{(k)}, \mathbf{A}\mathbf{d}^{(k')}\rangle = 0$  for  $k' < k$ . The method iterates until the residual norm (or “energy”)  $\|\mathbf{r}^{(k)}\|$  reduces below a user defined tolerance. Importantly, the convergence rate of CG depends on the spectral properties of the matrix  $\mathbf{A}$ , see e.g. [141]. In particular, it can be bounded proportionally to the square root of the condition number  $\kappa$ , defined as the ratio between its largest and smallest eigenvalue. A large value, as usually seen in composites, indicates that the system  $\mathbf{A}\tilde{\mathbf{u}} = \mathbf{b}$  is ill-conditioned. This means that  $\mathbf{u}$  is very sensitive to small changes in  $\mathbf{b}$ . For such cases, iterative solvers converge very slowly or even not at all, particularly when the problem size increases. A remedy is to *precondition* the system, that is to develop an operation  $\mathbf{M}^{-1}$  which is computationally cheap to construct and apply (in parallel) such that  $\mathbf{M}^{-1}\mathbf{A}\tilde{\mathbf{u}} = \mathbf{M}^{-1}\mathbf{b}$  is better conditioned and CG solvers converge quickly.

In `dune-composites` our main preconditioner is a two level additive Schwarz method.



**Figure 3-1:** (Left) Domain  $\Omega$  partitioned into non-overlapping subdomains  $\Omega'_j$  where colouring differentiates independent subdomains. (Middle) Shows overlapping subdomain  $\Omega_j$  with a single layer of overlap ( $O = 1$ ). Overlap region  $\Omega_j^o$  is shown in white. Transparent red regions show cells of the grid which belong to 'nearest neighbour' processors. (Right) Shows partition of unity (PoU) operator  $\Xi_j$  on a single processor, defined as in (3.12).

To construct this method we partition our domain  $\Omega$  into a set of non-overlapping subdomains  $\Omega'_j$  for  $j = 1$  to  $N$  resolved by  $\mathcal{T}_h$ , as shown in Fig. 3-1 (left). Each subdomain  $\Omega'_j$  is extended by  $O$ -layers of elements to give the overlapping subdomains  $\Omega_j$ , Fig. 3-1 (middle). For each subdomain  $1 \leq j \leq N$ , we denote the restriction of  $V_h$  to  $\Omega_j$  by  $V_h(\Omega_j)$ , whilst the space of FE functions with support contained in  $\Omega_j$  is called  $V_{h,0}(\Omega_j)$ .

*Remark:* In `dune-composites` the user can define the initial non-overlapping decomposition (or a default is used), the overlapping process is handled by DUNE's parallel structured grid class `Dune::YaspGrid` [136].

Any function  $v \in V_{h,0}(\Omega_j)$  is mapped onto  $V_h$  by the prolongation operator  $R_j^T : V_{h,0}(\Omega_j) \rightarrow V_h$ , which extends  $v$  by zeros, so that

$$R_j^T v(\mathbf{x}) = \begin{cases} v(\mathbf{x}), & \mathbf{x} \in \Omega_j \\ 0, & \mathbf{x} \in \Omega \setminus \Omega_j \end{cases}.$$

We therefore note that the restriction operator  $R_j : V_h \rightarrow V_{h,0}(\Omega_j)$ . In matrix form the restriction and prolongation operators  $R_j$  and  $R_j^T$ , are denoted  $\mathbf{R}_j$  and  $\mathbf{R}_j^T$  respectively. This allows us to define the subdomain stiffness matrices restricted to  $V_{h,0}(\Omega_j)$  as  $\mathbf{A}_j := \mathbf{R}_j \mathbf{A} \mathbf{R}_j^T$  for  $j = 1, \dots, N$ . In practice, we do not compute  $\mathbf{A}_j$  from  $\mathbf{A}$  via this double matrix product. Instead, we can equivalently assemble  $\mathbf{A}_j$  directly from the bilinear form (3.7) on  $\Omega_j$  with homogeneous Dirichlet boundary conditions on all artificial interior subdomain boundaries, i.e. all points  $\mathbf{x} \in \partial\Omega_j$  that satisfy  $\mathbf{x} \in \Omega_{j'}$  for some other  $j' \neq j$ .

The 1-level Additive Schwarz method can then be defined as a preconditioner of

---

(3.8) via the operator

$$\mathbf{M}_{AS,1}^{-1} = \sum_{j=1}^N \mathbf{R}_j^T \mathbf{A}_j^{-1} \mathbf{R}_j \quad (3.10)$$

Here, in the subscript, the  $AS$  denotes additive Schwarz and the 1 denotes a one-level method. In this case the preconditioner  $\mathbf{M}_{AS,1}^{-1}$  approximates the inverse operator  $\mathbf{A}^{-1}$  by a sum of local solves on overlapping subdomains, with homogeneous Dirichlet boundary conditions on interior boundaries. We will see in the numerical examples to follow that, for large problems, a single-level method is not sufficient, causing stagnation (high iteration counts) of the iterative solver. This stagnation of the iterative solver is caused by a few very small eigenvalues in the spectrum of the preconditioned problem. They are due to the lack of a global exchange of information in the preconditioner in the single-level method. A classical remedy is the introduction of a coarse grid problem that couples all subdomains at the second level [123]. To define our coarse problem, we introduce a coarse space  $V_H \subset V_h$  (which we define below). We denote the restriction from the fine to the coarse space by the operator  $R_H : V_h \rightarrow V_H$ , with matrix representation  $\mathbf{R}_H$ . The two-level additive Schwarz preconditioner (in matrix form) is given by

$$\mathbf{M}_{AS,2}^{-1} = \mathbf{R}_H^T \mathbf{A}_H^{-1} \mathbf{R}_H + \mathbf{M}_{AS,1}^{-1} \quad \text{where} \quad \mathbf{A}_H = \mathbf{R}_H \mathbf{A} \mathbf{R}_H^T. \quad (3.11)$$

Two natural questions arise:

- What is a good choice of coarse space  $V_H$  for composite applications?
- How do we construct  $\mathbf{A}_H$  efficiently on a distributed memory computer without assembling  $\mathbf{A}$  directly?

### 3.3.2 A robust coarse space via Generalised Eigenproblems in the Overlaps (GenEO)

The ideal coarse space would capture the global low energy modes of  $\mathbf{A}$  that jeopardise the performance of Krylov solvers. Specifically, in the two-level additive Schwarz setting, the modes not captured by the local solves are of interest. Yet, to compute those low-energy modes explicitly would be more expensive than inverting  $\mathbf{A}$  itself. Instead, the global low-energy modes can be approximated by stitching together local (optimal) approximations. These local approximations are solutions of specific Generalised Eigenproblems in the Overlaps, hence named **GenEO**, defined below. Importantly the local eigenproblems are independent and can trivially be computed in parallel. The robustness of **GenEO** has been proven for isotropic elasticity problems, Spillane et al.



---

[135], and numerically verified by the authors for anisotropic variants [25].

The construction of the **GenEO** coarse space has two key steps: the definition of the generalised eigenproblems on the subdomains and the stitching together of the resulting local eigenmodes from each subdomain to form a global basis. This stitching process by means of *partition of unity* (PoU) operators is also incorporated in the local eigenproblems, therefore we construct the PoU operators first.

**Definition 3.3.1.** (*Subdomain Overlap*). For each subdomain  $\Omega_j$ , the overlap region is defined by the set

$$\Omega_j^\circ := \{\mathbf{x} \in \Omega_j : \exists j' \neq j \text{ s.t. } \mathbf{x} \in \Omega_{j'}\},$$

i.e. the subset of  $\Omega_j$  which belongs to at least one other subdomain.

**Definition 3.3.2.** (*Partition of Unity*). The family of operators  $\Xi_j : V_h(\Omega_j) \rightarrow V_{h,0}(\Omega_j)$ ,  $j = 1, \dots, N$ , defines a *Partition of Unity* if

$$\sum_{j=1}^N R_j^T \Xi_j(v|_{\Omega_j}) = v, \quad \forall v \in V_h.$$

Since  $R_{j'}^T \Xi_{j'}(v|_{\Omega_{j'}}) = 0$  on  $\Omega_j \setminus \Omega_j^\circ$  for all  $j' \neq j$ , it follows from this definition that restricted to  $\Omega_j \setminus \Omega_j^\circ$  each  $\Xi_j$  has to be the identity operator. In the overlaps, the choice of  $\Xi_j$  is not unique. The simplest approach is to define  $\Xi_j(v)$  such that each coefficient of the FE function  $v$  is scaled by the number of subdomains the corresponding degree of freedom belongs to (see [135] for details). However, we also provide a smoother PoU as defined by Sarkis [142] in our implementation, but observe little difference in the performance of the overall solver (at most one iteration); we therefore keep the presentation here as simple as possible.

Given this set of local PoU operators  $\Xi_j(\cdot)$ , we can construct any global FE function  $v_h \in V_h$  from local functions  $v_h^{(j)} \in V_h(\Omega_j)$  as follows:

$$v_h = \sum_{j=1}^N R_j^T \Xi_j(v_h^{(j)}). \quad (3.12)$$

In particular, we can define the local generalised eigenproblems that (once collected from each subdomain) provide the basis of the **GenEO** coarse space. The following definition can be rigorously motivated from theoretical considerations and we refer again to [135]. For each subdomain  $\Omega_j$ ,  $j = 1, \dots, N$ , we define the generalised eigenproblem: Find  $(\lambda, p) \in \mathbb{R}^+ \times V_h(\Omega_j)$  such that

$$a_{\Omega_j}(p, v) = \lambda a_{\Omega_j^\circ}(\Xi_j(p), \Xi_j(v)), \quad \forall v \in V_h(\Omega_j), \quad (3.13)$$


---

---

where, for any  $D \subset \Omega$ , the bilinear form  $a_D$  is defined like  $a$  in (3.6) with the integral restricted to  $D$ .

**Definition 3.3.3.** (*GenEO coarse space*). For each subdomain  $\Omega_k$  let  $p_k^{(j)}$  be the eigenfunctions from (3.13) with associated eigenvalues  $\lambda_k^{(j)}$  in ascending order. Then, for some choice of  $m_j \in \mathbb{N}$ , the **GenEO** coarse space is defined as

$$V_H := \text{span}\{R_j^T \Xi_j(p_k^{(j)}) : k = 1, \dots, m_j, j = 1, \dots, N\}.$$

The only parameter that remains to be chosen is the number of eigenmodes  $m_j$  to be included in each subdomain  $\Omega_j$ . In order to ensure robustness and scalability of the solver, the condition number of the preconditioned system needs to be bounded from above, independent of  $N$ ,  $h$  and of the material properties. It has been shown in [135] that

$$\kappa(\mathbf{M}^{-1}_{AS,2}\mathbf{A}) \leq C \max_{1 \leq j \leq N} \left( 1 + \frac{1}{\lambda_{m_j+1}^{(j)}} \right). \quad (3.14)$$

where  $C > 0$  is a constant depending only on the geometry of the subdomains and where  $\lambda_{m_j+1}^{(j)}$  is the lowest eigenvalue whose eigenfunction is not added to the coarse space on  $\Omega_j$ . Thus, the desired robustness can be achieved by including all eigenfunctions in the coarse space whose eigenvalues are below an *a priori* chosen threshold. A particular threshold that turns out to provide an effective black-box choice for  $m_j$  and also depends only on the geometry of the subdomain partition is to include all eigenfunctions with  $\lambda_k^{(j)} \leq \text{diam}(\Omega_j)/\text{width}(\Omega_j)$ .<sup>1</sup> This simple threshold can be scaled by a constant factor, thus also scaling the condition bound of the preconditioned system by the same factor. As the number of iterations of the Krylov solver depends directly on the condition number, this allows us to balance the time spent in the iterative solver with the time spent on setting up the preconditioner.

The number of eigenfunctions that are used in the coarse space is problem-specific, but it turns out that for strongly structured coefficient distributions only a small number is typically sufficient. We will see in Section 3.5 that the calculation of these local eigenmodes is not prohibitively expensive, while yielding excellent condition numbers and, due to the independence of the individual eigenproblems, parallel scalability.

---

<sup>1</sup>For any  $D \subset \Omega$ ,  $\text{diam}(D)$  and  $\text{width}(D)$  refer to the radius of the largest circumscribed and inscribed circle, respectively.

---

### 3.3.3 Implementation of GenEO on a High Performance Computer

The two-level additive Schwarz preconditioner with GenEO coarse space is implemented within a collection of header files, which are located in `dune-pdelab` from `releases/2.6` and in the folder `solvers/geneo/` in prior releases. Here we describe our implementation. We are aware of only one other high performance implementation of GenEO, which can be found in the package HPDDM for which details are provided in Jolivet *et al.* [143, 144].

It is a main goal of such an implementation to fully exploit the excellent parallel scalability promised by the method’s construction and theoretical properties. Therefore, each process  $j$  will be assigned to subdomain  $\Omega_j$  and only store relevant fine-level operators and functions in the form of local restrictions to  $\Omega_j$ . Further, per-subdomain stiffness matrix and eigenproblem solves will be run in parallel on the respective processes. Only scalable nearest-neighbour communication is needed, with the exception of setting up the coarse matrix, which consequently requires particular care.

#### Partition of Unity (PoU) operator

The partition of unity operator as defined by Def. 3.3.2 is stored locally on each processor (see Fig. 3-1). In practice, the partition of unity operator  $\Xi_j$  is represented as a diagonal matrix  $\mathbf{X}^{(j)}$ . In the simplest case, each diagonal entry of  $\mathbf{X}^{(j)}$  is set to one divided by the number of subdomains containing the associated degree of freedom, except for the subdomain boundary where entries are set to zero. Therefore, if  $\mathbf{v}^{(j)}$  is a vector containing all nodal degrees of freedom of the FE function  $v_h \in V_h(\Omega_j)$  in subdomain  $\Omega_j$ , the operation  $\mathbf{X}^{(j)}\mathbf{v}^{(j)}$  automatically maps  $v_h$  into  $V_{h,0}(\Omega_j)$ . Such a PoU can be generated using existing parallel data structures in DUNE by adding a vector of ones using an `AddDataHandle` and by enforcing both global and subdomain boundary conditions before and after communication. The implementation of the PoU operators is within the header file `geneo/partitionofunity.hh` under the function `standardPartitionofUnity(...)`. As the choice of partition of unity operator is not unique, we also provide the PoU in [142], which is implemented in the same header file under the function `sarkisPartitionofUnity(...)`. This gives a ‘smoother’ PoU operator, which is however restricted to equally distributed subdomain sizes. Under testing, we noted no significant difference in the performance of the preconditioner when changing between the two different PoU operators.

---

### Subdomain eigenproblems

The local generalised eigenvalue problems (3.13) can be rewritten in matrix form as follows: Find eigenpairs  $(\lambda_i^{(j)}, \mathbf{p}_i^{(j)})$  with eigenvalues in ascending order and  $\|\mathbf{p}_i^{(j)}\| = 1$  such that

$$\mathbf{A}_{\Omega_j} \mathbf{p}_i^{(j)} = \lambda_i^{(j)} \left( \mathbf{X}^{(j)} \mathbf{A}_{\Omega_j^o} \mathbf{X}^{(j)} \right) \mathbf{p}_i^{(j)}, \quad \text{for } j = 1, \dots, N \quad \text{and} \quad i = 1, \dots, m_j \quad (3.15)$$

where  $\mathbf{A}_{\Omega_j}$  and  $\mathbf{A}_{\Omega_j^o}$  denote the stiffness matrices corresponding to the bilinear forms  $a_{\Omega_j}(\cdot, \cdot)$  and  $a_{\Omega_j^o}(\cdot, \cdot)$  on  $V_h(\Omega_j)$  and  $V_h(\Omega_j^o)$ , respectively. They are solved using ARPACK++ [145].

A customised wrapper has been developed to convert DUNE data structures into a suitable format for ARPACK++ [145]. In order to regularise the problem, we employ ARPACK++'s *shift and invert spectral transformation mode* and, since we are interested in the smallest  $m_j$  eigenvalues, we choose a small shift factor. The global coarse basis vectors  $\Phi_1, \dots, \Phi_{N_H}$  are obtained from the local eigenvectors by applying the PoU operator, i.e.,  $\Phi_{i(j,k)} := \mathbf{X}^{(j)} \mathbf{p}_k^j$ , and padding the rest of the global vector (outside  $\Omega_j$ ) with zeros. Here,  $N_H = m_j N$  is the total number of modes in the coarse space.

### Coarse space assembly

The parallel assembly of the coarse system  $\mathbf{A}_H = \mathbf{R}_H \mathbf{A} \mathbf{R}_H^T$  is not trivial in practice since process  $j$  only has local access to rows and columns of  $\mathbf{A}$  associated to degrees of freedom on sub-domain  $\Omega_j$ . We denote this submatrix  $\tilde{\mathbf{A}}_j$ . Note that  $\tilde{\mathbf{A}}_j$  differs from the matrix  $\mathbf{A}_j$  in (3.10) in that it does not incorporate Dirichlet conditions on interior subdomain boundaries.

Furthermore the coarse space prolongation matrix  $\mathbf{R}_H^T$  is only available in a distributed manner. Each basis vector  $\Phi_i$ ,  $i \in \{1, \dots, N_H\}$ , is available only on process  $j(i)$ , where the unique  $j(i) \in \{1, \dots, N\}$  denotes the index of the subdomain  $\Omega_{j(i)}$  associated with the eigenproblem (3.15) corresponding to  $\Phi_i$ . However, due to the local support of the basis functions, one can break down the global matrix product into local products

$$(\mathbf{A}_H)_{i,\ell} = (\mathbf{R}_H \mathbf{A} \mathbf{R}_H^T)_{i,\ell} = \left( \Phi_i^T \tilde{\mathbf{A}}_{j(i)} \right) \Phi_\ell, \quad \text{for } i, \ell = 1, \dots, N_H, \quad (3.16)$$

with a slight abuse of notation, denoting the local parts of the global vectors  $\Phi_i$  and  $\Phi_\ell$  restricted to  $\Omega_{j(i)}$  again by  $\Phi_i$  and  $\Phi_\ell$ . In the implementation, the matrix vector product in the bracket is local whereas the scalar product requires communication of (parts of) vector  $\Phi_\ell$  from processor  $j(\ell)$  to processor  $j(i)$ . This avoids having to

---

communicate the local matrices  $\tilde{\mathbf{A}}_j$ .

We note that the locality of the basis functions implies  $\Phi_i \tilde{\mathbf{A}}_{j(\ell)} \Phi_\ell = 0$  whenever  $\Omega_{j(i)} \cap \Omega_{j(\ell)} = \emptyset$ . Therefore, the parallel assembly of  $\mathbf{A}_H$  requires only communication between processes assigned to overlapping subdomains. This locality of communication, which is demonstrated in Fig. 3-1 (middle), can be exploited to set up  $\mathbf{A}_H$  as a banded sparse matrix. Its parallel communication is implemented using a customised `DataHandle` implemented within `geneo/multicommdatahandle.hh`, allowing to pass basis functions between all processes at the same time and therefore make best use of available bandwidth. Combining sparsity and efficient communication, linear complexity in basis size can be achieved for this step.

Since the number of coarse degrees of freedom per process is too small, i.e.,  $m_j$  on process  $j$ , after assembly we distribute the resulting global coarse matrix  $\mathbf{A}_H$  to all processors and duplicate the coarse solve on all processors to avoid fine-grained communication. The communication of  $\mathbf{A}_H$  is achieved directly with MPI calls, as the `dune-pdelab` communication infrastructure is not designed for such global operations.

In case of the restriction of a distributed vector  $\mathbf{v}_h$  representing a function  $v_h \in V_h$ , it follows that

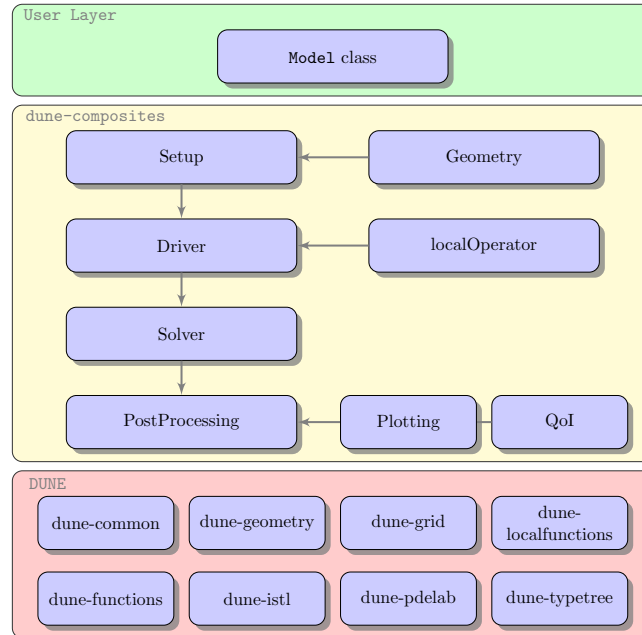
$$(\mathbf{R}_H \mathbf{v}_h)_i = \Phi_i^T \mathbf{v}_h. \quad (3.17)$$

So, each row  $i$  can be computed by the process associated with  $\Phi_i$ , and the rows can be exchanged among all processes via `MPI_Allgatherv`. Again, the communication effort increases with the dimension of  $V_H$ . On the other hand, the prolongation  $\mathbf{R}_H^T \mathbf{v}_H$  of a vector  $\mathbf{v}_H$  that is globally available on all processors, representing a  $v_H \in V_H$ , consists only of local contributions and hence can be computed in parallel without communication. Since the result lies in  $V_h(\Omega)$ , the regular PDELab communication patterns can be used. This only involves communication between adjacent subdomains, making this a highly scalable process.

Each process executes its associated subdomain solve as well as the coarse space solve (redundantly). Where possible we use a sparse direct solver (`UMFPack`) [146]. For very large problems and a large number of parallel processors the coarse space becomes too large, and must itself be solved with a preconditioned iterative solver; in that case we use by default preconditioned CG with the `BOOMERAMG` [8] preconditioner. It is important to note that in such cases since the coarse solve is inexact, the preconditioner for the (overall) Krylov method is now instationary. It is therefore necessary to switch from a standard preconditioned CG to a flexible Krylov solver. In our case we use Flexible GMRES as provided by `dune-istl` [7].

---

### 3.4 Using and extending dune-composites



**Figure 3-2:** *Code structure.*

In this section we provide an overview of the `dune-composites` code, sufficient to enable other scientists to leverage the framework. The code is structured so that little additional knowledge of DUNE and/or C++ is required to apply the code within the existing functionality. Figure 3-2 shows the code structure. A user can extend any of the functionality e.g. implement a new solver, define a more complex nonlinear problem (e.g. cohesive zone) or introduce new types of elements.

#### 3.4.1 Defining a Model

At the highest level an analysis is defined by a user-defined `BaseStructuredGridModel` class shown in green in Figure 3-2. This class defines all the key variables, functions and classes which describe the analysis, as well as storing any variables that are required for postprocessing or any later calculations. A base model class is provided (`baseStructureGridModel`), which can be inherited by each example. This provides default variables and functions, so that the user need only overwrite those functions which deviate from this base class. The `Model` class also defines the general loading on the structure and the boundary conditions, these include Dirichlet and Neumann conditions, but also thermal loading and multi-point constraints. Periodic boundary conditions are defined within the grid data structure, using `Model::LayerCake()`.

---

Examples of user defined `Model` classes for a series of applications are provided in Section 3.5.

### 3.4.2 Internals of dune-composites

The functions provided in the subfolder `/Setup` provide support for the geometric setup of the grid geometry, material properties and boundary conditions. This includes the composite layering (or stacking sequence), the structural geometric shape of the component and in some cases adding a perturbation to the geometry to form a defect (see for example Sec. 3.5.1). Because of uniform layering and planar anisotropy in composite laminates, our first version has focused on structured, overlapping grid implementations using `Dune::YaspGrid` and `Dune::GeometryGrid`. The `Dune::GeometryGrid` functionality allows us to apply any continuous transformation to the basic cartesian mesh provided by `Dune::YaspGrid`. Development of unstructured grid implementation using `Dune::UGGrid` [147] are the subject of future work.

The folder `/Driver` contains the key functions and classes which relate to the FE calculations beyond what is available directly from `Dune::PDELab`. In particular these include all element calculations, the definitions of new FEs, solvers and preconditioners. The functions and classes are split between three folders:

- `/localOperators` define the weak form of the equations to be solved on an element, along with any support functions. In our case for anisotropic linear elasticity equations we define the local operator `linearelasticity.hh` which returns the element stiffness matrix, load vector and residual as defined by equation (3.6).
- `/FEM` defines specialist finite elements beyond those defined by `Dune::PDELab`. In our case, these are the family of serendipity elements [148]. The use of these elements are then defined explicitly in the `Driver` class, where the FE space is set up on the grid.
- `/Solvers` defines specialist solvers and preconditioners beyond those defined in `Dune::PDELab::istl` [7]. The `Driver` uses the solver as defined by the `Model` class, defined by the templated class function `Model::solve()`. By default, as defined by `BaseStructuredGridModel`, for parallel calculations we use a CG Krylov solver, preconditioned with either a one or two level additive Schwarz method. Two-level methods use `GenEO` as the coarse space as long as `ARPACK++` is available. If not, the coarse space consists of only the zero energy modes [123]. In sequential mode, in particular for the coarse and the local solves, a sparse

---

direct solver `UMFPack` [146] and the iterative CG preconditioned with AMG as provided by `Dune::PDELab::ISTLBackend_SEQ_CG_AMG_SSOR`, are also available. In `/Solvers/hypre` we provide a wrapper to the external parallel solvers provided by `hypre` [149], including `boomerAMG` [8].

The analysis is defined via a `Driver`. This is a class which provides the complete solution procedure, from the setup of the grid, the definition of a finite element space, assembly of the global stiffness matrix and load vector, and finally, calling the solver, followed by the Post Processing routines, as defined by the `Model` class. In this version, we provide two driver classes for the examples considered in Section 3.5: `/FEMDriver/linearStatic.hh` and `/FEMDriver/ThermalStatic.hh`.

The folder `/PostProcessing` contains various classes and functions for the postprocessing of results. By default, after the solution has been calculated the driver initiates certain post processing steps, in particular the calculation of stresses, the creation of the necessary data for any plots and finally the calculation of any further quantities of interest. All of these routines can be modified by the user.

### 3.5 Examples

In this section, we introduce and demonstrate the functionalities of `dune-composites` using a series of examples of increasing complexity. The examples are intended as a starting point for researchers implementing their own studies, whilst also demonstrating the significant computational gains `dune-composites` is able to achieve in comparison to the commercial package `ABAQUS` [4].

To simplify the definition of the examples in the following we assume that all cases use the same material properties. The orthotropic fibrous layers are assumed to be of thickness  $t_p = 0.23\text{mm}$ , with elastic moduli

$$E_{11} = 162\text{GPa}, \quad E_{22} = E_{33} = 10\text{GPa}, \quad G_{12} = G_{13} = 5.2\text{GPa}, \quad G_{23} = 3.5\text{GPa}, \quad (3.18)$$

$$\nu_{12} = \nu_{13} = 0.35 \quad \text{and} \quad \nu_{23} = 0.5,$$

whereas the isotropic resin rich layers are assumed to be  $t_i = 0.02\text{mm}$  thick, with isotropic properties  $E = 10\text{GPa}$  and  $\nu = 0.35$ . These particular values are taken from a previous study by the authors [150].



---

### 3.5.1 Example 1: A flat composite plate

For the first two examples we consider a flat composite plate  $[0, 100\text{mm}] \times [0, 20\text{mm}]$  under various loading conditions. The laminate is made up of 12 identical composite layers arranged in the following composite stacking sequence

$$[\mp 45^\circ / 0^\circ / 90^\circ / \pm 45^\circ / \mp 45^\circ / 90^\circ / 0^\circ / \pm 45^\circ]. \quad (3.19)$$

The composite layers are separated by 11 isotropic resin interface layers, giving a total thickness of  $T = 2.98\text{mm}$ . In each of the examples, we discretise the geometry with quadratic, 20-node serendipity elements (with full Gaussian integration). For the base mesh, which will be refined, we take 20 elements in the  $x$ -direction, 5 in the  $y$ -direction and through thickness 2 per composite layer and 1 per interface layer. This gives a total number of 3,500 elements, with 13,608 degrees of freedom.

The geometry, the stacking sequence and the initial finite element mesh are introduced into a model by overwriting the base class function `Model::LayerCake()`, with user defined function given in Algorithm 4 in Appendix A. Here the geometry and grid are defined by a file `stackingSequences/example1.csv`.

In these first two examples, we demonstrate a very simple setup and run on a single processor as well as on a few processors. We consider a cantilever beam with a uniform pressure of 0.01 MPa applied to the top face and the following boundary conditions:

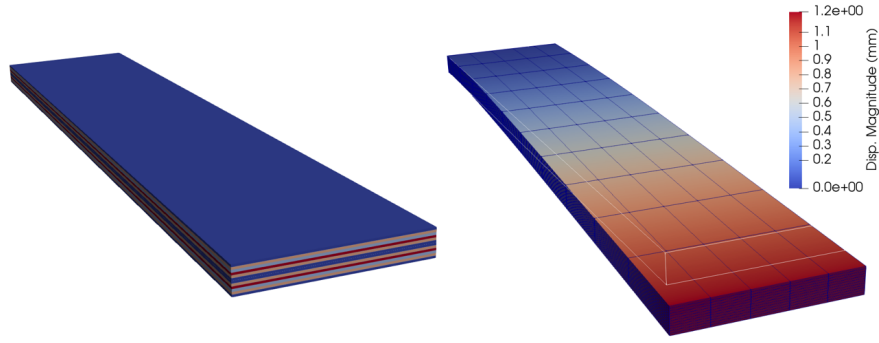
$$u_1 = u_2 = u_3 = 0 \quad \text{at} \quad x = 0 \quad \text{and} \quad \sigma_{33} \cdot \mathbf{n}_3 = -q \quad \text{at} \quad z = T. \quad (3.20)$$

All other boundary conditions are assumed to be homogeneous Neumann conditions, i.e.

$$\sigma_{ij} \cdot n_j = 0. \quad (3.21)$$

Boundary conditions are implemented by overwriting the two class functions `Model::isDirichlet()` and `Model::evaluateNeumann()`. The implementation can be found in Appendix A Algorithms 5 and 6, respectively.

We note that `Model::evaluateDirichlet()` need not be overwritten since by default it returns homogeneous boundary conditions (i.e.  $\mathbf{u}(\mathbf{x}) = \mathbf{0}$ ) for all those points  $\mathbf{x}$  marked as Dirichlet boundary conditions by `isDirichlet()`. Furthermore, by default loading under the weight of the structure is included by providing density as an input parameter. We do not wish to include it in this example and therefore we must also overwrite the function `Model::evaluateWeight()`



**Figure 3-3:** Visualisation of results for *Example01a* using PARAVIEW (left) Visual output of laminate and stacking sequence using `plotProperties()` function (right) Visualisation of solution, in deformed coordinates (scalar factor of displacement is 4).

### Example 1a: A flat composite plate – getting started

Our first study computes the maximum vertical deflection of the cantilever beam as our quantity of interest

$$Q(\mathbf{u}) = \max_{\mathbf{x} \in \Omega} u_3(\mathbf{x}).$$

This is done by providing the user defined function in Appendix A Algorithm 8. This function loops over the solution at each vertex `native(u)[i]`, and records the maximum vertical displacement `native(u)[i][2]`. Since for a parallel run, this maximum is only the maximum on the local subdomain associated with a given processor, the final command `MPI_Allreduce()` finds the maximum vertical displacement over all subdomains (processors). The final result is stored in `QoI`, a member of the `baseStructuredGridModel` class.

We use the default sequential and parallel solvers. On a single processor (e.g. with the call `./Example1a`) the sparse direct solver `UMFPack` [146] is used. Otherwise, if more than one processor is used (e.g. with the call `mpirun -np 8 ./Example1a`) the equations will be solved with CG, preconditioned with a one-level additive Schwarz preconditioner, as defined by (3.10) using `UMFPack` as the local solver on each subdomain.

As output, the quantity of interest is printed to the screen (**Maximum Vertical Displacement in Example01a = 1.23992mm**). Furthermore, the data for plots of the laminate stacking sequence, the solution (deformation) and the stress field are generated and provided in three files named `Example01a_xxx.vtu`. The stacking sequence and solution are shown in Fig. 3-3.

---

**Example 1b: A flat composite plate – testing preconditioners (up to 32 cores)**

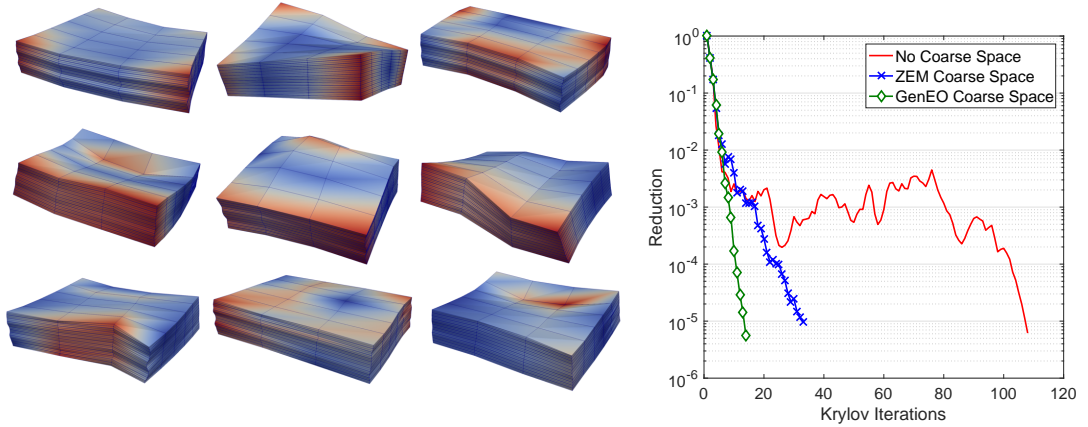
In `Example01b`, we test our new preconditioner `GenEO` on up to 32 processors. In this example we also demonstrate the inclusion of a failure criterion. To do this we change the quantity of interest to be the pressure  $q = q^*$  in the boundary condition (3.20) at which the laminate fails according to the Camanho criterion [151], defined by the functional

$$\mathcal{F}(\sigma(\mathbf{x})) = \sqrt{\left(\frac{\sigma_{33}^+}{s_{33}}\right)^2 + \left(\frac{\sigma_{13}}{s_{13}}\right)^2 + \left(\frac{\sigma_{23}}{s_{23}}\right)^2}. \quad (3.22)$$

We apply the Camanho criterion only in the resin-rich interface layers and we say that failure occurs at a load  $q^*$  if  $\max_{\mathbf{x} \in \Omega^{\text{Inter}}} \mathcal{F}(\sigma(\mathbf{x})) = 1$ . However, since the problem is linear it suffices to solve only one problem with an arbitrary load  $q$ . The failure load is then given by  $q^* := q / \max_{\mathbf{x} \in \Omega^{\text{Inter}}} \mathcal{F}(\sigma(\mathbf{x}))$ . Expression (3.22) is implemented in the file `PostProcessing/FailureCriterion/Camanho.hh` within the code, see Algorithm 9 in Appendix A. Within `linearStaticDriver`, by default, the stress field (per element) is stored within the container `stress_mech` (a  $6 \times 1$  vector). To compute  $q^*$ , the Camanho functional is first calculated in each element. The maximum is then found by once again overwriting the class function `Model::postprocess`. The material allowables,  $s_{33} = 61$  MPa,  $s_{13} = 97$  MPa and  $s_{23} = 94$  MPa, in Eq. (3.22) are stored in a `std::vector<double> p`.

Different failure criteria can be implemented by defining other user-defined functionals of the stress tensor, similar to `Dune::Composites::Camanho()`. In this simple test, we note that failure initiates due to high through thickness stresses in the interface between layers ( $\sigma_{13}$  and  $\sigma_{23}$ ) as the laminate bends. For further engineering discussion of the failure of composites under the Camanho criterion we point the reader to the original paper [151] and to [3, 150].

We use this test example to demonstrate the influence of the `GenEO` coarse space on the parallel iterative solver. For the first experiment we use 16 processors. Fig. 3-4 (left) shows the first nine non-zero energy modes of a subdomain with no global Dirichlet boundary. Linear combinations of these functions together with the zero energy modes (six rigid body translations and rotations) provide a good low dimensional representation of the system on that subdomain consisting of those modes most easily excited in the context of energy requirements. Fig. 3-4 (right) shows the influence of the coarse space on the number of iterations for the preconditioned Krylov Solver (pCG), comparing no coarse space (one-level additive Schwarz), only the zero energy modes (ZEM) and the `GenEO` coarse space. The need for a coarse space is clear; with no coarse space, even in this simple test case we observe the well-documented stagnation



**Figure 3-4:** (Left) The eigenvectors corresponding to the first nine non-zero eigenvalues on a subdomain with no global Dirichlet boundary. (Right) The reduction of the residual against CG iterations for *Example01b* using no coarse space, only zero energy modes (ZEM) and the full *GenEO* coarse space.

phenomenon for iterative solvers [123] between Iteration 10 – 100. With a coarse space (ZEM or *GenEO*), the convergence shows no stagnation and it is much faster – close to optimal with *GenEO*.

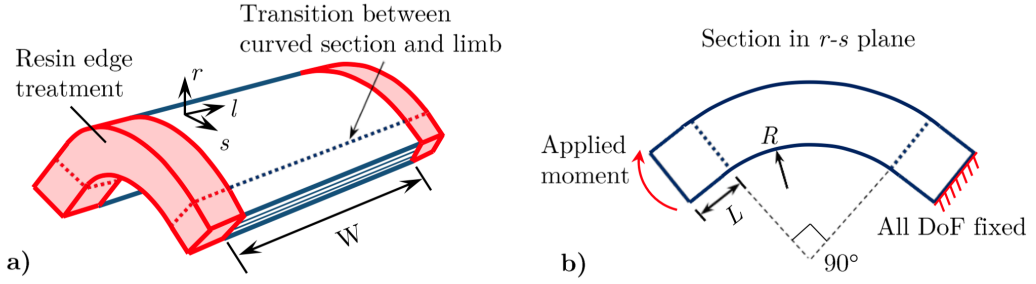
Next we want to study the robustness of *GenEO* as a function of the number of subdomains in comparison to one-level additive Schwarz (AS) and ZEM. We consider a fixed size problem and increase the number of subdomains. We note that the tests can be run with

```
mpirun -np 16 ./Example01b    or    mpirun -np 16 ./Example01bBoomerAMG
```

In each case we record the condition number, the dimension of the coarse space  $\dim(V_H)$  (if applicable) and the number of CG iterations to achieve a residual reduction of  $10^{-5}$ . The results are summarised in Table 3.1. We see that the iteration counts (and the condition number estimates) increase steadily with the number of subdomains when no coarse space is used. The condition number estimate is still fairly big if only the zero energy modes are used and the iteration counts also increase steadily with the number of subdomains. In contrast, the iterations and the condition number estimates remain constant for the *GenEO* preconditioner. We also add a comparison with *boomerAMG* [8] for this test problem. *BoomerAMG* provides a large number of parameters to fine-tune. We retained the defaults for most parameters (HMIS coarsening without aggressive refinement levels and a hybrid Gauss-Seidel smoother). We used blocked aggregation with block size 3 as recommended for elasticity problems. A strong threshold of 0.75 was chosen after testing values in the range from 0.4 to 0.9. Due to a lower setup cost with this parameter setting, the *boomerAMG* solver is faster in actual CPU time, but

$N$	AS		ZEM			GenEO				BOOMERAMG	
	it	cond $\kappa$	it	cond $\kappa$	$\dim(V_H)$	it	cond $\kappa$	$\dim(V_H)$		it	Num. levels
4	89	79,735	26	394	12	16	10	78		258	10
8	97	84,023	30	245	42	15	9	126		258	11
16*	107	98,579	36	177	84	16	10	182		257	12
32	158	226,871	42	230	168	16	9	526		263	12

**Table 3.1:** Demonstration of performance of different preconditioners for *Example01b* for fixed problem size (30,000 DOFs) but increasing the number of subdomains: Number of pCG iterations (it), coarse space dimension ( $\dim(V_H)$ ), an estimate of the condition number  $\kappa$ .



**Figure 3-5:** (Left) Diagram of the corner bend specimen with resin edge treatment. (Right) Cross section of the corner showing the loading conditions.

the numbers of iterations – albeit also constant – are more than  $10\times$  bigger. For more complex geometries, **boomerAMG** does not perform very well and in our tests it does not scale beyond about 100 cores in composite applications.

### 3.5.2 Example 2 : Corner unfolding – validation & performance comparison with Abaqus (up to 32 cores)

This example is motivated by the industrial challenge of certifying the corner-bend strength of a wingspar as its corner unfolds due to the internal fuel pressure in an aircraft wing. We use this example to demonstrate the validity of the results of **dune-composites** by comparing the stresses computed with those given by ABAQUS. We also make a cost comparison between the two software packages up to 32 cores.

The model setup is shown in Fig. 3-5. We consider the curvilinear coordinate system  $(s, r, \ell)$ , where  $s$  is around the radius,  $r$  is outwards (or normal) to the laminate and  $\ell$  runs along the length of the sample. For our particular test, the two limbs of the coupon are of length  $L = 3\text{mm}$  and border a corner with a radius of  $R = 6.6\text{mm}$ . The width is taken to be  $W = 15\text{mm}$ . The 12 plies and the 11 interfaces have the same

---

properties as in **Example01**, but the stacking sequence is slightly different, given by

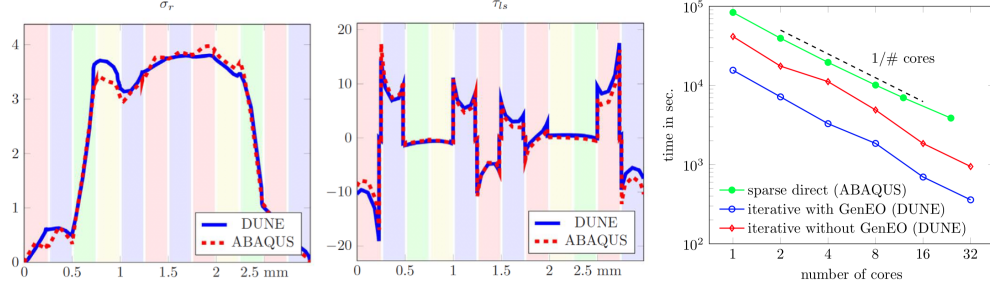
$$[\mp 45^\circ / 90^\circ / 0^\circ / \mp 45^\circ / \mp 45^\circ / 0^\circ / 90^\circ / \pm 45^\circ]. \quad (3.23)$$

Furthermore, we apply a resin treatment of 2mm to the free edges of the laminate as shown in Fig. 3-5 (left). The advantages of edge treatment have been shown in [150]. It reduces conservatism in the design of aircraft structures, as well as making the analyses more reliable by eliminating stress singularities at the free edges.

Away from the points of contact, a standard four-point bend test as detailed in ASTM D6415 [152] generates a pure moment on the corner. To simulate such a moment, all degrees of freedom at the boundary of one limb are clamped. At the other end, all nodes are tied with a multipoint constraint where a running moment of 96.8 Nmm/mm is applied. This is achieved by applying an offset load from the mid-plane of the laminate as a Neumann boundary condition, implemented with the user-defined function `evaluateNeumann()`.

The finite element mesh consists of  $56 \times 56$  columns of hexahedral 20-node serendipity elements (element **C3D20R** in ABAQUS, [4]) in its local  $l$  and  $s$  coordinates. In the  $r$  direction, each (fibrous and resin) layer is discretised into 6 elements, leading to an overall number of 432,768 elements. All of the geometry and mesh parameters are defined in `stackingSequences/example2.csv`. To ensure a sufficient resolution of the strong gradients of the solution at the free edges and at the material discontinuities, the mesh is graded along the width towards both free edges and in the radial direction towards each of the fibre-resin interfaces. Grading is defined by the ratio between largest and smallest elements in the mesh, called the bias ratio and chosen to be 400 between the center and the edge in the  $l$  direction and 10 between the layer centers and interfaces in the  $r$  direction. The specification of the geometry and of the mesh grading can be defined in the function `gridTransformation()`.

In Fig. 3-6 (left & middle), we compare the radius stress (denoted by  $\sigma_r$ ) and the through-thickness shear stress (denoted by  $\tau_{s\ell}$ ) recovered from **dune-composites** and ABAQUS. We see good agreement between the two codes. There are two small differences: ABAQUS uses reduced integration while our example uses full integration and the stresses are not recovered in an identical way from the displacements in the two codes. In Fig. 3-6 (right) we see the absolute cost and the parallel scalability of the sparse direct solver in ABAQUS and the iterative CG solver in **dune-composites** for a fixed total problem size, i.e. a strong scaling test. The red and blue curves show one-level and two level overlapping Schwarz methods respectively, both of which perform better than the sparse direct solver (green) in ABAQUS. However, both codes show optimal

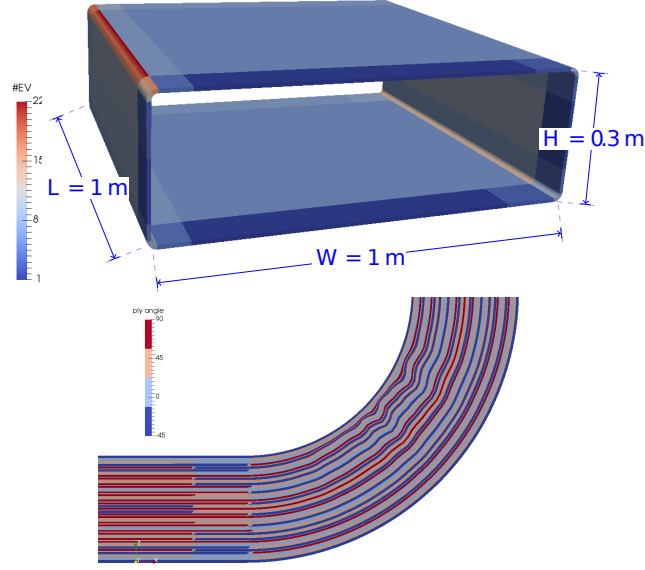


**Figure 3-6:** (Left & Middle) Stresses (in MPa) as functions of the distance  $r$  from the outer radius at the apex of the curve, at 2.156mm from the edge of the resin-edge-treated laminate (*dune-composites*, solid blue; ABAQUS, dotted red). The background colours indicate the stacking sequence:  $+45^\circ$ =red,  $-45^\circ$ =blue,  $90^\circ$ =green,  $0^\circ$ =yellow. (Right) Cost comparison between the sparse direct solver implemented in ABAQUS and the iterative preconditioned CG solver in *dune-composites*.

parallel scalability up to 32 cores. In *dune-composites* the problem is decomposed into 8 subdomains for 1 – 8 cores, distributed evenly to the available cores. On 16 and 32 cores, each core is passed exactly one subdomain, i.e., the number of subdomains is 16 and 32, respectively. The local problems on each subdomain are solved using the sparse direct solver **UMFPack** [146]. The simulations in ABAQUS are with a parallel sparse direct solver, based on a parallel multi-frontal method similar to [153]. ABAQUS’s iterative solver, which is based on CG preconditioned with ML [154] (another black-box AMG preconditioner), does not converge in a reasonable time for this problem due to the preconditioner’s aggregation strategy which is non optimal for composites. Therefore the computational gains observed here are really the difference between using a direct and a robust iterative solver. Importantly, we note that the parallel sparse direct solver, available in ABAQUS does not scale beyond 64 cores [4], making it unsuitable for problems much bigger than that considered here, and reinforcing the need for robust iterative solvers and therefore *dune-composites* as a package.

### 3.5.3 Example 3 : Large composite structure – parallel efficiency of *dune-composites* (up to 15,360 cores)

The industrially motivated problem described in this section is to assess the strength of a wingbox with a small localised wrinkle defect. Wrinkle defects, which can form during the manufacturing process [9, 155], occur at the layer scale. They lead to strong local stress concentrations [25, 156], causing premature failure. Naturally, good mesh resolution around the defect is required, leading to finite element calculations with very large number of degrees of freedom. We leave the engineering discussion of the results to a future engineering publication, using it instead to demonstrate both weak and



**Figure 3-7:** (Left) Geometry of the wingbox with dimensions; the colouring shows the number of eigenmodes used in *GenEO* in each of the subdomains of Setup 6 in Table 3.2. (Right) Close-up plot of the corner of the wingbox using *plotProperties()*, which shows the wrinkle and the inter-lacing of the different stacking sequences in the corner, cover and spar regions.

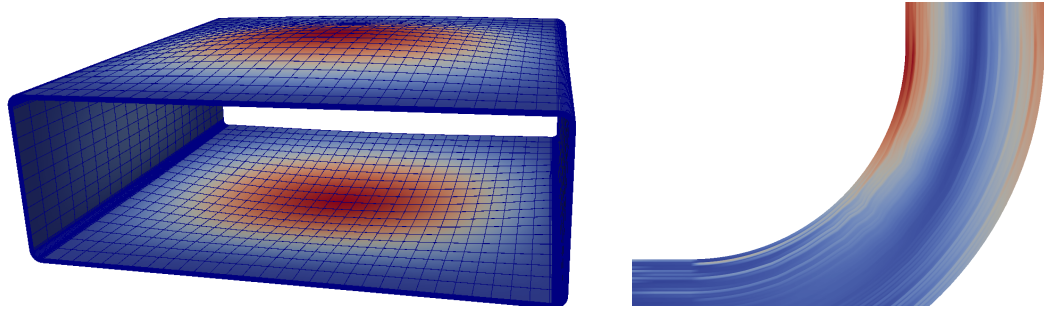
strong scalability of *dune-composites* up to 15,360 subdomains. The experiments in this section were performed using the UK national HPC cluster ARCHER, which has 4,920 Cray XC30 nodes with two 2.7 GHz, 12-core E5-2697 v2 CPUs each.

For these tests we model a single bay of a wingbox of width  $W = 1\text{ m}$ , height  $H = 300\text{ cm}$  and length  $L = 1\text{ m}$ , as shown by the schematics in Fig. 3-7 (left). The laminates were assumed to be of constant thickness, made up of 39 composite layers (as well as 38 interfaces) giving a total thickness of  $T = 9.93\text{ cm}$  with an internal radius of 15mm in the corners. As in a typical aerospace application, the stacking sequence differs in the covers (top and bottom), corners and in the spar (sides) with the following approximate percentage breakdowns of  $0^\circ$ ,  $\pm 45^\circ$  and  $90^\circ$ :

$$\begin{aligned} [50\%, 40\%, 10\%] & \quad (\text{covers}); \\ [20\%, 60\%, 20\%] & \quad (\text{corners}); \\ [15\%, 70\%, 15\%] & \quad (\text{spars}) \end{aligned} \tag{3.24}$$

We reiterate that this example serves to represent structural scale modelling. Therefore, sub-structural phenomena, such as stiffening of the upper and lower covers, are not modelled here. The specific layer-sequencing has been chosen, using a discrete optimiser, to ensure that each laminate is balanced, symmetric with no bend-twist





**Figure 3-8:** *FE solution for Example 3: (Left) Overall deformation of the wingbox with colours showing the magnitude of the displacements in cm. (Right) Camanho failure criterion (3.22) in a close-up of the corner containing the wrinkle defect.*

coupling, whilst maximising the number of continuous orientations around the wing box. Transitions between each of the stacking sequences are achieved over a relatively short segment of 5cm, and the chosen stacking sequence is in no way optimised for strength in these regions, as considered for example by Dillinger et al. [157]. In practice, this change of stacking sequence is easy for the user to specify using a `.csv` file specifying different `Regions` for each segment of the wingbox and providing the required different stacking sequence. The wingbox geometry is again achieved by specifying a `gridTransformation()`, which now becomes slightly more complex, in order to handle each of the different regions. To create the closed curve of this wingbox, periodic boundary conditions are imposed. In this application, we consider two forms of loading. Firstly, an internal pressure of 0.109MPa, arising from the fuel, is applied to the internal surface. Secondly, a thermal pre-stress induced by the manufacturing process is imposed, using the user-defined function `evaluateHeat()`.

We approximate the influence of the ribs that constrain the wingbox in the  $y$  direction, by clamping all degrees of freedom at one end, whilst tying all other degrees of freedom at the other end using a multipoint constraint. Elements to be included in the multipoint constraint are marked with the user-defined function `isMPC(FieldVec& x)`. A localised wrinkle defect is introduced into one of the corner radii, as shown in Fig. 3-7. The defect is introduced by adapting the function `gridTransformation()`. The wrinkle geometry is defined by a random field, parameterised by a Karhunen-Lo  ve expansion. The actual parametrisation of the wrinkle is chosen to match an observed defect in a CT-Scan of a real corner section. Further details of this methodology are provided in Section 4.2.

We firstly carry out a weak scaling experiment, increasing the problem size proportionally to the number of cores used. For iterative solvers that scale optimally with respect to problem size and with respect to the number of cores, the computational

---

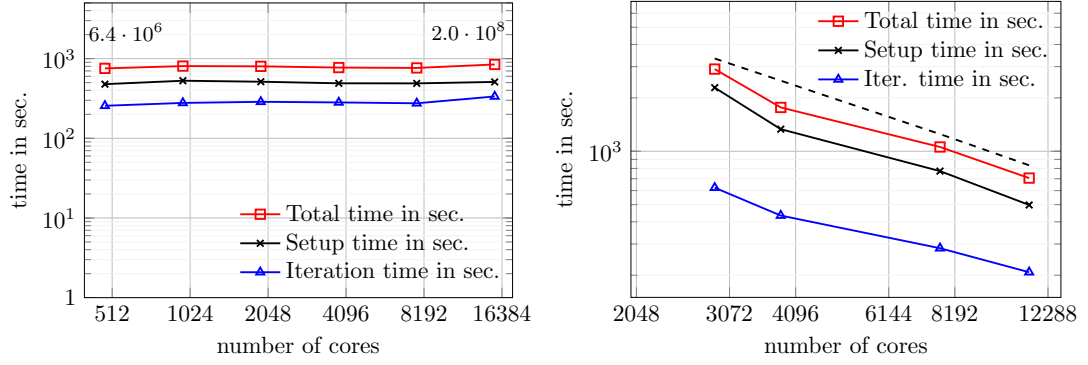
Setup	$N_{cores}$	Spar	Cover	$R_d$	$R_{nd}$	Length	DOF	iter.	$\kappa(A_H)$	$\dim V_H$	Time (sec)
1	480	34	14	40	20	20	$6.4 \cdot 10^6$	156	445	5025	734
2	960	34	14	40	20	40	$1.3 \cdot 10^7$	154	421	7840	806
3	1920	68	28	80	40	40	$2.6 \cdot 10^7$	152	322	18752	800
4	3840	68	28	80	40	80	$5.1 \cdot 10^7$	144	287	29444	772
5	7680	216	64	80	40	80	$1.0 \cdot 10^8$	132	303	50930	764
6	15360	216	64	80	40	160	$2.0 \cdot 10^8$	102	245	94527	845

---

**Table 3.2:** *Details of the six setups and results used in the weak scaling test. In all of the tests, we used two layers of 20-node serendipity elements per fibrous layer and only one layer of elements in each of the interface layers. The number of elements per core was fixed at 2808.*

time should remain constant. To scale the problem size as the number of cores  $N_{cores}$  grows, we refine the mesh, doubling the number of elements as we double the number of cores. The number of elements for each setup are detailed in Table 3.2, separately listing the number of elements across the spar and the cover, around the corners and along the length of the wingbox. The defective corner, denoted  $R_d$ , contains twice as many elements as the other three corners, denoted by  $R_{nd}$ . Table 3.2 also details the resulting number of degrees of freedom, iteration numbers for the preconditioned CG, an estimate of the condition number of the coarse matrix  $\kappa(A_H)$ , the dimension of the coarse space  $\dim V_H$ , as well as the total run time. Fig. 3-9 (left) shows that the weak scaling of the iterative CG solver in `dune-composites` with `GenEO` preconditioner is indeed almost optimal up to at least 15,360 cores (the limiting capacity available on ARCHER for our experiments). We also include a more detailed subdivision of the computational time into Setup Time (for the assembly of the FE stiffness matrix and for the construction of the `GenEO` coarse space) and Iteration Time (for the preconditioned CG iteration). Both scale almost optimally. This test demonstrates the capability of increasing the size of the tests at a nearly constant run time and thus, to solve a problem with 200 million degrees of freedom in just over 14 minutes.

Next we carry out a small strong scaling experiment. The mesh is that of Setup 5 in Table 3.2 and the results of the strong scaling test are given in Figure 3-9 (right) and in Table 3.3. We see that the iterative CG solver in `dune-composites` with `GenEO` preconditioner scales almost optimally to at least 11320 cores, with the time taken approximately halving as the number of cores is doubled. Again, both the Setup and the Iteration scale optimally.



**Figure 3-9:** Parallel performance of *dune-composites* on ARCHER: (Left) A weak scaling test, as summarised in Table 3.2. (Right) A strong scaling test using Setup 5 in Table 3.2, with the dashed line showing perfect scaling, as summarised in Table 3.3.

$N_{cores}$	EpC	$\dim(V_H)$	num it.	$T_{it}$	$T_{setup}$	$T_{total}$	TCT (days)
2880	3132	18843	167	623s	2283s	2906s	96.9
3840	2340	26333	153	434s	1332s	1766s	78.5
7680	2008	52622	132	284s	773s	1057s	94.0
11320	1392	78233	162	208s	498s	706s	92.5

**Table 3.3:** Strong scaling test with Setup 5 in Table 3.2, demonstrating near optimal strong scaling up to at least 11,320 cores. Note: EpC is Elements per Core and TCT is Total Core Time i.e.  $TCT = \frac{T_{total} * N_{cores}}{60 \times 60 \times 24}$ .

### 3.5.4 Subsurface flow applications: Strong scaling for the SPE10 benchmark

In this section we demonstrate the application of the **GenEO** solver to an elliptic partial differential problem outside of elasticity equations. We therefore test **GenEO** on Darcy’s law for a highly heterogeneous media. A challenging test case in computational geosciences is the SPE10 benchmark [158]. This problem features high contrast, discontinuous coefficients making it a challenge for most iterative solvers [159].

We consider the SPE10 domain  $\Omega := [0, 1200] \times [0, 2200] \times [0, 170]$  (feet), divided into a tensor product grid  $\mathcal{T}_h$  with  $60 \times 220 \times 85 = 1.122 \times 10^6$  cells. The domain  $\Omega$  has boundary  $\partial\Omega = \Gamma_D \cup \Gamma_N$ , where we define  $\Gamma_D := \{\mathbf{x} \in \partial\Omega : z = 0\}$  as the Dirichlet part of the boundary and  $\mathbf{n} \in \mathbb{R}^3$  as the outward normal to  $\partial\Omega$ . We calculate the steady-state fluid pressure  $u(\mathbf{x}) \in \Omega$  which obey’s Darcy’s law. This is given by the linear, scalar elliptic partial differential equations

$$-\nabla \cdot (\mathbf{K}(\mathbf{x})\nabla u) = f, \quad \forall \mathbf{x} \in \Omega \quad (3.25)$$

---

subject to boundary conditions

$$u(\mathbf{x}) = 0 \quad \text{on} \quad \Gamma_D \quad \text{and} \quad -\mathbf{K}(\mathbf{x})\nabla u \cdot \mathbf{n} = 0 \quad \text{on} \quad \Gamma_N = \partial\Omega \setminus \Gamma_D \quad (3.26)$$

and uniform source term  $f \equiv 1$ . The SPE dataset gives a spatially varying permeability tensor

$$\mathbf{K}(\mathbf{x}) = \begin{bmatrix} \mathbf{K}_x(\mathbf{x}) & 0 & 0 \\ 0 & \mathbf{K}_y(\mathbf{x}) & 0 \\ 0 & 0 & \mathbf{K}_z(\mathbf{x}) \end{bmatrix} \quad \forall \mathbf{x} \in \Omega \quad (3.27)$$

Figure 3-10 (top) shows the permeability field, it is constant in each cell, but varies strongly over the domain. The parameters  $\mathbf{K}_x$  and  $\mathbf{K}_y$  vary from  $6.64 \times 10^{-4}$  to  $2.0 \times 10^4$  and the parameter  $\mathbf{K}_z$  varies between  $6.65 \times 10^{-8}$  and  $6.0 \times 10^3$ . We define the function space for the pressure  $u$  as  $V := \{v \in H^1(\Omega) : v(\mathbf{x}) = 0, \mathbf{x} \in \Gamma_D\}$ , and from the finite element space  $V_h \subset V$  from the set of piecewise linear functions on  $\mathcal{T}_h$ . The weak, finite element equations for Eq. (3.25) are then; find  $u_h \in V_h$  such that

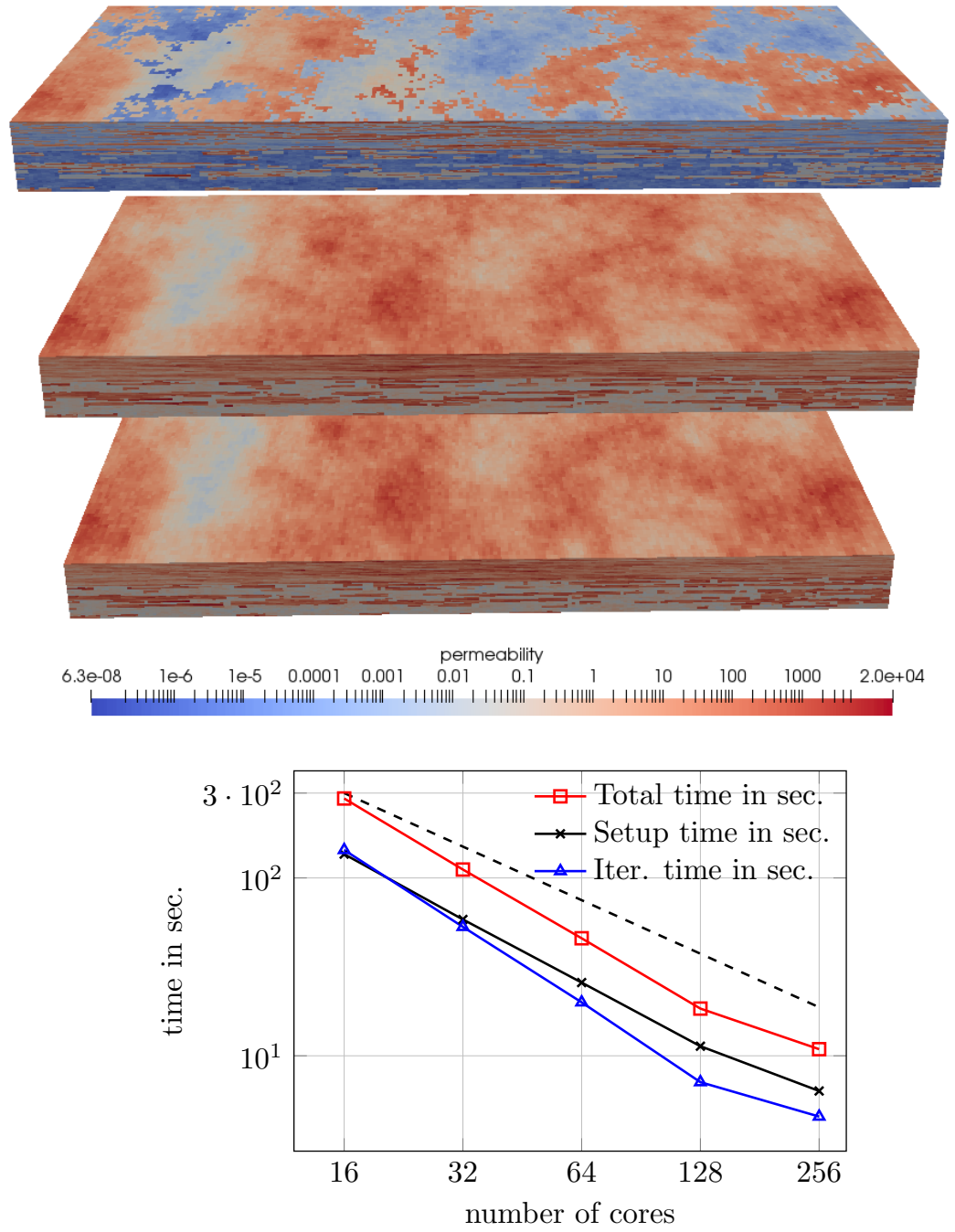
$$\int_{\Omega} \mathbf{K}(\mathbf{x})\nabla v_h dx + \int_{\Omega} f v_h dx = 0 \quad \forall v_h \in V_h \quad (3.28)$$

By defining  $u_h = \sum_{i=1}^n u^{(i)} \phi_i(\mathbf{x})$ , again we obtain the sparse system of equations

$$\mathbf{A} \tilde{\mathbf{u}}, \quad \text{where} \quad \tilde{\mathbf{u}} = [u^{(1)}, u^{(2)}, \dots, u^{(n)}]^T \quad (3.29)$$

is a vector of pressures at each cell vertex assembled element-wise from Eq. (3.28) using standard Gaussian intergration.

In Fig. 3-10 (bottom) and Table 3.4, we show a small strong scaling experiment performed with this challenging setup. The parameter constrast for this benchmark is on the order of  $10^{11}$ . Nevertheless, we show that the iterative CG solver in `dune-composites` with the `GenEO` preconditioner scales almost optimally to at least 256 cores. At 512 cores with only around 2000 elements per core, the strong scaling test begins to breakdown. Due to the layered structure of the material parameters our domain decomposition is two dimensional. Each subdomain extends through the full length of in the z-direction. We used a minimal overlap of only one element. Table 3.4 also details the number of cores, size of the coarse space  $\dim(V_h)$ , iteration numbers for preconditioned CG, as well as the time spent in CG iterations, setup time and total run time.



**Figure 3-10:** (Top) Logarithm of the permeability field  $\mathbf{K}$  for the SPE10 benchmark, from bottom to top:  $K_x$ ,  $K_y$  and  $K_z$ . (Bottom) A strong scaling test using the SPE10 dataset, with the dashed line showing perfect scaling.

---

$N_{cores}$	$\dim(V_H)$	it.	$T_{it}$	$T_{setup}$	$T_{total}$
16	149	167	136.11	143.621	279.721
32	225	203	58.42	53.065	111.485
64	379	206	25.81	19.982	45.787
128	527	224	11.32	7.107	18.427
256	930	232	6.34	4.552	10.892
512	1737	234	5.18	3.795	8.975

---

**Table 3.4:** *A strong scaling test using the SPE10 dataset.*

### 3.6 Concluding remarks

In this chapter, we described the new high performance package **dune-composites**, designed to solve massive finite element problems for anisotropic linear elasticity equations. The chapter provides both the mathematical foundations of the methods, their implementation within a state-of-the-art software platform on modern distributed memory computer architectures, as well as details of how to set up a problem and carry out an analysis, illustrated via a series of increasingly complex examples. In addition, we demonstrate the scalability of the new solver on over 15,000 cores on the UK national supercomputer ARCHER, solving industrially motivated problems with over 200 million degrees of freedom within minutes. This scale of computations brings composites problems that would otherwise be unthinkable into the feasible range.

The disadvantage of **dune-composites** as a package over commercial counterparts is the (currently) limited functionality in considering more general problems; this includes complex geometries, unstructured grids and nonlinear problems. This is the first release of **dune-composites**, and therefore the functionality is naturally still limited, but it will increase over time, driven by the industrial questions we seek to solve as a community of developers.



---

## CHAPTER 4

---

# A BAYESIAN FRAMEWORK FOR ASSESSING THE STRENGTH DISTRIBUTION OF COMPOSITE STRUCTURES WITH RANDOM DEFECTS

### 4.1 Introduction

In Chapter 2, we identified the need for more comprehensive uncertainty quantification. Pertaining to wrinkle defects, we identified the need for better parameterization and non-deterministic analyses in Section 2.1 where we also review effects of defects, their formation and detection. In this chapter, we develop a framework to compute the statistics of the strength penalty imposed by defects in composites. We develop a stochastic methodology to explore the distribution of strength of defective components by integrating finite element modelling of defects with observed measurement data about their size, location and morphology. Motivated by industry, we focus our study on out-of-plane defects, yet note that the general stochastic framework is applicable across a broad range of defect types, measurement data and modelling choices. Here, we use these Bayesian tools to integrate high-fidelity finite element modelling capabilities of defective composites with NDT measurements of wrinkles. The reader is referred



---

to Section 2.2 for a review of uncertainty quantification using Bayesian tools. The main goal is to determine the (inaccessible) distribution of wrinkle parameters based on measured data. Samples from the inaccessible distribution of wrinkles are drawn via MCMC. In our case these wrinkle samples can be embedded into a high-fidelity finite element model and solved using `dune-composites` [3] to predict the strength of each sample in a Monte Carlo step. The output is the distribution of component strength given measurements of observed defects.

We begin by introducing multi-disciplinary concepts that constitute this framework. The general description of the framework is provided starting with wrinkle parameterization for a generic basis, followed by posterior sampling and finally, Monte Carlo simulations to determine expected strength. An application of this framework is demonstrated through an industrial case study where we first describe the model problem, followed by the implementation of the method. Simulation results are presented alongside user inputs. In light of the findings and supporting literature [43, 45, 46] for slope-failure dependency, an engineer’s model is proposed to predict failure based on maximum misalignment.

## 4.2 Bayesian approach to construct defect distributions from measured data

In this section we describe a Bayesian approach to construct a distribution of wrinkle defects from observed data, and how this distribution links to stochastic Monte Carlo simulations with a finite element model to predict the distribution of component strength. We have intentionally left the description general to show that the methodology works for a broad definition of wrinkle defects. In fact, provided an adequate basis is chosen, a variety of defects can be modelled within this framework. Here, a specific industrially motivated case study is considered in a later section.

### 4.2.1 Parameterizing a wrinkle defect

An important step, and one that requires a modelling choice, is to define the method by which a wrinkle defect in a composite part is parameterized. In this contribution, and others, a wrinkle defect is defined by a deformation field  $W : \Omega \rightarrow \mathbb{R}^3$  mapping a composite component from a pristine state (occupying  $\Omega \subset \mathbb{R}^3$ ) to the defected state (occupying  $W(\Omega) \subset \mathbb{R}^3$ ), Fig. 4-1. To make our approach amenable to analysis we

---

define this map by the finite dimensional representation

$$W(\mathbf{x}, \boldsymbol{\xi}) = \sum_{i=1}^N a_i \psi_i(\mathbf{x}, \mathbf{b}), \quad \text{for } \mathbf{x} \in \Omega \subset \mathbb{R}^3 \quad \text{and} \quad \boldsymbol{\xi} = [\mathbf{a}, \mathbf{b}]^T \in \mathbb{R}^{N_w}. \quad (4.1)$$

where the set  $\{\psi_i(\mathbf{x})\}$  defines the orthonormal basis over which wrinkles are defined, and  $\boldsymbol{\xi}$  is a vector of coefficients  $N_w$  in length parameterizing the wrinkle. Here, we leave this choice open to show that the methodology presented is largely independent of parameterization of wrinkle since different choices have been made in the literature. Having said this, the choice of basis  $\psi_i$  is important as it constrains the representation of wrinkles. Therefore, it should be left as general as possible, with two important considerations

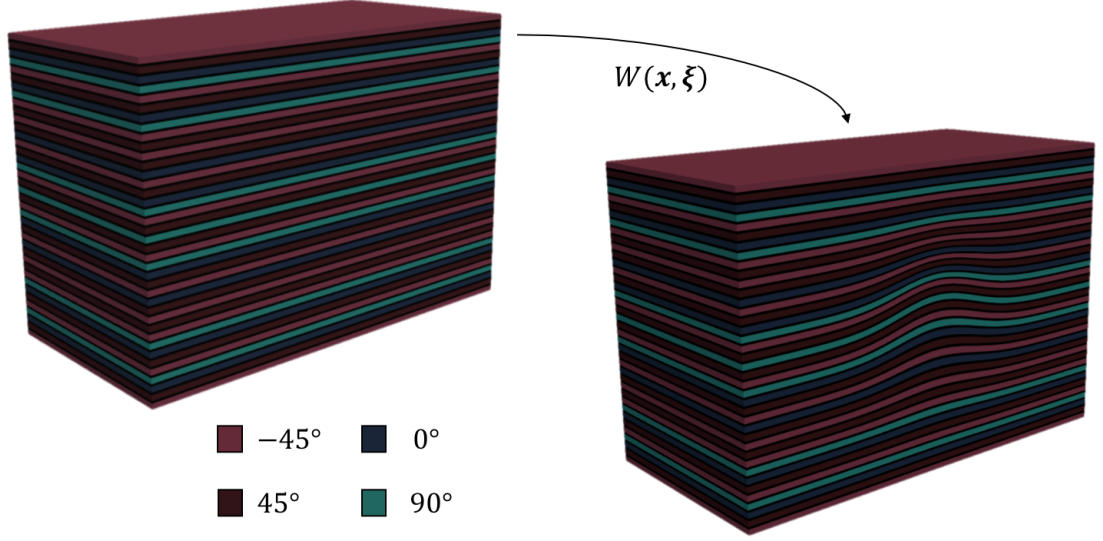
- The deformation induced by  $W(\mathbf{x}, \boldsymbol{\xi})$  should not self-intersect. This is equivalent to the constraint that the  $\det \mathcal{J}(\mathbf{x}, \boldsymbol{\xi}) > 0$ , for all  $\mathbf{x} \in \Omega$ , where  $\mathcal{J}(\mathbf{x}, \boldsymbol{\xi})$  is the Jacobian of the deformation map  $W(\mathbf{x}, \boldsymbol{\xi})$ . At this stage it is sufficient to choose  $\psi_i$  not self-intersecting, and impose the constraint  $\det \mathcal{J}(\mathbf{x}, \boldsymbol{\xi}) > 0$  during the posterior sampling (see below).
- Since the data for which we tune our wrinkle distribution is estimated from approximations of the misalignment of plies, the basis functions  $\psi_i(\mathbf{x}, \mathbf{b})$  should have well-defined first derivatives in the  $x_1$  and  $x_2$  directions. Moreover, the misalignment is computed as follows

$$\tan \phi_j(\mathbf{x}, \boldsymbol{\xi}) = \sum_{i=1}^{N_w} a_i \frac{d\psi_i(\mathbf{x}, \mathbf{b})}{dx_j} \quad \text{for } j = 1 \text{ and } 2. \quad (4.2)$$

#### 4.2.2 Posterior Sampling using a Metropolis-Hastings algorithm

Let the vector valued random variable  $\boldsymbol{\xi} \in \mathbb{X} \subset \mathbb{R}^{N_w}$  denote the  $N_w$ -dimensional coefficient vector representing a random wrinkle profile. We will assume this has the general form defined by (4.1). Let  $\mathcal{D}_{obs} := \{\mathbf{d}_{obs}^{(1)}, \dots, \mathbf{d}_{obs}^{(n)}\}$  denote the set of data measured from  $n$  observed independent wrinkles, each characterized by  $\mathbf{d}_{obs}^{(i)} = \{\phi_1^{(i)}, \phi_2^{(i)}, \dots, \phi_{N_\phi}^{(i)}\} \in \mathbb{D} \subset \mathbb{R}^{N_\phi}$ .

The *Forward Model*  $F(\boldsymbol{\xi}) : \mathbb{X} \rightarrow \mathbb{D}$  maps a set of wrinkle coefficients  $\boldsymbol{\xi} \in \mathbb{X}$  to the observable model output  $\mathbf{d}_{obs}^{(i)} \in \mathbb{D}$ . In this chapter the observable data is the misalignment field of the wrinkle profile in the  $x_1x_3$  plane, and therefore we define the



**Figure 4-1:** Illustration of Eq. (4.1) showing the transformation from pristine to defective state for a 39 ply composite with a representative stacking sequence

Forward model as

$$\phi_j = \tan^{-1} \left( \sum_{i=1}^{N_w} \xi^{(i)} \frac{d\psi_i(\mathbf{x}_j)}{dx_1} \right) \quad \text{at measurement points } \mathbf{x}_j \quad \text{for } j = 1 \dots N_\phi. \quad (4.3)$$

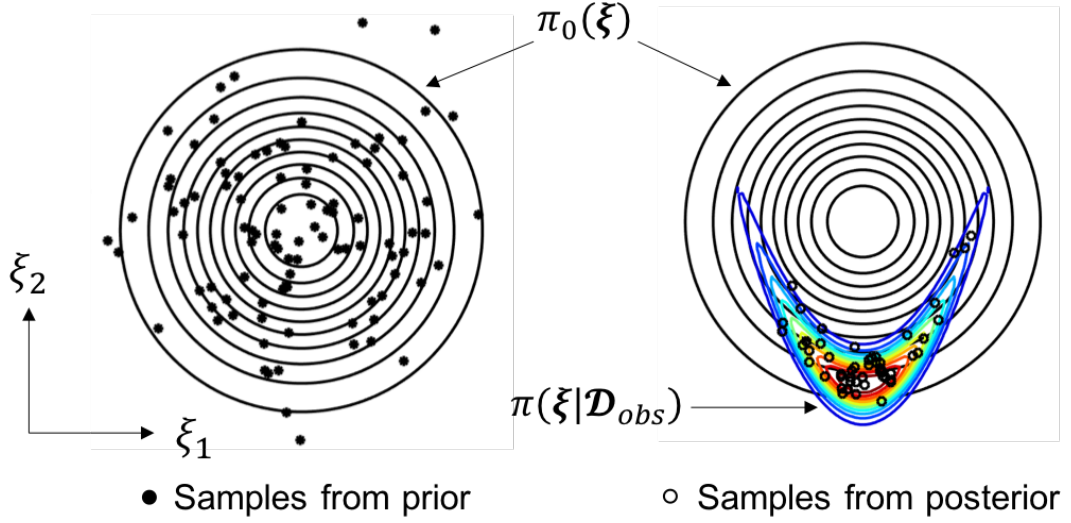
In a Bayesian setting the first task is to construct the prior model and the likelihood function as probability distributions. The prior density is a stochastic model representing knowledge of the unknown  $\boldsymbol{\xi}$  before Bayesian inversion on the data, denoted by the distribution  $\pi_0(\boldsymbol{\xi})$ . The likelihood function specifies the probability density of the observation  $\mathcal{D}_{obs}$  for a given set of parameters  $\boldsymbol{\xi}$ , denoted by  $\mathcal{L}(\mathcal{D}_{obs}|\boldsymbol{\xi})$ . We assume that the data and the model parameters have the following stochastic relationship

$$\mathbf{d}_{obs} = F(\boldsymbol{\xi}) + \boldsymbol{\epsilon} \quad (4.4)$$

where the random vector  $\boldsymbol{\epsilon} \in \mathbb{R}^{N_\phi}$  captures the measurement noise and other uncertainties in the observation-model relationship. Without additional knowledge of the measurement errors,  $\boldsymbol{\epsilon}$  is modelled as a zero mean Gaussian  $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \Sigma_\epsilon)$ , for covariance  $\Sigma_\epsilon$ .

Let the misfit function for the observation  $\mathbf{d}_{obs}^{(i)}$  be defined in the standard way, so that

$$\delta_i(\boldsymbol{\xi}) = \frac{1}{2} \left\| \Sigma_\epsilon^{-\frac{1}{2}} \left( F(\boldsymbol{\xi}) - \mathbf{d}_{obs}^{(i)} \right) \right\|_2 \quad (4.5)$$



**Figure 4-2:** Representation of Bayesian approach in a simplified 2D parameter space for a known likelihood function (a variant of the Rosenbrock function). (Left) Sampling from isotropic Gaussian prior  $\pi_0(\xi)$ . (Right) MCMC sampling from posterior  $\pi(\xi|\mathcal{D}_{obs})$ .

Whilst the misfit over the complete data set  $\mathcal{D}_{obs}$  is defined by

$$\Delta(\xi) = \min_{i=1, \dots, N_\phi} \delta_i(\xi), \quad (4.6)$$

which can be interpreted as mis-fit compared with the closest observed data point. The likelihood function  $\mathcal{L}(\mathcal{D}_{obs}|\xi)$  is proportional to  $\exp(-\Delta(\xi))$ , and by Bayes' formula, the posterior probability density is

$$\pi(\xi|\mathcal{D}_{obs}) = \frac{1}{Z} \exp(-\Delta(\xi)) \pi_0(\xi), \quad (4.7)$$

where  $Z$  is a normalizing constant (which there is no need to compute).

The posterior distribution (4.7) can be sampled using MCMC methods such as the standard random walk algorithm [6]. We now provide a brief review of the Metropolis-Hastings algorithm used in this contribution. The first step is to define our prior distribution  $\pi_0(\xi)$  and pull a random sample as the starting point of our Markov Chain, say  $\xi^{(0)}$ . Subsequent points on the Markov Chain  $\xi^k$  are generated by making a proposal  $\xi'$  defined by

$$\xi' = \sqrt{(1 - \beta^2)} \xi^{k-1} + \beta \omega \quad (4.8)$$

This is a preconditioned Crank-Nicholson (PCN) proposal with  $\beta \in \mathbb{R}$ , a tuning parameter designed to enhance the efficiency of the standard Markov chain algorithm [6]. The value  $\beta$  controls the step size of a proposal. In Eq. (4.8),  $\omega \in \mathbb{R}^{N_w}$  such that

---

$\omega_j \sim \mathcal{N}(0, \sigma_{PCN}^2)$ , are vectors of normally distributed random variables with standard deviation  $\sigma_{PCN}$ . The proposal  $\xi'$  is accepted for the next sample in the Markov chain  $\xi^{(k)}$  with the following probability

$$\alpha(\xi', \xi^{k-1}) = \min \left\{ 1, \frac{\mathcal{L}(\mathcal{D}_{obs}|\xi')}{\mathcal{L}(\mathcal{D}_{obs}|\xi^{k-1})} \right\} \quad \text{and} \quad \det \mathcal{J}(\mathbf{x}, \xi') > 0 \quad (4.9)$$

otherwise  $\xi^k = \xi^{k-1}$ . This process generates a series of samples which have the conditional probability distribution  $\pi(\xi|\mathcal{D}_{obs})$  which is the distribution of coefficients given the set of observations  $\mathcal{D}_{obs}$ .

Since we will use the  $\xi^{(k)}$  in Monte Carlo simulations we require  $N_{MC}$  independent samples. Samples close to one another in a Markov chain are strongly correlated. By estimating the integrated autocorrelation time for each component of Markov Chain  $\Lambda_i$  (see details in [160, Ch. 5.8]) we can approximate a subsampling interval  $\Lambda = \max(\Lambda_i)$  for which the samples are independent. Therefore in calculations which follow sampling only occurs after a *burn-in* period of  $b \gg \Lambda$  samples, to remove the influence of initial start of the chain  $\xi^{(0)}$  on the distribution of  $\xi$ . Then samples are taken every  $\Lambda$ , generating the set of  $N_{MC}$  independent random wrinkles from  $\pi(\xi|\mathcal{D}_{obs})$ ,

$$\Xi = \{\xi^{(b+\Lambda)}, \xi^{(b+2\Lambda)}, \dots, \xi^{(b+N_{MC}\Lambda)}\}. \quad (4.10)$$

and  $\Lambda$  is inversely related to  $\beta$ . However, the relationship between  $\beta$  and acceptance ratio, another diagnostic property of MCMC, is more complex. Acceptance ratio is the ratio of the number of accepted proposals to the total number of a proposals made. Intuition suggests that a higher proposal density will lead to a higher percentage of wiser moves resulting in a greater acceptance ratio. A natural question then arises - what is the optimal proposal density or acceptance ratio? Reference [82] defines a metric of efficiency (Langevin diffusion) in terms of acceptance ratio. This metric effectively quantifies the *diffusion rate* of a chain through some unknown posterior distribution. Then the optimal acceptance ratio is one that maximizes the diffusion rate. The mathematical proof suggests that the asymptotically optimal acceptance ratio is approximately 0.25 however, in practice a ratio lower than 0.3 may be unachievable [82]. The value of  $\beta$  can be tuned to achieve this acceptance ratio.

The convergence of the chain to the posterior distribution (4.7), can be monitored by running multiple independent parallel chains, and observing the convergence of  $\mathbb{E}[\Xi]$  (and perhaps higher moments) between all chains. Largely varying means of each chain would indicate the chains have yet to converge to a stationary distribution, and the burn-in period should be extended. This is particularly important in our case since

---

with multiple independent observations  $\mathcal{D}_{obs}$ , the likelihood defined by (4.6) represents a multi-modal posterior with maxima at each data point.

### 4.2.3 Monte Carlo Simulations

Having generated a distribution of wrinkle profiles from observed data we are interested in computing the strength distribution of the defected components, by propagating these defects through a model and observing the distribution of component failure load. For this we introduce a finite element model  $Q_M(\boldsymbol{\xi}) : \mathbb{X} \rightarrow \mathbb{R}$  which maps a given wrinkle profile to an engineering quantity of interest, e.g. the expected load or moment at failure, or the probability of the failure occurring below a prescribed loading condition. The details of the particular finite element model and setup used in this contribution are provided in Section 4.3.4. The subscript  $M$  indicates the number of degrees of freedom in that model, so that as  $M \rightarrow \infty$  (under uniform mesh refinement) the expected value converges for some (inaccessible) random variable  $Q : \mathbb{X} \rightarrow \mathbb{R}$ , i.e.  $\mathbb{E}[Q_M] \rightarrow \mathbb{E}[Q]$ . We therefore seek to estimate

$$\mathbb{E}[Q] = \int_{\mathbb{X}} Q(\boldsymbol{\xi}) \pi(\boldsymbol{\xi} | \mathcal{D}_{obs}) d\boldsymbol{\xi}. \quad (4.11)$$

This can be estimated by  $N_{MC}$  posterior samples  $\boldsymbol{\xi}^{(i)} \sim \pi(\boldsymbol{\xi} | \mathcal{D}_{obs})$  and the Monte Carlo estimate

$$\hat{Q}_M = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} Q_M(\boldsymbol{\xi}^{(i)}) \quad (4.12)$$

which is a biased estimator with the mean square error

$$\varepsilon(\hat{Q}_M)^2 = \mathbb{E}[Q - Q_M]^2 + \frac{\mathbb{V}(\hat{Q}_M)}{N_{MC}}, \quad \text{such that} \quad \mathbb{V}[\hat{Q}] \approx \frac{1}{N_{MC} - 1} \sum_{i=1}^{N_{MC}} (Q_M(\boldsymbol{\xi}^{(i)}) - \hat{Q}_M)^2. \quad (4.13)$$

The first term in this expression represents the bias error in the model, arising from the quantity of interest  $Q$  being approximated by a finite element calculation on a finite-dimensional grid. This error can be estimated from mesh analysis over a number of samples, as given in Section 4.3.4. The second term is the sampling error, arising from approximating  $\mathbb{E}[Q_M]$  with only a finite number of samples. Care should be taken to balance these two errors to avoid unnecessary and expensive forward FE runs.

---

### 4.3 Industrially Motivated Case Study

A case study based on industrial data is conducted to demonstrate the methodology described in Section 4.2 and to introduce some bespoke but essential peripheral developments required to build theoretical models ( $F(\boldsymbol{\xi}) : \mathbb{X} \rightarrow \mathbb{D}$  and  $Q : \mathbb{X} \rightarrow \mathbb{R}$ ) from empirical data (B-scans). These developments are explained here in the context of a model problem derived from an aircraft wing. The following section provides details and assumptions made in the model by briefly introducing the particular non-destructive testing (NDT) technique used to access internally contained wrinkles invisible from the outside. Section 4.3.2 then provides details on the Multiple Field Image Analysis (MFIA) algorithm that extracts alignment information from NDT results (B-scans). This is a necessary step prior to wrinkle parameterization as it constitutes the left hand side of Eq. (4.2). Next, we define an appropriate basis from which the right hand side of Eq. (4.2) is derived to compute coefficients of best fit. Once parameterization is complete, all wrinkles considered here can be represented by some linear combination of the basis. Thus, in the posterior - the distribution of coefficients within the parameterized space  $\mathbb{R}^{N_w}$  - one wrinkle only differs from another in the content of its coefficient vector,  $\mathbf{a}$ . It is important to understand that the method described in Section 4.2 is directly applied within this parameterized space. Note, any mathematical operation applied to a *wrinkle* hereafter, should be interpreted as a treatment of its corresponding coefficient vector.

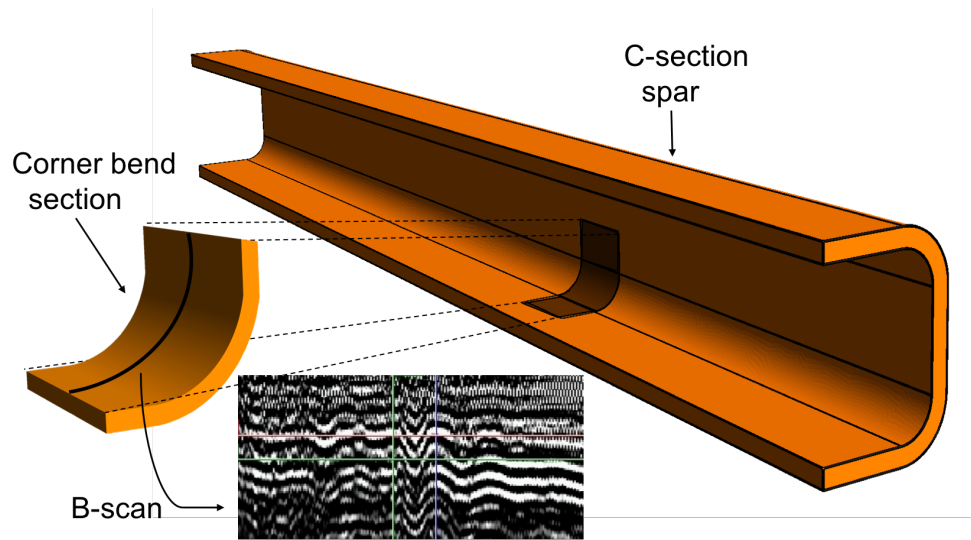
Now we wish to draw some conclusions about the posterior and model its evolution into a strength distribution in  $\mathbb{R}$ . An infinity of samples would be required to find the true distribution of wrinkles but we can produce useful results by pulling an appropriate number of independent and identically distributed samples or *iids* from the posterior. These *iids* or Monte Carlo samples are passed through an FE model  $Q : \mathbb{X} \rightarrow \mathbb{R}$  that outputs a scalar strength. Details of the FE simulation are provided in Section 4.3.4. Continuous distributions can be approximated from the *iids* within some confidence bounds. We attempt to interpret the results thus generated in terms of Weibull statistics to understand the cumulative effects of misalignments. However, an engineering result of crucial importance is presented in light of the findings up to this point. It is a parameterized exponential relationship between the first derivative of a wrinkle,  $W'(\mathbf{x}, \boldsymbol{\xi})$  and its failure moment. It offers a major time advantage by replacing lengthy FE calculations with an analytical formula.

---

#### 4.3.1 Model Problem and its industrial application

Figure 4-3 (right) illustrates a typical aircraft development program followed by industry. As more exotic materials and/or technology are introduced into the aerospace industry, development costs continually increase. In the interest of preserving the health of the aviation economy, a new initiative encouraging modelling alongside physical testing is gaining traction. To achieve similar levels of robustness, the modelling track follows a conceptually similar development pyramid whereby the number of simulations conducted at the coupon level are much greater than components or systems. Motivated by this philosophy, one corner of a C-section composite (CFRP) wing spar is considered for the model problem. Figure 4-3 shows the corner bend coupon as a building block of a element-level part (spar) to clarify that this study explores the coupon level exclusively. Additional detail about the model problem is provided in Fig. 4-6.

Upper and lower wing covers, bound together by the fore and aft spars form the fuel tanks in an aircraft. For a multitude of reasons, fuel must be stored under pressure which exerts an opening moment along the inner radius of the spar. During manufacture these regions (highlighted in Fig. 4-3) are susceptible to wrinkle formation rendering them of critical importance for failure initiation. This case study focuses on characterization of such defects to simulate their effects on part strength using a corner bend sample representative of the region shown in Fig. 4-3.



**Figure 4-3:** Schematic of a wing spar highlighting the region of showing a B-scan at a defect location.

Visualizing and measuring the wrinkles that we intend to model can be particularly challenging since their parent components may be inconveniently large ( $> 10\text{m}$ ). To



---

overcome problems like immersion in ultrasonic imaging, a *phased array* is used for scanning. There are two major advantages of a phased array: 1) it eliminates motion of probes by using multiple sensors arranged so that they can be fired individually to allow beam steering and wavefront manipulation (focus) [41, 161] and, 2) scanning can be conducted in-situ. Of greater importance perhaps, are the two key limitations since, in this case, they simplify our problem.

Firstly, the resultant image produced is a slice through thickness rather than a volume (see Fig. 4-3 (left)) meaning that we only have two dimensional information about a wrinkle. Due to the geometry of the spar, wrinkles form in a way that its span in  $x_2$  is orders of magnitude greater than perturbations in  $x_1$  and  $x_3$ . Therefore, within the vicinity of the B-scan, we can safely assume the wrinkle to be prismatic in  $x_2$ .

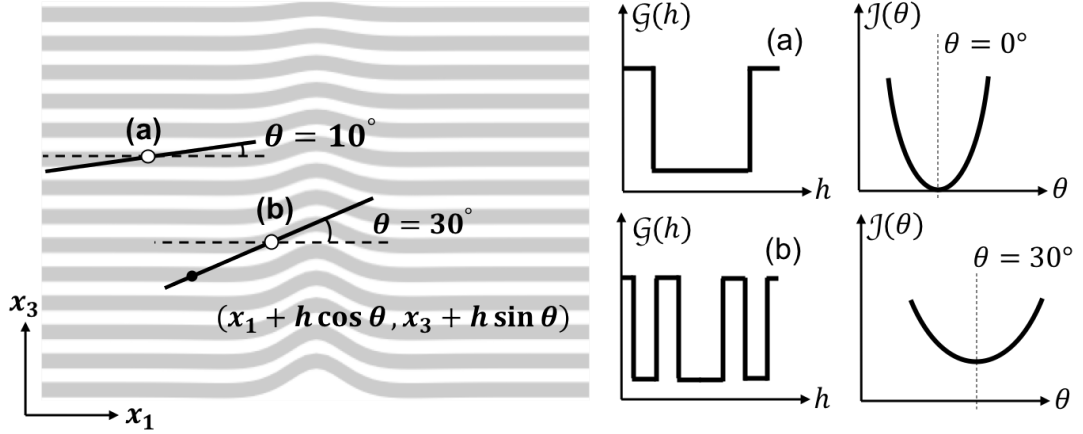
Secondly, the ultrasonic beams are focused at a particular depth such that scans contain a corresponding high resolution region. Ply boundaries begin to fade into the surroundings further from the focused band (see lower half of B-scan in Fig. 4-3). Therefore, at a sampling location outside the focused region, the existence of a global minimum in gray scale variance is not guaranteed. Since the wrinkle shows no clear signs of decay in  $x_3$  within the subregion, we discard the  $x_3$  coordinate of the sampling points. We have thus reduced the number of dependent variables of the alignment map to one.

### 4.3.2 Extracting wrinkle data from B-Scans using Multiple Field Image Analysis (MFIA)

A variety of image processing tools for investigating alignment exist, see for example the review by Smith *et al.* [162] and other contributions [161, 163]. In this contribution we use Multiple Field Image Analysis (MFIA) algorithm introduced by Creighton *et al.* [164] to estimate the misalignment of a wrinkled ply at a given position in the B-Scan image. We briefly review the method and describe some adaptations made to handle low resolution B-Scan images and the computational efficiency of the original approach [164].

Multiple Field Image Analysis (MFIA) [164] uses a pixelated gray-scale image. At a given point  $\mathbf{x} = (x_1, x_3)$ , a *trial fibre* is introduced. This is an array of pixels of length  $H$ , centered about  $\mathbf{x}$ , and orientated at an angle  $\theta$  to the  $x_1$  or  $x_2$  axis. At each sampling point, the algorithm finds the orientation  $\theta$  of the trial fibre which minimizes variance in gray scale along its length, i.e. the misalignment at point is the defined by

$$\phi = \arg \min (\mathcal{J}(\theta)), \quad \text{where} \quad \mathcal{J}(\theta) := \frac{1}{H} \int_{-H/2}^{H/2} (\mathcal{G}(h, \theta) - \bar{\mathcal{G}}(\theta))^2 dh, \quad (4.14)$$



**Figure 4-4:** Estimating alignment at a point by minimizing the integral of the gray scale over the trial fibre using the MFIA algorithm [164]. Randomly sampled points are used to reconstruct an alignment over the domain.

$\bar{\mathcal{G}}(\theta) = \frac{1}{H} \int_{-H/2}^{H/2} \mathcal{G}(h, \theta) dh$  is the mean gray-scale along the fibre, with  $\mathcal{G}(h, \theta)$  defining the gray-scale at a point  $(x_1 + h \cos \theta, x_3 + h \sin \theta)$  along its length. The procedure is repeated for an array of sample points  $\mathbf{x}^{(k)} = (x_1^{(k)}, x_3^{(k)})$  for  $k \in \{1, 2, \dots, N_\phi\}$  where  $N_\phi$  is number of pixels sampled per image.

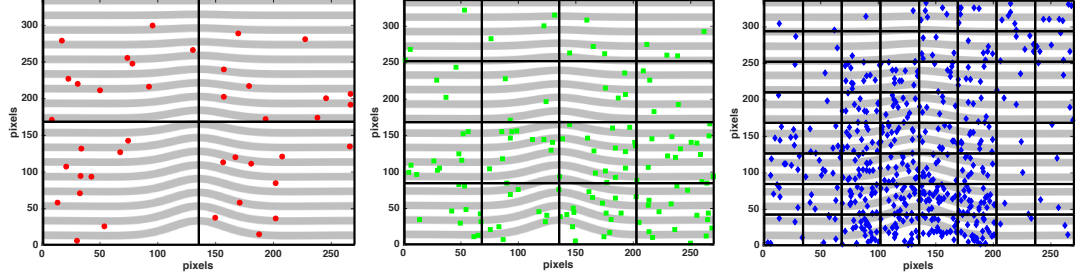
A single point represents an individual optimization problem defined in Eq. (4.14). Sampling every pixel is prohibitively expensive, therefore we want a method to selectively distribute sampling points in such a way that concentrate evaluations to regions of misalignment. To do this we develop a hierarchical approach. The B-scan image is divided into a coarse rectangular mesh (level  $j = 0$ ) with  $m_0$  cells. We generate a sequence of levels by uniformly refining the mesh, given  $m_j = 4^j m_0$  on level  $j$ . The method starts by computing the misalignment  $\phi(\mathbf{x}_k^{(j)})$  at  $n_j^{(i)} = \lceil N_\phi^{(j)} / m_j \rceil$  randomly sampled points in each cell,  $\mathbf{x}_k^{(j)} \in \Omega_i$  for  $k = 1, \dots, n_j^{(i)}$ ,  $i = 1, \dots, m_j$  and  $j = 0$ . In each cell  $\Omega^{(i)}$  we compute the mean absolute misalignment

$$\bar{\phi}_i^{(j)} = \frac{1}{|\mathcal{X}_i^j|} \sum_{\mathbf{x} \in \mathcal{X}_i^j} |\phi(\mathbf{x})|, \quad \text{where the set is defined } \mathcal{X}_i^j = \{\mathbf{x}_k : \mathbf{x}_k \in \Omega_i^j\} \quad (4.15)$$

For each cell on level  $j$  we compute the misalignment and its percentage contribution to the total mean absolute misalignment i.e.

$$\gamma_i^{(j)} = \bar{\phi}_i^{(j)} / \sum_{c=1}^{m_j} \bar{\phi}_c^{(j)} \quad (4.16)$$

For the next level  $j + 1$ ,  $N_\phi^{(j+1)}$  more samples are taken for the four cells  $\Omega^{(j+1)}$  created



**Figure 4-5:** A hierarchical multilevel sampling scheme that is biased towards regions of high misalignment is used draw sampling locations for the trial fibre. The number of new samples per cell on every level are proportional to the relative average misalignment observed in that cell on the previous level. The figure shows level 1 through 3 from left to right and each picture only shows the samples drawn on that level.

from subdividing  $\Omega_i^{(j)}$  we take

$$n_i^{(j+1)} = \lceil \gamma_i^{(j)} N_\phi^{(j)} \rceil \quad (4.17)$$

more samples where  $\bar{\phi}_i$  is the average misalignment of the  $i$ th cell. This is one of the key peripherals developed for the MFIA framework that allows us to generate a continuous alignment field from a manageable number of samples.

*Remark:* We note that MFIA is sensitive to trial fibre length  $H$ . In our experience,  $H = 3t$  produces reliable results where  $t$  is the average ply thickness in pixels. Moreover, these B-scans are flat representations of cornerbends thus requiring a geometric transformation to undo the apparent corner opening.

### 4.3.3 Defining a Wrinkle, Prior and likelihood definition

In the methodology (Section 4.2), the parameterization of a wrinkle defect is left general (4.1). In this section we refine this definition towards the particular application and available data that we consider. The wrinkles are defined by the wrinkle functions

$$W(\mathbf{x}, \boldsymbol{\xi}) = g_1(x_1)g_3(x_3) \sum_{i=1}^{N_w} a_i f_i(x_1, \lambda). \quad (4.18)$$

where  $g_i(x_i)$  are decay functions (defined in Eq. (4.20)),  $f_i(x_1, \lambda)$  are the first  $N_w$  Karhunen-Loève (KL) modes parameterized by the length scale  $\lambda$  and  $a_i$  the amplitudes. In the results which follow both the amplitude modes and the length scale are taken as random variables, so that the stochastic vector is defined by  $\boldsymbol{\xi} = [a_1, a_2, \dots, a_{N_w}, \lambda]^T$ . We now briefly discuss the assumptions under which this choice of wrinkle function has been made.

- 
- **Prismatic in  $x_2$ .** The wrinkle function Eq. (4.18) is assumed to have no  $x_2$  dependency, and therefore the wrinkles are prismatic along the width. Inline with the ASTM standardization [165], four point bend tests were conducted on 52mm wide corner bend samples. In all four samples considered, wrinkles were prismatic over this coupon width.
  - **Karhunen-Loève (KL) modes for wrinkle.** KL modes are widely used in generating random fields, since they generate random functions which display an underlying spatial correlation structure. We note other choices could also have been used e.g. piecewise cubic splines, wavelets or Fourier modes. The set of function  $f_i(x_1, \lambda)$  are proportional the first  $N_w$  (normalized) 1D eigenfunctions associated with the  $N_w$  largest eigenvalues of the one-dimensional two-point, squared exponential covariance operator

$$C(x, y) = \sigma_f^2 \exp\left(-\frac{(x - y)^2}{\lambda^2}\right), \quad \text{for any } x, y \in \mathbb{R} \quad (4.19)$$

The normalizing constant for each mode is the square root of its associated eigenvalues. Further information of KL modes and their use as random fields is widely available, see for example [51]. We note that owing to their natural ordering (in decreasing eigenvalue and wavelength) the modes can efficiently represent functions which display behaviour with a characteristic length scale  $\lambda$  yet offer more flexibility than a simple choice such as  $\sin(\lambda x_1)$ . The runs which follow take  $N_w = 30$ , which was chosen since with an approximate value of  $\lambda = 12.9\text{mm}$ , higher KL modes give undulations on a wavelength shorter than pixels of the B-Scan. Furthermore, from the data we estimate  $\sigma_f = 0.1425$ , but note the output of the model is insensitive to the choice of this value.

- **Wrinkle Decay out side of B-Scan.** For the limited data we have, each B-Scan is centered to the midpoint of the corner radius  $x_1^* = R\pi/4$ , and focuses at a fixed depth  $x_3^* = 4.8\text{mm}$ . For the type of wrinkles considered here, no perturbations are visible on the inside or outside face of the component and the wrinkles were always localized to the corner radius. Again, without further data on their spatial statistics of the wrinkle distribution, we make the simplifying assumption of introducing decay function in both the  $x_1$  and  $x_3$  direction (as also considered in other publications [25, 45], defined by

$$g_i(x_i) = \exp\left(-\left(\frac{x_i - x_i^*}{\eta_{g_i}}\right)^n\right) \quad (4.20)$$

---

For the simulations which follow we take  $\eta_i = -(x_i^*)^4 / \log(10^{-6})$ . This choice of  $\eta_i$  gives the assumption that the wrinkle height is at most  $10^{-6}$ mm outside of the corner radius and on the inner/outer face. For both  $x_1$  and  $x_3$  directions, we choose the hyper-parameter  $n = 4$ . This value is selected to provide the best fit to the observed wrinkle profiles. Better decay functions can be derived from higher quality scans, however, little difference in output was observed for even values of  $n > 4$ .

With more available data, which includes a broader class of wrinkle defects [10, 25, 35, 36], this definition could be generalized. Yet, here the choice is sufficient to demonstrate the methodology, and draw some interesting preliminary engineering results.

Now that we have defined our parameterization of a wrinkle, it remains to define the prior distribution for the random parameters and the parameter  $\Sigma_\epsilon$  for the misfit function. First, denoted  $\pi_0(\boldsymbol{\xi})$ , we define the prior. This is done by analyzing each of four B-Scans and fitting the wrinkle function (4.18) in the least-squared sense using an optimizer (e.g. `fminsearch` in MATLAB [166]). This then provides just four values for each parameter. We approximate the prior as an independent multidimensional Gaussian distribution with mean taken over all measurements and a variance of all 4 samples multiplied by the student t-test factor to account for uncertainty due to only four data points. We note that with just 4 samples and a two sided confidence bounds of 95% this is a factor of 3.18. Secondly, in the definition of the misfit function Eq. (4.5), we require the user-defined correlation matrix  $\Sigma_\epsilon$ , which defines the uncertainty in the measured data. In our case measurement error comes from two sources (1) the accuracy of B-Scan data and analysis method (MFIA) (2) the sampling error of the data since (in our case) we only have 4 samples. To account for the first of these sources of measurement error we assume that all data points are accurate up to  $\pm 2.5^\circ \approx \pm 0.044\text{rad}$ . This was estimated from comparing MFIA outputs to micrographs of wrinkle sections. Although  $2.5^\circ$  seems significant, it is within the error threshold of the scan itself. Further details are not given on how this is constructed, as it is well documented that the MCMC outputs are not sensitive to the fine scale accuracy of  $\Sigma_\epsilon$ . Secondly to account for sample data we rescale  $\Sigma_\epsilon$  by the student t-test factor to (95%) confidence, which we denote  $\tau_{N_\phi}$ . We remark that  $\tau_{N_\phi} \rightarrow 1$  and  $N_\phi \rightarrow \infty$ , and in this case  $\Sigma_\epsilon$  is purely driven by the accuracy of the B-Scan data. For our example we therefore set  $\Sigma_e = \tau_{N_\phi} 0.044\mathbb{I}$ .

---

#### 4.3.4 Finite Element Modelling

For each wrinkle sample generated using the MCMC approach, a finite element analysis is used to predict the corner bend strength (CBS) of that defected component. Finite element modelling was conducted using high performance finite element code **dune-composites** [3]. In the model, the curved laminates were assumed to have the nominal width of 52 mm. The plies were assumed to have a thickness of 0.24 mm, with a 0.015 mm interface layer of pure resin between each ply. This is based upon measurements taken from micrograph images of the curved laminates as described by Fletcher *et al.* [44]. The assumed mechanical properties for both the fibrous ply material and the resin rich interface material are given in Section 4.3.4. A discussion on how they have been chosen from various sources is given by Fletcher *et al.* [44].

Modelling the full 3D bending test (according the ASTM standard [**astm**]) with rollers and contact analysis would be extremely computationally expensive. Therefore a simplified model was used. Curved laminates were modelled with shortened limbs; of length 10mm, approximately equal to the thickness of the laminate. A unit moment was applied to the end of one limb using a multi-point constraint (MPC), with all degrees of freedom fixed at the end of the opposite limb. Whilst this does not accurately model stresses in the limbs, it gives the same stress field towards the apex of the curved section as a full model with rollers. In this region there is a pure moment (without shear) caused by the roller displacement. Since this is the critical region where both wrinkles and failure occurs during the tests, it implies the simplified model is suitable for predicting CBS. The setup of the model is summarized in Fig. 4-6.

Each finite element model contains approximately 1.1 million 3D 20-node serendipity elements, with 8 elements per ply thickness and 4 in the interply regions adding up to roughly 2 million nodes (or 6 million degrees of freedom). This model resolution follows from the mesh convergence study as presented by Reinartz *et al.* [3]. Failure of the coupon is measured according to Camanho's failure criterion [167], whereby a numeric value is assigned to a particular combination of peak tensile and shear stresses.

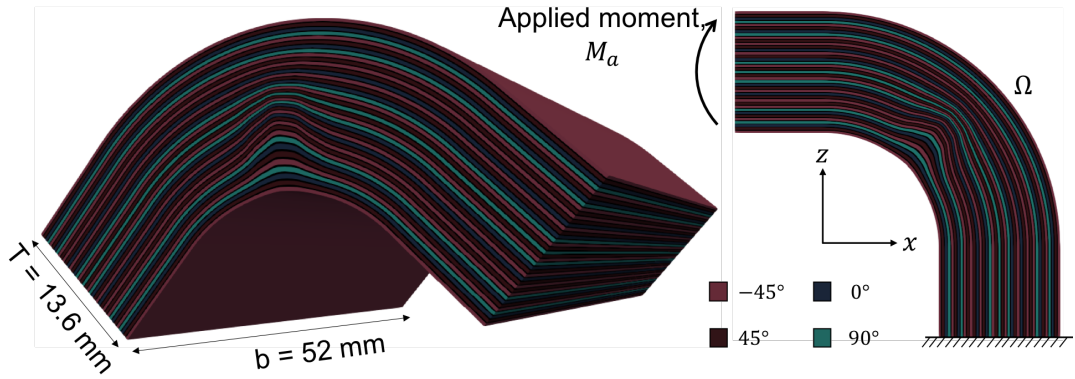
$$\mathcal{F}(\sigma) = \sqrt{\left(\frac{\sigma_{33}^+}{s_{33}}\right)^2 + \left(\frac{\sigma_{23}}{s_{23}}\right)^2 + \left(\frac{\sigma_{13}}{s_{13}}\right)^2} \quad (4.21)$$

Here the subscripts denote the direction of stresses in local coordinates and  $s_{ij}$  denotes allowable stresses. Note that  $\sigma_{33}^+$  is set to 0 if the stress component is negative. Failure occurs when  $\mathcal{F}(\sigma) = 1$ . Usually, failure of a system such as Fig. 4-6 occurs due to delamination which indicates that the peak stresses is likely occur in the resin rich interply regions. Fletcher *et al.* have shown that the Camanho failure criterion predicts

Geometry		Ply properties		Resin properties	
number of plies	39	$E_{11}$	162 GPa	$E$	10 GPa
radius	22 mm	$E_{22}, E_{33}$	10 GPa	$\nu$	0.35
limb length	10 mm	$G_{12}, G_{13}$	5.2 GPa	<b>Allowables</b>	
ply thickness	0.24 mm	$G_{23}$	3.5 GPa	$s_{13}, s_{23}$	97 MPa
interply thickness	0.015 mm	$\nu$	0.35	$s_{33}$	61 MPa

**Table 4.1:** Assumed mechanical properties for CFRP material (M21/IMA), where 1 is the fibre direction in-plane, 2 is perpendicular to the fibre direction in-plane and 3 is out-of-plane.  $s_{33}$  is the tensile through-thickness strength and  $s_{13}$  is the transverse shear strength.

failure to within 5% of average experimental test values [44] with treated edges to mitigate premature failure. Here, for simplicity, we discount the edge effects by not evaluating the failure criterion close to the boundary to isolate the effects of wrinkles. More precisely,  $\mathcal{F}(\sigma)$  is evaluated within a subregion such that  $x_2 \in [15\text{mm}, 37\text{mm}]$  which is 15mm away from each edge.

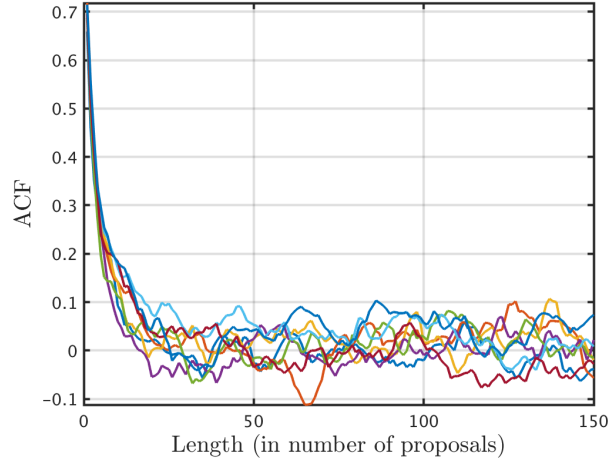


**Figure 4-6:** FE model showing the true geometry of the part with a sample wrinkle amplified for visual clarity. Note that it is a fully internal wrinkle with no trace at the surfaces.

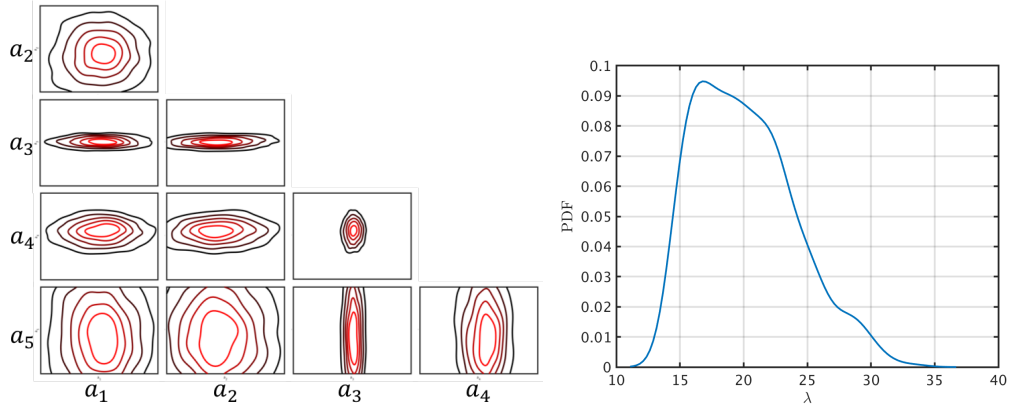
## 4.4 Results

### 4.4.1 Bayesian Sampling of wrinkles

To improve the exploration of the posterior space we initialize five independent Markov chains. For the pCN proposal distribution (4.8) we take  $\beta = 0.25$  and  $\sigma_{PCN} = 1$ . These values were tuned to give an acceptance ratio of approximately 30% as is widely suggested [82]. We first estimate the integrated autocorrelation time  $\Lambda$  for each chain. From Fig. 4-7, we note that  $\Lambda < 100$  in all cases. Random starting positions are sampled



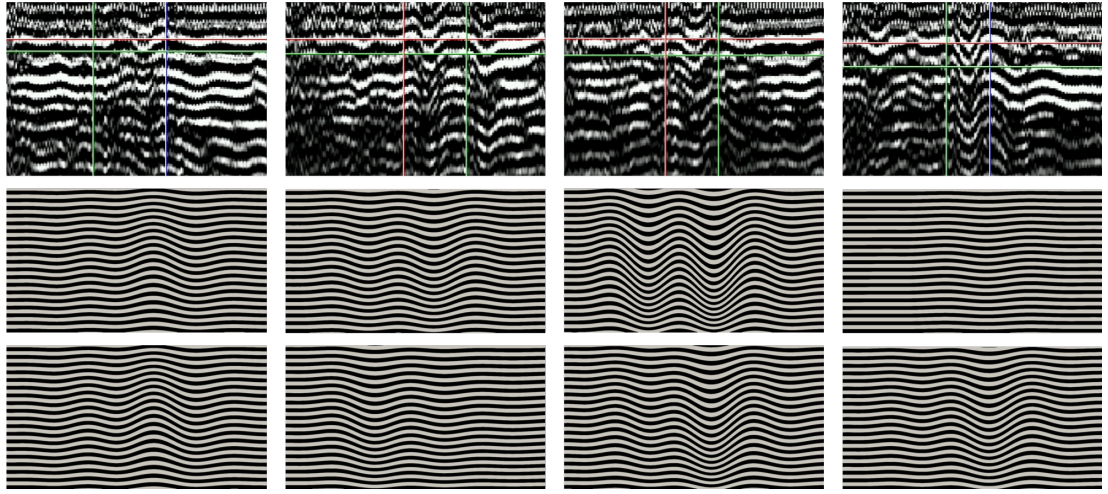
**Figure 4-7:** The ACF showing the longest autocorrelation length across all dimensions of MCMC is illustrated here. Monte Carlo samples of wrinkles are obtained by subsampling every  $\Lambda = 100$  samples.



**Figure 4-8:** (Left) Two-dimensional posterior distributions of the first five coefficients  $a_i$  in Eq. (4.1), note 2-D plot axes are plotted on a scale of  $\pm 0.25$  to visualize dependencies. For example, the  $a_3$  plots suggest that a relatively constant amount of the 3<sup>rd</sup> KL mode compared to others is present in all wrinkles studied here. (Right) Posterior distribution of covariance length scale parameter  $\lambda$  plotted separately.

from the prior, then each chain is ‘burnt-in’ over  $10\Lambda \approx 1000$  MCMC steps. Post burn-in, each chain is subsampled at intervals of  $2\Lambda$  till  $N_{MC} = 200$  independent MCMC samples (or 8,000 dependent samples per chain) are obtained. Posterior distributions of the first five coefficients  $a_i$  from the combined dataset of all chains are visualized in Fig. 4-8. Fig. 4-9 shows a subset of 8 wrinkles out of the 200 samples from the posterior distribution along with the four B-scans.





**Figure 4-9:** Top row shows B-Scan data, bottom two rows show 8 independent posterior samples of wrinkles in B-Scan coordinates.

#### 4.4.2 Monte Carlo simulations

As a benchmark we first calculate the CBD for a pristine part, from which we calculate  $M_c^* = 8.93$  kNmm/mm. We also calculate knock downs for each of the B-Scan samples, by using the maximum *a priori* (MAP) estimates for each scan

$$M_d = \{8.61, 8.88, 8.62, 8.91\} \text{ kNmm/mm} \quad (4.22)$$

Using the MCMC methodology we generate  $N_{MC} = 200$  independent samples from the posterior distributions. All simulations were carried out on 400 cores of the HPC cluster *Balena*, taking approximately 6 minutes per sample. The cluster comprises 192 nodes, each with two 8-core Intel Xeon E5-2650v2 Ivybridge process running at 2.6 GHz. Therefore total core time was approximately 20 hours of computation. In practice, simulation time was less since a number of samples could be run in parallel by using the cluster's  $\sim 3000$  available cores.

From these samples we estimate a mean of  $\mathbb{E}[M_c] \approx 8.72$  kNmm/mm, equating to an average knock down of 2.4%. With a variance of  $\mathbb{V}[M_c] \approx 0.094$  the 200 samples we estimate the 95% one-sided confidence interval of 0.053 (0.6% of the mean value). Given that the finite element error at the mesh resolution chosen is approximately 0.5% (as taken from [3]), the number of samples is sufficient to estimate the mean at the same accuracy as the discretization error given by the finite element model. Therefore no further samples were generated.

Whilst the mean seems like a small deviation from pristine strength, the most severe wrinkle out of 200 samples knocks the strength down by 26%. Therefore, rather than

---

the mean itself we are more interested from an engineering viewpoint in the distribution of strength, particularly in the tails of the distribution, which represent larger knock downs in strength. Figure 4-10 (left) shows the CDF of strength distribution, along with Weibull fits; a common engineering way of quantifying material variability.

The Weibull model for failure assumes that fracture initiates at the weakest link. As Fig. 4-12 shows the existence of a strong nonlinear correlation between the maximum gradient of a wrinkle and its corresponding  $M_c$ , fitting a Weibull curve to the CDF is an appropriate choice. The Weibull curve, defined by Eq. (4.23), predicts the probability of failure  $P$ ;

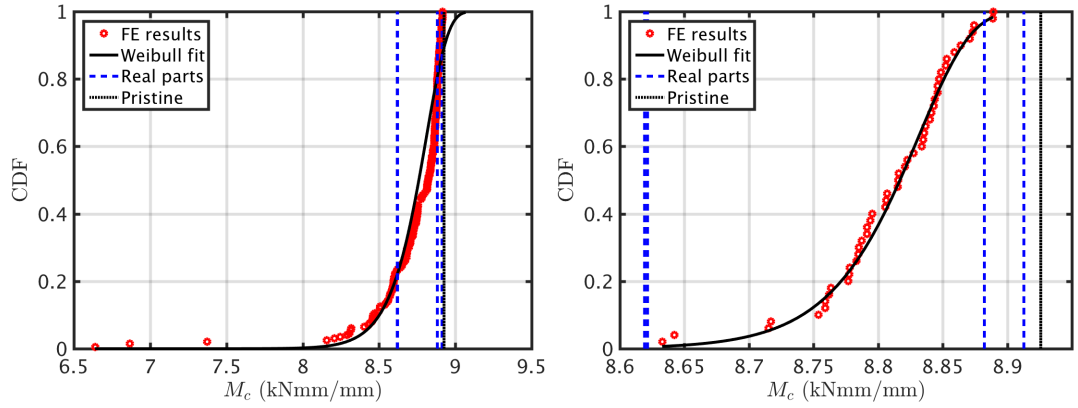
$$\mathbb{P}(M_c|M_W, M_S) = 1 - \exp \left[ - \left( - \frac{M_c}{M_S} \right)^{M_W} \right] \quad (4.23)$$

where  $M_c$  is the critical moment and  $M_S$  is a scale parameter. Perhaps the most important parameter is the Weibull modulus  $M_W$  that can be thought of as a dispersion of defects in a part. A high dispersion is interpreted as an *amorphous* presence of defects thus lacking a clear origin of failure. As a result, the loading bandwidth over which all parts fail is relatively narrow. Conversely, a low  $M_W$  means that defects are concentrated in certain regions in such way that failure usually originates from these hot spots.

Figure 4-10 provides a comparison between the distributions obtained by sampling using the Markov chain methodology (left) in contrast to normal sampling (right) whereby coefficients  $\xi^{(k)}$  are drawn at random from a normal distribution centered about the observed mean with population standard deviation corrected for  $n-1$  degrees of freedom with 95% confidence bounds. For samples shown in Fig. 4-10 (left) the Weibull modulus  $M_w = 62.9$ . On the other hand, for samples shown in Fig. 4-10 (right), the Weibull modulus was found to be  $M_w = 218.6$ . The higher modulus indicates an even more uniform spread of softer regions. The modulus demonstrates the overly conservative nature of normal sampling. It does not take into account the intrinsic correlation structure of the coefficients by assuming all coefficients are independent of each other. As a result, wrinkles generated this way are not likely to resemble observed defects, therefore, Markov chain sampling is preferred. This is an important result because it further strengthens the case for application of Bayesian methods to the wrinkle problem by clearly demonstrating the conservative nature of the current approach.

#### 4.4.3 An ‘engineering model’ for Corner Bend Strength

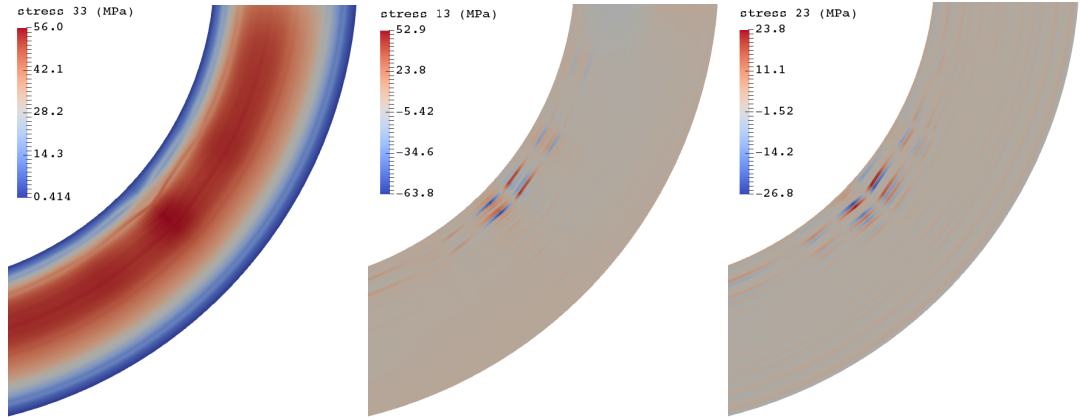
In this section we describe how the Bayesian methodology alongside a finite element model can be used to derive a distribution of corner bend strength due to random



**Figure 4-10:** (Left) CDF of critical or failure moment  $M_c$  per unit width of a part, where wrinkle distributions are using the Bayesian framework introduced within this chapter. (Right)  $M_c$  of samples obtained by assuming a Gaussian prior with mean and variance derived from data.

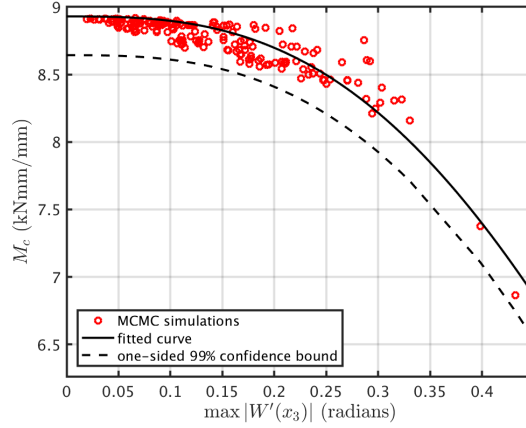
wrinkle defects, which in turn is parameterized by a Weibull model. We show how the results from these large finite element calculations can be distilled into a much simpler engineering approach or ‘look up’ model, which provides some practical means of assessment on determining the influence of an observed wrinkle.

Figure 4-11 shows the wrinkle extracted from the B-scan in Fig. 4-3 embedded into the corner bend sample. The magnitude of the wrinkle in the B-scan is misleading since the pixels represent a length approximately 8 times larger in the vertical in comparison to the horizontal. In reality, the wrinkle is much smaller as shown in stress plots.



**Figure 4-11:** Left to right showing  $\sigma_{33}$ ,  $\tau_{13}$  and  $\tau_{23}$  respectively at  $M_a = 1$  kNmm/mm.

An inspection of Fig. 4-11 shows that interlaminar tensile stress ( $\sigma_{33}$ ) concentrations form at peak wrinkle curvature whereas interlaminar shear ( $\tau_{13}, \tau_{23}$ ) is concentrated around maximum wrinkle slope. Moreover, the allowable tensile stress ( $s_{33}$ ), 60% lower



**Figure 4-12:** Approximating the relation between  $M_c$  and maximum wrinkle slope with Eq. (4.24).  $q = 2.867$  and  $\lambda_q = 4.212$  for the fitted curve.  $q = 2.587$  and  $\lambda_q = 3.834$  for lower 99% confidence bound.

than allowable shear  $(s_{13}, s_{23})$ , lends a greater contribution to  $\mathcal{F}(\sigma)$  in Eq. (4.21). We therefore explore the correlation between maximum slope a knock down in strength. For this we parameterize the slope-failure dependency with an exponential relationship defined by

$$M_c = M_c^* \exp \left[ - \frac{W'(x_3)^q}{\lambda_q} \right] \quad (4.24)$$

where  $M_c^*$  is the strength of the pristine part. Model parameters,  $q = 2.867$  and  $\lambda_q = 4.212$  for the fitted curve in Fig. 4-12, enable prediction of critical moment for a given maximum gradient. The lower 99% confidence bound for that prediction is computed with  $q = 2.587$  and  $\lambda_q = 3.834$ .

## 4.5 Concluding remarks

This chapter proposes a generalized framework to quantify the effects of wrinkles in large composite structures. It combines ideas from NDT, image processing, Bayesian inference and FE modelling to create a rigorous methodology for visualizing, parameterizing and computing strength of wrinkles. The methodology is demonstrated through an industrially motivated case study with field data where we show the over conservative nature of the current design approach in comparison to the Bayesian, data-driven method to strengthen the certification by simulation idea.

Two dimensional ultrasonic scans (B-scans) are used to visualize wrinkles that form inside manufactured parts. They are parameterized using a Karhunen-Lo  ve basis due to their suitability for capturing multiple localized features. A possibly

---

true distribution is inferred from observed wrinkles. The forward model - a finite element model - then determines the strength of a composite corner bend with a wrinkle embedded in it. Evaluating the forward model for a variety of defects, elucidates knockdown distribution. Whilst we focus on the influence of wrinkle defects, the general framework could be readily applied to other types defects, for example porosity provided an adequate basis is selected.

The theoretical strength for 200 independent MCMC wrinkles is evaluated to give an expected value approximately 2% lower than the pristine strength. The same test for normally sampled wrinkles gives a much poorer estimate as the results fail to capture some of the observed wrinkles. The worst MCMC case however, suffers from a knockdown of approximately 26%. The cumulative strength distribution is well approximated by a Weibull curve with a relatively high Weibull modulus which is interpreted as the lack of a dominant failure mechanism or origin. In other words, it is difficult to find one particular wrinkle parameter universally responsible for failure.

An engineering model is constructed based on the significant negative correlation found between maximum gradient and critical moment of failure. Owing to its non-linearly decreasing nature, the gradient-failure relationship is parameterized by a negative exponential to produce a directly usable look up chart to estimate knockdown of a particular wrinkle.

We emphasize that this work demonstrates a method and the selected case study represents a narrow bandwidth of possible wrinkles due to a small set of observations. The available training data is a set of scans of pronounced wrinkles only, which makes it impossible to deduce their probability of occurrence in the first place. Consequently, all parts simulated here have wrinkles and suffer some strength knockdown. We do not claim our algorithm provides minimal error since the limited data set restricts the performance of the algorithm - a problem expected to be mitigated by a richer data set. Instead, we argue that industry design standards may now be challenged or reformulated. Moreover, there remains some room to develop application specific bases and more accurate estimates of decay functions and location parameters. In this way, building true representations of the parameters at the coupon level affords us a sampling space from which defects can be generated and embedded into much larger components models.



---

---

# CHAPTER 5

---

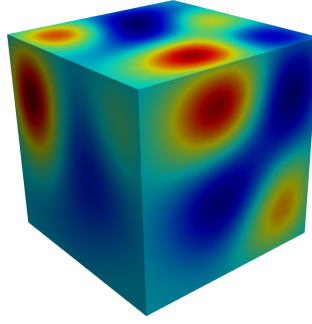
## MULTISCALE METHODS FOR COMPOSITES

### 5.1 What is a good multiscale model?

In the context of this thesis, a good multiscale model must be able to do two things. Firstly, it must capture important information across all scales that may lead to an uncertain mechanical response. Secondly, it must be inexpensive to solve relative to a corresponding fine scale model with full resolution.

The motivation for multiscale modelling of composites derives from the lack of scale separation exhibited by them. For instance, introducing a ply at  $45^\circ$  in a stack of two  $0^\circ$  plies creates bend-twist coupling such that any bending in the new stack will invoke some degree of twist. To model this modified stack, now requires finite elements that can resolve stresses and strains in the  $45^\circ$  ply in order to capture the twisting behaviour. As the complexity of stacking sequences increases, interactions between plies give rise to complicated mechanics. Even the resin rich interply regions become important which are typically a fraction of the ply thickness. This places an upper limit on the size of the mesh parameters  $h$  which in turn produces large systems of equations to be solved. Such systems can only be solved iteratively. Furthermore, the contrasting properties of plies and resin makes this system of equations ill-conditioned. This is important because ill-conditioning can be beneficial in the sense that a stiffness matrix containing kernels demonstrates potential for a multiscale approach.

To understand the beneficially ill-conditioned nature in a mathematical way, let

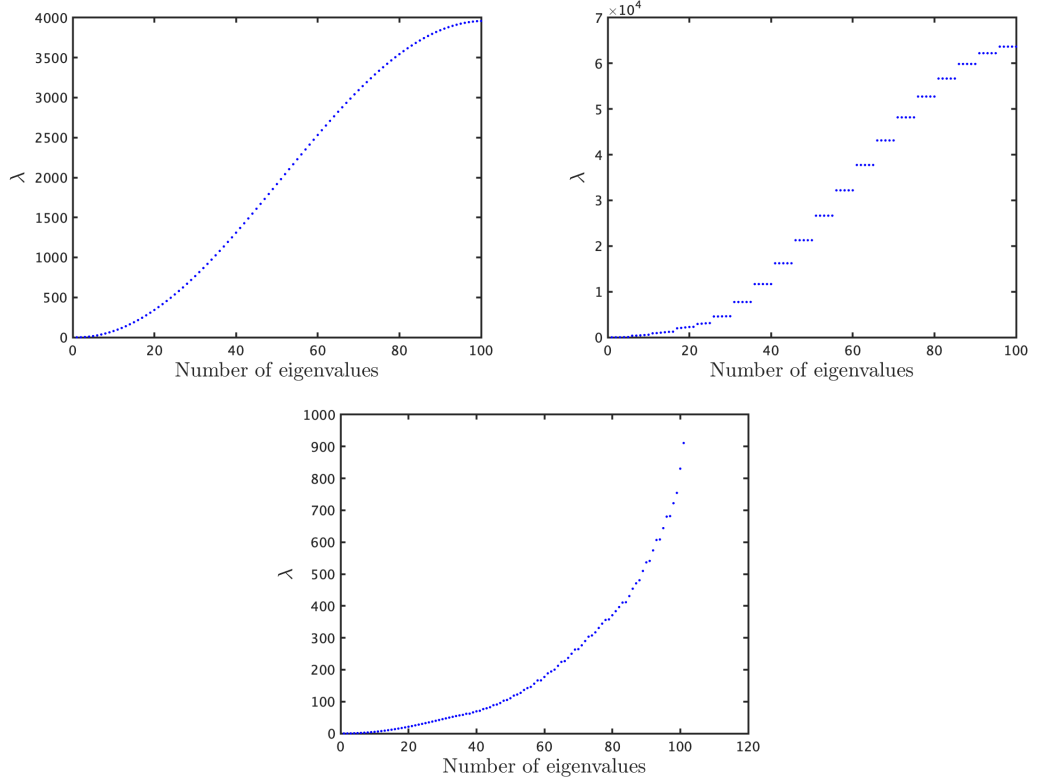


**Figure 5-1:** *A Gaussian random field showing a more continuous distribution of scales such that features are not readily distinguishable using discrete lengths scales.*

us reconsider the 1 dimensional composite beam problem shown in Fig. 2-14. We consider a one dimensional isotropic bar with  $E_{resin} = 10$  GPa and a composite bar with the same resin used to bind composite layers with  $E_{composite} = 162$  GPa in order to investigate their eigenvalue spectrum, see Fig. 5-2. The eigenvalue spectrum of the isotropic material is noticeably smoother than the composite counterpart. From the perspective of a composite material under load, it is easier to deform the softer resin rich zones as opposed to the stiffer carbon fibre rich zones. So the softer, resin rich regions accommodate more strain than the stiffer, carbon fibre dominated zones. This is represented by the lower part of the spectrum where the eigenvalues are smoothly increasing because most of the initial strain is absorbed by resin deformation. Beyond a certain load however, it becomes more energy efficient to share the deformation between strong and weak zones causing a jump in the spectrum every time a new energy demanding (composite) mode is excited. For every carbon fibre dominated eigenmode, there exist several arrangements of resin deformation with similar (slightly greater) energy requirements which is why the composite eigenspectrum is stepped. Once a composite eigenmode has exhausted all its resin deformation configurations, it must invoke the next high energy mode if a solution has not yet been obtained. In generalizing this idea to heterogeneous materials, the scales may not be separated but smeared. It is difficult to pick out discrete length scales that describe the overall structure of the material. For example, Fig. 5-1 shows a 3D Gaussian random field on a unit cube. On inspection it is clear that the mechanics of this material cannot be segregated into high and low energy regimes, as the heterogeneities occur across many length scales throughout the domain. This is reflected in Fig. 5-2 (bottom) which shows a smoother eigenspectrum for a 1D Gaussian random field. Therefore, ill-conditioning occurs due to the presence of low energy deformation modes. A good multiscale model should be able to capture these.

A composite type eigenspectrum resulting from strong spatial variation of material





**Figure 5-2:** *Eigenvalue spectrum of (left) an isotropic material, (right) a composite and (bottom) a Gaussian random field.*

properties gives rise to convergence issues. This can be overcome in one of two ways; 1) refine the mesh till the transition in strain across the material boundary is smoother or 2) refine the basis to include more information which may be in the form of higher order elements (p-refinement) or multiscale functions. The former approach known as h-refinement increases the number of unknowns which, on occasion may be the only way to a solution, assuming it can be solved. It is the less interesting of two options because refining a mesh almost always improves the solution. Therefore, this chapter elaborates on the second approach whereby we wish to improve accuracy via multiscale enrichment.

In continuum mechanics, the second approach led to the development of higher order models like Cosserat [30] and strain-gradient continuum [30, 168] and other multiscale methods like GMSFEM [90], reviewed in Section 2.5. We showed in Chapter 2 the unsuitability of RVEs and other homogenization methods due to restrictive assumptions on the material such as periodicity and scale separation. Spectral methods such as GMSFEM are the most promising since they provide a naturally ordered list of deformation modes by solving a suitable eigenvalue problem. They rely on multiscale

---

basis functions, obtained from eigenproblems, that form a coarse space which is used to construct multiscale models. GMSFEM is similar to p-refinement however, unlike p-refinement, the multiscale basis is physics based from which macroscopically relevant low energy modes are selected for enrichment. All such methods are model reduction concepts, the major difference being in the construction of the multiscale bases. More precisely, the difference is in the choice of eigenvalue problem. In GMSFEM, the coarse space is a ranked collection of orthogonalized displacement solutions known as snapshots.

A similar approach is called Multiscale Finite Element Method (MFEM) [169] where the basis functions on each finite element are adapted to the local properties of its differential operator. This encodes the fine scale variation into the finite element basis functions. We note that the idea of using special basis functions for FE methods is not new and has been used previously in [170, 171], although mostly restricted to scalar valued porous flow problems. MFEM requires individual realizations for varying parameters and may not be a viable method in a stochastic sense. However, an extension of MFEM proposed by Aarnes *et al.* remedies the issue [172]. The key idea is to construct multiscale basis functions that not only resolve fine scale information but also the spatial variability across realizations in a stochastic media. This is done by computing bases for selected samples from the stochastic permeability field. The sampling of these realizations however, is not discussed in their paper. In this chapter, we focus on coarse space construction from the viewpoint of a single realization and leave the stochastics to Chapter 6.

In Chapter 3, we develop a bespoke two level additive Schwarz preconditioner to solve large ill-conditioned systems of equations. This is important because the spectral coarse space used to build the second level of the **GenEO** preconditioner captures individual modes of overall mechanical response and ranks them according to their energy requirement. The lowest ranked are the zero energy modes which form the Nicolaides coarse space [126]. Multiscale models built with the Nicolaides coarse space can only capture translations and rotations but not deformations. Therefore, it is the information content of the coarse space that determines the quality of the resulting multiscale model. For example, in Fig. 2-15 we showed that convergence of an iterative solver is significantly improved by adding the Nicolaides coarse space on the second level. However, in a multiscale sense, this coarse space does not contain sufficient information. Several alternative choices of spaces exist that may be chosen for the second level of the preconditioner and the multiscale model as discussed in [90]. The eigenproblem in our multiscale method is identical to GMSFEM (see Eq. (3.13)) except in the choice of the right hand side matrix. GMSFEM uses the mass matrix whereas our method

---

employs the stiffness matrix  $\mathbf{A}_{\Omega_j^\circ}$ . We note that the right hand side contains a matrix product between the subdomain matrix and the partition of unity (PoU) operator, both of which contribute to the quality of the coarse space built in this way.

The aim of this chapter is to demonstrate via a proof of concept the conversion of theoretical mathematical methods into real engineering applications. Typically this is done by solving 2D scalar problems to model porous media with synthetic data sets. In this chapter, we use the **GenEO** basis to build multiscale models and explore the effects of various PoUs in both 2D and 3D. We also demonstrate a prototype for a bespoke physics based PoU to obtain high quality multiscale models. Stress results obtained via this multiscale method are post processed to further improve stress solutions locally through stress recovery.

## 5.2 Preliminaries

This section is a continuation of the preliminaries in Section 3.2 so we begin with the system of linear equations  $\mathbf{A}\tilde{\mathbf{u}} = \mathbf{b}$ . Although, some of the notation was previously defined in Chapter 3, we reiterate it for convenience.

Let us partition the domain  $\Omega$  into  $N$  non-overlapping subdomains denoted  $\Omega'_j$  where  $j = 1, \dots, N$  and then extend each  $\Omega'_j$  by  $O$  layers of elements to build overlapping partitions denoted as  $\Omega_j$  such that  $\Omega = \cup_{j=1}^N \Omega'_j$ . The overlapped degrees of freedom in the  $j$ -th subdomain then belong to the set

$$\Omega_j^\circ := \{\mathbf{x} \in \Omega_j : \exists i \neq j \text{ such that } \mathbf{x} \in \Omega_i\} \quad (5.1)$$

From the example in Fig. 5-3 (right)  $\Omega_1^\circ = \{2, 3, 4\}$ . The internal dofs of a subdomain are defined by the following set

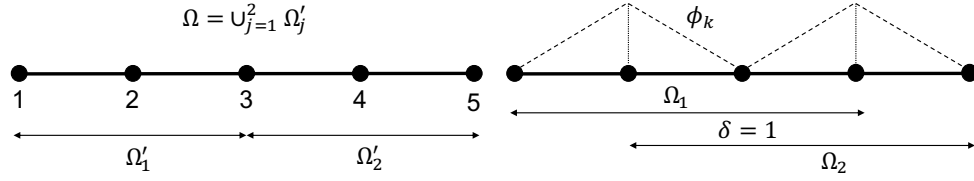
$$\text{dof}(\Omega_j) := \{k : 0 \leq k \leq n \text{ such that } \text{supp}(\phi_k) \subset \Omega_j\} \quad (5.2)$$

and all *active* dofs on that subdomain are denoted

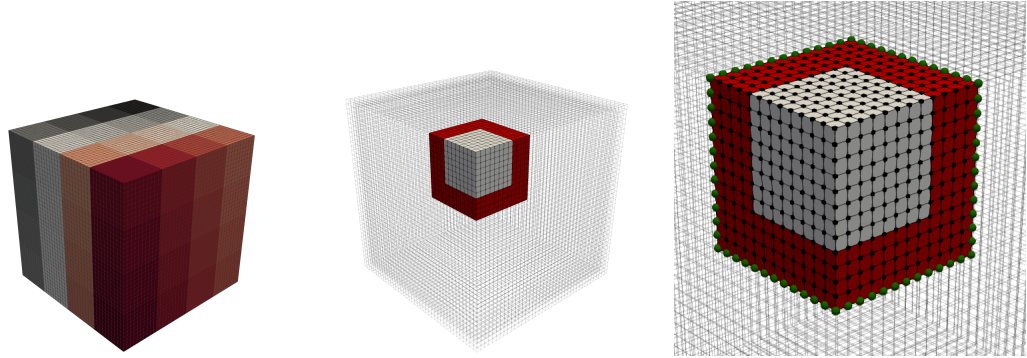
$$\overline{\text{dof}}(\Omega_j) := \{k : 0 \leq k \leq n \text{ such that } \text{supp}(\phi_k) \cap \Omega_j \neq \emptyset\} \quad (5.3)$$

Let us demonstrate these definitions with a simple 1D example shown in Fig. 5-3 where  $\text{dof}(\Omega_1) = \{1, 2, 3\}$  and  $\overline{\text{dof}}(\Omega_1) = \{1, 2, 3, 4\}$ . Now, we define the function space  $V_h(\Omega_j)$  on a subdomain

$$V_h(\Omega_j) := \{v|_{\Omega_j} : v \in V_h\} \quad (5.4)$$



**Figure 5-3:** One dimensional example to clarify definitions in Section 5.2. (Left) Non-overlapping partitions of  $\Omega$ . (Right) Partitions of  $\Omega$  overlapped by one element giving an overlap 2 elements wide.



**Figure 5-4:** (Left) The domain  $\Omega$  divided into 64 non-overlapping subdomains  $\Omega'$ . (Middle) The solid coloured region shows overlapping subdomain  $\Omega_j$  where the red subregion is denoted by  $\Omega^o$  in our notation and the white subregion is  $\Omega_j \setminus \Omega_j^o$ . (Right) Close up of  $\Omega_j$  where the set  $\text{dof}(\Omega_j)$  is marked by square black grid points while the spherical green grid points on the boundaries denote the set  $\overline{\text{dof}}(\Omega_j) \setminus \text{dof}(\Omega_j)$ . Note: we assume that all surfaces are Dirichlet boundaries.

---

and restrict  $V_h(\Omega_j)$  to only contain functions with complete support on  $\Omega_j$ . We define this restricted space as

$$V_{h,0} := \{v \in V_h(\Omega_j) : \text{supp}(v) \subset \Omega_j\} \quad (5.5)$$

Therefore, a function  $v \in V_{h,0}(\Omega_j)$  exists on a subset of dofs in  $\Omega_j$ . To build a global basis, we must be able to extend any function  $v \in V_{h,0}(\Omega_j)$  to  $V_h$ . This transformation is achieved by an extension operator  $R_j^T : V_{h,0}(\Omega_j) \rightarrow V_h$  whose transpose then becomes the restriction operator  $R_j : V_h \rightarrow V_{h,0}(\Omega_j)$ . In matrix form,  $\mathbf{R}_j$  and  $\mathbf{R}_j^T$  are sparse binary matrices mapping global dofs to local dofs or vice-versa. For our 1D example shown in Fig. 5-3, the operators would be defined as

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{R}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

With the restriction operator, Neumann matrices on subdomain can be defined as  $\mathbf{A}_j = \mathbf{R}_j \mathbf{A} \mathbf{R}_j^T$  for  $j = 1, \dots, N$ . For large systems, however, constructing  $\mathbf{A}$  is a time consuming task which we wish to avoid. The **dune-composites** package [3] achieves this by local assembly of  $\mathbf{A}_j$  on each subdomain. We introduced in Eq. (3.11), a coarse grid problem that couples all subdomains, giving a two-level preconditioner. Like before, we require another operator to restrict the fine space  $V_h$  to the coarse space  $V_H$  which we denote  $R_H : V_h \rightarrow V_H$ . Once again, its transpose defines a prolongation operation, both of which in matrix form are  $\mathbf{R}_H$  and  $\mathbf{R}_H^T$ . Now, we must define a suitable coarse space  $V_H$ .

### 5.2.1 Customized coarse space - GenEO

We have highlighted in Chapter 3 that convergence of iterative solvers deteriorates in the presence of global low energy modes in  $\mathbf{A}$  and proposed a way to overcome it using a two level additive Schwarz framework. This is important because the key ingredient at the second level is the **GenEO** coarse space. To build this coarse space we need to be able to do two things,

1. Define a suitable generalized eigenvalue problem on each subdomain. This is given in Eq. (3.13).
2. Combine the local solutions to build a global coarse space. This is done via a partition of unity operator given in Definition 3.3.2. We explore two other PoUs

---

in this chapter, the definitions of which are given in Section 5.2.2.

In this way we have a coarse space  $V_H$  that captures a user controlled amount of fine scale information with which we can formulate our multiscale problem. In matrix form, the number of columns in the mapping  $\mathbf{R}_H$  is equal to  $\dim(V_h)$  with  $Nm$  rows. Each row is computed as a matrix product  $\mathbf{R}_j \mathbf{X}_j \mathbf{p}_i^{(j)}$  for  $j = 1, \dots, N$  and  $i = 1, \dots, m$ . With all the operators defined, we reformulate the problem in the customized coarse space so that

$$\mathbf{A}_H \tilde{\mathbf{u}}_H = \mathbf{b}_H \quad (5.7)$$

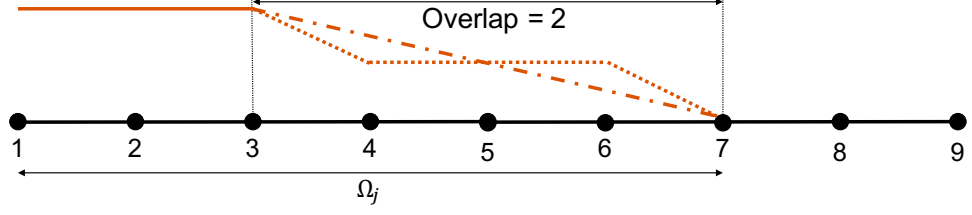
where  $\mathbf{A}_H = \mathbf{R}_H \mathbf{A} \mathbf{R}_H^T$ ,  $\tilde{\mathbf{u}}_H$  is the solution vector in  $V_H$  and  $\mathbf{b}_H$  is the coarse load vector. Transforming the solution and load vectors from  $V_H$  to  $V_h$  is also achieved with the extension operator such that the fine scale solution  $\tilde{\mathbf{u}} = \mathbf{R}_H^T \tilde{\mathbf{u}}_H$ . In this way we recover the fine solution from a coarse solve.

We emphasize that the generalized eigenvalue problems per subdomain are independent and can be solved in parallel thus reducing the problem from one expensive fine scale solve to  $N$  independent parallel solves that enable one cheap coarse solve. However, compatibility of eigenmodes on adjoining subdomains is a necessary requirement. It is for this reason that nearest neighbour communication between subdomains becomes necessary for efficiently parallelized assembly of  $\mathbf{A}_H$  and  $\mathbf{b}_H$ . Therefore,  $\mathbf{A}_H$  is a small but dense matrix.

### 5.2.2 Partition of Unity

In Chapter 3 we observe that the influence of the choice of PoU in constructing the two level additive Schwarz preconditioner has little bearing on the convergence of the solution. For the tests conducted in Chapter 3, the smooth PoU reduced the iteration count by 1 at most. However, when building a multiscale model using the **GenE0** coarse space, the treatment of overlaps is not trivial and significantly affects the solution. In order to gain a better understanding of this influence, we create a new solution based PoU in addition to the smooth and non-smooth.

The formal definition of the non-smooth PoU is given in Definition 3.3.2. A one dimensional visual example is provided in Fig. 5-5 where the dotted line represents this PoU. The unsmooth operator is exceedingly simple whereby every node is assigned a scalar integer  $\xi_k$  equal to the number of subdomains sharing that node. The contribution of any node to all the subdomains it belongs to is equally divided such that individual contributions are proportional to  $\xi_k^{-1}$ . The operators  $\Xi_j$  are encoded as



**Figure 5-5:** Visual comparison of the two partition of unity operators in one dimension.

diagonal matrices  $\mathbf{X}_j$  that satisfy the following criteria

$$I = \sum_{j=1}^N \mathbf{R}_j^T \mathbf{X}_j \mathbf{R}_j \quad (5.8)$$

where  $I$  is an identity matrix.

Like the non-smooth PoU, the smooth PoU as defined by Sarkis [142] also introduces a scalar per node such that

$$\tilde{\xi}_k = \begin{cases} 1 & \text{on all nodes in } \Omega_k^0 \\ 1 - \frac{\delta}{l} & \text{on all nodes in } \Omega_k^\delta \setminus \bar{\Omega}_k^{\delta-1} \forall \delta \in [1, l] \end{cases} \quad (5.9)$$

where  $\bar{\Omega}_j^{m-1}$  denotes the inner nodes of the  $m$ -th layer of overlap. Unlike the non-smooth PoU, the smooth PoU is a piecewise linear function in overlaps. With  $\tilde{\xi}_k$  defined, the local partition is computed as

$$\xi_k = \frac{\tilde{\xi}_k}{\sum_{j \in \Omega_k^0} \tilde{\xi}_j | \Omega_k^l \cap \Omega_j^l |} \quad (5.10)$$

We now demonstrate the two partition of unity operators for the example in Fig. 5-3. Both PoUs are identical for  $\delta = 1$  so they can be expressed for two subdomains with matrices  $\mathbf{X}_1$  and  $\mathbf{X}_2$

$$\mathbf{X}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{X}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.11)$$

The two PoUs presented so far are purely geometrical in nature and can be expected to represent isotropic overlaps reasonably well. However, they do not capture the anisotropy in the overlaps which is why the eigenmodes computed over two neighbour-

---

ing domains, although compatible, do not represent well the mechanical response of anisotropic materials. Motivated by this requirement, we define a solution based PoU which we refer to as *uPoU*. This particular partition is based on the displacement solution in the overlaps. In order to construct the uPoU on subdomain  $j$ , we must define two new quantities. First, the interior boundary of its overlap  $\partial\Omega_{j,\text{int}}^\circ$ . Second, the exterior boundary of  $\Omega_j^\circ$  defined as  $\partial\Omega_{j,\text{ext}}^\circ$ . The interior and exterior boundaries are shown in red and black nodes respectively, for an arbitrary 2D subdomain in Fig. 5-6. This PoU, like any other, must satisfy  $\partial\Omega_{j,\text{int}}^\circ = 1$  and  $\partial\Omega_{j,\text{ext}}^\circ = 0$  along with Eq. (5.8). An important consideration is that any vector valued solution in the presence of Poisson's effects will invariably generate negative displacements that violate the partition of unity. As a remedy, we propose decoupling the mechanical response in the principal directions i.e. remove Poisson's effects and ensure the stiffness matrix on the overlap  $\mathbf{A}_{\Omega_j^\circ}$  is block diagonal. This is equivalent to solving independent scalar valued problems in each direction. We note that the decoupling only applies to uPoU construction and the solution includes Poisson's effects. With these in mind, we can now construct the uPoU on the  $j$ th subdomain using the following recipe,

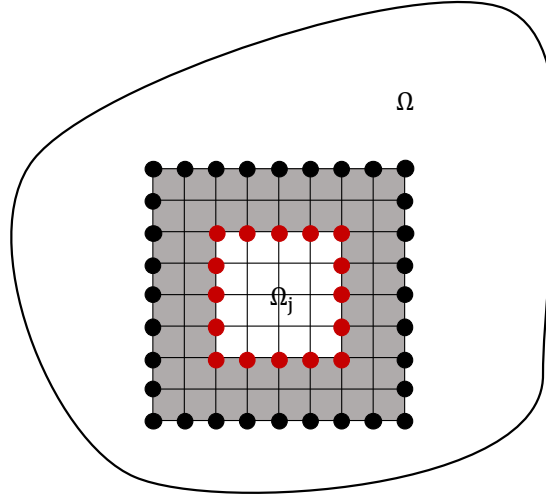
1. Assemble uncoupled  $\mathbf{A}_{\Omega_j^\circ}$
2. Apply boundary conditions  $\partial\Omega_{j,\text{int}}^\circ = 1$  and  $\partial\Omega_{j,\text{ext}}^\circ = 0$ .
3. Solve for displacements  $\tilde{\mathbf{u}}_h(\Omega_j^\circ)$
4. Compute global  $\tilde{\xi}_k = \sum_{j=1}^N \tilde{\mathbf{u}}_h(\Omega_j^\circ)$  to get nodal values of uPoU on  $\Omega_j$

Since  $\tilde{\xi}_k$  is the sum of displacements in  $\Omega_j^\circ$  for all  $j$ , it is not 1 everywhere. In order to satisfy Eq. (5.8), we recompute  $\xi_k$  using Eq. (5.10).

### 5.3 Numerical experiments

In this section we demonstrate the performance of **GenEO** as a multiscale method. We begin with two dimensional simulations of a composite laminate, followed by a heterogeneous material modelled with a Gaussian random field. We use Gaussian random fields because, like composites they too exhibit insufficient scale separation and a high contrast spatial variation of material properties thus enabling us to demonstrate the scope of the method. We then extend the analysis to three dimensional bending of a composite beam.





**Figure 5-6:** The red nodes show the interior boundary  $\partial\Omega_{j,int}^o$  while the black nodes show the exterior boundary  $\partial\Omega_{j,ext}^o$  of an arbitrary 2D subdomain.

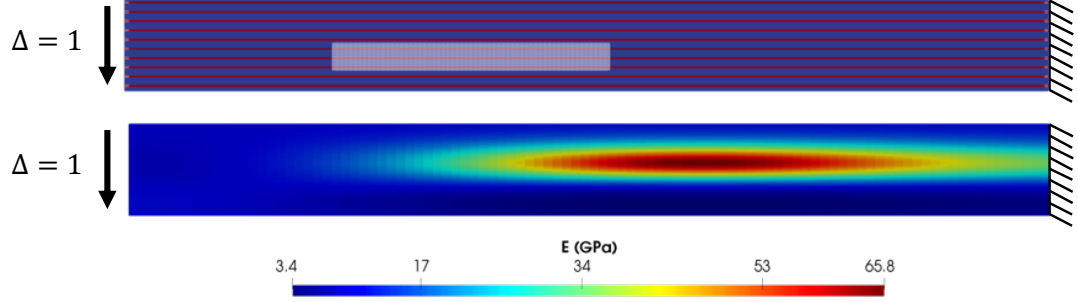
#### GenEO coarse space as a multiscale method in 2D

We consider a rectangular two-dimensional domain with vertical Dirichlet boundaries and horizontal Neumann boundaries. We use this domain to conduct two separate tests by modelling;

1. A unidirectional composite with properties given in Table 5.1, shown in Fig. 5-7 (top)
2. A heterogeneous material modelled with a Gaussian random field without classical scale separation, see Fig. 5-7 (bottom).

Since we are particularly interested in the composite case, we will consider it first. We use these simulations to compare all three partition of unity operators for different overlaps and  $\dim(V_H)$  in order to demonstrate the performance of **GenEO** as a multiscale method. For the composite test, we consider a beam of length  $L_x = 10\text{mm}$  and thickness  $L_y = 1\text{mm}$ , made up of 11 carbon fibre plies  $0.08\text{mm}$  in thickness, orientated at  $0^\circ$ . They are separated by  $0.02\text{mm}$  thick resin rich interplies. The beam is discretized into a mesh comprising 200 elements in each direction such that there are 40,000 elements or 40,401 nodes adding up to 80,802 degrees of freedom.

Each interply layer is modelled with 4 linear quadrilateral elements while the plies are 16 elements thick with mechanical properties provided in Table 5.1. The setup of the unidirectional laminate with  $0^\circ$  plies along the x-axis and the applied constraints are shown in Fig. 5-7 (top).



**Figure 5-7:** (Top) 2D composite elastic field showing the layered material with applied constraints. The domain is discretized into  $4 \times 4$  subdomains and  $\Omega_6$  is overlaid in a white mesh. Deformations for  $\Omega_6$  are shown in Fig. 5-9. (Bottom) Gaussian random field for the same problem with a high contrast  $\approx 100$ .

Ply properties		Resin properties	
$E_{11}$	162 GPa	$E$	10 GPa
$E_{22}, E_{33}$	10 GPa	$\nu$	0.25
$G_{12}, G_{13}$	5.2 GPa		
$G_{23}$	3.5 GPa		
$\nu_{12}$	0.25		

**Table 5.1:** Assumed mechanical properties for CFRP material (M21/IMA), where 1 is the fibre direction in-plane, 2 is perpendicular to the fibre direction in-plane.

---

The mesh is subdivided into  $N = 16$  regular subdomains such that individual GenEO problems are solved for each to give the required number of eigenmodes. The eigenmodes from each subdomain are extended with the  $\mathbf{R}_j^T$  operator to build the second level restriction operator  $\mathbf{R}_H$  to reduce model order according to Eq. (5.7). For this problem, the  $\mathbf{R}_H$  matrix is  $16m \times 80,802$  in size and  $\mathbf{A}$  is of size  $80,802 \times 80,802$ . In practice, the full  $\mathbf{R}_H$  and  $\mathbf{A}$  are never constructed. Instead, only the local contribution of subdomain  $j$  to  $\mathbf{A}_H$  is computed as

$$\mathbf{A}_H^{((j-1)m+k, (i-1)m+k)} = \mathbf{R}_{H,j}^{(k)}(\Omega_j) \mathbf{A}_j \mathbf{R}_{H,i}^{T,(k)}(\Omega_j) \quad \forall i \in \hat{N} \quad (5.12)$$

where  $\hat{N}$  is the set of immediate neighbours of subdomain  $j$  including itself and  $k = [1, 2, \dots, m]$ .

For a good coarse space, the coarse approximation should converge rapidly to the fine solution as the dimensionality of the coarse space is increased. For this particular problem, tests were conducted with  $m = [5, 10, \dots, 30]$  modes per subdomain. To quantify the quality of the coarse representation, we define the error as a difference between solutions in the energy norm such that

$$\varepsilon_e = \log_{10} \left( \frac{\mathbf{u}_e^T \mathbf{A} \mathbf{u}_e}{\tilde{\mathbf{u}}_h^T \mathbf{A} \tilde{\mathbf{u}}_h} \right) \quad (5.13)$$

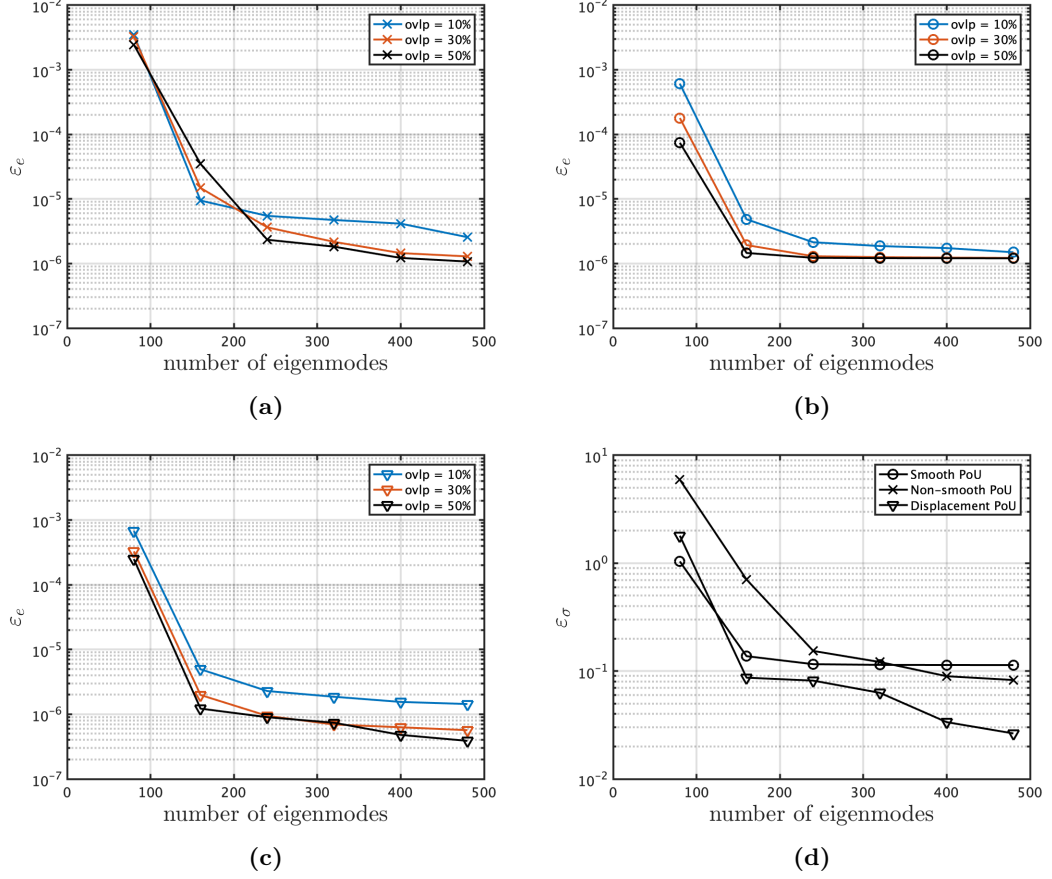
where  $\mathbf{u}_e = \tilde{\mathbf{u}}_h - \mathbf{R}_H^T \tilde{\mathbf{u}}_H$  is the difference between the directly obtained fine solution and fine solution recovered from the coarse solve. Similarly, we compute stress error as

$$\varepsilon_\sigma = \log_{10} \left( \frac{\|\sigma_{v,h} - \sigma_{v,H}\|}{\|\sigma_{v,h}\|} \right) \quad (5.14)$$

where subscript  $v$  represents Von Mises stress and  $h$  and  $H$  represent fine scale and coarse scale solutions respectively. We note that the extension operator is globally applied to demonstrate good agreement of the coarse and fine solutions. This operation would normally not be done since its cost is of the same order as an efficient iterative solver for finding  $\tilde{\mathbf{u}}_h$  itself. In practice, it often suffices to recover the fine scale solution in localized regions of interest. With the problem setup, we now discuss the experiment in more detail.

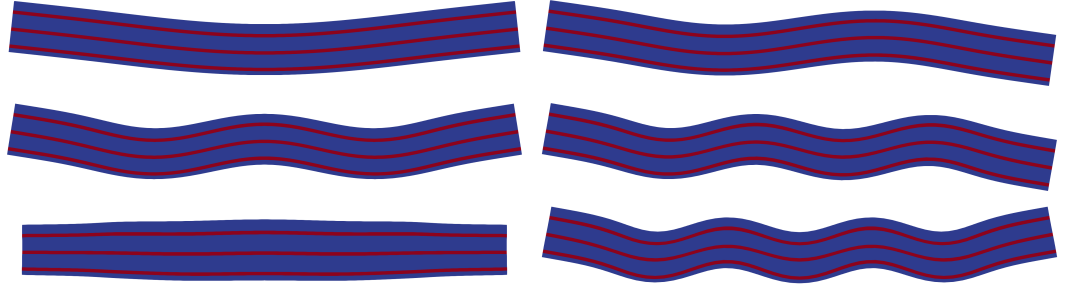
Convergence of multiscale methods in a domain decomposition setting is strongly influenced by the way subdomains interact with their neighbours. This interaction is encoded in the partition of unity operator, the optimal choice for which is currently unknown. In these experiments, we explore the difference between a piecewise constant, and two piecewise linear PoUs. We refer to them as non-smooth, smooth and uPoU,

respectively. Figure 5-8 shows the error convergence for the composite beam test for all three PoUs for a range of overlaps.



**Figure 5-8:** Sensitivity of *GenEO* multiscale method to 3 PoUs for the problem shown in Fig. 5-7(top). (a) Non-smooth PoU (b) Smooth PoU and (c) uPoU. In (d) we present the Von Mises stress error in  $L_2$  norm relative to fine scale solution, assumed to be the truth. The stress error curves correspond to 50% overlap for all three PoUs and are identified by their markers.

For a floating subdomain ( $\Omega_6$ ) i.e. one that does not share any degrees of freedom with the domain boundaries, the first 6 eigenmodes (excluding rigid body modes and using the smooth PoU) are pictured in Fig. 5-9 to provide a visual idea of the deformed shapes when stiffening inclusions are present. We note that the treatment of rigid body modes is non trivial in the sense that constrained subdomains sharing nodes with Dirichlet boundaries produce no rigid body modes from the local eigensolve. Translations and rotations must be added explicitly. The eigenmodes in Fig. 5-9 help illustrate the sensitivity of the method to the treatment of overlaps. That is to say, the neighbours of  $\Omega_6$  must deform in ways that are compatible with these modes. Therefore, the deformation contributed to the overlapped region of  $\Omega_6$  by its neighbours must be



**Figure 5-9:** First 6 eigenmodes in 2D elasticity excluding the zero energy rigid body modes. The deformations are shown for a subdomain that does not share any nodes with the domain boundary i.e. floating subdomain marked by a white mesh overlaid on Fig. 5-7(top). Red cells represent softer interply regions of a laminate and blue cells show carbon fibre layers.

shared amongst them in some proportion, the optimum of which is unknown. So, we assume that the contributions must be equally shared between neighbours. This gives us the non-smooth PoU as shown by the dotted line in Fig. 5-5. A major advantage of the non-smooth or piecewise constant PoU is its simple implementation however, it is relatively slow to converge to a fixed accuracy i.e. several modes per subdomain are required. On the other hand, the smooth or piecewise linear PoU is more difficult to implement but converges much more rapidly as shown in Fig. 5-8. Therefore, for a given accuracy,  $\dim(V_h)$  is much smaller for the smooth PoU. However, uPoU outperforms the other two PoUs but is more expensive to construct since it requires a solve for every pair or overlapped subdomains. Consequently, it stores information about the elastic field in the overlaps making it the most accurate out of the three PoUs tested here. The difference in accuracy is relatively small when considering errors in energy norm or even the displacement solution. But, error in the stress solution (see Fig. 5-8(d)) shows a greater difference since element strains converge at a slower rate than displacements.

These plots also show the error reduction based on the amount overlap between two subdomains. For simplicity, consider a 1D case such that any subdomain can have a maximum of two neighbours. Intuition suggests that increasing the overlap must improve the solution since the eigenvectors obtained better represent deformations over a larger region of a subdomain and its neighbours. Mathematically speaking, the matrix  $\mathbf{A}_{\Omega_j^\circ}$  on the right hand side of Eq. (3.13) is non-zero only in dofs indexed by overlapping nodes although it is equal in size to  $\mathbf{A}_{\Omega_j}$ . Therefore, it is rank deficient and the maximum number of real eigenvalues is bounded above by  $\text{rank}(\mathbf{A}_{\Omega_j^\circ})$ . A greater overlap admits more eigenvalues which improves accuracy of the coarse model however, enriching the coarse space beyond a certain limit, the increase in cost is disproportionate

---

to the error reduction. This is confirmed by the plots in Fig. 5-8. However, one should bear in mind that the overlap must not exceed 50% of a subdomain's length in a one dimensional problem so that it does not interact with a subdomain that is not its immediate neighbour. For this reason, the plots are terminated at 50% overlap.

In all three cases, the finest multiscale model has 480 dofs, reducing the model order by a factor of 168. The global accuracy achieved in the energy norm with 480 multiscale dofs, is rarely demanded by users within a stochastic framework. Depending on application, it often suffices to approximate a solution within 1% error tolerance. Based on these tests, we achieve that for multiscale models that are up to a 1000 times smaller than the fine model. However, solutions can be improved locally in the regions of interest. To demonstrate this, we now consider the stress solutions for the composite beam.

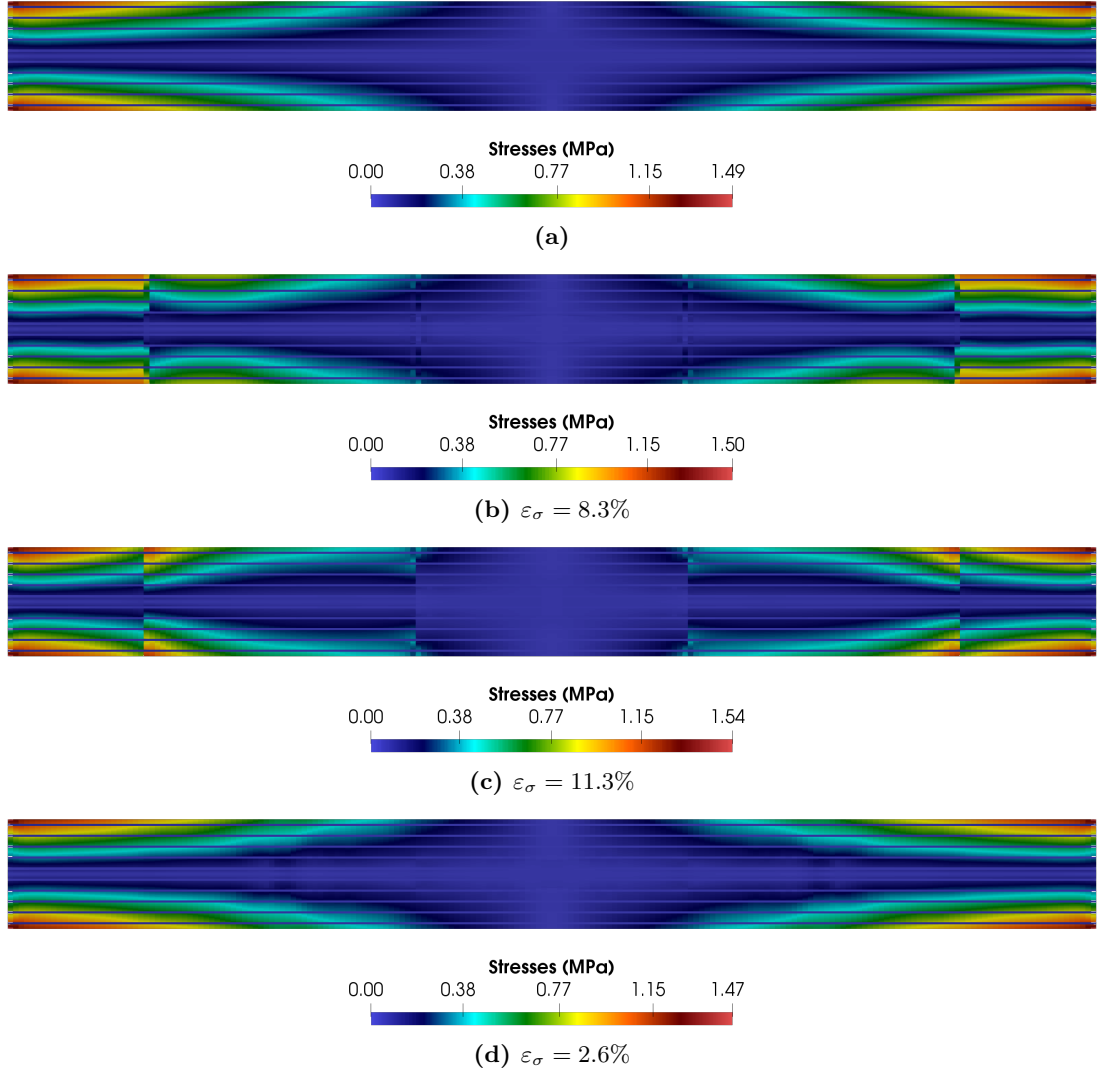
We begin by computing the Von Mises stress for displacement solutions presented in Fig. 5-10.

$$\sigma_v = \sqrt{\sigma_{11}^2 - \sigma_{11}\sigma_{22} + \sigma_{22}^2 + 3\sigma_{12}^2} \quad (5.15)$$

Consider the non-smooth PoU from Fig. 5-5 (dotted line). In this case, the mismatch in the deformations of  $\Omega_j$  (nodes =  $[1, 2, \dots, 7]$ ) and its neighbour to the right (nodes =  $[3, 4, \dots, 9]$ ) is accommodated in elements 3 and 6 since the elements in between (elements 4 and 5) contain the average solution of both subdomains. So, there is a displacement jump in elements at subdomain boundaries. As a result, we recover a piecewise constant strain over the entire domain with errors concentrated on the edges of overlaps. This phenomenon, generalized to 2D, can be seen in Fig. 5-10(b) for stresses obtained with the non-smooth PoU. We would expect this phenomenon to be eliminated when using a smooth PoU instead but, as shown in Fig. 5-10(c), we observe similar discontinuity. In fact the error  $\varepsilon_\sigma$  increases for the smooth PoU. This is because the smooth PoU error convergence stagnates. However, the subdomain boundaries are much better approximated. This is evidenced by a more significant improvement in the solution after stress recovery. The stagnation occurs due to the purely geometric nature of the smooth and non-smooth PoUs since they contain no information about the mechanics of the material. The error is analogous to the error that would result from finding a displacement solution by simply interpolating from boundary conditions over the domain instead of an FE based displacement solution.

On the other hand, the uPoU produces a stress solution Fig. 5-10 (d) that takes into account the mechanics of the overlaps. By inspection, we can conclude that the uPoU offers a far more superior performance for the problem at hand. For this reason, we use the uPoU for the rest of the simulations that follow in this chapter.

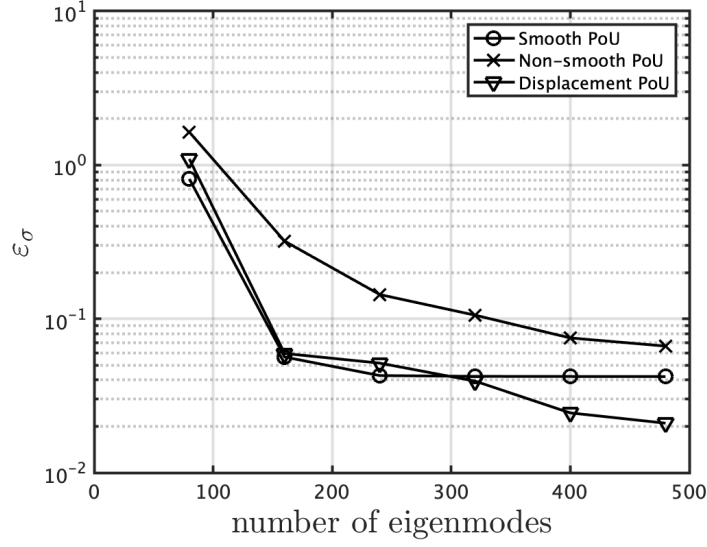
Treating the fine scale stress solution as the truth, the error in the uPoU is visu-



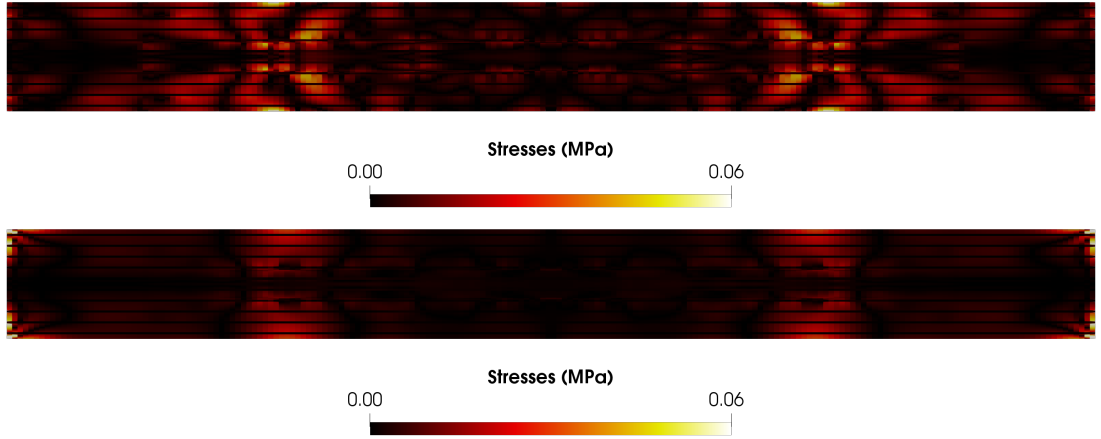
**Figure 5-10:** Von Mises stress plots for the layered composite beam shown in Fig. 5-7. (a) Fine scale solution with 80,802 dofs (b) Non-smooth PoU, (c) Smooth PoU and (d) uPoU. (b), (c) and (d) were computed with 480 dofs. The solution in (a) is treated as the truth relative to which the error  $\varepsilon_\sigma$  is computed.

alized in Fig. 5-12 (top). Peak errors are observed at subdomain boundaries. These concentrations are artefacts of the independent eigensolves due to Dirichlet boundary conditions being applied on all subdomain boundaries except global Neumann boundaries. In order to correct for this mismatch, a post processing technique called stress recovery is used.

It is common knowledge that convergence of a finite element displacement solution is better than convergence of its derivative (strain). As a result, stress solutions are less accurate than displacement based solutions. In a domain decomposition setting,



**Figure 5-11:** Stresses after patch recovery at 50% overlap for all three PoUs.



**Figure 5-12:** Von Mises stress error relative to the fine scale solution pictured in Fig. 5-10(top). Both plots shown here are solved using uPoU. (Top) Before stress recovery,  $L_2$  norm error  $\approx 2.6\%$ . (Bottom) Post stress recovery,  $L_2$  norm error =  $2.1\%$ .

this is remedied with a post process called stress or patch recovery, the key steps of which are as follows:

1. Project the coarse solution onto a region of interest i.e. a subdomain giving  $\tilde{\mathbf{u}}_h(\Omega_j)$ . This operation requires nearest neighbour communication due to overlapped regions. Say, the neighbours of  $\Omega_j$  are denoted as  $\Omega_{j,1}, \Omega_{j,2}, \dots, \Omega_{j,k}$ , then



---

we can write this projection as

$$\tilde{\mathbf{u}}_h(\Omega_j) = \mathbf{R}_H^T(\Omega_j)\tilde{\mathbf{u}}_H(\Omega_j) + \sum_{i=1}^k \mathbf{R}_H^T(\Omega_{j,i}^\circ)\tilde{\mathbf{u}}_H(\Omega_{j,i}^\circ) \quad (5.16)$$

where the superscript  $\circ$  denotes overlaps.

2. Assemble local stiffness matrix on  $\Omega_j$ .
3. Apply boundary conditions. Let us denote a processor or subdomain boundary as  $\partial\Omega_j = \overline{\text{dof}(\Omega_j)} \setminus \text{dof}(\Omega_j)$ . All processor boundaries are treated as Dirichlet boundaries except for those that coincide with global non-Dirichlet boundaries. We then apply the displacement boundary conditions  $\tilde{\mathbf{u}}_h(\partial\Omega_j)$  on  $\partial\Omega_j$  and solve to obtain the corrected solution  $\hat{\mathbf{u}}_h(\Omega_j)$ .
4. Project the local solution  $\hat{\mathbf{u}}_h(\Omega_j)$  back to  $\Omega$  by computing

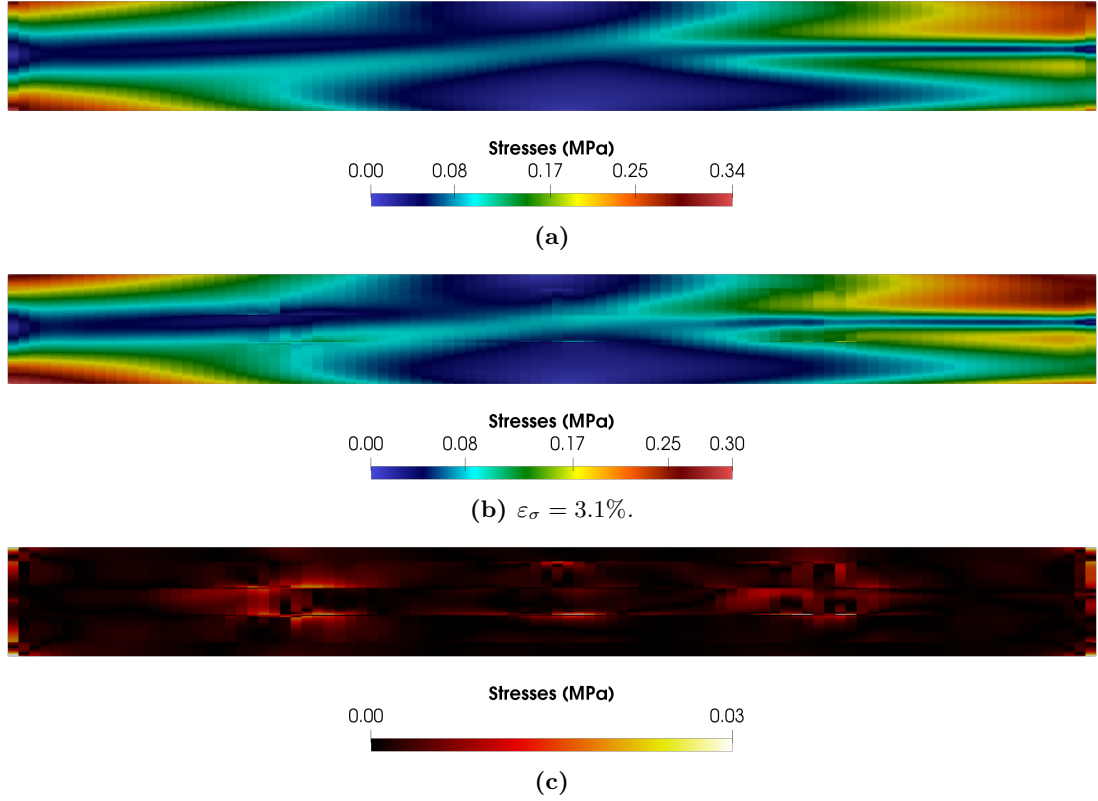
$$\hat{\mathbf{u}}_h(\Omega) = \sum_{j=1}^N \mathbf{R}_H^T(\Omega_j)\hat{\mathbf{u}}_h(\Omega_j) \quad (5.17)$$

In real world applications, the coarse solution is not usually projected back to the fine scale since the cost is comparable to that of solving the problem iteratively. Therefore, fine scale solutions are only recovered in regions or subdomains of interest. To enable a comparison over the entire domain we do however, recover the fine scale solution everywhere. Using this strategy the error is reduced by an order of magnitude as shown in Fig. 5-12 (bottom). Here, stress recovery was used to improve the  $L_2$  norm error of Von Mises stress over the entire domain from  $\approx 2.6\%$  to  $2.1\%$ . The improvement is small because uPoU is sufficiently accurate without stress recovery.

It is important to note that none of the three PoU operators tested here are the optimum choice for this problem. In fact, the optimal PoU design is a subject of future work.

Now, we test the method with a random heterogeneous material modelled as a Gaussian random field. One distinctive difference between composites and random heterogeneous materials is the lack of scale separation and it is with respect to this that we wish to demonstrate the performance of **GenEO** as multiscale method. The Gaussian random field was created by computing 9 lowest Karhunen-Lo  ve eigenpairs for the Mat  rn covariance function with a correlation length  $\lambda = 1.2$ , smoothness parameter  $\nu = 0.5$  and variance  $\sigma = 1.3$ .

Random heterogeneous materials show better convergence than composites. The reason for better convergence relative to the composite beam is a smoother eigenspec-

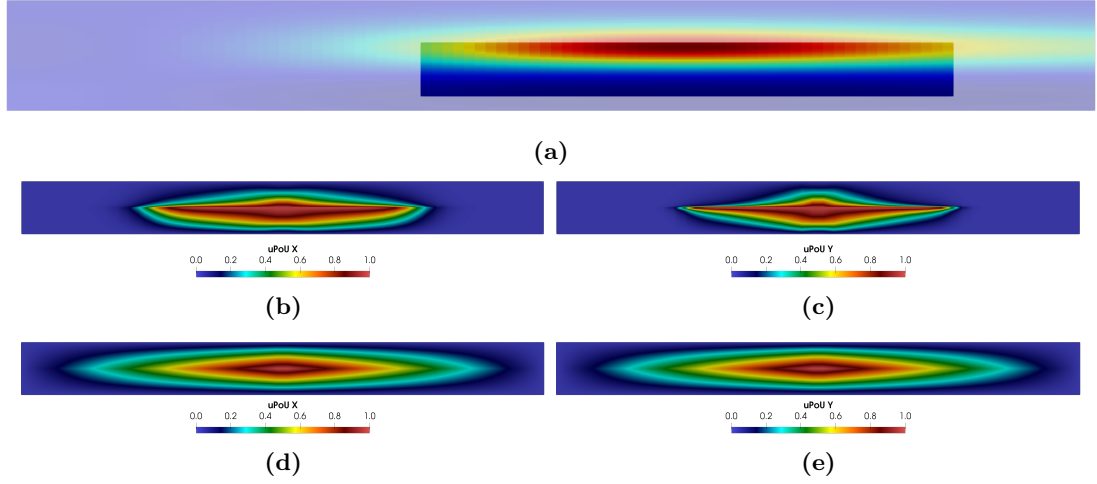


**Figure 5-13:** (a) Fine scale solution for heterogeneous material shown in Fig. 5-7. (b) Multiscale (640 dofs) solution projected on the fine grid. (c) Difference between fine and multiscale solutions.

trum in the absence of scale separation, see Fig. 5-2. Due to this, for a given accuracy, multiscale models for heterogeneous materials can be much smaller than composites. In this way we show that **GenEO** as a multiscale method is not restricted to composites.

For stress analysis, we use only the smooth PoU along with stress recovery to demonstrate the performance of this multiscale method with respect to random heterogeneous materials. The stress field in Fig. 5-13 (top) shows the fine scale Von Mises stress for the problem given in Fig. 5-7 (bottom) using 15 eigenvectors per subdomain with 50% overlap. The middle plot shows the multiscale solution post stress recovery and the bottom plot shows the difference between the two solutions.

Problems smaller than  $10^5$  dofs can efficiently be solved on a desktop computer however, beyond that stiffness matrices and its factors cannot be stored in memory. The main advantage of this method is that it only requires nearest neighbour communication and only ever needs to assemble stiffness matrices on individual subdomains at most. This makes it possible to solve much larger problems on a desktop computer. Crucially,



**Figure 5-14:** Visualizing  $uPoU$  on  $\Omega_7$  of the heterogeneous beam with 50% overlap.

it can be readily implemented in parallel.

### GenEO coarse space as a multiscale method in 3D

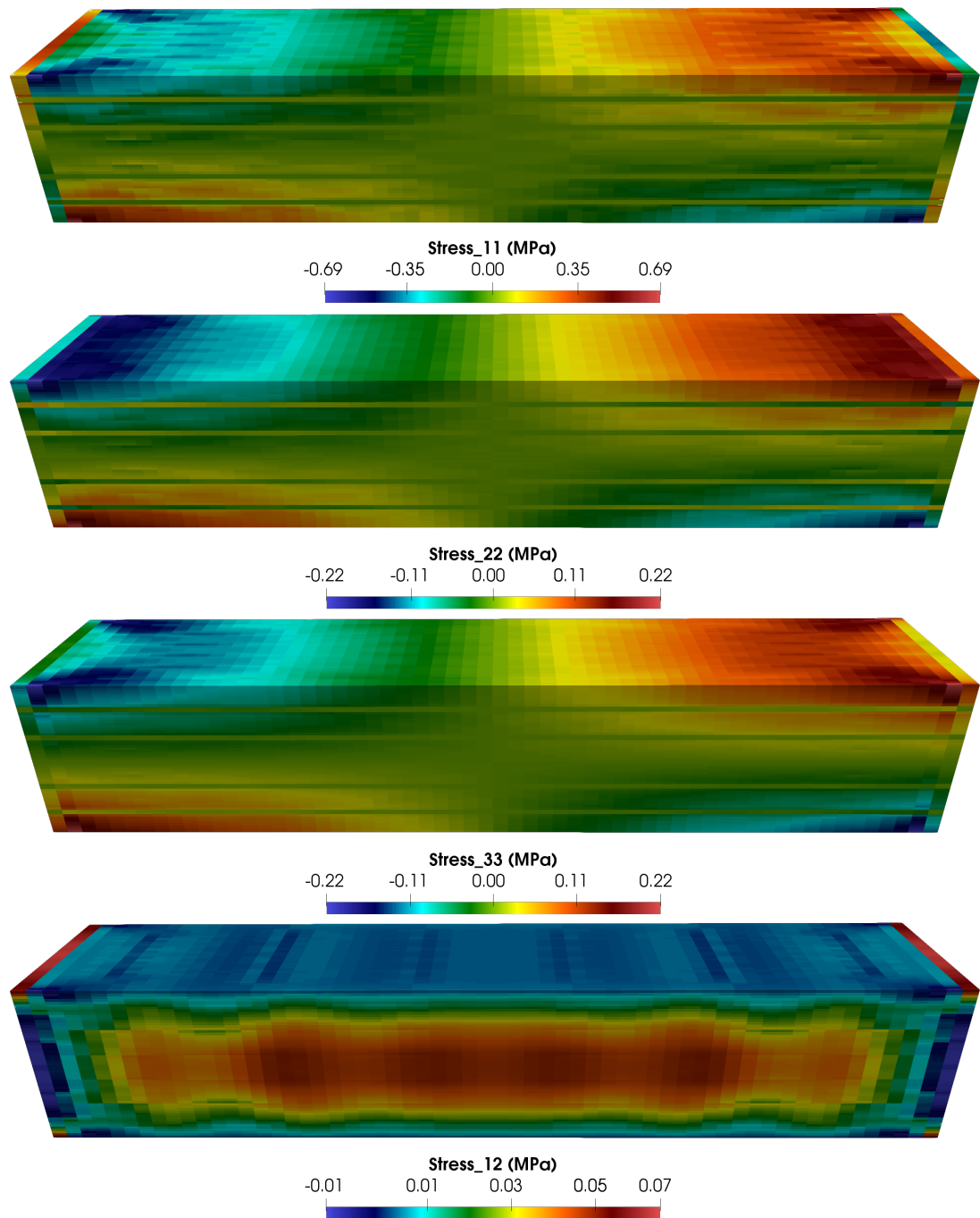
As we are primarily interested in composites, we now demonstrate the performance for a three-dimensional composite beam with 6 composite plies 0.25mm thick orientated at  $0^\circ$  along the x-axis. They are separated by resin rich interply regions 0.0625mm thick. The mechanical properties for both composite plies and resin are given in Table 5.1, however, in 3D  $\nu = 0.25$ . The dimensions of the beam in x, y and z directions are 10mm, 1.8mm and 2mm respectively. The domain is discretized with 60 elements in each direction giving approximately 700,000 dofs on the fine scale. The fine scale problem is decomposed into 216 ( $N = 6$  in each direction) regular subdomains with 50% overlap. On the coarse scale, we compute  $m = 25$  eigenmodes per subdomain resulting in 5400 degrees of freedom on the coarse scale. The multiscale model is therefore, almost 130 times smaller than the fine model. The stress plots for this problem are given in Fig. 5-15.

Now, we add a wrinkle to the composite beam simulated above. The wrinkle is defined as

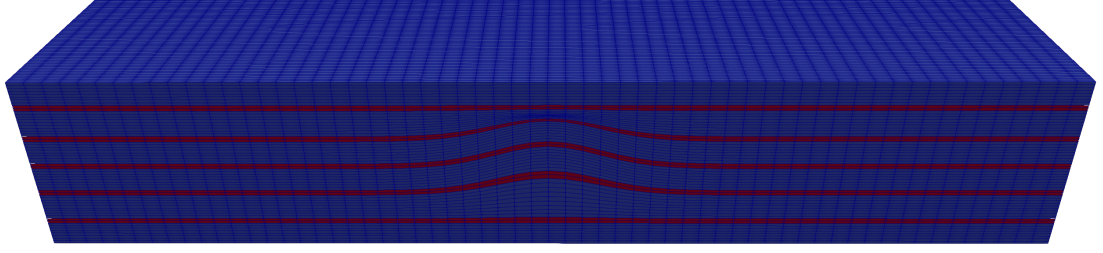
$$y = \frac{g(x, y)}{4} \operatorname{sech}^2 \left( x - \frac{L_x}{2} \right) \quad (5.18)$$

where  $g(x, y)$  is the following envelope function

$$g(x, y) = \exp \left( - \left( x - \frac{L_x}{2} \right)^2 - 20 \left( y - \frac{L_y}{2} \right)^4 \right) \quad (5.19)$$



**Figure 5-15:** *Stress plots for 3D composite beam using the smooth PoU.*



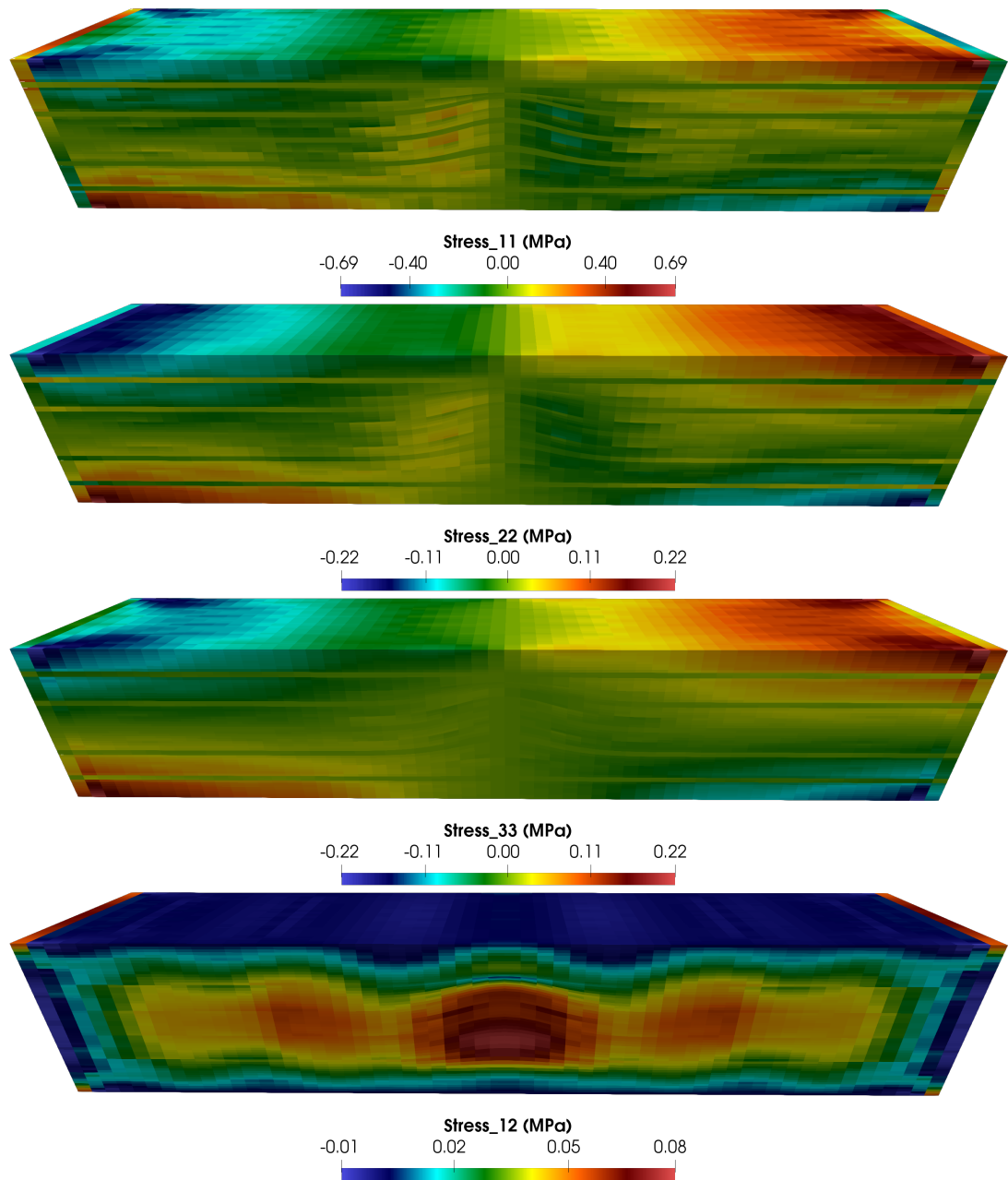
**Figure 5-16:** *Wrinkle defined in Eq. (5.18) embedded in a laminate.*

to contain the wrinkle entirely within the laminate except in the  $z$ -direction where it is prismatic as shown in Fig. 5-16. The corresponding stress plots are shown in Figs. 5-17 and 5-18 to show a comparison between the smooth PoU and uPoU, respectively.

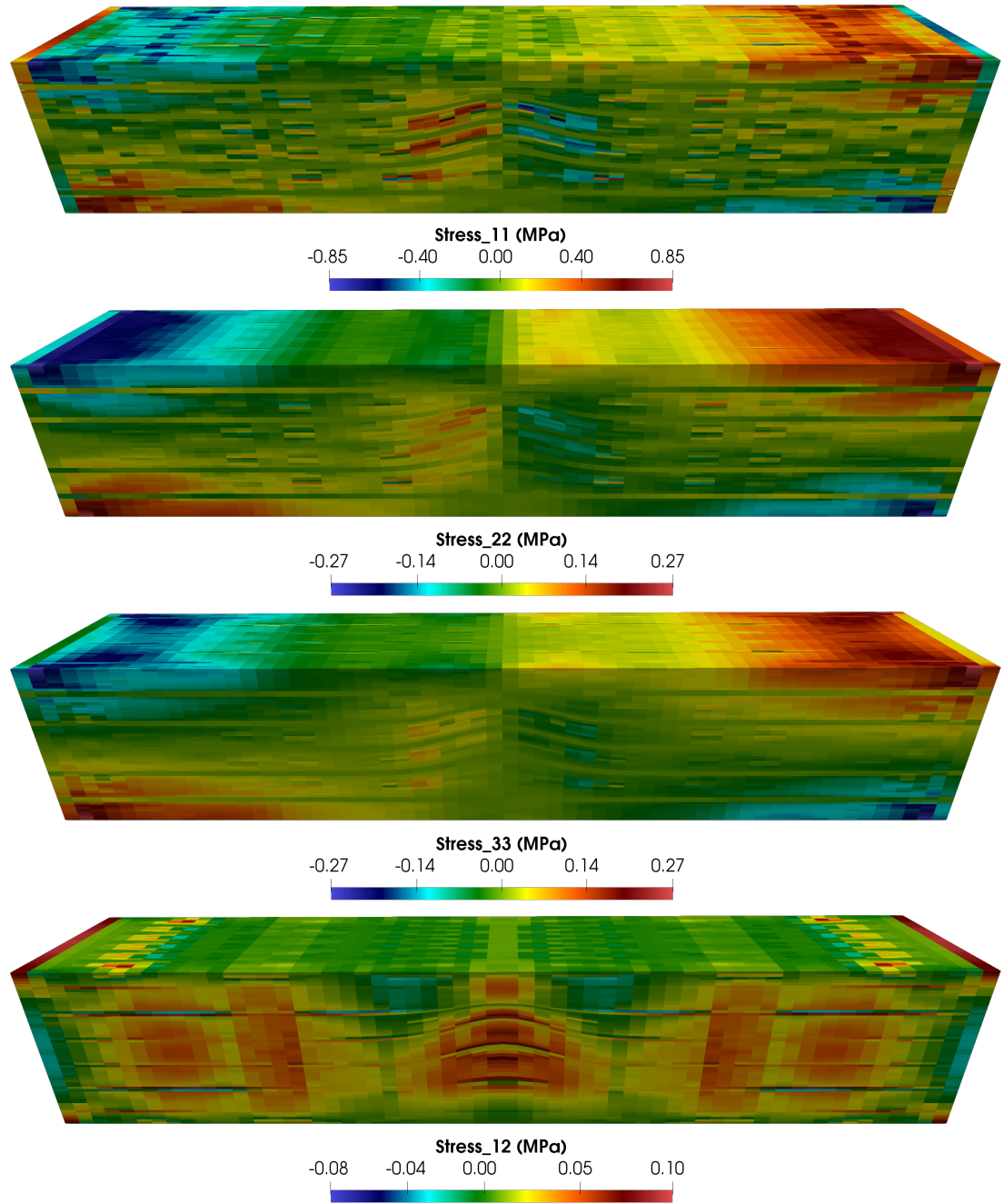
## 5.4 Concluding remarks

In this chapter we have demonstrated the performance of **GenEO** coarse space as a multiscale method for both two and three dimensional problems. The bespoke multiscale bases spaces are built using local spectral information stitched together with a partition of unity operator. The choice of this operator has a significant bearing on the accuracy of the solution, the optimum choice for which is currently unknown. However, from our numerical experiments we can conclude that the smooth PoU demonstrates a superior performance relative to the piecewise constant PoU but both are outperformed by uPoU.

This method permits the construction of coarse models with a fraction of the degrees of freedom of their fine scale representation, which still produce solutions within 1% of the fine solution. For the examples presented, particularly the composite case, the models are reduced by a factor greater than 550 in 2D and 130 in 3D and yet achieve a solution within 1% of the fine scale solution. Even greater model order reduction was achieved in the heterogeneous case while maintaining similar accuracy. For composite or heterogeneous cases, where macroscale modes of deformation can be unclear, as seen in Fig. 5-9, use of local eigenfunctions in the customized coarse space construction seems natural. The most important detail about the customized coarse space built in this chapter is that it becomes available from two-level additive Schwarz implemented in **dune-composites** and does not need to be recomputed. The importance of this availability will become clear in the next chapter.



**Figure 5-17:** *Stress plots for 3D composite beam containing a wrinkle solved using the smooth PoU.*



**Figure 5-18:** *Stress plots for 3D composite beam containing a wrinkle calculated with uPoU.*





---

---

## CHAPTER 6

---

# DATA-DRIVEN MULTISCALE MODELS FOR HIGH DIMENSIONAL MCMC

### 6.1 Introduction

The traditional approach to quantify uncertainty in both, materials and models involves running several experiments to heuristically characterize the spread of a quantity of interest (QoI) so that an expected QoI can be estimated with confidence proportional to the amount of data available. Naturally, sets of test data get smaller as one moves up the test pyramid (see Fig. 2-1). Consequently, confidence in outputs is lowered and wider safety margins have to be applied. This is one reason for the drive to construct virtual twins of the test pyramid that permit a comprehensive grasp of uncertainty leading to more efficient designs - the third and final challenge this thesis set out in Chapter 1 which we are now in a position to address.

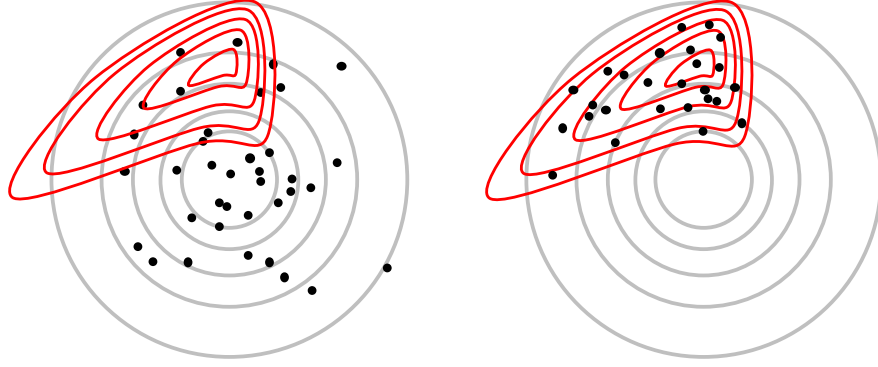
The question we now ask is that given some observed deformation, what is the underlying elastic field of that component? We cast this as an inverse problem in a Bayesian setting. In order to conceptualize the inverse problem, let us consider a rectangular composite plate tested under compression. We measure the strain or deflection at a set of predetermined points on its surface during the experiment. The goal then is to divine a Gaussian field with parameters sampled at random from the prior so that the model outputs at those predetermined points agree with experimental

---

measurements. If the model and experimental outputs agree, within some tolerance, we can conclude that the elastic field of the composite is well approximated by the current instance of the random field.

Random parameters or fields are sampled using a MCMC method however, the scale of calculations presents a sizeable challenge. To achieve proper mixing of a chain through the posterior, the theoretically optimum acceptance ratio is 0.23 [82]. Therefore, 77% of the samples drawn will be rejected. The accept-reject step for every sample requires a solution to the forward model used to define the likelihood function, a computationally intensive undertaking. For models with more than a few million unknowns, this cost is unaffordably expensive. One way to relieve the computational burden is by replacing the model with a cheaper substitute. An alternate way is to draw samples in a manner that maximizes acceptance ratio without compromising mixing properties of the chain. The former is addressed in Chapter 5 with the help of which we pursue the latter strategy in this chapter.

UQ literature for composites is sparse in general so we draw on ideas mostly from the fields of geothermal reservoir modelling [173], subsurface flow [174], remote sensing [175] and other similar scalar valued problems. The idea of using cheap replacements of forward models has been previously explored but limited to low dimensional scalar valued problems [69]. Surrogates have been applied to inverse problems in different ways. For example, [70, 176] use generalized polynomial chaos, [177] employed Gaussian process regression and [178–181] used projection-based reductions in model order. We are interested in the projection-based approach that reduces model order by solving a projection of the full model in a subspace. Our innovation lies in realizing the suitability of the **GenEO** spectral coarse space for model order reduction. For our choice of reduction method, constructing the **GenEO** coarse space requires solving the full model at representative samples from the posterior which are not known *a priori*. The span of each of their coarse spaces orthogonalized in some way, defines an appropriate subspace applicable all over the posterior. Crucially, the quality of the subspace is dictated by the choice of representative samples. We extend the delayed acceptance approach designed by Christen and Fox [88], further developed in [69] to quantify uncertainty in materials and models. We adapt innovations by Cui *et al.* [69] whereby the coarse space construction process is embedded into an MCMC algorithm that simultaneously explores the posterior and selects samples for enriching the coarse space. The enrichment sample selection is based on the idea that a coarse model for any point in the posterior can be expected to be a good substitute for all samples near that point. Therefore, a metric is defined by which we dynamically monitor the quality of the coarse space such that it can be enriched for any region of the posterior that has not yet been visited by



**Figure 6-1:** Representation of Bayesian approach in a simplified 2D parameter space for some likelihood function. (Left) Sampling from isotropic Gaussian prior  $\pi_0(\xi)$ . (Right) MCMC sampling from posterior  $\pi(\xi|\mathcal{D}_{obs})$ .

the Markov chain. Compared to the classical approach whereby subspaces are built offline, the adaptive approach is computationally more efficient as the coarse space built this way is tailored to a particular posterior thus requiring a smaller basis to retain the accuracy within a concentrated region of the parameter space. Representative samples in the offline approach are usually drawn from the prior and so need to be much richer to compete with online enrichment.

An added benefit of this MCMC strategy is that we obtain a coarse space tailored to a particular model. We use this to run a coarse MCMC process and provide a comparison of results showing how well coarse MCMC approximates the actual distributions. This is useful because we can estimate an unknown distribution at a much lower cost using a tailored coarse space for model order reduction thus bypassing full scale evaluations entirely.

## 6.2 Standard Markov Chain Monte Carlo

Standard MCMC process for sampling posteriors using Metropolis-Hastings random walk is described in Section 4.2.2. In a Markov chain, samples close to one another

---

### Algorithm 1 Metropolis-Hastings algorithm

---

```

1 generate the first proposal  $\xi^0$  while
2  $0 \leq k \leq n$  given  $\xi^k$ , generate a proposal
3  $\xi'$  from given proposal distribution
4  $q(\xi'|\xi^k)$  Accept
5  $\xi'$  with the probability
6  $\alpha_\xi(\xi^k, \xi') = \min \left\{ 1, \frac{\pi(\xi')q(\xi^k|\xi')}{\pi(\xi^k)q(\xi'|\xi^k)} \right\}$   $k = k + 1$  end % end while
```

---

---

are strongly correlated but we require independent samples to estimate expectations of functions. By estimating the integrated autocorrelation time for each component of a chain  $\Lambda_i$ , we can approximate a subsampling interval  $\Lambda = \max(\Lambda_i)$  for which the samples are independent. The number of independent samples in a chain is called its effective sample size or ESS. Computational efficiency of an MCMC algorithm is defined by ESS per unit CPU time. The efficiency of standard MCMC is driven by two factors;

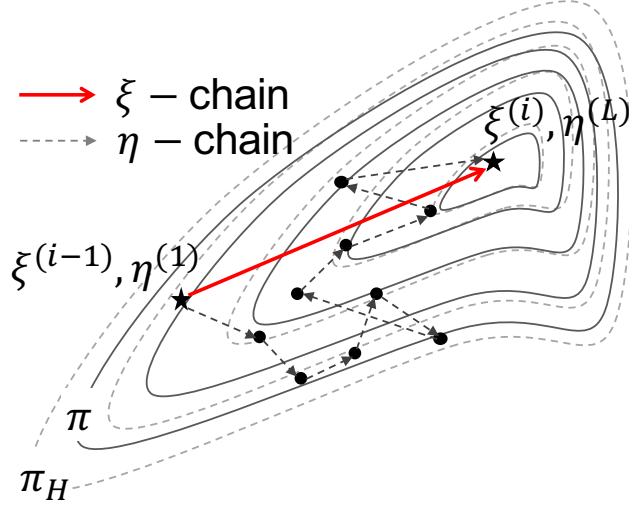
1. Cost of the forward model  $F(\boldsymbol{\xi})$ : The number of MCMC samples drawn for a given CPU budget is directly proportional to the cost of solving the forward model. If the cost of evaluating the posterior probability density in Eq. (4.7) can be reduced with a faster evaluation of  $F(\boldsymbol{\xi})$ , we can improve computational efficiency. We explore this idea further in Section 5.2.1.
2. The autocorrelation time of a chain: For a finite length Markov chain, ESS is inversely proportional to  $\Lambda$  which can be reduced, for instance, by exploiting the gradient information of the posterior on-the-fly [182] thus making more effective proposals. Here too it is beneficial to have a quicker solve of  $F(\boldsymbol{\xi})$  from which gradients can be computed at a reduced cost. However, in this contribution we choose to reduce  $\Lambda$  by using the delayed acceptance MCMC methodology [88] for reasons described in Section 6.3.

### 6.3 Multiscale Markov Chain Monte Carlo

In Section 6.2 we identified two factors that improve the efficiency of MCMC - faster forward solves and shorter autocorrelation time. We begin by exploring the idea of reducing  $F(\boldsymbol{\xi})$  solve time which we later combine with delayed acceptance MCMC to reduce autocorrelation time of a chain.

In Chapter 3, we claimed that the **GenEO** preconditioner produces a well-conditioned system accelerating convergence of Krylov solvers, particularly for large anisotropic systems. The success of the two-level additive Schwarz method depends on the quality of the **GenEO** coarse space. In Chapter 5 we observed that a good coarse space also doubles as an effective reduced order or multiscale representation of the full model. Here, we use the multiscale model to reduce both, forward solve time and autocorrelation time.

To help understand the algorithm we define two Markov chains - the primary chain called  $\xi$ -chain and the secondary or subchain denoted the  $\eta$ -chain. The  $\eta$ -chain only exists between two neighbouring samples of the  $\xi$ -chain,  $\boldsymbol{\xi}^{i-1}$  and  $\boldsymbol{\xi}^i$  such that  $\boldsymbol{\eta}^1 = \boldsymbol{\xi}^{i-1}$ , see Fig. 6-2. The key idea is that the  $\xi$ -chain computes expensive fine scale solutions



**Figure 6-2:** The solid contours denote some (true) posterior distribution  $\pi$  and the dashed contours mark a coarse approximation of it  $\pi_H$ . Generally these are unknown quantities but we draw them here to demonstrate the delayed acceptance MCMC scheme. The stars denote two consecutive samples in the  $\xi$ -chain (red arrow) pulled from  $\pi$  while the black dots show the subchain or  $\eta$ -chain (dashed black arrows) sampling from  $\pi_H$  between  $\xi^{i-1}$  and  $\xi^i$ .

---

**Algorithm 2** Delayed acceptance MCMC

---

- 1 pass a proposal  $\xi^{k-1}$  from  $\xi$ -chain to  $\eta$ -chain such that  $\eta^0 = \xi^{k-1}$
  - 2 for  $j = 0:L-1$  Given  $\eta^j$ , generate new proposal on  $\eta$ -chain  $\eta'$  from given proposal distribution  $q(\eta'|\eta^j)$
  - 3 accept with probability  $\alpha_\eta(\eta', \eta^{j-1}) = \min \left\{ 1, \frac{\pi_m(\eta')q(\eta^j|\eta')}{\pi_m(\eta^j)q(\eta'|\eta^j)} \right\}$  end hand  $\eta^j$  over to  $\xi$ -chain such that  $\xi' = \eta^j$  accept with probability  $\alpha_\xi(\xi', \xi^{k-1}) = \min \left\{ 1, \frac{\pi(\xi')\pi_m(\xi^{k-1})}{\pi(\xi^{k-1})\pi_m(\xi')} \right\}$   $\xi^k = \xi'$  with probability  $\alpha_\xi$  and  $\xi^k = \xi^{k-1}$  with probability  $1 - \alpha_\xi$
-

---

to the forward model  $F(\boldsymbol{\xi})$  thus exploring the full posterior distribution. The  $\eta$ -chain evaluates coarse approximations  $F_H(\boldsymbol{\eta})$ , enabling it to construct a computationally cheaper posterior approximation. Here,  $F_H(\boldsymbol{\eta}) : \tilde{\mathbf{u}}_H \rightarrow \mathbf{d}_{obs}$ .

Given the approximate forward model  $F_H(\boldsymbol{\xi})$  based on the customized coarse space, we must redefine the misfit function and approximate posterior distribution. The approximate misfit is as before with the forward model replaced by its approximation

$$\Phi_H(\boldsymbol{\eta}) = \frac{1}{2} \left\| \Sigma^{-\frac{1}{2}} (F_H(\boldsymbol{\eta}) - \mathcal{D}_{obs}) \right\|_2 \quad (6.1)$$

and the resulting posterior approximation has the form

$$\pi_H(\boldsymbol{\eta} | \mathcal{D}_{obs}) = \frac{1}{Z_H} \exp(-\Phi_H(\boldsymbol{\eta})) \pi_0(\boldsymbol{\eta}) \quad (6.2)$$

and once again,  $Z_H$  need not be computed.

Now, let us assume we have a sufficiently rich coarse space constructed using the method described in Eq. (3.13), equation Eq. (5.7) can be solved to approximate the displacement field with some error threshold  $\epsilon$ . For the  $i$ -th iteration of MCMC sampling, we exploit the shorter run time of the  $\eta$ -chain to explore the posterior using the pCN random walk whereby a proposal has the form given in Eq. (2.7) The proposal  $\boldsymbol{\eta}'$  is accepted for the next sample  $\boldsymbol{\eta}^k$  with the following probability

$$\alpha_\eta(\boldsymbol{\eta}', \boldsymbol{\eta}^{k-1}) = \min \left\{ 1, \exp(\Phi_H(\boldsymbol{\eta}^{k-1}) - \Phi_H(\boldsymbol{\eta}')) \right\} \quad (6.3)$$

If the current sample is rejected then  $\boldsymbol{\eta}^k = \boldsymbol{\eta}^{k-1}$ . This chain is run for a predetermined number of steps  $L$ , chosen such that its initial and final states are uncorrelated. The last step of the subchain is treated as a new proposal  $\boldsymbol{\xi}' = \boldsymbol{\eta}^L$  of the  $\xi$ -chain and is passed through the accept/reject step Eq. (6.4) using the ratio of full posterior density to the approximate posterior density to ensure the  $\xi$ -chain samples from the full posterior.

$$\alpha_\xi(\boldsymbol{\xi}', \boldsymbol{\xi}^{i-1}) = \min \left\{ \frac{\pi(\boldsymbol{\xi}' | \mathcal{D}_{obs}) \pi_H(\boldsymbol{\xi}^{i-1} | \mathcal{D}_{obs})}{\pi(\boldsymbol{\xi}^{i-1} | \mathcal{D}_{obs}) \pi_H(\boldsymbol{\xi}' | \mathcal{D}_{obs})} \right\} \quad (6.4)$$

Again, if  $\boldsymbol{\xi}'$  is rejected,  $\boldsymbol{\xi}^i = \boldsymbol{\xi}^{i-1}$ .

As we are only interested in the independent samples drawn from the full posterior, we wish to ensure that the  $\xi$ -chain has a minimal rejection rate i.e. acceptance ratio close to 1. One can imagine that if the subchain samples from a distribution which is a poor approximation of the full posterior, more samples are likely to be rejected by the  $\xi$ -chain reducing the effective sample size. Naturally, the accuracy of the approximate posterior is driven by the accuracy of the coarse model. The question then becomes,

---

how can we build a sufficiently accurate coarse model without prior knowledge of the posterior to maintain statistical efficiency?

To improve the statistical efficiency, we use adaptive reduced basis enrichment developed by Cui *et al.* [69] with a slight variation. After each accept/reject test on the  $\xi$ -chain, we compute the model error in the coarse solve by treating the  $F(\xi_i)$  as the truth.

$$t_H(\xi^i) = \left\| \Sigma_\epsilon^{-\frac{1}{2}} (F(\xi^i) - F_H(\xi^i)) \right\|_\infty \quad (6.5)$$

If this error exceeds a prescribed threshold  $\epsilon$ , we augment the coarse space with the current solution  $\tilde{\mathbf{u}}^{(i)}$  also called a *snapshot*. This can be interpreted as a new eigenmode, previously unseen. However, it is not orthonormal to the existing coarse space. So we must orthonormalize the coarse space. This is a non-trivial step that allows us to retain control of the maximum allowable error.

### 6.3.1 Multiscale model

Here we use the multiscale model we built in Chapter 5. We wish to clarify that this chapter is intended as a proof-of-concept rather than a thorough treatment of multiscale MCMC. So, we restrict our multiscale model in order to facilitate laying the groundwork for a fully parallel multiscale MCMC implementation within the `dune` framework which is part of the future work. In this chapter we implement a sequential version and discuss the extension to a fully parallel platform.

The first restriction is applied on the coarse space  $\mathbf{R}_H$ . We will consider only the zero energy modes per subdomain (Nicolaidis coarse space) as it affords us a simpler way of implementing Dirichlet boundary conditions. The non-zero energy modes per subdomain are replaced by local solutions which are essentially a combination of local eigenmodes. The multiscale model then learns the topology of the posterior and is capable of adapting the coarse space online. This offers significant advantages over the traditional offline coarse space as discussed in Section 2.5.

#### Coarse space learning and Gram-Schmidt orthonormalization

Suppose we have a reduced order model  $\mathbf{A}_H$  constructed from an initial coarse space  $\mathbf{R}_H$  that produces  $t_H(\xi^i) > \epsilon$ . This condition informs the algorithm that the coarse space is not rich enough and triggers the coarse space update process. The update process is designed to simply add a snapshot to  $\mathbf{R}_H$  but the implementation is non-trivial. At the start of the update process, the coarse space is augmented with the current snapshot to give  $\mathbf{R}'_H$ , the dimension of which exceeds  $\dim(\mathbf{R}_H)$  by 1. There are two important considerations here. First, the initial coarse space comprises zero

---

energy modes which are also included in the snapshot. To prevent double accounting, the zero energy modes are subtracted from the snapshot. Second, the snapshot must be orthonormal to the coarse space in the absence of zero energy modes. This is achieved via a Gram-Schmidt process. In this way, we develop a data-adjusted coarse space.

This strategy allows delayed acceptance MCMC to achieve two important objectives simultaneously; a) full posterior exploration and b) online model order reduction.

### 6.3.2 The algorithm

---

#### Algorithm 3 Delayed acceptance MCMC

---

```

1 generate the first proposal  $\xi^1$   $k = 2$  while  $k < N$ 
2  $\eta^1 = \xi^{k-1}$  compute
3  $F_H(\eta^1)$  % reduced order model
4 for  $i = 2 : L$  generate new proposal  $\eta'$ 
5 compute  $F_H(\eta')$  evaluate acceptance probability
6  $\alpha_\eta(\eta^{i-1}, \eta')$  if
7  $(\alpha_\eta > r)$  % r is a random number uniformly distribution
8 between  $[0, 1]$ 
9  $\eta^i = \eta'$  else
10  $\eta^i = \eta^{i-1}$  end % end if
11 end % end for
12 set  $\xi' = \eta^L$  evaluate
13  $F(\xi')$  evaluate acceptance probability
14  $\alpha_\xi(\xi^{k-1}, \xi')$  if
15  $(\alpha_\xi > r)$  % r is a random number uniformly distribution
16 between  $[0, 1]$ 
17  $\xi^k = \xi'$  else
18  $\xi^k = \xi^{k-1}$  end % end if
19 compute model error  $t_H$  if  $(t_H > \epsilon)$  update coarsespace end  $k = k + 1$  end % end while

```

---

## 6.4 Numerical experiments

Any factory manufactured composite structure is expected to contain some degree of out-of-plane waviness however, not all such perturbations are seen as wrinkles even though they have an effect on part performance. In addition, that part is also expected to have other defects or variations in its elastic properties across its domain. The variety of defects however, is too wide to be rigorously classified into wrinkles, voids and so on. Therefore, as a generalization step, we assume that all these variations can be modelled with finite dimensional Gaussian random fields. We note that a Gaussian random field may not be the optimum description of a laminate but as this is a demonstration of methodology, we do not require a thorough treatment of the elastic



---

field as long as it contains the salient features such as high contrast anisotropy and localized variations. Furthermore, the method presented in this chapter is not bound to Gaussian representations. If a finite dimensional parametric description of a field exists, it can be plugged into this method.

In order to demonstrate the concept proposed in this chapter, we consider a 2D elasticity problem. Let the domain be  $\Omega = [0, 1]^2$ , with vertical Dirichlet boundaries and horizontal Neumann boundaries. One Dirichlet boundary is fixed while the other is displaced by 1 in the x-direction. The domain is discretized using a structured mesh with constant mesh parameter  $h = 1/200$ . This produces a total of 40,000 elements with over 80,000 dofs. The elastic field over the domain is defined by a random Gaussian field, the weights of which constitute the unknowns to be determined. We consider a 100 dimensional field with 100 unknowns to demonstrate the proof of concept.

The random field is defined as

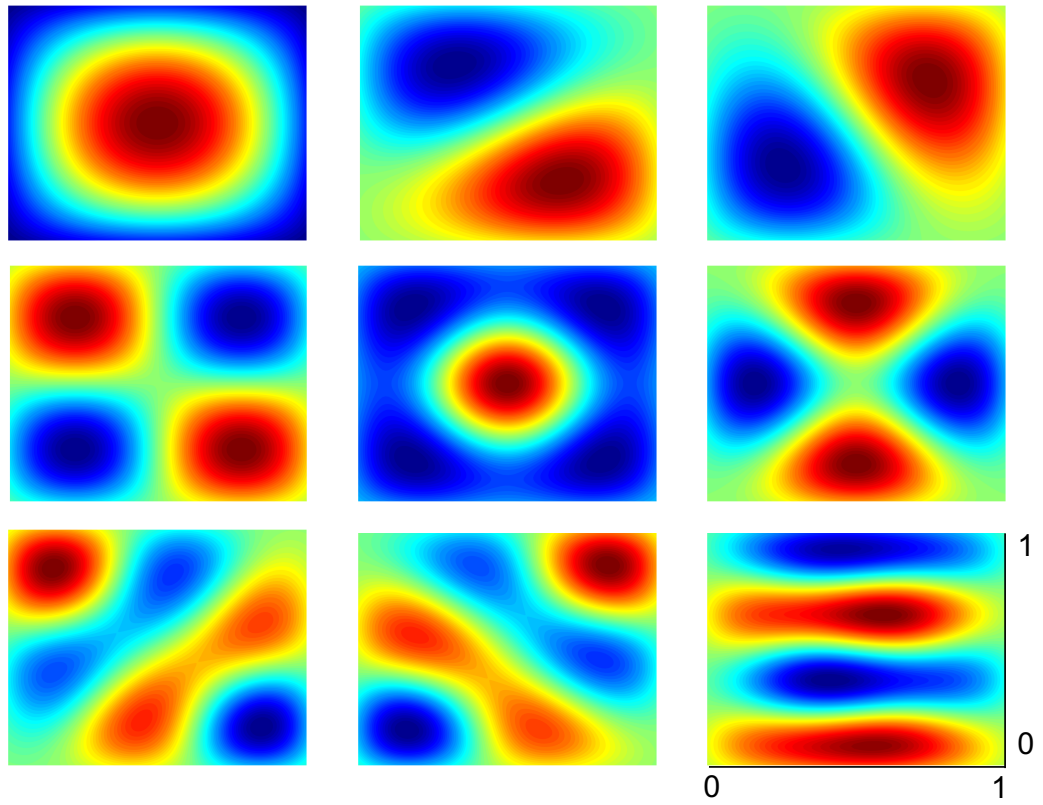
$$k(\mathbf{x}) = \sum_{i=1}^{N_d} \xi_i \sqrt{\lambda_i} \phi_i(\mathbf{x}) \quad (6.6)$$

where  $N_d = 100$  is the dimensionality of this random field,  $\boldsymbol{\xi} = [\xi_1, \xi_2, \dots, \xi_{N_d}]$  is a random vector that defines the weights assigned to each basis. In order to construct the basis functions, we first build a covariance matrix  $C$  over the domain  $\Omega$ . In this particular example, we use the Matérn covariance function such that

$$C_{ij} = \sigma^2 \exp\left(1 + \frac{\sqrt{3}r}{\ell}\right) \exp\left(-\frac{\sqrt{3}r}{\ell}\right) \quad (6.7)$$

where  $r$  is the radial distance between points  $i$  and  $j$ . Here,  $\sigma^2 = 4$  and  $\ell = 1.25 \times 10^{-3} \text{m}$ . Then we compute eigenfunctions  $\phi(\mathbf{x})$  of  $C$  corresponding to  $N_d$  largest eigenvalues  $\lambda$  which, scaled by  $\sqrt{\lambda}$  produce the required basis vectors, the first nine of which are shown in Fig. 6-3. We note that the choice of covariance function is user dependent and can be substituted with any other suitable kernel. The computation of  $C$  is an expensive task with a memory requirement inversely proportional to  $h^d$ , where  $d$  is the dimension of  $\Omega$ . However, as our mesh is constant, we compute  $C$  and its eigendecomposition offline prior to MCMC and store the basis functions. This is one reason why a multilevel MCMC approach that continuously adapts  $h$  would become infeasible for large problems, as the covariance matrix would need to be recomputed every time mesh adaptation occurs.

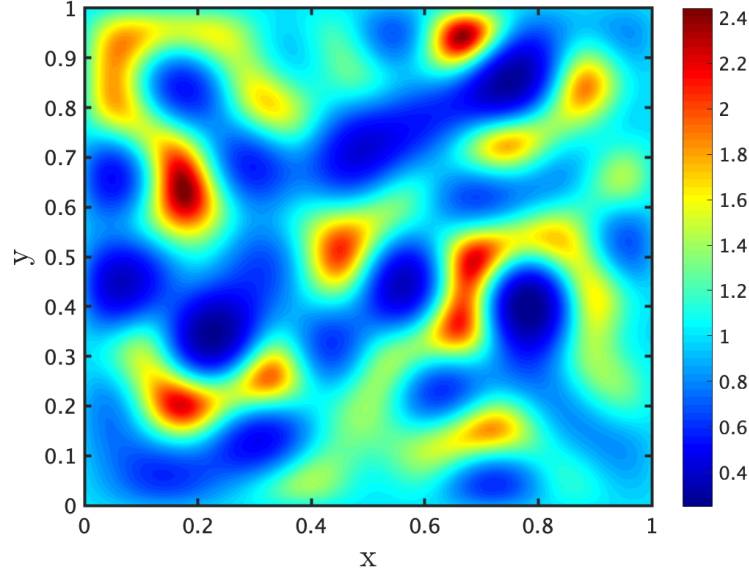
To test the method, we use a random vector  $\boldsymbol{\xi}_r$  to create a reference random field. This random field, shown in Fig. 6-4, we consider to be our *truth* that the delayed acceptance MCMC algorithm must infer. The corresponding finite element displacement



**Figure 6-3:** *Basis functions used to create a Gaussian random field. All plots range between -1 and 1 on the domain  $[0,1]^2$ .*

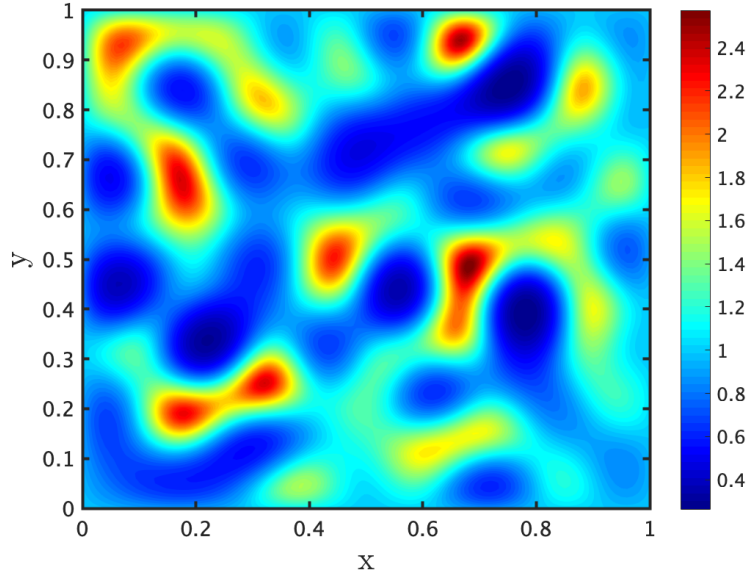
---

solution for the reference field is our quantity of interest. We chose the displacement solution as the direct observation for two reasons; firstly, using techniques such as digital image correlation, a richer surface displacement field of a test specimen can be obtained as opposed to say, strain measurements. Secondly, using displacements saves significant computation effort as it avoids strain calculations.



**Figure 6-4:** *The reference elastic field created using the basis functions shown in Fig. 6-3. This is treated as the truth to be determined by solving the inverse problem.*

To construct the multiscale model, the problem domain is divided into a  $4 \times 4$  grid of structured subdomains with 25 elements of overlap. In the numerical experiments of Chapter 5, we refer to this overlap as the 50% overlap case which we have shown achieves the greatest accuracy. So, for a total of 16 subdomains, the initial coarse space matrix  $\mathbf{R}_H^T$  is of size  $80,801 \times 48$ . As a result, all fine or  $\xi$ -chain samples require evaluating  $80,801 \times 80,801$  sized models but the first sample on the coarse or  $\eta$ -chain is reduced to  $48 \times 48$  dense matrix. The subchain length is fixed at 50 such that every 50th sample is evaluated by both chains and the error indicator  $t_H(\xi)$  is checked. If it exceeds  $\epsilon$ , and the current sample has been accepted, the solution is added to  $\mathbf{R}_H^T$  which is then normalized. The normalization step is non trivial. Before the first update occurs,  $\mathbf{R}_H^T$  includes only zero energy modes per subdomain therefore, any solution that is added to the coarse space must first be stripped of the zero energy modes as they have already been accounted for. In the absence of zero energy modes, every solution that is added is orthonormalized with respect to all other solutions added previously using the Gram-Schmidt process. The number of updates are governed by the accuracy



**Figure 6-5:** *Mean field for computed parameters*

demanded from the multiscale model. We demonstrate the performance of the method for  $\epsilon = 10^{-1}$ .

We use here the pCN proposal to draw a fresh sample for the  $\eta$ -chain. This too is user defined and can be substituted with any other form of proposal. One may use elaborate schemes such as delayed rejection to procure a new sample for the  $\eta$ -chain since the delayed acceptance algorithm ensures that every sample passed to the  $\xi$ -chain comes from the true posterior and not its approximation. Therefore, further gains can be achieved by increasing the efficiency of the  $\eta$ -chain. Since this chapter is intended as a proof-of-concept, these possible gains are not investigated any further and remain the subject of future work.

To verify the methodology, we compare the delayed acceptance algorithm to the standard MCMC algorithm which samples from the true posterior using pCN proposals with the same sampling parameters. In order to make a fair comparison, we carry out an equal number of model evaluations for both algorithms. The first 2000 samples of the delayed acceptance chains constitute the burn-in period and are discarded. Since the  $\eta$ -chain length is fixed at 50 samples, the first  $1 \times 10^5$  samples of the standard chains are also discarded. We run multiple chains with randomized start points for each algorithm, all with a pCN step size of 0.009. The step size can be tuned to achieve a desirable acceptance ratio however, smaller values increase autocorrelation length of a chain thus reducing the effective sample size (ESS). Alternatively, large values

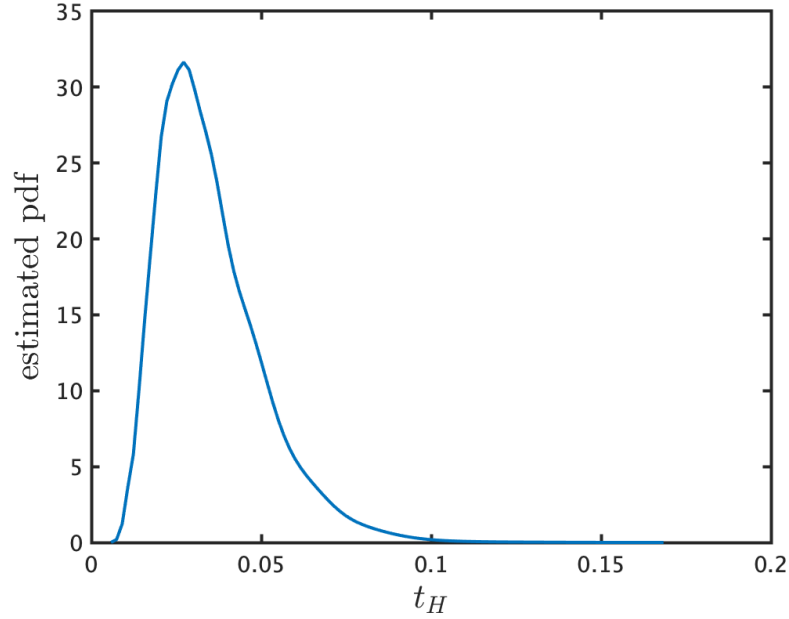
---

can increase ESS but only up to a limit beyond which proposals generated are so far away that their likelihood is almost always worse and chances of accepting it diminish. Consequently, the acceptance rate drops and autocorrelation length is again increased. The step size chosen here is based on multiple short runs (not reported), optimized to achieve the largest ESS whilst efficiently exploring the posterior without stagnation. Under these parameters, the average acceptance rate of the standard MCMC chains is 55%. This means that 45% of the full model evaluations are rejected before uncorrelated samples are extracted. Comparatively, the average acceptance ratio of the delayed acceptance chains is 99%. It makes use of almost all full model evaluations thus improving statistical efficiency relative to standard MCMC.

Another loss of efficiency comes in the form of autocorrelation length of a chain which is inversely proportional to ESS. As mentioned previously, the only negotiation of ESS a standard MCMC chain with pCN proposal permits is tuning the step size which has limited potential. Comparatively, the delayed acceptance scheme offers more flexibility. Autocorrelation length of the  $\xi$ -chain can be reduced by increasing the length of the  $\eta$ -chain. This has two advantages. First, the effective sample size is increased. Second, the path to the next effective sample can be traversed faster as the number of evaluations of the coarse model exceed the number of full model evaluations. The average cost of solving this particular fine model is 0.35s. The corresponding coarse model solve time, after the updates have been completed, is 0.0028s.

The acceptance rate of the  $\xi$ -chain is determined by the accuracy of the coarse model which in turn is controlled by the user defined error threshold  $\epsilon$ . The error indicator  $t_H$  serves as a measure of quality of the multiscale model relative to the fine scale model. If  $t_H > 0.1$ , an update is triggered. The plot in Fig. 6-6 shows the distribution of  $t_H$ . The variance of the error indicator is  $2.3 \times 10^{-4}$  for over 2000 independent samples which suggests a very accurate multiscale model. In other words, it may suffice to use a greater error threshold thus reducing the number of updates. However, altering the update threshold value affects the efficiency of the algorithm. It is easy to imagine that a higher threshold would trigger fewer updates and the coarse model would be cheaper to solve. But, too large a threshold would be more forgiving of bad samples which in turn would increase sampling error thus requiring more samples to achieve a given accuracy. So, the choice of the threshold value is problem specific, depending on what level of accuracy is affordable or desirable.

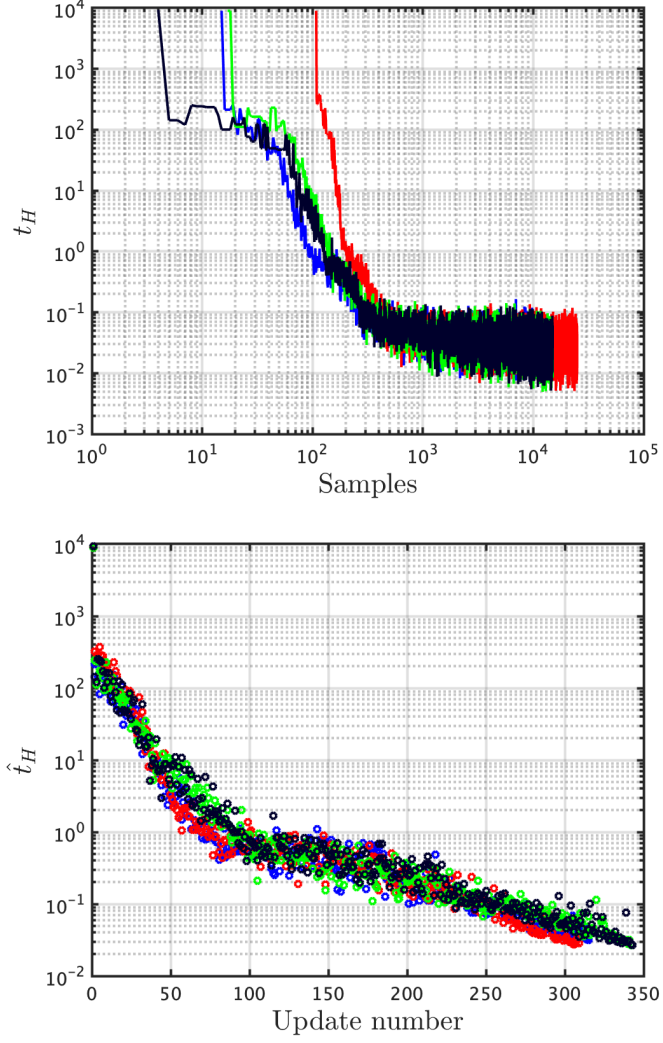
In Fig. 6-7(top), we see 4 chains progressing through the posterior and the reduction in true scaled error as the accuracy of the multiscale model improves. Figure 6-7(bottom) shows the error convergence with every update for each of the 4 chains. The values plotted are of  $\hat{t}_H$  which represents the average error of the chain between



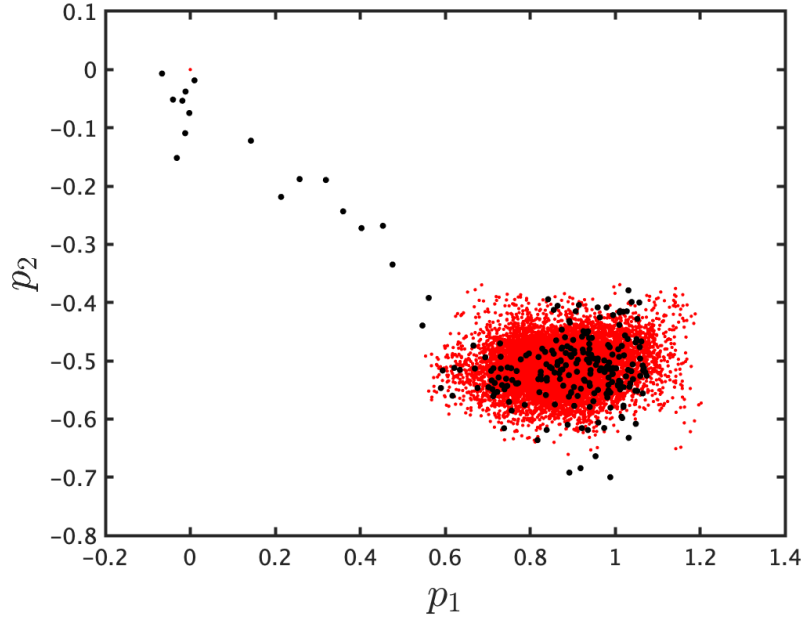
**Figure 6-6:** Probability density of the error between fine scale and multiscale models  $t_H$ . The variance in error  $\mathbb{V}(t_H) = 2.3 \times 10^{-4}$ .

two consecutive updates to make the data clearer. On average each chain updates the coarse space  $\mathbf{R}_H^T$  328 times.

The multiscale model used to solve the first sample on the  $\eta$ -chain comprises only zero energy modes thus producing an inaccurate solution, the maximum error of which exceeds the update threshold. Therefore, coarse space updates begin from the first sample. In the initial stages of the Markov chain, the coarse space is updated for nearly every new sample till it contains a sufficient number of snapshots that it can accurately approximate a multiscale solution. So, most of the updates lie within the burn-in period of the chain. Since the chain is moving towards the posterior and every new sample is significantly different from its predecessor, previously computed snapshots are unable to represent the solution for new samples and thus the current snapshot is added to the coarse space. In this way, the coarse space continuously learns about the posterior. This phenomenon is visualized in Fig. 6-8 where the black dots mark the snapshots chosen to update the coarse space. The update process also explains the decrease in the rate of error reduction beyond approximately 100 updates, as seen in Fig. 6-7(bottom). Once it settles into the posterior, the updates become increasingly infrequent till they stop entirely. This is because, once within the posterior, the samples do not differ as much as they do in the burnin. One might argue that some of the samples in the posterior are very close to each other and only one of those should suffice to make the coarse



**Figure 6-7:** *Top: Actual scaled error  $t_H$  for a chain can be seen converging to the user provided threshold  $\epsilon = 10^{-1}$ . Bottom: Error reduces as the multiscale model learns the posterior distribution. The error plotted here,  $\hat{t}_H$ , is the mean of the scaled error  $t_H$  of a chain between two consecutive updates. The plots are shown for 4 chains.*

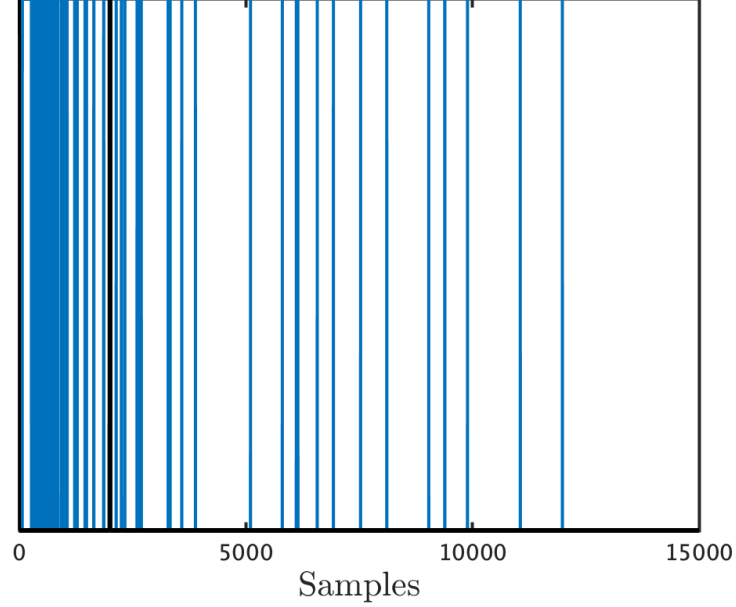


**Figure 6-8:** Black dots highlight the samples where the updated condition was triggered and the solution added to the coarse space with respect to the first two parameters  $p_1$  and  $p_2$ . The red dots mark all other samples drawn by the MCMC process. Note the burn-in where almost every sample requires an update.

space applicable to that region. However, samples that appear close to each other in Fig. 6-8 are only close with respect to the first two parameters. In at least one pair of parameters, they must be sufficiently far apart so as to trigger an update of the multiscale model. As a results, the multiscale model acquires enough information over time that with each update it becomes applicable to a larger region of the posterior. As the size of the coarse space increases and its quality improves, the effect of a single snapshot diminishes and fewer updates are required till the updates stop entirely. The plot in Fig. 6-9 visualizes the increasing sparsity of updates for one of the delayed acceptance chains. It also shows visually, the percentage of updates before and after burn-in.

At the end of a delayed acceptance MCMC process, we are left with a coarse space tailored to the posterior distribution of parameters. We now use this coarse space to achieve further gains by completely bypassing full model evaluations. However, a coarse model built from this coarse space can only sample from the approximate posterior  $\pi_H$  and will therefore have some bias. In the following test, we investigate the performance of this approximate or coarse MCMC. For this test, the coarse space is used to construct directly the multiscale model which is solved in the standard MCMC setting using pCN



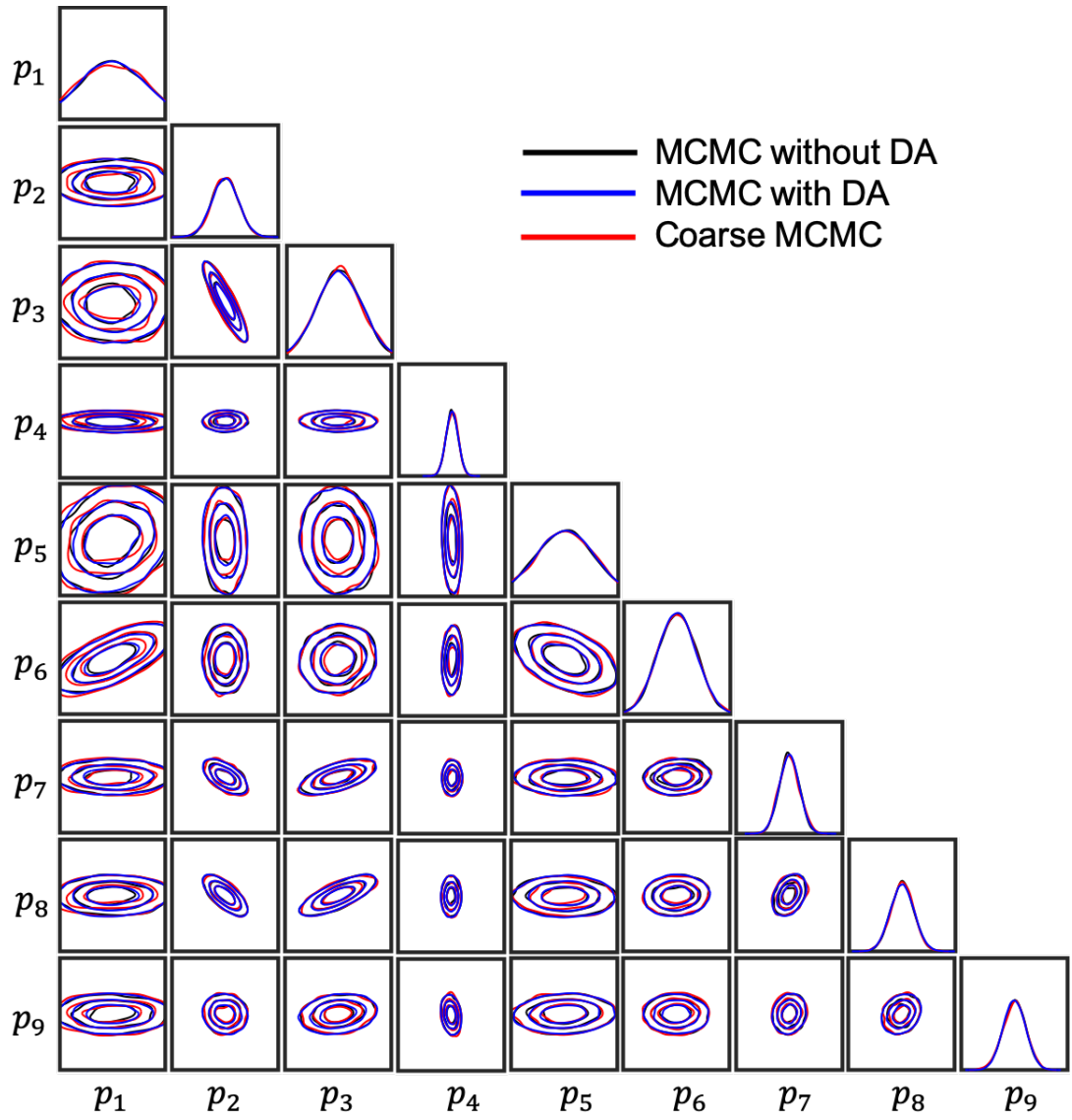


**Figure 6-9:** The  $x$ -axis shows sample of a chain and the vertical lines mark the samples where coarse space updates occurred. As the chain propagates, the frequency of updates reduces visualized here by the vertical lines becoming sparse as the coarse space learns about the posterior distribution. The vertical black line marks the burn-in length of the chain.

proposals with the same step size as before. Importantly, no full model evaluation is ever conducted by any of the coarse chains. The coarse space used here has a dimension of 376, the sum of 328 updates from the posterior and 48 zero energy modes. Six chains are initiated with random initial conditions and each evaluates  $5 \times 10^5$  coarse models.

In Fig. 6-10, we show the comparison of estimated posterior densities for all pairs of the first 9 parameters out of 100 for all three algorithms; standard MCMC (black), delayed acceptance MCMC (blue) and coarse MCMC (red). This demonstrates the sampling accuracy of the three algorithms. The black line represent results generated using standard MCMC to sample the full posterior. This we treat as our reference algorithm. The delayed acceptance MCMC results are shown in red which coincide with the black contours as this too samples from the full posterior. However, the slight differences that can be seen in some of the marginal densities are probably due to Monte Carlo error. The blue contours representing coarse MCMC show the marginal densities when samples are drawn from an approximate posterior. This bias would reduce as the coarse space dimension increases but that defeats the purpose of having a cheaper to solve coarse model.

We provide Table 6.1 to numerically demonstrate convergence of the all three



**Figure 6-10:** Posterior densities for the first 9 parameters of the random field.

---

	ESS	ESS/s	mean	$\sigma_{SE}$	95% confidence bounds
MCMC without DA	3436	0.08	0.8791	0.0032	[0.8728, 0.8854]
MCMC with DA	2514	0.6	0.8723	0.0039	[0.8647, 0.8798]
Coarse MCMC	2726	2.5	0.8810	0.0045	[0.8722, 0.8897]

---

**Table 6.1:** *Convergence properties of all three MCMC algorithms tested for the first parameter  $p_1$ .*

MCMC processes. We only study convergence in terms of the most dominant parameter  $p_1$  to aid readability. The table shows that 95% confidence bounds for the mean value of  $p_1$ . In order to compute confidence bounds, we require the sampling error  $\sigma_{SE}$  which is defined as

$$\sigma_{SE} = \sqrt{\frac{\mathbb{V}(p_1)}{\text{ESS}}} \quad (6.8)$$

where  $\mathbb{V}(p_1)$  is the variance of  $p_1$ . The overlapping confidence bounds show that all three algorithms converge to the mean. From this we can conclude that running any of the three MCMC algorithms will simulate the distribution of the parameters correctly with 95% confidence.

From Table 6.1 we can also compare the sampling accuracy of the algorithms. We observe that standard MCMC without DA has the lowest sampling efficiency as it requires approximately 13 seconds to draw an independent sample from the posterior. Comparatively, delayed acceptance MCMC pulls an independent sample in 1.7 seconds thus arriving at the next independent sample 7.5 times faster. The coarse MCMC algorithm requires even less time per independent sample. It offers a speed up factor of 31, requiring only 0.38s to the next independent sample.

#### 6.4.1 Parallel implementation

To validate the methodology presented in this chapter, we test the delayed acceptance MCMC algorithm on a single processor. More precisely, the forward model is solved on one processor. The ultimate aim is uncertainty quantification for large models such as the wingbox problem (see Section 3.5.3) which is impossible to solve using one processor only. Furthermore, the sequential implementation of delayed acceptance MCMC masks its true potential so that major gains are not immediately clear.

In the current implementation, for each forward solve, the full stiffness matrix is assembled to which the reduction operator  $\mathbf{R}_H^T$  is applied thus obtaining the multiscale model. In `dune-composites`, the full stiffness matrix is never assembled, only the local stiffness matrix is. Consequently, the full coarse space does not need to be assembled either but it does require nearest neighbour communication. Therefore, the cost of

---

building the multiscale model restricted to one processor is of the order  $n^2$  however the cost of obtaining the solution scales linearly which is shown by the weak scaling test in Fig. 3-9. This is one avenue for efficiency gain. In this sequential example, the cost of building the multiscale model on average is 1.43s. Dividing the computation between 16 processors, the approximate cost will be 0.09s. Combining that with the coarse solve gives a total time of 0.092s. This is nearly 4 times faster than the fine solve time of 0.35s. Importantly, the speed up factor increases as the problem size grows due to the rates at which the cost of fine and coarse model solves increases.

On a wingbox type simulation, where we are primarily interested in the mechanics of the region containing defects, we can exploit the parallel setup for further speed up. In the example above, the solution is queried at 441 grid points where the solutions of the two models are compared to find the worst error. This requires another expensive operation of elongating the multiscale solution to recover the fine scale solution. In the wingbox simulation, this too can be avoided by only recovering the fine solution in the region of interest.

We have shown in Chapter 5, the multiscale model can be reduced by a factor of over 500 thus the cost of solving it will be much faster than a fine solve especially when the bottleneck of entire matrix assembly is bypassed. However, such problems now require preconditioned iterative solvers. The number of iterations depends on the quality of the preconditioner. Since the cost of delayed acceptance MCMC is dominated by fine solves, we wish to improve upon them. This can be achieved by learning the preconditioner. As the coarse space is learnt over the posterior, the second level restriction operator can be applied in reverse as an elongation operator to continuously improve the second level of a two level additive Schwarz preconditioner. In this way, the cost of fine solves can be maintained or prevented from increasing. In fact, as the reusability of the preconditioner increases, the setup cost will eventually be amortized. In this way, uncertainty quantification can be implemented within the `dune-composites` framework.

It is important to note that additional errors will appear in a parallel implementation of this methodology. As we have shown in Chapter 5, the partition of unity operator has a significant influence on the solution. Therefore, the multiscale solution calculated in this way will lose some accuracy however the choice of partition of unity has little effect on the preconditioned CG solve of the fine scale model. So, the cost of evaluating the full model would change insignificantly since we show that alternative partition of unity operators affect the convergence of CG by at most 1 iteration. However, the cost of evaluating the multiscale model could fluctuate depending on the accuracy of the coarse space which in turn determines the number of required updates.

---

Nevertheless, the cheaper model is expected to be at least a few hundred times smaller and thus much faster.

## 6.5 Concluding remarks

In this chapter, we have demonstrated a proof of concept for data adjusted model reduction to quantify uncertainty in material modelling. The ultimate goal is to integrate this approach into the `dune-composites` module in order to perform UQ analysis on large composite structures where it becomes necessary to use available data in a way that reduces computational burden since large scale models are already on the edge of current computational capabilities.

In this approach, we address two bottlenecks of MCMC algorithms; the cost of solving forward models and the autocorrelation time of a chain. The former is reduced by replacing several expensive forward solves by cheaper multiscale solves. The latter is addressed by coupling the multiscale model with the delayed acceptance MCMC framework. In order to achieve this, we adapt existing algorithms to more efficiently explore posterior distributions. Firstly, the delayed acceptance algorithm draws independent samples from the true posterior at a lower cost than standard MCMC. Secondly, coarse MCMC samples from an approximate posterior at a much lower cost but introduces a bias. The amount of bias that is acceptable depends on the application.

The key idea underlying both algorithms is the self learning coarse space tailored to the problem at hand. As opposed to typical methods that learn from the prior offline, this method learns from the posterior in a more targeted way by updating the coarse space on-the-fly thus improving accuracy as it progresses. Using this approach we demonstrate that both algorithms accelerate MCMC beyond the capabilities of offline approaches since coarse spaces built online can be much smaller to represent regions of high probabilities.

Finally, we provide a discussion about the parallel implementation of these algorithms to achieve our ultimate goal. We believe that the numerical analysis presented in this chapter undersells the concept as it masks its true potential. In a large scale parallel setting, the delayed acceptance holds great promise and will be implemented in the future release of `dune-composites`.



---

---

## CHAPTER 7

---

# CONCLUDING REMARKS AND FUTURE WORK

The advantages offered by composites are well known however, exploiting them to our benefit is becoming increasingly challenging as the scale of applications grows. The challenges investigated in this thesis are outlined in Chapter 1. In this work, we address three challenges pertaining to composite modelling simultaneously identifying avenues of future work from which the composite industry would benefit greatly.

In Chapter 3, we tackle the first two challenges i.e. the scale of calculations and material anisotropy. Existing parallel solvers' limitations in terms of scalability and performance are the primary reasons why large problems lie outside their scope. We offer a collaboratively developed high performance package **dune-composites** that shows near optimal scaling on over 15,000 cores. With this package we can efficiently solve real world problems in excess of 200 million degrees of freedom in minutes. In the context of composite laminates, 200 million degrees affords us the resolution to model sub millimeter wrinkles say, in metre scale aircraft wingbox, see Section 3.5.3. The package is validated by a standard benchmark test in the field of subsurface flow called SPE10.

With this capability, we investigated the effects of wrinkle defects that produce variability in materials (part of challenge 3). So far, wrinkles have been modelled in a deterministic way described by sinusoidal waveforms. We are aware of only one study that employs Fourier series. In Chapter 4, we showed the inadequacy of such descriptions of wrinkles and provide a novel parameterization that captures salient

---

features such as their non-periodic nature. We extended an existing method to extract wrinkle information from ultrasonic scans and present a method to predict the strength knockdown caused by their presence. We showed that, given sufficient data, a lookup chart can be created to estimate knockdown based on maximum wrinkle slope thus bypassing expensive model evaluations.

The remainder of challenge 3 is addressed in Chapters 5 and 6 where we explored variability in models. We demonstrate the procedure for building a multiscale model from the **GenEO** coarse space which itself is a byproduct of a solve using the two-level additive Schwarz preconditioner within **dune-composites**. A key ingredient for a multiscale model built in this way is the partition of unity operator that specifies how subdomains are stitched together. In order to understand their importance, we investigated the performance of two purely geometric and one solution based PoU. From our tests, we concluded that the solution based PoU outperforms the others but comes at the cost of an additional solve. The smooth geometric PoU is a close second but the solution may require post processing which too can include an additional solve. However, post processing need only be done on a region of interest within the problem domain. So, the choice of PoU is largely determined by application. Importantly, we do not claim that any of the partitions tested are optimal which remains the subject of future work. These multiscale models are shown to reduce model orders by factors as great as 550 in 2D and 130 in 3D.

Model order reduction is particularly useful in order to quantify uncertainty. The prohibitive cost of multiple evaluations of a fine scale model makes UQ infeasible so we exploit the cheaper model which still retains sufficient accuracy. The cheaper model is embedded in a delayed acceptance MCMC framework which does two things well. Firstly, it learns the posterior distribution on the fly and is therefore able to achieve good accuracy that is tailored to the particular problem being solved such that the multiscale model remains cheap to solve. Secondly, it increases statistical efficiency by filtering out samples with low likelihoods at a much lower cost. This particular avenue for UQ is treated as a proof of concept in this thesis. The full scale parallel implementation in **dune-composites** is a part of the future work that will follow.

## 7.1 Future work

### 7.1.1 Nonlinear mechanics

The current limitations of this work open up new areas of research that we now discuss so as to motivate researchers to further the field of composite modelling. The work presented here is focussed on linear material behaviour and nonlinear mechanics such



---

as failure or fatigue modelling have not been implemented although implementation of cohesive zone models is currently underway. However, FE modelling of fracture mechanics has well documented problems. There are 3 notable issues;

- mesh dependency i.e. crack paths can vary significantly when the mesh parameter is changed
- usually a crack must be seeded within the problem domain and its path of propagation must be preallocated
- the choice of failure criteria affects initiation and propagation

Peridynamics is a novel particle based method developed by Silling *et al.* [183] in order to address these issues by modelling microscale behaviour but that approach makes it computationally expensive for a problem of say, wingbox dimensions (see Section 3.5.3). However, coupling cheaper macro scale FEM with peridynamics models localized to cracks is a promising direction of future research.

Peridynamics is a nonlocal alternative to classical continuum mechanics which is mathematically consistent for both continuous and discontinuous displacement fields. A typical peridynamic solution procedure involves discretizing the domain into interconnected particles. Crucially, each particle  $\mathbf{x}$  has a horizon  $\mathcal{H}_x$  such that all other particles that lie within the horizon are connected to  $\mathbf{x}$ , see Fig. 7-1. Numerous models have been developed that describe the mechanics of these connections but we will not discuss details about them. Interested readers are referred to [184, 185].

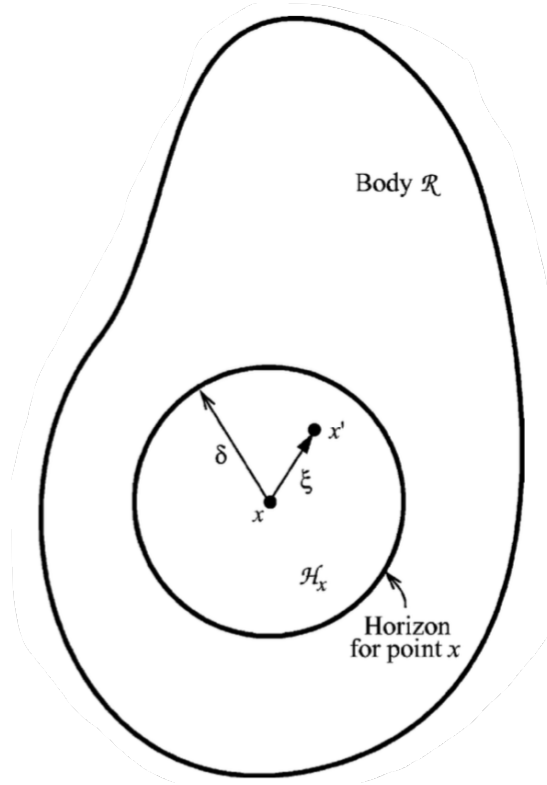
Here, we consider the simplest peridynamic model known as the Prototype Microelastic Brittle (PMB) model, derived in [186] where particles are essentially connected by springs. Consider a 2 particle system as shown in Fig. 7-2. Let  $\mathbf{u}(\mathbf{x}, t)$  be the displacement of particle  $\mathbf{x}$  at time  $t$  and  $\mathbf{y}(\mathbf{x}, t)$  be its position. Then the equation of equilibrium within the peridynamic framework is written as

$$\rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x}, t) = \int_{\mathcal{H}_x} f(\xi, \eta) dV \quad (7.1)$$

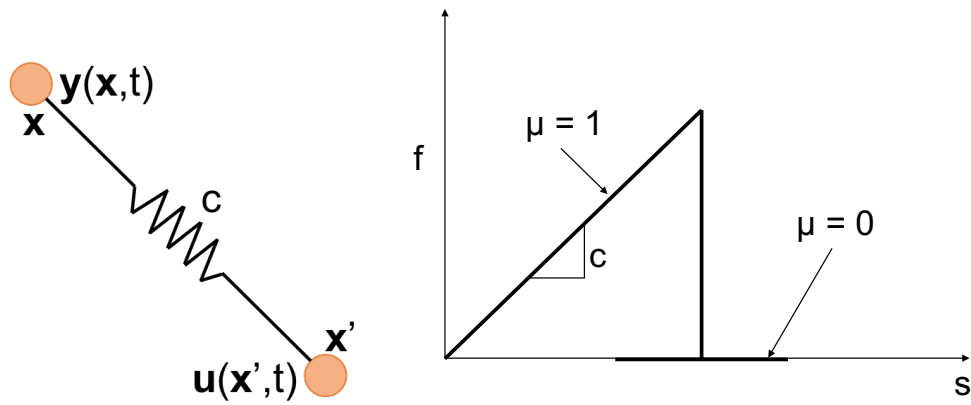
where  $\rho(\mathbf{x})$  is the density of particle  $\mathbf{x}$ , and

$$f(\xi, \eta) = \frac{1}{2} \frac{18K}{\pi\delta^4} \frac{\|\xi + \eta\| - \|\xi\|}{\|\xi\|} \quad (7.2)$$

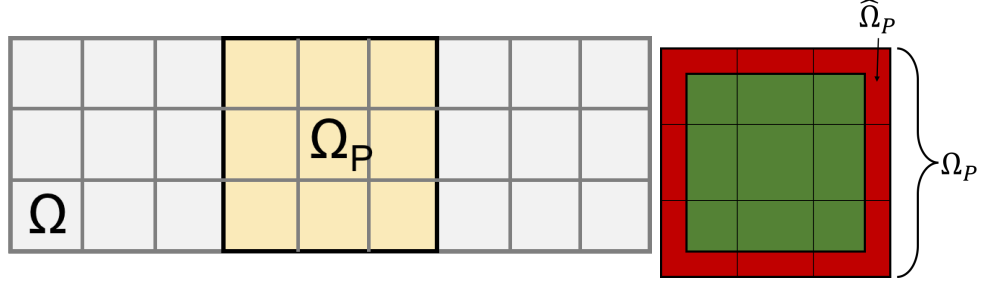
where  $\xi = \mathbf{y}(\mathbf{x}', t) - \mathbf{y}(\mathbf{x}, t)$  and  $\eta = \mathbf{u}(\mathbf{x}', t) - \mathbf{u}(\mathbf{x}, t)$ . The fraction  $\frac{18K}{\pi\delta^4}$  is called the micromodulus, denoted  $c$ , which in this model is based on the bulk modulus of the material  $K$  and horizon radius  $\delta$ . Equation (7.2) gives the scalar force experienced by



**Figure 7-1:** An arbitrary body showing a particle and the relevant features in a peridynamic framework.



**Figure 7-2:** Left: Two particle PMB model where  $c$  is the micromodulus or stiffness of the spring. Right: Constitutive law for the spring connection.



**Figure 7-3:** Left: A peridynamic subdomain embedded within a finite element domain. Right: More detailed drawing of the peridynamic subdomain showing the handshake region in red and the unconstrained peridynamic particles in green.

each particle in the two particle system. The force vector is obtained by computing

$$\mathbf{f}(\eta, \xi) = f(\eta, \xi) \frac{\eta + \xi}{\|\eta + \xi\|} \quad (7.3)$$

The bond stretch,  $s = \frac{\|\xi + \eta\| - \|\xi\|}{\|\xi\|}$  is the peridynamic equivalent of one dimensional strain and it is the quantity used to define failure criteria. In the PMB model, the bond has a linear elastic response as shown in Fig. 7-2(right). Once a bond exceeds its *critical stretch* value, it is permanently broken. The ratio of the number of broken bonds to the total number of bonds a particle has, is its damage parameter. Mathematically, this is written as

$$\varphi(t, \eta, \xi) = 1 - \frac{\int_{\mathcal{H}_x} \mu(t, \eta, \xi) dV_x}{\int_{\mathcal{H}_x} dv_x} \quad (7.4)$$

where  $\mu$  is a binary value equal to 1 while a bond is intact, 0 otherwise.

Using the peridynamic theory, we propose a multiscale formulation to model failure whereby a subregion of a problem domain where defects may exist and failure initiation is likely to occur. Consider a domain  $\Omega \in \mathbb{R}^2$  with a peridynamic  $\Omega_P$  model embedded within. The problem can be written as; find  $\mathbf{u}(\mathbf{x}) \in \mathbb{R}^2$  such that

$$\nabla \cdot \sigma(\mathbf{u}) + \mathbf{r}(\mathbf{u}) = 0 \quad \forall \mathbf{x} \in \Omega \quad (7.5)$$

with a function  $h(\mathbf{x})$  defining boundary conditions on Dirichlet boundaries  $\partial\Omega_D$  and  $g(x)$  defining Neumann boundary conditions on  $\partial\Omega_N$  with the standard FE constitutive law (see Eqs. (3.3) and (3.4)). In the absence of body forces,  $\mathbf{r}(\mathbf{u})$  is considered a force corrector which is obtained from the local peridynamic model. Essentially, at the macroscale, a crack exists in the form of internal forces pushing the crack surfaces away from each other. Naturally, this corrector only applies where the peridynamic model is defined such that  $\mathbf{r}(\mathbf{u}) = \mathbf{0} \quad \forall \mathbf{x} \in \Omega \setminus \Omega_P$ .

---

At the microscale,  $\Omega_P$  is discretized into a lattice of particles defined by the set

$$\mathcal{X} := \{\mathbf{x}_k : k = 1, 2, \dots, N \in \Omega_P\} \quad (7.6)$$

The particles are further divided into the *handshake* region  $\hat{\Omega}_P$  and unconstrained region  $\tilde{\Omega}_P = \Omega_P \setminus \hat{\Omega}_P$  as seen in Fig. 7-3. Macroscale deformations are interpolated onto the particles using the macroscale shape functions  $\phi_j(\mathbf{x})$  such that

$$\mathbf{y}(\mathbf{x}, t = 0) = \sum_{j=1}^n \mathbf{u}_h(\mathbf{x}) \phi_j(\mathbf{x}) \quad (7.7)$$

Importantly, microscale boundary conditions are only applied on the handshake region and the peridynamics solution  $\mathbf{u}_P(\mathbf{x})$  is obtained via energy minimization. Using  $L_2$  projection, we obtain a macroscale approximation  $\hat{\mathbf{u}}_h(\mathbf{x})$  by computing

$$\begin{aligned} (\mathbf{u}_P, \mathbf{v}_h) &= (\hat{\mathbf{u}}_h, \mathbf{v}_h) \\ \int_{\Omega_P} \mathbf{u}_P(\mathbf{x}) \mathbf{v}_h \, d\Omega_P &= \int_{\Omega_P} \hat{\mathbf{u}}_h(\mathbf{x}) \mathbf{v}_h \, d\Omega_P \\ \hat{\mathbf{u}}_h(\mathbf{x}) &= \mathbf{u}_h(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega \setminus \Omega_P \end{aligned} \quad (7.8)$$

In order to solve the integral in Eq. (7.8) we treat all the particles as equally weighted quadrature points for the macroscale finite element that contains them. With the projected solution, we can now write the force corrector in matrix form

$$\mathbf{K}(\hat{\mathbf{u}}_h(\mathbf{x}) - \mathbf{u}_h(\mathbf{x})) = \mathbf{r}(\mathbf{u}_h) \quad \forall \mathbf{x} \in \Omega \quad (7.9)$$

Finally, we have all the ingredients to define the nonlinear iteration scheme. Let  $\mathbf{u}_h^i(\mathbf{x})$  denote the  $i$ -th iteration. The system of equations then becomes

$$\mathbf{K}\mathbf{u}_h^i + \mathbf{r}(\mathbf{u}_h^i) = \mathbf{f} \quad (7.10)$$

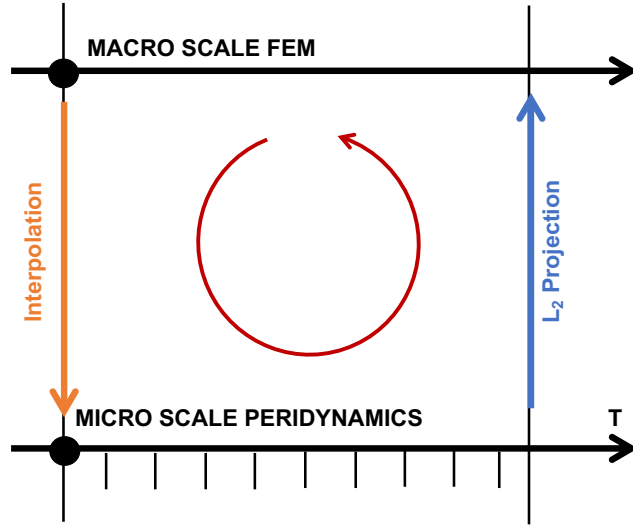
$$\mathbf{K}\mathbf{u}_h^{i+1} + \mathbf{r}(\mathbf{u}_h^{i+1}) = \mathbf{f} \text{ where } \mathbf{u}_h^{i+1} = \mathbf{u}_h^i + \Delta\mathbf{u}_h \quad (7.11)$$

Subtracting Eqs. (7.10) and (7.11), we get

$$\mathbf{K}\Delta\mathbf{u}_h + \mathbf{r}(\mathbf{u}_h^{i+1}) = \mathbf{r}(\mathbf{u}_h^i) \quad (7.12)$$

and rearrange to compute the solution

$$\mathbf{K}\Delta\mathbf{u}_h = \mathbf{r}(\mathbf{u}_h^i) - \mathbf{r}(\mathbf{u}_h^{i+1}) \quad (7.13)$$

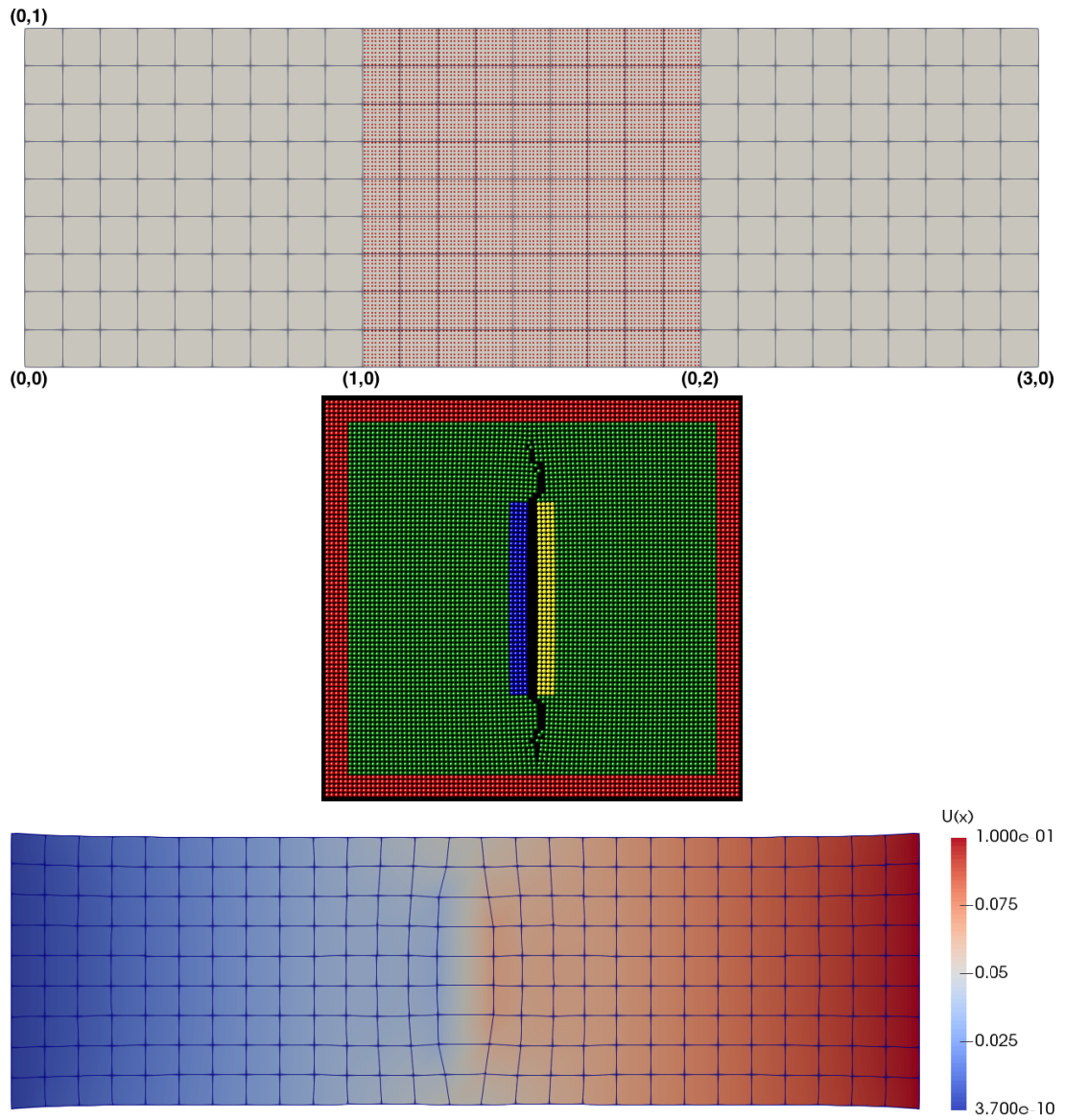


**Figure 7-4:** Solution strategy for multiscale modelling of failure using peridynamics on the microscale and FEM on the macroscale.

Solution converges as  $RHS \rightarrow 0$ . The coupled multiscale strategy is summarized in Fig. 7-4 and has the following steps,

1. Solve macroscale FEM to get  $\mathbf{u}(\mathbf{x}) \forall \mathbf{x} \in \Omega$
2. Interpolate macroscale solution onto microscale,  $\mathbf{y}(\mathbf{x}, 0) \forall \mathcal{X}$  and apply boundary conditions
3. Solve peridynamic system to get  $\mathbf{u}_P(\mathbf{x}, T) \forall \mathcal{X}$
4. Project microscale solution back to macroscale to get  $\hat{\mathbf{u}}_h \in \Omega$
5. Calculate force corrector  $\mathbf{r}(\mathbf{u}_h)$

Let us look at an example to clarify the coupled procedure. In Fig. 7-5 (top), a macroscale finite element grid is shown with a microscale grid embedded within. A crack is seeded between the blue and yellow band on Fig. 7-5(middle) which are equal in width to the radius of the horizon such that no bonds cross the blue-yellow particle interface. As in Fig. 7-3(right), the red particles lie in the handshake region which must be greater than  $\delta$  so as to avoid numerical instabilities which may cause premature failure at the boundary. The bottom picture in Fig. 7-5 shows the macroscale solution after the force corrector has been applied. At the microscale the crack is evident and propagates without a predefined path. At the macroscale, it is seen in the form of relatively large displacements. We note that it is not necessary to seed a crack in the microgrid but it is helpful for proving a concept as it makes the problem predictable.



**Figure 7-5:** *Coupled multiscale example*

The peridynamic subproblem is solved using a molecular dynamics code called LAMMPS. This particular implementation along with examples can be found in [187]. The LAMMPS code was combined and built with `dune-composites` to solve the coupled nonlinear problem.

### 7.1.2 Grids, subdomains and PoUs

Another limitation of the thesis is the strict adherence of `dune-composites` to structured grids on the fine level and structured subdomains on the coarse level. There are

---

two reasons to justify non-existence of unstructured grids and irregular subdomains within **dune-composites**. On the fine grid level, discretization of composite laminates using unstructured grids is very challenging for large problems. A much greater amount of data has to be stored in memory regarding the mesh whereas in structured grids that information can be computed as and when required. Besides memory requirements, the lack of sufficient scale separation in laminates compromises the stress solution if the mesh resolution is low in regions of high stress gradients such as interlaminar resin rich zones. In a structured grid, these regions can be meshed in the standard way using elements of constant size, at first. A geometric transformation is then applied to distort the mesh in a desired way thus achieving a sufficiently high resolution where required. Transforming unstructured grids in this way is far more difficult especially for conforming meshes.

On the subdomain level, the difficulty lies in discretizing meshes into overlapping subdomains, both regular and irregular. In the case of regular subdomains carved out of structured grids, the way subdomains are arranged affects the convergence of iterative solvers and an optimal decomposition for an arbitrary domain is not yet known. As we cannot have unstructured grids, some regularity is automatically imposed on the shape of subdomains which can no longer be completely arbitrary. But we cannot say whether the optimal decomposition will comprise regular or irregular subdomains. However, a major cost benefit in constructing the second level of the **GenEO** preconditioner stems from the structure of subdomains that is exploited to minimize communications between processors. With unstructured subdomains, a loss in efficiency is likely. This concern is deepened further when we consider the way subdomains are stitched together using a PoU operator. Treatment of overlaps has been observed to have a significant influence on the solution obtained from the **GenEO** multiscale model although, it does not affect much the quality of the **GenEO** preconditioner. In the context of large scale uncertainty quantification where multiscale models become indispensable, assuming that we somehow have an optimal arrangement of subdomains, we can then ask what is the optimal way to stitch them together? This too is unknown.





---

---

# APPENDIX A

---

## CODE BLOCKS FOR CHAPTER 3

In this appendix we provide C++ code blocks referenced in Chapter 3. These are meant to aid understanding and help users manipulate the `dune-composites` source code for their benefit.

---

**Algorithm 4** User defined function called `Model::LayerCake()`

---

```
1 void inline LayerCake(){ std::string example01a_Geometry =  
2   "stackingSequences/example1a.csv";  
3   LayerCakeFromFile(example01a_Geometry); GeometryBuilder()  
   ; }
```

---

---

**Algorithm 5** `Model::isDirichlet()`

---

```
1 bool inline isDirichlet(FieldVec& x, const int dof){ return  
   (x[0] <  
2   1e-6); }
```

---

---

**Algorithm 6** Model::evaluateNeumann()

---

```
1 inline void evaluateNeumann(const FieldVec& x, FieldVec& h,
    const
2 FieldVec& normal) const{ h = 0.0; // initialise to zero
    double T =
3 R[0].L[2]; // Thickness if (x[2] > T - 1e-6){ h[2] += q;
    } }
```

---

---

**Algorithm 7** Model::evaluateWeight()

---

```
1 inline void evaluateWeight(FieldVec& f, int id) const{ f =
    0; }
```

---

---

**Algorithm 8** Find maximum vertical deflection

---

```
1 template<class GO, class V, class GFS, class C> void inline
2 postprocess(const GO& go, V& u, const GFS& gfs, const C& cg
    ){ using
3 Dune::PDELab::Backend::native; double local_u3_max = 0.0;
    for (int i
4 = 0; i < native(u).size(); i++){ // Loop over each vertex
    double u3
5 = std::abs(native(u)[i][2]); if (local_u3_max < u3) {
    local_u3_max
6 = u3; } } MPI_Allreduce(&local_u3_max, &QoI, 1,
    MPI_DOUBLE,
7 MPI_MAX, MPI_COMM_WORLD); }
```

---

---

**Algorithm 9** Failure calculation according to Camanho criterion

---

```
1 template<....> void inline postprocess(...){ //material
    allowables in
2 MPa const std::vector<double> p = {61., 97., 94.}; double
    Fm = 0.0;
3 for (int i = 0; i < stress_mech.size(); i++){ double F =
4 Camanho(stress_mech[i], elemIndx2PG[i], p); if (Fm < F)
    { Fm = F;
5 } } double Fm_all; MPI_Allreduce(&Fm, &Fm_all, 1,
    MPI_DOUBLE,
6 MPI_MAX, MPI_COMM_WORLD); Q = pressure / Fm_all; //
    Failure load }
```

---



---

## REFERENCES

1. *The 2016 UK composites strategy: Delivering UK growth through the multi-sector application of composites* 2016.
2. Hughes, T. J. *The finite element method: Linear static and dynamic finite element analysis* (Courier Corporation, 2012).
3. Butler, R. *et al.* High-performance dune modules for solving large-scale, strongly anisotropic elliptic problems with applications to aerospace composites. *arXiv preprint arXiv:1901.05188* (2019).
4. Systèmes, D. *Abaqus analysis user's manual* (2007).
5. Groetsch, C. W. *Generalized inverses of linear operators: representation and approximation* (Dekker, New York, NY, 1977).
6. Cotter, S. L., Roberts, G. O., Stuart, A. M. & White, D. MCMC Methods for Functions: Modifying Old Algorithms to Make Them Faster. *Statist. Sci.* **28**, 424–446 (Aug. 2013).
7. Blatt, M. & Bastian, P. *The iterative solver template library* in *International Workshop on Applied Parallel Computing* (2006), 666–675.
8. Yang, U. M. & Henson, V. E. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics* **41**, 155–177 (2002).
9. Dodwell, T., Butler, R. & Hunt, G. Out-of-plane ply wrinkling defects during consolidation over an external radius. *Composites Science and Technology* **105**, 151–159 (Dec. 2014).

- 
10. Lightfoot, J. S., Wisnom, M. R. & Potter, K. A new mechanism for the formation of ply wrinkles due to shear between plies. *Composites Part A: Applied Science and Manufacturing* **49**, 139–147 (2013).
  11. Lubin, G. *Handbook of composites* (Springer Science & Business Media, 2013).
  12. Advisory Circular (AC) 20-107B - Composite Aircraft Structure. *U.S. Dept. of Transportation - Federal Aviation Authority (FAA)* (2009).
  13. EASA, A. AMC 20-29. *Composite aircraft structure* (2010).
  14. Pleshakov, E. *Own work* <https://commons.wikimedia.org/w/index.php?curid=3912586>.
  15. Hill, R. Elastic properties of reinforced solids: some theoretical principles. *Journal of the Mechanics and Physics of Solids* **11**, 357–372 (1963).
  16. Heinrich, C. *et al.* The influence of the representative volume element (RVE) size on the homogenized response of cured fiber composites. *Modelling and Simulation in Materials Science and Engineering* **20**, 075007 (Sept. 2012).
  17. Herakovich, C. T. *Mechanics of fibrous composites* (1998).
  18. Hill, R. Theory of mechanical properties of fibre-strengthened materials: I. Elastic behaviour. *Journal of the Mechanics and Physics of Solids* **12**, 199–212 (1964).
  19. Aboudi, J. Micromechanical analysis of composites by the method of cells. *Applied Mechanics Reviews* **42**, 193–221 (1989).
  20. Paley, M. & Aboudi, J. Micromechanical analysis of composites by the generalized cells model. *Mechanics of materials* **14**, 127–139 (1992).
  21. Bednarczyk, B. A. & Arnold, S. M. Micromechanics-based deformation and failure prediction for longitudinally reinforced titanium composites. *Composites Science and Technology* **61**, 705–729 (2001).
  22. Reuss, A. Berechnung der fließgrenze von mischkristallen auf grund der plastizitätsbedingung für einkristalle. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* **9**, 49–58 (1929).
  23. Voigt, W. Ueber die Beziehung zwischen den beiden Elasticitätsconstanten isotroper Körper. *Annalen der physik* **274**, 573–587 (1889).
  24. Reiner, M. Rheology. *Encyclopedia of Physics, Elasticity and Plasticity* **6**, 522–534 (1958).
  25. Reinartz, A. *et al.* Dune-composites - A new framework for high-performance finite element modelling of laminates. *Composite Structures* **184**, 269–278 (2018).
-

- 
26. Bargmann, S. *et al.* Generation of 3D representative volume elements for heterogeneous materials: A review. *Progress in Materials Science* **96**, 322–384 (2018).
  27. Kurtyka, P. & Rylko, N. Structure analysis of the modified cast metal matrix composites by use of the RVE theory. *Archives of Metallurgy and Materials* **58**, 357–360 (2013).
  28. Cunsolo, S. *et al.* Monte Carlo determination of radiative properties of metal foams: Comparison between idealized and real cell structures. *International Journal of Thermal Sciences* **87**, 94–102 (2015).
  29. Hashin, Z. & Shtrikman, S. A variational approach to the theory of the elastic behaviour of multiphase materials. *Journal of the Mechanics and Physics of Solids* **11**, 127–140 (1963).
  30. Mühlhaus, H.-B. in *Analysis and Design Methods* 209–230 (Elsevier, 1993).
  31. Kunin, I. A. *Elastic media with microstructure I: one-dimensional models* (Springer Science & Business Media, 2012).
  32. Potter, K. *Understanding the origins of defects and variability in composites manufacture* in *International Conference on Composite Materials (ICCM)-17, Edinburgh, UK* (2009).
  33. Purslow, D. On the optical assessment of the void content in composite materials. *Composites* **15**, 207–210 (1984).
  34. Liu, D., Fleck, N. & Sutcliffe, M. Compressive strength of fibre composites with random fibre waviness. *Journal of the Mechanics and Physics of Solids* **52**, 1481–1505 (2004).
  35. Butler, R. *et al.* *Uncertainty quantification of composite structures with defects using multilevel monte carlo simulations* in *17th AIAA Non-Deterministic Approaches Conference* (2015), 1598.
  36. Mukhopadhyay, S., Jones, M. I. & Hallett, S. R. Compressive failure of laminates containing an embedded wrinkle; experimental and numerical study. *Composites Part A: Applied Science and Manufacturing* **73**, 132–142 (2015).
  37. Boisse, P., Hamila, N. & Madeo, A. Modelling the development of defects during composite reinforcements and prepreg forming. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* **374** (2016).
  38. Sutcliffe, M., Lemanski, S. & Scott, A. Measurement of fibre waviness in industrial composite components. *Composites Science and Technology* **72**, 2016–2023 (2012).
-

- 
39. Smith, R. A., Mukhopadhyay, S., Lawrie, A. & Hallett, S. R. *Applications of ultrasonic NDT to aerospace composites* in *Proceedings of the 5th International Symposium on Aerospace NDT, Singapore* (2013).
  40. Nikishkov, G., Nikishkov, Y. & Makeev, A. Finite element mesh generation for composites with ply waviness based on X-ray computed tomography. *Advances in Engineering Software* **58**, 35–44 (2013).
  41. Meola, C. *et al.* Nondestructive evaluation of carbon fibre reinforced composites with infrared thermography and ultrasonics. *Composite Structures* **134**, 845–853 (2015).
  42. Mizukami, K., Mizutani, Y., Todoroki, A. & Suzuki, Y. Detection of in-plane and out-of-plane fiber waviness in unidirectional carbon fiber reinforced composites using eddy current testing. *Composites Part B: Engineering* **86**, 84–94 (2016).
  43. Lemanski, S. *et al.* Modelling failure of composite specimens with defects under compression loading. *Composites Part A: Applied Science and Manufacturing* **48**, 26–36 (2013).
  44. Fletcher, T. A. *et al.* Resin treatment of free edges to aid certification of through thickness laminate strength. *Composite Structures* **146**, 26–33 (2016).
  45. Xie, N., Smith, R. A., Mukhopadhyay, S. & Hallett, S. R. A numerical study on the influence of composite wrinkle defect geometry on compressive strength. *Materials and Design* **140**, 7–20 (2018).
  46. Wang, J., Potter, K., Hazra, K. & Wisnom, M. Experimental fabrication and characterization of out-of-plane fiber waviness in continuous fiber-reinforced composites. *Journal of Composite Materials* **46**, 2041–2053 (2012).
  47. Elhajjar, R. F. & Shams, S. S. Compression testing of continuous fiber reinforced polymer composites with out-of-plane fiber waviness and circular notches. *Polymer Testing* **35**, 45–55 (2014).
  48. Adams, D. O. & Hyer, M. Effects of Layer Waviness on the Compression Strength of Thermoplastic Composite Laminates. *Journal of Reinforced Plastics and Composites* **12**, 414–429 (1993).
  49. Kratmann, K. K. *et al.* A novel image analysis procedure for measuring fibre misalignment in unidirectional fibre composites. *Composites Science and Technology* **69**, 228–238 (2009).
  50. Wade, M. Effects of periodic and localized imperfections on struts on nonlinear foundations and compression sandwich panels. *International Journal of Solids and Structures* **37**, 1191–1209 (2000).
-

- 
51. Ghanem, R. G. & Spanos, P. D. *Stochastic finite elements: a spectral approach* (Courier Corporation, 2003).
  52. Conrad, P. *et al.* Parallel local approximation MCMC for expensive models. *SIAM/ASA Journal on Uncertainty Quantification* **in press** (2018).
  53. Rosas-Carbajal, M., Linde, N., Kalscheuer, T. & Vrugt, J. A. Two-dimensional probabilistic inversion of plane-wave electromagnetic data: methodology, model constraints and joint inversion with electrical resistivity data. *Geophysical Journal International* **196**, 1508–1524 (2013).
  54. Sorensen, D. & Gianola, D. *Likelihood, Bayesian, and MCMC methods in quantitative genetics* (Springer Science & Business Media, 2007).
  55. Wilks, D. S. *Statistical methods in the atmospheric sciences* (Academic press, 2011).
  56. Olsson, A., Sandberg, G. & Dahlblom, O. On Latin hypercube sampling for structural reliability analysis. *Structural safety* **25**, 47–68 (2003).
  57. Glynn, P. W. & Iglehart, D. L. Importance sampling for stochastic simulations. *Management Science* **35**, 1367–1392 (1989).
  58. Carter, C. K. & Kohn, R. On Gibbs sampling for state space models. *Biometrika* **81**, 541–553 (1994).
  59. Gamerman, D. & Lopes, H. F. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference* (Chapman and Hall/CRC, 2006).
  60. Stefanou, G. The stochastic finite element method: Past, present and future. *Computer Methods in Applied Mechanics and Engineering* **198**, 1031–1051 (2009).
  61. Liu, W. K., Belytschko, T. & Mani, A. Probabilistic finite elements for nonlinear structural dynamics. *Computer Methods in Applied Mechanics and Engineering* **56**, 61–81 (1986).
  62. Wiener, N. The homogeneous chaos. *American Journal of Mathematics* **60**, 897–936 (1938).
  63. Ghanem, R. G. & Spanos, P. D. *Stochastic finite elements: a spectral approach* (Courier Corporation, 2003).
  64. Mathelin, L., Hussaini, M. Y. & Zang, T. A. Stochastic approaches to uncertainty quantification in CFD simulations. *Numerical Algorithms* **38**, 209–236 (2005).
  65. Xiu, D. *Numerical methods for stochastic computations: a spectral method approach* (Princeton university press, 2010).
-



- 
66. Rasmussen, C. E. *Gaussian processes in machine learning* in *Summer School on Machine Learning* (2003), 63–71.
67. Krauth, K., Bonilla, E. V., Cutajar, K. & Filippone, M. AutoGP: Exploring the capabilities and limitations of Gaussian process models. *arXiv preprint arXiv:1610.05392* (2016).
68. Fishman, G. *Monte Carlo: concepts, algorithms, and applications* (Springer Science & Business Media, 2013).
69. Cui, T., Marzouk, Y. M. & Willcox, K. E. Data-driven model reduction for the Bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering* **102**, 966–990 (2015).
70. Marzouk, Y. M., Najm, H. N. & Rahn, L. A. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics* **224**, 560–586 (2007).
71. Parno, M. & Marzouk, Y. Transport map accelerated markov chain monte carlo. *arXiv preprint arXiv:1412.5492* (2014).
72. Dodwell, T. J., Ketelsen, C., Scheichl, R. & Teckentrup, A. L. A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. *SIAM/ASA Journal on Uncertainty Quantification* **3**, 1075–1108 (2015).
73. Gilks, W. R., Richardson, S. & Spiegelhalter, D. *Markov chain Monte Carlo in practice* (CRC press, 1995).
74. Metropolis, N. *et al.* Equation of state calculations by fast computing machines. *The journal of chemical physics* **21**, 1087–1092 (1953).
75. Tierney, L. & Mira, A. Some adaptive Monte Carlo methods for Bayesian inference. *Statistics in medicine* **18**, 2507–2515 (1999).
76. Gelfand, A. E. & Smith, A. F. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association* **85**, 398–409 (1990).
77. Cowles, M. K. & Carlin, B. P. Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review. *Journal of the American Statistical Association* **91**, 883–904 (1996).
78. Gelman, A., Rubin, D. B., *et al.* Inference from iterative simulation using multiple sequences. *Statistical science* **7**, 457–472 (1992).
79. Raftery, A. E. & Lewis, S. *How many iterations in the Gibbs sampler?* tech. rep. (Washington University Seattle Dept Of Statistics, 1991).
-

- 
80. Geyer, C. J. Practical markov chain monte carlo. *Statistical science*, 473–483 (1992).
81. Raftery, A. L. & Lewis, S. One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. *Statistical Science, Hayward* **7**, 475–485 (1992).
82. Roberts, G. O., Gelman, A. & Gilks, W. R. Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms. *The Annals of Applied Probability* **7**, 110–120 (1997).
83. Plummer, M., Best, N., Cowles, K. & Vines, K. CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News* **6**, 7–11 (2006).
84. MacEachern, S. N. & Berliner, L. M. Subsampling the Gibbs sampler. *The American Statistician* **48**, 188–190 (1994).
85. Chen, M.-H., Shao, Q.-M. & Ibrahim, J. G. *Monte Carlo methods in Bayesian computation* (Springer Science & Business Media, 2012).
86. Roberts, G. O., Tweedie, R. L., *et al.* Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* **2**, 341–363 (1996).
87. Roberts, G. O. & Rosenthal, J. S. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **60**, 255–268 (1998).
88. Christen, J. A. & Fox, C. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical statistics* **14**, 795–810 (2005).
89. Roberts, G. O. & Rosenthal, J. S. Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics* **18**, 349–367 (2009).
90. Efendiev, Y., Galvis, J. & Hou, T. Y. Generalized multiscale finite element methods (GMsFEM). *Journal of Computational Physics* **251**, 116–135 (2013).
91. Jacobi, C. G. J. Ueber ein leichtes Verfahren, die in der Theorie der Sacularstörungen vorkommenden Gleichungen numerisch aufzulösen. *J. reine angew. Math.* 51–94 (1846).
92. Yoon, S. & Jameson, A. Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations. *AIAA journal* **26**, 1025–1026 (1988).
93. Hadjidimos, A. Accelerated overrelaxation method. *Mathematics of Computation* **32**, 149–157 (1978).
94. Gutknecht, M. H. in *Frontiers of Computational Science* 53–62 (Springer, 2007).
-

- 
95. Young, D. M. *Iterative solution of large linear systems* (Elsevier, 2014).
  96. Golub, G. H. & O’Leary, D. P. Some history of the conjugate gradient and Lanczos methods. *Siam review* **31**, 50–102 (1989).
  97. Saad, Y. *Iterative methods for sparse linear systems* (siam, 2003).
  98. Saad, Y. & Van Der Vorst, H. A. in *Numerical Analysis: Historical Developments in the 20th Century* 175–207 (Elsevier, 2001).
  99. Hestenes, M. R. & Stiefel, E. *Methods of conjugate gradients for solving linear systems* **1** (NBS Washington, DC, 1952).
  100. Lanczos, C. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bur. Standards* **49**, 33–53 (1952).
  101. Paige, C. C. & Saunders, M. A. Solution of sparse indefinite systems of linear equations. *SIAM journal on numerical analysis* **12**, 617–629 (1975).
  102. Fletcher, R. in *Numerical analysis* 73–89 (Springer, 1976).
  103. Saad, Y. & Schultz, M. H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing* **7**, 856–869 (1986).
  104. Freund, R. W. & Nachtigal, N. M. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numerische mathematik* **60**, 315–339 (1991).
  105. Van der Vorst, H. A. 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**, 631–644 (1992).
  106. Trefethen, L. N. & Bau III, D. *Numerical linear algebra* (Siam, 1997).
  107. Turing, A. M. Rounding-off errors in matrix processes. *The Quarterly Journal of Mechanics and Applied Mathematics* **1**, 287–308 (1948).
  108. Evans, D. J. The use of pre-conditioning in iterative methods for solving linear equations with symmetric positive definite matrices. *IMA Journal of Applied Mathematics* **4**, 295–314 (1968).
  109. Cesari, L. *Sulla risoluzione dei sistemi di equazioni lineari per approssimazioni successive* (G. Bardi, 1937).
  110. Bodewig, E. *Matrix calculus* (1956).
  111. Engeli, M. *et al. Refined iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems* (Springer, 1959).
-

- 
- 112. Meijerink, J. & Van Der Vorst, H. A. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of computation* **31**, 148–162 (1977).
  - 113. Kershaw, D. S. The incomplete Cholesky conjugate gradient method for the iterative solution of systems of linear equations. *Journal of computational physics* **26**, 43–65 (1978).
  - 114. Gustafsson, I. A class of first order factorization methods. *BIT Numerical Mathematics* **18**, 142–156 (1978).
  - 115. Watts III, J. W. *et al.* A conjugate gradient-truncated direct method for the iterative solution of the reservoir simulation pressure equation. *Society of Petroleum Engineers Journal* **21**, 345–353 (1981).
  - 116. Saad, Y. ILUT: A dual threshold incomplete LU factorization. *Numerical linear algebra with applications* **1**, 387–402 (1994).
  - 117. Benson, M. W. Iterative solution of large scale linear systems. (1973).
  - 118. Benson, M. W. Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems. *Util. Math.* **22**, 127–140 (1982).
  - 119. Fedorenko, R. P. The speed of convergence of one iterative process. *USSR Computational Mathematics and Mathematical Physics* **4**, 227–235 (1964).
  - 120. Brandt, A. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation* **31**, 333–390 (1977).
  - 121. Brandt, A. *Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems* in *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics* (1973), 82–89.
  - 122. Hackbusch, W. *Multi-grid methods and applications* (Springer Science & Business Media, 2013).
  - 123. Toselli, A. & Widlund, O. *Domain decomposition methods-algorithms and theory* (Springer Science & Business Media, 2006).
  - 124. Karypis, G. & Kumar, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing* **20**, 359–392 (1998).
  - 125. Pellegrini, F. Scotch and libScotch 5.1 user’s guide (2008).
  - 126. Nicolaides, R. A. Deflation of conjugate gradients with applications to boundary value problems. *SIAM Journal on Numerical Analysis* **24**, 355–365 (1987).
-

- 
- 127. Babuška, I. in *Numerical Solution of Partial Differential Equations–III* 89–116 (Elsevier, 1976).
  - 128. Farhat, C. & Roux, F.-X. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering* **32**, 1205–1227 (1991).
  - 129. Langer, U. & Pechstein, C. Coupled finite and boundary element tearing and interconnecting solvers for nonlinear potential problems. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics* **86**, 915–931 (2006).
  - 130. Klawonn, A. & Rheinbach, O. Robust FETI-DP methods for heterogeneous three dimensional elasticity problems. *Computer Methods in Applied Mechanics and Engineering* **196**, 1400–1414 (2007).
  - 131. Mandel, J. & Tezaur, R. Convergence of a substructuring method with Lagrange multipliers. *Numerische Mathematik* **73**, 473–487 (1996).
  - 132. Klawonn, A. & Widlund, O. FETI and Neumann-Neumann iterative substructuring methods: connections and new results. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* **54**, 57–90 (2001).
  - 133. Pechstein, C. & Scheichl, R. Analysis of FETI methods for multiscale PDEs. *Numerische Mathematik* **111**, 293–333 (2008).
  - 134. Toselli, A. & Widlund, O. *Domain decomposition methods-algorithms and theory* (Springer Science & Business Media, 2006).
  - 135. Spillane, N. *et al.* Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numerische Mathematik* **126**, 741–770 (2014).
  - 136. Bastian, P. *et al.* A generic grid interface for parallel and adaptive scientific computing. Part I: Abstract framework. *Computing* **82**, 103–119 (2008).
  - 137. Bastian, P. *et al.* A generic grid interface for parallel and adaptive scientific computing. Part II: Implementation and tests in DUNE. *Computing* **82**, 121–138 (2008).
  - 138. Blatt, M. & Bastian, P. On the generic parallelisation of iterative solvers for the finite element method. *International Journal of Computational Science and Engineering* **4**, 56–69 (2008).
-

- 
139. Belytschko, T., Ong, J. S.-J., Liu, W. K. & Kennedy, J. M. Hourglass control in linear and nonlinear problems. *Computer Methods in Applied Mechanics and Engineering* **43**, 251–276 (1984).
  140. Ting, T. C.-t. *Anisotropic elasticity: theory and applications* **45** (Oxford University Press on Demand, 1996).
  141. Saad, Y. *Iterative methods for sparse linear systems* (siam, 2003).
  142. Sarkis, M. Nonstandard coarse spaces and Schwarz methods for elliptic problems with discontinuous coefficients using non-conforming elements. *Numerische Mathematik* **77**, 383–406 (1997).
  143. Jolivet, P. *et al.* High performance domain decomposition methods on massively parallel architectures with FreeFem++. *Journal of Numerical Mathematics* **20**, 287–302 (2012).
  144. Jolivet, P., Hecht, F., Nataf, F. & Prud’Homme, C. Scalable domain decomposition preconditioners for heterogeneous elliptic problems. *Scientific Programming* **22**, 157–171 (2014).
  145. Gomes, F. M. & Sorensen, D. C. ARPACK++: A C++ implementation of ARPACK eigenvalue package. *CRPC, Rice University, Houston, TX, Tech. Rep. TR97729* (1997).
  146. Davis, T. A. Algorithm 832: UMFPACK V4. 3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software (TOMS)* **30**, 196–199 (2004).
  147. Bastian, P. *et al.* UG - A flexible software toolbox for solving partial differential equations. *Computing and Visualization in Science* **1**, 27–40 (1997).
  148. Arnold, D. N. & Awanou, G. The serendipity family of finite elements. *Foundations of Computational Mathematics* **11**, 337–344 (2011).
  149. Falgout, R. D., Jones, J. E. & Yang, U. M. in *Numerical solution of partial differential equations on parallel computers* 267–294 (Springer, 2006).
  150. Fletcher, T. A. *et al.* Resin treatment of free edges to aid certification of through thickness laminate strength. *Composite Structures* **146**, 26–33 (2016).
  151. Camanho, P. P., Davila, C. G. & De Moura, M. F. Numerical simulation of mixed-mode progressive delamination in composite materials. *Journal of composite materials* **37**, 1415–1438 (2003).
  152. ASTM D6415 / D6415M-06a(2013). *Standard test method for measuring the curved beam strength of a fiber-reinforced polymer-matrix composite* 2013.
-

- 
153. Amestoy, P. R., Guermouche, A., L'Excellent, J.-Y. & Pralet, S. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing* **32**, 136–156 (2006).
  154. Gee, M. W. *et al.* *ML 5.0 smoothed aggregation user's guide* tech. rep. (Technical Report SAND2006-2649, Sandia National Laboratories, 2006).
  155. Belnoue, J. P.-H. *et al.* Understanding and predicting defect formation in automated fibre placement pre-preg laminates. *Composites Part A: Applied Science and Manufacturing* **102**, 196–206 (2017).
  156. Sandhu, A., Reinarz, A. & Dodwell, T. A Bayesian framework for assessing the strength distribution of composite structures with random defects. *Composite Structures* **205**, 58–68 (2018).
  157. Dillinger, J., Abdalla, M., Klimmek, T. & Gurdal, Z. Stiffness Optimization of Composite Wings with Aeroelastic Constraints. *Journal of Aircraft* **50**, 1159–1168 (July 2013).
  158. Christie, M. A. & Blunt, M. J. Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques (Aug. 2001).
  159. Bastian, P., Müller, E. H., Müthing, S. & Piatkowski, M. Matrix-free multigrid block-preconditioners for higher order Discontinuous Galerkin discretisations. *arXiv e-prints*, arXiv:1805.11930 (May 2018).
  160. Liu, J. S. *Monte Carlo strategies in scientific computing* (Springer Science & Business Media, 2008).
  161. Smith, R. *et al.* in *52nd Annual Conference of the British Institute of Non-Destructive Testing 2013, NDT 2013* 198–209 (British Institute of Non-Destructive Testing, Jan. 2013).
  162. Smith, R. *et al.* Progress in 3D characterisation and modelling of monolithic carbon-fibre composites. *Insight-Non-Destructive Testing and Condition Monitoring* **57**, 131–139 (2015).
  163. Smith, R. A., Nelson, L. J., Mienczakowski, M. J. & Challis, R. E. Automated analysis and advanced defect characterisation from ultrasonic scans of composites. *Insight-Non-Destructive Testing and Condition Monitoring* **51**, 82–87 (2009).
  164. Creighton, C. J., Sutcliffe, M. P. F. & Clyne, T. W. A multiple field image analysis procedure for characterisation of fibre alignment in composites. *Composites:Part A* **32**, 221–229 (2001).
-

- 
165. ASTM D6272-17. Standard test methods for flexural properties of unreinforced and reinforced plastics and electrical insulating materials by four point bending. *Annual book of ASTM Standards* (2017).
  166. mathworks, T. MATLAB User's Guide. *Inc., Natick, MA* **5**, 333 (1998).
  167. Camanho, P. P., Davila, C. G. & de Moura, M. F. Numerical Simulation of Mixed-Mode Progressive Delamination in Composite Materials. *Journal of Composite Materials* **37**, 1415–1438 (2003).
  168. Dodwell, T. Internal wrinkling instabilities in layered media. *Philosophical Magazine* **95**, 3225–3243 (2015).
  169. Hou, T. Y. & Wu, X.-H. A Multiscale Finite Element Method for Elliptic Problems in Composite Materials and Porous Media. *Journal of Computational Physics* **134**, 169–189 (1997).
  170. Babuška, I. & Osborn, J. E. Generalized finite element methods: their performance and their relation to mixed methods. *SIAM Journal on Numerical Analysis* **20**, 510–536 (1983).
  171. Babuška, I., Caloz, G. & Osborn, J. E. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM Journal on Numerical Analysis* **31**, 945–981 (1994).
  172. Aarnes, J. & Efendiev, Y. Mixed Multiscale Finite Element Methods for Stochastic Porous Media Flows. *SIAM Journal on Scientific Computing* **30**, 2319–2339 (2008).
  173. Cui, T., Fox, C. & O'Sullivan, M. Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm. *Water Resources Research* **47** (2011).
  174. Berger, J., Dawid, A., Smith, D. & West, M. *Markov chain Monte Carlo-based approaches for inference in computationally intensive inverse problems in Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting* **181** (2003).
  175. Haario, H. *et al.* Markov chain Monte Carlo methods for high dimensional inversion in remote sensing. *Journal of the Royal Statistical Society: series B (statistical methodology)* **66**, 591–607 (2004).
  176. Marzouk, Y. M. & Najm, H. N. Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. *Journal of Computational Physics* **228**, 1862–1902 (2009).
-



- 
177. Bayarri, M. J. *et al.* Predicting vehicle crashworthiness: Validation of computer models for functional and hierarchical data. *Journal of the American Statistical Association* **104**, 929–943 (2009).
  178. Galbally, D., Fidkowski, K., Willcox, K. & Ghattas, O. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International journal for numerical methods in engineering* **81**, 1581–1608 (2010).
  179. Lipponen, A., Seppanen, A. & Kaipio, J. P. Electrical impedance tomography imaging with reduced-order model based on proper orthogonal decomposition. *Journal of Electronic Imaging* **22**, 023008 (2013).
  180. Lieberman, C., Willcox, K. & Ghattas, O. Parameter and state model reduction for large-scale statistical inverse problems. *SIAM Journal on Scientific Computing* **32**, 2523–2542 (2010).
  181. Wang, J. & Zabaras, N. Using Bayesian statistics in the estimation of heat source in radiation. *International Journal of Heat and Mass Transfer* **48**, 15–29 (2005).
  182. Girolami, M. & Calderhead, B. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73**, 123–214 (2011).
  183. Silling, S. A. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids* **48**, 175–209 (2000).
  184. Silling, S. A. & Askari, E. A meshfree method based on the peridynamic model of solid mechanics. *Computers & structures* **83**, 1526–1535 (2005).
  185. Silling, S. A. *et al.* Peridynamic states and constitutive modeling. *Journal of Elasticity* **88**, 151–184 (2007).
  186. Seleson, P. & Parks, M. On the role of the influence function in the peridynamic theory. *International Journal of Multiscale Computational Engineering* **9**, 689–706 (2011).
  187. Parks, M. L., Lehoucq, R. B., Plimpton, S. J. & Silling, S. A. Implementing peridynamics within a molecular dynamics code. *Computer Physics Communications* **179**, 777–783 (2008).