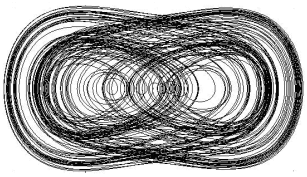
Advanced Stochastic Finite Element Simulations and Reliability Analyses



Dimitris G. Giovanis
School of Civil Engineering
National Technical University of Athens

Supervisor: Vissarion Papadopoulos

A thesis submitted for the degree of
PhilosophiæDoctor (PhD)

April, 2014

1. Supervisor: Vissarion Papadopoulos

2. Member of the committee: Manolis Papadrakakis

3. Member of the committee: Constantinos Trezos

4. Member of the committee: George Deodatis

5. Member of the committee: Constantinos Papadimitriou

6. Member of the committee: Dimitrios Koutsogiannis

7. Member of the committee: Dimitrios Vamvatsikos

Στους γονείς μου

Che Fece … Il Gran Rifiuto

Σε μερικούς ανθρώπους ανθρώπους έρχεται μια μέρα που πρέπει το μεγάλο Ναι ή το μεγάλο Όχι να πούνε. Φανερώνεται αμέσως όποιος το’χει έτοιμο μέσα του το Ναι, και λέγοντάς το πέρα πηγαίνει στην τιμή και στην πεποίθησή του. Ο αρνηθείς δεν μετανοιώνει. Αν ρωτιούνταν πάλι, όχι θα’ξανάλεγε. Κι όμως τον καταβάλλει εκείνο το όχι—το σωστό—εις όλην τη ζωή του.

Κ. Π. Καβάφης (1901)

For some people the day comes when they have to declare the great Yes or the great No. Its clear at once who has the Yes ready within him; and saying it, he goes from honor to honor, strong in his conviction. He who refuses does not repent. Asked again, hed still say no. Yet that nothe right no drags him down all his life.

C. P. Cavafy (1901)

Acknowledgements

It would not have been possible to write this doctoral thesis without the help and support of the kind people around me. Above all, I would like to thank my parents, for they have given me their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice. The sacrifices that you have made on my behalf and your prayer for me was what sustained me thus far.

This thesis would not have been possible without the help, support, patience and friendship of my supervisor, Ass. Prof. Vissarion Papadopoulos, not to mention his unsurpassed knowledge of stochastic mechanics. You have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Your advice both on research and on personal level have been priceless. Thank you.

I would like to thank Prof. Manolis Papadrakakis for giving me the opportunity to work in the institute of static and anti-seismic research (ISAAR). The belief, advice and support throughout my diploma, M.Sc and Ph.d have been invaluable for which i am extremely grateful. Special thanks to Ass.Prof. Costas Trezos for being the third member of my committee.

I am most grateful to Dr. George Stefanou for encouraging me during my M.Sc to pursue a doctoral degree and for all the fruitful conversations we had regarding stochastic mechanics. His insight in the topic is of a great importance. I would also like to thank Dr George Markou for all the help he provided me in the first years of my Ph.d. Thanks to all my colleagues for their support and cooperation.

Finally, I gratefully acknowledge the funding received towards my Ph.d from the European Research Council Advanced Grant MASTER Ű- Mastering the computational challenges in numerical modeling and optimum design of CNT reinforced composites.

Abstract

It is common knowledge that the analysis of structural systems with ambiguous characteristics has been the focal point of research work over the course of the last two decades. Epistemic uncertainty can be taken into consideration in simulation by setting the problem in a probabilistic framework, resulting to governing equations that are, in general, stochastic partial differential equations. Throughout this period of time, a plethora of stochastic finite element methodologies (SFEM) for the numerical solution of the stochastic partial differential equations involved in such problems have been developed. It could be argued that one of the most blurry areas in these methodologies is the computational effort involved. This thesis presents a series of methodologies that have been implemented in the framework of SFEM and reliability analysis in order to reduce the computational effort involved.

The first methodology is a neural network-based subset simulation in which neural networks are effectively trained and then used as robust meta-models in order to increase the efficiency of subset simulation by increasing significantly the samples per level with a minimum additional computational effort. In the second methodology neural networks are used in the framework of MCS for computing the response statistics (moments and pdf) as well as the reliability of stochastic structural systems, whose material properties are described as random fields, by providing robust neural network estimates of the structural response. The third methodology consists of constructing an adaptive sparse polynomial chaos (PC) expansion of the response of stochastic systems in the framework of spectral stochastic finite element method (SSFEM). The proposed methodology utilizes the concept of variability response function (VRF) in order to compute an a priori low cost estimation of the spatial distribution of the second-order error of the response as a function of the number of terms used in the truncated Karhunen-Loève (KL) series representation of the random field involved in the problem. Finally, a parametric study of Monte Carlo simulation versus SSFEM in large-scale systems is performed.

ΠΕΡΙΛΗΨΗ ΤΗΣ
ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ

με τίτλο

'Προχωρημένες μέθοδοι προσομοίωσης
στοχαστικών πεπερασμένων στοιχείων
και αξιοπιστίας των κατασκευών'

Οι σύγχρονες τάσεις ανάλυσης προβλημάτων της επιστήμης του μηχανικού εστιάζονται συνήθως στην αύξηση της ακρίβειας του αριθμητικού προσομοιώματος. Επικρατεί δηλαδή η άποψη ότι, ένα λεπτομερές προσομοίωμα με πυκνό δίκτυο διακριτοποίησης με πεπερασμένα στοιχεία, με μεγάλο πλήθος βαθμών ελευθερίας ανταποκρίνεται πιστότερα στην πραγματική συμπεριφορά του φυσικού προβλήματος. Ωστόσο, οι συγκρίσεις των αποτελεσμάτων των αναλύσεων με τις πειραματικές μετρήσεις δεν επιβεβαιώνουν την άποψη αυτή. Ένα λεπτομερές προσομοίωμα πεπερασμένων στοιχείων ελαττώνει απλώς το σφάλμα διακριτοποίησης. Η ύπαρξη άλλων παραμέτρων που οδηγούν στην ασυμφωνία μεταξύ πρόβλεψης (ανάλυσης) και μέτρησης (πείραμα) δεν λαμβάνονται υπόψη. Αυτό συμβαίνει για τους εξής λόγους:

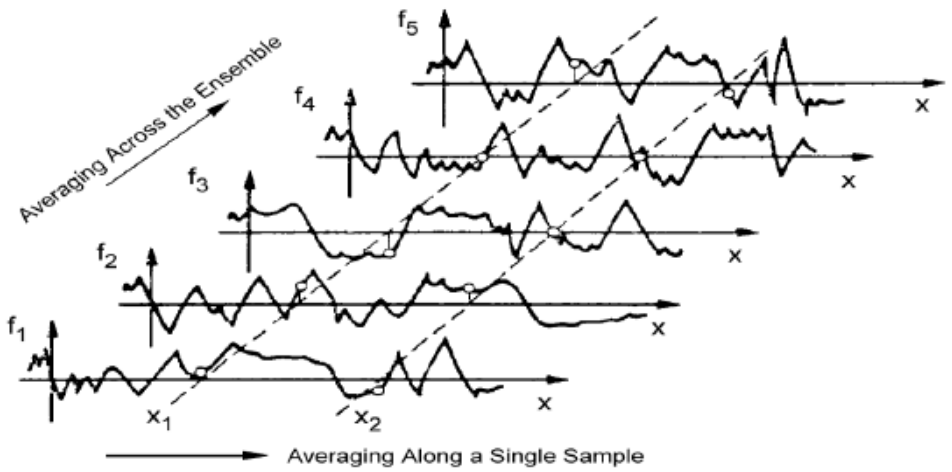
a) Το εξιδανικευμένο αριθμητικό προσομοίωμα των πεπερασμένων στοιχείων δεν απεικονίζει με ακρίβεια την πραγματική συμπεριφορά του φορέα (μη γραμμικά φαινόμενα αλληλεπίδρασης, ευκαμψίες κόμβων, συνοριακές συνθήκες).
b) Οι παράμετροι που καθορίζουν την απόκριση της κατασκευής είναι συνήθως αβέβαιες (μέτρο ελαστικότητας, γεωμετρικές ατέλειες, στατικά και δυναμικά φορτία).
c) Τα αποτελέσματα των πειραματικών μετρήσεων υπόκεινται σε ανεξέλεγκτα τυχαία φαινόμενα τα οποία εμποδίζουν την αναπαραγωγή και επιβεβαίωση όλων των μετρήσεων

Από τα παραπάνω γίνεται προφανές ότι ένα λεπτομερέστερο προσομοίωμα πεπερασμένων στοιχείων δεν εξασφαλίζει την αύξηση της αξιοπιστίας της ανάλυσης. Αντίθετα, έχει μεγάλη σημασία η προσομοίωση των τυχαίων φαινομένων που επηρεάζουν την απόκριση της κατασκευής σε συνδυασμό με ένα αξιόπιστο αριθμητικό προσομοίωμα της κατασκευής. . Η στοχαστική ανάλυση, στην ευρεία έννοια, αφορά στον άμεσο χειρισμό αβέβαιων παραμέτρων, η αβεβαιότητα των οποίων δεν λαμβάνεται υπόψη στις συμβατικές νομοτελειακά καθορισμένες προσδιοριστικές (deterministic) θεωρήσεις.

Η επιρροή των αβεβαιοτήτων στο σχεδιασμό των κατασκευών αποτελεί ένα σημαντικό πεδίο με συνεχώς αναπτυσσόμενο ενδιαφέρον. Ενώ οι περισσότεροι μοντέρνοι σχεδιαστικοί κώδικες στηρίζονται σε συντελεστές ασφαλείας, η στοχαστική μοντελοποίηση των αβεβαιοτήτων που αφορούν στη φόρτιση, τη

γεωμετρία και τις ιδιότητες των υλικών γίνεται όλο και περισσότερο θελκτική, οδηγώντας σε πιο ρεαλιστικές εκτιμήσεις της ασφάλειας και της αξιοπιστίας των κατασκευών. Αρκετές πιθανοτικές μέθοδοι έχουν προταθεί στο παρελθόν, με την απλούστερη από όλες να είναι εκείνη στην οποία οι αβεβαιότητες περιγράφονται από ένα σετ συσχετισμένων τυχαίων μεταβλητών, όπου κάθε τυχαία μεταβλητή αναπαριστά ιδιότητα υλικού, συντελεστή φόρτισης ή γεωμετρική παράμετρο. Σε αρκετές όμως εφαρμογές, η περιγραφή αυτή κρίνεται ανεπαρκής λόγω του γεγονότος ότι αρκετές φυσικές ποσότητες που συσχετίζονται με το πρόβλημα ενδέχεται να μεταβάλλονται τυχαία είτε στο χρόνο είτε στο χώρο. Η πιθανοτική περιγραφή τέτοιων ποσοτήτων απαιτεί τη χρησιμοποίηση της θεωρίας των στοχαστικών πεδίων.

Στη θεωρία των στοχαστικών πεδίων θεωρείται ότι η διακύμανση μίας παραμέτρου, π.χ. του μέτρου ελαστικότητας, περιγράφεται από μία συνάρτηση μίας ή περισσοτέρων μεταβλητών (στοχαστική διαδικασία ή πεδίο, Σχήμα 1).



<u>Σχήμα 1</u>: Μέσος όρος στοχαστικού πεδίου σε δύο σημεία κατά μήκος 5 δειγματοσυναρτήσεων

Ο αριθμός των μεταβλητών εξαρτάται από το είδος του προβλήματος (μονοδιάστατη, διδιάστατη, τριδιάστατη ένταση) και από το είδος της ανάλυσης (στατική, δυναμική). Για την προσομοίωση των στοχαστικών συναρτήσεων, απαιτείται ο προσδιορισμός της κατανομής των τυχαίων μεταβλητών (π.χ. Gauss), της μέσης τιμής, της διασποράς καθώς και της συσχέτισης (correlation) μεταξύ όλων των τιμών της στοχαστικής διαδικασίας ή του πεδίου. Η ποσοτικοποίηση της αβεβαιότητας στις τιμές των παραμέτρων με χρήση της μαθηματικής θεωρίας των στοχαστικών συναρτήσεων (διαδικασιών/πεδιών) αποτελεί πεδίο εκτεταμένων ερευνών τις τελευταίες δύο δεκαετίες. Η πλειοψηφία των μεθόδων αυτών έχει επικεντρωθεί στην ανάπτυξη στοχαστικών μεθόδων πεπερασμένων στοιχείων (ΣΠΣ) για την αριθμητική επίλυση

των στοχαστικών μερικών διαφορικών εξισώσεων που κυβερνούν τέτοια προβλήματα. Ανάμεσα σε αυτές οι περισσότερο γνωστές είναι οι μη-διεισδυτικές Monte Carlo μέθοδοι.

Οι μέθοδοι Monte Carlo είναι μια κατηγορία υπολογιστικών αλγορίθμων που στηρίζονται σε επαναλαμβανόμενες τυχαίες δειγματοληψίες για τον υπολογισμό των αποτελεσμάτων τους. Ακριβής υπολογισμός της διακύμανσης της απόκρισης οποιουδήποτε συστήματος, υπό την προϋπόθεση ότι η προσδιοριστική λύση του προβλήματος είναι γνωστή μπορεί να πραγματοποιηθεί μόνο μέσω της άμεσης προσομοίωσης Monte Carlo (brute force Monte Carlo simulation (MCS)). Στην πραγματικότητα, η μέθοδος αυτή είναι η μόνη ικανή να χειριστεί στοχαστικά προβλήματα στα οποία εμπλέκονται μη γραμμικότητες, δυναμικές φορτίσεις, προβλήματα αστάθειας, παραμετρικές διεγέρσεις κλπ. Γι' αυτό και χρησιμοποιείται ως μέθοδος αναφοράς με τα αποτελέσματα της οποίας συγκρίνονται τα αποτελέσματα άλλων μεθόδων. Το μειονέκτημα της μεθόδου είναι ότι συνήθως είναι υπερβολικά απαιτητική σε υπολογιστική ισχύ και χρόνο εκτελέσεως. Παρόλο που οι εξελίξεις που έχουν επιτευχθεί σε θεωρητικό επίπεδο, έχουν φτάσει σε ένα ικανοποιητικό σημείο, ανυπέρβλητες δυσκολίες συναντώνται κατά την επίλυση ρεαλιστικών προβλημάτων με τη μέθοδο προσομοίωσης MCS. Το μειονέκτημα της άμεσης μεθόδου MCS είναι περισσότερο εμφανές στην περίπτωση της ανάλυσης αξιοπιστίας ρεαλιστικών και πολύπλοκων συστημάτων, αφού για κάθε προσομοίωση απαιτείται η πραγματοποίηση μιας πλήρους μη γραμμικής ανάλυσης του συστήματος, καθώς επίσης και για τον υπολογισμό μικρών πιθανοτήτων αστοχίας ($\leq 10^{-4}$) όπου απαιτείται αρκετά μεγάλος αριθμός δειγμάτων ($\sim 10^{6}$).

Για την αντιμετώπιση του υπολογιστικού κόστους που έχει η MCS, διάφορες μέθοδοι αναπτύχθηκαν στο πλαίσιο της ανάλυσης της αξιοπιστίας των κατασκευών με στόχο τη μείωση του αριθμού των προσομοιώσεων που απαιτούνται για τον ακριβή υπολογισμό της πιθανότητας αστοχίας. Τέτοιες μέθοδοι είναι η μέθοδος προσομοίωσης των υποσυνόλων (subset simulation), η μέθοδος δειγματοληψίας σημασίας (importance sampling) καθώς επίσης και η μέθοδος προσαρμοστικής δειγματοληψίας (adaptive sampling). Επίσης, τα νευρωνικά δίκτυα έχουν εφαρμοσθεί ως μετά-μοντέλα με επιτυχία στο πλαίσιο μεθόδων εκτίμησης της αξιοπιστίας των κατασκευών αντικαθιστώντας το μοντέλο πεπερασμένων στοιχείων.

Τις τελευταίες δεκαετίες αναπτύχθηκε η Μέθοδος των Φασματικών Στοχαστικών Πεπερασμένων Στοιχείων (ΦΣΠΣ) η οποία προτάθηκε από τους

Ghanem and Spanos και βασίζεται στην ταυτόχρονη διακριτοποίηση της "τυχαίας διάστασης" (random space) του συστήματος χρησιμοποιώντας αναπτύγματα σε σειρά βασικών ποσοτήτων του προβλήματος. Η διαδικασία αυτή οδηγεί τελικά σε ένα γραμμικό σύστημα εξισώσεων διαστάσεων (NP)×(NP), όπου N είναι το πλήθος των βαθμών ελευθερίας του ντετερμινιστικού προβλήματος και P είναι ο αριθμός των όρων που χρησιμοποιείται για τη διακριτοποίηση της τυχαίας διάστασης. Στο πλαίσιο της μεθόδου αυτής, τα στοχαστικά πεδία *f(x)* που περιγράφουν τις αβέβαιες παραμέτρους, στην περίπτωση που οι αυτές ακολουθούν την Γκαουσιανή κατανομή, προσομοιώνονται με το ανάπτυγμα Karhunen-Loeve

$$f(x) = \sum_{n=1}^{M} \sqrt{\lambda_n} \cdot \varphi_n(x) \cdot \xi_n \tag{1}$$

ενώ κάθε τυχαία ποσότητα της απόκρισης *Y* του συστήματος αντιπροσωπεύεται από τις συντεταγμένες της σε μία κατάλληλη βάση του χώρου των τυχαίων μεταβλητών, το πολυωνυμικό χάος (polynomial chaos) που εισήγαγε ο N. Wiener.

$$Y = \sum_{j=0}^{P-1} y_j \Psi_j(\boldsymbol{X}) \tag{2}$$

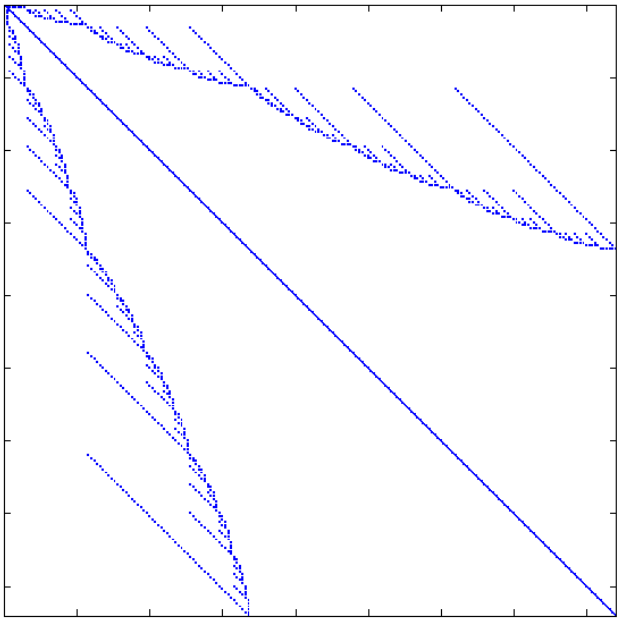
Τα αναπτύγματα πεπερασμένου πλήθους όρων που χρησιμοποιούνται εισάγουν ένα σφάλμα προσέγγισης. Η βέλτιστη προσέγγιση της ακριβούς λύσης του προβλήματος στον χώρο που ορίζεται από τα πολυώνυμα χάους επιτυγχάνεται με την ελαχιστοποίηση του σφάλματος κατά τη μέση τετραγωνική έννοια (μέθοδος Galerkin). Δηλαδή, επιθυμούμε το σφάλμα να είναι ορθογωνικό ως προς τη πολυωνυμική βάση στην οποία δημιουργείται η λύση

$$\begin{aligned} e_{M,p} &= \sum_{i=0}^{M+1} \sum_{j=0}^{P-1} \mathbf{K}_i \xi_i(\theta) \mathbf{U}_j(\theta) \Psi_j(\theta) - \mathbf{F} \\ E\left[e_{M,p} \cdot \Psi_k \right] &= 0 \end{aligned} \tag{3}$$

Προκειμένου να περιληφθούν και μη Γκαουσιανά πεδία στο πλαίσιο της μεθόδου, έχει προταθεί η χρήση του αναπτύγματος πολυωνυμικού χάους για την προσομοίωση των τυχαίων παραμέτρων του προβλήματος.

$$\left(\sum_{i=0}^{P-1}\mathbf{K}_i\Psi_i(\theta)\right)\cdot\left(\sum_{j=0}^{P-1}\mathbf{U}_j\Psi_j(\theta)\right)-\mathbf{F}=0, \tag{4}$$

Ωστόσο, η διπλή χρήση του πολυωνυμικού χάους για την περιγραφή των αβέβαιων (μηχανικών και γεωμετρικών) ιδιοτήτων και της απόκρισης του συστήματος επιβαρύνει την ακρίβεια της μεθόδου. Αυτό έχει ως αποτέλεσμα, στη μέθοδο των φασματικών πεπερασμένων στοιχείων το υπολογιστικό κόστος για την επίλυση του διευρυμένου (στοχαστικού) συστήματος εξισώσεων να είναι κατά P τάξεις μεγέθους μεγαλύτερο από εκείνο της προσδιοριστικής ανάλυσης του ίδιου προβλήματος.



Σχήμα 2: Διευρυμένο στοχαστικό μητρώο της μεθόδου των φασματικών στοχαστικών πεπερασμένων στοιχείων στην περίπτωση Γκαουσιανού πεδίου

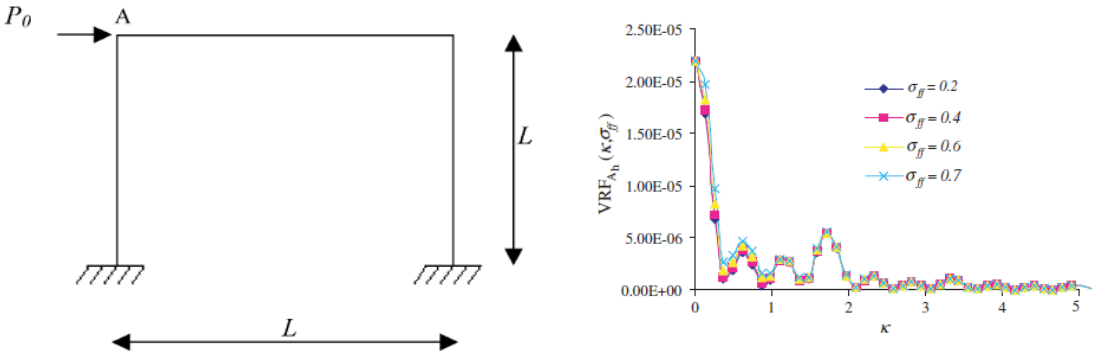
Ιδιαίτερα στην περίπτωση χρησιμοποίησης διπλού πολυωνυμικού χάους το μητρώο είναι πού πιο πυκνό από το αντίστοιχο Γκαουσιανο. Κατά συνέπεια, η χρησιμοποίηση της μεθόδου των φασματικών στοχαστικών πεπερασμένων στοιχείων απαιτεί σημαντικά μεγαλύτερες υπολογιστικές δυνατότητες συγκριτικά με εκείνες που χρειάζονται για την κλασική προσδιοριστική ανάλυση. Για αυτό τον λόγο, η μέθοδος αυτή έχει περιοριστεί πρακτικά έως τώρα σε προβλήματα μικρής στοχαστικής διάστασης. Συνήθως, απαιτούνται από 2-3 έως μερικές δεκάδες

συντελεστών για τη διακριτοποίηση της "τυχαίας διάστασης" 'και για την ακριβή πιθανοτική περιγραφή της κάθε ποσότητας της απόκρισης, χωρίς αυτός ο αριθμός να είναι εκ των προτέρων γνωστός. Σε μια προσπάθεια μείωσης της διάστασης αλλά και της πυκνότητας του διευρυμένου μητρώου της μεθόδου αυτής προκύπτει η ανάγκη μίας προσαρμοστικής διατύπωσης της μεθόδου των φασματικών στοχαστικών πεπερασμένων στοιχείων. Διάφορες μέθοδοι έχουν διατυπωθεί τα τελευταία χρόνια, οι οποίες αντιμετωπίζουν το πρόβλημα αυτό με επιτυχία.

Ένας άλλος παράγοντας που επηρεάζει την απόδοση την προαναφερθέντων στοχαστικών μεθόδων είναι ότι η γνώση της συνάρτησης διακύμανσης καθώς επίσης και η γνώση της συνάρτηση κατανομής της πιθανότητας των στοχαστικών μεταβλητών που περιγράφουν το στοχαστικό πρόβλημα είναι απαραίτητη για τη την εκτίμηση της διακύμανσης της απόκρισης του. Συνήθως όμως η έλλειψη πειραματικών δεδομένων για την ποσοτικοποίησης των διαφόρων αβέβαιων παραμέτρων οδηγεί σε αναλύσεις ευαισθησίας της κατασκευής για τις διάφορες στοχαστικές παραμέτρους. Σε αυτή την περιπτώσεις, έχουμε δυσανάλογη αύξηση του υπολογιστικού κόστους και έλλειψη πως αυτές οι παράμετροι ελέγχουν τη διακύμανση της απόκρισης του συστήματος. Στο πλαίσιο αυτό η ιδέα της συνάρτησης διακύμανσης της απόκρισης (variability response function) προτάθηκε στα τέλη της δεκαετίας του 80. Με βάση τη μέθοδο αυτή η διακύμανση της απόκρισης μιας κατασκευής δίνεται από την παρακάτω σχέση

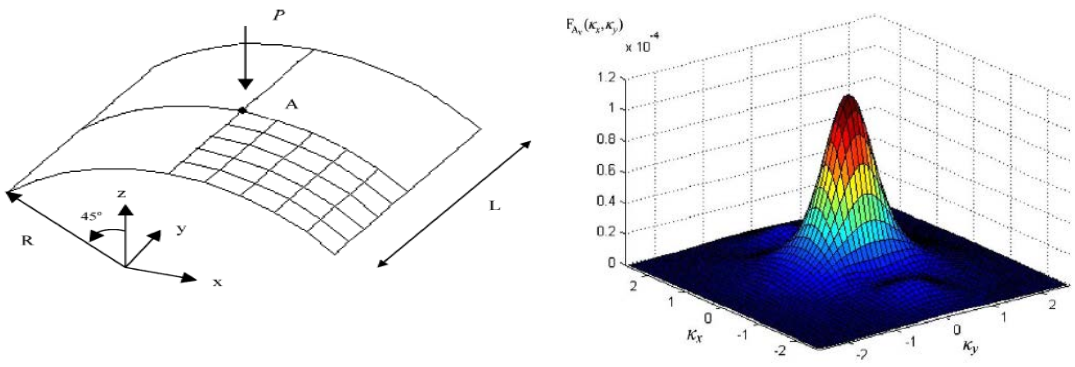
$$\mathrm{Var}(u,\mathbf{x}) = \int_{-\infty}^{\infty} \mathrm{VRF}(\boldsymbol{\kappa},\mathbf{x},\sigma_f)\cdot S_{ff}(\boldsymbol{\kappa})d\boldsymbol{\kappa} \tag{5}$$

VRF είναι η συνάρτηση διακύμανση της απόκρισης η οποία εξαρτάται από ντετερμινιστικά στοιχεία του συστήματος και από την διακύμανση σ του στοχαστικού πεδίου. Όπως προκύπτει επίσης η συνάρτηση διακύμανσης της απόκρισης είναι ανεξάρτητη της μορφής που έχει η συνάρτηση αυτοσυσχέτισης του στοχαστικού πεδίου.

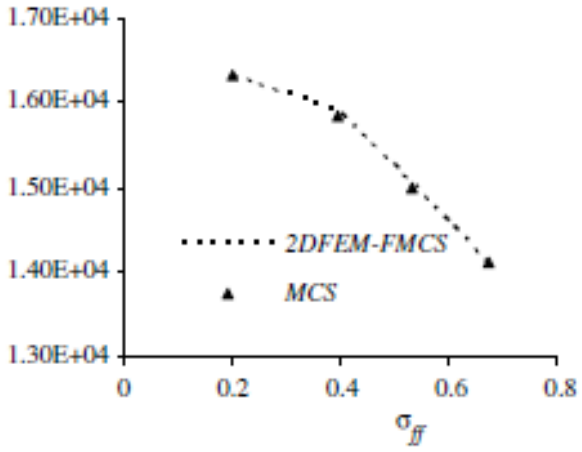


Σχήμα 3: Τιμές της συνάρτησης διακύμανσης της απόκρισης για διάφορες τιμές της τυπικής απόκλισης σ του πεδίου

Όπως έχει αποδειχτεί η συνάρτηση διακύμανσης της απόκρισης ορίζεται με κλειστή ολοκληρωτική σχέση μόνο σε γραμμικά στατικά στοχαστικά συστήματα με ντετερμινιστική φόρτιση καθώς επίσης για στατιστικά ομογενή στοχαστικά πεδία. Για τη εύρεση της συνάρτησης διακύμανσης της απόκρισης απαιτείται η διαδικασία της γρήγορης Monte Carlo προσομοίωσης η οποία παράγει συνημιτονοειδής πραγματοποιήσεις του στοχαστικού πεδίου.



Σχήμα 4: Συνάρτηση διακύμανσης της απόκρισης για δισδιάστατο στοχαστικό πεδίο



Σχήμα 5: Σύγκριση των μεθόδων άμεσης προσομοίωσης Monte Carlo (MCS) με την γρήγορη Monte Carlo (FMCS)

Στο πρώτο μέρος της διδακτορικής διατριβής πραγματοποιείται μία παραμετρική διερεύνηση της συμπεριφοράς της μεθόδου των Φασματικών πεπερασμένων στοιχείων για διάφορες τιμές παραμέτρων του στοχαστικού πεδίου όπου αποτιμούνται τα πλεονεκτήματα και μειονεκτήματα της, σε συνάρτηση με τον αριθμό των όρων που απαιτούνται για τη σύγκλιση της μεθόδου, σε σύγκριση με τη μέθοδο Monte Carlo, η οποία θεωρείται ως μέθοδος αναφοράς. Σε μια δεύτερη φάση της παρούσας διδακτορικής διατριβής αναπτύσσεται μια μεθοδολογία για τον υπολογισμό της πιθανότητας αστοχίας ενός στοχαστικού συστήματος η οποία χρησιμοποιεί τα τεχνητά νευρωνικά δίκτυα στο πλαίσιο της ανάλυσης αξιοπιστίας με τη μέθοδο των υποσυνόλων (Subset Simulation) καθώς και επίσης στο πλαίσιο της μεθόδου Monte Carlo στην περίπτωση όπου η ποσοτικοποίηση της αβεβαιότητας των τυχαίων παραμέτρων γίνεται με τη χρήση της θεωρίας των στοχαστικών πεδίων και ειδικότερα της μεθόδου φασματικής απεικόνισης. Στην πρώτη περίπτωση (Subset Simulation), η βασική ιδέα της μεθόδου είναι η σταδιακή εκπαίδευση των νευρωνικών δικτύων σε κάθε υποχωρίο (subset) ούτως ώστε να χρησιμοποιηθούν για τον εμπλουτισμό της περιοχής αστοχίας δίχως πρόσθετο υπολογιστικό κόστος. Για την προσομοίωση των δειγμάτων που χρησιμοποιούνται για την εκπαίδευση των νευρωνικών δικτύων επιλέγεται η τεχνική της δειγματοληψίας του λατινικού υπερκύβου, η οποία έχει το πλεονέκτημα ότι όλη η περιοχή της κατανομής που χρησιμοποιείται προσομοιώνεται. Ειδικότερα στην περίπτωση όπου οι τυχαίες μεταβλητές ακολουθούν την ομοιόμορφη κατανομή η δειγματοληψία γίνεται με τέτοιο τρόπο ούτως ώστε όλος ο χώρος να προσομοιώνεται δίχως την ύπαρξη του φαινόμενου της υπερκάλυψης διαφόρων περιοχών του.

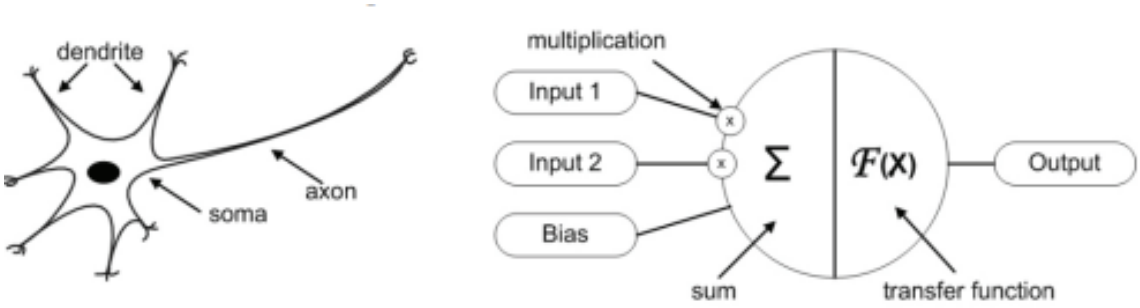
Σε επόμενο στάδιο της διδακτορικής διατριβής τα νευρωνικά δίκτυα χρησιμοποιούνται ως μετά-μοντέλα για την εκτίμηση της συνάρτησης κατανομής πιθανότητας της απόκρισης ενός συστήματος καθώς επίσης για την ανάλυση της αξιοπιστίας των κατασκευών, στο πλαίσιο της προσομοίωσης Monte Carlo. Στα πλαίσια της προτεινόμενης μεθοδολογίας η δημιουργία πραγματοποιήσεων (προσομοίωση) του στοχαστικού πεδίου γίνεται με τη μέθοδο της φασματικής απεικόνισης, της οποίας οι ανεξάρτητες τυχαίες γωνίες φάσης χρησιμοποιούνται ως διάνυσμα εισόδου για την εκπαίδευση του νευρωνικού δικτύου, επιτυγχάνοντας με αυτό τον τρόπο σημαντική μείωση της διάστασης του διανύσματος αυτού σε σχέση με τις έως τώρα προταθείσες μεθοδολογίες στις οποίες το διάνυσμα εισόδου περιέχει τις τιμές των τυχαίων μεταβλητών που χρησιμοποιούνται για την περιγραφή των αβέβαιων παραμέτρων του συστήματος ως στοχαστικού πεδίου. Με αυτό τον τρόπο αυξάνεται η αξιοπιστία της εκτίμησης του νευρωνικού δικτύου.

Στην συνέχεια πραγματοποιείται μια προσαρμοστική διατύπωση των φασματικών στοχαστικών πεπερασμένων στοιχείων (spectral stochastic finite element method) με μεθόδους Galerkin χρησιμοποιώντας τη μέθοδο διακύμανσης της απόκρισης (Variability response function) για την εύρεση της χωρική κατανομή των όρων του αναπτύγματος Karhunen-Loeve. Η κατανομή των όρων πραγματοποιείται με τέτοιο τρόπο ώστε το δευτέρας τάξεως σφάλμα να είναι ομοιόμορφα κατανεμημένο. Η διατύπωση αυτή οδηγεί σε σημαντικά αραιότερα διευρυμένα μητρώα με αποτέλεσμα τη βελτίωση της υπολογιστική επίδοσης της μεθοδου. Η αποτίμηση αυτής της βελτίωσης πραγματοποιείται με τη χρήση της προσταθεροποιημένης μεθόδου των συζυγών διανυσματικών κλίσεων για την επίλυση του διευρυμένου μητρώου.

Τα τεχνητά νευρωνικά δίκτυα έχουν με επιτυχία χρησιμοποιηθεί στο πλαίσιο της μεθόδου αξιοπιστίας των κατασκευών οδηγώντας σε αποδεκτές εκτιμήσεις της πιθανότητας αστοχίας αλλά με μειωμένο υπολογιστικό κόστος. Τα τεχνητά νευρωνικά δίκτυα (artificial neural networks) αποτελούν μια σχετικά νέα περιοχή στις φυσικές επιστήμες, καθόσον έχουν γίνει γνωστά και έχουν αναπτυχθεί σε διεθνές επίπεδο μόνο κατά τις τελευταίες δεκαετίες. Αρχικά προτάθηκαν ως ένα μαθηματικό μοντέλο προσομοίωσης της πολύπλοκης λειτουργίας του ανθρώπινου εγκεφάλου. Η δομή του τεχνητού νευρωνικού δικτύου μιμείται κατά το δυνατό εκείνη του βιολογικού νευρωνικού δικτύου, ώστε να εμφανίζει παρόμοιες ιδιότητες. Κατ' αναλογία επομένως με ένα δίκτυο νευρώνων εγκεφάλου, ένα τεχνητό δίκτυο

αποτελείται από ένα σύνολο τεχνητών νευρώνων που αλληλεπιδρούν, συνδεόμενοι μεταξύ τους με τις λεγόμενες συνάψεις (synapses). Ο βαθμός αλληλεπίδρασης είναι διαφορετικός για κάθε ζεύγος νευρώνων και καθορίζεται από τα λεγόμενα συναπτικά βάρη (synaptic weights). Συγκεκριμένα, καθώς το νευρωνικό δίκτυο αλληλεπιδρά με το περιβάλλον και μαθαίνει από αυτό, τα συναπτικά βάρη μεταβάλλονται συνεχώς, ενδυναμώνοντας ή αποδυναμώνοντας την ισχύ του κάθε δεσμού. Όλη η εμπειρική γνώση που αποκτά επομένως το νευρωνικό δίκτυο από το περιβάλλον κωδικοποιείται στα συναπτικά βάρη. Αυτά αποτελούν το χαρακτηριστικό εκείνο που δίνει στο δίκτυο την ικανότητα για εξέλιξη και προσαρμογή στο περιβάλλον.

Σε αναλογία με το βιολογικό νευρώνα του εγκεφάλου, ο τεχνητός νευρώνας (artificial neuron) είναι η δομική μονάδα του τεχνητού νευρωνικού δικτύου. Σε αυτόν συντελείται όλη η επεξεργασία της πληροφορίας. Κάθε νευρώνας δέχεται πληροφορία, την επεξεργάζεται και δίνει μία τιμή εξόδου. Οι είσοδοί του είναι είτε οι έξοδοι άλλων νευρώνων, είτε το πρωταρχικό σήμα εισόδου του δικτύου. Υπάρχουν διάφορα είδη νευρώνα. Το είδος που θα επιλεχθεί για να δομηθεί ένα συγκεκριμένο τεχνητό νευρωνικό δίκτυο, εξαρτάται από τη φύση του εκάστοτε προβλήματος που εξετάζουμε. Σε πολλές περιπτώσεις χρησιμοποιείται συνδυασμός διαφορετικών ειδών νευρώνα.



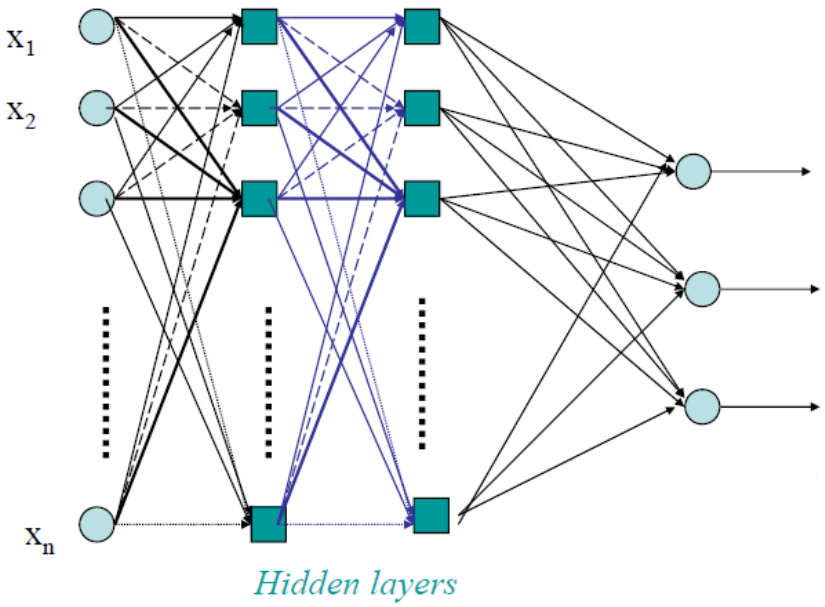
Σχήμα 6: Ανθρώπινος νευρώνας και νευρώνας τεχνητού νευρωνικού δικτύου

Ο πρωταρχικός σκοπός της λειτουργίας ενός τεχνητού νευρωνικού δικτύου είναι να μπορεί να λύνει συγκεκριμένα προβλήματα ή να επιτελεί από μόνο του ορισμένες διεργασίες. Για να γίνει αυτό, το νευρωνικό δίκτυο προηγουμένως πρέπει να εκπαιδευθεί κατάλληλα. Υπάρχουν πολλές μέθοδοι με τις οποίες μπορούμε να εκπαιδεύσουμε ένα τεχνητό νευρωνικό δίκτυο. Η κάθε μία από αυτές έχει σαφή πλεονεκτήματα και μειονεκτήματα. Καμία από αυτές δεν είναι πανάκεια. Υπάρχουν δύο βασικές μέθοδοι εκπαίδευσης:

a) με εποπτευόμενο τρόπο

b) με μη–εποπτευόμενο τρόπο (ή αυτο–εποπτευόμενο τρόπο).

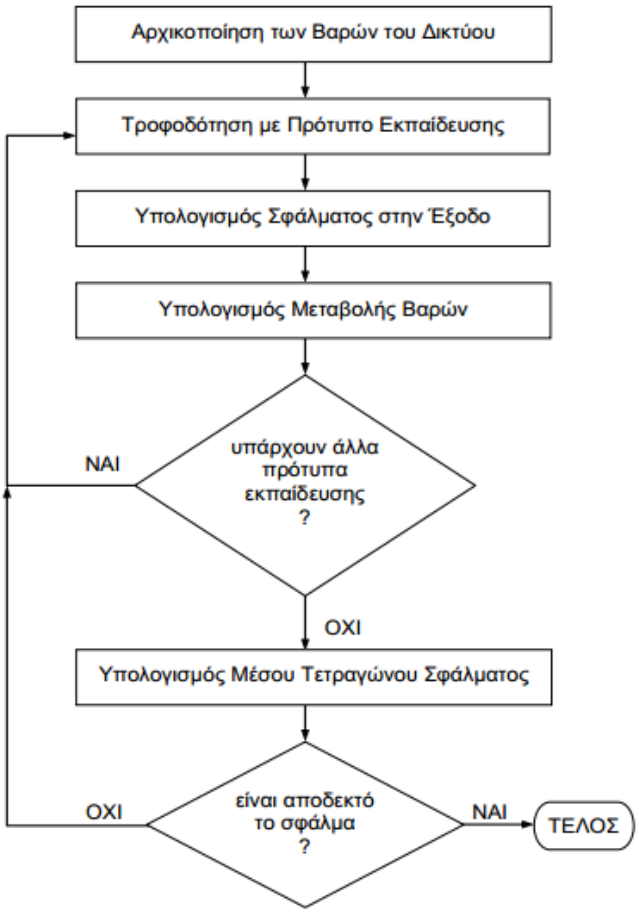
Η εποπτευόμενη μάθηση είναι και ο πιο συχνός τρόπος στην εκπαίδευση των νευρωνικών δικτύων και πραγματοποιείται με την παροχή των δεδομένων εισόδου και εξόδου (input/output) στο δίκτυο. Η ροή της πληροφορίας σε ένα τέτοιο δίκτυο γίνεται πάντα από τα αριστερά προς τα δεξιά, δεν υπάρχει κανένας βρόχος ανάδρασης. Θεωρούμε επίσης ότι, οι νευρώνες σε κάθε επίπεδο αλληλεπιδρούν μόνο με εκείνους τους νευρώνες που ανήκουν στα άμεσα γειτονικά τους επίπεδα. Δηλαδή το πρώτο κρυφό επίπεδο δέχεται τις τιμές του επιπέδου εισόδου, τα αποτελέσματα του πρώτου κρυφού επιπέδου περνάνε στο δεύτερο κρυφό, του οποίου τα αποτελέσματα στη συνέχεια περνάνε στο επίπεδο εξόδου.



Σχήμα 7: Τεχνητό νευρωνικό δίκτυο εμπρός-τροφοδότησης πολλαπλών επιπέδων

Ο αλγόριθμος οπισθοδιάδοσης του σφάλματος, είτε αυτούσιος, είτε με κάποιες παραλλαγές ή προσθήκες είναι ο πιο ευρέως διαδεδομένος αλγόριθμος. Η βασική ιδέα του αλγόριθμου οπισθοδιάδοσης είναι να προσαρμόζονται τα συναπτικά βάρη, σε κάθε βήμα της επανάληψης, έτσι ώστε η συνάρτηση της ενέργειας του σφάλματος να μειώνεται. Στην πραγματικότητα επομένως, η προσαρμογή των βαρών γίνεται σύμφωνα με τα σφάλματα που υπολογίζονται σε κάθε βήμα της διαδικασίας,

δηλαδή για κάθε δεδομένο παράδειγμα. Ο μέσος όρος της μεταβολής των βαρών σε όλο το σύνολο των σετ δεδομένων, που προκύπτει με αυτόν τον τρόπο, είναι επομένως μία εκτίμηση της μεταβολής που θα προέκυπτε, αν ελαχιστοποιούσαμε τη συνάρτηση της ενέργειας του σφάλματος όλου του συνόλου.



Σχήμα 8: Αλγόριθμος οπισθοδιάδοσης του σφάλματος

Τα τεχνητά νευρωνικά δίκτυα χρησιμοποιήθηκαν με επιτυχία σε προβλήματα που αφορούν στοχαστική ανάλυση κατασκευών σε συνδυασμό με πρώτου και δεύτερου βαθμού μεθόδους αξιοπιστίας των κατασκευών (FORM, FOSM) για τον χειρισμό των αβεβαιοτήτων. Επίσης, παρόμοιες προσεγγίσεις παρουσιάζονται στα όπου τα νευρωνικά δίκτυα χρησιμοποιήθηκαν στο πλαίσιο μεθόδων επιφάνειας της απόκρισης (response surface).

Μια από τις μεθόδους μείωσης της διασποράς είναι η μέθοδος των υποσυνόλων, η οποία χρησιμοποιείται στο πλαίσιο της μελέτης της αξιοπιστίας των κατασκευών για την εύρεση της πιθανότητας αστοχίας ενός συστήματος και ειδικότερα για τον υπολογισμό μικρών πιθανοτήτων. Με τη μέθοδο αυτή, η

πιθανότητα αστοχίας εκφράζεται ως ένα γινόμενο κάποιων υπό συνθήκη πιθανοτήτων που αντιστοιχούν σε κάποια ενδιάμεσα γεγονότα αστοχίας, ο υπολογισμός των οποίων απαιτεί την προσομοίωση ποίο συχνών γεγονότων, δηλαδή απαιτεί λιγότερες προσομοιώσεις. Κατά συνέπεια το πρόβλημα του υπολογισμού μιας μικρής πιθανότητας αστοχίας στον αρχικό πιθανοτικό χώρο αντικαθίσταται από μια αλληλουχία προσομοιώσεων πιο συχνών γεγονότων σε υπό συνθήκη πιθανοτικούς χώρους. Για τον υπολογισμό των υπό συνθήκη πιθανοτήτων απαιτείται η χρησιμοποίησης της προσομοίωσης μιας αλυσίδας Markov (Markov chain). Εάν συμβολίσουμε με F το γεγονός αστοχίας το οποίο καθορίζει την περιοχή αστοχίας στον αβέβαιο παραμετρικό χώρο, τότε μια μειούμενη αλληλουχία γεγονότων αστοχίας ορίζεται ως

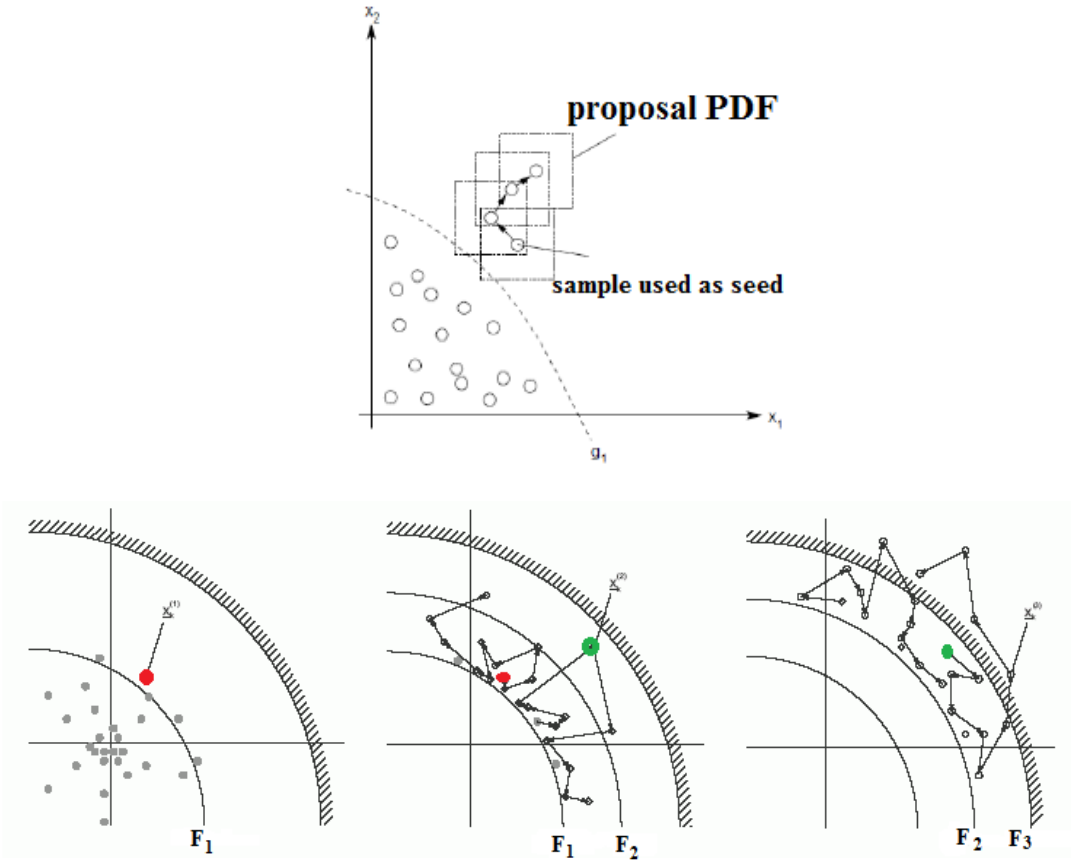
$$F_1 \supset F_2 \supset ... \supset F_m = F \tag{6}$$

Μια υπό συνθήκη πιθανότητα ορίζεται ως

$$P(F_m) = P\left(\bigcap_{i=1}^{m} F_i\right) = P(F_1)\prod_{i=1}^{m-1} P(F_{i+1} \mid F_i) \tag{7}$$

Η ιδέα της μεθόδου προσομοίωσης των υποσυνόλων είναι να υπολογίζει την πιθανότητα αστοχίας $P(F_m)$ εκτιμώντας της πιθανότητες $P(F_1)$, $P(F_{i+1}|F_i)$. Η πιθανότητα $P(F_1)$ υπολογίζεται με την κλασσική προσομοίωση Monte Carlo, ενώ οι υπόλοιπες πιθανότητες $P(F_{i+1}|F_i)$ απαιτούν τη χρησιμοποίηση της τεχνικής προσομοίωσης της αλυσίδας Markov και ειδικότερα της μεθόδου Metropolis-Hastings. Με τη μέθοδο αυτή μπορούμε να προσομοιώσουμε δείγματα από μια τυχαία συνάρτηση κατανομής πιθανότητας ως καταστάσεις μιας αλυσίδας Markov, η οποία υπό την προϋπόθεση της εργοδικότητας θα συγκλίνει στην συνάρτηση πυκνότητας πιθανότητας την οποία επιθυμούμε. Σύμφωνα με τη μέθοδο του Metropolis κάθε δείγμα κάνει μια τυχαία βόλτα στην γειτονία του η οποία εξαρτάται από μια ‘προτεινόμενη’ συνάρτηση κατανομής πιθανότητας την οποία εμείς επιλέγουμε. Το κάθε δείγμα τοποθετείται στο κέντρο της προτεινόμενης συνάρτησης κατανομής και από αυτό δημιουργείται ένα υποψήφιο δείγμα ως η επόμενη κατάσταση της αλυσίδας με βάση έναν αλγόριθμο αποδοχής/απόρριψης του δείγματος. Η επιλογή ως προς την κατανομή αυτή γίνεται με κριτήριο την ευκολία στον χειρισμό της, (π.χ ομοιόμορφη). Ιδιαίτερη σημασία συνεπώς δεν έχει η μορφή της προτεινόμενης συνάρτησης κατανομής πιθανότητας άλλα όπως προκύπτει από το

εύρος στο οποίο μπορεί να κινηθεί κάθε δείγμα σε αυτό τον τυχαίο περίπατο στη γειτονία του. Έχει παρατηρηθεί ότι μικρό εύρος οδηγεί σε δείγματα της αλυσίδας Markov τα οποία έχουν μεγάλη συσχέτιση ενώ μεγάλο εύρος οδηγεί σε επαναλαμβανόμενες αλυσίδες, τα οποί και τα δύο επηρεάζουν την σύγκλιση της μεθόδου.



Σχήμα 9: Τυχαίος περίπατος του δείγματος στην περιοχή αστοχίας με βάση το εύρος της προτεινόμενης συνάρτησης πιθανότητας

Με τη μέθοδο της αλυσίδας Markov επιτυγχάνεται ταχύτερη εξερεύνηση της περιοχής αστοχίας αφού η μέθοδος του Metropolis εξασφαλίζει ότι όλα τα δείγματα βρίσκονται μέσα στην περιοχή αστοχίας.

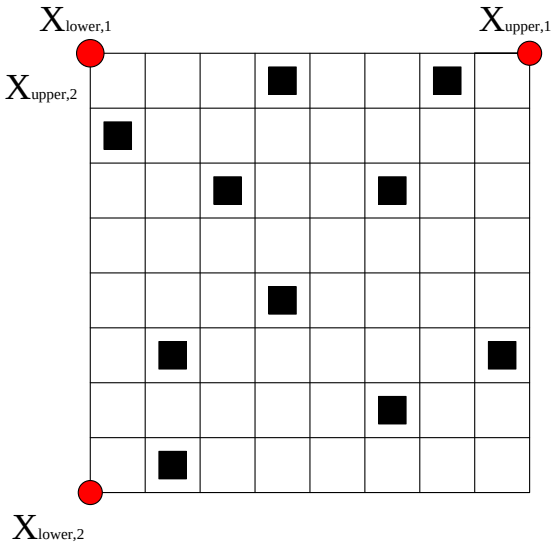
Στα πλαίσια της παρούσας διδακτορικής διατριβής μια μεθοδολογία για τον υπολογισμό της πιθανότητας αστοχίας ενός στοχαστικού συστήματος αναπτύχθηκε η οποία χρησιμοποιεί τα τεχνητά νευρωνικά δίκτυα στο πλαίσιο της μεθόδου των υποσυνόλων με αποτέλεσμα την ενίσχυση της υπολογιστικής απόδοσης της. Η βασική ιδέα της μεθόδου είναι η σταδιακή εκπαίδευση των νευρωνικών δικτύων σε

κάθε υποσύνολο ούτως ώστε να χρησιμοποιηθούν για τον εμπλουτισμό της περιοχής αστοχίας δίχως αυξανόμενο υπολογιστικό κόστος. Για την προσομοίωση των δειγμάτων που θα χρησιμοποιηθούν για την εκπαίδευση των νευρωνικών δικτύων επιλέχθηκε η τεχνική της δειγματοληψίας του λατινικού υπερκύβου, η οποία έχει το πλεονέκτημα ότι όλη η περιοχή της κατανομής που χρησιμοποιείται προσομοιώνεται. Και ειδικότερα στην περίπτωση της ομοιόμορφης κατανομής η δειγματοληψία γίνεται με τέτοιο τρόπο ούτως ώστε δεν υπάρχει το φαινόμενο της υπερ-δειγματοληψίας. Η διαδικασία είναι η εξής: Ο υπολογισμός του κριτηρίου αστοχίας που αντιστοιχεί στην πρώτη πιθανότητα είναι παρόμοιος με την κλασσική μέθοδο. Για κάθε ένα από τα δείγματα που επιλέγονται ως υποψήφια για το επόμενο υποσύνολο και βρίσκονται μέσα στην περιοχή αστοχίας εκπαιδεύεται ένα νευρωνικό δίκτυο.

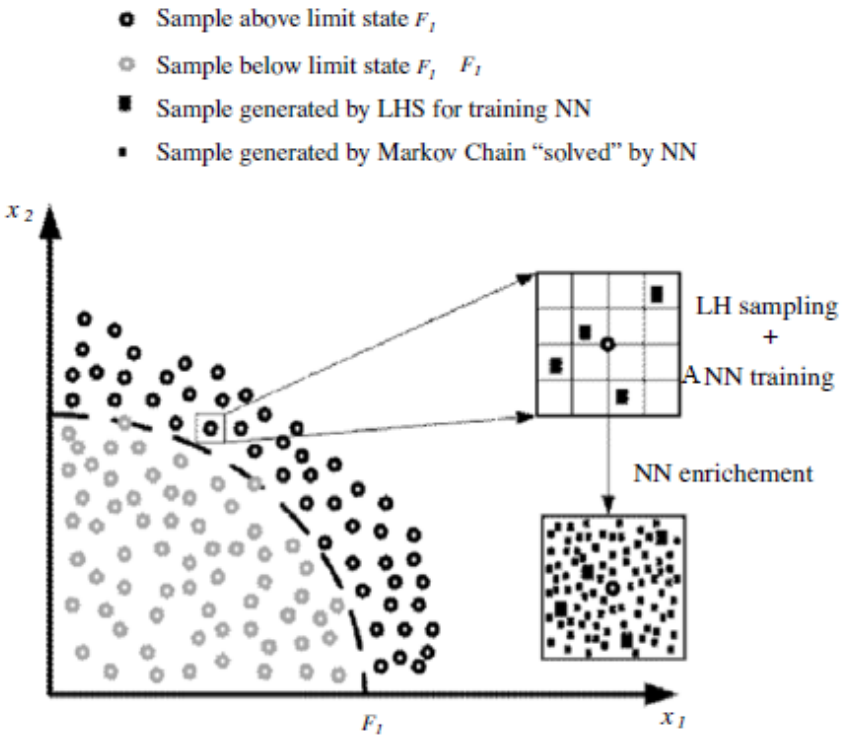
Όπως αναφέρθηκε παραπάνω για την εκπαίδευση του χρησιμοποιείται η τεχνική δειγματοληψίας του υπερκύβου για την επιλογή των κατάλληλων ζευγαριών εισόδου και εξόδου. Σημαντικό ρόλο στην επιτυχία της μεθόδου παίζει η επιλογή των διαστάσεων του υπερκύβου. Για τις ανάγκες τις μεθόδου δύο τρόποι εφαρμόστηκαν. Στην πρώτη περίπτωση το κάθε δείγμα τοποθετείται στο κέντρο του υπερκύβου ο οποίος έχει πλευρά το διπλάσιο του εύρος της προτεινόμενης συνάρτησης κατανομής που χρησιμοποιείται στην προσομοίωση των αλυσίδων Markov. Στην δεύτερη περίπτωση από κάθε δείγμα δημιουργούμε μια μεγάλη αλυσίδα Markov από όπου και βρίσκουμε το μέγιστη και ελάχιστη τιμή που αντιστοιχεί σε κάθε δείγμα. Ο υπερκύβος σε αυτή την περίπτωση έχει κέντρο το μηδέν και πλευρά που κυμαίνεται από την ελάχιστη μέχρι την μέγιστη τιμή.

$$\boldsymbol{x}^{(i)} = \begin{bmatrix} \mathrm{x}_1^{(i)} \\ \mathrm{x}_2^{(i)} \\ \cdot \\ \cdot \\ \cdot \\ \mathrm{x}_n^{(i)} \end{bmatrix} \overset{MCMC}{\rightarrow} \begin{bmatrix} \mathrm{x}_{11}^{(k)} & \mathrm{x}_{12}^{(k)} & \mathrm{x}_{13}^{(k)} & \cdot & \cdot & \cdot & \mathrm{x}_{1N_{MCMC}}^{(k)} \\ \mathrm{x}_{21}^{(k)} & \mathrm{x}_{22}^{(k)} & \mathrm{x}_{23}^{(k)} & \cdot & & & \mathrm{x}_{2N_{MCMC}}^{(k)} \\ \cdot & & & & \cdot & & \cdot \\ \cdot & & & & & \cdot & \cdot \\ \cdot & & & & & & \cdot \\ \mathrm{x}_{n1}^{(k)} & \mathrm{x}_{n2}^{(k)} & \mathrm{x}_{n3}^{(k)} & & & & \mathrm{x}_{nN_{MCMC}}^{(k)} \end{bmatrix} \tag{8}$$

$$\left.\begin{aligned} \mathrm{x}_{loweer,j}^{(i)} &= \mathrm{minimum}\left(\mathrm{x}_{\mathrm{jk}}^{(i)}\right) \\ \mathrm{x}_{upper,j}^{(i)} &= \mathrm{maximum}\left(\mathrm{x}_{\mathrm{jk}}^{(i)}\right) \end{aligned}\right|_{\substack{j=1,\ldots,n \\ k=1,\ldots,N_{MCMC}}} \tag{9}$$



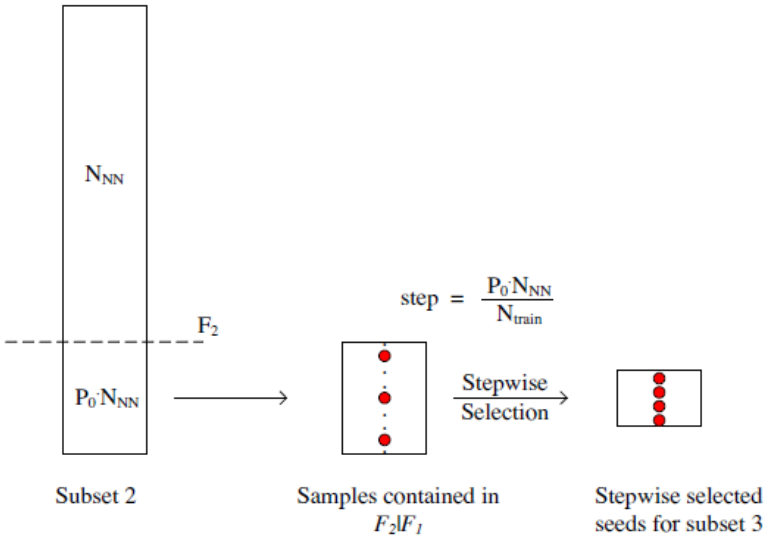
Σχήμα 10: Διαστάσεις του λατινικού υπερκύβου στην πρώτη περίπτωση εκπαίδευσης



Σχήμα 11:Προτεινόμενη μεθοδολογία για τον εμπλουτισμένο της περιοχής αστοχίας σε κάθε υποσύνολο.

Ο αριθμός των σημείων τα οποία χρησιμοποιούνται για την εκπαίδευση του κάθε νευρωνικού δικτύου είναι τέτοιος ώστε ο αριθμός των αναλύσεων που απαιτούνται να παραμένει ο ίδιος με της κλασσική μεθόδου των υποσυνόλων. Επίσης πρέπει να αναφερθεί ότι επειδή ο αριθμός των δειγμάτων που δημιουργούνται με τα νευρωνικά δίκτυα είναι πολύ μεγαλύτερος από αυτόν που κανονικά είναι η επιλογή των δειγμάτων που θα χρησιμοποιηθούν ως υποψήφια για το επόμενο υποσύνολο γίνεται βηματικά. Εφόσον το νευρωνικό δίκτυο εκπαιδευθεί κατάλληλα

μπορούμε να το χρησιμοποιήσουμε ούτως ώστε να εμπλουτίσουμε την περιοχή αστοχίας δίχως περαιτέρω υπολογιστικό κόστος και συνεπώς να βελτιώσουμε την ποιότητα των αποτελεσμάτων της μεθόδου.



Σχήμα 12: Βηματική επιλογή των υποψήφιων δειγμάτων για το επόμενο υποσύνολο

Στο επόμενο μέρος της διδακτορικής διατριβής, τα νευρωνικά δίκτυα χρησιμοποιήθηκαν ως μετά-μοντέλα, στο πλαίσιο της προσομοίωσης Monte Carlo, για την εκτίμηση των δύο πρώτων ροπών της απόκρισης ενός συστήματος καθώς επίσης για την ανάλυση της αξιοπιστίας των κατασκευών. Η προτεινόμενη μεθοδολογία που αναπτύχθηκε στηρίχθηκε στη μέθοδο της φασματικής απεικόνισης για την δημιουργία πραγματοποιήσεων του στοχαστικού πεδίου. Ένα από τα κύρια χαρακτηριστικά της μεθόδου αυτής είναι ότι είναι ανεξάρτητη από το δίκτυο των πεπερασμένων στοιχείων αφού η διαδικασία αναπαραγωγής του στοχαστικού μοντέλου είναι ανεξάρτητη από αυτό και εξαρτάται κυρίως από το φάσμα ισχύος. Η εξίσωση που δίνει τις δειγματοσυναρτήσεις είναι

$$f(x) = \sqrt{2} \cdot \sum_{n=1}^{N} A_n \cdot \cos(\omega_n + \varphi_n) \tag{10}$$

$$A_0 = 0, \qquad A_n = \sqrt{2 S_f(\omega_n) \cdot \Delta\omega}$$
$$\Delta\omega = \omega_{up} / N \tag{11}$$

Όπου S_f είναι το φάσμα ισχύος, N ο αριθμός που διακριτοποιούμε τον άξονα των συχνοτήτων, ω_{up} είναι η συχνότητα αποκοπής και φ_n ανεξάρτητες τυχαίες γωνίες φάσεις που ακολουθούν την ομοιόμορφη κατανομή και παράγονται από μια γεννήτρια τυχαίων αριθμών. Για την εύρεση της συχνότητας αποκοπής χρησιμοποιούμε τη σχέση

$$\int_0^{\omega_{up}} S_f(\omega)d\omega = (1-\varepsilon)\int_0^{\infty} S_f(\omega)d\omega \tag{12}$$

Κατά την παραγωγή των δειγματοσυναρτήσεων το βήμα που πρέπει να χωρίζει τις παραγόμενες τιμές του πρέπει να είναι $\Delta x \le \pi / \omega_{up}$ για να αποφευχθεί το φαινόμενο της επικάλυψης.

Η προτεινόμενη μεθοδολογία χρησιμοποιεί ως διάνυσμα εισόδου για την εκπαίδευση του νευρωνικού δικτύου απευθείας τις τιμές των ανεξάρτητων γωνιών φάσεις που χρησιμοποιούνται στη μέθοδο της φασματικής απεικόνισης και παράγονται χρησιμοποιώντας τη μέθοδο δειγματοληψίας του υπερκύβου, ενώ ως διάνυσμα εξόδου χρησιμοποιεί τις αντίστοιχες τιμές τις απόκρισης. Οι συμβατικές μέθοδοι στοχαστικών πεπερασμένων στοιχείων που χρησιμοποιούν τα νευρωνικά δίκτυα ως μετά-μοντέλα, χρησιμοποιούν ως διάνυσμα εισαγωγής τις πραγματοποιήσεις του στοχαστικού πεδίου. Αυτό έχει ως αποτέλεσμα οι διαστάσεις του διανύσματος να εξαρτώνται από τη πυκνότητα του δικτύου των πεπερασμένων στοιχείων, εάν χρησιμοποιείται λ.χ. η μέθοδος του μέσου σημείου. Στόχος της μεθόδου είναι η μείωση των διαστάσεων του διανύσματος εισόδου που θα οδηγήσει στην καλύτερη εκπαίδευση του νευρωνικού δικτύου.

Στα πλαίσια της εφαρμογής της μεθόδου, η κλασσική στοχαστική μέθοδος του μέσου σημείου χρησιμοποιείται με την οποία στο κέντρο κάθε πεπερασμένου στοιχείου ορίζεται μια τυχαία μεταβλητή $f(x_i)$ που παράγεται με την μέθοδο της φασματικής απεικόνισης. Κατά συνέπεια το μητρώο της κατασκευής μπορεί να θεωρηθεί ως συνάρτηση της μεταβλητής αυτής και η απόκριση να δίνεται από τη σχέση

$$\mathbf{u} = \mathbf{K}^{-1}\left(f(x_i)\right)\mathbf{F} \tag{13}$$

Η συμβατικές μέθοδοι χρησιμοποιούν το μητρώο των τιμών αυτών ως διάνυσμα εισόδου

$$\mathbf{Input} = \begin{Bmatrix} f(x_1) \\ f(x_2) \\ . \\ . \\ . f(x_{N_e}) \end{Bmatrix} \tag{14}$$

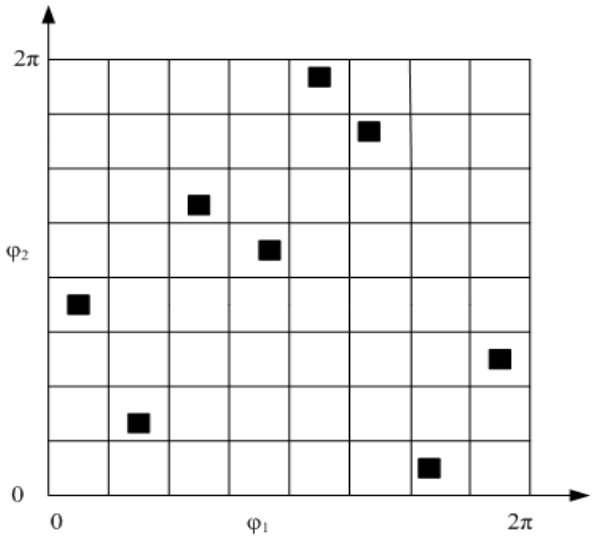
Όπου N_e ο αριθμός των πεπερασμένων στοιχείων του μοντέλου. Οι πραγματοποιήσεις αυτές εξαρτώνται από τις τυχαίες γωνίες φάσεις φ_n και κατά συνέπεια μπορούμε να γράψουμε

$$\mathbf{u} = \mathbf{K}^{-1}(\varphi_n)\mathbf{F} \tag{15}$$

Μπορούμε να πούμε συνεπώς ότι η διάσταση του διανύσματος εισόδου που χρησιμοποιείται για την εκπαίδευση του νευρωνικού δικτύου εξαρτάται από τον αριθμό των σημείων της διακριτοποίησης του πεδίου των συχνοτήτων και άρα από τις διαστάσεις του προβλήματος (μονοδιάστατο, δισδιάστατο, τρισδιάστατο). Σε κάθε μία από αυτές τις περιπτώσεις η διάσταση θα'ναι

$$\begin{aligned} \dim &= N_x && \textbf{(1-D)} \\ \dim &= 2N_xN_y && \textbf{(2-D)} \\ \dim &= 4N_xN_yN_z && \textbf{(3-D)} \end{aligned} \tag{16}$$

Ιδιαίτερη σημασία για την καλή εκπαίδευση του νευρωνικού δικτύου έχει ο τρόπος με τον οποίο πραγματοποιείται η επιλογή των γωνιών φάσης. Για το λόγο αυτό η μέθοδος δειγματοληψίας του υπερκύβου χρησιμοποιήθηκε. Το γεγονός ότι η γωνίες φάσεις ακολουθούν την ομοιόμορφη κατανομή στο [0,2π] με την ταυτόχρονη χρησιμοποίησης της μεθόδου αυτής εξασφαλίζει ότι δεν θα υπάρξει υπερκάλυψη κάποιου διαστήματος και θα έχουμε καλύτερη κάλυψη του δειγματικού χώρου των τυχαίων μεταβλητών.



Σχήμα 13: Μέθοδος δειγματοληψίας του λατινικού υπερκύβου για τις τυχαίες γωνίες φάσης στην περίπτωση της ομοιόμορφης κατανομής

Η συμπεριφορά της διαδικασία εκπαίδευσης του νευρωνικού δικτύου ελέγχεται από ένα κριτήριο σύγκλισης το δίνεται από την παρακάτω σχέση

$$\hat{e}_{NN} = \frac{|Out_{real} - Out_{NN}|}{Out_{real}} \times 100\ (\%) \qquad (17)$$

Όταν η τιμή του λάθους e_{NN} γίνει μικρότερη από μια προβλεπόμενη (π.χ 5%) τότε χρησιμοποιούμε το εκπαιδευμένο νευρωνικό δίκτυο για να υπολογίσουμε τις τιμές της απόκριση του συστήματος για διάφορες τιμές των γωνιών φάσης δίχως περαιτέρω υπολογιστικό κόστος. Τα βήματα της προτεινόμενης μεθοδολογίας παρουσιάζονται συνοπτικά παρακάτω.

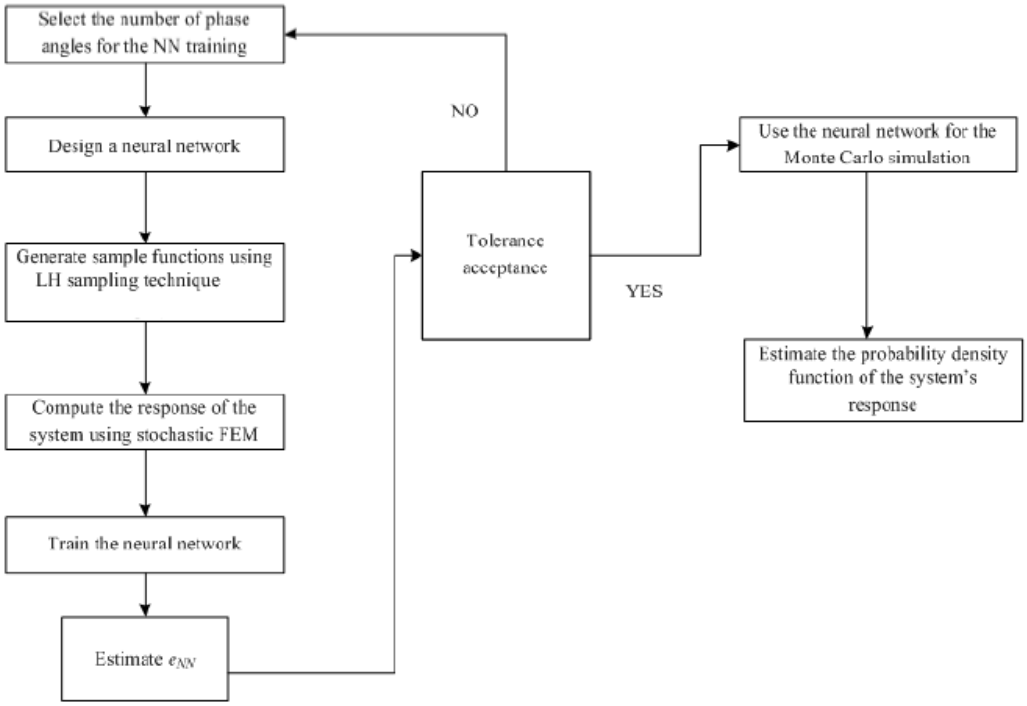
[1]. Υπολογισμός του αριθμού των γωνιών φάσης που θα χρησιμοποιηθούν για την εκπαίδευση του νευρωνικού δικτύου (από τον αριθμό της διακριτοποίηση του πεδίου συχνοτήτων)

[2]. Εφαρμογή της μεθόδου δειγματοληψίας του υπερκύβου για την δημιουργία των τυχαίων γωνιών φάσης που χρησιμοποιούνται ως διάνυσμα εισόδου στο νευρωνικό δίκτυο. Οι αντίστοιχες αποκρίσεις αποτελούν το διάνυσμα εξόδου.

[3]. Εκπαίδευση του νευρωνικού δικτύου επιλέγοντας την κατάλληλη αρχιτεκτονική και τον κατάλληλο αλγόριθμο εκπαίδευσης

[4]. Υπολογισμός του κριτηρίου σύγκλισης το οποίο καθορίζει κατά πόσο η εκπαίδευση ήταν επαρκής.

Εάν επιτεύχθηκε το κριτήριο σύγκλισης τότε χρησιμοποιούμε το εκπαιδευμένο νευρωνικό δίκτυο για τον υπολογισμό της απόκρισης του συστήματος, ειδάλλως επαναλαμβάνουμε τη διαδικασία χρησιμοποιώντας είτε αλλάζοντας την αρχιτεκτονική του δικτύου είτε χρησιμοποιώντας μεγαλύτερο αριθμό δειγμάτων για την εκπαίδευση του νευρωνικού δικτύου. Ποιο από αυτά καθορίζει την επιτυχία της μεθόδου εξαρτάται από τον τύπο του προβλήματος. Στο παρακάτω σχήμα βλέπουμε τα διάφορα βήματα της προτεινόμενης μεθοδολογίας.



Σχήμα 14: Βήματα προτεινόμενης μεθοδολογίας

Στο τελευταίο μέρος της διδακτορικής διατριβής πραγματοποιείται μια προσαρμοστική διατύπωση των φασματικών στοχαστικών πεπερασμένων στοιχείων (spectral stochastic finite element method) με μεθόδους Galerkin χρησιμοποιώντας τη μέθοδο διακύμανσης της απόκρισης (Variability response function) για τη χωρική κατανομή των όρων της Karhunen-Loeve. Η προτεινόμενη μέθοδος αποτελείται από δυο φάσεις. Στην πρώτη φάση πραγματοποιείται η εύρεση των όρων της Karhunen-Loeve που απαιτούνται σε κάθε θέση με τη χρησιμοποίηση της συνάρτησης διακύμανσης της απόκρισης. Τα βήματα που ακολουθούνται είναι τα παρακάτω:

[1]. Εύρεση της συνάρτησης διακύμανσης στον φορέα

$$\mathrm{Var}^{\mathrm{T}}(u,\mathbf{x}) = 2\int_0^{\infty} \mathrm{VRF}(\boldsymbol{\kappa},\mathbf{x},\sigma_f)\cdot S_{ff}(\boldsymbol{\kappa})d\boldsymbol{\kappa} \tag{18}$$

[2]. Για συγκεκριμένο φάσμα υπολογίζουμε τη διακύμανση της απόκρισης χρησιμοποιώντας τη μέθοδο της γρήγορης Monte Carlo. Η διακύμανση αυτή αποτελεί το σημείο αναφοράς μας για την συνέχεια.

[3]. Δημιουργούμε πραγματοποιήσεις του στοχαστικού πεδίου χρησιμοποιώντας το ανάπτυγμα Karhunen-Loeve για διαφορετικές τιμές του αριθμού *M* των όρων της σειράς

$$f^{M}(\mathbf{x}) = \sum_{n=1}^{M} \sqrt{\lambda_n}\cdot\varphi_n(\mathbf{x})\cdot\xi_n \tag{19}$$

[4]. Χρησιμοποιώντας το μετασχηματισμό Fourier δημιουργούμε το αντίστοιχο φάσμα που παράγεται από κάθε τέτοια δειγματοσυνάρτηση

$$S_{ff}{}^{M}(\boldsymbol{\kappa}) = \frac{1}{2\pi T}\left|\int_0^{T} f^{M}(\mathbf{x})\cdot e^{-i\boldsymbol{\kappa}\mathbf{x}}d\mathbf{x}\right| \tag{20}$$

[5]. Εισάγουμε το φάσμα στην ολοκληρωτική σχέση που μας δίνει τη συνάρτηση διακύμανση της απόκρισης και υπολογίζουμε την αντίστοιχη διακύμανση

$$\mathrm{Var}^{M}(u,\mathbf{x}) = 2\int_0^{\infty} \mathrm{VRF}(\boldsymbol{\kappa},\mathbf{x},\sigma_f)\cdot S_{ff}{}^{M}(\boldsymbol{\kappa})d\boldsymbol{\kappa} \tag{21}$$

[6]. Υπολογίζουμε το σφάλμα στην πρόβλεψη της διακύμανσης που οφείλεται στην τάξη του αναπτύγματος της Karhunen-Loeve

$$\hat{e}(\mathbf{x}) = \frac{\left|\mathrm{Var}^{T}(u,\mathbf{x}) - \mathrm{Var}^{M}(u,\mathbf{x})\right|}{\mathrm{Var}^{T}(u,\mathbf{x})}\times 100\% \tag{22}$$

Απαιτώντας το σφάλμα να είναι ομοιόμορφα κατανεμημένο σε όλο το φορέα βρίσκουμε τον απαιτούμενο αριθμό *M* που απαιτείται σε κάθε θέση και ο οποίος γενικά θα διαφέρει από θέση σε θέση. Η επόμενη φάση περιλαμβάνει τα εξής στάδια:

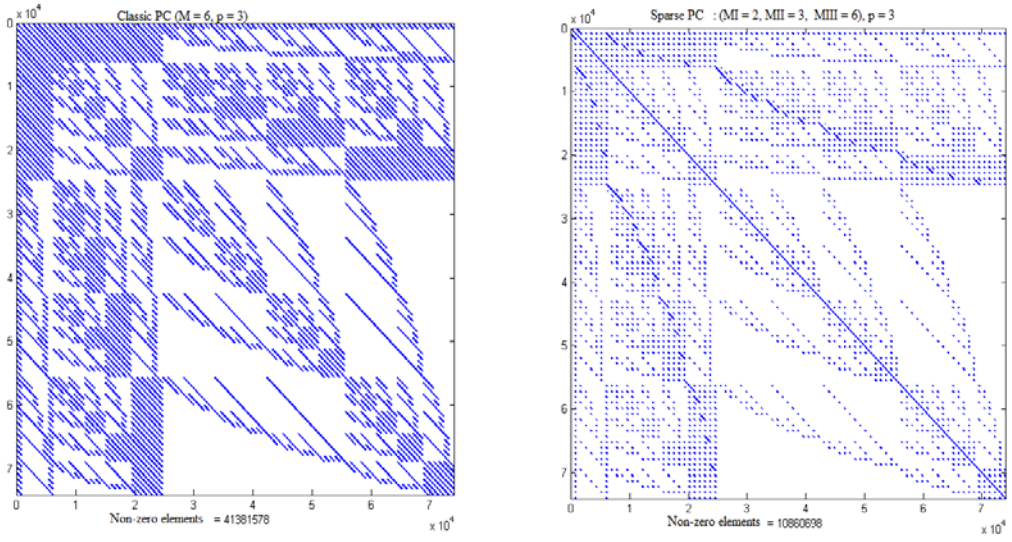
[1]. Εύρεση του αριθμού διακριτοποίησης της τυχαίας διάστασης σε κάθε θέση του φορέα

$$P_i = \begin{pmatrix} M_i + p \\ p \end{pmatrix} \tag{23}$$

[2]. Κάθε βαθμός ελευθερίας του συστήματος αναπτύσσεται πάνω στην αντίστοιχη βάση που προκύπτει από την παραπάνω διαδικασία

$$\begin{aligned} U_1 &= \sum_{i=0}^{P_1-1} u_i \Psi_i \\ U_2 &= \sum_{j=0}^{P_2-1} u_j \Psi_j \\ &. \\ &. \\ U_N &= \sum_{k=0}^{P_N-1} u_k \Psi_k \end{aligned} \tag{24}$$

Το αποτέλεσμα της μεθόδου αυτής έχει μεγάλη σημασία γιατί οδηγεί σε μείωση της πυκνότητας του διευρυμένου μητρώου του συστήματος και συνεπώς μείωση του υπολογιστικού κόστους που απαιτείται για την επίλυση του διευρυμένου συστήματος εξισώσεων. Τα αποτελέσματα είναι ιδιαίτερα εμφανή σε προβλήματα μοντέλων με μεγάλο αριθμό βαθμών ελευθερίας καθώς επίσης και όταν χρησιμοποιούνται επαναληπτικοί αλγόριθμοι για την επίλυση του.



<u>Σχήμα 15</u>: Διευρυμένο στοχαστικό μητρώο τη κλασσικής μεθόδου των φασματικών στοχαστικών πεπερασμένων στοιχείων και της προσαρμοστικής μεθόδου στην περίπτωση λογαριθμικού πεδίου

Contents

List of Figures **ix**

List of Tables **xiii**

1 Introduction **1**
1.1 Motivation and scope 1
1.2 Outline 5

2 Probability and uncertainty **9**
2.1 Probability theory 9
2.2 Random Variables 11
2.2.1 Properties 11
2.2.2 Some common random variables 13
2.3 Multiple random variables 15
2.3.1 Random vectors 15
2.3.2 The multivariate Gaussian distribution 15
2.3.3 Central Limit theorem 16
2.4 Random processes and Random fields 16
2.4.1 Homogeneous random fields 18
2.4.2 Ergodicity 20
2.4.3 Gaussian random fields 21
2.4.4 Non-Gaussian random fields 21
2.5 Discretization of random fields 22
2.5.1 Spectral representation method 22
2.5.2 The Karhunen-Loève expansion 25

3 The Finite Element method **29**
3.1 Formulation for plane stress and plane strain 32
3.1.1 Two-dimensional triangular element 36
3.1.2 Two-dimensional isoparametric elements 37
3.2 Three-dimensional isoparametric elements 39

4 Basic principles of artificial neural networks **43**
4.1 Artificial neuron . . . 45
4.2 Architecture of ANN . . . 47
4.2.1 Feedforward ANN . . . 48
4.2.2 Recurrent ANN . . . 49
4.3 The learning algorithm . . . 50
4.3.1 Supervised learning . . . 50
4.3.1.1 Global adaptive techniques . . . 51
4.3.1.2 Local adaptive techniques . . . 51
4.3.2 Unsupervised learning . . . 52
4.3.3 Reinforcement learning . . . 53

5 Structural Reliability analysis **55**
5.1 An overview on reliability analysis . . . 55
5.2 Crude Monte Carlo simulation . . . 59
5.3 Latin Hypercube sampling (LHS) . . . 61
5.4 The Subset Simulation (SS) . . . 61
5.4.1 Subset simulation algorithm . . . 62
5.5 Markov chain Monte Carlo simulation . . . 63
5.5.1 Modified Metropolis algorithm . . . 63
5.5.2 Conditional probability estimators . . . 65

6 Accelerated subset simulation with neural networks for reliability analysis **67**
6.1 ANN architecture . . . 68
6.2 First ANN training approach (SS-ANN1) . . . 69
6.2.1 LHS based training . . . 71
6.3 Second ANN training approach (SS-ANN2) . . . 72
6.3.1 Choice of witdh for the ANN training in SS-ANN2 . . . 74
6.4 Numerical examples . . . 75
6.4.1 Mathematical model 1 . . . 75
6.4.1.1 Subset Simulation . . . 75
6.4.1.2 Proposed SS-ANN methodologies . . . 76
6.4.2 Mathematical model 2 . . . 77
6.4.3 6-storey reinforced concrete building . . . 79
6.4.4 SS-ANN versus ANN-MCS . . . 81
6.5 Conclusions . . . 82

7 Stochastic finite element method (SFEM) **85**
7.1 Introductions . . . 85
7.2 Spectral representation-based MC . . . 87
7.3 Karhunen-Loève-based MC . . . 88

7.4 Closed form solution using variability Response functions (VRF) . . . 88
7.4.1 Fast Monte Carlo simulation . . . 89
7.5 Spectral Stochastic Finite element method (SSFEM) . . . 90
7.5.1 SSFEM of Gaussian random fields . . . 90
7.5.2 SSFEM of log-normal random fields . . . 94
7.6 Parametric study of SSFEM vs MC in large-scale systems . . . 96
7.6.1 Computation of the second moments of the response field . . . 98
7.6.2 Reliability analysis . . . 104
7.7 Conclusions . . . 105

8 Neural network assisted stochastic structural mechanics 107
8.0.1 Selection of training points . . . 109
8.0.1.1 Latin Hypercube sampling . . . 109
8.0.2 Steps of proposed methodology . . . 110
8.1 Numerical Examples . . . 112
8.1.0.1 2D cantilever with a 2D Gaussian random field . . . 112
8.1.1 2D perforated plate with 1D non-Gaussian random field . . . 118
8.1.2 3D cantilever with 3D Gaussian random field . . . 121
8.2 Conclusions . . . 126

9 An adaptive Galerkin SSFEM approach 127
9.1 Phase 1: Error estimation . . . 127
9.2 Phase 2: Building a sparse PC coefficient matrix . . . 129
9.3 Numerical examples . . . 129
9.3.1 3D Cantilever . . . 130
9.3.2 2D plane stress plate . . . 135
9.4 Solution of the system of stochastic partial differential equations . . . 143
9.4.1 Monte Carlo methods . . . 143
9.4.2 Solution of the augmented SSFEM systems of equations . . . 144
9.5 Computational merits of the proposed methodology . . . 145

10 Summary - Innovation of thesis 147

Bibliography 151

List of Figures

2.1 Example of probability computation 12
2.2 pdf for the Uniform distribution for various a, b 13
2.3 pdf for the Normal distribution with $\mu = 0$ and for various values of σ^2 14
2.4 pdf for the Beta distribution for various values of the parameters a, b . 14
2.5 Realizations of a random process 17
2.6 Power spectral density function (psd) 20

3.1 A mesh of finite elements over a rectangular region having a central hole. 30
3.2 A solid of domain Ω. 32
3.3 Support conditions. 32
3.4 Triangular finite element. 36
3.5 Linear and quadratic finite elements and their representation in the local coordinate system. 37
3.6 Linear and quadratic three-dimensional finite elements and their representation in the local coordinate system. 39

4.1 Working principle of an artificial neuron (source : Artificial Neural networks, wikibooks.org, 2013. 44
4.2 Example of simple artificial neural network. 44
4.3 Biological and artificial neuron design. 45
4.4 Step function. 46
4.5 Sigmoid function. 47
4.6 A taxonomy of feed-forward and recurrent/feedback network architectures (source : Artificial neural networks- a tutorial, 1996) 48
4.7 Feed-forward (Feed-forward artificial neural network. 49

5.1 Joint density function $f_{R,S}(r, s)$, marginal density functions $f_R(r)$ and $f_S(s)$ and failure domain D, (Melchers, 1999) 56
5.2 Basic $R - S$ problem: $f_R(r)$ $f_S(s)$ representation 57
5.3 Basic $R - S$ problem: $f_R(r)$ $f_S(s)$ representation 57
5.4 Distribution of safety margin $Z = R - S$ 58

5.5 Illustration of the two-dimensional case of a linear limit state function and standardized normally distributed variables U 59
5.6 Schematic representation of generation of samples using Markov Chain Monte Carlo algorithm 64

6.1 Schematic representation of progressive ANN training windows in two subsequent SS levels. 71
6.2 LHS training scheme for two random variables and three subsequent SS levels. 72
6.3 Schematic representation of steps 1 and 2 for the second NN training approach. 73
6.4 Schematic representation of step 4. 74
6.5 Views of the 6-storey RC building. 80

7.1 Sparsity pattern of $\boldsymbol{K}$ for the Gaussian case ($M = 2,\ p = 2$) 93
7.2 Sparsity pattern of $\boldsymbol{K}$ for the Gaussian case ($M = 6,\ p = 4$) 93
7.3 Sparsity pattern of $\boldsymbol{K}$ for the non-Gaussian case ($M = 2,\ p = 2$) . . . 95
7.4 Sparsity pattern of $\boldsymbol{K}$ for the non-Gaussian case ($M = 6,\ p = 4$) . . . 96
7.5 Element mesh of a quarter of the deterministic soil problem with 10k dof 97
7.6 Step 1: Response's c.o.v (%) convergence error of MC for the Gaussian field with c.o.v = 15% 99
7.7 Step 1: Response's c.o.v (%) convergence error of MC for the log-normal field with c.o.v 30% 99
7.8 Step 1: Response's c.o.v (%) convergence error of MC for the log-normal field with c.o.v = 80% 100
7.9 Step 3: Response's c.o.v (%) convergence error of the SSFEM for the Gaussian field with c.o.v = 15% and $p = 2, 4$. 101
7.10 Step 3: Response's c.o.v (%) convergence error of the SSFEM for the log-normal field with c.o.v = 30% and $p = 2, 3, 4$. 101
7.11 Step 3 : Response's c.o.v (%) convergence error of the SSFEM for the log-normal field with c.o.v = 80% and $p = 2, 3, 4$. 102
7.12 Settlement's c.o.v for random field with c.o.v = 15% , correlation length 2m, and $M = 12$ for both MC and SSFEM. 103
7.13 Settlement's c.o.v for random field with c.o.v = 30% , correlation length 2m, and $M = 10$ for both MC and SSFEM 103
7.14 Settlement's c.o.v for random field with c.o.v = 80% , correlation length 2m, and $M = 4$ for MC and SSFEM 104

8.1 Pairing two random variables $\varphi_1,\ \varphi_2$ using LH sampling 110
8.2 Schematic representation of the proposed methodology. 112
8.3 Example 1: 2D cantilever beam. 113

8.4 Example 1: Spectral density function. 113
8.5 Histogram of the displacement of MC simulation 114
8.6 Histogram of the displacement for the proposed method with 1000 training points for the ANN 115
8.7 Histogram of the displacement for the proposed method with 4000 training points for the ANN 115
8.8 Probability $P(U_{tip} > d_{cr}) = 0.0384$ estimated with 50000 MC simulations 116
8.9 Probability $P(U_{tip} > d_{cr}) = 0.0324$ estimated with 50000 ANN samples 117
8.10 Probability $P(U_{tip} > d_{cr}) = 0.0382$ estimated with 50000 ANN samples 117
8.11 Example 1: ANN Validation error with respect to the LHS training points. 118
8.12 Example 2: 2D plane stress plate 119
8.13 Example 2: Probability density functions for the "U-shaped" beta random field for 150, 400 and 1000 training points. 121
8.14 Example 3: Short cantilever. 122
8.15 Example 3:Histogram of the displacement obtained with brute force MC simulation 123
8.16 Example 3:Histogram of the displacement obtained with the proposed method 124
8.17 Example 3:Probability $P(U_{tip} > d_{cr})$ for the MC 125
8.18 Example 3: Probability $P(U_{tip} > d_{cr})$ for the proposed method 125

9.1 Finite element mesh (HEXA) 131
9.2 Estimation of **VRF** in three locations along the beam 131
9.3 **VRF** at points x_j, j=1,2,3 132
9.4 SDFs from KL for different M and target SDF 133
9.5 Joint plot of S^5_{ff} with **VRF**($x_1 = 3\ m$), **VRF**($x_2 = 7\ m$), **VRF**($x_3 = 10\ m$) 133
9.6 Standard deviation error (%) as a function of M 134
9.7 Non-zero elements of the : (a) classical PC and (b) sparse PC 135
9.8 Spectral density function of the underlying Gaussian random field . . 136
9.9 Points in which **VRF** is estimated 137
9.10 Variability response functions at three representative points 137
9.11 Standard deviation error (%) as a function of M in three representative locations x_1, x_2 and x_3 138
9.12 Subdomains with equal M values required to reach a target error of about 2% 139
9.13 Relative error (%) of the mean displacement as a function of p 140
9.14 Relative error (%) of the standard deviation of U_x as a function of p . 140
9.15 Sparsity of the augmented matrix 141
9.16 Increase of sparsity as a function of p 142
9.17 Non-zero elements of the : (a) classical PC and (b) sparse PC 142

LIST OF FIGURES

List of Tables

6.1 Estimation of failure probability P_F with SS 76
6.2 SS-ANN estimation of failure probability for the case of 3 random variables. 76
6.3 SS-ANN estimation of failure probability for the case of 10 random variables. 76
6.4 SS-ANN estimation of failure probability for the case of 100 random variables. 77
6.5 SS estimation for 3, 10 and 100 random variables for the 2^{nd} mathematical function 78
6.6 SS-ANN estimation of failure probability for the case of 3 random variables. 78
6.7 SS-ANN estimation of failure probability for the case of 10 random variables. 78
6.8 SS-ANN estimation of failure probability for the case of 100 random variables. 78
6.9 Random variable parameters of the Gaussian distribution, for the 6-storey RC building. 80
6.10 SS-ANN estimation of the failure probability P_F for the 6-storey RC building. 81
6.11 SS-ANN vs ANN-MC estimation of failure probability for the case of 3 random variables. 82
6.12 SS-ANN vs ANN-MC estimation of failure probability for the case of 100 random variables. 82

7.1 Required number of MC simulations for achieving a response's c.o.v error less than 0.5%. 100
7.2 Step 2: Response's c.o.v (%) convergence errors for the various KL expansion orders 100
7.3 Convergence errors for the SSFEM 102
7.4 Settlements (m) with 0.1% probability of failure 105

7.5 Probability of failure as computed by SSFEM for the settlements of Table 7.4 for the limit states 105
7.6 Probability of failure as computed by SSFEM for the settlements of Table 7.4 and the necessary PC order 105

8.1 Example 1: Monte Carlo simulation 114
8.2 Example 1: ANN assisted SFEM 116
8.3 Example 1: CPU time for MC and ANN-SFEM 118
8.4 Example 2: Monte Carlo simulation 120
8.5 Example 2: ANN assisted SFEM 120
8.6 Example 2: CPU time for MC and ANN-SFEM 121
8.7 Example 3: Monte Carlo simulation 123
8.8 Example 3: ANN assisted SFEM 124
8.9 Example 3: CPU time for MC and ANN-SFEM 126

9.1 Test cases - input random variable 130
9.2 Test cases - Correlation structures 130
9.3 Standard PC expansion 134
9.4 Adaptive PC expansion 135
9.5 Standard PC expansion 139
9.6 Adaptive PC expansion 140
9.7 The PCG algorithm 143
9.8 Computational performance of classical SSFEM (Full problem) vs adaptive SSFEM (Sparse problem) 145

1

Introduction

1.1 Motivation and scope

The impact of uncertainties in the design process of engineering structures is an important field with growing interest. While most of modern design codes rely on partial safety factors, calibrated to target structural reliability, stochastic modeling of the uncertainties in the loading, geometrical, and material properties becomes more and more attractive leading to more rational estimations of structural safety and reliability. Several probabilistic structural analysis methods have been proposed in the past, the simplest being the description of the uncertainties by a set of correlated random variables, where each variable represents a material parameter, load factor, or geometrical property. In several engineering applications however, the description of uncertain parameters using independent random variables can be insufficient. This is due to the fact that certain physical quantities are often expected to vary randomly in space or time. The probabilistic description of such quantities requires the consideration of random fields. This approach is generally referred as stochastic structural analysis.

The analysis of structural systems with uncertain properties modeled as random fields has been the subject of extensive research in the past two decades. Over these years, the majority of research work has focused on developing various stochastic finite element methodologies for the numerical solution of the stochastic partial differential equations involved in such problems. Among these, the most commonly used are the non intrusive Monte Carlo (MC) methods (17, 121, 127)and the intrusive Galerkin-based spectral stochastic finite element methods (SSFEM) (54, 55, 142, 157). Non intrusive methods sample the stochastic PDEs at a set of points in the random space leading to a corresponding set of independent deterministic problems. On the other hand, in Galerkin-based methods the system response is spanned by a set of polynomials of the basic random variables, namely the polynomial chaos expansion (PC)(54, 55, 142, 157). The resulting stochastic finite element approximation space

becomes a tensor product space defined on the cartesian product of the physical and random domains. The polynomial chaos coefficients are consequently calculated from the solution of a system of linear equations representing the set of coupled deterministic PDEs defined in the above mentioned tensor product space.

The most suitable, versatile and easily implemented method to solve the aforementioned stochastic problems is the Monte Carlo approach. Despite its generality, MC has been used mostly as a means of verifying the accuracy of approximate and less costly procedures due to its usually high computational cost even if structural analysis is accelerated by advanced solution techniques and/or Neumann series expansion methods (26, 116, 121). In particular, when dealing with deterministic external loading, MC features the solution of successive linear systems with multiple left-hand sides, since the coefficient matrix $\mathbf{K}$ changes in every simulation. This means that each simulation i ($i = 1, ..., n_{sim}$), the solution of a problem of the form:

$$\mathbf{K}_i \mathbf{u}_i = \mathbf{f} \quad i \, (i = 1, ..., n_{sim}) \tag{1.1}$$

where $\mathbf{K}_i$ is the stiffness matrix corresponding to the stochastic realization of the i^{th} simulation, $\mathbf{u}_i$ is the corresponding vector of unknown nodal displacements, n_{sim} is the number of Monte Carlo simulations and $\mathbf{f}$ is the vector of nodal loads. The size of the stiffness matrix and the corresponding vectors is equal to the size of the equivalent deterministic problem. Thus, if $\mathbf{K}_0$ is the stiffness matrix of the deterministic problem with dimensions $N \times N$, eq.(1.1) can be written as

$$(\mathbf{K}_0 + \Delta\mathbf{K}_i)\mathbf{u}_i = \mathbf{f}\,, \quad i = 1, ..., n_{sim} \tag{1.2}$$

which specifies a set of near-by problems. The solution of the emerging systems of equations (1.1) or (9.8) can be solved using standard numerical methods, with he standard direct method based on Cholesky factorization remaining the most popular solution scheme for FE equations. However, this solution procedure exhibits two major drawbacks; poor performance when the stiffness matrices involved have large bandwidths, mainly in 3D problems and incapability of exploiting the near-by nature of the problem. Due to the fact that the solution process has to start from the beginning every time a new $\mathbf{K}_i$ is formed, the solution of the n_{sim} systems of equation (1.1) becomes a major computational task that hinders the overall simulation process for large scale problems. In order to circumvent this difficulty of direct schemes to deal with the peculiarity of the resulting systems of equations, iterative solvers have been adopted which are customized to the particular properties of the equilibrium equations arising in the context of the FE formulation used. Such iterative solvers have been presented for sequential (53, 121, 124) as well as for parallel environments (27, 105, 124).

Also, in the direction of overcoming the large computational effort involved in MC, advanced variance reduction-based simulation methods have been proposed in the context of reliability analysis, in order to reduce the number of MC required for

an accurate prediction of the probability of failure, such as adaptive sampling, importance sampling, line sampling and subset simulation (12, 13, 14, 22, 24, 101, 130). In addition to the aforementioned methodologies, meta-models, such as artificial neural networks (ANN) have been successfully implemented in the framework of reliability analysis leading to cost-efficient yet acceptable predictions of the probability of failure (13, 23, 36, 74, 75, 76, 77, 87, 108, 110, 120, 122). This is due to the associative memory properties featured by these artificial intelligence devices that allow them to become efficient surrogates to the numerical solver of the mechanical model which is repeatedly invoked in the MC. Following these early approaches, ANN approximations of the limit-state function were proposed in (86, 103) combined either with MC or with first and second order reliability methods (FORM, SORM) for handling the uncertainties. Similar approaches were presented in (23, 29, 89) where ANN based response surface methods were implemented in order to estimate the reliability integral over the failure domain. Comparative studies of ANN-based reliability analysis with corresponding polynomial approximations of the response surface as well as FORM and SORM methods are presented in (23, 60, 74, 146). In these papers, the ability of ANN to predict efficiently and accurately enough the reliability of large and complex structural systems is demonstrated. Furthermore is is shown that the advantage of using the ANN approaches is that they adapt efficiently the input/output (I/O) relations allowing for more accurate mappings of the basic random variables than in the corresponding response surface polynomial approximations of the failure functions, especially in complex failure domains (74).

As mentioned before, more advanced techniques which have become quite popular recently, are the intrusive stochastic collocation and Galerkin methods. Both methods are using tensor product spaces for the spatial and stochastic discretizations. The stochastic Galerkin method is more widely used and relies on the spectral finite element (SFEM) approach to transform a stochastic PDE into a coupled set of deterministic PDEs. In particular, solving stochastic problems using the spectral stochastic finite element approach (SSFEM) is a method which involves the solution of an augmented problem of the form (1.1)(6, 28, 31, 79, 99, 154, 155). It could be argued that one of the most blurry areas in SSFEM is the computational effort involved because the dimensionality of the corresponding deterministic matrix that has to be inverted is increased by P times, P being the dimensionality of the random space, leading in many cases in a dramatic increase on the required computational effort, especially in cases where the corresponding random input is described by a non-Gaussian marginal distribution and the truncation order of the KL as well as the PC order are high. Current literature has provided solution procedures for solving this augmented system but mainly address small to medium problems. For larger scale problems however, such a solution can become quite challenging due to the enormous memory and computational resources required. Contemporary efficient solution techniques are based on iterative solvers like the block Gauss-Jacobi (6, 32, 54, 90), CG (32, 45, 49, 53, 57, 81, 99, 112, 124)

and the block PCG (57). The latter proves to be quite efficient since it exploits the block-sparse structure of the augmented stiffness matrix while utilizing a domain decomposition solver, optimized for multiple right-hand sides. In these cases, the sparsity of the corresponding deterministic matrix that has to be inverted is increased, leading in a dramatic multifaceted of the required computational effort. Although adaptivity in the physical domain is a well established topic with a family of **p** and/or **h** adaptivity methodologies proposed for the enhancement of the FEM solution of the physical problem (58, 118, 119, 159, 160), a limited number of works exist for stochastic adaptive methodologies in random space. Stochastic adaptivity and model reduction techniques have been proposed in (1, 18, 19) in an effort to reduce the overall problem dimensionality and augment the sparsity of the deterministic matrix, in tandem with efficient preconditioning and domain decomposition-based approaches which reduce the computational cost encountered in the inversion of coupled deterministic problems.

Another parameter that plays a role of paramount importance as far as the performance of the aforementioned stochastic methodologies is concerned, is the knowledge of the correlation structure and the marginal probability distribution function (pdf) of the stochastic fields underpinning the vague system properties, which functions as a prerequisite for the prediction of the response variability of such a system. Since there is more than often lack of experimental data with regard to the quantification of such probabilistic quantities, a sensitivity analysis regarding multiple stochastic parameters is often embedded. In this case, the most commonly emergent issues are the disproportional augmentation of the computational effort and the ellipsis of insight in relation to the ways these elements affect the response variability of the actual system. This is the rationale that underpins the concept of the variability response function (VRF) introduced by Shinozuka in the late 1980s (138). VRF provides information of the spatial distribution of the variability of a structure's response with no information of the spatial correlation structure of the uncertain system parameters. The VRF is a deterministic function dependent on the structure, its boundary conditions and loading, assumed to be independent of the distributional and spectral characteristics of the uncertain system parameters. It identifies the sensitivity of the correlation structure of the uncertain parameters on the variability of the response. Although VRFs were initially proposed for relatively simple statically determinant structures, subsequent developments allowed for their successful implementation to general stochastic finite element systems. Different aspects and applications of the VRF were introduced in (10, 38, 107, 113, 114, 151), while an efficient fast Monte Carlo simulation for the numerical computation of VRF of this approach was provided in (113, 117). A development of this approach, which further boosted the validity of the assumption of independence of VRF to the stochastic parameters of the problem, was proposed in (94) where the concept of GVRF was introduced which is derived from a family of different VRFs for corresponding combinations of different pdfs with different sets of power spectral density functions.

1.2 Outline

This thesis is organized in ten chapters, which are outlined as follows:

Chapter 2 first provides an introduction to the background of the theory of probability and random variables and reviews basic distribution models. Then, the representation of quantities that vary in time (processes) or space (fields) is addressed by introducing the mathematical theory of stochastic processes (or random fields). The cases of Gaussian and non-Gaussian random processes are described in detail.

Chapter 3 is an overview to the finite element method (FEM). Today the FEM is considered as one of the well established and convenient technique for the computer solution of complex problems in different fields of engineering. The success of FEM is based largely on the basic finite element procedures used: the formulation of the problem in variational form, the finite element dicretization of this formulation and the effective solution of the resulting finite element equations. These basic steps are the same whichever problem is considered and together with the use of the digital computer present a quite natural approach to engineering analysis. In this chapter its basic concept is described and the formulas for some common finite elements are given.

Chapter 4 introduces neural networks and its use in the framework of structural engineering and moreover reliability analysis with finite elements. First, a review of its basic principles and the engineering approach is provided, followed by a discreption of the architectures of neural network. Subsequently, the three major learning technique categories are described, namely the supervised, the unsupervised and the reinforcement learning algorithms. For the first category two techniques are presented, the Global adaptive and the local adaptive.

Chapter 5 of the thesis presents structural reliability analysis and its basic principles. In this task, the simulation methods used in the context of reliability analysis. First, the fundamental concepts of reliability analysis is discussed. Then, an extended review of simulation methods is performed.

In chapter 6 a new methodology for computing the probability of structural failure by incorporating artificial neural network (ANN) in the framework of the subset simulation (SS) method is introduced. The basic concept and two alternative training strategies implemented for the ANN training procedure are described analytically in this chapter. The first introduces a closed multidimensional set Ω^M, subset of the total random variable space $\boldsymbol{\Theta}$($\Theta^M \subset \boldsymbol{\Theta}$) defined by the upper and lower bounds of the random variables as these are generated by the increased, with respect to standard SS, space of MCMCS samples at each SS level. Subsequently, the ANN is effectively trained over progressive and relatively small sets Ω^M instead of $\boldsymbol{\Theta}$, with the minimum required training points using a Latin Hypercube Sampling (LHS) training scheme. The second strategy introduces a width l around each initial seed that the MCMCS

uses as center in order to generate a random walk according to the proposal pdf. This width defines a closed subset Ω^j around each seed j, defined by a lower and upper bound at distance l from the center. The ANN is effectively trained at each Ω^j with a minimum computational cost using the LHS. Finally, numerical examples are presented in order to demonstrate the efficiency of the proposed methodology.

In chapter 7 of the thesis, a description of the stochastic finite element method (SFEM) is provided. Monte Carlo simulation and its various forms, depending on the method used for the representation of the random field (Spectral representation and Karhunen-Loève expansion), is presented. Next, the variability response functions are introduced and a discussion on their implementation is made based on results reported in the literature. Subsequently, a the description of the general formulation of the spectral stochastic finite element method (SSFEM) is made for both Gaussian and non-Gaussian random fields. Finally, a comparison of the Monte Carlo method and SSFEM in large-scale structural systems is made. The comparison is based on the computation of the second order moments of the response field as well as on a reliability analysis. In order to set an objective basis for assessing the performance of these methods, a parametric study is conducted, regarding different values for standard deviation σ_E and correlation length b. A numerical test is presented demonstrating the efficiency of the proposed implementations on a problem with different stochastic characteristics. Especially, for the computation of the second moments of the response field, the following procedure is followed: A series of Monte Carlo analyses is carried out with a fixed value of M, in order to estimate the necessary number of simulations for a convergence error of less than 1% for each value of σ_E and b examined. Assuming that the convergence behavior remains invariant for increasing M, another series of Monte Carlo analyses is carried out, in order to estimate the appropriate order of the KL expansion for a prescribed convergence error. The same procedure is carried out next performing SSFEM analyses, in order to estimate the appropriate order of the PC expansion required for convergence to the corresponding MC results. Finally, a reliability analysis is performed, in which the order of the PC expansion is being modified in order to reach a prescribed convergence error in the estimation of the probability of failure, compared to the corresponding MC results.

Chapter 8 provides a novel idea for computing the response variability of stochastic structural systems, whose material properties are described as random fields, by incorporating ANN in the framework of MC simulation. First, the basic idea of the proposed methodology is presented. Then, the basic steps required for the implementation of the this method are discussed in detail. The basic idea of the proposed approximation is to achieve a dimensionality reduction of the aforementioned training space, by using as input vector directly the random phase angles φ of the spectral representation method. Since the number of the random phase angles depend on the discretization of the power spectrum in the wave number domain the resulting input vector will be independent of the finite element discretization and the number of random variables used for the

stochastic description of the uncertain parameter(s). A further improvement in the efficiency of the proposed method may be achieved by exploiting the uniform distribution of the random phase angles in the range $[0, 2\pi]$ performing sampling of this uniform training space with a Latin hypercube (LHS) technique. Three numerical examples are presented next in order to demonstrate the efficiency of the proposed methodology in the case of Gaussian and non-Gaussian random fields.

Chapter 9 introduces an adaptive sparse polynomial chaos (PC) expansion for the response of stochastic systems whose input parameters are modeled with random fields. The proposed methodology uses Karhunen-Loève (KL) expansion for describing the random field and utilizes the concept of VRF in order to compute an a priori low cost estimation of the spatial distribution of the second-order error of the response. This way a significantly reduced number of PC coefficients, with respect to classical PC expansion, is required in order to reach a target second-order error. This variation of the KL terms results in an increase of sparsity of the coefficient matrix of the corresponding augmented linear system of equations leading to an enhancement of the computational efficiency of the SSFEM.

Finally, Chapter 10 presents a summary of the contributions of this work.

2

Probability and uncertainty

2.1 Probability theory

The theory of probability (123) deals with averages of mass phenomena occurring sequentially or simultaneously. Its purpose is to describe and predict such averages in terms of probabilities of events. In probability theory the following set terminology is used: The set (a collection of objects called elements) of all possible outcomes of an experiment is called the sample space (certain event) Ω and its subsets (sets whose elements are also elements of Ω) are called events. The empty or null set $\varnothing$ is by definition the set that contains no elements and is the complement of Ω. An experiment is specified in terms of the following concepts:

1. The set Ω of all possible outcomes.
2. The Borel field of all events of Ω.
3. The probabilities of these events.

For two events A and B the following axiomatic definitions are given

- Union: $A \cup B = \{\omega | \omega \in A \ \ or \ \ \omega \in B\}$
- Intersection: $A \cap B = \{\omega | \omega \in A \ \ and \ \ \omega \in B\}$
- Complement: $A^c = \{\omega | \omega \notin A\}$
- $(A \cup B)^c = A^c \cap B^c$

The axiomatic approach to probability is based on the following three postulates (85): The probability of an event A is a positive number $P(A)$ assigned to this event.

$$P(A) \geq 0 \tag{2.1}$$

The probability of the certain event Ω equals 1:

$$P(\Omega) = 1 \tag{2.2}$$

The probability of the null set $\varnothing$ equals 0:

$$P(\varnothing) = 0 \tag{2.3}$$

If the events $A_1, A_2, ...$ are mutually exclusive (there is no element common to both events: $A_1 \cap A_2 \cap ... = \varnothing$), then

$$P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i) \tag{2.4}$$

Let A and B events in Ω, and suppose that $P(B) > 0$. The probability of an event A given the occurrence of some other event B is formally done by considering the conditional probability of A given that B occurs. This event is denoted by $A \mid B$. The conditional probability of A given B is by definition the ratio:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \tag{2.5}$$

The following properties follow readily from the definition:

1. If $B \subset A$ then $p(A \mid B) = 1$
2. If $A \subset B$ then $p(A \mid B) = \frac{P(A)}{P(B)} \geq P(A)$

Bayes' theorem relates the conditional probability $P(A \mid B)$ and the probabilities $P(A), P(B)$ of events A and B, provided that the probability of B does not equal zero:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{2.6}$$

In Bayes' theorem, each probability has a conventional name: $P(A)$ is the prior probability (or "unconditional" or "marginal" probability) of A in the sense that it does not take into account any information about B, $P(A \mid B)$ is the conditional (posteriori) probability of A, given B, because it is derived from or depends upon the specified value of B, $P(B \mid A)$ is the conditional probability of B, given A and $P(B)$ is the prior or marginal probability of B.

2.2 Random Variables

A random variable X is the assignment of a number $x(\theta)$ to each sample point θ in sample space, $X : \Omega \rightarrow \mathbb{R}$. Thus, a random variable can be considered a function whose domain is a set and whose range are a subset of the real line. A random variable can be discrete or continuous. In order to specify the probability measures used when dealing with random variables, it is often convenient to specify alternative functions from which the probability measure governing an experiment immediately follows. We define the cumulative distribution function (cdf) $F_X : \mathbb{R} \rightarrow [0, 1]$ of a random variable as

$$F_X(x) = P(X \leq x) \tag{2.7}$$

Note that X denotes the random variable and x denotes the argument of the distribution function.

2.2.1 Properties

- $0 \leq F_X(x) \leq 1$
- $Lim_{x \rightarrow -\infty} F_X(x) = 0$
- $Lim_{x \rightarrow \infty} F_X(x) = 1$
- $x \leq y \Rightarrow F_X(x) \leq F_X(y)$

The probability density function (pdf) $f_X(x)$ is defined to be that function when integrated yields the distribution function:

$$F_X(x) = \int_{-\infty}^{x} f_X(x)dx \Rightarrow f_X(x) = \frac{dF_X(x)}{dx} \tag{2.8}$$

- $f_X(x) \geq 0$
- $\int_{-\infty}^{\infty} f_X(x)dx = 1$
- $\int_{x \in A} f_X(x)dx = P(x \in A)$

The probability of the random variable falling within a particular range of values (a, b) is given by the integral of this variableŠs density function pdf over that range

$$P(a < X \leq b) = F(b) - F(a) = \int_{a}^{b} f_X(x)dx \tag{2.9}$$

The probability is equal to area under the density function but above the horizontal axis and between the lowest and greatest values of the range (Fig. 2.1).

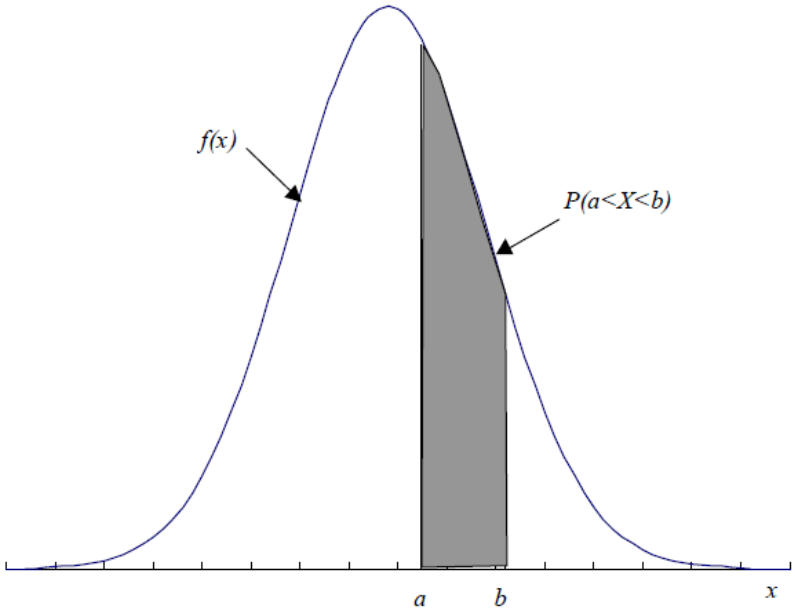


Figure 2.1: Example of probability computation

Intuitively, the expectation of a function of a random variable $E[g(X)]$ can be thought of as a Şweighted averageŤ of the values that the function $g(x)$ can take for different values of x, where the weights are given by $f_X(x)$. As a special case of the above, note that the expectation, $E[X]$ of a random variable itself is found by letting $g(x) = x$; this is also known as the mean of the random variable X.

- $E[a] = a$ for any constant $a \in \mathbb{R}$
- $E[af(X)] = a \cdot E[f(X)]$ for any constant $a \in \mathbb{R}$
- Linearity of expectation:$E[f(X) + g(X)] = E[f(X)] + E[g(X)]$

The variance of a random variable X is the measure of how concentrated the distribution of a random variable X is around its mean. Formally, the variance of a random variable X is defined as

$$Var[(X)] = E[(X - E[X])^2] = E[X^2] - E[X]^2 \tag{2.10}$$

2.2.2 Some common random variables

- $X \sim \text{Uniform}(a, b)$ where $(a < b)$: equal probability function to every value between a and b on the real line.

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{if } a \leq x \leq b \\ 0, & \text{otherwise} \end{cases} \tag{2.11}$$

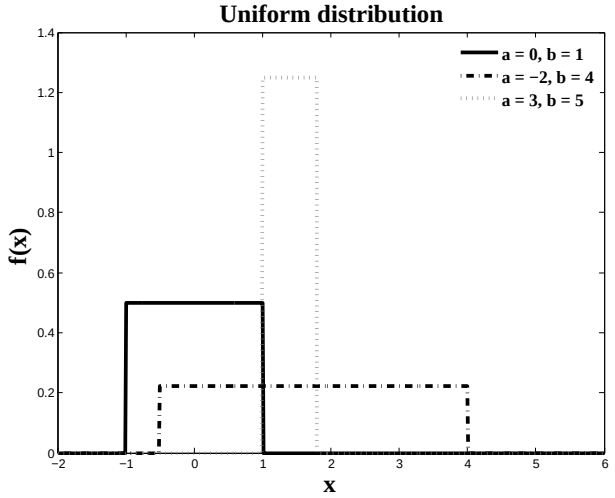


Figure 2.2: pdf for the Uniform distribution for various a, b

- $X \sim \text{Normal}(\mu, \sigma^2)$: also known as the Gaussian distribution

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.12}$$

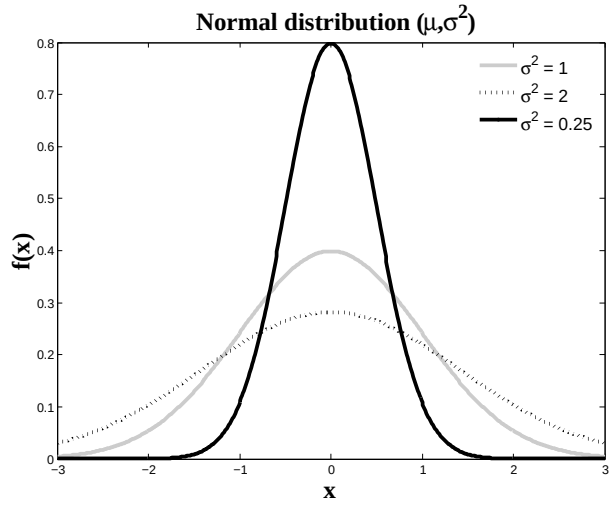


Figure 2.3: pdf for the Normal distribution with $\mu = 0$ and for various values of σ^2

- $X \sim \text{Beta}(a, b)$: is defined in interval $[0, 1]$ parameterized by two positive shape parameters a, b.

$$f(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} \tag{2.13}$$

where $\Gamma(z)$ is the gamma function.

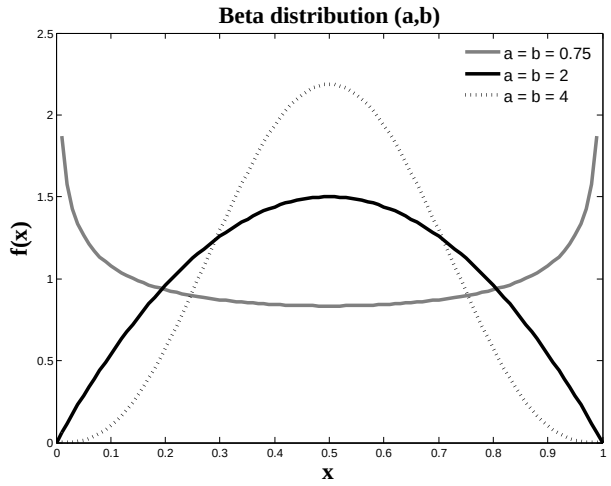


Figure 2.4: pdf for the Beta distribution for various values of the parameters a, b

2.3 Multiple random variables

Suppose we have n continuous random variables $X_1, X_2, ..., X_n$. We can define the joint distribution function,the joint probability density function and the marginal probability density function of X_1 as:

- $F_{\mathbf{X}}(\mathbf{x}) = F_{X_1,X_2,...,X_n}(x_1, x_2, ..., x_n) = P(X_1 \leq x_1, X_2 \leq x_2, ..., X_n \leq x_n)$
- $f_{\mathbf{X}}(\mathbf{x}) = f_{X_1,X_2,...,X_n}(x_1, x_2, ..., x_n) = \frac{\vartheta^n F_{X_1,X_2,...,X_n}(x_1,x_2,...,x_n)}{\vartheta x_1 \vartheta x_2 ... \vartheta x_n})$
- $f_{X_1} = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} f_{X_1,X_2,...,X_n}(x_1, x_2, ..., x_n) dx_2 ... dx_n$

2.3.1 Random vectors

When working with all these random variables together, we will often find it convenient to put them in a vector $\mathbf{X} = [X_1, X_2, ..., X_n]^T$. We call the resulting vector a random vector (more formally, a random vector is a mapping from Ω to $\mathbb{R}^n$. It should be clear that random vectors are simply an alternative notation for dealing with n random variables, so the notions of joint pdf and cdf will apply to random vectors as well. Consider an arbitrary function $g(\mathbf{X}) : \mathbb{R}^n \to \mathbb{R}$ of the random vector $\mathbf{X}$. The mathematical expectation of this function can be defined as:

$$E[g(\mathbf{X})] = \int_{\mathbb{R}^n} g(x_1, x_2, ..., x_n) f_{X_1,X_2,...,X_n}(x_1, x_2, ..., x_n) dx_1 dx_2 ... dx_n \tag{2.14}$$

where $\int_{\mathbb{R}^n}$ is n consecutive integrations from $-\infty$ to ∞. If g is a function from $\mathbb{R}^n \to \mathbb{R}^n$, then the value of g is the element-wise expected values of the output vector. The covariance matrix $\mathbf{C}$ of the random vector is the $n \times n$ matrix whose entries are given by $\Big[C_{ij}\Big]_{n\times n}$=Cov$[X_i, X_j] = E\Big[(X_i - \mu_{X_i})(X_j - \mu_{X_j})\Big]$. The covariance matrix has a number of useful properties like being positive semi-definite and symmetric. The mean value vector $\boldsymbol{\mu}_{\mathbf{X}}$ of a random vector $\mathbf{X}$ is a vector containing the mean value of each component random variable

$$\boldsymbol{\mu}_X = [\mu_{X_1}, \mu_{X_2}, ..., \mu_{X_n}]^T \tag{2.15}$$

2.3.2 The multivariate Gaussian distribution

One particularly important example of a probability distribution over random vectors $\boldsymbol{X}$ is called the multivariate Gaussian or multivariate normal distribution. A random

vector $\boldsymbol{X} \in \mathbb{R}^n$ is said to have a multivariate normal (or Gaussian) distribution with mean $m \in \mathbb{R}^n$ and covariance matrix $\mathbf{C}$.

$$f_{\mathbf{X}}(\mathbf{x}) = f_{X_1,X_2,...,X_n}(x_1, x_2, ..., x_n; \boldsymbol{\mu}; \mathbf{C}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}} exp\Big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu})\Big) \quad (2.16)$$

We write this as $\mathbf{X} \sim (\boldsymbol{\mu}, \mathbf{C})$.

2.3.3 Central Limit theorem

Let $\mathbf{X}_n$ denote a sequence of independent, identically distributed, random variables. Assuming they have zero means and finite variances (equaling σ^2), the Central Limit Theorem states

$$\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{X}_i \sim N(0, \sigma^2) \quad for \quad n \to \infty \quad (2.17)$$

Because of its generality, this theorem is often used to simplify calculations involving finite sums of non- Gaussian random variables. However, attention is seldom paid to the convergence rate of the Central Limit Theorem. Consequently, the Central Limit Theorem, as a finite-sample distributional approximation, is only guaranteed to hold near the mode of the Gaussian, with huge numbers of observations needed to specify the tail behavior.

2.4 Random processes and Random fields

As we recall, a random variable is a rule for assigning to every outcome θ of an experiment a number $x(\theta)$. Thus, a stochastic process (or field) is a family of time (or space) functions depending on the domain θ which is the set of all experimental outcomes (Fig. 2.5). A stochastic process or field $X(t)$ is indexed by a continuous parameter $t \in T$, where T is a continuous set $\mathbb{R}$ of real numbers. This means that for any $t_i \in T$, $X(t_i)$ is a random variable. Usually, a random process refers to the case where the set T denotes a continuous time interval. The notion for a random field is used for describing spatially varying random quantities, i.e. $T = \Omega \subset \mathbb{R}^d$ $d = 1, 2, 3$. Hence, in the general case of some spatial domain Ω, the random field is denoted by $X(\mathbf{t})$, where $\mathbf{t}$ is a location vector. In the following we will use this more general notation, unless otherwise noted. For the determination of the statistical properties of a stochastic process, knowledge of the joint pdf of the random variables is required. However, for many applications, only certain averages are used, in particular, the expected value of $X(\mathbf{t})$

and of $X(\mathbf{t})^2$. Denoting by $f_{X(\mathbf{t})}(x, \mathbf{t})$ the marginal pdf of the random variable $X(\mathbf{t})$, for some $\mathbf{t} \in T$, the mean and variance functions are defined as follows:

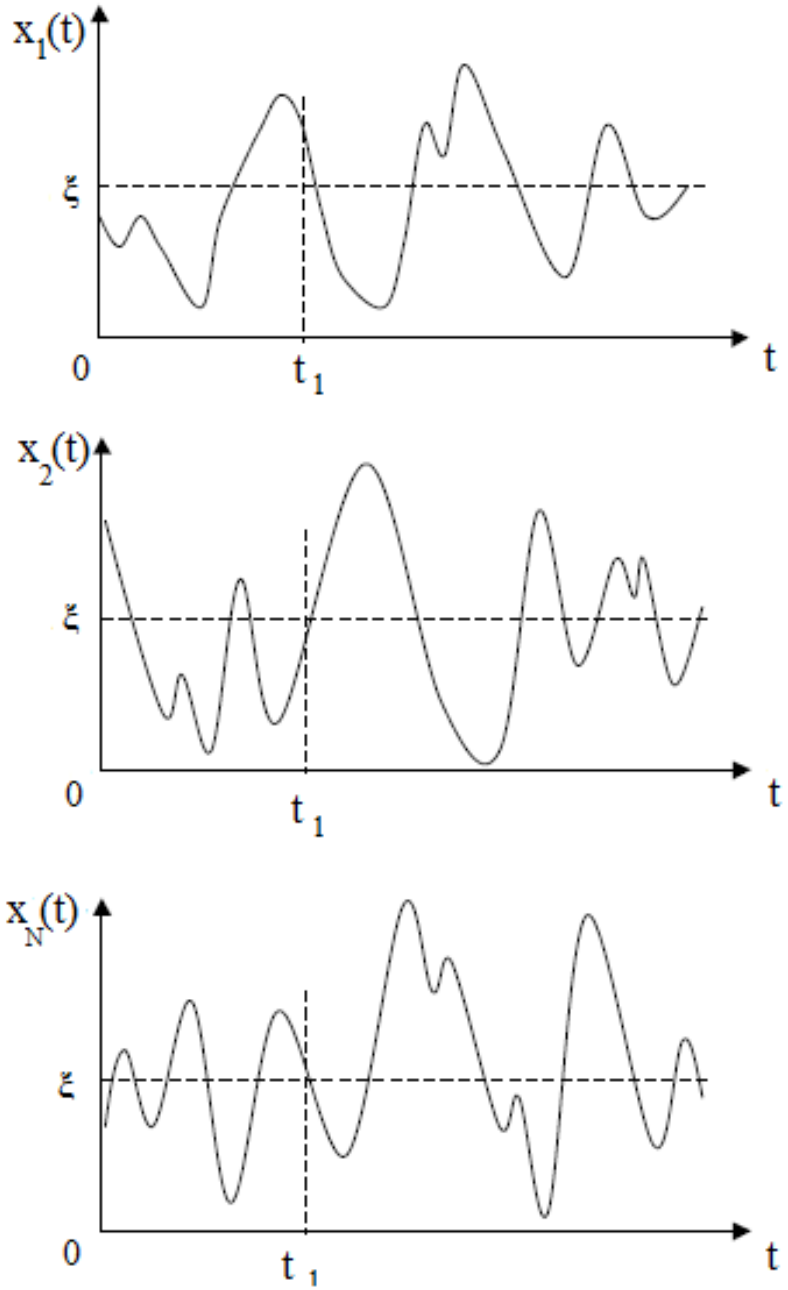


Figure 2.5: Realizations of a random process

$$\mu_X(\mathbf{t}) = E[X(\mathbf{t})] = \int_{-\infty}^{\infty} x f_{X(\mathbf{t})}(x, \mathbf{t}) dx \tag{2.18}$$

$$\sigma_X^2(\mathbf{t}) = E[(X(\mathbf{t}) - m_X(\mathbf{t}))^2] = \int_{-\infty}^{\infty} x f_{X(\mathbf{t})}(x, \mathbf{t}) dx \tag{2.19}$$

The second moment function or autocorrelation function is defined as:

$$R(\mathbf{t}_1, \mathbf{t}_2) = E[X(\mathbf{t}_1)X(\mathbf{t}_2)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 x_2 f_{X(\mathbf{t}_1)} f_{X(\mathbf{t}_2)}(x_1, X_2, \mathbf{t}_1, \mathbf{t}_2) dx_1 x_2 \tag{2.20}$$

The value of $R(\mathbf{t}_1, \mathbf{t}_2)$ on the diagonal $\mathbf{t}_1 = \mathbf{t}_2 = t$ is the average power of $X(\mathbf{t})$: $E\{X^2(\mathbf{t})\} = R(\mathbf{t}, \mathbf{t})$. The autocovariance function is defined as:

$$C(\mathbf{t}_1, \mathbf{t}_2) = R(\mathbf{t}_1, \mathbf{t}_2) - \mu(\mathbf{t}_1)\mu(\mathbf{t}_2) \tag{2.21}$$

The value of $C(\mathbf{t}, \mathbf{t})$ on the diagonal $\mathbf{t}_1 = \mathbf{t}_2 = \mathbf{t}$ equals the variance of $X(\mathbf{t})$. The ratio:

$$r(\mathbf{t}_1, \mathbf{t}_2) = \frac{C(\mathbf{t}_1, \mathbf{t}_2)}{\sqrt{C(\mathbf{t}_1, \mathbf{t}_1)C(\mathbf{t}_2, \mathbf{t}_2)}} \tag{2.22}$$

is the correlation coefficient of the process $X(\mathbf{t})$. Furthermore, $|r(\mathbf{t}_1, \mathbf{t}_2)| \leq 1$ and $r(\mathbf{t}, \mathbf{t}) = 1$.

2.4.1 Homogeneous random fields

A random field $X(\mathbf{t})$ is said to be strictly homogeneous if its probabilistic structure is invariant to a shift in the parameter origin. This means that the field $X(\mathbf{t})$ and $X(\mathbf{t} + \mathbf{c})$ have the same statistics for any $\mathbf{c}$.

$$f(x_1, x_2, ..., x_n; \mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_n) = f(x_1, x_2, ..., x_n; \mathbf{t}_1 + \mathbf{c}, \mathbf{t}_2 + \mathbf{c}, ..., \mathbf{t}_n + \mathbf{c}) \tag{2.23}$$

This leads to the conclusion that

$$f(x_1, x_2; \mathbf{t}_1, \mathbf{t}_2) = f(x_1, x_2; \boldsymbol{\tau}) \quad \boldsymbol{\tau} = \mathbf{t}_1 - \mathbf{t}_2 \tag{2.24}$$

Hence, the mean and standard deviation functions are constant and the autocorrelation, autocovariance and autocorrelation coefficient functions are also functions of $\boldsymbol{\tau}$.

$$\mu(\mathbf{t}) \to \mu \tag{2.25}$$

$$\sigma(\mathbf{t}) \to \sigma \tag{2.26}$$

$$R(\mathbf{t}_1, \mathbf{t}_2) \rightarrow R(\boldsymbol{\tau}) \tag{2.27}$$

$$C(\mathbf{t}_1, \mathbf{t}_2) \rightarrow C(\boldsymbol{\tau}) \tag{2.28}$$

$$r(\mathbf{t}_1, \mathbf{t}_2) \rightarrow r(\boldsymbol{\tau}) \tag{2.29}$$

If the probabilistic structure of a stochastic field $X(\mathbf{t})$ is invariant to a shift in the parameter origin only up to a second order then the process is said to be weakly homogeneous. For a homogeneous random field, the bounds of $R(\boldsymbol{\tau})$ and $C(\boldsymbol{\tau})$ read:

$$|R(\boldsymbol{\tau})| \leq R(0), \qquad R(0) = E[X(\mathbf{t})^2] = const. \tag{2.30}$$

$$C(\boldsymbol{\tau}) \leq \sigma^2 \tag{2.31}$$

The second-order moment functions of a real valued homogeneous field are even functions (i.e. symmetric with respect to the origin $\boldsymbol{\tau} = 0$), e.g:

$$R(\boldsymbol{\tau}_1, ..., \boldsymbol{\tau}_i, ...\boldsymbol{\tau}_d) = R(-\boldsymbol{\tau}_1, ..., -\boldsymbol{\tau}_i, ... - \boldsymbol{\tau}_d), \quad 1 \leq i \leq d \tag{2.32}$$

Several models of autocorrelation coefficient functions for homogeneous random fields have been proposed. Common models are:

$$r(\boldsymbol{\tau}) = e^{-\frac{|\boldsymbol{\tau}|}{b}} \tag{2.33}$$

$$r(\boldsymbol{\tau}) = e^{-(\frac{|\boldsymbol{\tau}|}{b})^2} \tag{2.34}$$

The parameter b is the correlation length of the respective correlation models. A small correlation length signifies fast reduction of the correlation coefficient as the distance $\boldsymbol{\tau}$ increases and thus a high variability in the random field realization. Conversely, large correlation lengths correspond to slowly varying realizations. Moreover, the limit case of an infinite correlation length can be modeled by one random variable.

The Fourier transform of the autocorrelation function $R(\boldsymbol{\tau})$ of a weakly homogeneous stochastic field $X(\mathbf{t})$ gives the power spectral density function (psd) $S(\boldsymbol{\omega})$

$$S(\boldsymbol{\omega}) = \int_{-\infty}^{\infty} R(\boldsymbol{\tau}) e^{-j\boldsymbol{\omega}\boldsymbol{\tau}} d\boldsymbol{\tau} \tag{2.35}$$

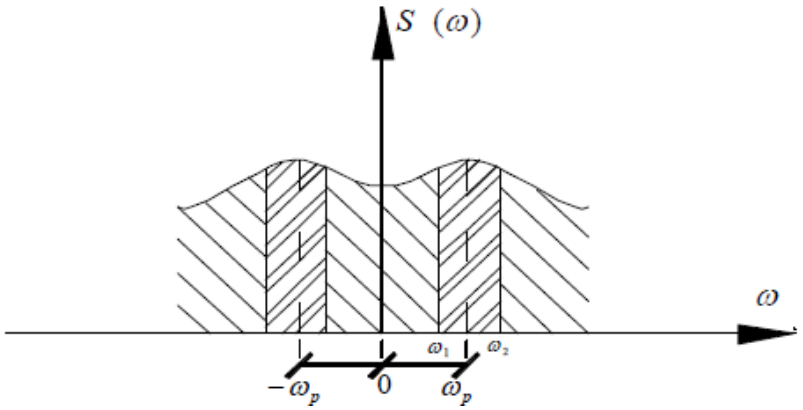


Figure 2.6: Power spectral density function (psd)

As one can notice from from Figure 2.6, the area enclosed by the psd function in the range $[\omega_1, \omega_2]$ represents the variance of the amplitudes of the process in that particular range. The predominant frequency of the process, ω_p, indicates the frequency around whom most of the power of the process is concentrated. From the Fourier inversion formula, it follows that

$$R(\boldsymbol{\tau}) = \int_{-\infty}^{\infty} S(\boldsymbol{\omega}) e^{j\omega\tau} d\boldsymbol{\omega} \tag{2.36}$$

If $X(\mathbf{t})$ is a real valued field, then $R(\boldsymbol{\tau})$ is real and even; hence $S(\boldsymbol{\omega})$ is real and even. In this case

$$R(\boldsymbol{\tau}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\boldsymbol{\omega}) \cos \boldsymbol{\omega\tau} d\boldsymbol{\omega} = \frac{1}{\pi} \int_{0}^{\infty} S(\boldsymbol{\omega}) \cos \boldsymbol{\omega\tau} d\boldsymbol{\omega} \tag{2.37}$$

$$S(\boldsymbol{\omega}) = \int_{-\infty}^{\infty} R(\boldsymbol{\tau}) \cos \boldsymbol{\omega\tau} d\boldsymbol{\tau} = 2 \int_{0}^{\infty} R(\boldsymbol{\tau}) \cos \boldsymbol{\omega\tau} d\boldsymbol{\tau} \tag{2.38}$$

2.4.2 Ergodicity

A second-order homogeneous random field is called ergodic if its second-order information can be obtained from a single realization $x(t)$ of the field. For a one-dimensional random field we have:

$$m = E[X(\mathbf{t})] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(\mathbf{t}) d\mathbf{t} \tag{2.39}$$

$$R(\boldsymbol{\tau}) = E[X(\mathbf{t}+\boldsymbol{\tau})X(t)] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} (\mathbf{t}+\boldsymbol{\tau}) x(\mathbf{t}) d\mathbf{t} \tag{2.40}$$

The concept of ergodicity is of practical importance, since it allows the estimation of the statistics of a random field from a single time or space record. Note that an ergodic field is always homogeneous, but the reverse does not necessary hold.

2.4.3 Gaussian random fields

A random field $X(\mathbf{t})$ is Gaussian if the random variables $\{X(\mathbf{t}_1), X(\mathbf{t}_2), ..., X(\mathbf{t}_n)\}$ are jointly Gaussian for any $\{n, \mathbf{t}_1, \mathbf{t}_2, ...\mathbf{t}_n\}$. Therefore, any linear operation of a Gaussian field result also in a Gaussian field. A Gaussian random field can be completely defined by its mean function $m(\mathbf{t})$ and either its autocovariance function $C(\mathbf{t}_1, \mathbf{t}_2)$ or its autocorrelation function $R(\mathbf{t}_1, \mathbf{t}_2)$. Alternatively, the correlation structure can be defined by the standard deviation function $\sigma(\mathbf{t})$ and the autocorrelation coefficient function $r(\mathbf{t}_1, \mathbf{t}_2)$. If a Gaussian random field is homogeneous, it can be completely defined by its mean value m, standard deviation σ and autocorrelation coefficient function r. The marginal pdf of the Gaussian random field is given by:

$$f_{X(\mathbf{t})}(x, \mathbf{t}) = \frac{1}{\sigma(\mathbf{t})\sqrt{2\pi}} \exp\left[-\frac{(x - \mu(\mathbf{t}))^2}{2\sigma^2(\mathbf{t})} \right] \tag{2.41}$$

2.4.4 Non-Gaussian random fields

In the general case where the random field $X(\mathbf{t})$ in non-Gaussian it is very difficult to obtain the joint pdf of the random variables. However, a class of non-Gaussian random fields with given marginal distribution and second moment information can be defined by s nonlinear marginal transformation (translation) of an underlying Gaussian random field $g(\mathbf{t})$.

$$X(\mathbf{t}) = F^{-1}_{X(\mathbf{t})} \cdot \Phi[g(\mathbf{t})] \tag{2.42}$$

where Φ is the standard Gaussian cumulative distribution function and F is the non-Gaussian marginal cumulative distribution function of $X(\mathbf{t})$. An important issue arising in the context of translation fields is that the choice of the marginal distribution of $X(\mathbf{t})$ imposes constrains to its correlation structure. In other words, F and $S_X(\boldsymbol{\omega})$ have to satisfy a specific compatibility condition derived directly from the definition of the autocorrelation function of the translation field:

$$R^T_X(\boldsymbol{\xi}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F^{-1}[\Phi(g_1)] F^{-1}[\Phi(g_2)] \cdot f[g_1, g_2; R_g(\boldsymbol{\xi})] dg_1 dg_2 \tag{2.43}$$

where, $g_1 = g(\mathbf{t})$, $g_2 = g(\mathbf{t} + \boldsymbol{\xi})$, $f[g_1, g_2; R_g(\boldsymbol{\xi})]$ denotes the joint density of $[g_1, g_2]$. If F and $S^T_X(\boldsymbol{\omega})$ are proven to be incompatible there is no translation field with the prescribed characteristics.In this case, one has to resort to translation fields that match the

target marginal distribution and/or the spectral density function approximately. Within this framework, several researchers have developed techniques for simulating non-Gaussian fields which utilize the translation field theory described above (20, 39, 62, 66, 67, 68, 96, 97, 126, 128, 135, 136, 137, 158).

A different type of translation field can be defined by expansion of the original field using one-dimensional Hermite polynomials with the underlying Gaussian field as argument (128, 132). In addition, it is shown in (125) that a non-Gaussian field can be defined by its marginal distribution and second moment functions in conjunction with the spectral expansion methods without the need for a translation from an equivalent Gaussian field.

2.5 Discretization of random fields

The discretization of a continuous random field $X(\mathbf{t})$ consists in its approximation by a discrete $\hat{X}(\mathbf{t})$ one. Several methods have been proposed for the discretization of random fields- a comprehensive review is given in (78, 145). These methods can be divided in the following three categories:

1 Point discretization methods

2 Average discretization methods

3 Series expansion methods

The most commonly used discretization techniques of the first two categories are, for the point discretization method: the midpoint method, the shape function method and the optimal linear estimation method, while for the average discretization method is the spatial discretization technique. In the next section an overview of the most famous series expansion methods (Spectral representation method and Karhunen-Loève expansion) is provided.

2.5.1 Spectral representation method

Spectral representation method (139) expands the stochastic field as a series of trigonometric functions with random phase angles and amplitudes. Spectral representation generates sample functions that are ergodic in the mean value and autocorrelation (63, 139). For an one-dimensional homogeneous field $X(\mathbf{t})$ with mean value equal to zero, autocorrelation function $R(\boldsymbol{\tau})$ and two-sided power spectral function $S(\omega)$, it is assumed that there exists an upper cut-off frequency ω_u beyond which $S(\omega)$ is zero for either mathematical or physical reasons, i.e.

$$S(\omega) = 0 \qquad for \qquad |\omega| \geqslant \omega_u \tag{2.44}$$

The power spectrum S is related to the autocorrelation function R of the random field through the Wiener–Khintchine transform pair of equations **??**.
In most applications, the following criterion is used for the determination of ω_u,

$$\int_0^{\omega_u} S(\omega)d\omega = (1-\epsilon)\int_0^{\infty} S(\omega)d\omega \tag{2.45}$$

where $\epsilon \ll 1$ is the 'admissible relative error'. The target autocorrelation function is given by

$$R(\boldsymbol{\tau}) = \int_{-\infty}^{\infty} S(\omega)e^{i\omega\boldsymbol{\tau}}d\omega = \int_0^{\infty} 2S(\omega)\cos\omega\boldsymbol{\tau}d\omega \tag{2.46}$$

which is reduced to

$$R^*(\boldsymbol{\tau}) = \int_{-\omega_u}^{\omega_u} S(\omega)e^{i\omega\boldsymbol{\tau}}d\omega = \int_0^{\omega_u} 2S(\omega)\cos\omega\boldsymbol{\tau}d\omega \tag{2.47}$$

The difference between these two functions

$$\epsilon_X^*(\boldsymbol{\tau}) = R(\boldsymbol{\tau}) - R^*(\boldsymbol{\tau}) = \int_{\omega_u}^{\infty} 2S(\omega)\cos\omega\boldsymbol{\tau}d\omega \tag{2.48}$$

corresponds to the simulation error due to the truncation of the spectral density function for $|\omega| \geqslant \omega_u$, which it is termed 'truncation error'.
The simulation formula reads:

$$\hat{X}(\mathbf{t}) = \sqrt{2}\sum_{n=0}^{N-1} A_n \cos(\omega_n \mathbf{t} + \varphi_n) \tag{2.49}$$

where $\varphi_n (n = 0, 1, ...N-1)$ are independent random phase angles uniformly distributed in the range $[0, 2\pi]$; the frequencies are set to

$$\omega_n = n\Delta\omega = n\frac{\omega_u}{N} \qquad for \qquad n = 0, 1, ...N-1 \tag{2.50}$$

and the coefficients A_n are defined as follows

$$A_0 = 0 \qquad A_n = \sqrt{2S(\omega_n)\Delta\omega} \qquad for \qquad n = 0, 1, ...N-1 \tag{2.51}$$

The coefficient A_0 is chosen zero such that the temporal mean value averaged over the whole simulation time $T_0 = \frac{2\pi}{\Delta\omega}$ of the generated stochastic process $x(t)$ remains zero. The simulated stochastic process of eq.(2.49) constitutes the spectral representation of

the random field. It is shown by Shinozuka and Deodatis (140) that $X(\mathbf{t})$ is asymptotically Gaussian as $N \rightarrow \infty$ and ergodic in the mean and in correlation due to the central limit theorem. Its ensemble mean value and the temporal mean value averaged over the whole simulation time T_0 are identical to the corresponding target zero. The ensemble autocorrelation function and the temporal correlation function averaged over the whole simulation time T_0 are identical and converge to the corresponding target $R(\tau)$ linearly as $N \rightarrow \infty$.

For the cases of two and three dimensional random fields, eq.(2.49) takes the form:

$$\hat{X}(\mathbf{t}_1, \mathbf{t}_2) = \sqrt{2} \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} A_{ij}[\cos(\omega_{i\mathbf{t}_1}\mathbf{t}_1 + \omega_{j\mathbf{t}_2}\mathbf{t}_2 + \varphi_{ij}^1) + \\ + \cos(\omega_{i\mathbf{t}_1}\mathbf{t}_1 - \omega_{j\mathbf{t}_2}\mathbf{t}_2 + \varphi_{ij}^2)] \tag{2.52}$$

and

$$\hat{X}(\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3) = \sqrt{2} \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} \sum_{k=0}^{N_3-1} A_{ijk}[\cos(\omega_{i\mathbf{t}_1}\mathbf{t}_1 + \omega_{j\mathbf{t}_2}\mathbf{t}_2 + \omega_{k\mathbf{t}_3}\mathbf{t}_3 + \varphi_{ijk}^1) + \\ + \cos(\omega_{i\mathbf{t}_1}\mathbf{t}_1 + \omega_{j\mathbf{t}_2}\mathbf{t}_2 - \omega_{k\mathbf{t}_3}\mathbf{t}_3 + \varphi_{ijk}^2) + \\ + \cos(\omega_{i\mathbf{t}_1}\mathbf{t}_1 - \omega_{j\mathbf{t}_2}\mathbf{t}_2 + \omega_{k\mathbf{t}_3}\mathbf{t}_3 + \varphi_{ijk}^3) + \\ + \cos(\omega_{i\mathbf{t}_1}\mathbf{t}_1 - \omega_{j\mathbf{t}_2}\mathbf{t}_2 - \omega_{k\mathbf{t}_3}\mathbf{t}_3 + \varphi_{ijk}^4)] \tag{2.53}$$

respectively, with

$$A_{ij} = \sqrt{2S(\omega_{\mathbf{t}_1}, \omega_{\mathbf{t}_2})\Delta\omega_{\mathbf{t}_1}\Delta\omega_{\mathbf{t}_2}} \tag{2.54}$$

$$A_{ijk} = \sqrt{2S(\omega_{\mathbf{t}_1}, \omega_{\mathbf{t}_2}, \omega_{\mathbf{t}_3})\Delta\omega_{\mathbf{t}_1}\Delta\omega_{\mathbf{t}_2}\Delta\omega_{t_3}} \tag{2.55}$$

$$\Delta\omega_{\mathbf{t}_1,\mathbf{t}_2,\mathbf{t}_3} = \frac{\omega_{(\mathbf{t}_1,\mathbf{t}_2,\mathbf{t}_3)u}}{N_{\mathbf{t}_1,t_2,\mathbf{t}_3}} \tag{2.56}$$

$$\omega_{\mathbf{t}_1,\mathbf{t}_2,\mathbf{t}_3} = (\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3)\Delta\omega_{\mathbf{t}_1,\mathbf{t}_2,\mathbf{t}_3} \tag{2.57}$$

and φ_{ijk}^m with $m = 1, ...4$ are independent angle phases generated randomly in the range $[0, 2\pi]$. The number of the angle phases for the cases of the two and three-dimensional random fields are

$$N_{xy} = 2 \times N_x \times N_y \tag{2.58}$$

and

$$N_{xyz} = 4 \times N_x \times N_y \times N_z \tag{2.59}$$

respectively.

2.5.2 The Karhunen-Loève expansion

The Karhunen-Loève (KL) expansion (64, 65, 73, 93) of a one-dimensional random field $X(\mathbf{t})$ with mean value $\mu(\mathbf{t})$ is written as:

$$X(\mathbf{t}, \theta) = \mu(\mathbf{t}) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \xi_i(\theta) f_i(\mathbf{t}) \tag{2.60}$$

$\mathbf{t}$ being a position variable and θ the random event, omitted in the following. In eq.(2.60) λ_i and $f_i(\mathbf{t})$ are the eigenvalues and eigenfunctions of its covariance function $C(\mathbf{t}_1, \mathbf{t}_2)$. By definition, $C(\mathbf{t}_1, \mathbf{t}_2)$ is bounded, symmetric and positive definite with the following spectral or eigen-decomposition:

$$C(\mathbf{t}_1, \mathbf{t}_2) = \sum_{i=1}^{\infty} \lambda_i f_i(\mathbf{t}_1) f_i(\mathbf{t}_2) \tag{2.61}$$

The eigenvalues and eigenfunctions of $C(\mathbf{t}_1, \mathbf{t}_2)$ may be calculated, in the range D of the random field $X(\mathbf{t})$, from the solution of the homogeneous Fredholm integral equation of the second kind given by:

$$\int_D C(\mathbf{t}_1, \mathbf{t}_2) f_i(\mathbf{t}_1) = \lambda_i f_i(\mathbf{t}_2) \tag{2.62}$$

The $\xi_i(\theta)$ in eq.(2.60) is a set of uncorrelated Gaussian random variables, representing the random event θ, which can be expressed as

$$\xi_i(\theta) = \frac{1}{\sqrt{\lambda_i}} \int_D [X(\mathbf{t}, \theta) - \mu(\mathbf{t})] f_i(\mathbf{t}) d\mathbf{t} \tag{2.63}$$

with mean and covariance function given by:

$$\begin{aligned} E[\xi_i(\theta)] &= 0 \\ E[\xi_i(\theta)\xi_j(\theta)] &= \delta_{ij} \end{aligned} \tag{2.64}$$

For all practical purposes, the KL series expansion of eq.(2.60) is approximated by a finite number of M terms, giving

$$\hat{X}(\mathbf{t}) = \mu(\mathbf{t}) + \sum_{i=1}^{M} \sqrt{\lambda_i}\xi_i(\theta) f_i(\mathbf{t}) \tag{2.65}$$

and corresponding covariance function

$$\widehat{C}(\mathbf{t}_1\mathbf{t}_2) = \sum_{i=1}^{M} \lambda_i f_i(\mathbf{t}_1) f_i(\mathbf{t}_2) \tag{2.66}$$

From eq.(2.65), we can compute the variance of the truncation error, which is shown to be:

$$Var[X(\mathbf{t}) - \hat{X}(\mathbf{t})] = \sigma^2(\mathbf{t}) - \sum_{i=0}^{M} \lambda_i \varphi_i^2(\mathbf{t}) \tag{2.67}$$

The integral eigenvalue problem of eq.(2.62) can be solved analytically only for rectangular domains and a few autocovariance functions - see (42, 142) for the triangular and exponential correlation models. In any other case eq.(2.62) can be solved numerically -see reference (11) for an overview on the numerical solution of Fredholm integral equations.

A standard procedure for solving eq.(2.62) in the case where the domain D has an arbitrary shape is the Galerkin method in which a basis function is used in order to represent the eigenfunctions. The shape function $N_i(\mathbf{t})$ $i = 1, .., n$ of the finite element (FE) mesh are used for this purpose. In this sense, the eigenfunctions are represented as:

$$f_i(\mathbf{t}) = \sum_{j=1}^{n} d_i^j N_j(\mathbf{t}) \tag{2.68}$$

where n is the number of nodes of the FE mesh and d_i^j are unknown coefficients. Substituting eq.(2.68) into eq.(2.62) we obtain the following generalized eigenvalue equation:

$$\mathbf{CD} = \mathbf{\Lambda M D} \tag{2.69}$$

where:

$$\mathbf{M} = \left[\int_D N_i(\mathbf{t}) N_j(\mathbf{t}) d\mathbf{t} \right]_{n \times n} \tag{2.70}$$

$$\mathbf{C} = \left[\int_D \int_D C_{XX}(\mathbf{t}_1, \mathbf{t}_2) N_i(\mathbf{t}_1) N_j(\mathbf{t}_2) d\mathbf{t}_1 d\mathbf{t}_2 \right]_{n \times n} \tag{2.71}$$

$$\mathbf{D} = \left[d_i^j\right]_{n\times n} \tag{2.72}$$

$$\mathbf{\Lambda} = \mathrm{diag}\left[\lambda_i\right]_{n\times n} \tag{2.73}$$

If instead of the FE shape functions we use as basis functions a set of orthogonal functions then the derived solution approach is the spectral Galerkin method. The KL expansion can be applied for the discretization of Gaussian random fields. Moreover, if the field is weakly homogeneous, the covariance function is periodic with period equal to the rectangular domain of interest and each eigenfunction is represent by its Fourier series, the KL expansion coincides with the spectral representation.

3

The Finite Element method

The finite element method (FEM) sometimes referred as finite element analysis (FEA) (8, 9, 131, 161), is a computational technique used to obtain approximate solutions of boundary value problems in engineering. Simple stated, a boundary value problem is a mathematical problem in which one or more dependent variables must satisfy a differential equation everywhere within a known domain of independent variables and satisfy specific conditions on the boundary of the domain. Boundary value problems are often called field problems. The field is a domain of interest and most often represents a physical structure.

Certain steps in formulating a finite element analysis of a physical problem are embodied in commercial finite element software packages and are summarized below:

1 **Preprocessing**: The preprocessing step include:

(i). Defining the geometric domain of the problem.

(ii). Defining the element type(s) to be used.

(iii). Define the material properties of the elements.

(iv). Define the geometrical properties of the elements (length, area, etc)

(v). Define the element connectivities (mesh the model).

(vi). Define the physical constraints (boundary conditions).

(vii). Define the loading.

2 **Solution**: During the analysis phase, finite element software assembles the governing algebraic equations in matrix form and computes the unknown values of the primary field variable(s).

$$\mathbf{K}_{ij}\mathbf{u}_j = \mathbf{f}_i \tag{3.1}$$

where **u** and **f** are the displacements and externally applied forces at the nodal points.

3 **Postprocessing**: Analysis and evaluation of the solution results with postprocessor software containing sophisticated routines used for sorting, printing and plotting selected results from a finite element solution.

The geometry depicted in Fig. 3.1 represents the finite element model of several physical problems. For plane stress analysis, the geometry represents a thin plate with a central hole subjected to edge loading in the plane depicted.

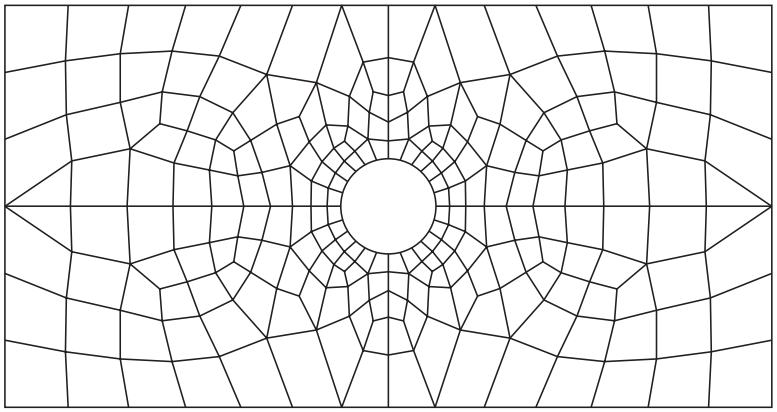


Figure 3.1: A mesh of finite elements over a rectangular region having a central hole.

A linear differential equation can be of the following form:

$$\mathfrak{L}\mathbf{u} + \mathbf{q} = 0 \tag{3.2}$$

where u is the vector of primary variables of the problem, which are functions of the coordinates, $\mathfrak{L}$ is the differential operator and q is the vector of known functions. This differential equation will be subjected to boundary conditions, which are usually of two types:

(i) the essential boundary conditions

(ii) the natural boundary conditions

The essential boundary conditions are the set of boundary conditions that are sufficient for solving the differential equations completely. The natural boundary conditions are the boundary conditions involving higher order derivative terms and are not sufficient

for solving the differential equation completely, requiring at least one essential boundary condition.

Two popular FEM formulations are Galerkin formulation and Ritz formulation. In Galerkin formulation, the primary variable is approximated by a continuous function inside the element. When the approximate primary variable $\mathbf{u}^e$ is substituted in eq.(3.2), we shall get residue depending on the approximating function, i.e.,

$$\mathfrak{Q}\mathbf{u}^e + \mathbf{q} = \mathbf{R} \tag{3.3}$$

Ideally, the residue should be zero everywhere. In that case, approximation becomes equal to true value. As it is very difficult to make the residue 0 at all points, we make the weighted residual equal to zero, i.e.,

$$\int_D \mathbf{w}\mathbf{R}dA \tag{3.4}$$

where $\mathbf{w}$ is the weight function. In order to weaken the requirement on the differentiability of the approximating function, we integrate eq.(3.4) by parts to redistribute the order of derivative in $\mathbf{w}$ and $\mathbf{R}$.

In Galerkin method, the weight function is chosen of the same form as the approximating function. The approximating function is some algebraic function. It is common to replace the unknown coefficients of the function by unknown nodal degrees of freedom. Thus, typically,

$$\mathbf{u}^e = [\mathrm{N}]\mathbf{u}^{ne} \tag{3.5}$$

where [N] is the matrix of shape functions and $\mathbf{u}^{ne}$ is the nodal degrees of freedom.

In Ritz formulation, the differential equation eq.(3.2) is converted into an integral form using calculus of variation. (Sometimes the integral form itself may be easily derivable from the physics of the problem.) The approximation (eq.(3.5)) is substituted in the integral form and the form is extremized by partially differentiating with respect to $\mathbf{u}^{ne}$.

After obtaining the elemental equations, the assembly is performed. A simple way of assembly is to write equations for each element in global form and then add each similar equations of all the elements, i.e., we add the eq.(3.2) from each element to obtain the first global equation, all eq.(3.3) are added together to give second equation, and so on. The boundary conditions are applied to assembled equation and then are solved by a suitable solver. Then, post-processing is carried out to obtain the derivatives

3.1 Formulation for plane stress and plane strain

Consider a linear elastic solid of domain and having uniform thickness bounded by two parallel planes on any closed boundary Γ as shown in Fig. 3.2. The meaning of boundary conditions is explained in Fig. 3.3. If the thickness in z direction is small compared with the size of the domain, the problem may be approximated as a plane stress problem. The following assumptions are made. The body forces, if any exist, cannot vary in the thickness direction and cannot have components in the z direction; the applied boundary forces must be uniformly distributed across the thickness (i.e. constant in the z direction); and no loads can be applied on the parallel planes bounding to the bottom surfaces. The assumption that the forces are zero on the parallel planes implies that for plane stress problems the stresses in the z direction are negligibly small i.e.,

$$\sigma_{xz} = \sigma_{yz} = \sigma_z = 0 \tag{3.6}$$

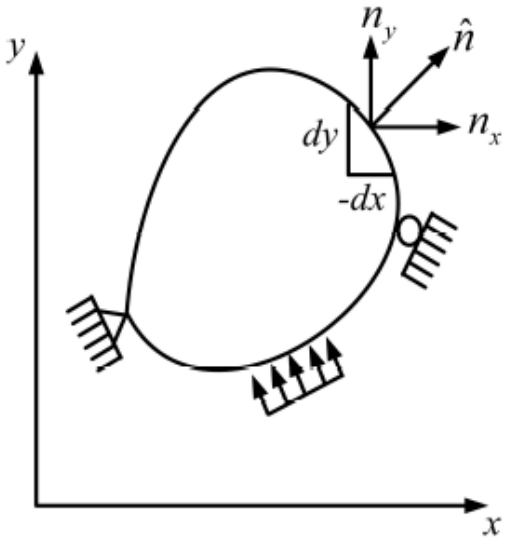


Figure 3.2: A solid of domain Ω.

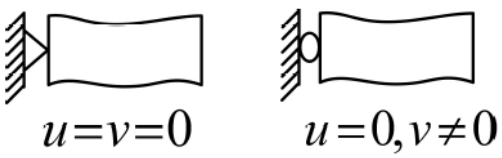


Figure 3.3: Support conditions.

Plane strain is defined as a deformation state in which there is no deformation in z-direction and deformations in other directions are functions of x and y but not of z. Thus, stain components $\varepsilon_z = \varepsilon_{yz} = \varepsilon_{zx} = 0$. In plane strain problems non-zero stress components are σ_x, σ_y, σ_{xy} and σ_z. However, σ_z is not an independent component and can be obtained if σ_x and σ_y are known. This makes the FEM formulation for plane stress and plane strain problems similar. Only difference is in the constitutive matrices for both problems.

The governing equations for the plane elasticity problems are given by

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + f_x = \rho \frac{\partial^2 u}{\partial_t^2} \tag{3.7}$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + f_y = \rho \frac{\partial^2 v}{\partial_t^2} \tag{3.8}$$

where f_x and f_y denote the body forces per unit volume along the x and y directions, respectively and ρ is the density of the material. σ_x and σ_y are the normal stresses and u, v are the displacements in x and y directions respectively, σ_{xy} is the shear stress on the xz and yz planes. Strain-displacement relations are given by

$$\varepsilon_x = \frac{\partial u}{\partial x}, \quad \varepsilon_y = \frac{\partial v}{\partial y}, \quad 2\varepsilon_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \tag{3.9}$$

For plane stress problems, stress and strain are related by the constitutive matrix **D**, in the following manner:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} = \begin{bmatrix} d_{11} & d_{12} & 0 \\ d_{21} & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \cdot \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ 2\varepsilon_{xy} \end{Bmatrix}$$

where d_{ij} ($d_{ij} = d_{ji}$) are the elasticity (material) constants for an orthotropic material with the material principal directions coinciding with the co-ordinate axes (x, y) used to describe the problem. For an isotropic material in plane stress d_{ij} are given by

$$d_{11} = d_{22} = \frac{E}{1-\nu^2}, \quad d_{12} = d_{21} = \frac{E\nu}{1-\nu^2}, \quad d_{33} = \frac{E}{2(1+\nu)} \tag{3.10}$$

where E is Young's modulus of the material and ν is Poisson's ratio. For plane strain problems:

$$d_{11} = d_{22} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}, \quad d_{12} = d_{21} = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad d_{33} = \frac{E}{2(1+\nu)} \tag{3.11}$$

For the given problem, essential or geometric boundary conditions are

$$u = \bar{u} \quad v = \bar{v} \quad \text{on } \Gamma_u \tag{3.12}$$

and natural boundary conditions are

$$t_x = \sigma_x n_x + \sigma_{xy} n_y = \bar{t}_x \tag{3.13}$$
$$t_y = \sigma_{xy} n_x + \sigma_y n_y = \bar{t}_y \quad \text{on } \Gamma_t \tag{3.14}$$

wheren_x, n_y are the components of the unit normal vector n on the boundary Γ. Γ_u and Γ_t are portions of the boundary Γ ($\Gamma = \Gamma_u \bigcup \Gamma_t$). $\bar{t}_x$, $\bar{t}_y$ are specified boundary stresses or tractions, and $\bar{u}$, $\bar{v}$ specified displacements. Only one element of each pair, (u, t_x) and (v, t_y) may be specified at a boundary point.

For plane elastic body, the total potential energy of an element is given by (using index notations),

$$\Pi^e = \int_{V^e} \frac{1}{2} \sigma_{ij} \varepsilon_{ij} dV - \int_{V^e} f_i u_i dV - \int_{\Gamma^e} \bar{t}_i v_i dS \tag{3.15}$$

where, V^e denotes the volume of element e, Γ^e is the boundary of domain Ω^e, σ_{ij} and ε_{ij} are components of stress and strain tensors, respectively and f_i and t_i are the components of body force and boundary stress vectors, respectively. Note that

$$\sigma_{11} = \sigma_x, \ \sigma_{12} = \sigma_{xy}, \ \sigma_{22} = \sigma_y \tag{3.16}$$
$$f_1 = f_x, \ f_2 = f_y, \ t_1 = t_x, \ t_2 = t_y \tag{3.17}$$

The first term in equation (3.15) corresponds to strain energy stored in the element, the second represents the work potential of the body force, and the third represent the work potential of surface forces. For plane stress problems with thickness h_e, it is assumed that all quantities are independent of the thickness co-ordinates z. Hence,

$$\Pi^e = h_e \int_{\Omega^e} \frac{1}{2} (\sigma_x \varepsilon_x + \sigma_y \varepsilon_y + 2\sigma_{xy} \varepsilon_{xy}) dxdy - h_e \int_{\Omega^e} (f_x u + f_y v) dxdy - h_e \int_{\Gamma^e} (t_x u + t_y v) dS \tag{3.18}$$

The finite element model of the plane elasticity equations is developed using eq.(3.18). The displacements u and v are approximated by the Lagrange family of interpolation functions (shape functions). Let u and v are approximated over Ω^e by the finite element interpolations

$$u \approx \sum_{i=1}^{n} u_i^e N_i^e(x, y), \quad v \approx \sum_{i=1}^{n} v_i^e N_i^e(x, y) \tag{3.19}$$

where n is the number of nodes representing the element e, N_i^e are the displacement shape functions, u_i^e and v_i^e are the nodal displacements in x and y directions respectively. The displacements and strains over element e are given by

$$\begin{Bmatrix} u^e \\ v^e \end{Bmatrix} = \begin{Bmatrix} \sum_{i=1}^n u_i^e N_i^e \\ \sum_{i=1}^n v_i^e N_i^e \end{Bmatrix} = \begin{bmatrix} N_1 & N_2 & \cdots & N_n & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & N_1 & N_2 & \cdots & N_n \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \\ v_1 \\ v_2 \\ \ddots \\ v_n \end{Bmatrix}$$

or

$$\begin{Bmatrix} u^e \\ v^e \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & \cdots & N_n & 0 \\ 0 & N_1 & 0 & N_2 & \cdots & 0 & N_n \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{Bmatrix} \equiv [N^e]\{\Delta^e\}$$

and

$$\{\varepsilon^e\} = [B^e]\{\Delta^e\}, \quad \{\sigma^e\} = [D^e][B^e]\{\Delta^e\} \tag{3.20}$$

where $[B^e] = [T^e][N^e]$ is called Gradient matrix and $[T^e]$ is the matrix of differential operators. Substituting these expressions for the displacements and strains into (3.18), minimizing and differentiating with respect to $\{\Delta^e\}$ we get

$$[k^e]\{\Delta^e\} = \{f^e\} + \{Q^e\} \tag{3.21}$$

where,

$$[k^e] = h_e \int_{\Omega^e} [B^e]^T [D^e][B^e] dxdy \tag{3.22}$$

$$\{f^e\} = h_e \int_{\Omega^e} [N^e]^T \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} dxdy \tag{3.23}$$

$$\{Q^e\} = h_e \int_{\Gamma^e} [N^e]^T \begin{Bmatrix} t_x \\ t_y \end{Bmatrix} ds \tag{3.24}$$

The element stiffness matrix $[k^e]$ is of order $2n \times 2n$ and the elemental load vector $[F^e] = \{f^e\} + \{Q^e\}$ is of order $2n \times 1$ where n is the number of nodes of the element.

3.1.1 Two-dimensional triangular element

Triangular finite element (Fig. 3.4) was the first finite element proposed for continuous problems. Because of simplicity it can be used as an introduction to other elements. The shape function are of the form

$$\begin{aligned} N_i &= \frac{1}{2\triangle}(a_i + b_i x + c_i y) \\ a_i &= x_{i+1}y_{i+2} - x_{i+2}y_{i+1} \\ b_i &= y_{i+1} - y_{i+2} \qquad (3.25) \\ c_i &= x_{i+2} - x_{i+1} \\ \triangle &= \frac{1}{2}(x_2y_3 + x_3y_1 + x_1y_2 - x_2y_1 - x_3y_2 - x_1y_3) \qquad (3.26) \end{aligned}$$

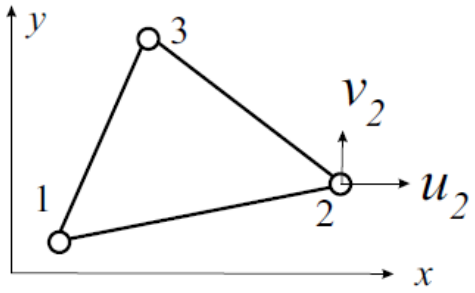


Figure 3.4: Triangular finite element.

where $\triangle$ is the element area. The matrix $[B]$ for interpolating strains using nodal displacements is equal to:

$$[B] = \frac{1}{2}\begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix}$$

It is taken into account that both matrices $[B]$ and $[D]$ do not depend on coordinates. It is also assumed that the element has unit thickness. Since the matrix $[B]$ is constant inside the element the strains and stresses are also constant inside the triangular element.

3.1.2 Two-dimensional isoparametric elements

Isoparametric finite elements are based on the parametric definition of both coordinate and displacement functions. The term ŤisoparametricŤ means that geometry and displacement field are specified in parametric form and are interpolated with the same functions.

Linear and quadratic two-dimensional isoparametric finite elements are presented in Figure 3.5. Shape functions N_i are defined in local coordinates $\xi,\ \eta\ (-1 \leq \xi,\ \eta \leq 1)$. The same shape functions are used for interpolations of displacements and coordinates:

$$u = \sum N_i u_i, \quad v = \sum N_i v_i,$$
$$x = \sum N_i x_i, \quad y = \sum N_i y_i, \tag{3.27}$$

where u, v are displacement components at point with local coordinates (ξ, η); $u_i i, v_i$ are displacement values at the nodes of the finite element; x, y are point coordinates and x_i, y_i are coordinates of element nodes.

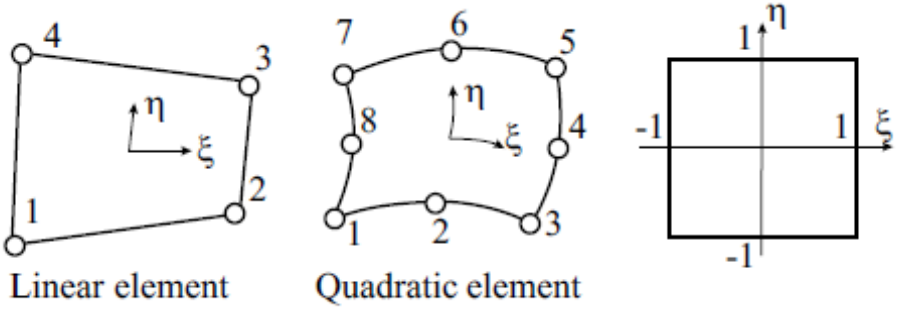


Figure 3.5: Linear and quadratic finite elements and their representation in the local coordinate system.

Shape functions for linear and quadratic two-dimensional isoparametric elements are given by:

linear element (4 nodes):

$$N_i = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i) \tag{3.28}$$

quadratic element (8 nodes):

$$
\begin{aligned}
N_i &= \frac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i) - \frac{1}{4}(1+\xi^2)(1+\eta\eta_i) - \frac{1}{4}(1+\xi\xi_i)(1-\eta^2), \quad i = 1,3,5,7 \\
N_i &= \frac{1}{2}(1-\xi^2)(1+\eta\eta_i), \quad i = 2,6 \\
N_i &= \frac{1}{2}(1+\xi\xi_i)(1-\eta^2), \quad i = 4,8
\end{aligned} \tag{3.29}
$$

The strain-displacement matrix which is employed to compute strains at any point inside the element using nodal displacements is:

$$
[B_i] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix}
$$

While shape functions are expressed through the local coordinates (ξ, η) the strain-displacement matrix contains derivatives in respect to the global coordinates x, y. Derivatives can be easily converted from one coordinate system to the other by means of the chain rule of partial differentiation:

$$
\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \cdot \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix}
$$

where $[J]$ is the Jacobian matrix. The derivatives in respect to the global coordinates are computed with the use of inverse of the Jacobian matrix:

$$
\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix}
$$

The components of the Jacobian matrix are calculated using derivatives of shape functions N_i in respect to the local coordinates (ξ, η) and global coordinates of element nodes x_i, y_i:

$$
\begin{aligned}
\frac{\partial x}{\partial \xi} &= \sum \frac{\partial N_i}{\partial \xi} x_i, \quad \frac{\partial x}{\partial \eta} = \sum \frac{\partial N_i}{\partial \eta} x_i \\
\frac{\partial y}{\partial \xi} &= \sum \frac{\partial N_i}{\partial \xi} y_i, \quad \frac{\partial y}{\partial \eta} = \sum \frac{\partial N_i}{\partial \eta} y_i
\end{aligned} \tag{3.30}
$$

The determinant of the Jacobian matrix $|J|$ is used for the transformation of integrals from the global coordinate system to the local coordinate system:

$$
dV = dxdy|J|d\xi d\eta \tag{3.31}
$$

Integration of expressions for stiffness matrices and load vectors can not be performed analytically for general case of isoparametric elements. Instead, stiffness matrices and load vectors are typically evaluated numerically using Gauss quadrature over quadrilateral regions. The Gauss quadrature formula for the volume integral in two-dimensional case is of the form:

$$I = \int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)d\xi d\eta = \sum_{i=1}^{n}\sum_{j=1}^{n} f(\xi_i,\eta_j)w_i w_j \tag{3.32}$$

where ξ_i, η_j are abscissae and w_i are weighting coefficients of the Gauss integration rule.

3.2 Three-dimensional isoparametric elements

Hexahedral (or brick-type) linear 8-node and quadratic 20-node three-dimensional isoparametric elements are depicted in Fig. 3.6.

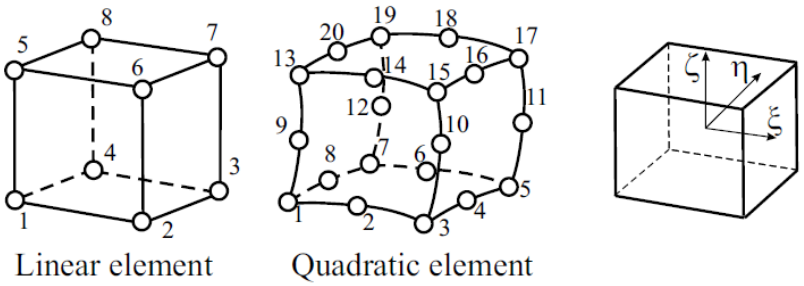


Figure 3.6: Linear and quadratic three-dimensional finite elements and their representation in the local coordinate system.

Shape functions of the linear element are equal to:

$$N_i = \frac{1}{8}(1+\xi\xi_i)(1+\eta\eta_i)(1+\zeta\zeta_i) \tag{3.33}$$

For the quadratic element with 20 nodes the shape functions can be written in the following form:

$$
\begin{aligned}
N_i &= \frac{1}{8}(1+\xi\xi_i)(1+\zeta\zeta_i)(\xi\xi_i+\eta\eta_i+\zeta\zeta_i-2) \quad \text{vertices} \\
N_i &= \frac{1}{4}(1-\xi^2)(1+\eta\eta_i)(1+\zeta\zeta_i), \quad i=2,6,14,18 \qquad (3.34) \\
N_i &= \frac{1}{4}(1-\eta^2)(1+\xi\xi_i)(1+\zeta\zeta_i), \quad i=4,8,16,20 \qquad (3.35) \\
N_i &= \frac{1}{4}(1-\zeta^2)(1+\xi\xi_i)(1+\eta\eta_i), \quad i=9,10,11,12 \qquad (3.36)
\end{aligned}
$$

In the above relations ξ_i, η_i, ζ_i are values of local coordinates ξ, η, ζ at nodes. The strain-displacement matrix for three-dimensional elements has the following appearance:

$$
[B_i] = \begin{bmatrix}
\frac{\partial N_i}{\partial x} & 0 & 0 \\
0 & \frac{\partial N_i}{\partial y} & 0 \\
0 & 0 & \frac{\partial N_i}{\partial z} \\
\frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial z} & 0 \\
0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\
\frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x}
\end{bmatrix}
$$

Derivatives of shape functions in respect to global coordinates are obtained as follows:

$$
\begin{Bmatrix}
\frac{\partial N_i}{\partial x} \\
\frac{\partial N_i}{\partial y} \\
\frac{\partial N_i}{\partial z}
\end{Bmatrix}
= [J]^{-1}
\begin{Bmatrix}
\frac{\partial N_i}{\partial \xi} \\
\frac{\partial N_i}{\partial \eta} \\
\frac{\partial N_i}{\partial \zeta}
\end{Bmatrix}
$$

where the Jacobian matrix has the appearance:

$$
[J] = \begin{bmatrix}
\frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\
\frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\
\frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta}
\end{bmatrix}
$$

The partial derivatives of x, y, z in respect to ξ, η, ζ are found by differentiation of displacements expressed through shape functions and nodal displacement values:

$$
\begin{aligned}
\frac{\partial x}{\partial \xi} &= \sum \frac{\partial N_i}{\partial \xi} x_i, \quad \frac{\partial x}{\partial \eta} = \sum \frac{\partial N_i}{\partial \eta} x_i, \quad \frac{\partial x}{\partial \zeta} = \sum \frac{\partial N_i}{\partial \zeta} x_i \\
\frac{\partial y}{\partial \xi} &= \sum \frac{\partial N_i}{\partial \xi} y_i, \quad \frac{\partial y}{\partial \eta} = \sum \frac{\partial N_i}{\partial \eta} y_i, \quad \frac{\partial y}{\partial \zeta} = \sum \frac{\partial N_i}{\partial \zeta} y_i \qquad (3.37) \\
\frac{\partial z}{\partial \xi} &= \sum \frac{\partial N_i}{\partial \xi} z_i, \quad \frac{\partial z}{\partial \eta} = \sum \frac{\partial N_i}{\partial \eta} z_i, \quad \frac{\partial z}{\partial \zeta} = \sum \frac{\partial N_i}{\partial \zeta} z_i \qquad (3.38)
\end{aligned}
$$

The transformation of integrals from the global coordinate system to the local coordinate system is performed with the use of determinant of the Jacobian matrix:

$$dV = dxdydz|J|d\xi d\eta d\zeta \tag{3.39}$$

The elasticity matrix $[D]$ is

$$[D] = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}$$

whereλ and μ are elastic Lame constants which can be expressed through the elasticity modulus E and PoissonŠs ratio ν:

$$\begin{aligned} \lambda &= \frac{\nu E}{(1+\nu)(1-2\nu)} \\ \mu &= \frac{E}{2(1+\nu)} \end{aligned} \tag{3.40}$$

Integration of the stiffness matrix for three-dimensional isoparametric elements is carried out in the local coordinate system ξ, η, ζ :

$$[k] = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} [B(\xi,\eta,\zeta)]^T [E][B(\xi,\eta,\zeta)]|J|d\xi d\eta d\zeta \tag{3.41}$$

Three-time application of the one-dimensional Gauss quadrature rule leads to the following numerical integration procedure:

$$I = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta)d\xi d\eta d\zeta = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} f(\xi_i,\eta_j,\zeta_k)w_i w_j w_k \tag{3.42}$$

Usually $2 \times 2 \times 2$ integration is used for linear elements and integration $3 \times 3 \times 3$ is applied to the evaluation of the stiffness matrix for quadratic elements. Fore more efficient integration, a special 14- point Gauss-type rule exists, which provides sufficient precision of integration for three-dimensional quadratic elements.

4

Basic principles of artificial neural networks

Artificial Neural Networks (ANN) (2, 3, 4, 16, 25, 30, 34, 46, 82, 109, 152, 153) are information processing models configured for a specific application through a training process. Trained ANN provide with rapid mapping of a given input into the desired output quantities, thereby can be used as meta-models that enhance the computational efficiency of a numerical analysis process. This major advantage of a trained ANN over a conventional numerical analysis procedures, under the provision that the predicted results fall within acceptable tolerances, is that results can be produced in a fraction of wall clock time, requiring orders of magnitude less computational effort than the conventional procedure (7, 21, 70, 71, 87, 95, 104, 129). Basic building block of every ANN is artificial neuron, that is, a simple mathematical model (function). Such a model has three simple sets of rules: multiplication, summation and activation. At the entrance of artificial neuron the inputs are weighted what means that every input value is multiplied with individual weight. In the middle section of artificial neuron is sum function that sums all weighted inputs and bias. At the exit of artificial neuron the sum of previously weighted inputs and bias is passing trough activation function that is also called transfer function (Fig. 4.1).

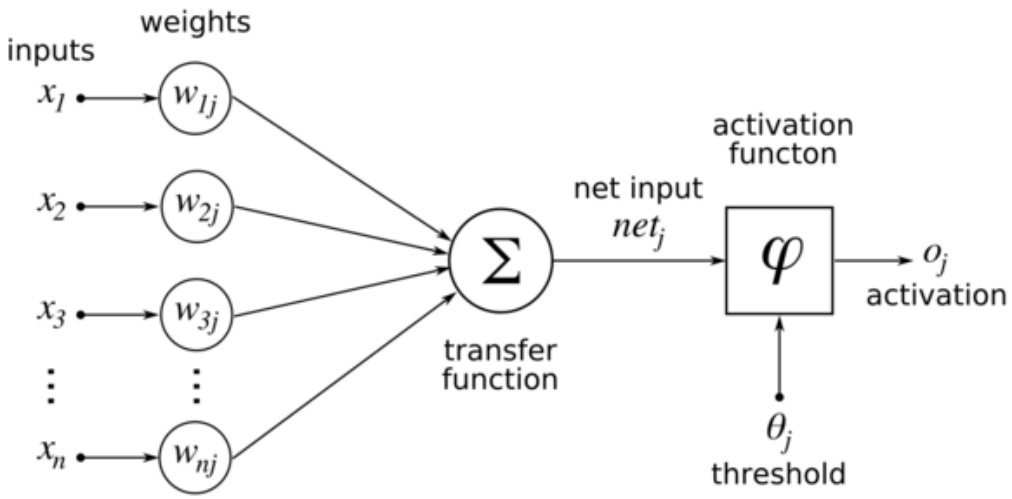


Figure 4.1: Working principle of an artificial neuron (source : Artificial Neural networks, wikibooks.org, 2013.

In order to fully harvest the benefits of mathematical complexity that can be achieved through interconnection of individual artificial neurons we do not interconnect these artificial neurons randomly. In the past, researchers have come up with several ŞstandarisedŤ architectures (topographies)) of artificial neural networks. Different types of artificial neural network architectures are suited for solving different types of problems.

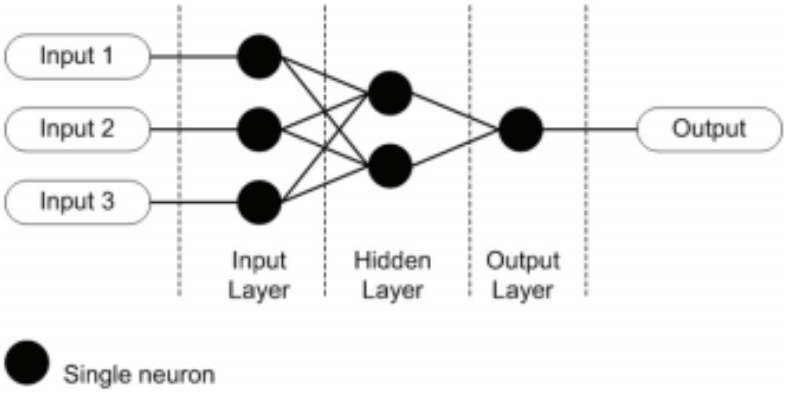


Figure 4.2: Example of simple artificial neural network.

After determining the type of given problem we need to decide for topology of artificial neural network we are going to use and then fine-tune it. We need to fine-tune the topology itself and its parameters.Before we can use our artificial neural network

we need to teach it solving the type of given problem. There are three major learning paradigms: supervised learning, unsupervised learning and reinforcement learning. We choose learning paradigm similar as we chose artificial neuron network topography - based on the problem we are trying to solve. Although learning paradigms are different in their principles they all have one thing in common; on the basis of Şlearning dataŤ and Şlearning rulesŤ (chosen cost function) artificial neural network is trying to achieve proper output response in accordance to input signals. After choosing topology of an artificial neural network, fine-tuning of the topology and when ANN has learn a proper behaviour we can start using it for solving given problem.

4.1 Artificial neuron

Artificial neuron is a basic building block of every artificial neural network. Its design and functionalities are derived from observation of a biological neuron that is basic building block of biological neural networks (systems) which includes the brain, spinal cord and peripheral ganglia. Similarities in design and functionalities can be seen in Fig.4.3. where the left side of a figure represents a biological neuron with its soma, dendrites and axon and where the right side of a figure represents an artificial neuron with its inputs, weights, transfer function, bias and outputs.

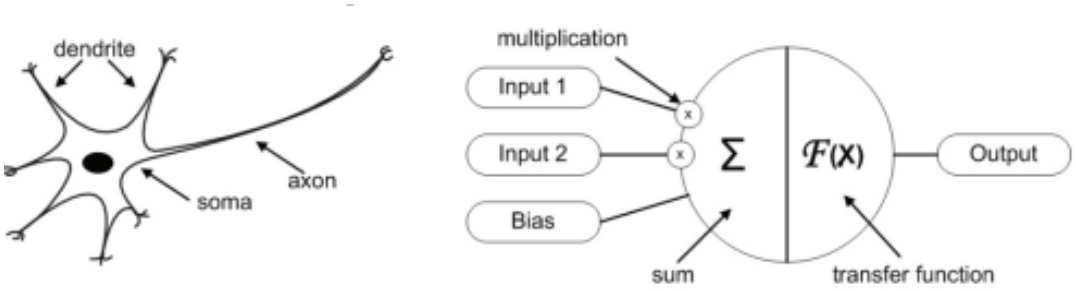


Figure 4.3: Biological and artificial neuron design.

In case of biological neuron information comes into the neuron via dendrite, soma processes the information and passes it on via axon. In case of artificial neuron the information comes into the body of an artificial neuron via inputs that are weighted (each input can be individually multiplied with a weight). The body of an artificial neuron then sums the weighted inputs, bias and ŞprocessesŤ the sum with a transfer function. At the end an artificial neuron passes the processed information via output(s). Benefit of artificial neuron model simplicity can be seen in its mathematical description

below:

$$y(\mathbf{t}) = F\left(\sum_{i=0}^{m} w_i(\mathbf{t}) \cdot x_i(\mathbf{t}) + b\right) \tag{4.1}$$

where,

- $x_i(\mathbf{t})$ is input value
- $w_i(\mathbf{t})$ is weight value
- b is bias
- F is transfer function
- $y_i(\mathbf{t})$ is output value

As seen from a model of an artificial neuron and its eq.(4.1) the major unknown variable of our model is its transfer function. Transfer function defines the properties of artificial neuron and can be any mathematical function. We choose it on the basis of problem that ANN needs to solve and in most cases we choose it from the following set of functions: Step function, Linear function and Non-linear (Sigmoid) function.

Step function is binary function that has only two possible output values (e.g. zero and one). That means if input value meets specific threshold the output value results in one value and if specific threshold is not meet that results in different output value. Situation can be described with eq.(4.2).

$$y = \begin{cases} 1, & \text{if } w_i x_i \geq \text{threshold} \\ 0, & \text{if } w_i x_i < \text{threshold} \end{cases} \tag{4.2}$$

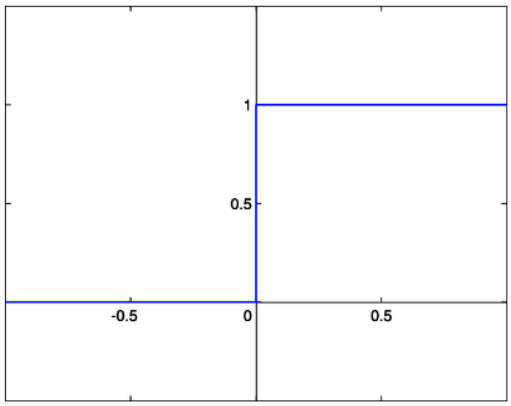


Figure 4.4: Step function.

When this type of transfer function is used in artificial neuron we call this artificial neuron perceptron (106). Perceptron is used for solving classification problems and as such it can be most commonly found in the last layer of artificial neural networks. In case of linear transfer function artificial neuron is doing simple linear transformation over the sum of weighted inputs and bias. Such an artificial neuron is in contrast to perceptron most commonly used in the input layer of artificial neural networks. When we use non-linear function the sigmoid function is the most commonly used. Sigmoid function has easily calculated derivate, which can be important when calculating weight updates in the artificial neural network.

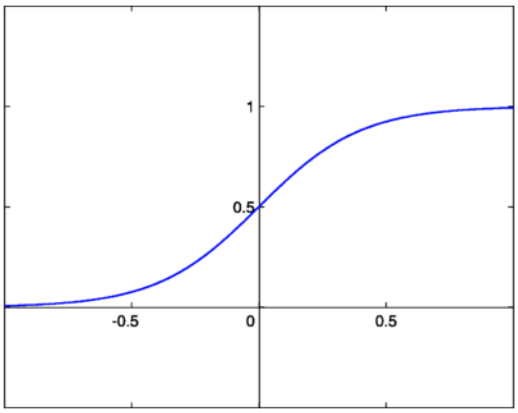


Figure 4.5: Sigmoid function.

4.2 Architecture of ANN

When combining two or more artificial neurons we are getting an artificial neural network. If single artificial neuron has almost no usefulness in solving real-life problems the artificial neural networks have it. In fact artificial neural networks are capable of solving complex real-life problems by processing information in their basic building blocks (artificial neurons) in a non-linear, distributed, parallel and local way.

The way that individual artificial neurons are interconnected is called topology, architecture or graph **A** of an artificial neural network. The fact that interconnection can be done in numerous ways results in numerous possible topologies that are divided into two basic classes. Fig.4.6 shows these a taxonomy of these two topologies; In a simple feed-forward topology (acyclic graph) information flows from inputs to outputs in only one direction while in a simple recurrent topology (semi-cyclic graph) some information flows not only in one direction from input to output but also in opposite direction. It must be mentioned that for easier handling and mathematical describing of an artificial network individual neurons are grouped into layers.

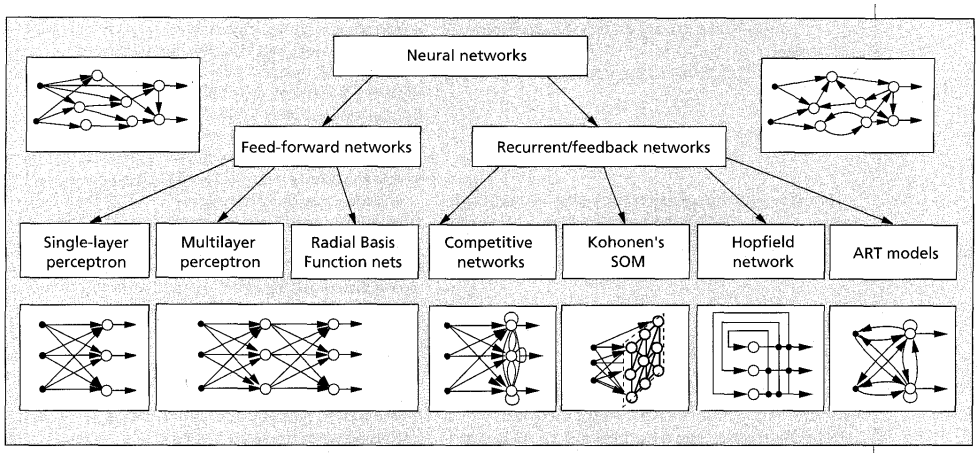


Figure 4.6: A taxonomy of feed-forward and recurrent/feedback network architectures (source : Artificial neural networks- a tutorial, 1996)

The next step is to learn proper response of an ANN and this can be achieved through learning (supervised, un-supervised or reinforcement learning). No matter which method we use, the task of learning is to set the values of weight and biases on basis of learning data to minimize the chosen cost function

4.2.1 Feedforward ANN

ANN with feed-forward architecture is called feedforward artificial neural network and as such has only one condition: information must flow from input to output in only one direction with no back-loops. There are no limitations on number of layers, type of transfer function used in individual artificial neuron or number of connections between individual artificial neurons.

Feed-forward architectures contain no cycles, i.e., the architecture of a feed-forward neural network can then be represented as an acyclic graph. Hence, neurons in a feed-forward neural network are grouped into a sequence of $c + 1$ layers $L^1, ...L^{c+1}$, so that neurons in any layer are connected only to neurons in the next layer. The input layer L^0 consists of n external inputs and is not counted as a layer of neurons; the hidden layers $L^1, ...L^{c+1}$ contain $h_1, ..., h_c$ hidden neurons, respectively; and the output layer L^{c+1} is composed of m output neurons. Communication proceeds layer by layer from the input layer via the hidden layers up to the output layer.

The simplest feed-forward artificial neural network is a single perceptron that is only capable of learning linear separable problems. Simple multi-layer feed-forward artificial neural network is shown in Fig.4.7.

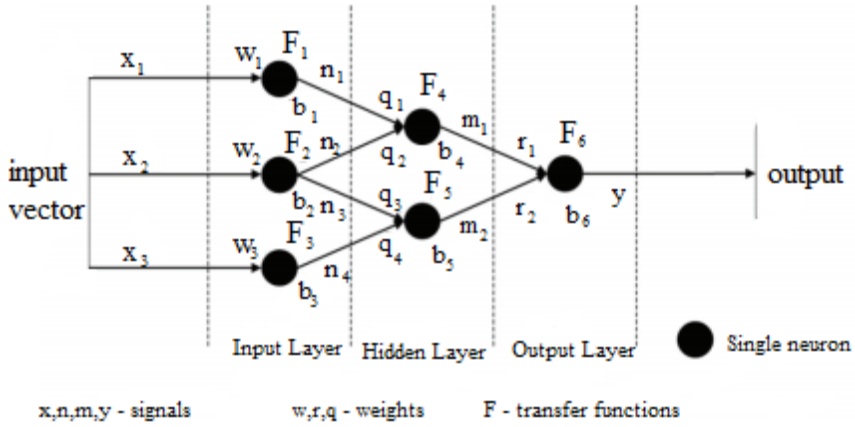


Figure 4.7: Feed-forward (Feed-forward artificial neural network.

Eqs.(4.3),(4.4) and (4.5) give an analytical description of how this feed-forward neural network works

$$\begin{aligned} n_1 &= F_1(w_1 x_1 + b_1) \\ n_2 &= F_2(w_2 x_2 + b_2) \\ n_3 &= F_2(w_2 x_2 + b_2) \\ n_4 &= F_3(w_3 x_3 + b_3) \end{aligned} \tag{4.3}$$

$$\begin{aligned} m_1 &= F_4(q_1 n_1 + q_2 n_2 + b_4) \\ m_2 &= F_5(q_3 n_3 + q_4 n_4 + b_5) \end{aligned} \tag{4.4}$$

$$y = F_6(r_1 m_1 + r_2 m_2 + b_6) \tag{4.5}$$

4.2.2 Recurrent ANN

Artificial neural network with the recurrent topology is called Recurrent artificial neural network. It is similar to feed-forward neural network with no limitations regarding back-loops. In these cases information is no longer transmitted only in one direction but it is also transmitted backwards. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. Recurrent artificial neural networks can use their internal memory to process any sequence of inputs.

The most basic topology of recurrent artificial neural network is fully recurrent artificial network where every basic building block (artificial neuron) is directly connected to every other basic building block in all direction. Other recurrent artificial

neural networks such as Hopfield, Elman, Jordan, bi-directional and other networks are just special cases of recurrent artificial neural networks.

4.3 The learning algorithm

There are three major learning paradigms; supervised learning, unsupervised learning and reinforcement learning. Usually they can be employed by any given type of artificial neural network architecture. Each learning paradigm has many training algorithms.

4.3.1 Supervised learning

Supervised learning is a machine learning technique that sets parameters of an artificial neural network from training data. The task of the learning artificial neural network is to set the value of its parameters for any valid input value after having seen output value. The training data consist of pairs of input and desired output values that are traditionally represented in data vectors. Supervised learning can also be referred as classification, where we have a wide range of classifiers, each with its strengths and weaknesses (Multilayer perceptron, Support Vector Machines, k-nearest neighbour algorithm, Gaussian mixture model, Gaussian, naive Bayes, decision tree, radial basis function classifiers, e.t.c).

In order to solve a given problem of supervised learning various steps have to be considered. In the first step we have to determine the type of training examples. In the second step we need to gather a training data set that satisfactory describe a given problem. In the third step we need to describe gathered training data set in form understandable to a chosen artificial neural network. In the fourth step we do the learning and after the learning we can test the performance of learned artificial neural network with the test (validation) data set. Test data set consist of data that has not been introduced to artificial neural network while learning.

A learning algorithm tries to determine the values of $\boldsymbol{w}$, in order to achieve the correct response for each input vector applied to the network by minimizing the value of $\mathbf{E}_D$.

$$\mathbf{E}_D(\mathbf{D}|\boldsymbol{w}, \mathbf{A}) = \sum_m \frac{1}{2}(\boldsymbol{y}(\boldsymbol{x}^m; \boldsymbol{w}, \mathbf{A}) - \boldsymbol{t}^m)^2 \tag{4.6}$$

The numerical minimization algorithms used for the training generate a sequence of weight parameters $\boldsymbol{w}$ through an iterative procedure. To apply an algorithmic operator $\mathbf{O}$ the starting weight parameters $\boldsymbol{w}^{(0)}$ are needed, while the update formula can be

written as follows:

$$\boldsymbol{w}^{(t+1)} = \mathbf{O}(\boldsymbol{w}^{(t)}) = \boldsymbol{w}^{(t)} + \Delta\boldsymbol{w}^{(t)} \tag{4.7}$$

The increment of the weight parameter $\Delta\boldsymbol{w}^{(t)}$ is further decomposed into

$$\Delta\boldsymbol{w}^{(t)} = a_t\boldsymbol{d}^t \tag{4.8}$$

where $\boldsymbol{d}^t$ is a search direction vector and a_t is the step size along this direction.

The training methods can be divided into two categories. Algorithms that use global knowledge of the state of the entire network, such as the direction of the overall weight update vector, which are referred to as global techniques. In contrast local adaptation strategies are based on weight specific information only such as the temporal behavior of the partial derivative of this weight. The local approach is more closely related to the ANN concept of distributed processing in which computations can be made independent to each other. Furthermore, it appears that for many applications local strategies achieve faster and reliable prediction than global techniques despite the fact that they use less information (133).

4.3.1.1 Global adaptive techniques

The algorithms most frequently used in the ANN training are the steepest descent, the conjugate gradient and the Newton's methods with the following direction vectors:

- Steepest descent method: $\mathbf{d}^{(t)} = -\nabla E(\mathbf{w}^{(t)})$
- Conjugate gradient method: $\mathbf{d}^{(t)} = -\nabla E(\mathbf{w}^{(t)}) + \beta_{t-1}\mathbf{d}^{(t-1)}$ where β_{t-1} is defined: $\beta_{t-1} = \nabla E_t \cdot \nabla E_t / \nabla E_{t-1} \cdot \nabla E_{t-1}$ Fletcher-Reeves
- Newton's method: $\mathbf{d}^{(t)} = -[H(\mathbf{w}^{(t)})]^{-1}\nabla E(\mathbf{w}^{(t)})$

The convergence properties of optimization algorithms for differentiable functions depend on properties of the first and/or second derivatives of the function to be optimized. When optimization algorithms converge slowly for ANN problems, this suggests that the corresponding derivative matrices are numerically ill-conditioned (88).

4.3.1.2 Local adaptive techniques

To improve the performance of weight updating, two completely different approaches have been proposed, namely Quickprop (47) and Rprop (130).

The Quickprop method

This method is based on a heuristic learning algorithm for a multi-layer perceptron, developed by Fahlman (47), which is partially based on the Newton's method. Quickprop is one of most frequently used adaptive learning paradigms. The weight updates are based on estimates of the position of the minimum for each weight, obtained by solving the following equation for the two following partial derivatives

$$\frac{\partial E_{t-1}}{\partial \mathbf{w}_{ij}} \quad \text{and} \quad \frac{\partial E_t}{\partial \mathbf{w}_{ij}} \tag{4.9}$$

and the weight update is implemented as follows:

$$\Delta \mathbf{w}_{ij}^{(t)} = \frac{\frac{\partial E_t}{\partial w_{ij}}}{\frac{\partial E_{t-1}}{\partial \mathbf{w}_{ij}} - \frac{\partial E_t}{\partial \mathbf{w}_{ij}}} \Delta \mathbf{w}_{ij}^{(t-1)} \tag{4.10}$$

The learning time can be remarkably improved compared to the global adaptive techniques.

The Rprop method

Another heuristic learning algorithm with locally adaptive learning rates based on an adaptive version of the Manhattan-learning rule and developed by Riedmiller and Braun is the **R**esilient back**prop**agation abbreviated as Rprop (130). The weight updates can be written

$$\Delta \mathbf{w}_{ij}^{(t)} = -\eta_{ij}^{(t)} sgn(\frac{\partial E_{t-1}}{\partial \mathbf{w}_{ij}}), \tag{4.11}$$

where

$$\eta_{ij} = \begin{cases} min(\alpha \cdot \eta_{ij}^{(t-1)}, \eta_{max}), & if \frac{\partial E_t}{\partial \mathbf{w}_{ij}} \cdot \frac{\partial E_{t-1}}{\partial \mathbf{w}_{ij}} > 0 \\ max(b \cdot \eta_{ij}^{(t-1)}, \eta_{min}), & if \frac{\partial E_t}{\partial \mathbf{w}_{ij}} \cdot \frac{\partial E_{t-1}}{\partial \mathbf{w}_{ij}} < 0 \\ \eta_{ij}^{(t-1)} & otherwise \end{cases} \tag{4.12}$$

where $\alpha = 1.2$, $b = 0.5$, $\eta_{max} = 50$ and $\eta_{min} = 0.1$ (95). The learning rates are bounded by upper and lower limits in order to avoid oscillations and arithmetic underflow. It is interesting to note that, in contrast to other algorithms, Rprop employs information about the sign and not the magnitude of the gradient components.

4.3.2 Unsupervised learning

Unsupervised learning is a machine learning technique that sets parameters of an artificial neural network based on given data and a cost function which is to be minimized. Cost function can be any function and it is determined by the task formulation.

Unsupervised learning is mostly used in applications that fall within the domain of estimation problems such as statistical modeling, compression, filtering, blind source separation and clustering. In unsupervised learning we seek to determine how the data is organized. It differs from supervised learning and reinforcement learning in that the artificial neural network is given only unlabeled examples. One common form of unsupervised learning is clustering where we try to categorize data in different clusters by their similarity. Among above described artificial neural network models, the Self-organizing maps are the ones that the most commonly use unsupervised learning algorithms.

4.3.3 Reinforcement learning

Reinforcement learning is a machine learning technique that sets parameters of an artificial neural network, where data is usually not given, but generated by interactions with the environment. Reinforcement learning is concerned with how an artificial neural network ought to take actions in an environment so as to maximize some notion of long-term reward. Reinforcement learning is frequently used as a part of artificial neural networkŠs overall learning algorithm.

After return function that needs to be maximized is defined, reinforcement learning uses several algorithms to find the policy which produces the maximum return. Naive brute force algorithm in first step calculates return function for each possible policy and chooses the policy with the largest return. Obvious weakness of this algorithm is in case of extremely large or even infinite number of possible policies. This weakness can be overcome by value function approaches or direct policy estimation. Value function approaches attempt to find a policy that maximizes the return by maintaining a set of estimates of expected returns for one policy; usually either the current or the optimal estimates. These methods converge to the correct estimates for a fixed policy and can also be used to find the optimal policy. Similar as value function approaches the direct policy estimation can also find the optimal policy. It can find it by searching it directly in policy space what greatly increases the computational cost.

4. BASIC PRINCIPLES OF ARTIFICIAL NEURAL NETWORKS

5

Structural Reliability analysis

5.1 An overview on reliability analysis

The inherent probabilistic nature of design parameters, material properties and loading conditions involved in structural analysis is an important factor that influences structural safety. Reliability analysis leads to safety measures that a design engineer has to take into account due to the aforementioned uncertainties. Consider the reliability problem

$$P_f = P(R - S \leq 0) \leq P_d \tag{5.1}$$

wherein the limit-state criterion is defined in terms of the load effect S and the resistance R, each of which is described by a known probability density function $f_S(s)$, $f_R(r)$ respectively and P_d is a predefined probability of failure. The probability of failure P_f of the structural component is as follows:

$$P_f = P(R - S \leq 0) = P[g(R - S) \leq 0] \tag{5.2}$$

where $g(\cdot)$ is termed "limit-state function" with negative values defining the failure scenario. The random variables R, S have a joint probability density function denoted by $f_{R,S}(r, s)$. So, the failure probability of eq.(5.2) becomes:

$$P_f = P(R - S \leq 0) = \int\int_{r-s\leq 0} f_{R,S}(r, s) dr ds \tag{5.3}$$

If the two variables are statistically independent, then the joint PDF is expressed as:

$$f_{R,S}(r, s) = f_R(r) \cdot f_S(s) \tag{5.4}$$

In this case eq.(5.3) gives

$$P_f = P(R - S \leq 0) = \int_{-\infty}^{\infty} \int_{-\infty}^{r \leq s} f_R(r) f_S(s) dr ds \tag{5.5}$$

Quite general density functions f_R and f_S for R and S respectively are shown in Fig 5.1 together with the joint (bivariate) density function $f_{R,S}(r, s)$. For any infinitesimal element $(\Delta r \Delta s)$ the latter represents the probability that R takes on a value between r and $r + \Delta r$ and S a value between s and $s + \Delta s$ as Δr and Δs each approach zero.

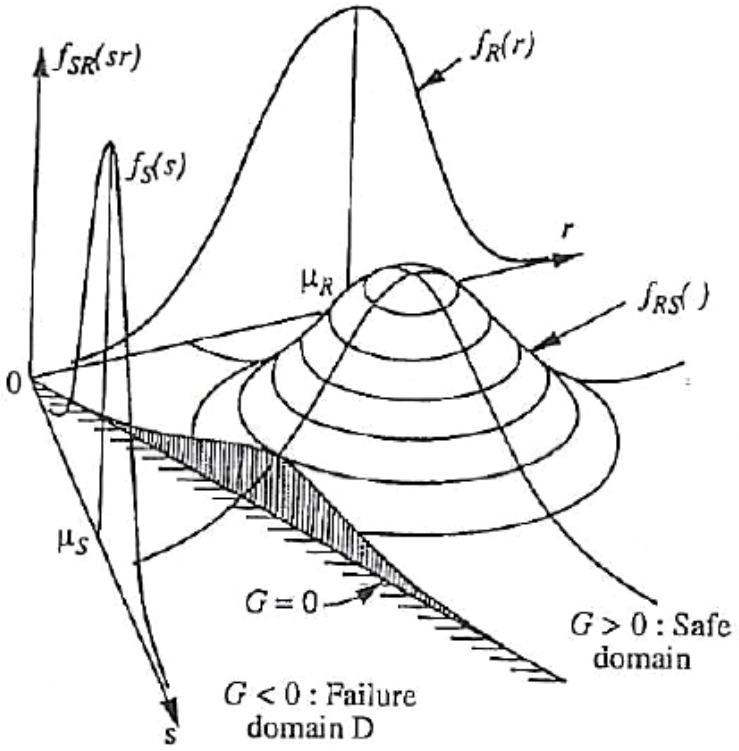


Figure 5.1: Joint density function $f_{R,S}(r, s)$, marginal density functions $f_R(r)$ and $f_S(s)$ and failure domain D, (Melchers, 1999)

Equation (5.3) also known as a convolution integral with meaning easily explained by reference to Fig. 5.2 (102). $F_R(x)$ is the probability the actual resistance R of the member is less than some value x. Let this represent failure. The term $f_S(x)$ represents the probability that the load effect S acting in the member has a value between x and $x + \Delta x$ in the limit as $\Delta x \to 0$. By considering all possible values of x, i.e. by taking the integral over all x, the total probability of failure is obtained. This is also seen in

Fig. 5.3 where the density functions $f_R(r)$ and $f_S(s)$ have been drawn along the same axis.

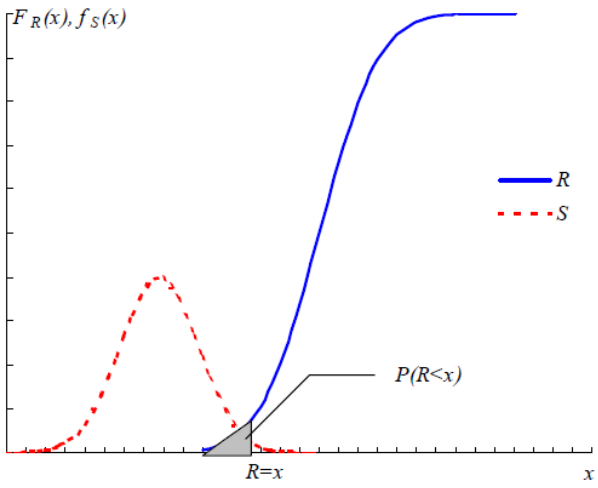


Figure 5.2: Basic $R - S$ problem: $f_R(r)$ $f_S(s)$ representation

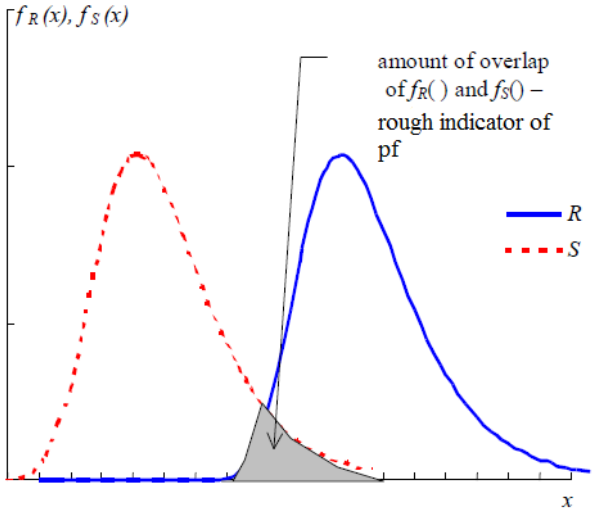


Figure 5.3: Basic $R - S$ problem: $f_R(r)$ $f_S(s)$ representation

For a few distributions of R and S it is possible to integrate the convolution integral of eq.(5.3) analytically. One notable example is when both are normal random variables with means μ_R and μ_S and variances σ_R^2 and σ_S^2 respectively. The safety margin $Z = R - S$ then has a mean and variance given by well-known rules for addition of normal random variables:

$$\mu_Z = \mu_R - \mu_S$$
$$\sigma_Z^2 = \sqrt{\sigma_R^2 + \sigma_S^2} \tag{5.6}$$

Equation (5.3) becomes

$$P_f = P(R - S \leq 0) = P(Z \leq 0) = \Phi\left(\frac{0 - \mu_Z}{\sigma_Z}\right) \tag{5.7}$$

where $\Phi()$ is the standard normal distribution function (zero mean and unit variance). The random variable $Z = R - S$ is shown in Figure 5.4, in which the failure region $Z \leq 0$ is shown shaded. Using (5.6) and (5.7) it follows that (33)

$$P_f = \Phi\left[\frac{-(\mu_R - \mu_S)}{(\sigma_R^2 + \sigma_S^2)^{1/2}}\right] = \Phi(-\beta) \tag{5.8}$$

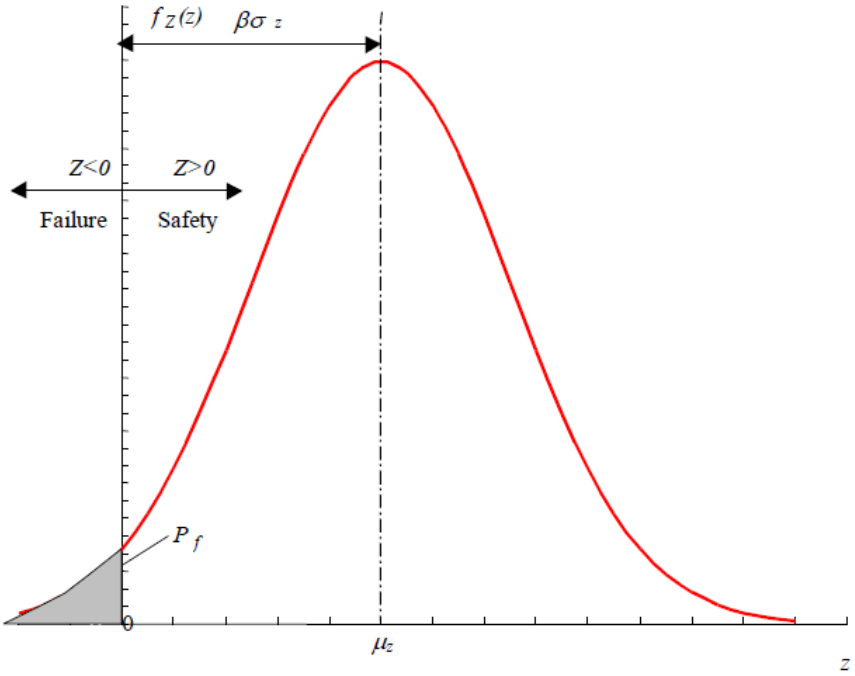


Figure 5.4: Distribution of safety margin $Z = R - S$

where $\beta = \mu_Z/\sigma_Z$ is defined as reliability (safety) index. If either of the standard deviations σ_R and σ_S or both are increased, the term in square brackets in (5.8) will become smaller and hence P_f will increase. Similarly, if the difference between the

mean of the load effect and the mean of the resistance is reduced, P_f increases. These observations may be deduced also from Figure 4.3, taking the amount of overlap of $f_R()$ and $f_S()$ as a rough indicator of P_f.

The reliability index β has the simple geometrical interpretation as the smallest distance from the line (or generally the hyper-plane) forming the boundary between the safe domain and the failure domain, i.e. the domain defined by the failure event and the origin in standard normal distributed space U

$$U_i = \frac{X_i - \mu_{X_i}}{\sigma_{X_i}} \tag{5.9}$$

in which case the components of U have zero means and variances equal to 1. It should be noted that this definition of the reliability index (due to Hasofer and Lind (69)) does not depend on the limit state function but rather the boundary between the safe domain and the failure domain. The point on the failure surface with the smallest distance to origin is commonly denoted the design point or most likely failure point.

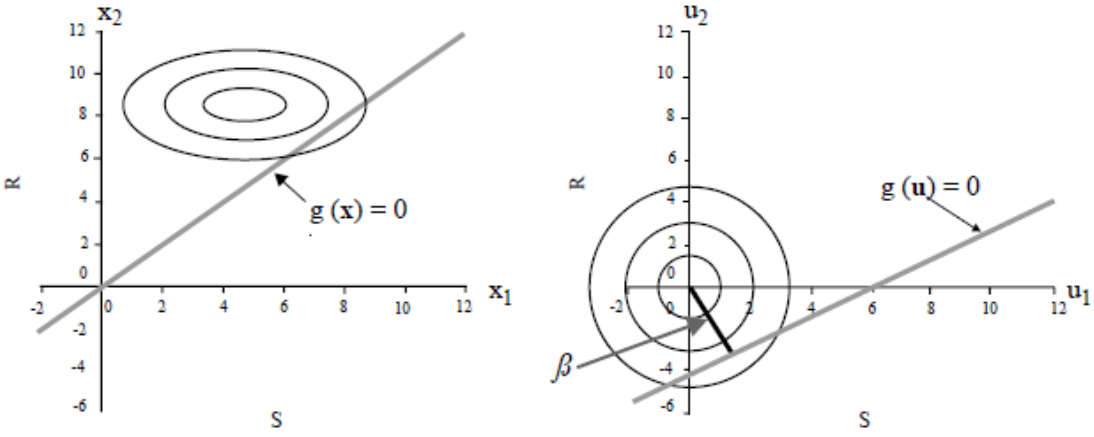


Figure 5.5: Illustration of the two-dimensional case of a linear limit state function and standardized normally distributed variables U

5.2 Crude Monte Carlo simulation

In reliability analysis of structures the Monte Carlo simulation (MC) method is particularly applicable when an analytical solution is not attainable and the failure domain cannot be expressed or approximated by an analytical form. This is mainly the case in problems of complex nature with a large number of basic variables where all other reliability analysis methods are not applicable. Despite the fact that the mathematical formulation of the MC is relatively simple and the method has the capability of handling practically every possible case regardless of its complexity, this approach has not

received an overwhelming acceptance due to the excessive computational effort that is required. Several sampling techniques, also called variance reduction techniques, have been developed in order to improve the computational efficiency of the method by reducing the statistical error that is inherent in MC methods and keeping the sample size to the minimum possible.

Monte Carlo simulation can be stated as follows in structural reliability problems. Expressing the limit state function as $g(x) \leq 0$ where $x = (x_1, x_2, ..., x_M)$ is the vector of the random variables, eq.(5.5) can be written as

$$P_f = \int_{g(x)\leq 0} f_x(x)dx \tag{5.10}$$

where $f_x(x)$ denotes the joint probability of failure for all random variables. Since MC approach is based on the theory of large numbers (N_∞) an unbiased estimator of the probability of failure is given by

$$P_f = \frac{1}{N_\infty} \sum_{j=1}^{N_\infty} I(x_j) \tag{5.11}$$

in which $I(x_j)$ is an indicator for successful and unsuccessful simulations defined as

$$I(x_j) = \begin{cases} 1, & if \quad g(x_j) \leq 0, \\ 0, & if \quad g(x_j) > 0. \end{cases} \tag{5.12}$$

In order to estimate P_f an adequate number of N independent random samples is produced using a specific, usually uniform, probability density function of the vector $\boldsymbol{x}$. The value of the failure function is computed for each random sample x_j and the Monte Carlo estimation of P_f is given in terms of sample mean by

$$P_f = \frac{N_s}{N} \tag{5.13}$$

where where N_s is the number of successful simulations and N the total number of simulations. However, simulations are often costly in computation time and the uncertainty of the estimate is thus of interest. It is easily realized that the coefficient of variation of the estimate is proportional to $1/\sqrt{(N_s)}$ meaning that if Monte Carlo simulation is pursued to estimate a probability in the order of 10^{-6} it must be expected that approximately 10^8 simulations are necessary to achieve an estimate with a coefficient of variance in the order of 10%. A large number of simulations are thus required using Monte Carlo simulation and all refinements of this crude technique have the purpose of reducing the variance of the estimate.

5.3 Latin Hypercube sampling (LHS)

Latin hypercube sampling (LHS) (87, 100, 111) is a simulation method which generates samples that are better distributed in the sample space compared to those obtained by pseudo-random generators. The advantage of the LHS approach is that in the case of uniform distribution field, the random samples are generated from all ranges of possible values, in a way that no sub-domain is over-sampled, while in the cases where non-uniform distributions are used LHS insures that the tails of the probability distributions are sampled. According to this approach, the range of each random variable is divided into N sets with equal probability mass, where N is the total number of samples. Then, one sample is generated from each of the N sets. The advantage of this method is that usually converges faster than the crude Monte Carlo simulation.

5.4 The Subset Simulation (SS)

The estimation of small failure probabilities P_f with the aid of MC requires an excessive number of simulations in order to capture rare events. The basic idea of subset sampling is the subdivision of the failure event $F = \{g(R - S) \leq 0\}$ into a sequence of M partial failure events (subsets) $F_1 \supset F_2 \supset ... \supset F_M = F$. The division into subsets (sub-problems) offers the possibility to transform the simulation of rare events into a set of simulations of more frequent events. The determination of the failure events F_i can be determined by presetting a series of limit values $g_i, \quad i = 1, ..., M$ such as (12)

$$F_i = \{\boldsymbol{x} : g(\boldsymbol{x}) \leq g_i\} \tag{5.14}$$

where $\boldsymbol{x}$ being the vector of independent and identically distributed (i.i.d) samples according to a probability density function pdf q, and $g(x)$ is the system performance function, usually a non-linear function of $\boldsymbol{x}$. This enables the computation of the failure probability as a product of conditional probabilities $P(F_{i+1}|F_i$ and $P(F_1)$ as follows:

$$P_F = P(F_M) = P(F_1) \cdot \prod_{i=1}^{M-1} P(F_{i+1}|F_i) \tag{5.15}$$

The determination of the failure events F_i and the partial conditional failure probabilities $P_i = P(F_{i+1}|F_i)$ strongly affects the accuracy of the simulation. Usually, the limit values $g_i|i = 1, .., M$ are selected in such way that nearly equal partial failure probabilities $P_i|i = 1, .., M$ are obtained for each subset. However, it is difficult to specify in advance the limit values g_i according to a prescribed probability P_i. Therefore the limit values have to be determined adaptively within the simulation, as a function of the prescribed (13).

5.4.1 Subset simulation algorithm

In the first step the probability P_1 is determined by direct Monte Carlo simulation:

$$P_1 = P(F_1) = \frac{1}{N} \cdot \sum_{k=1}^{N_i} I_{F_1}(x_k^{(1)}) \tag{5.16}$$

where $[x_k^{(1)} : k = 1, ..., N]$ are independent and identically distributed samples simulated according to pdf q and $I_{F_1}(x_k^{(1)})$ is the indicator

$$I_{F_1}(x_k^{(1)}) = \begin{cases} 0, & if \quad x_k^{(1)} \notin F_1, \\ 1, & if \quad x_k^{(1)} \in F_1. \end{cases} \tag{5.17}$$

To obtain the conditional probabilities $P(F_{i+1}|F_i)$ of eq.(5.15), the evaluation of the respective conditional probability density functions is required:

$$q(x|F_i) = \frac{I_{F_1}(x) \cdot q(x)}{P(F_i)} \tag{5.18}$$

Samples that follow the $q(x|F_i)$ conditional pdf can be generated numerically using a Markov Chain Monte Carlo procedure. Then the conditional probability at subset level $i + 1$ can be estimated as:

$$P(F_{i+1}|F_i) = P_{i+1} \approx \frac{1}{N_{i+1}} \cdot \sum_{k=1}^{N_{i+1}} I_{F_{i+1}}(x_k^{(i+1)}) \tag{5.19}$$

The samples of $i + 1$ subset are generated from the samples of subset i that are located in the failure region F_i as follows

$$x^i : g(x^i) > g_i \quad i = 1, 2, ..., M - 1 \tag{5.20}$$

The limit value g_i of the i^{th} partial subset is determined during the simulation from a list of sorted pairs $(x_k^{(i)}), \quad g(x_k^{(i)}), \quad i = 1, 2, .., N_i$ in ascending order, according to values of $g(x_k^{(i)})$. The limit value g_i corresponds to the value of the performance function of the j^{th} sorted sample given by

$$g_i = g(x_i^{(i)}), \quad j = P_i \cdot N_i \tag{5.21}$$

The number of samples N_i is selected in a way that partial probabilities P_i at each subset level are accurately estimated, in the context of MC. The last failure probability $P(F_M|F_{M-1})$ can be estimated with the following expression:

$$P(F_M|F_{M-1}) = P_M \approx \frac{1}{N_M} \cdot \sum_{k=1}^{N_M} I_{F_M}(x_k^{(M)}) \tag{5.22}$$

The total failure probability P_F may then be computed as

$$P_F = \prod_{i=1}^{M} P_i \tag{5.23}$$

5.5 Markov chain Monte Carlo simulation

Markov Chain Monte Carlo Simulation (MCMCS), in particular, the Metropolis method, is a powerful simulation technique for simulating samples according to an arbitrary pdf. It originates from the method developed by Metropolis and his co-workers for applications in statistical physics (104). The advantages of MCMCS for solving reliability analysis problems is that it succeeds in simulating samples compatible to a conditional probability distribution $q(\cdot|F)$, which has been the main challenge in simulation–based reliability analysis. The algorithm used here is a modified version of the original Metropolis algorithm (13).

5.5.1 Modified Metropolis algorithm

For every $j = 1, .., n$ let $p_j^*(y|x)$, called the 'proposal pdf' be one-dimensional PDF for y centered at x with the symmetry property $p_j^*(y|x) = p_j^*(x|y)$. Generate a sequence of samples $\{x_1, x_2, ...\}$ from a given sample x_1 by computing x_{k+1} from $x_k = [x_k(1), ...x_k(n)], \quad k = 1, 2, ...$ as follows:

1 Generate a 'candidate' state x^*: for each component $j = 1, ..., n$ simulate y_j from $p_j^*(\cdot|x_k(j))$.

Compute the ratio $r_j = \frac{q_j(y_j)}{q_j(x_k(j))}$

Set

$$x^*(j) = \begin{cases} y_j, & \text{with probability} \quad \min(1, r_j), \\ x_k(j), & \text{with probability} \quad 1 - \min(1, r_j) \end{cases} \tag{5.24}$$

2 Accept/reject x^*: Check the location of x^* If $x^* \in F_i$ accept it as the next sample, i.e. $x_{k+1} = x^*$. Otherwise reject it and take the current sample as the next sample i.e. $x_{k+1} = x_k$.

$$x_{k+1}(j) = \begin{cases} x_{k+1}^*(j, & \text{when} \quad x^* \in F_i, \\ x_k, & \text{when} \quad x^* \notin F_i. \end{cases} \tag{5.25}$$

Step 1 can be viewed as a ŚlocalŠ random walk in the neighborhood of the current state x_k , while Step 2 ensures that the next sample always lies in F_i, so as to produce the correct conditioning in the samples. The choice of the proposal pdfs affects the deviation of the candidate state from the current one and controls the efficiency of the Markov Chain samples.

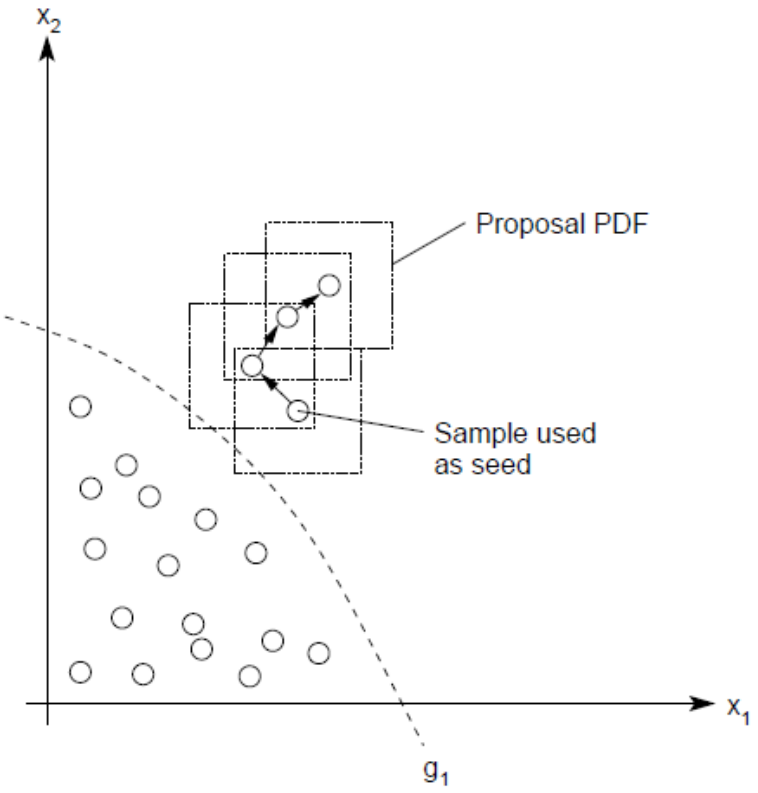


Figure 5.6: Schematic representation of generation of samples using Markov Chain Monte Carlo algorithm

Simulations show that the efficiency of the method is insensitive to the type of the proposal pdfs, and hence those which can be operated easily are the most preferable. For example, the uniform PDF centered at the current sample with width $2l_j$ is a good candidate for p_j^*. However, the spread of the proposal pdfs is of crucial importance since it affects the size of the region covered by the Markov chain samples, and consequently it controls the efficiency of the method. Very small spreads of the proposal pdfs tends to increase the dependence between successive samples due to their proximity, thus slowing down convergence of the estimator and occasionally causing ergodic-

ity problems. On the other hand, excessive large spreads may reduce the acceptance rate, by increasing the number of repeated Markov Chain samples and thus slowing down convergence. The optimal choice for the spread of the proposal pdfs depends on a trade-off between acceptance rate and correlation due to proximity. Usually, the spread is chosen to be a fraction of the standard deviation of the starting samples, since no a priori knowledge is available that would lead to a convenient choice of spread (12, 13).

The choice of the intermediate failure events also plays a key role in the SS robustness since it affects the values of the conditional level probabilities and hence the efficiency of the method. A commonly applied procedure is to choose a priori the intermediate threshold values g_i in eq.(5.14) in such way that nearly equal partial failure probabilities $P_i|i = 1, .., M$ are obtained for each subset. However, as it is often difficult to specify the limit values g_i in advance according to a prescribed partial probability P_i, these values are obtained during the SS procedure from a list of sorted pairs in ascending order so that the estimated conditional probabilities are equal to predefined values. This choice of the intermediate limit thresholds implies that the limit values g_i are no longer deterministic, since they depend on the conditional samples.

5.5.2 Conditional probability estimators

The coefficient of variation (c.o.v) of the probability of the first subset δ_1, defined as the ration of the standard deviation to the mean of P_1, is given by:

$$\delta_1 = \sqrt{\frac{1 - P_1}{P_1 N}} \tag{5.26}$$

Since the Markov chains generated at each conditional level are started with samples (selcted from the previous simulation level) distributed as the corresponding target conditional pdf, the Markov chain samples used for computing the conditional probability estimators are identically distributed as the target conditional pdf. Thus, the conditional estimators P_i's are unbiased. Although the Markov chains samples are dependent, the estimator's still have convergence properties of the estimators using independent samples.

An expression for the c.o.v for P_i is given next. At the $(i - 1)$th conditional level, suppose that the number of Markov chains is N_c and $\frac{N}{N_c}$ samples have been simulated from each of these chains, so that the total number of Markov chain samples is N. Although the samples generated by different chains are in general dependent, it is assumed for simplicity that they are uncorrelated through the indicator function $I_{F_i}(\cdot)$. The covariance sequence $R_i(k)$ $i = 0, ..., \frac{N}{N_c - 1}$ can be estimated using the Markov chain

samples at the $(i-1)$th conditional level by:

$$R_i(k) = \frac{1}{N - kN_c} \sum_{j=1}^{N_c} \sum_{l=1}^{\frac{N}{N_c}-k} I^i_{jl} I^i_{j,l+k} - P_i^2 \tag{5.27}$$

The correlation coefficient at lag k of the stationary sequence is given by $\rho_i(k) = \frac{R_i(k)}{R_i(0)}$. Finally, since I^i_{jk} is a Bernoulli random variable $R_i(0) = P_i(1-P_i)$ and so the variance of P_i is given by

$$\sigma_i^2 = \frac{P_i(1-P_i)}{N}[1+\gamma_i] \tag{5.28}$$

where

$$\gamma_i = \sum_{k=1}^{\frac{N}{N_c}-1} (1 - \frac{kN_c}{N})\rho_i(k) \tag{5.29}$$

The c.o.v δ_i of P_i is thus given by:

$$\delta_i = \sqrt{\frac{1-P_i}{P_i N}(1+\gamma_i)} \tag{5.30}$$

The value of γ_i depends on the choice of the spread of the proposal pdfs. The c.o.v of P_F is upper bounded and the upper bound corresponds to the case when the conditional probability estimators are fully correlated. The actual c.o.v depends on the correlation between the P_i's. If all the P_i's were uncorrelated, then

$$\delta^2 = \sum_{i=1}^{M} \delta_i^2 \tag{5.31}$$

Although the P_i's are generally correlated, simulations show that δ^2 may well be approximated by $\sum_{i=1}^{m} \delta_i^2$.

6

Accelerated subset simulation with neural networks for reliability analysis

In this chapter of the thesis a Neural Network-based SS (SS-ANN) methodology (115) is proposed in which ANN are effectively trained over smaller sub-domains of the total random variable space which are generated progressively at each SS level by the modified Metropolis algorithm. ANN are then used as robust meta-models in order to increase the efficiency of SS by increasing significantly the samples per SS level with a minimum additional computational effort. In the numerical examples considered, it is demonstrated that the training of a sufficiently accurate ANN meta-model in the context of SS simulation leads to more robust estimations of the probability of failure both in terms of mean and variance of the estimator.

In order to transfer information from the modified Metropolis algorithm to the ANN training process in order to train effectively the ANN in a sequence of moving ranges (windows), as these are produced by the upper and lower bounds of the random variables generated by Markov chains of the MCMCS at each successive SS level. The ANN is then used as a robust meta-model in order to increase the SS efficiency by significantly enriching the sample space at each SS level with a minimum additional computational cost. In order to achieve that, a large number, with respect to standard SS procedure, of N_{MC} samples is generated at each subsequent SS level using MCMC algorithm and the corresponding failure function values are evaluated with the trained ANN instead of the real numerical model.

Two alternative training strategies are implemented for the ANN training procedure. The first (SS-ANN1) introduces a closed multidimensional set Ω^M, subset of the total random variable space $\boldsymbol{\Theta}$($\Theta^M \subset \boldsymbol{\Theta}$) defined by the upper and lower bounds of the random variables as these are generated by the increased, with respect to standard SS, space of MCMCS samples at each SS level. Subsequently, the ANN is effectively trained over progressive and relatively small sets Ω^M instead of $\boldsymbol{\Theta}$, with the minimum required training points using a Latin Hypercube Sampling (LHS) training scheme.

The second strategy (SS-ANN1) introduces a width l around each initial seed that the MCMCS uses as center in order to generate a random walk according to the proposal pdf $q(x|F_i)$ of Eq.(5.18). This width defines a closed subset Ω^j around each seed j, defined by a lower and upper bound at distance l from the center. The ANN is effectively trained at each Ω^j with a minimum computational cost using the LHS. Obviously, the subset Ω^M in the case of SS-ANN2 is constructed by the union of all Ω^j at each SS level.

The basic difference between the proposed methodology and the classical ANN-based MCS is that the SS-ANN1 and SS-ANN2 training processes are more effective since in both cases the ANN is trained over smaller but sufficient progressive subsets of the total random variable space $\mathbf{\Theta}$. The ANN is used subsequently as a more accurate meta-model, compared to a ANN that is trained over the whole $\mathbf{\Theta}$, as in the case of classical ANN-MCS [5]. The trained ANN is then used to produce estimates of the failure function values required for the definition of the conditional parametric space $F_M|F_{M-1}$ over which the probability $P(F_M|F_{M-1})$ of eq.(5.22) is calculated at each SS level. Detailed pseudo-algorithms of the two ANN training alternatives are presented next.

6.1 ANN architecture

The Resilient Back Propagation (Rprop) algorithm (130) is used for the ANN training while also a modification to the error function of eq.(4.6) is performed by adding an extra regularizing term $\mathbf{E}_W(\boldsymbol{w})$. This term penalizes the large values of the weights in order to achieve a smoother mapping

$$\mathbf{E}_W(\boldsymbol{w}, \mathbf{A}) = \sum_i \sum_j \frac{1}{2} w_{ij}^2 \tag{6.1}$$

The $\mathbf{E}_W$ is called weight energy term, and the error function to be minimized becomes

$$E = \alpha \mathbf{E}_W(\boldsymbol{w}, \mathbf{A}) + \beta \mathbf{E}_D(\mathbf{D}|\boldsymbol{w}, \mathbf{A}) \tag{6.2}$$

the constant α is called is called regularizing constant. The two constants α and β are determined using the following two rules:

$$2\alpha \mathbf{E}_W = \mu \tag{6.3}$$

$$2\beta \mathbf{E}_D = N - \mu \tag{6.4}$$

with

$$\mu = \sum_{i}^{k} \frac{\lambda_i}{\lambda_i + i} \tag{6.5}$$

where λ_i are the eigenvalues of the quadratic form $\beta \mathbf{E}_D$, N is the number of output units times the number of the training pairs and k is the total number of the weight parameters.

6.2 First ANN training approach (SS-ANN1)

Step 1: At the M^{th} SS level, generate a large number of N_{MC} samples for each random $x_i^{(M)}$ variable satisfying failure criterion F_{M-1}, using the MetropolisŰHastings algorithm as follows:

$$x^{(M)} = \begin{bmatrix} x_1^{(M)} \\ x_2^{(M)} \\ \vdots \\ x_n^{(M)} \end{bmatrix} \rightarrow MCMC \rightarrow \begin{bmatrix} x_{1j}^{(M)} \\ x_{2j}^{(M)} \\ \vdots \\ x_{nj}^{(M)} \end{bmatrix}, \quad j = 1, .., N_{MC} \tag{6.6}$$

where $x_{ij}^{(M)} \quad i = 1, ..., n$ are the points of the MCMC random walk around $x_i^{(M)}$, n being the number of random variables.

Step 2: Define the limits of subset Ω^M by determining its lower and upper bounds. These limits define the window in which ANN is progressively trained at each SS level.

$$x_{lower}^{(M)} = \begin{bmatrix} x_{lower,1}^{(M)} \\ x_{lower,2}^{(M)} \\ \vdots \\ x_{lower,n}^{(M)} \end{bmatrix} = \left\{ \min \begin{bmatrix} x_{1j}^{(M)} \\ x_{2j}^{(M)} \\ \vdots \\ x_{nj}^{(M)} \end{bmatrix}, \quad j = 1, .., N_{MC} \right\} \tag{6.7}$$

$$x_{upper}^{(M)} = \begin{bmatrix} x_{upper,1}^{(M)} \\ x_{upper,2}^{(M)} \\ \vdots \\ x_{lower,n}^{(M)} \end{bmatrix} = \left\{ \max \begin{bmatrix} x_{1j}^{(M)} \\ x_{2j}^{(M)} \\ \vdots \\ x_{nj}^{(M)} \end{bmatrix}, \quad j = 1, .., N_{MC} \right\} \tag{6.8}$$

Step 3: Train the Neural Network within the multidimensional space Ω_M using a LHS scheme in the following range:

$$[x_{lower}^{(M)}, x_{upper}^{(M)}] = [x_{lower,i}^{(M)}, x_{upper,i}^{(M)}] \quad i = 1, ..., n \tag{6.9}$$

In case that the successive training windows overlap, ANN is trained over the remaining non-overlapping domain, which is defined as follows:

$$\Omega_M - (\Omega_M \cap \Omega_{(M-1)}) \tag{6.10}$$

A schematic representation of the aforementioned procedure for the progressive ANN training windows in two subsequent subset levels is presented in Fig.6.1 for an ideal case of two random variables. The number of points used for the ANN training at the M^{th} SS level spans the dashed area in Fig.6.1. This number is equal to the number of the required analyses of the detailed model (or function evaluations) and hence the computational effort involved at each SS level.

Step 4: Estimate the analysis results (function evaluations) at each one of the N_{MC} sample points of Ω_M using the trained ANN with practically zero additional computational cost.

$$X_j^{(M)} = \begin{bmatrix} x_{1j}^{(M)} \\ x_{2j}^{(M)} \\ \vdots \\ x_{nj}^{(M)} \end{bmatrix} \longrightarrow \text{ANN} \longrightarrow y^M = Y_j^{(M)}(x_j^{(M)}) \quad j = 1, ..., N_{MC} \tag{6.11}$$

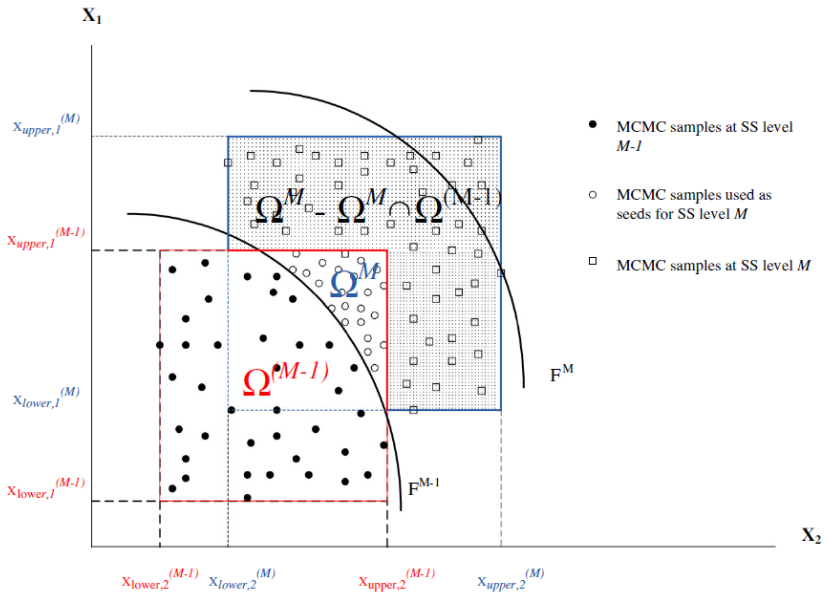


Figure 6.1: Schematic representation of progressive ANN training windows in two subsequent SS levels.

Step 5: Estimate the failure criterion F_M according to and return to Step 1 for the subsequent SS Level. This is done by specifying intermediate target probabilities $P(F_M|F_{M-1})$ at each SS level (usually set at 10^{-1}) and identifying the failure event F_M by sorting x_j and y^M according to the magnitude of the elements of y^M.

6.2.1 LHS based training

The appropriate selection of the input training data is an important factor for a successful training since the training set must include data over the entire range of the output space. Although the number of training patterns plays its own role in the accuracy of the predictions, the distribution of samples is of greater importance. The selection of the I/O training pairs is based on the requirement that the full range of possible results should be represented in the training procedure.

In the present study the sample space for each random variable is divided into equally spaced distances for the application of the ANN simulation and for the selection of the suitable training pairs. The range for the random vector $\boldsymbol{x}$ is defined as follows:

$$\left.\begin{aligned} x^{(M)}_{lower,i} &= \text{minimum}(x^{(M)}_{ij}) \\ x^{(M)}_{upper,i} &= \text{maximum}(x^{(M)}_{ij}) \end{aligned}\right|_{\substack{i=1,\ldots,n \\ j=1,..,N_{MC}}}$$

where n is the number of random variables, N_{MC} is the total number of MCMC samples generated at M^{th} SS level. The samples used at each SS level for the ANN training are generated with the Latin Hypercube Sampling method within the range $[x_{lower,i}^{(M)}, x_{upper,i}^{(M)}$ for each random variable. As mentioned previously, ANN is trained over the non-overlapping domains of the windows in two subsequent SS levels. A schematic representation of the LHS training scheme for an ideal case of two random variables and three subsequent SS levels is shown in Fig. 6.2.

The convergence of the training process is controlled by the prediction error. This is done either with a direct comparison of the predicted results with the target values computed by the conventional numerical procedure denoted as ŚŚexactŠŠ, or by means of the root mean square error e_{RMS}, which gives a measure of the difference between predicted at each ANN cycle and ŚŚexactŠŠ values.

$$e_{RMS}(\%) = \frac{|Out_{exact} - Out_{NN}|}{Out_{exact}} \cdot 100 \tag{6.12}$$

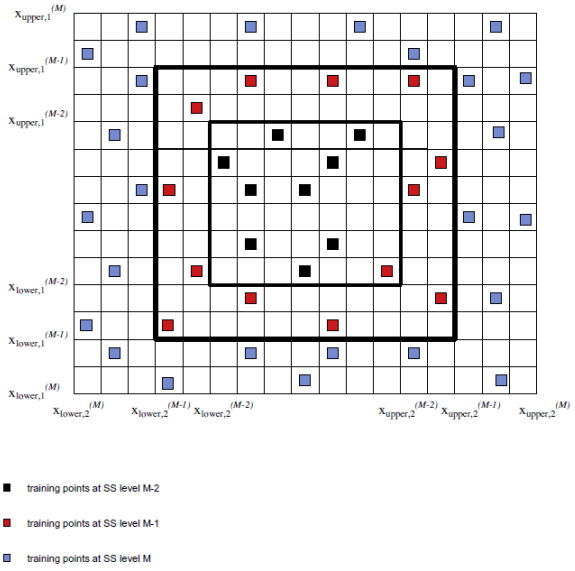


Figure 6.2: LHS training scheme for two random variables and three subsequent SS levels.

6.3 Second ANN training approach (SS-ANN2)

The pseudo-algorithm of the second ANN training strategy is as follows:

Step 1: At the first SS level, generate N_{MC} function evaluations, sort the samples in ascending order and determine the failure criterion for the first subset level and the samples which will be used as seeds for the next subset level. The number of the

samples will be $P_0 \cdot N_{MC}$, where P_0 is the target probability of the intermediate failure events usually taken as 0.1.

Step 2: For every seed $x^j_{center} = (x^j_{center,1}, ..., x^j_{center,n})$ satisfying the first failure criterion ($> F_1$), train a ANN within subset $\Omega^{(j)} \subset \Omega^M \subset \Theta$ defined as:

$$\Omega^{(j)} = [x^{(j)}_{lower}, x^{(j)}_{upper}] = [x^{(j)}_{center} - l, x^{(j)}_{center} + l] \tag{6.13}$$

where l is a width that defines Ω^j. Subset Ω^M in this case is given by the following union:

$$\Omega^M = \bigcap_{j}^{P_0 \cdot N} \Omega^j \tag{6.14}$$

The total number of the ANN training points in this step will be

$$N_{train} = T \cdot P_0 \cdot N_{MC} \tag{6.15}$$

where T is the number of points generated by LHS. A schematic representation of steps 1 and 2 is shown in Fig.6.3.

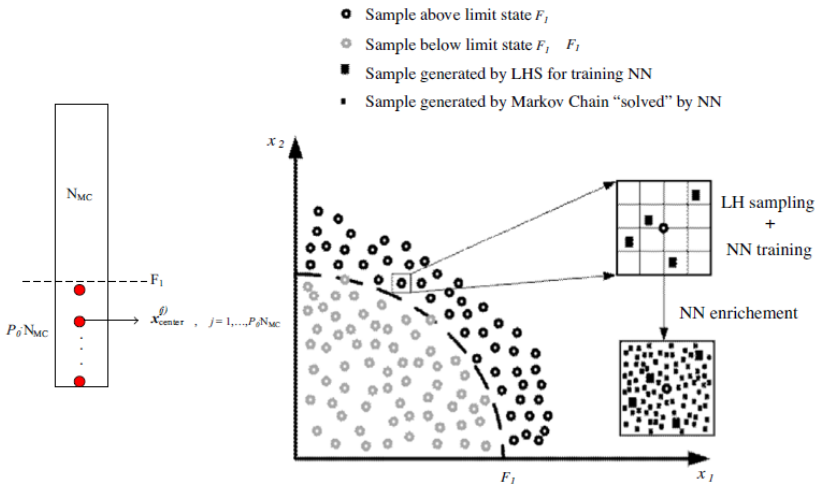


Figure 6.3: Schematic representation of steps 1 and 2 for the second NN training approach.

Step 3: Use the trained ANN to perform N_{NN} ($N_{NN} \geq N_{train}$) function evaluations in the context of SS procedure and modified Metropolis algorithm. Estimate the failure

criterion of the second subset and obtain the samples $P_0 \cdot N_{NN}$ used as seeds for the next SS level.

Step 4: Among the $P_0 \cdot N_{NN}$ samples select only N_{train} which will be used for the ANN training. The selection is made stepwise in order to achieve the best representation of the parametric space (Fig.6.4). In order to train the ANN accurately and effectively, replace the values of the performance function for the N_{train} points with a real function evaluations.

Step 5: Train the ANN and generate N_{NN} MCMC samples.

Step 6: Repeat steps 4 and 5 until the target failure criterion is achieved.

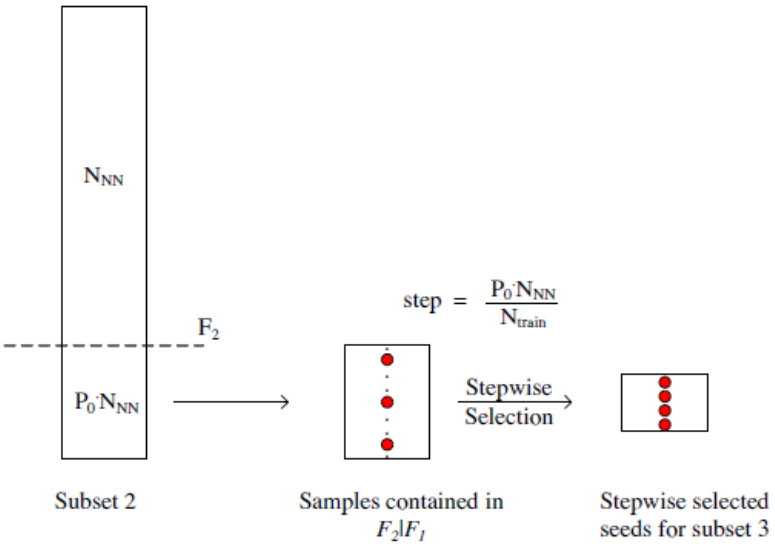


Figure 6.4: Schematic representation of step 4.

6.3.1 Choice of witdh for the ANN training in SS-ANN2

It must be mentioned here that the selection of an appropriate width l for the ANN training in SS-ANN2, is of crucial importance since it affects the ability of the ANN to accurately predict the values of the failure function within Ω^M. This is due to the fact that the ANN is not trained over the total subspace that covers the random walk around each seed of the MCMCS, as in the case of Ω^M SS-ANN1. Therefore, the accuracy of SS-ANN2 relies on the aliasing of the various Ω^j so that samples of the random walk around one seed that violate the limits of eq.(6.13) and eq.(6.14), i.e. plus and minus the width of the proposal pdf, are expected to fall in to the training window of

an adjacent seed. The selection of an appropriate width is not a priori known but it seems that a convenient choice is $l = 2$.

6.4 Numerical examples

In order to assess the performances of the SS-ANN methods, two test examples are considered, one based on mathematical models and one based on a structural analysis problem. For the mathematical models, all random variables $X_i,\ i = 1, ..., n$ are assumed to follow the Normal distribution with mean value $\mu = 0$ and variance $\sigma^2 = 1$ ($N \sim (0, 1)$). The partial failure probabilities are predefined to $P_i = P_0 = 0.1$ for each subset. For this example a number of $n = 3, 10$ and 100 random variables have been used in order to test the efficiency of the proposed method and its sensitivity to the dimensionality of the problem. The target probability of failure is taken $P_F = 10^{-4}$.

6.4.1 Mathematical model 1

The proposed method is first implemented in a simple linear analytic performance function, given by the following expression:

$$Y_n(x) = \sqrt{\sum_{j=1}^{n} X_j^2} \quad n = 3, 10, 100 \tag{6.16}$$

A brute force Monte Carlo simulation with 10^6 samples is initially performed in order to calculate the reference failure criterion g_n which corresponds to the target probability of failure (10^{-4}), for the cases of $n = 3, 10$ and 100 random variables, respectively. The results for g_n are 4.60, 5.98 and 12.70, respectively.

6.4.1.1 Subset Simulation

A standard SS is performed with 1000 and 10000 samples per subset level. The spread of the uniform proposal pdf is taken $l = 2$. Table 6.1 depicts the mean value and the coefficient of variation (c.o.v) of the estimated probability of failure according to eq.(5.31) which corresponds to the reference failure criterions g_n obtained with brute force MC. From this Table it can be seen that the efficiency of the SS is highly depended on the number of the samples generated in each Subset and less in the dimensionality of the problem.

The c.o.v for the estimated probability significantly reduces when increasing the number of analyses and reaches a $10-15\%$ value only when 10000 samples per subset are used. These results tend to improve when higher dimensionality problems are investigated (10 and 100 random variables). The bias of SS in the prediction of the

mean P_F, is 1.1%, 4.1% and 1.1% for the 1000 function evaluations per subset level and 3, 10 and 100 random variables, respectively, while the same results for 10000 function evaluations per SS level are 0.4%, 0.6% and 0.2%.

Subset Simulation (10 simulation runs)			
Failure function:		$Y_n = \sqrt{\sum_{j=1}^{n} X_j^2}$	
n	**Function evaluations/Subset**	P_F	**c.o.v**
3	1000	9.89×10^{-5}	0.40
	10000	9.96×10^{-5}	0.15
10	1000	9.59×10^{-5}	0.34
	10000	9.94×10^{-5}	0.15
100	1000	9.89×10^{-5}	0.29
	10000	9.98×10^{-5}	0.09

Table 6.1: Estimation of failure probability P_F with SS

6.4.1.2 Proposed SS-ANN methodologies

The same problem is now solved with the proposed SS-ANN approaches using the procedures described in sections 6.1 and 6.2. Tables 6.2 – 6.4 present the results for the two approaches for the cases of $n = 3$, 10 and 100 random variables, respectively.

Failure function: $Y_n = \sqrt{\sum_{j=1}^{n} X_j^2}$								
			SS		**SS-ANN1**		**SS-ANN2**	
n	**Function evaluations/subset**	**ANN-MCMC/subset**	P_F	**c.o.v**	P_F	**c.o.v**	P_F	**c.o.v**
3	1000	10^4	9.89×10^{-5}	0.40	9.86×10^{-5}	0.20	1.09×10^{-4}	0.14
		10^5			9.94×10^{-5}	0.18	1.01×10^{-4}	0.07
		10^6			1.01×10^{-4}	0.11	9.98×10^{-5}	0.06

Table 6.2: SS-ANN estimation of failure probability for the case of 3 random variables.

Failure function: $Y_n = \sqrt{\sum_{j=1}^{n} X_j^2}$								
			SS		**SS-ANN1**		**SS-ANN2**	
n	**Function evaluations/subset**	**ANN-MCMC/subset**	P_F	**c.o.v**	P_F	**c.o.v**	P_F	**c.o.v**
10	1000	10^4	9.59×10^{-5}	0.34	9.84×10^{-5}	0.24	9.93×10^{-5}	0.19
		10^5			1.07×10^{-4}	0.15	1.01×10^{-4}	0.12
		10^6			1.01×10^{-4}	0.12	9.98×10^{-5}	0.09

Table 6.3: SS-ANN estimation of failure probability for the case of 10 random variables.

Failure function: $Y_n = \sqrt{\sum_{j=1}^{n} X_j^2}$								
			SS		**SS-ANN1**		**SS-ANN2**	
n	**Function evaluations/subset**	**ANN-MCMC/subset**	P_F	**c.o.v**	P_F	**c.o.v**	P_F	**c.o.v**
100	1000	10^4	9.89×10^{-5}	0.29	9.85×10^{-5}	0.23	9.81×10^{-5}	0.21
		10^5			9.91×10^{-5}	0.16	9.94×10^{-5}	0.13
		10^6			1.05×10^{-4}	0.12	9.93×10^{-5}	0.12

Table 6.4: SS-ANN estimation of failure probability for the case of 100 random variables.

From these Tables it can be seen that the performance of the two ANN training alternatives is almost equivalent in terms of the c.o.v of their estimations. For SS-ANN1, the bias for the cases of 3, 10 and 100 variables in the mean P_F is 1%, 1% and 5%, respectively, while SS-ANN2 produces improved predictions with a corresponding bias of 0.2%, 0.2% and 0.7%. In all cases examined, very good predictions of the mean P_F with a relatively small c.o.v (10-Ű20%) where achieved with the proposed approach using 1000 training points for the ANN (function evaluations) and 10^6 MCMC samples per SS level, which is an order of magnitude less computational effort with respect to the 10000 number of function evaluations per subset required by standard SS in order to reach the same accuracy.

6.4.2 Mathematical model 2

The proposed method is also implemented in a non-linear analytic performance function, given by the following expression:

$$Y_n(x) = \sum_{i=1}^{n} X_i^5 - 5 \sum_{i=1}^{n} X_i + 12 \quad n = 3, 10, 100 \tag{6.17}$$

Again, a brute force Monte carlo was performed in order to obtain the reference failure criterion for the three cases. The results for g are 1.217, 1.664 and 2.012 respectively. The results of the standard SS and of the proposed methodologies are presented in the next Tables.

Subset Simulation (10 simulation runs)			
Failure function:		$Y = \sum_{i=1}^{n} X_i^5 - 5\sum_{i=1}^{n} X_i + 12$	
n	**Function evaluations/subset**	P_F	**c.o.v**
3	1000	9.75×10^{-5}	0.42
	10000	9.93×10^{-5}	0.17
10	1000	9.82×10^{-5}	0.32
	10000	9.96×10^{-5}	0.15
100	1000	9.89×10^{-5}	0.29
	10000	1.02×10^{-4}	0.11

Table 6.5: SS estimation for 3, 10 and 100 random variables for the 2nd mathematical function

From Table 6.5 it can be seen that the bias of SS in the prediction of the mean P_F, is 2.5%, 2% and 1.1% for the 1000 function evaluations per subset levels and 3, 10 and 100 random variables, respectively, while the same results for 10000 function evaluations per SS level are 0.7%, 0.4% and 2%.

Failure function: $Y = \sum_{i=1}^{n} X_i^5 - 5\sum_{i=1}^{n} X_i + 12$								
			SS		**SS-ANN1**		**SS-ANN2**	
n	**Function evaluations/subset**	**ANN-MCMC/subset**	P_F	**c.o.v**	P_F	**c.o.v**	P_F	**c.o.v**
3	1000	10^4	9.75×10^{-5}	0.42	9.85×10^{-5}	0.21	1.09×10^{-4}	0.17
		10^5			9.90×10^{-5}	0.19	1.04×10^{-4}	0.12
		10^6			9.89×10^{-5}	0.13	9.92×10^{-5}	0.08

Table 6.6: SS-ANN estimation of failure probability for the case of 3 random variables.

Failure function: $Y = \sum_{i=1}^{n} X_i^5 - 5\sum_{i=1}^{n} X_i + 12$								
			SS		**SS-ANN1**		**SS-ANN2**	
n	**Function evaluations/subset**	**ANN-MCMC/subset**	P_F	**c.o.v**	P_F	**c.o.v**	P_F	**c.o.v**
10	1000	10^4	9.82×10^{-5}	0.32	1.08×10^{-4}	0.25	9.89×10^{-5}	0.18
		10^5			9.88×10^{-5}	0.15	9.92×10^{-5}	0.14
		10^6			1.03×10^{-4}	0.10	9.96×10^{-5}	0.09

Table 6.7: SS-ANN estimation of failure probability for the case of 10 random variables.

Failure function: $Y = \sum_{i=1}^{n} X_i^5 - 5\sum_{i=1}^{n} X_i + 12$								
			SS		**SS-ANN1**		**SS-ANN2**	
n	**Function evaluations/subset**	**ANN-MCMC/subset**	P_F	**c.o.v**	P_F	**c.o.v**	P_F	**c.o.v**
100	1000	10^4	9.89×10^{-5}	0.28	9.75×10^{-5}	0.22	9.81×10^{-5}	0.21
		10^5/subset			9.88×10^{-5}	0.15	9.90×10^{-5}	0.13
		10^6			1.06×10^{-4}	0.12	9.95×10^{-5}	0.11

Table 6.8: SS-ANN estimation of failure probability for the case of 100 random variables.

From Tables 6.6-6.8 it can be concluded that again the performance of the two ANN training alternatives is almost equivalent in terms of the c.o.v of their estimations.

For SS-ANN1, the bias for the cases of 3, 10 and 100 variables in the mean P_F is 1%, 3% and 6%, respectively, while SS-ANN2 produces improved predictions with a corresponding bias of 0.8%, 0.4% and 0.5%. Thus, in all cases examined, very good predictions of the mean P_F with a relatively small c.o.v $(10-20\%)$ where achieved with the proposed approach using 1000 training points and 10^6 ANN-MCMC samples per SS level, which is an order of magnitude less computational effort with respect to the 10000 number of function evaluations per subset required by standard SS in order to reach the same accuracy.

6.4.3 6-storey reinforced concrete building

The third test example considered in this study is the evaluation of the failure probability of a 6-storey reinforced concrete building of Fig.6.5. The base nodes are fixed while the slabs are considered to act as diaphragms. The building is analyzed with a Non-linear Static Pushover (NSP) analysis. A lateral load distribution corresponding to the fundamental mode is implemented. The building has been designed to meet the Eurocode requirements of EC8 and EC2 design codes. Concrete of class 16/20 (nominal cylindrical strength of 16 MPa) and class $S500$ steel (nominal yield stress of 500 MPa) is assumed. The base shear is obtained from a response spectrum of soil type B characteristic periods: ($T_B = 0.15\ s$, $T_C = 0.50\ s$, $T_D = 2.00\ s$) while the Peak Ground Acceleration (PGA) considered is equal to $0.31g$. Moreover, the importance factor γ_I was taken equal to 1.0, while the damping correction factor is equal to 1.0, since a damping ratio of 5% has been considered. The slab thickness is equal to 15 cm and is considered to contribute to the moment of inertia of the beams with an effective flange width. In addition to the self weight of beams and slabs, a distributed permanent vertical load of 2 KN/m^2 due to floor finishing partitions and an imposed load with nominal value of 1.5 KN/m^2, is considered. Following EC8 for the seismic loading combination, dead loads are considered with their nominal values, while live loads with 30% of their nominal value.

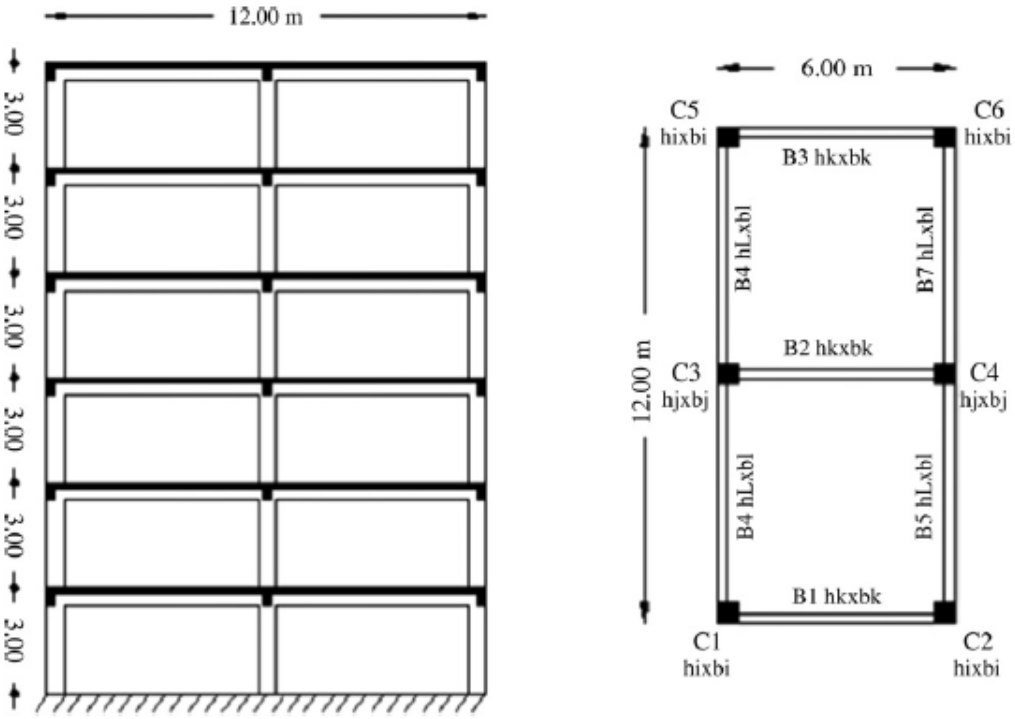


Figure 6.5: Views of the 6-storey RC building.

Beam-column members are modeled with the inelastic force-based fibber element. Three random variables were introduced, namely the concrete compression strength f_c, the steel tensile strength f_s and the steel Young modulus E_s which were considered to differ in each storey, resulting a total number of $n = 18$ random variables. All random variables were assumed to follow the Gaussian distribution with parameters presented in Table 6.9.

Random variables	Mean value	std
E_s	220 GPa	11 GPa
f_s	500000 KPa	25000 KPa
f_c	20000 kPa	3000 KPa

Table 6.9: Random variable parameters of the Gaussian distribution, for the 6-storey RC building.

For this case, the target probability of failure P_F is set at 10^{-3} and the reference value for the failure criterion is obtained via crude MC with 10^5 samples ($g = 1.7219$ m). As long as the network architecture is concerned, 1 hidden layer with 10 nodes (hidden layer size = 10) is used while the Rprop algorithm is implemented for the training. Both the approaches of the proposed methodology are studied.

The same problem is solved using the proposed SS-ANN method. The number of samples generated by the ANN are 10^4, 10^5 and 10^6, in order to enrich the sampling space at each subsequent SS step, while 1000 training points are used in each subset in order to train the ANN with a total of 4000 NSP analyses. Table 6.10 depicts the mean and c.o.v of P_F obtained for the SS and the two ANN training approaches. Again SS-ANN2 produces better predictions, with respect to SS-ANN1, on the mean P_F, with a bias of 2% instead of 3% and a c.o.v of 0.08 instead of 0.13 estimated through SS-ANN1. The corresponding c.o.v of the SS is 0.38.

Non-linear static pushover analysis (NSP)								
			SS		**SS-ANN1**		**SS-ANN2**	
n	**NPS/subset**	**ANN-MCMC/subset**	P_F	**c.o.v**	P_F	**c.o.v**	P_F	**c.o.v**
18	1000	10^4	9.86×10^{-4}	0.38	9.81×10^{-4}	0.22	9.91×10^{-4}	0.19
		10^5			9.94×10^{-4}	0.19	1.03×10^{-3}	0.11
		10^6			1.03×10^{-3}	0.13	1.02×10^{-3}	0.08

Table 6.10: SS-ANN estimation of the failure probability P_F for the 6-storey RC building.

6.4.4 SS-ANN versus ANN-MCS

The results of the SS-ANN2 procedure, considered to be more accurate, are further compared against results obtained via classical ANN-based brute-force Monte Carlo Simulation (ANN-MC) as this was proposed in. In order to perform a fair comparison between the two approaches, the total number of required detailed numerical analyses (or function evaluations) is kept the same for both methodologies. In the case of ANN-MC the training set of the random variable space is obtained with the previously described LHS training scheme in the range $[-5\sigma, +5\sigma]$, σ being the standard deviation of the random variables. The brute-force MC was performed for 10^6 samples. The results are obtained for the cases of 3 and 100 random variables and presented in Tables 6.11 and 6.12, respectively for the mathematical models of examples 1 and 2. At this point it is reminded that in all cases, considered the target ŚexactŠ probability of failure is set at $p = 10^{-4}$. The c.o.v for the Monte Carlo Simulation is

$$\delta = \sqrt{\frac{(1-p)}{N \cdot p}} \tag{6.18}$$

Failure function: $Y_n = \sqrt{\sum_{i=1}^{n} X_i^2}$					
		ANN-MC		**SS-ANN2**	
n	**Function evaluations**	P_F	**c.o.v**	P_F	**c.o.v**
3	4000	8.98×10^{-5}	0.10	9.98×10^{-5}	0.06
Failure function :$Y = \sum_{i=1}^{n} X_i^5 - 5 \cdot \sum_{i=1}^{n} X_i + 12$					
		ANN-MC		**SS-ANN2**	
n	**Function evaluations**	P_F	**c.o.v**	P_F	**c.o.v**
100	4000	8.72×10^{-5}	0.10	9.93×10^{-5}	0.08

Table 6.11: SS-ANN vs ANN-MC estimation of failure probability for the case of 3 random variables.

Failure function:$Y = \sqrt{\sum_{i=1}^{n} X_i^2}$					
		ANN-MC		**SS-ANN2**	
n	**Function evaluations**	P_F	**c.o.v**	P_F	**c.o.v**
3	4000	8.98×10^{-5}	0.10	1.005×10^{-5}	0.12
Failure function:$Y = \sum_{i=1}^{n} X_i^5 - 5 \sum_{i=1}^{n} X_i + 12$					
		ANN-MC		**SS-ANN2**	
n	**Function evaluations**	P_F	**c.o.v**	P_F	**c.o.v**
100	4000	8.72×10^{-5}	0.10	9.95×10^{-5}	0.11

Table 6.12: SS-ANN vs ANN-MC estimation of failure probability for the case of 100 random variables.

From these Tables it can be seen that SS-ANN2 produces significantly improved predictions of the mean P_F, with a bias of less than 1% in all cases considered. The corresponding ANN-MCS bias of the estimation is more than 10%. Thus, the proposed methodology leads to a more than one order of magnitude reduction in the bias of the P_F estimation.

6.5 Conclusions

A methodology is proposed for accurate and efficient system reliability analysis using a Neural Network-based Subset Simulation approach. This methodology takes advantage of the special characteristics of the SS in order to exploit the capability of a ANN to efficiently approximate limit state structural performance, provided that it is trained over a properly selected domain. The basic idea is to progressively train the ANN at each SS level in a sequence of moving windows defined by the upper and lower bounds of the samples generated by the Markov Chains of the modified Metropolis algorithm.

This way the ANN are effectively trained and used subsequently as robust meta-models in order to improve the efficiency of SS. This is accomplished by significantly increasing the samples at each SS level with a minimum additional computational effort. It is shown that the proposed approach improves the efficiency of classical SS reaching the same accuracy, both in terms of mean and c.o.v of the estimation, with one order of magnitude less computational effort. In addition it leads to more robust predictions, with respect to classical ANN-based brute force Monte Carlo simulation, with more than one order of magnitude less error in the estimation of the mean probability of failure.

6. ACCELERATED SUBSET SIMULATION WITH NEURAL NETWORKS FOR RELIABILITY ANALYSIS

7

Stochastic finite element method (SFEM)

7.1 Introductions

The most important issue in stochastic mechanics is the propagation of uncertainty through the system and the assessment of its stochastic response and is mainly addressed today in the framework of the stochastic finite element method (SFEM). SFEM is an extension of the classical deterministic approach for the solution of stochastic (static and dynamic) problems and has received considerable attention especially in the last two decades, due to the technological advances in the available computational power. SFEM involves finite elements whose properties are random. From a mathematical point of view, SFEM is a powerful tool for the solution of stochastic partial differential equations (PDEs) and it has been treated as such in numerous publications where convergence and error estimation issues are examined in detail. All the methods found in the literature under the denomination SFEM have the following common characteristics :

-a finite element model, i.e. the discretized version of the equations governing a physical phenomenon such as solid mechanics, heat transfer, etc.

-a probabilistic model of the input parameters : random variables and/or random fields are introduced for this purpose

Apart from these common points, the methods referred to as stochastic finite element analysis are rather different in nature. According to (144, 146), they may be classified as follows (48, 83, 98, 134):

-**Second moment methods**: these methods essentially aim at computing the variations of the mechanical response around its mean value, i.e. they provide

the mean and standard deviation of response quantities such as displacements or stresses. The perturbation method applied in (72, 84, 91, 92), falls within this category. So does the weighted integral method proposed in (37, 41, 148, 149).

-**Reliability methods**: these methods aim at computing the probability of failure of a mechanical system with respect to a failure criterion represented by a limit state function (43, 44).

-**Polynomial chaos expansion methods**(PCEM) : these methods aim at representing the full probabilistic content of the mechanical response as a polynomial series expansion in standard normal variables. In this respect, PCEM provide an intrinsic representation of the response, since each response quantity is characterized as a random variable through expansion coefficients. The spectral stochastic finite element method (SSFEM) proposed by Ghanem and Spanos (55) pertains to this category. This representation can be used together with Monte Carlo simulation to obtain the probability density function (PDF) of response quantities or second moment information. The use of SSFEM for finite element reliability analysis has also been demonstrated in (147).

The first step in the analysis of uncertain systems is the representation of the input of the system. This input usually consists of the mechanical and geometric properties as well as of the loading of the system (left and right hand side of the equilibrium equation, respectively). Characteristic examples are the Young modulus, Poisson ratio, yield stress, cross section geometry of physical systems, material and geometric imperfections of shells, earthquake loading, wind loads, waves etc. In the framework of SFEM The equilibrium equations for a finite element system are written as follows:

$$\boldsymbol{K}(\theta)\boldsymbol{U}(\theta) = \boldsymbol{F} \tag{7.1}$$

where the stochastic stiffness matrix $\mathbf{K}(\theta)$ is linearly dependent on a random process (or field) $X(\mathbf{t}, \theta) : D \times \Omega \rightarrow \mathbb{R}$. This relation is expressed using the following operator:

$$\mathbf{K}(\theta) = \boldsymbol{K}(X(\mathbf{t}, \theta)) \tag{7.2}$$

Despite the fact that most of the uncertain quantities appearing in engineering systems are non-Gaussian in nature, the Gaussian assumption is often used due to its simplicity and the lack of relevant experimental data. Furthermore, Gaussian random fields occur naturally in applications as a result of the central limit theorem and are the model of maximum entropy when only information on the second-order moments is available. The most common methods used in literature for the representation of Gaussian random fields $X(\mathbf{t}, \theta)$ are the spectral representation method (139) and the Karhunen-Loève (KL) decomposition (64, 65, 73, 93**?**).

The SFEM equations of the static problem become:

$$\boldsymbol{K}(X(\mathbf{t},\vartheta))\mathbf{U}(\theta) = \mathbf{F} \tag{7.3}$$

7.2 Spectral representation-based MC

The classical model for Monte Carlo simulation is based on a discrete approximation of the spectral representation of the random process $X(\mathbf{t},\theta)$ and it made in 2 steps.

Step A. **Realization the random field**
The realization involves three steps

1. First, a cut-off frequency $\overline{\omega} \in (0,\infty)$ needs to be selected such that most of the variance of the process is retained.

2. Second, the integration interval $(0,\overline{\omega})$ can be descritized in n equal intervals of length $\Delta\omega > 0$ with mid-points $\omega_q = (q - \frac{1}{2})\Delta\omega, \quad q = 1,2,...,n.$

3. Third, the classical model approximating $X(\mathbf{t})$ is

$$a_n(\mathbf{t}) = \sum_{k=1}^{n} \sigma_k [A_k \cos(\omega_k \mathbf{x}) + B_k \sin(\omega_k \mathbf{t})], \tag{7.4}$$

   where

$$\sigma_k^2 = \int_{\omega_k - \Delta\omega/2}^{\omega_k + \Delta\omega/2} g(\omega) d\omega \tag{7.5}$$

   and $\{A_k, B_k\}$ are uncorrelated random variables with mean 0 and variance 1.

Step B. **Solve the stochastic equation**
The stiffness matrix $\boldsymbol{K}(X(\mathbf{t},\vartheta))$ is calculated for every realization of the random field and the stochastic equilibrium is solved in order to find the statistics (moments, pdf) of the response $\boldsymbol{U}(\theta)$.

7.3 Karhunen-Loève-based MC

The random process $X(\mathbf{t}, \theta)$ is discretized according to eq.(2.60). The stiffness matrix is decomposed accordingly:

$$\mathbf{K}(\theta) = \boldsymbol{K}(X(\mathbf{t}, \theta)) \approx \mathbf{K}_0 + \sum_{j=1}^{M} \mathbf{K}_j \xi_j(\theta) \tag{7.6}$$

$\mathbf{K}_0$ is the stiffness matrix of the mean system. The deterministic matrices $\mathbf{K}_j$ are given by the operator:

$$\mathbf{K}_j = \begin{cases} \boldsymbol{K}(m(\mathbf{t})) & j = 0 \\ \boldsymbol{K}(\sqrt{\lambda_i} f_j(\mathbf{t})) & j = 1, ..., M \end{cases} \tag{7.7}$$

The SFEM equations of the static problem become:

$$\left(\mathbf{K}_0 + \sum_{j=1}^{M} \mathbf{K}_j \xi_j(\theta)\right) \mathbf{U}(\theta) = \mathbf{F} \tag{7.8}$$

The Monte Carlo simulation consists of the following steps:

1. Assembly the system matrices $\boldsymbol{K}_j$ according to eq.(7.7)
2. Generate of the sets $\{\xi_1(\theta_i), ..., \{\xi_M(\theta_i)\}$ with $i = 1, ..., n_{MCS}$ of independent standard Gaussian variables.
3. Assembly of the stiffness matrix $\boldsymbol{K}(\theta_i)$ for every realization i and solution of the deterministic system of equations $\boldsymbol{K}(\theta_i)\boldsymbol{U}(\theta_i) = \boldsymbol{F}$.
4. Estimation of the PDF of the response based on the statistics of $\{\boldsymbol{U}(\theta_i)\}_i$

7.4 Closed form solution using variability Response functions (VRF)

The concept of variability response functions (VRF) has been proposed in the late 1980s (138), along with different aspects and applications of the VRF (61, 151). A development of this approach was presented in a series of papers (10, 38, 40, 107, 113, 114, 117, 141), where the existence of a closed-form integral expression for the variance of the response displacement of the form

$$Var[u] = \int_{-\infty}^{\infty} VRF(\omega, \sigma) \cdot S(\omega) d\omega \tag{7.9}$$

was conjectured for linear frame-type structures. The variability of a response quantity into a deterministic function (VRF) that is solely dependent on the deterministic components of the structure (geometry, boundary conditions, (mean) material properties and loading of the structural system) and the Spectral Density Function (SDF) of the uncertain system parameters modeled as a homogeneous random field. There are two major drawbacks to the VRF concept. First, the VRF concept is only applicable for statistically homogeneous random fields. No theoretical developments have been made with regards to establishing VRFs for structures with statistically inhomogeneous uncertainties. The second drawback is that the existence of VRFs has only been derived for statically determinate linear structures. A recent methodology, termed the Generalized Variability Response Function (GVRF) Methodology (117, 138, 139), is a Monte Carlo based approach that has been used to establish approximate VRFs for statically indeterminate linear structures where the stochasticity is modeled as a one-dimensional random field.

7.4.1 Fast Monte Carlo simulation

The variability response function can be estimated numerically using a fast Monte Carlo simulation (FMCS) approach whose basic idea of considering stochastic field $X(\mathbf{t})$ as a random sinusoid. The numerical estimation of the VRF through FMCS is extremely important as the closed-form analytic expressions involve modulating functions that are very difficult to establish even in the simplest cases of statically indeterminate beams (138, 139). For this reason, FMCS is used exclusively to determine the VRF.

The basic steps of the FMCS approach are described in the following.

1 Generate N sample functions of a random sinusoid with standard deviation σ and wave number $\overline{\omega}$ modeling stochastic a field $X(\mathbf{t})$ that describes the random property:

$$X_j(\mathbf{t}) = \sqrt{2}\sigma\cos(\overline{\omega}\mathbf{t} + \phi_j) \quad j = 1, 2, ..., N \tag{7.10}$$

where ϕ_j are random phase angles uniformly distributed in the range $[0, 2\pi]$. Rather than picking up the ϕ_j Šs randomly in $[0, 2\pi]$, they can be selected at N equal intervals in $[0, 2\pi]$ for significant computational savings.

2 Using these N generated sample functions of $X_j(\mathbf{t})$, it is straightforward to compute the corresponding N displacement responses either analytically or numerically. Then, the mean value of the response $\varepsilon[u(\mathbf{t})]_{\overline{\omega}}$ and its variance $Var[u(\mathbf{t})]_{\overline{\omega}}$ can be easily determined for the specific value of $\overline{\omega}$ considered by ensemble averaging the N computed responses.

3 The value of the variability response function (VRF) of the statically indeterminate beam at wave number $\overline{\omega}$ and for standard deviation σ is computed from

$$VRF(\mathbf{t}, \overline{\omega}, \sigma) = \frac{Var[u(\mathbf{t})]_{\overline{\omega}}}{\sigma^2} \tag{7.11}$$

4 Steps 1 – 3 are repeated for different values of the wave number $\overline{\omega}$ of the random sinusoid. Consequently $VRF(\mathbf{t}, \overline{\omega}, \sigma)$ are computed over a wide range of wave numbers, wave number by wave number. The entire procedure can be eventually repeated for different values of the standard deviation σ and for different locations $\mathbf{t}$ along the domain.

It should be pointed out that the fast Monte Carlo simulation procedure can be implemented into the framework of a deterministic finite element code making this approach very general. Specifically, the N displacement responses in the second step of the aforementioned procedure can be computed numerically using any general purpose finite element code. To do this, every generated sample function modeling the elastic modulus is used as a direct input in a FEM code.

7.5 Spectral Stochastic Finite element method (SSFEM)

The original work by Ghanem and Spanos (54, 55) deals with linear stochastic boundary value problems in which the spatial variability of a material property (e.g. YoungŠs modulus) is modeled using a random field, which is discretized using the Karhunen-Loève expansion. Later on, the same approach has been applied to transport in porous media (50), heat transfer (52, 156) and soil-structure interaction (56) and structural dynamics (150). In all these applications, the spatial variability of one or more material properties is represented by a Gaussian or log-normal random field. Attempts to applying the approach to non linear problems can be found in (5, 6) (bounds on the solution are derived) as well as in (80). A general framework for stochastic mechanics based on these ideas is described by Ghanem in (51). Babuska et al. propose a similar framework and discuss convergence issues and error estimators (15, 35)

7.5.1 SSFEM of Gaussian random fields

In the SSFEM approach, the system response is projected in a polynomial chaos (PC) basis as follows

$$\mathbf{u}(\theta) = \sum_{j=0}^{Q-1} \mathbf{u}_j \Psi_j(\boldsymbol{\xi}) \tag{7.12}$$

where $\{\Psi_j(\boldsymbol{\xi})\}_{j=0}^{Q-1} = \{\Psi_j((\xi_1(\theta), ..., \xi_M(\theta)))\}_{j=0}^{Q-1}$ is the PC basis, consisting of the $M-$dimensional zero mean and orthogonal Hermite polynomials of order p, satisfying:

$$\begin{aligned} \Psi_0 &= 1 \\ E[\Psi_j] &= 0 \quad j > 0 \\ E[\Psi_j(\theta)\Psi_k(\theta)] &= 0 \quad j \neq k \end{aligned} \tag{7.13}$$

The value of Q in eq.(7.12) is determined by the following formula

$$Q = \binom{M+p}{p} \tag{7.14}$$

In this case eq.(7.7) can be written as

$$\left(\sum_{i=0}^{M} \mathbf{K}_i \xi_i(\theta)\right) \cdot \left(\sum_{j=0}^{Q-1} \mathbf{u}_j \Psi_j(\theta)\right) = \mathbf{f} \tag{7.15}$$

with the following residual due to the truncation error

$$\boldsymbol{\epsilon}_{M,Q} = \sum_{i=0}^{M} \sum_{j=0}^{Q-1} \mathbf{K}_i \mathbf{u}_j \xi_i(\theta) \Psi_j(\theta) - \mathbf{f} \tag{7.16}$$

The best approximation of the exact solution $\mathbf{u}(\theta)$ spanned by $\{\Psi_k\}_{k=0}^{Q-1}$ is obtained by minimizing this residual in a mean square sense:

$$E[\boldsymbol{\epsilon}_{M,Q} \cdot \Psi_k] = 0 \quad k = 0, 1, ..., Q-1 \tag{7.17}$$

By introducing the following notation:

$$c_{ijk} = E[\xi_i \Psi_j \Psi_k] \tag{7.18}$$

$$\mathbf{f}_k = E[\Psi_k \mathbf{f}] \tag{7.19}$$

eq.(7.15) can be rewritten as

$$\sum_{i=0}^{M} \sum_{j=0}^{Q-1} c_{ijk} \mathbf{K}_i \mathbf{u}_j = \mathbf{f}_k \quad k = 0, ..., Q-1 \tag{7.20}$$

and by defining for the sake of simplicity

$$\mathbf{K}_{jk} = \sum_{i=0}^{M+1} c_{ijk} \mathbf{K}_i \tag{7.21}$$

eq.(7.20) rewrites as follows:

$$\sum_{j=0}^{Q-1} \mathbf{K}_{jk}\mathbf{u}_j = \mathbf{f}_k \tag{7.22}$$

In the above equation, each $\mathbf{u}_j$ is a $N-$dimensional vector and each $\mathbf{K}_{jk}$ is a matrix of size $N \times N$. The Q different equations can be cast in a matrix form of size $(N \cdot Q \times N \cdot Q)$ as follows:

$$\boldsymbol{K} \cdot \boldsymbol{u} = \boldsymbol{f} \tag{7.23}$$

where,

$$\boldsymbol{K} = \begin{bmatrix} \sum_{i=0}^{M} c_{i,0,0}\mathbf{K}_i & \sum_{i=0}^{M} c_{i,1,0}\mathbf{K}_i & \cdots & \sum_{i=0}^{M} c_{i,Q-1,0}\mathbf{K}_i \\ \sum_{i=0}^{M} c_{i,0,1}\mathbf{K}_i & \sum_{i=0}^{M} c_{i,1,1}\mathbf{K}_i & \cdots & \sum_{i=0}^{M} c_{i,Q-1,1}\mathbf{K}_i \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{M} c_{i,0,Q-1}\mathbf{K}_i & \sum_{i=0}^{M} c_{i,1,Q-1}\mathbf{K}_i & \cdots & \sum_{i=0}^{M} c_{i,Q-1,Q-1}\mathbf{K}_i \end{bmatrix}$$

and

$$\boldsymbol{u} = \begin{bmatrix} \mathbf{u}_0, & \mathbf{u}_1, & \cdots, & \mathbf{u}_{Q-1} \end{bmatrix}^T$$

$$\boldsymbol{f} = \begin{bmatrix} \mathbf{f}_0, & \mathbf{f}_1, & \cdots, & \mathbf{f}_{Q-1} \end{bmatrix}^T$$

The sparsity patterns of the $(N \times N)$ non-zero block sub-matrices of $\boldsymbol{K}$ for two dimensional elasticity cases is shown in Figs. 7.1 and 7.2.

$\mathbf{K}_0$	$\mathbf{K}_1$	$\mathbf{K}_2$	0	0	0
$\mathbf{K}_1$	$\mathbf{K}_0$	0	$2\mathbf{K}_1$	$\mathbf{K}_2$	0
$\mathbf{K}_2$	0	$\mathbf{K}_0$	0	$\mathbf{K}_1$	$2\mathbf{K}_2$
0	$2\mathbf{K}_1$	0	$2\mathbf{K}_0$	0	0
0	$\mathbf{K}_2$	$\mathbf{K}_1$	0	$\mathbf{K}_0$	0
0	0	$2\mathbf{K}_2$	0	0	$2\mathbf{K}_0$

Figure 7.1: Sparsity pattern of $\boldsymbol{K}$ for the Gaussian case ($M = 2,\ p = 2$)

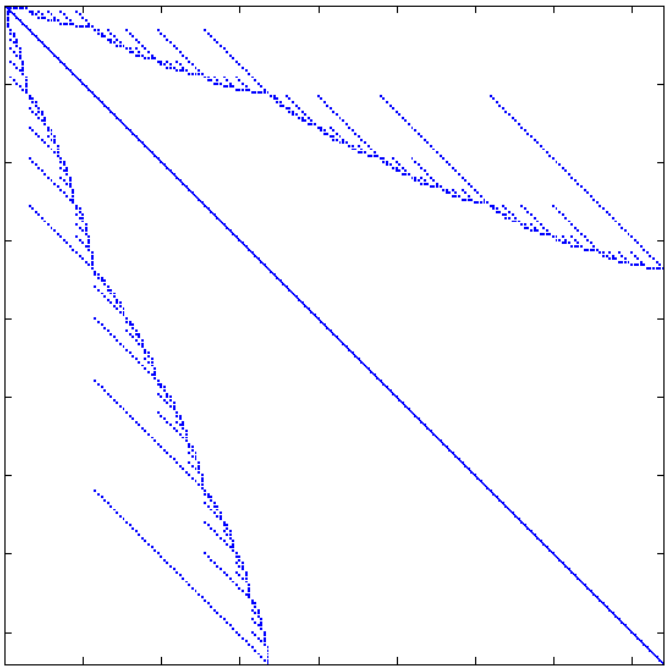


Figure 7.2: Sparsity pattern of $\boldsymbol{K}$ for the Gaussian case ($M = 6,\ p = 4$)

When dealing with Gaussian stochastic fields, each block of the diagonal is comprised of the deterministic stiffness matrix $\mathbf{K}_0$ scaled by an integer. After solving the aug-

mented system for $\boldsymbol{u} = \{\mathbf{u}_k, k = 0, ..., Q-1\}$, the required $\mathbf{u}(\theta)$ is computed from:

$$\mathbf{u}(\theta) = \sum_{j=0}^{Q-1} \mathbf{u}_j \Psi_j(\theta) \tag{7.24}$$

Once the coefficients $\mathbf{u}_j$ of the expansion are computed, approximate statistics of the solution can be derived by MC simulations. In this case however, the MC simulation computational effort is trivial since it is applied directly to the polynomial representation of eq.(7.24) without the need of solving a system of equations at each simulation.

7.5.2 SSFEM of log-normal random fields

If the random field is considered to be non-Gaussian the probabilistic structure of $\xi_i(\theta)$ in eq.(2.65) cannot be determined so the KL expansion is of no practical interest. For the case of log-normal random fields $l(\mathbf{x}, \theta)$ the truncated KL expansion of an underlying Gaussian field can be defined as follows

$$\hat{l}(\mathbf{x}, \theta) = e^{\{g_0(\mathbf{x}) + \sum_{i=1}^{M} \sqrt{\lambda_i}\xi_i(\theta)\phi_i(\mathbf{x})\}} \tag{7.25}$$

Ghanem proposed in (52) an expansion of $l(\mathbf{x}, \theta)$ into a polynomial basis in order to be able to include these fields in the context of the spectral stochastic finite element method. The truncated PC expansion of a log-normal random field $l(\mathbf{x}, \theta)$ reads

$$\hat{l}(\mathbf{x}, \theta) = \sum_{i=0}^{Q-1} l_i(\mathbf{x}) \Psi_i(\boldsymbol{\xi}) \tag{7.26}$$

In this case the stiffness matrix of eq.(7.7) becomes

$$\mathbf{K}(\theta) = \sum_{i=0}^{Q-1} \mathbf{K}_i \Psi_i(\boldsymbol{\xi}) \tag{7.27}$$

where $\Psi_i(\boldsymbol{\xi})$ is the PC basis. Substituting eq.(7.27) in eq.(7.15), the stochastic equilibrium equation can be written as follows:

$$\left(\sum_{i=0}^{Q-1} \mathbf{K}_i \Psi_i(\theta) \right) \cdot \left(\sum_{j=0}^{Q-1} \mathbf{u}_j \Psi_j(\theta) \right) = \mathbf{f} \tag{7.28}$$

The Galerkin minimization of error leads to a system of linear equations similar to Eq.(7.20) where the coefficients c_{ijk} are replaced by:

$$d_{ijk} = E[\Psi_i \Psi_j \Psi_k] \quad i, j, k = 0, ...Q-1 \tag{7.29}$$

The matrix $\boldsymbol{K}$ retains the block-sparsity nature and its diagonally block-dominance albeit to a lesser extent than the Gaussian case.

In log-normal stochastic fields, each block of the diagonal is comprised either of the deterministic stiffness matrix $\mathbf{K}_0$ scaled by an integer or of a linear combination of the deterministic stiffness matrix $\mathbf{K}_0$ and stochastic matrices $\mathbf{K}_1$ to $\mathbf{K}_n$. Matrices $\mathbf{K}_1$ to $\mathbf{K}_n$ may differ up to orders of magnitude when compared to $\mathbf{K}_0$, so matrix $\boldsymbol{K}$ is also considered block-dominant. Figs. 7.3 and 7.4 depict the sparsity pattern of the non-zero block sub-matrices of $\boldsymbol{K}$ in the case of a log-normal distribution for the same M and p values considered in the Gaussian case of Figs. 7.1 and 7.2. It can be seen that the augmented matrix $\boldsymbol{K}$ in the case of a log-normal random field is far more dense than the one obtained in the case of a Gaussian input.

K_0	K_1	K_2	$2K_3$	K_4	$2K_5$
K_1	(K_0+2K_3)	K_4	$2K_1$	K_2	0
K_2	K_4	(K_0+2K_5)	0	K_1	$2K_2$
$2K_3$	$2K_1$	0	$(2K_0+8K_3)$	K_4	0
K_4	K_2	K_1	K_4	(K_0+2K_3)	K_4
$2K_5$	0	$2K_2$	0	K_4	$(2K_0+8K_5)$

Figure 7.3: Sparsity pattern of $\boldsymbol{K}$ for the non-Gaussian case ($M = 2,\ p = 2$)

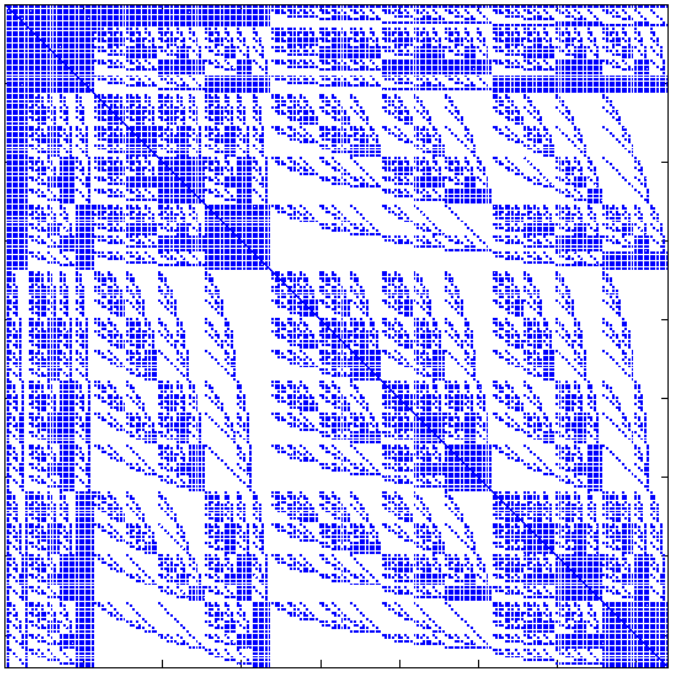


Figure 7.4: Sparsity pattern of $\boldsymbol{K}$ for the non-Gaussian case ($M = 6,\ p = 4$)

7.6 Parametric study of SSFEM vs MC in large-scale systems

Regarding reliability analysis in the case that uncertainties are described as a set of random fields brute force MC approach is the most suitable, versatile and easily implemented method to solve the aforementioned problems. Despite its generality, MC has been used mostly as a means of verifying the accuracy of approximate and less costly procedures due to its usually high computational cost. Alternatively, SSFEM is a method which involves the solution of an augmented problem. In order to perform an assessment of the range of the relative superiority of MC and SSFEM with regard to a variety of stochastic parameters a numerical test is considered in which the second order moments and a reliability analysis are carried out for both MC and SSFEM methods are estimated (143).

For this purpose, a soil prisma of $10 \times 10 \times 20$ meters under load in the center of its upper surface due to a large footing was considered, resulting to a finite element mesh of 10k dof approximately. A multi-parametric study has been carried out first, considering both Gaussian and log-normal stochastic fields.

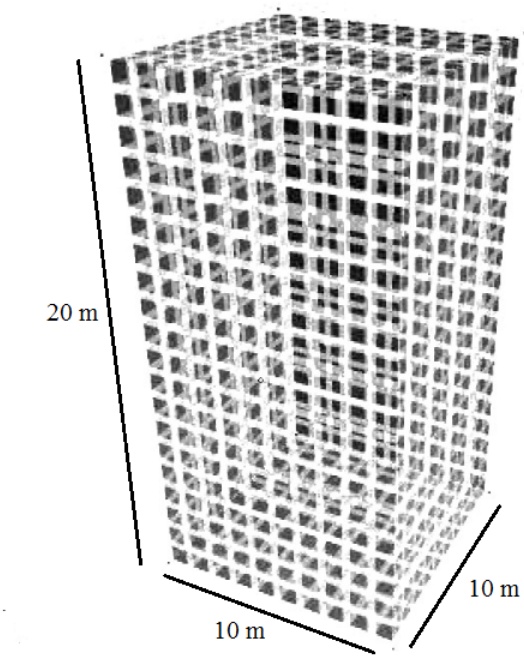


Figure 7.5: Element mesh of a quarter of the deterministic soil problem with 10k dof

One dimensional stochastic fields are used to describe the spatial variation of the system's modulus of elasticity E around its mean as $E = E_0 \cdot (1 + f(\mathbf{x}))$, where E_0 is the mean value of E and $f(\mathbf{x})$ a zero-mean homogeneous stochastic field with variance σ_E^2. The covariance function of the random field $f(\mathbf{x})$ is assumed to be exponential:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \sigma_E^2 e^{-\frac{|\Delta \mathbf{x}|}{\mathrm{b}}} \tag{7.30}$$

where $\Delta \mathbf{x} = \mathbf{x}_2 - \mathbf{x}_1$. In order to set an objective basis three test cases regarding the random field's distribution and c.o.v are examined: i) Gaussian distribution with c.o.v = 15% , ii) Log-normal distribution with c.o.v = 30% and iii) Log-normal distribution with c.o.v = 80% . Moreover, four correlation length values are assumed: (a) b= $0.1a$, (b) b= $1a$, (c) b= $10a$ and (d) b= $100a$, with $a = 20m$ being the height of the prisma. For all these test cases, two separate problems are addressed: evaluation of the second moments of the response field and a reliability analysis with 0.1% probability of failure. For the computation of the second moments of the response field, the following procedure was followed:

Step 1. A series of Monte Carlo analyses of $100k$ simulations was carried out, using $M = 1$ as the order of the KL expansion, in order to estimate the necessary number of simulations for a convergence error of less than 0.5% for each value of c.o.v and b examined. This error is computed as the normalized difference of

the response's c.o.v (%) at each simulation with respect to the response's c.o.v (%) computed at the end of the $100k$ simulations.

Step 2. Assuming that the convergence behavior of the previous step remains invariant for increasing M, another series of Monte Carlo analyses was carried out, in the range of $M = 2$ to $M = 12$, in order to estimate the appropriate order of the KL expansion for a convergence error of less than 1%. In this case an "exact" solution was assumed at $M = 12$ in order to compute the relative error (%) for different M.

Step 3. Using the results of step 2, the same procedure as in step 1 was carried out performing SSFEM analyses, in order to estimate the appropriate order of the PC expansion required for convergence to the corresponding MC results.

Step 4. For the case of reliability analysis with 0.1% target probability of failure, the order of the PC expansion is being modified, with respect to step 3 (convergence in response's c.o.v (%), in order to reach a convergence error in the estimation of the probability of failure of less than 10%, compared to the corresponding MC results. The number of simulations for both MC and SSFEM is in this case $100k$.

7.6.1 Computation of the second moments of the response field

Figs. 7.6 to 7.8 show the convergence error for each field as per step 1 of the assessment procedure. Based on these figures, the number of simulations necessary for evaluating the second moments of the response field are shown in Table 7.1.

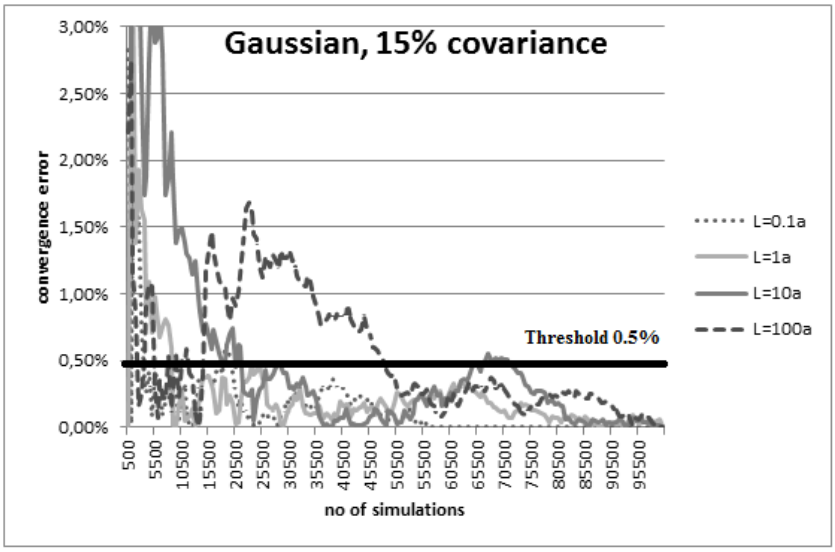


Figure 7.6: Step 1: Response's c.o.v (%) convergence error of MC for the Gaussian field with c.o.v = 15%

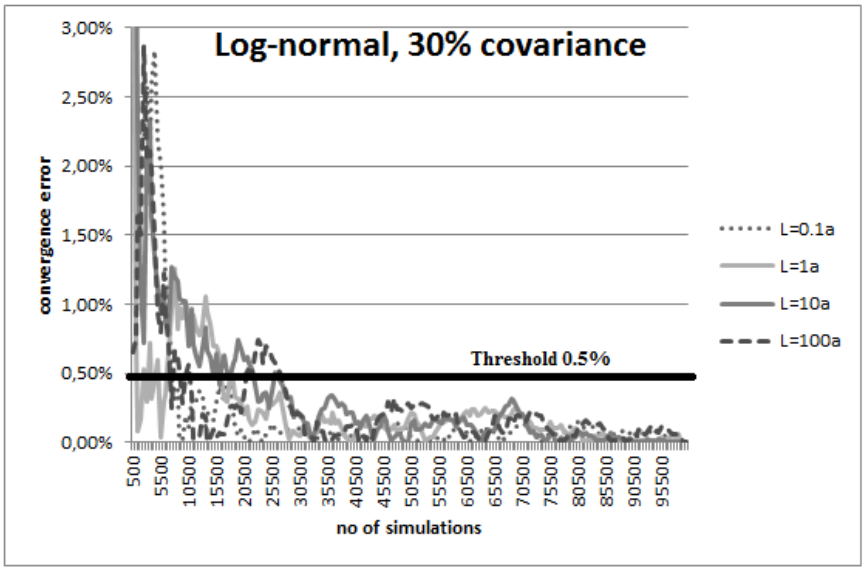


Figure 7.7: Step 1: Response's c.o.v (%) convergence error of MC for the log-normal field with c.o.v 30%

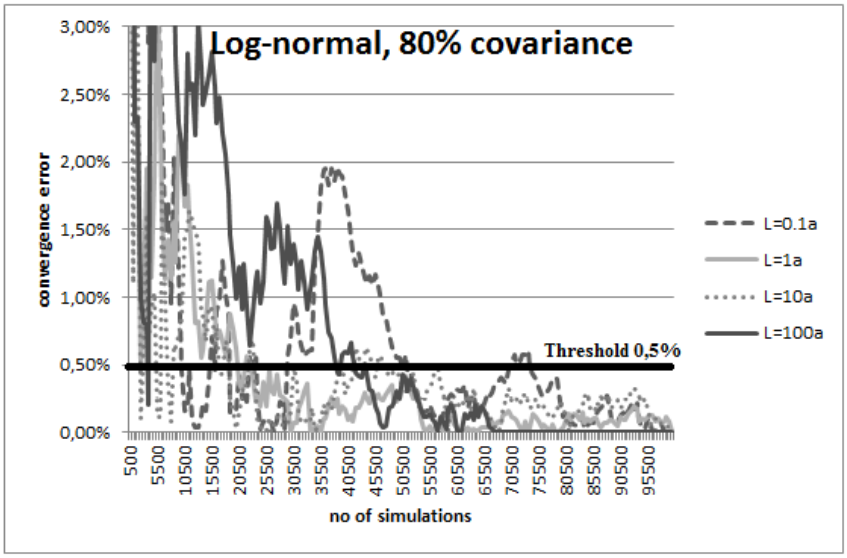


Figure 7.8: Step 1: Response's c.o.v (%) convergence error of MC for the log-normal field with c.o.v = 80%

Correlation length b	**c.o.v** = 15%	**c.o.v** = 30%	**c.o.v** = 80%
0.1a	20000	10000	50500
1a	23000	16000	25000
10a	20500	20000	26000
100a	46000	26000	40500

Table 7.1: Required number of MC simulations for achieving a response's c.o.v error less than 0.5%.

Correlation length b	**c.o.v** = 15%		**c.o.v** = 30%		**c.o.v** = 80%	
	M	**Error** (%)	***M***	**Error** (%)	***M***	**Error** (%)
0.1a	12	"exact"	10	0.43	4	0.75
1a	6	0.93	4	0.75	4	0.57
10a	2	0.36	2	0.85	4	0.26
100a	2	0.48	2	0.53	4	0.96

Table 7.2: Step 2: Response's c.o.v (%) convergence errors for the various KL expansion orders

Following step 2, Figs. 7.9 to 7.11 show the convergence error for each field as per step 3 of the assessment procedure for the selection of PC expansion order (p) required

for the SSFEM to converge at an error less than 1% using the KL expansion orders M shown in Table 7.2. This relative error is computed with respect to the corresponding MC simulations with the same parameter M.

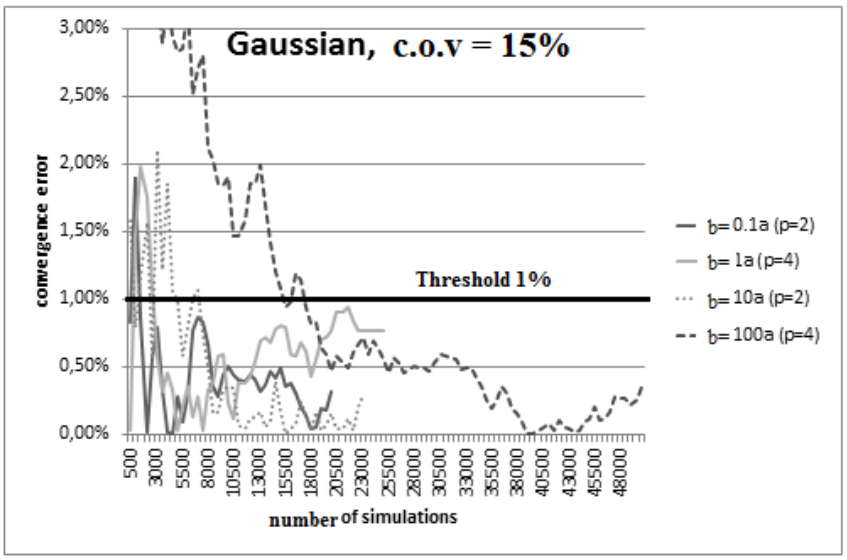


Figure 7.9: Step 3: Response's c.o.v (%) convergence error of the SSFEM for the Gaussian field with c.o.v = 15% and $p = 2, 4$.

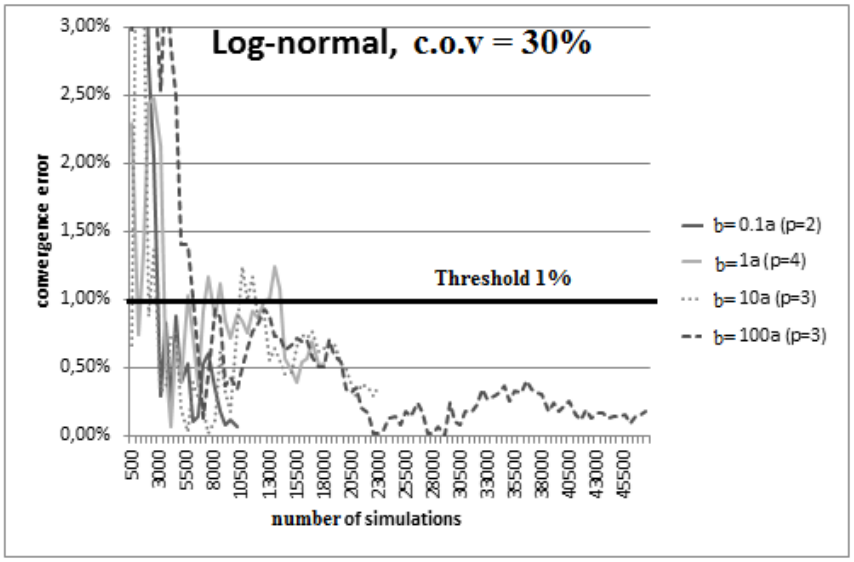


Figure 7.10: Step 3: Response's c.o.v (%) convergence error of the SSFEM for the log-normal field with c.o.v = 30% and $p = 2, 3, 4$.

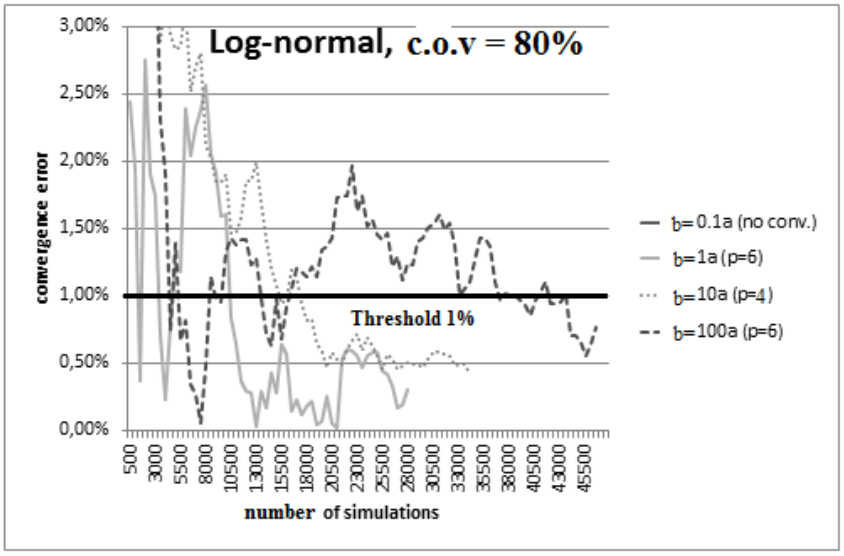


Figure 7.11: Step 3 : Response's c.o.v (%) convergence error of the SSFEM for the log-normal field with c.o.v = 80% and $p = 2, 3, 4$.

Figs. 7.12 to 7.14 depict some indicative graphs of the convergence behavior of the SSFEM in specific cases. Table 7.3 summarizes the convergence of the SSFEM (relative error %) with respect to MC, for all cases considered.

Correlation	**c.o.v** = 15%		**c.o.v** = 30%		**c.o.v** = 80%	
length b	***p***	**Error** (%)	***p***	**Error** (%)	***p***	**Error** (%)
$0.1a$	2	0.23	2	0.07	6	30.00
$1a$	4	0.09	4	0.69	6	0.52
$10a$	2	0.03	3	0.36	4	0.68
$100a$	4	0.36	3	0.74	6	0.88

Table 7.3: Convergence errors for the SSFEM

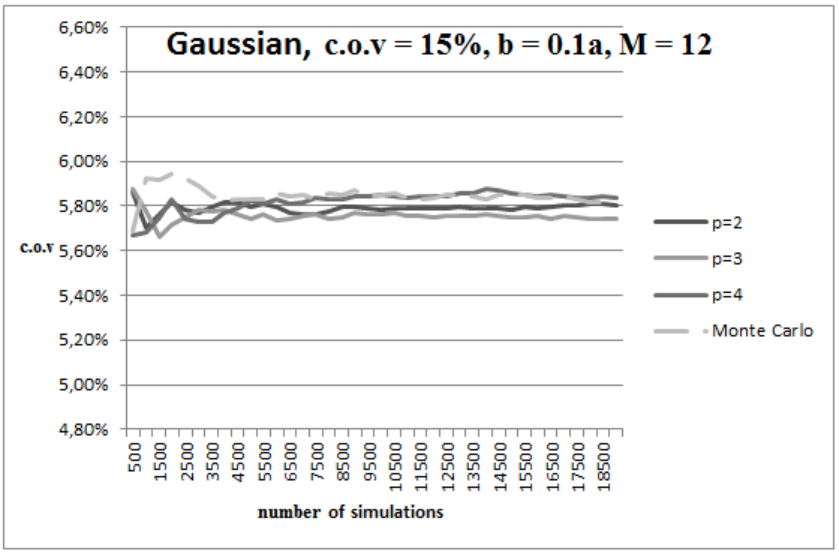


Figure 7.12: Settlement's c.o.v for random field with c.o.v = 15% , correlation length 2m, and $M = 12$ for both MC and SSFEM.

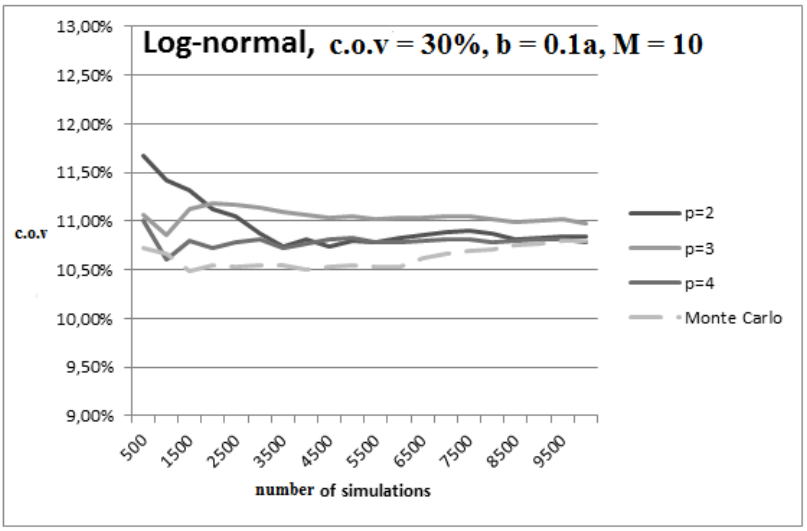


Figure 7.13: Settlement's c.o.v for random field with c.o.v = 30% , correlation length 2m, and $M = 10$ for both MC and SSFEM

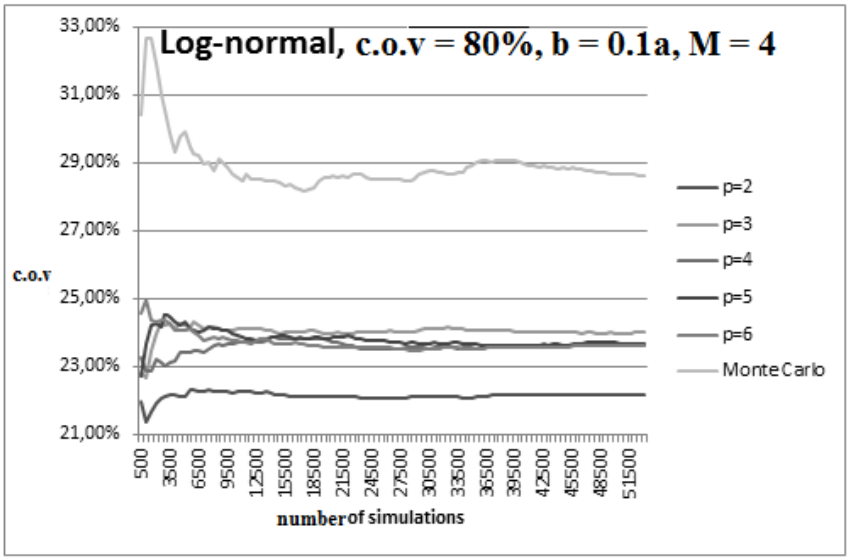


Figure 7.14: Settlement's c.o.v for random field with c.o.v = 80% , correlation length 2m, and $M = 4$ for MC and SSFEM

It is worth noting that for the case of b=0.1a, the SSFEM failed to provide a solution within the acceptable error margin when compared to the MC solution. While increasing the p-order of the PC expansion, the SFFEM method was asymptotically converging to a solution which exhibited a 30% error when compared to the corresponding Monte Carlo solution.

7.6.2 Reliability analysis

Utilizing the values of M obtained at step 2 of the solver assessment procedure for the computation of the second moments of the response field, we performed reliability analysis on the same test problem. Table 7.4 shows the settlement values which correspond to a probability of failure of 0.1%, as estimated by MC with 100k simulations, for various stochastic parameters (c.o.v and b) considered. Table 7.5 shows the probability of failure for the corresponding limit state settlements of Table 7.4 using SSFEM with the KL and PC order expansions used for the second order moments analysis as shown in Tables 7.2 and 7.3, respectively. The settlement values of Table 7.4 are used as reference values, i.e. as the limit states that correspond to a probability of failure 0.1% for all cases considered.

Correlation length b	MC		
	c.o.v = 15%	**c.o.v** = 30%	**c.o.v** = 80%
0.1a	0.019405	0.023468	0.076088
1a	0.024796	0.032947	0.118410
10a	0.028214	0.038950	0.166543
100a	0.029439	0.039841	0.168306

Table 7.4: Settlements (m) with 0.1% probability of failure

Correlation length b	SSFEM		
	c.o.v = 15%	**c.o.v** = 30%	**c.o.v** = 80%
0.1a	0.06 %	0.04 %	-
1a	0.09 %	0.09 %	0.07 %
10a	0.03 %	0.09 %	0.01 %
100a	0.07 %	0.10 %	0.10 %

Table 7.5: Probability of failure as computed by SSFEM for the settlements of Table 7.4 for the limit states

Table 7.6 shows the same probability of failure with the PC order expansion needed to reach almost the same accuracy with the "reference" MC solution. Values marked in bold correspond to analyses that needed an increase of the PC order expansion.

Correlation length b	**c.o.v** = 15%		**c.o.v** = 30%		**c.o.v** = 80%	
	p	**prob**	***p***	**prob**	***p***	**prob**
0.1a	**4**	**0.09**%	**5**	**0.09**%	-	-
1a	4	0.09%	4	0.09%	**7**	0.09 %
10a	**4**	**0.09**%	3	0.09 %	**8**	**0.10**%
100a	**6**	**0.09**%	3	0.10 %	6	0.10 %

Table 7.6: Probability of failure as computed by SSFEM for the settlements of Table 7.4 and the necessary PC order

For the case of b=0.1a and c.o.v = 80%, a series of analyses were performed with various expansion orders M and p, going up to $M = 12$ and $p = 6$. However, SSFEM failed to converge to an acceptable solution resulting to a minimum convergence error of 20%.

7.7 Conclusions

The efficiency comparison of the Monte Carlo method and SSFEM was made in this work and it was based on the computation of the second order moments of the re-

sponse field as well as on reliability analysis. For the first case, SSFEM proved to be more efficient when dealing with input fields exhibiting small to medium covariance. However, for the case of large covariance, the Monte Carlo outperformed the SSFEM in most cases with the latter being unable to converge in one of the problems cases. Results are similar for the reliability analysis tests with the addition that the SSFEM needed a greater polynomial chaos order to converge to the MC solution, compared to the first uncertainty analysis tests.

8

Neural network assisted stochastic structural mechanics

An efficient ANN-based methodology (59) is developed in the present chapter of the thesis for computing the response statistics (moments and pdf) as well as the reliability of stochastic structural systems, whose material properties are described as random fields, by providing robust ANN estimates of the structural response in the framework of MCS. The corresponding deterministic solution of the problem is performed with the finite element method. To this purpose, a reduced number of input random variables is proposed for training the ANN in the context of SFEM and SFEM-based reliability analysis. The methodology developed is based on the multi–dimensional spectral representation method (140) for the modeling of the involved random quantities where a dimensionality reduction of the stochastic variables is achieved by considering directly the random phase angles of the truncated series expansion as the input vector for the ANN training, instead of the random variables describing the uncertain input parameters. The proposed methodology is applied to Gaussian as well as to non-Gaussian random fields. Since neural networks require a training phase for tuning their parameters, some conventional MCS are needed at the initial phase of the procedure. For this purpose, the Latin hypercube sampling (LHS) technique (87, 100, 111) is applied in a straightforward manner in order to effectively span the random space of the phase angles which, according to the spectral representation, are assumed to be uniformly distributed in the range $[0, 2\pi]$. Thus, the effectiveness of the proposed ANN methodology is further enhanced by the aforementioned uniformly spanned training space. Numerical results are presented demonstrating the efficiency of the proposed methodology in two and three dimensional stochastic structures. Furthermore, it is shown that the methodology can be easily integrated in commercial finite element codes for solving real world problems within an affordable computational time.

The use of artificial neural networks as surrogates to the FEM model in the context of SFEM analysis is prohibitive due to the large dimensionality of the required

input vector in the ANN training process. For example, in the case of a FEM model with 10^6 finite elements which utilizes the midpoint method for the random field mapping, an input vector of up to 10^6 random variables would be required in the ANN architecture. This number is impossible to be handled by any ANN. The basic idea of the proposed approximation is to achieve a dimensionality reduction of the aforementioned training space, by using as input vector directly the random phase angles φ of the spectral representation method. Since the number of the random phase angles depend on the discretization of the power spectrum in the wave number domain the resulting input vector will be independent of the finite element discretization and the number of random variables used for the stochastic description of the uncertain parameter(s). Thus, it is expected that the dimensionality of input vector will be reduced by orders of magnitude, especially in cases where a large scale detailed FEM model is required to describe the structural behavior. A further improvement in the efficiency of the proposed method may be achieved by exploiting the uniform distribution of the random phase angles in the range $[0, 2\pi]$ performing sampling of this uniform training space with a Latin hypercube (LHS) technique, as described below.

It must be made clear however that a sufficiently discretized FEM mesh, both in terms of providing a "convergable" solution as well as of having a mesh size of a fraction of the correlation length in order to accurately represent the fluctuations of the random input parameter, remains a prerequisite for an accurate SFEM analysis. However, this prerequisite is not yet correlated to the size of the input training vector of the ANN since the input consists of the random phase angles and not the realizations of stochastic field modeling the uncertain parameter.

In order to demonstrate the proposed methodology the classical midpoint SFEM approach is considered where, a different random variable $f(x_i)$ must be assigned at the centroid of each element using the spectral representation method (eq.(2.49)). Hence, the structure nodal displacements $\mathbf{u}$ are computed as:

$$\mathbf{u} = \mathbf{K}^{-1}(f(x_i))\mathbf{F}. \tag{8.1}$$

where the stiffness matrix of the structure $\mathbf{K}$ is clearly a function of the random variables $f(x_i)$, $\mathbf{F}$ being the nodal force vector. Conventional implementation of a ANN model as a surrogate to the FE model would use $f(x_i), i = 1, ...N_e$ as the ANN input, where N_e is the number of elements. However, eq.(2.49) indicates that $f(x_i)$ depends on the random phase angles $\varphi_n, n = 1, ..., N$, where N is the number of intervals of the power spectrum discretization. Therefore, eq.(8.1) may be rewritten as:

$$\mathbf{u} = \mathbf{K}^{-1}(\varphi_n)\mathbf{F}. \tag{8.2}$$

Thus, the ANN may be designed having the reduced space of random phase angles φ_n as input vector instead of $f(x_i)$ replacing the dependence of $\mathbf{K}$ on $f(x_i)$ with a dependence on φ_n.

8.0.1 Selection of training points

The appropriate selection of the input training data is also an important factor for a successful training since the training set must include data over the entire range of the output space. Although the number of training patterns plays its own role in the accuracy of the predictions, the distribution of samples is of greater importance. The selection of the input training pairs is based on the requirement that the full range of possible results should be represented in the training procedure. In the present study the selection of the random phase angles of the truncated spectral representation allows for an efficient implementation of LHS sampling due to the fact that the sample space is the same for each random variable and is restricted to the range of $[0, 2\pi]$. Thus, this range is divided into equally spaced distances for the selection of suitable ANN training pairs covering efficiently the total random variables space. The convergence of the training process is controlled by the prediction error according to:

$$\widehat{e}_{NN} = \frac{|Out_{Real} - Out_{NN}|}{Out_{Real}} \times 100 \tag{8.3}$$

where Out_{Real} are the output values of the function evaluations computed by the conventional numerical procedure and the Out_{NN} correspond to the ANN predictions.

8.0.1.1 Latin Hypercube sampling

The advantage of using harmonics-based spectral representation is that they incorporate bounded in $[0, 2\pi]$ and uniformly distributed random phase angles, making this way the ANN training more efficient. In conjunction to the spectral representation and eq.(8.3), the LHS sampling method (87, 100, 111) can be implemented in order to generate pseudo-random numbers at equally spaced iso-probability distances in the range $[0, 2\pi]$ as follows:

- Generate N realizations $[e^{(1)}, ...e^{(N)}]$ of a uniform random vector.
- Define the vectors $\mathbf{u}^{(i)} = \frac{(e^{(i)}-1)}{N}$, $i = 1, ..., N$, each component $u_j^{(i)}$ of $\mathbf{u}^{(i)}$ is located in the interval $[(i-1)N, iN]$.
- Randomly pair, without replacement, the N components of vector $\mathbf{u}^{(1)}$ with those of vector $\mathbf{u}^{(2)}$. The resulting N pairs are then randomly combined with the N components of $\mathbf{u}^{(3)}$. This procedure continues until a set of N samples is formed.
- The obtained set is finally transformed into a set of pseudo-random numbers $[x^{(1)}, ...x^{(N)}]$ that are distributed according to the input joint PDF $f_{\mathbf{X}}(\boldsymbol{x})$.

The LHS technique is further demonstrated for the case of two independent uniformly distributed random phase angles φ_1 and φ_2 in the range of $[0, 2\pi]$ and $N = 8$ (Fig.8.1). The domain $[0, 2\pi]^2$ is first split into N^2 equiprobable segments. Then N points are randomly generated in such a way that there is only one point in each column and in each row.

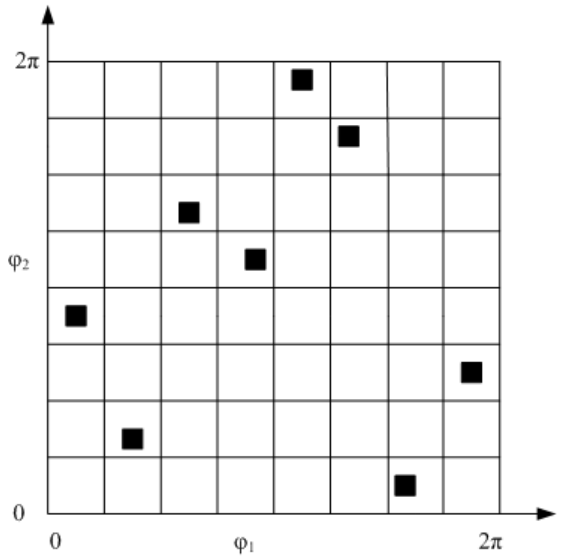


Figure 8.1: Pairing two random variables φ_1, φ_2 using LH sampling

8.0.2 Steps of proposed methodology

The basic steps of the proposed methodology are presented below for the simplest case of one dimensional random fields. The generalization however to 2D and 3D fields is straightforward.

1. Choose the number of terms N for the truncated series of eq.(2.49) which correspond to the number of independent phase angles φ_i used as input vector for the ANN training.

2. Choose an appropriate architecture of the ANN. The input layer consists of N neurons.

3. Use LHS technique in order to generate random phase angles uniformly distributed over $[0, 2\pi]$, as defined in section 8.0.1.1

4. Generate a number of sample functions of the stochastic properties from eq.(2.49) using the phase angles generated in the previous step.

5. Compute the response of the system for the generated sample functions with the finite element method, using the midpoint method, i.e by assigning a corresponding random variable at the centroid of each finite element of the discretized structure.

6. Use the computed response in order to obtain the input/output pair for the ANN training.

7. Estimate $\widehat{e}_{NN}$ from eq.(8.3) in order to evaluate the ANN performance:

   If $\widehat{e}_{NN}$ is smaller than a target value (usually $0.1 \sim 1.0\%$) then go to the next step otherwise go to step 2 using a more refined LHS discretization.

8. Proceed in the MC using ANN.

A schematic representation of the proposed methodology is shown in the Fig.8.2. The advantage of the proposed methodology is that the size of the input vector for the ANN training depends only on the number of terms of the truncated series expansion of the power spectrum. This number orders of magnitude smaller than the number of random variables, describing the random process as the input vector, which makes feasible the ANN implementation in the context of stochastic finite element analysis (SFEM). Two numerical examples are presented in the next section in order to demonstrate the efficiency of the proposed methodology. It must be mentioned that the computer platform used for the computational task was a Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, with 6 GB RAM memory.

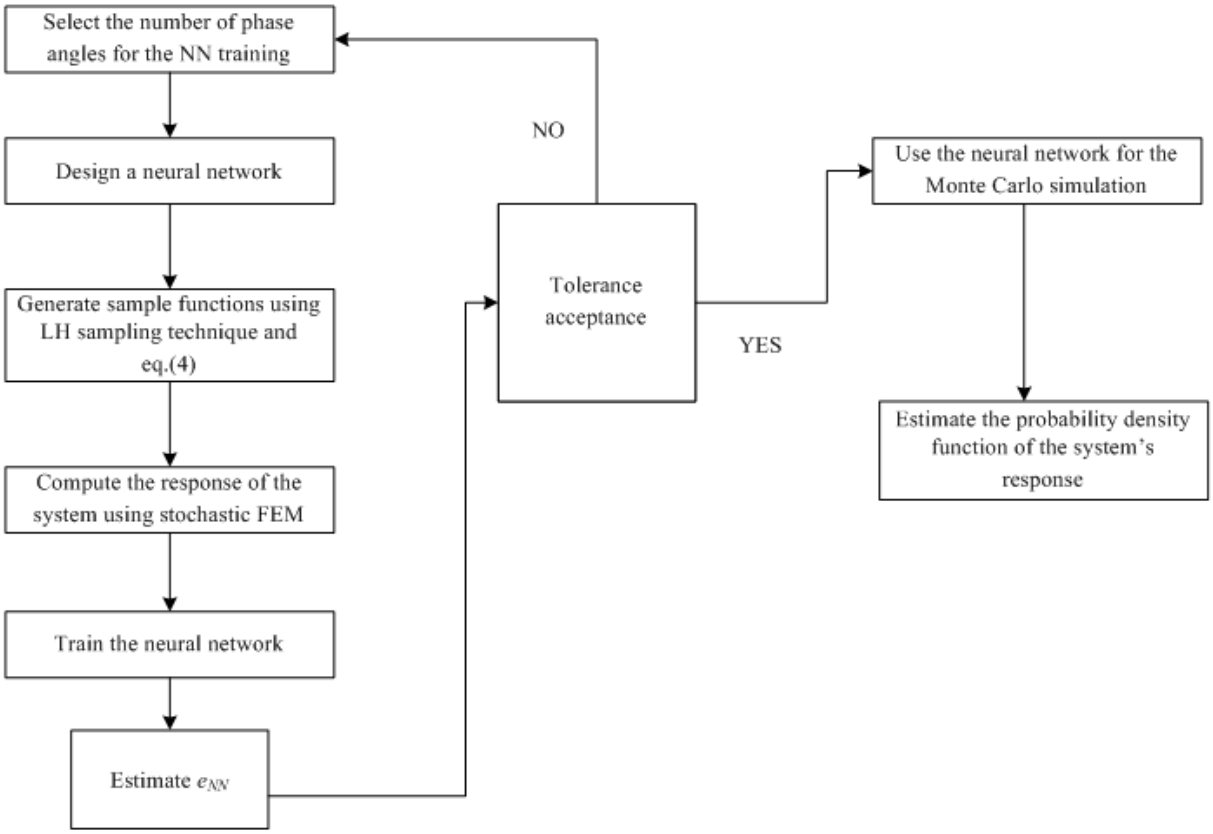


Figure 8.2: Schematic representation of the proposed methodology.

8.1 Numerical Examples

8.1.0.1 2D cantilever with a 2D Gaussian random field

Consider the 2D cantilever of Fig.8.3 with the nodes at the left edge of the cantilever fixed against translations. The dimensions of the beam are $L_x = 10\ m$ and $L_y = 1\ m$ and the cantilever tip is subjected to a deterministic vertical point load $F = 10\ kN$. The cantilever is discretized with 1500 quadrilateral plane stress elements with a total number of 1756 degrees of freedom.

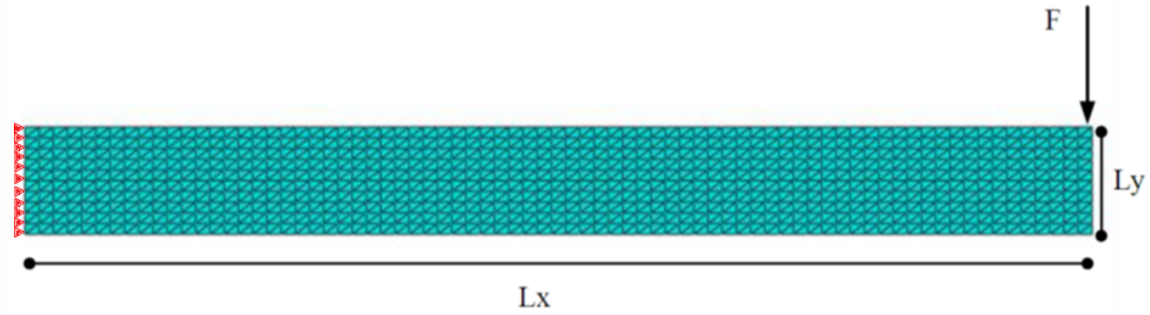


Figure 8.3: Example 1: 2D cantilever beam.

The elasticity modulus $E(x, y)$ is given $E(x, y) = E_0(1 + f(x, y))$ with $E_0 = 21\ GPa$ its mean value and Poisson's ratio $\nu = 0.3$. $f(x, y)$ is a two-dimensional, zero-mean stochastic field with an exponential covariance structure given by:

$$C(x_1, x_2; y_1, y_2) = \sigma^2 e^{-\frac{|\Delta x|}{b_x} - \frac{|\Delta y|}{b_y}} \tag{8.4}$$

where $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$ and $\sigma = 0.15$ is the standard deviation of the random field. In this example,the values of $b_x = 0.1m$ and $b_y = 0.1m$ are chosen as correlation length parameters. The corresponding spectral density function is plotted in Fig.8.4. The upper cut-off frequencies are $\omega_{x_u} = \omega_{y_u} = 30.2\ Hz$.

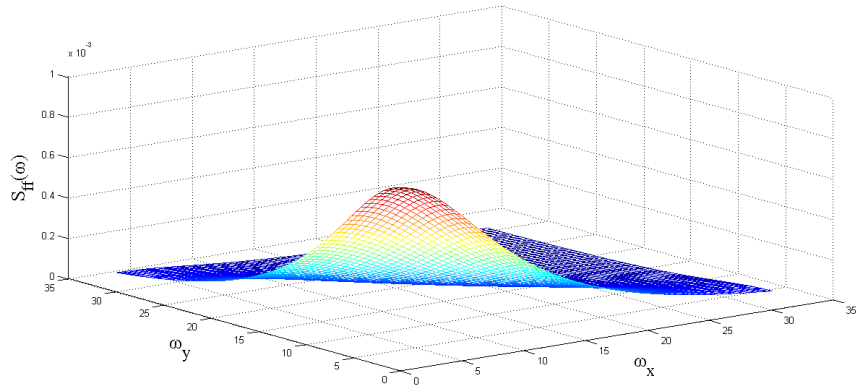


Figure 8.4: Example 1: Spectral density function.

An initial MC is performed in order to estimate the values of $N_x = N_y$ of the truncated series in eq.(2.52) that provide with a reasonable accurate description of the stochastic field. A relative sparse discretization in the wave number domain ($N_x = N_y = 20$) is sufficient to get accurate estimation of the response statistics, as the results for this values are almost identical to the corresponding results for higher values of

N, $(50, 100)$. Therefore, this value is selected for all subsequent applications in this example. Following this preliminary phase, the ANN architecture selected includes $2 \times N_x \times N_y = 800$ neurons (see eq.(2.58)) for the input layer, 10 neurons for the 1st hidden layers and 1 neuron for the output layer.

The system is first solved by means of a brute force MC simulation with $N = 50000$. This number of MC was selected in order to accurately estimate a failure probability in the order of $10^{-2} \sim 10^{-3}$. For each realization, the vertical displacement U_y is calculated. Response statistics are used to estimate the first two moments, namely the mean value and the variance as well as its histogram. The results of the MC are shown in Fig. 8.5 and in Table 8.1 and those of the proposed methodology are depicted in Figs. 8.6 and 8.7 and in Table 8.2, for the cases of 1000 and 4000 training points, respectively. In order to have a fair comparison the results of the proposed ANN-SFEM surrogate model were obtained for the same number of simulations (50000).

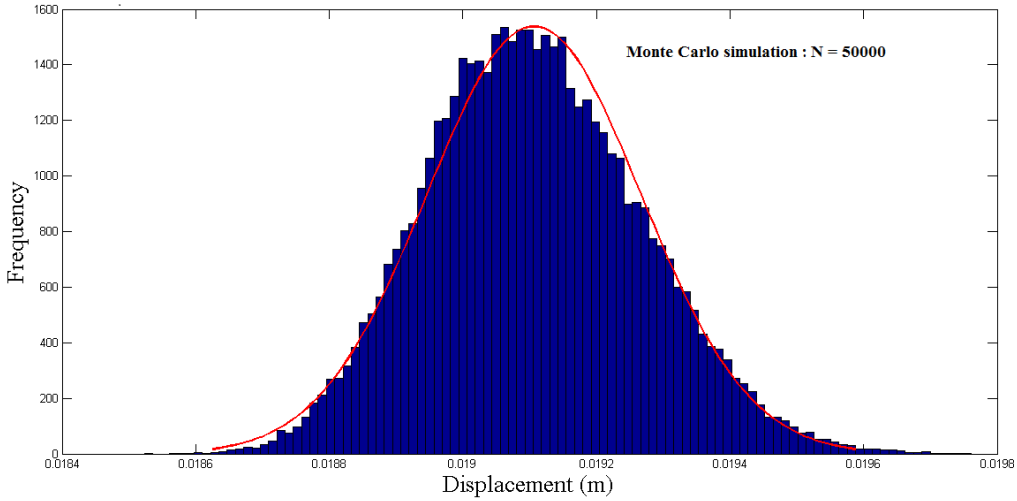


Figure 8.5: Histogram of the displacement of MC simulation

FE analyses	**Moments of U_y**	
	mean (m)	*std*
50000	0.0190	2.1477×10^{-4}

Table 8.1: Example 1: Monte Carlo simulation

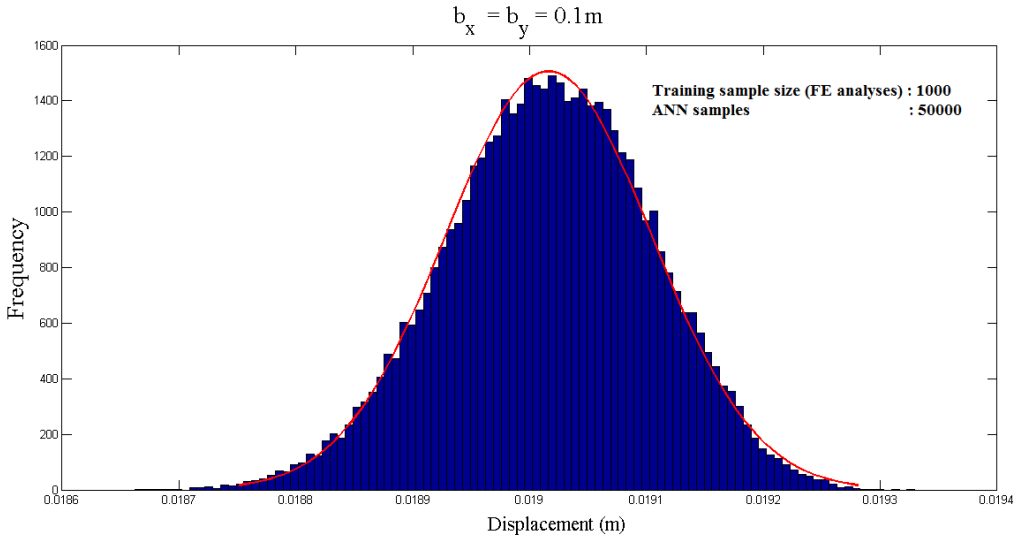


Figure 8.6: Histogram of the displacement for the proposed method with 1000 training points for the ANN

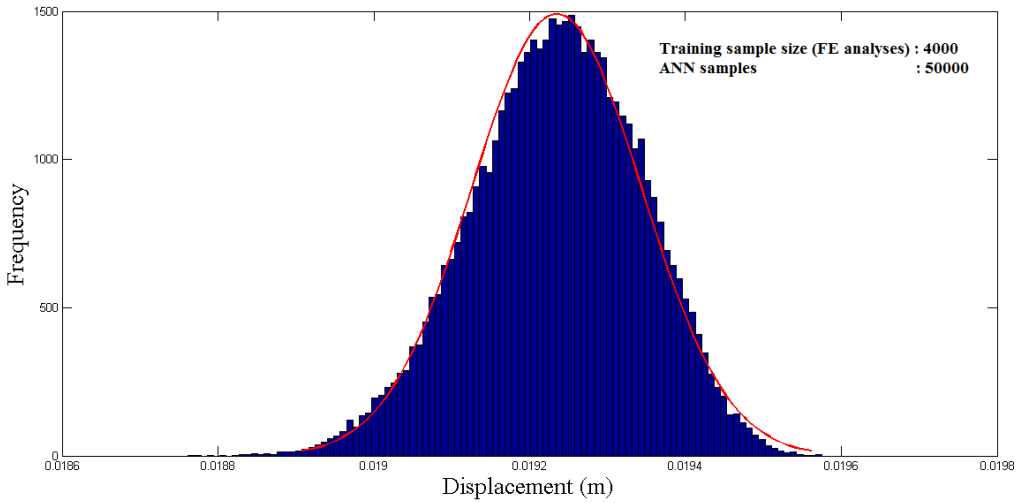


Figure 8.7: Histogram of the displacement for the proposed method with 4000 training points for the ANN

ANN samples	Training sample size (FE analyses)	ANN training error (%)	Moments of U_y	
			mean (m)	**std** (m)
50000	1000	3.54	0.019297	1.6055×10^{-4}
	4000	0.9	0.01901	2.1439×10^{-4}

Table 8.2: Example 1: ANN assisted SFEM

From these Tables it can be seen that very close predictions of the response statistics are obtained with the proposed methodology with respect to MC. A critical role plays the number of training samples used for training the ANN. Specifically for the case of 4000 LHS samples, the relative error of the ANN-SFEM procedure, with respect to MC is less than 0.1% in all cases. As can be observed from these figure, for the case of 4000 training points the distribution of the displacement of the proposed methodology becomes almost identical to one obtained by the conventional MC, while for the case of 1000 training points the results slightly deviate from the MC.

A more clear view of the advantage of the proposed methodology over the standard Monte Carlo simulation is presented in Figs.8.8- 8.10. In these figures the probability of the beam tip displacement to exceed a critical value $d_{cr} = 0.0194$ is computed as a function of the number of samples using MC and the ANN-SFEM proposed methodology. This value of d_{cr} is located 2×std far from the mean displacement response. From these figures it can be seen that using the standard MC, the value of $P(U_{tip} > d_{cr}) = 0.0384$ is computed after 50000 simulations. Almost the same value is obtained with the proposed ANN-SFEM approach and 4000 training points at almost two orders of magnitude less computing time.

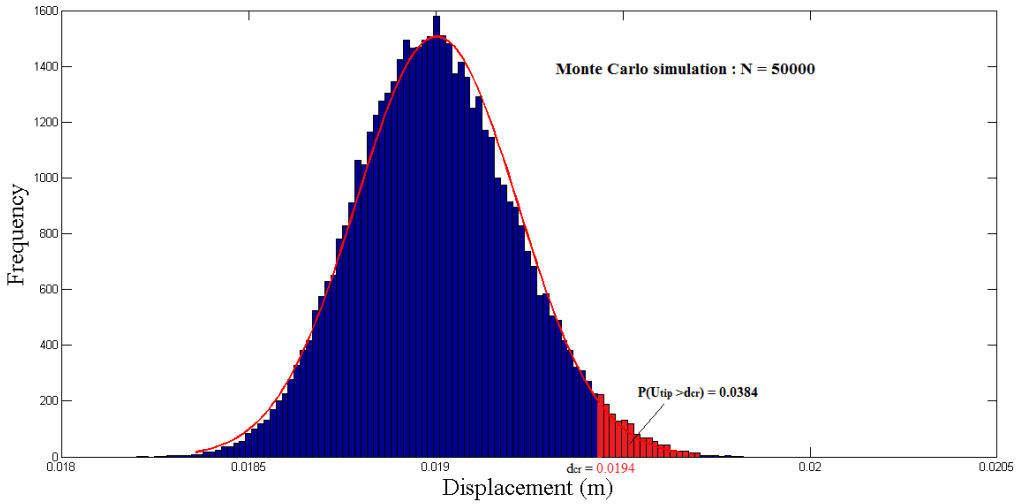


Figure 8.8: Probability $P(U_{tip} > d_{cr}) = 0.0384$ estimated with 50000 MC simulations

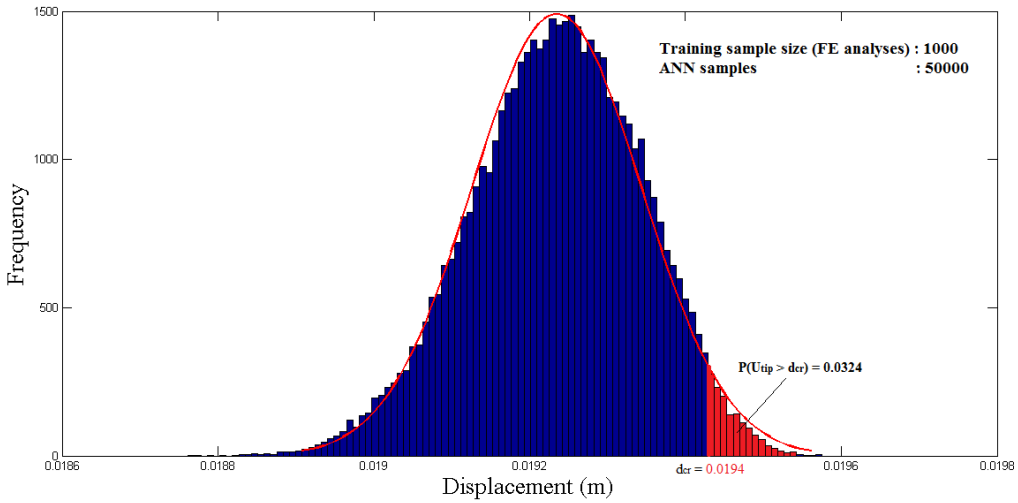


Figure 8.9: Probability $P(U_{tip} > d_{cr}) = 0.0324$ estimated with 50000 ANN samples

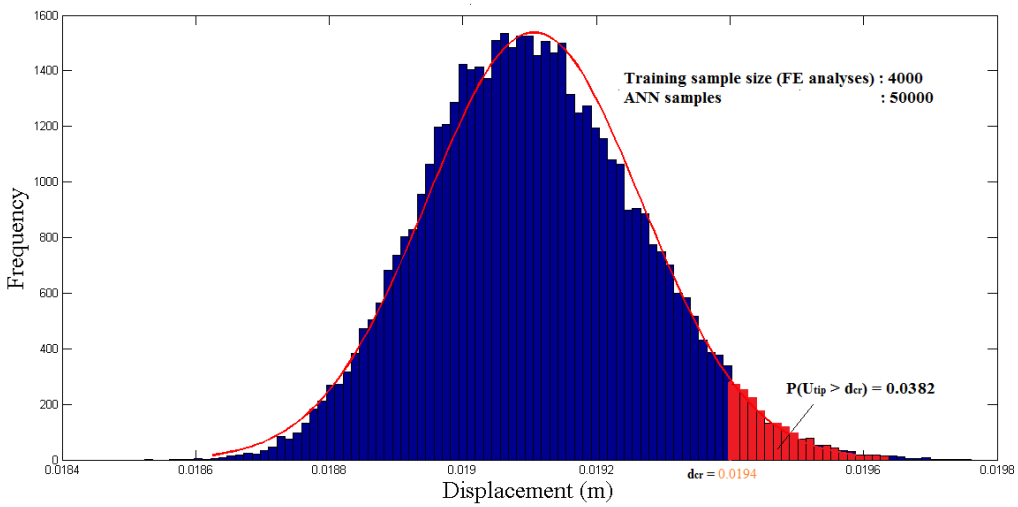


Figure 8.10: Probability $P(U_{tip} > d_{cr}) = 0.0382$ estimated with 50000 ANN samples

Finally, a plot of the ANN prediction error is presented in Fig.8.11 for different values of training samples. From this figure it can be seen that the ANN prediction error decreases rapidly with increasing number of LHS training points. When 500 training points are used, the error is around 6.2%, while by increasing the training points to 4000 the error is reduced to 0.9%. The CPU time required for MC and for the proposed methodology are depicted in Table 8.10. In this Table it can be seen that

the CPU time required for MC is 148299 *sec* while the CPU time required for the same number of simulations with ANN-SFEM with 1000 and 4000 training samples is 2962 *sec* and 13523 *sec*, respectively.

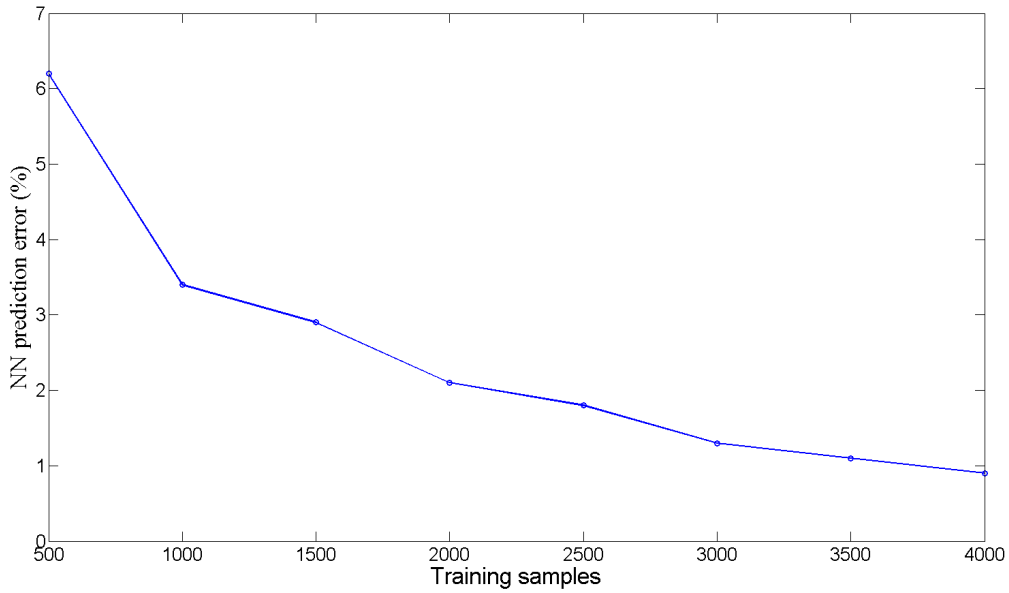


Figure 8.11: Example 1: ANN Validation error with respect to the LHS training points.

Method	FE analyses	ANN samples	CPU time (s)
MC	50000		148299
ANN-SFEM	1000	50000	2962
	4000		13523

Table 8.3: Example 1: CPU time for MC and ANN-SFEM

8.1.1 2D perforated plate with 1D non-Gaussian random field

The proposed methodology can be applied also in cases involving strongly non-Gaussian stochastic fields. A 2D perforated plate of Fig. 8.9 is considered for this case. The two-dimensional domain is a rectangle of length $L_x = 1\ m$ and width $L_y = 1\ m$, with holes in the center of the domain of radius $R = 0.1\ m$ and two symmetric cut-offs at the middle, with dimensions $a = b = 0.33\ m$. The domain is discretized with 400 plane stress quadrilateral elements. The model is subjected to a constant uniform tension load $p = 25\ KN/m$ along its boundary at its upper side.

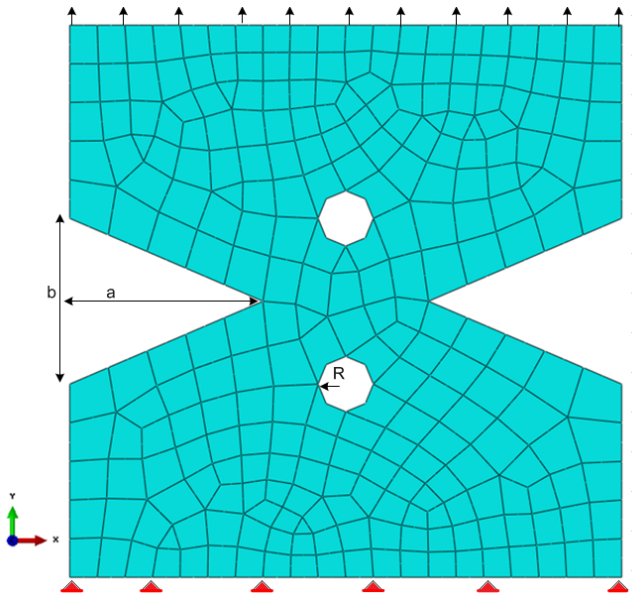


Figure 8.12: Example 2: 2D plane stress plate

The modulus of elasticity is expressed as $E(x) = E_0 \cdot (1 + f(x))$, where $f(x)$ is considered to be a one-dimensional, homogeneous random field varying along the $x-$direction, defined as

$$f(x) = F^{-1}\Phi[g(x)] \tag{8.5}$$

where Φ is the standard Gaussian cumulative distribution function and F is non-Gaussian "U-shaped" beta marginal cumulative distribution function given by:

$$F = \frac{\Gamma(C+D)}{\Gamma(C)\Gamma(D)(B-A)^{C+D-1}}(x-A)^{C-1}(B-x)^{D-1} \tag{8.6}$$

The values of parameters are selected as follows: $A = -1.1$, $B = 1.7$, $C = 0.5$, $D = 0.5$, while the mean value is chosen as $E_0 = 10^5\ KN/m^2$. The transformation $F^{-1} \cdot \Phi$ is a memory-less translation since the value of $f(x)$ at an arbitrary point x depends only on the value of $g(x)$ at the same point and the resulting non-Gaussian field is called translation field. The underlying Gaussian field $g(x)$ is simulated according to the SDE

$$S_g(k) = \sigma_g^2 \cdot \frac{b}{\pi \cdot (1 + b^2k^2)} \tag{8.7}$$

using the spectral representation method with $\sigma_g = 1$ and $b = 5$. For the generation of the Gaussian random field, $N = 128$ terms in the truncated spectral representation are selected. For each realization,the y-displacement of the upper right node is monitored, namely U_y. Response statistics are used to estimate the mean value and the standard deviation σ as well as histograms of U_y.

First, a brute MC simulation is performed. As in the previous example a number of n_{MCS} = 50000 simulations are considered in all cases. The results are gathered in Table 8.4.

FE analyses	**Moments of** U_y	
	mean (m) ($\times 10^{-2}$)	**std** (m) ($\times 10^{-3}$)
50000	1.87	8.6

Table 8.4: Example 2: Monte Carlo simulation

Next, the proposed ANN-SFEM methodology is applied. The ANN architecture contains one hidden layer with 10 neurons, while the first layer (input) has 128 neurons, since the number of terms in the truncated spectral representation is $N = 128$. The results of the proposed methodology for different training sample sizes for the ANN are depicted in Table 8.5.

ANN samples	**Training sample size (FE analyses)**	**Moments of** U_y	
		mean (m) ($\times 10^{-2}$)	**std** (m)
50000	150	1.87	7.8 (9.3%)
	400	1.87	8.5 (1.16%)
	1000	1.87	8.6

Table 8.5: Example 2: ANN assisted SFEM

Finally, the histograms computed using brute force MC and the proposed approach for various training sample sizes are plotted in Fig. 8.10.

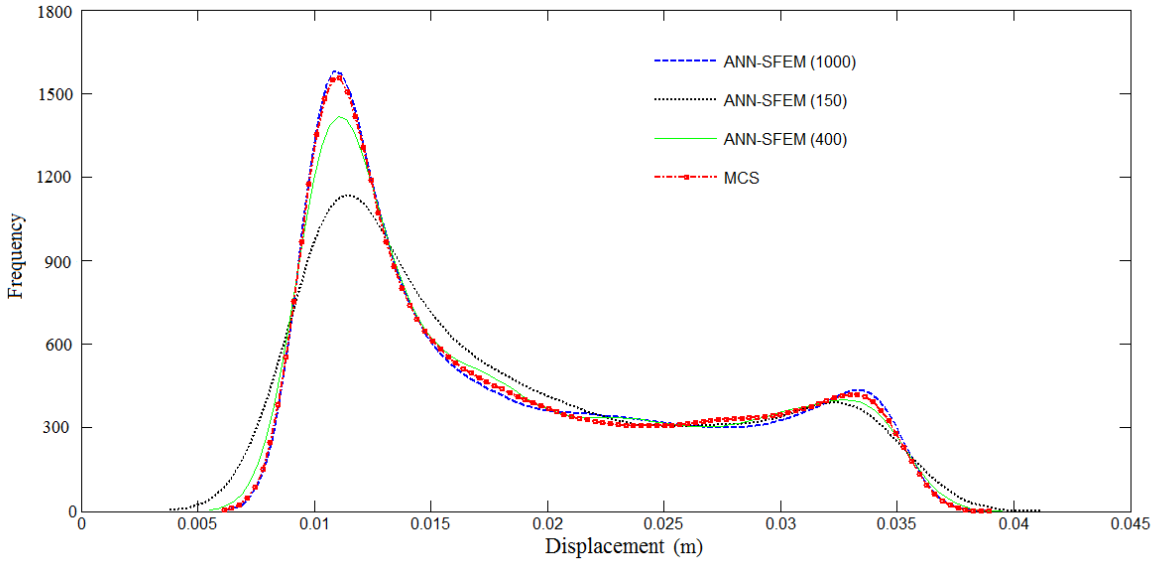


Figure 8.13: Example 2: Probability density functions for the "U-shaped" beta random field for 150, 400 and 1000 training points.

From these results it can be observed that in this strongly non-Gaussian case the proposed methodology also converges to the results of the brute force MC in a robust and efficient manner at a fraction of computational time as it can be seen from Table 8.6.

Method	FE analyses	ANN samples	CPU time (s)
MC	50000		51490
ANN-SFEM	150	50000	155
	400		412
	1000		1030

Table 8.6: Example 2: CPU time for MC and ANN-SFEM

8.1.2 3D cantilever with 3D Gaussian random field

In order to further demonstrate the potential of the proposed combination of ANN and MC for stochastic finite element analysis, the 3D short cantilever of Fig. 8.14 is considered. This example is chosen in order to demonstrate the efficiency of the proposed LHS technique even if a relatively large number of input random phase angles is used in the case of 3D random fields. The dimensions of the cantilever are $L_x = 4m$, $L_y = 2m$ and $L_z = 4m$ with an applied load of $F_z = 1000KN/m$ at its free edge . The finite element mesh consists of 15000 8-node brick elements with a total number of 14236 degrees of freedom.

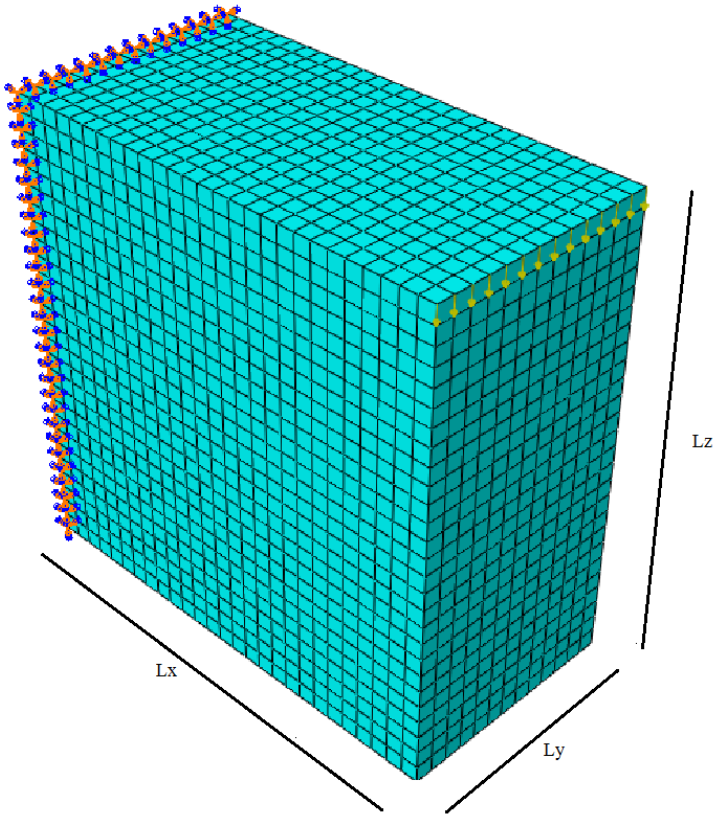


Figure 8.14: Example 3: Short cantilever.

The Young's modulus E is varying randomly and its values are $E = E_0(1 + f(x, y, z))$, where $f(x, y, z)$ is a zero-mean, three-dimensional random field. $E_0 = 21\ GPa$ and $\nu = 0.3$ are the mean value of the Young's modulus and the Poisson's ratio, respectively. The correlation structure of the field is considered to be exponential:

$$C(x_1, x_2; y_1, y_2; z_1, z_2) = \sigma^2 e^{-\frac{|\Delta x|}{b_x} - \frac{|\Delta y|}{b_y} - \frac{|\Delta z|}{b_z}} \tag{8.8}$$

where $\sigma = 0.15$ and $b_x = b_y = b_z = 0.1m$ are the correlation length parameters along the dimensions x, y and z, respectively. The power spectrum is generated by a three-dimensional Fourier transformation of the autocorrelation kernel. The upper cut-off frequency is set at $\omega_u = 10.76Hz$.

A relative sparse discretization in the wave number domain ($N_x = N_y = N_z = 15$) is sufficient to provide accurately enough estimates of the response statistics, therefore, the value of $N = 15$ is selected for all subsequent applications of this case. This leads to a ANN architecture with $4 \times N_x \times N_y \times N_z = 13500$ neurons, according to eq.(2.59), for the input layer while 12 neurons are used in the hidden layer and one in the output.

The system is first solved by means of a brute force MC with $N = 50000$. Again, this number of MC is selected in order to accurately estimate a failure probability in the order of $10^{-2} \sim 10^{-3}$. For each realization, the displacement U_z along the z-axis at the tip of the beam is calculated. Response statistics are used to estimate the mean value, the standard deviation and the histogram of the response. The results of the MC and of the proposed methodology are depicted in Tables 8.5 and 8.6 and in Figs. 8.15 and 8.16, respectively. The ANN-SFEM results are shown for 1000 training points. In order to have a fair comparison the results of the proposed ANN-SFEM surrogate model were obtained for the same number of samples (50000).

FE analyses	**Moments of** U_z	
	mean (m) $(\times 10^{-2})$	**std** (m) $(\times 10^{-3})$
50000	5.399	2.92

Table 8.7: Example 3: Monte Carlo simulation

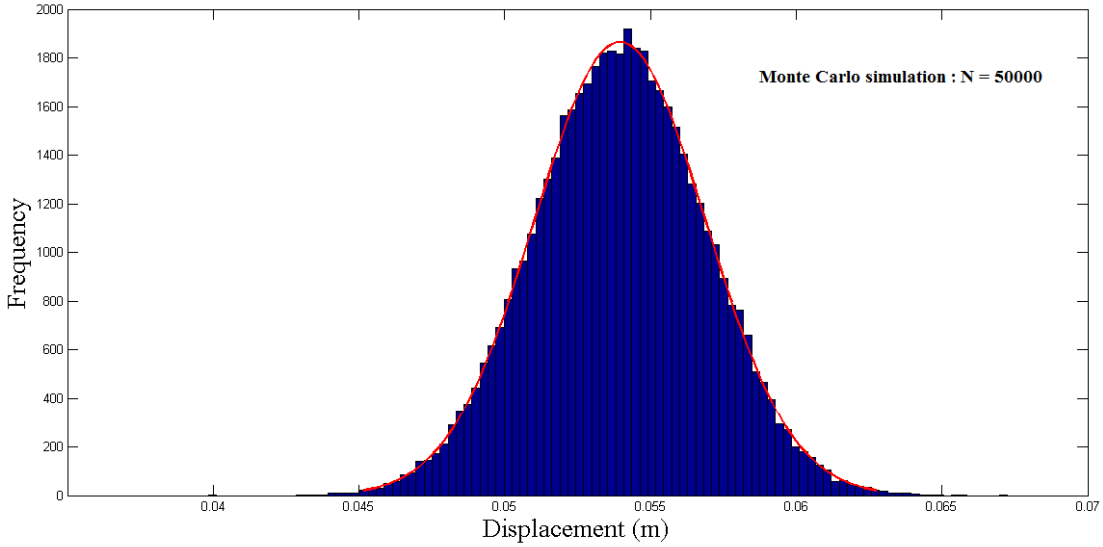


Figure 8.15: Example 3:Histogram of the displacement obtained with brute force MC simulation

ANN samples	Training sample size (FE analyses)	ANN training error (%)	Moments of U_z	
			mean (m) ($\times 10^{-2}$)	**std** (m) ($\times 10^{-3}$)
50000	1000	0.85	5.397 (0.04%)	2.94 (0.68%)

Table 8.8: Example 3: ANN assisted SFEM

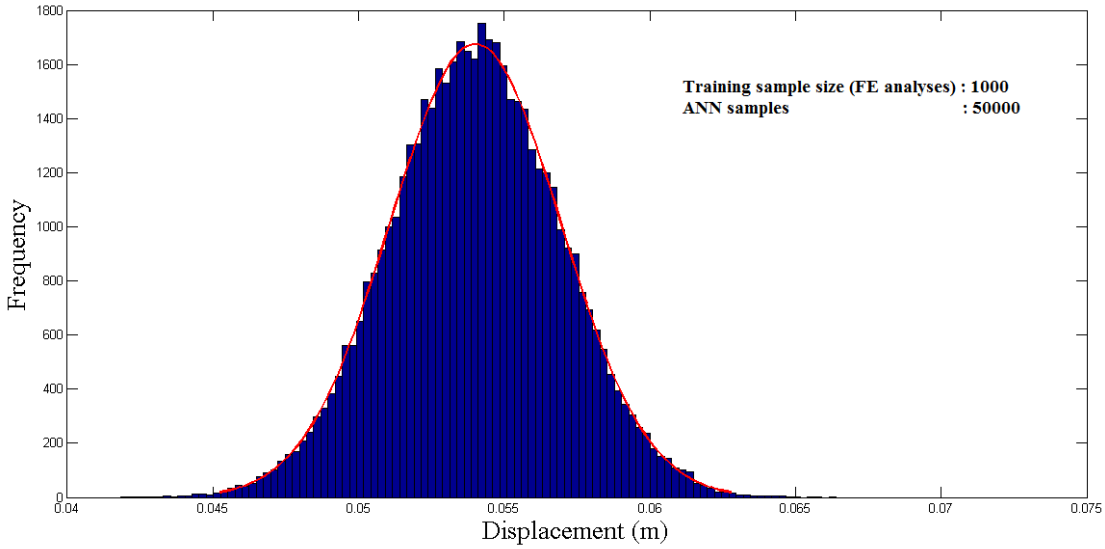


Figure 8.16: Example 3:Histogram of the displacement obtained with the proposed method

From these Tables it is clear that a significant reduction of the computational cost has been achieved by implementing ANN in the framework of the MC simulation without affecting the accuracy of the results. As mentioned above, only 1000 samples where used in order to train the ANN effectively leading to a mean error of the training less than 1%. The relative error of the MC-ANN prediction in the first three moments is of the order of 1%.

Finally, a more clear view of the advantage of the proposed methodology over the standard MC is presented in Figs. 8.17 and 8.18. In these figures the probability of the beam tip displacement to exceed the value $d_{cr} = 0.0613\ m$ ($2.5 \times std$ far from the mean value ($0.05399\ m$)) is computed. From these Figs it can be seen that the value of $P(U_{tip} > d_{cr}) = 0.0068$ is obtained with the crude MC simulation after 50000 simulations while for with the proposed method it is calculated $P(U_{tip} > d_{cr}) = 0.0066$

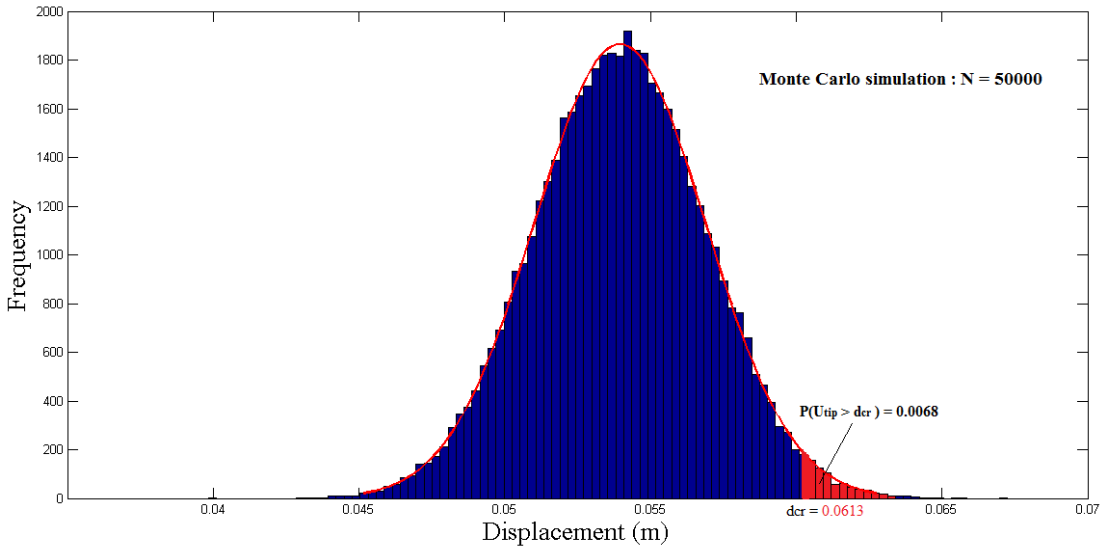


Figure 8.17: Example 3:Probability $P(U_{tip} > d_{cr})$ for the MC

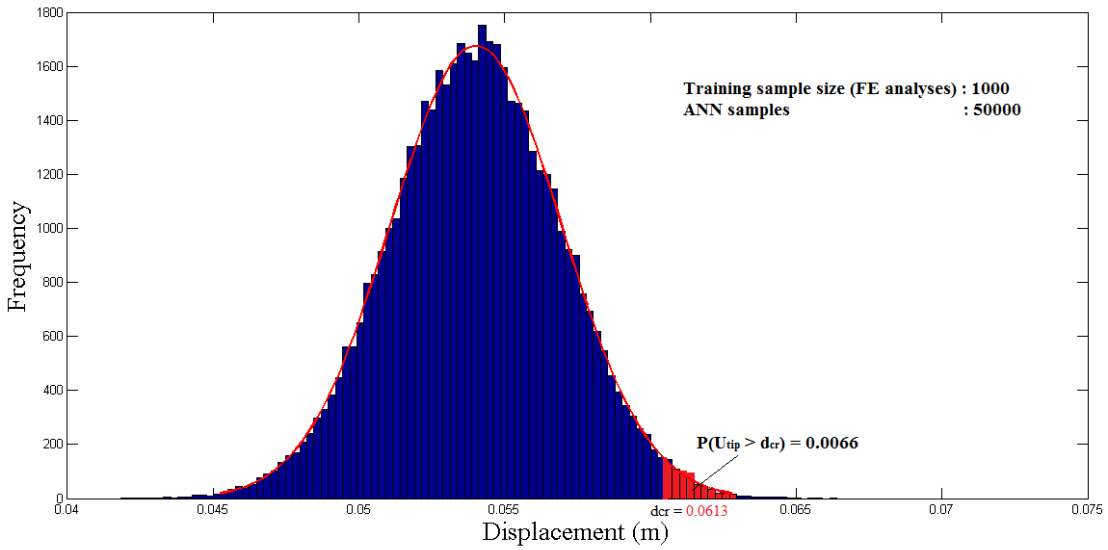


Figure 8.18: Example 3: Probability $P(U_{tip} > d_{cr})$ for the proposed method

The CPU time required in each case is depicted in Table 9. The distinctive characteristic of the adopted approach is that the CPU time necessary to generate the realizations is 1.5x order of magnitude less than the time needed in the conventional MC simula-

tion which makes this method a powerful tool, especially for highly computationally demanding problems.

Method	FE analyses	ANN samples	CPU time (s)
MC	50000		471243
ANN-SFEM	1000	50000	9310

Table 8.9: Example 3: CPU time for MC and ANN-SFEM

8.2 Conclusions

A neural network assisted stochastic finite element analysis is proposed in the context of Monte Carlo simulation for assessing the spread of stochastic response under the presence of uncertain parameters described as random fields. The basic idea is to achieve a dimensionality reduction of the I/O ANN training space by using as input vector the random phase angles of the spectral representation method. A further improvement of the efficiency of the proposed approach is achieved by exploiting the uniform distribution of the random phase angles in the range $[0, 2\pi]$, by efficiently sampling this uniform training space using a LHS technique. The presented test examples demonstrated the efficiency of an ANN predictions of the stochastic quantities in the framework of finite element analysis based on Monte Carlo simulation. Acceptable results were achieved at a fraction of computing time compared to the standard Monte Carlo simulation with finite element analyses. Consequently, this method seems to be an efficient tool towards the direction of accurately assessing the effect of a random spread of design quantities in real world structures at an affordable computational cost. We can thus summarize that neural networks offer an alternative way of performing stochastic analyses of structural systems especially for problems where conventional methods cannot be efficiently applied.

9

An adaptive Galerkin SSFEM approach

In this chapter, a methodology is developed (58) in order to construct an adaptive sparse polynomial chaos (PC) expansion of the response of stochastic systems whose input parameters are modeled with random fields. The proposed methodology utilizes the concept of VRF in order to compute an a priori low cost estimation of the spatial distribution of the second-order error of the response as a function of the number of terms used in the truncated Karhunen-Loève (KL) series representation of the random field involved in the problem. This way the sensitivity of the response variance to the spectral content (correlation structure) of the random input is taken into account through a spatial variation of the truncated KL terms. The criterion for selecting the number of KL terms at different parts of the structure is the uniformity of the spatial distribution of the second order error. This way a significantly reduced number of PC coefficients, with respect to classical PC expansion, is required in order to reach a target second-order error. This variation of the KL terms results in an increase of sparsity of the coefficient matrix of the corresponding augmented linear system of equations leading to an enhancement of the computational efficiency of the SSFEM. The improvement of the computational performance of the solution of the augmented matrices is more pronounced in the case where a block diagonal preconditioner is used.

9.1 Phase 1: Error estimation

(1a) Estimate the variability response function $\mathbf{VRF}(\mathbf{x}, \kappa)$ numerically, using a fast Monte Carlo simulation as described in section 5.1, over a wide range of wave numbers and in different locations x_i $i = 1, 2, ...$ along the domain ($\mathbf{x} = \bigcup x_i$).

(1b) For a target autocorrelation function with corresponding two sided power spec-

trum $S_{ff}(\kappa)$ calculate an "exact" value for the response variance as

$$\mathrm{Var}^T[u(\mathbf{x})] = 2 \cdot \int_0^\infty \mathbf{VRF}(\mathbf{x}, \kappa) \cdot S_{ff}(\kappa) d\kappa \tag{9.1}$$

(1c) Generate zero-mean Gaussian sample functions of the stochastic field using Karhunen-Loève expansion for various number of terms M:

$$f_M(\mathbf{x}) = \sum_{i=1}^{M} \sqrt{\lambda_i} \xi_i \phi_i(\mathbf{x}) \tag{9.2}$$

and estimate the corresponding spectral density function (SDF) of the sample function in an ensemble average sense as follows:

$$S_{ff}^M(\kappa) = \frac{1}{2\pi T} \left| \int_0^T f_M(\mathbf{x}) e^{-i\kappa\mathbf{x}} d\mathbf{x} \right|^2 \tag{9.3}$$

where T is the length of the sample functions of the random fields modeling the uncertainties.

(1d) Calculate the variance of the displacement from:

$$\mathrm{Var}^M[u(\mathbf{x})] = 2 \cdot \int_0^\infty \mathbf{VRF}(\mathbf{x}, \kappa) \cdot S_{ff}^M(\kappa) d\kappa \tag{9.4}$$

(1e) Estimate the error:

$$\mathrm{Error}^M(\mathbf{x}) = \frac{\left| \mathrm{Var}^T[u(\mathbf{x})] - \mathrm{Var}^M[u(\mathbf{x})] \right|}{\mathrm{Var}^T[u(\mathbf{x})]} \cdot 100(\%) \tag{9.5}$$

(1f) Repeat steps $(1c) - (1f)$ with increasing number of terms M until the error of Eq.(9.5) reaches a target value ($e.g. < 10\%$) at all selected points x_i of the domain. In general, a different number M_j will be required at different locations in order to have a uniform error distribution all over the domain. By grouping the x_j's with the same number of KL order M we divide the domain in subdomains. This guarantees a uniform error distribution all over the domain.

9.2 Phase 2: Building a sparse PC coefficient matrix

(2a) For each sub-domain i estimate the number of terms $P_i, \quad i = 1, ..., N$ required in the PC expansion according to the formula $P_i = \sum_{k=0}^{p} \binom{M_i+k-1}{k}$.

(2b) The nodal displacement vector is decomposed as $\mathbf{U} = [\mathbf{U}_1, ...\mathbf{U}_N]^T$ where $\mathbf{U}_i$ are the nodal displacements at each sub-domain. In each section the PC expansion gives

$$\begin{aligned} \mathbf{U}_1 &= \sum_{i=0}^{P_1} u_i \Psi_i \\ \mathbf{U}_2 &= \sum_{j=0}^{P_2} u_j \Psi_j \\ &\vdots \\ \mathbf{U}_N &= \sum_{k=0}^{P_N} u_k \Psi_k \end{aligned}$$

The overlapping nodes are assumed to belong to the domain with the higher M

(2c) Assemble the corresponding sparse PC coefficient matrix and solve the linear system according to eq.(7.23)

The advantage of the proposed methodology is that the sparsity of matrix $\mathfrak{K}$ in eq.(7.23) is significantly increased compared to the classical Galerkin approach. The increase in sparsity of the PC coefficients matrix may be calculated as

$$\sum_{i=1}^{N} (\max(P_1, P_2, ..., P_N) - P_i) \cdot \text{dof}_i \tag{9.6}$$

where dof_i are the number of degrees of freedom in each sub-domain.

9.3 Numerical examples

In order to assess the performances of the proposed methodology, a detailed comparison of the classical full PC representation and the proposed sparse one, in terms of by-products of interest, namely the first two statistical moments is performed. Two test examples are considered, one based on a 3D cantilever model and one based on a

plain stress model. In both cases the elastic modulus is assumed to vary according to the formula,

$$E(x,\theta) = E_0 \cdot \Big(1 + f(\mathbf{x},\theta)\Big) \tag{9.7}$$

where E_0 is the mean value, $f(\mathbf{x},\theta)$ is assumed to be a zero-mean homogeneous Gaussian stochastic field in the first case which varies along its length, while in the second example it follows a log-normal distribution, which varies spatially in both directions. The mean values and standard deviations as well as the autocorrelation functions (ACF) and their corresponding SDFs for the two test cases are reported in Tables 9.1 and 9.2, respectively.

Test Case	Distribution	Standard deviation
3D Cantilever	Gaussian	0.1
2D Plate	Log-normal	0.5

Table 9.1: Test cases - input random variable

Test Case	SDF	ACF
3D Cantilever	$\sigma^2 \frac{2b}{\pi\cdot(1+b^2\kappa^2)}$	$\sigma^2 e^{\frac{\lvert\tau\rvert}{b}}$
2D Plate	$\sigma^2 \frac{2b}{\pi\cdot(1+b^2\kappa^2)}$	$\sigma^2 e^{\frac{\lvert\tau\rvert}{b}}$

Table 9.2: Test cases - Correlation structures

In Table 9.2, σ is the standard deviation of the stochastic field and b a correlation length parameter.

9.3.1 3D Cantilever

Consider the 3D cantilever structure of Fig. 9.1 with length $L = 10\ m$ and section $A = 1 \times 1\ m^2$, loaded with a uniform load $Q = 1000\ N/m$, partially distributed along its length, and a concentrated moment $M = 10000\ N \cdot m$ at its right end, applied as equivalent nodal forces at the nodes of the two extreme edges of the last section at $x = 10\ m$. Both loads are assumed to be deterministic. The cantilever is descritized with 450 HEXA finite elements resulting in a problem with 5412 dofs.

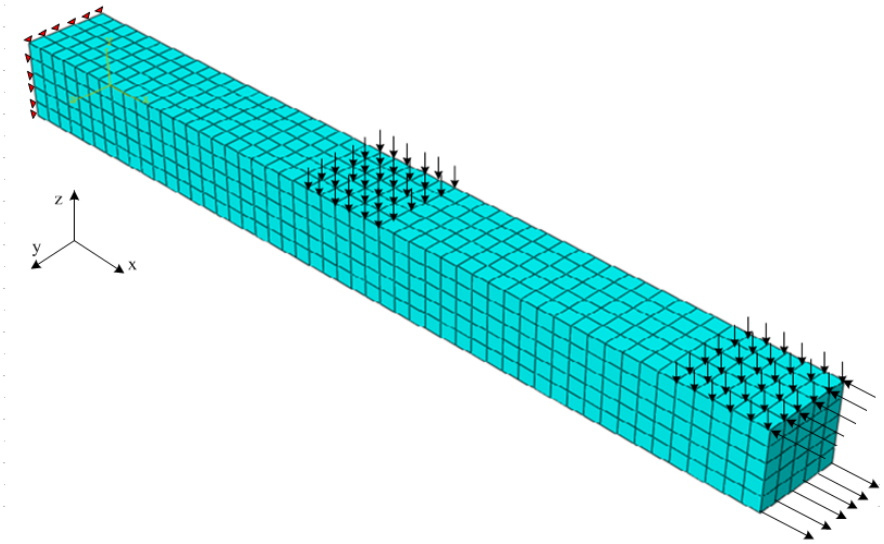


Figure 9.1: Finite element mesh (HEXA)

The influence of the scale of correlation on the quality of the simulations is of a great importance but is not under investigation in this work, The value $b = 10m$ is selected for this example. The variability response function $\mathbf{VRF}(\mathbf{x}, \kappa)$ is estimated numerically at x_j $j = 1, ..., 30$ locations along the x-axis (Fig.9.2), in the range $\kappa \in [0, \kappa_u]$ with a step of 0.02, κ_u being an upper cut-off frequency, taken equal to 0.2. Indicative plots of calculated VRFs in location $x_1 = 1\ m$, $x_2 = 3\ m$ and $x_3 = 10\ m$ are presented in Fig.9.3

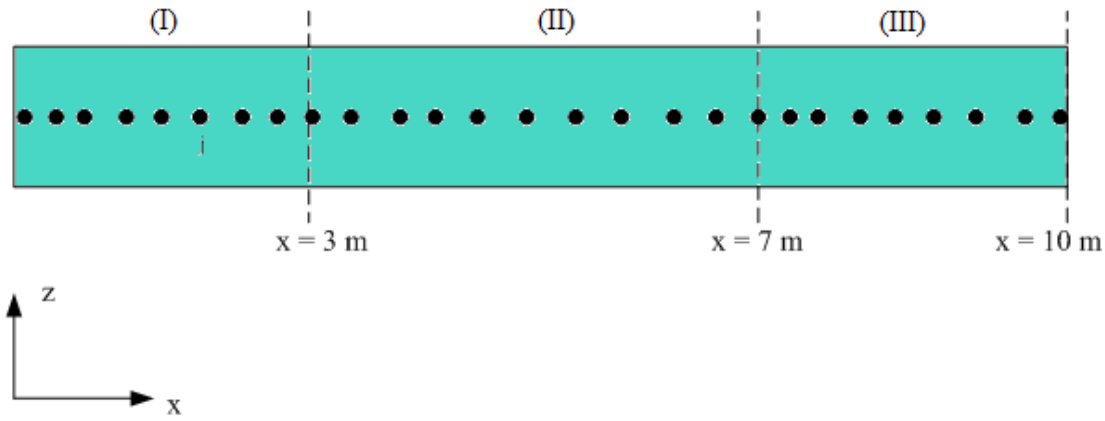


Figure 9.2: Estimation of **VRF** in three locations along the beam

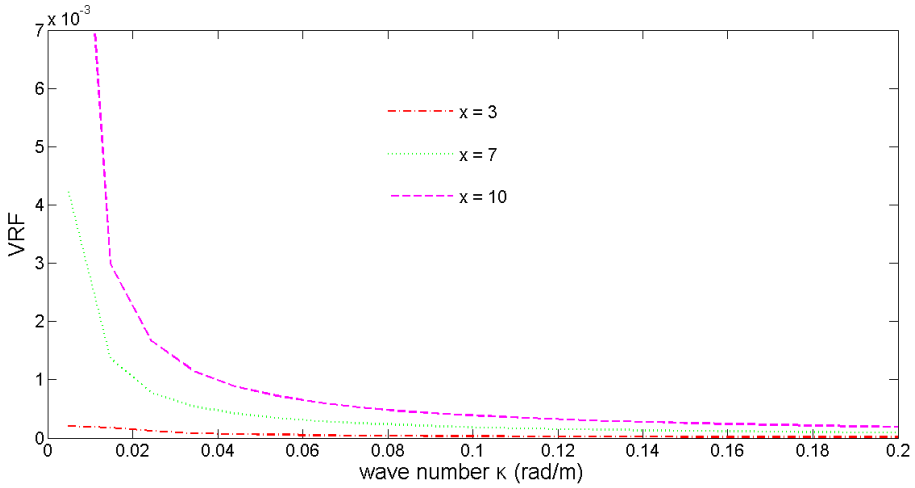


Figure 9.3: VRF at points x_j, j=1,2,3

Figure 9.4 presents the plots of S_{ff}^{M} for various values of M computed via eq.(9.3), together with the target SDF. Inspection of this figure, keeping in mind that the variance is given by the integral form expression of eq.(7.9), reveals that the error of the variance depends not only on the number M but also on the values of the **VRF** at the frequencies that are not well represented in the power spectral density due to the KL truncation. For example, as shown in Fig. 9.5, the error for $x_1 = 3\ m$ and $M = 5$ is expected to be much smaller than the same error for $x_2 = 7\ m$ or $x_3 = 10\ m$ since, at the latest location the **VRF** has significant contributions at frequencies larger than the frequency at which S_{ff}^{5} becomes zero ($\kappa_u^5 = 0.05$), while for $x_1 = 3\ m$ and $\kappa > 0.05$, **VRF** becomes almost zero. Indeed, the corresponding errors computed via eq.(9.5) are found to be 17%, 20% and 31% respectively.

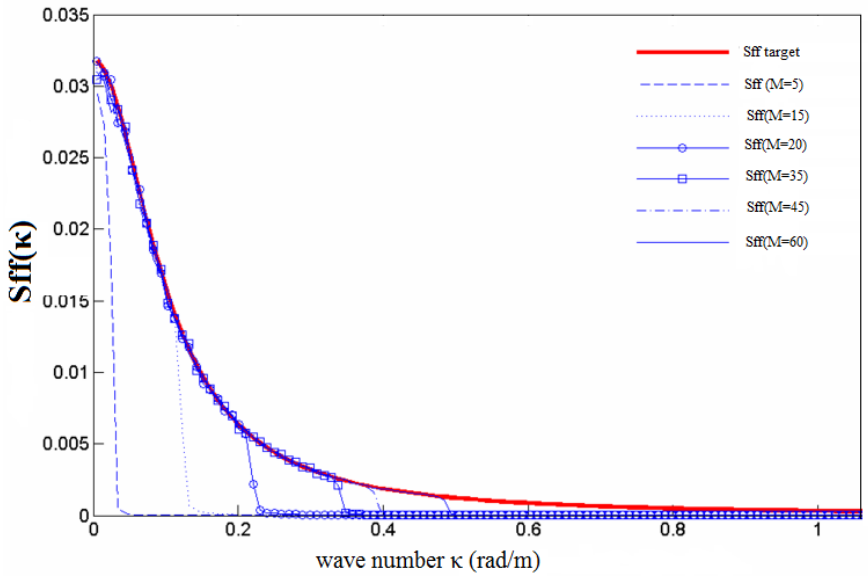


Figure 9.4: SDFs from KL for different M and target SDF

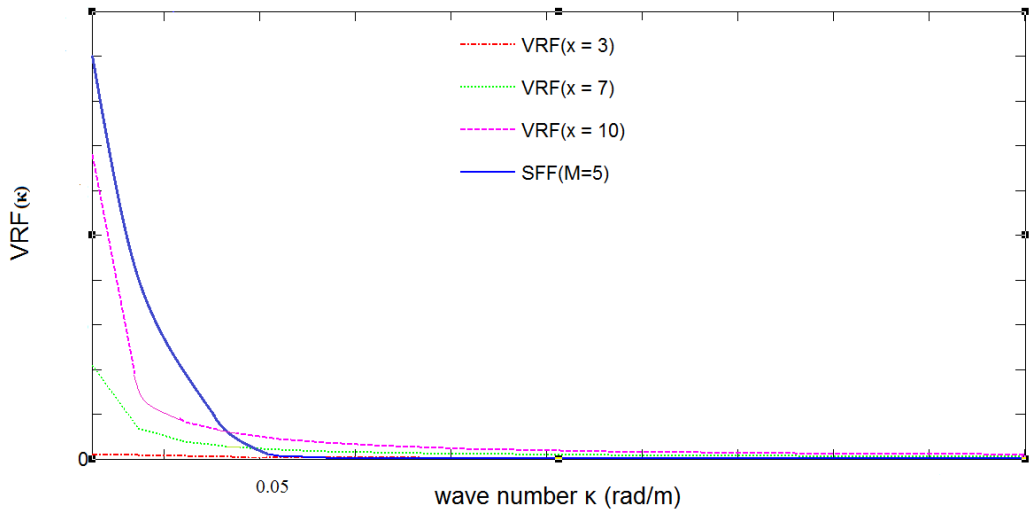


Figure 9.5: Joint plot of S^5_{ff} with **VRF**$(x_1 = 3\ m)$, **VRF**$(x_2 = 7\ m)$, **VRF**$(x_3 = 10\ m)$

This error behavior is presented concisely in Fig. 9.6, which plots the (%) error for the three representative locations x_1, x_2 and x_3 as a function of M. As we can see, the error is much smaller for $x_1 = 3$ and small number of M for the reasons previously mentioned. Figure 9.2 depicts the subdomains with values up to a certain M required to reach a target error less than a fixed value. More specifically, if we require a uniform

error less than 20% then for all the points in subdomain (I) the value of up to M_I = 2 is required, for points inside subdomain (II) we need M_{II}=6, while for all the other points in subdomain (III), M_{III}=10 is satisfying the requirement. Thus, we choose to represent all stochastic degrees of freedom in subdomain I with $M = 2$, in subdomain II with $M = 6$ and $M = 10$ for subdomain III. If we require a target error less than 10% the corresponding values will become $M = 12,\ 14$ and 18 for domains (I), (II) and (III), respectively.

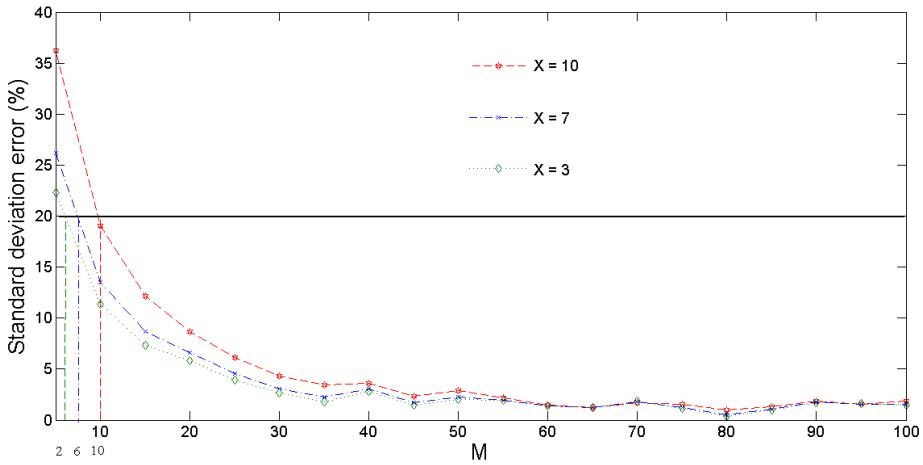


Figure 9.6: Standard deviation error (%) as a function of M

This problem is solved with the classical PC expansion method with M=2, 6 and 10 and p=2, 3 and 4. The z-axis displacement at the tip of the cantilever ($x_3 = 10\ m$), namely U_z, is monitored in order to obtain the results, which are presented in Table 9.3. From this Table it can be seen that the minimum error computed with eq.(9.5) is achieved for M=10 and p=4 as expected and it is 12.38 %.

(M, p)	$U_z\ (m)$	std	$Error^{(M)}(\%)$
(2,2)	-0.1371	0.0612	35.23
(6,2)	-0.1369	0.0698	26.13
(10,2)	-0.1368	0.0817	13.54
(2,3)	-0.1369	0.0639	32.38
(6,3)	-0.1369	0.0729	22.85
(10,3)	-0.1368	0.0825	12.69
(2,4)	-0.1368	0.0675	28.57
(6,4)	-0.1368	0.0754	20.21
(10,4)	-0.1368	0.0828	12.38

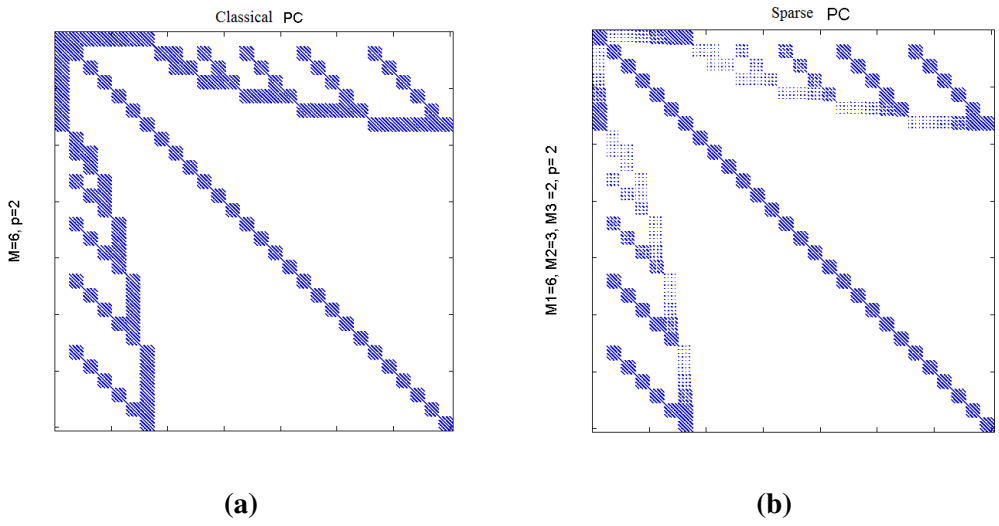
Table 9.3: Standard PC expansion

The same calculation is now repeated using the proposed adaptive sparse PC methodology described in section 9.2. For this calculation the different values of $M = 2$, 6 and 10 are used for sections (I), (II) and (III) respectively (see Fig. 9.2). The corresponding results for $p = 2$, 3 and 4 are presented in Table 9.4 together with the (%) relative error, with respect to the classical PC. This Table also depicts the (%) sparsity increase of the PC coefficient matrix for each case. From these results it can be seen that the same level of accuracy is achieved in all cases with the one of the classical PC using $M = 10$ everywhere. However, this accuracy is reached with a 40% sparsity increase of the coefficient matrix.

(M_1, M_2, M_3, p)	U_z (m)	std	$Error_{\text{rel}}(\%)$	$Sparsity$ $(\%)$
(2,6,10,2)	-0.1370	0.0795	2.69	+38.63
(2,6,10,3)	-0.1369	0.0817	0.96	+41.17
(2,6,10,4)	-0.1368	0.0827	0.12	+43.21

Table 9.4: Adaptive PC expansion

A visual representation of the sparsity increase for this problem is presented in Fig. 9.7.



(a) **(b)**

Figure 9.7: Non-zero elements of the : (a) classical PC and (b) sparse PC

9.3.2 2D plane stress plate

Next, the case of a 2D log-normal random field is investigated in order to show the performance of the method in a case where non-Gaussian random field is involved. The two-dimensional domain of Fig. 9.9 is a rectangle of length $L_x = 1$ m and width

$L_y = 1\ m$. The autocorrelation function of the underlying Gaussian field is selected for this example to be of exponential type. The values $b_x = b_y = 1\ m$ are selected arbitrary for this example. A schematic view of the spectral density function (SDF) of the underlying Gaussian field is given in Fig. 9.8

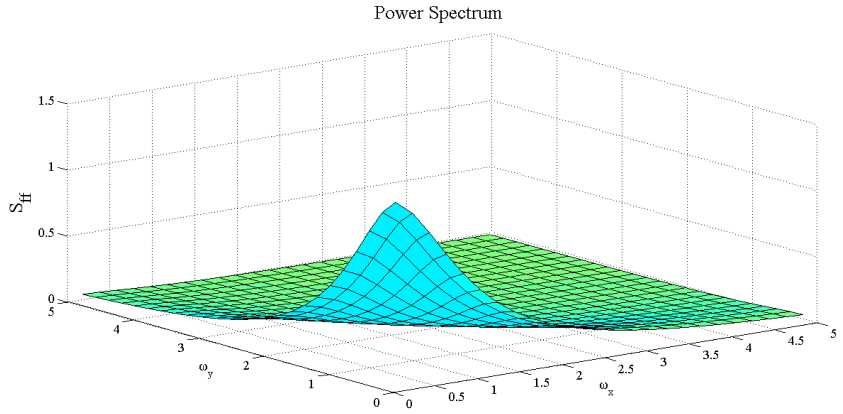


Figure 9.8: Spectral density function of the underlying Gaussian random field

In order to estimate the variability response function of the described model first we generate a number of sinusoid sample functions with fast MC technique according to eq.(7.10). The number required is usually 5-10, which must be added to the total computational cost of the procedure.

The spatial sensitivity of the model's response to the correlation structure of the Young's modulus is identified by estimating the variability response function numerically at 20 different points which can be seen in Fig. 9.9. Plots of the calculated **VRF**s at three representative points, namely x_1, x_2 and x_3, are depicted in Fig. 9.10

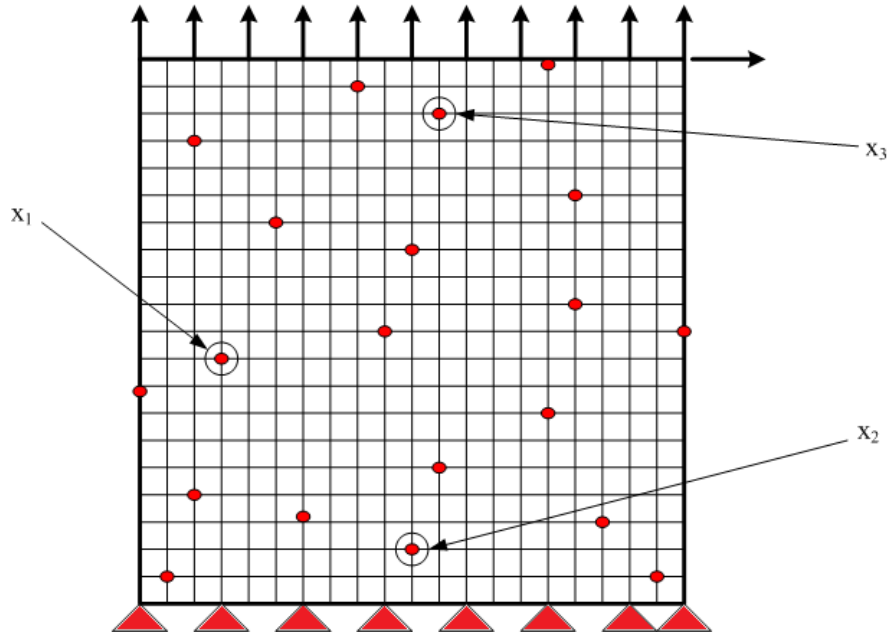


Figure 9.9: Points in which **VRF** is estimated

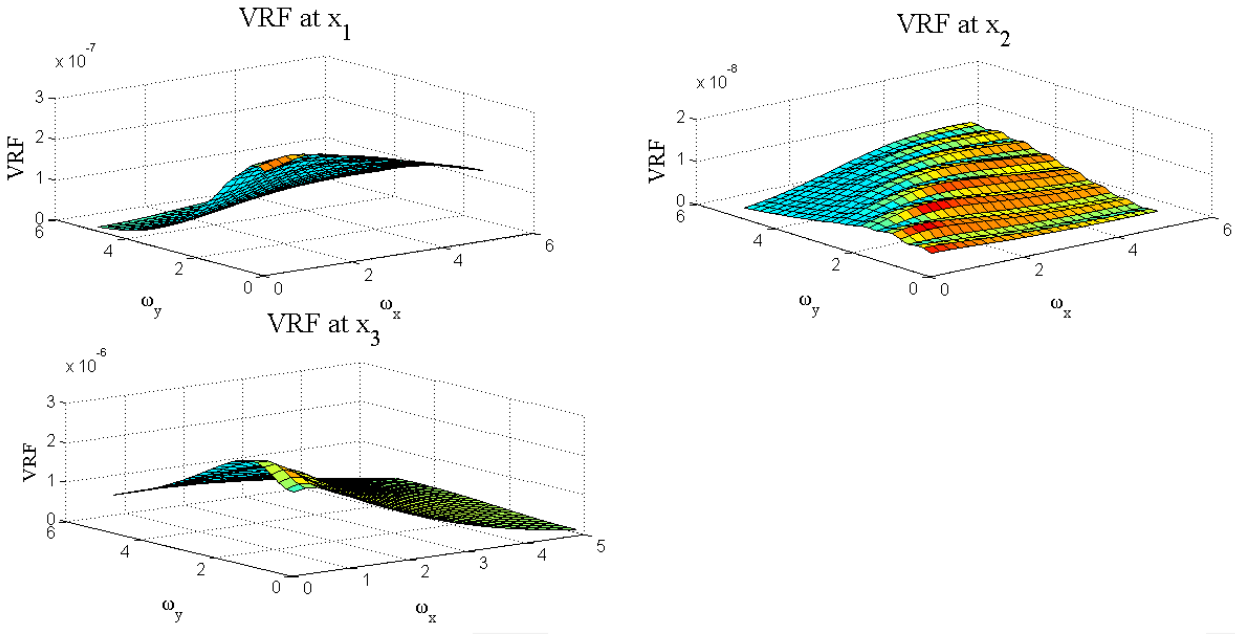


Figure 9.10: Variability response functions at three representative points

After estimating the variability response function in different locations, we calculate the power spectrum of the log-normal filed S_{ff}^{M}. Sample functions of the log-normal

random field are produced using the truncated KL expansion of eq.(7.25) for different values M. The resulting power spectrum of the log-normal field is calculated using eq.(9.3). The standard deviation in each location of the plate is calculated next using the analytic expression of eq.(7.9) and thus the error for all locations of the domain as a function of M is derived. This error behavior is presented concisely in Fig. 9.11, which plots the (%) error for the three representative locations x_1, x_2 and x_3 as a function of M. It must be mentioned that this variance error is computed with respect to the target variance obtained for $M = 10$ in the KL terms.

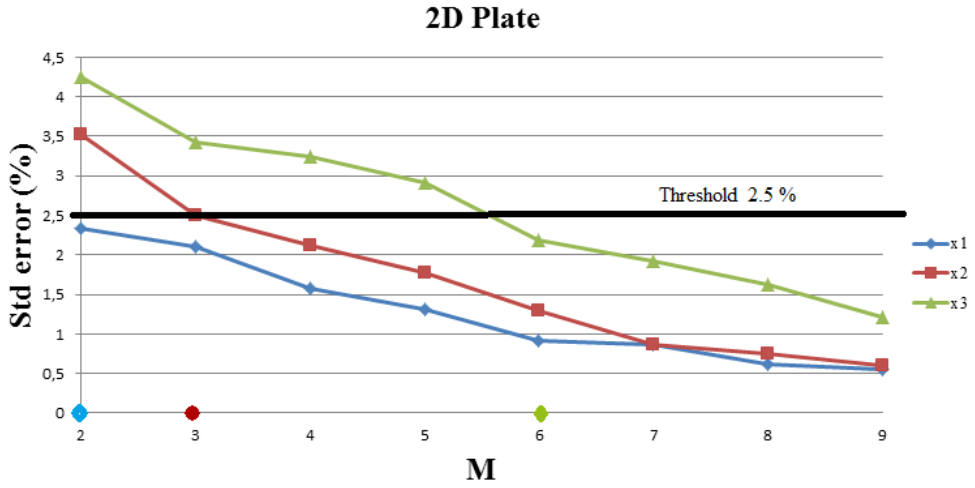


Figure 9.11: Standard deviation error (%) as a function of M in three representative locations x_1, x_2 and x_3

Figure 9.12 depicts the subdomains with values up to a certain M required to reach a target error of about 2.5%. More specifically, for the all the points in subdomain (I) the value of up to $M_I = 3$ is required, for points inside subdomain (II) we need M_{II}=2, while for all the other points in subdomain (III), M_{III}=6 is satisfying the requirement for a uniform error. Thus, we choose to represent all stochastic degrees of freedom in subdomain I with $M = 3$, in subdomain II with $M = 2$ and $M = 6$ for subdomain III.

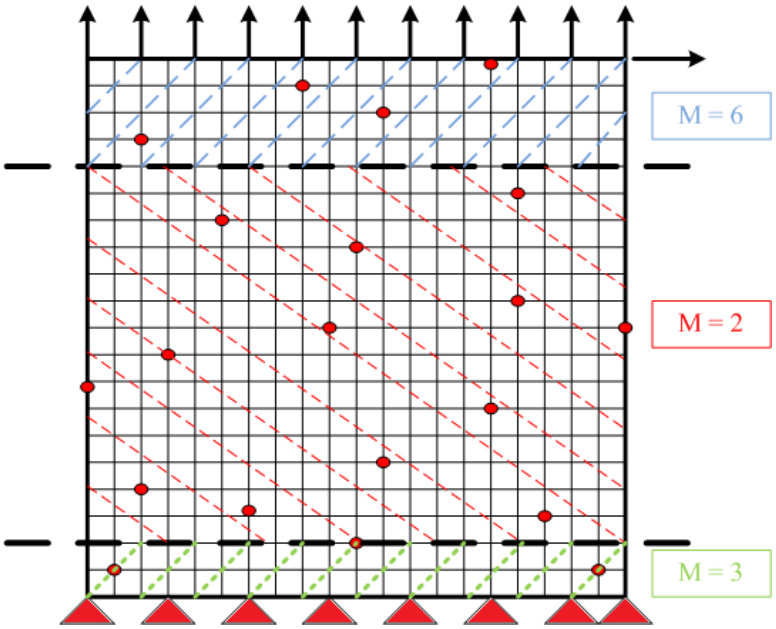


Figure 9.12: Subdomains with equal M values required to reach a target error of about 2%

This problem is initially solved with the classical PC expansion method for M=2, 3 and 6 and p=2, 3 and 4. The x-axis displacement of the node at the upper right corner of the plate namely U_x, is monitored in order to obtain the results, which are presented in Table 9.5.

M	**p**	**mean** $\mathbf{U}_x$ (10^{-4}) (m)	**Std** (10^{-5})
2	2	5.1996	6.7064
3		5.3336	6.8714
6		5.4495	7.0000
2	3	5.2451	6.7578
3		5.4056	6.9473
6		5.4825	6.9956
2	4	5.2385	6.7504
3		5.3894	6.9361
4		5.4553	6.9999

Table 9.5: Standard PC expansion

The same calculation is now repeated using the proposed adaptive sparse PC methodology. For this calculation the different values of M = 2, 3 and 6 are used for sections (I), (II) and (III) respectively (see Fig. 9.12). The corresponding results for p = 2, 3 and 4 are presented in Table 9.6.

$(\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3)$	**p**	**mean** $\mathbf{U}_x$ (10^{-4}) (m)	**Std** (10^{-5})
(2,3,6)	2	5.4093	7.0197
	3	5.5125	7.0125
	4	5.4757	7.0126

Table 9.6: Adaptive PC expansion

Figures 9.13 and 9.14 depict the relative error in the mean and standard deviation of U_x, as a function of p for both approaches, respectively. These errors correspond to the classical PC with $M = 2$ and $M = 3$ and to the sparse PC expansion, with respect to the classical PC with $M = 6$ for all cases. It must be highlighted that for the sparse PC this error is less than 1%.

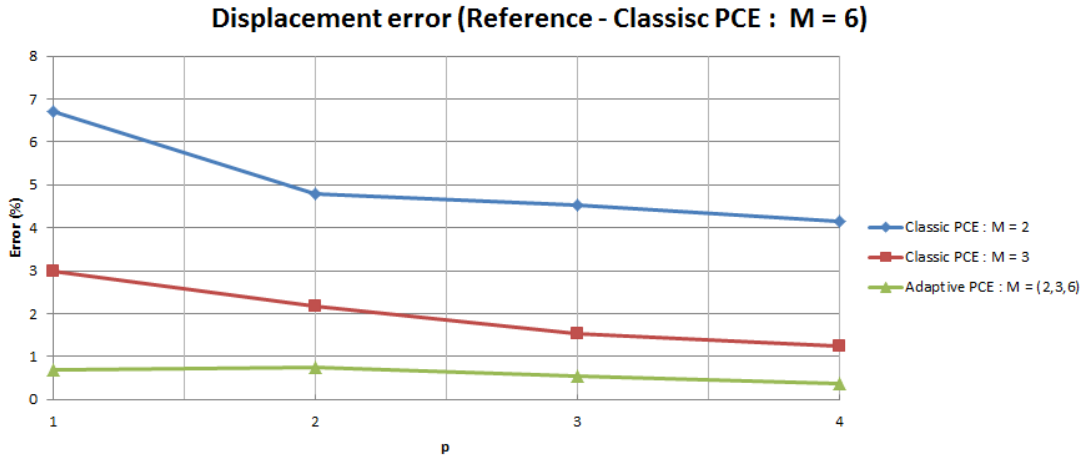


Figure 9.13: Relative error (%) of the mean displacement as a function of p

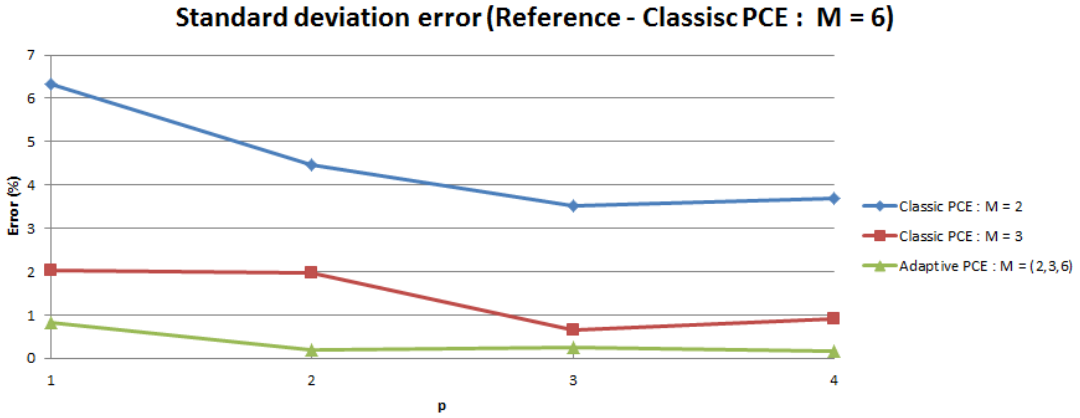


Figure 9.14: Relative error (%) of the standard deviation of U_x as a function of p

The benefits of the aforementioned sparsity increase in the computational efficiency of PC are expected to be significant especially if a preconditioned conjugate gradient (PCG) solver is used with a block diagonal preconditioner. This can be derived from Figs. 9.15 and 9.16 where the sparsity of the augmented matrix is shown as a function of p. From these Figs we can see that an increase of order $40 - 70\%$ can be reached in sparsity, depending on the value of p, with the sparse PC method. The non-zero elements of the augmented matrix are almost one order of magnitude less with the proposed methodology compared to the classic PC with $M = 6$ all over the domain.

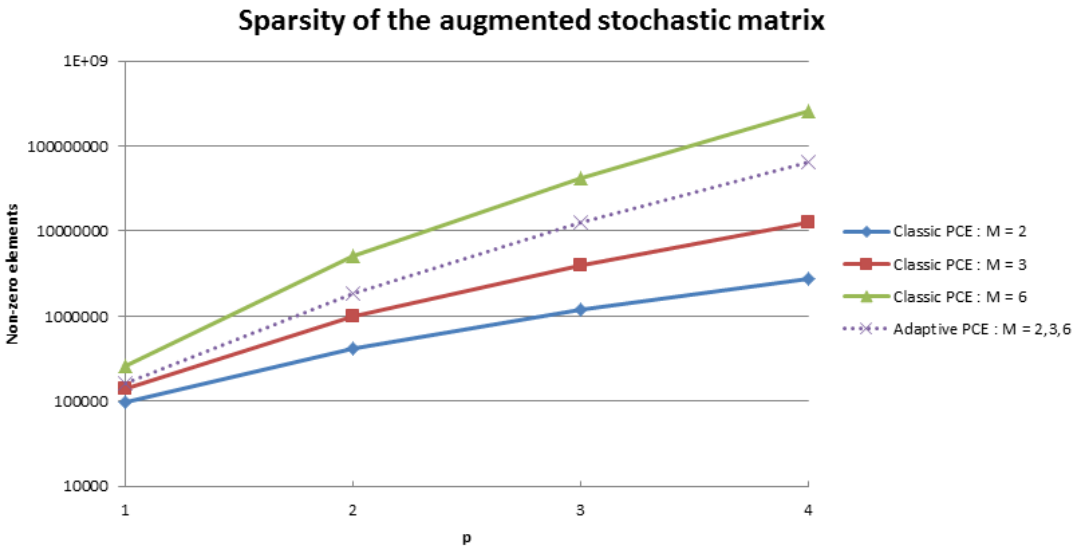


Figure 9.15: Sparsity of the augmented matrix

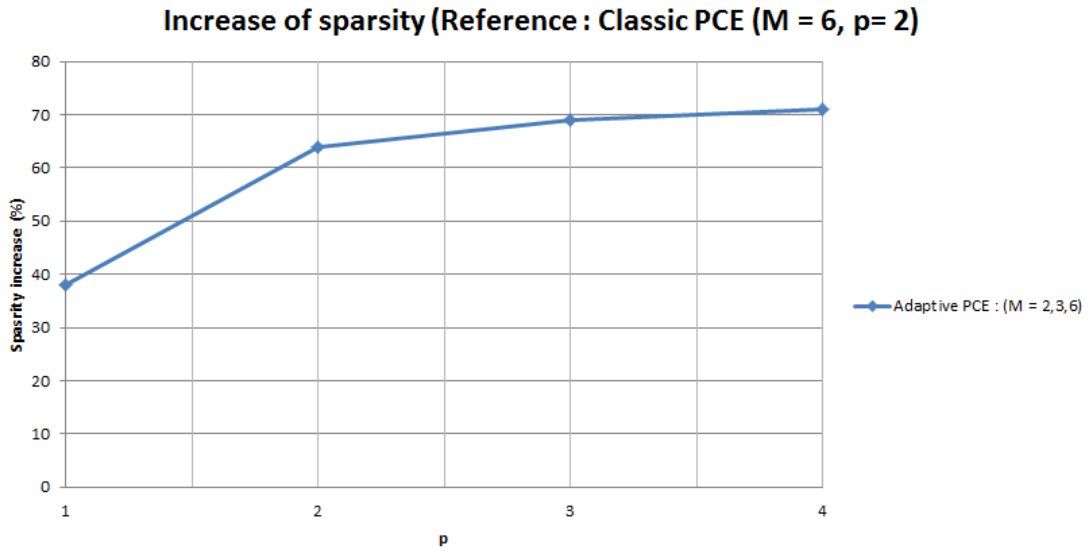
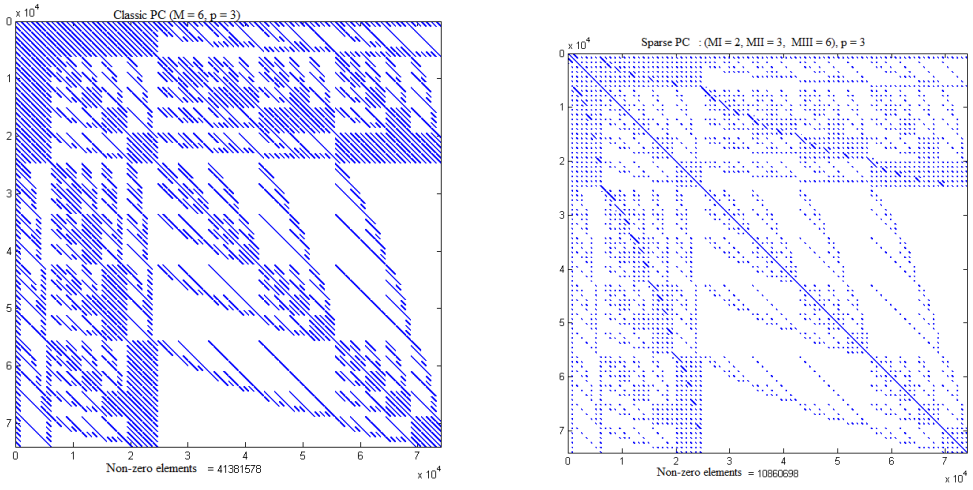


Figure 9.16: Increase of sparsity as a function of p

Finally, plots of the non-zero elements of the augmented matrix is shown in Fig. 9.17 for the classical PC with $M = 6$ and $p = 3$ and for the sparse PC approach with $M_I = 3,\ M_{II} = 2\ M_{III} = 6$ and $p = 3$, respectively.



(a) **(b)**

Figure 9.17: Non-zero elements of the : (a) classical PC and (b) sparse PC

9.4 Solution of the system of stochastic partial differential equations

9.4.1 Monte Carlo methods

When dealing with deterministic external loading, the Monte Carlo simulation procedure for the numerical estimation of the **VRF** features the solution of successive linear systems with multiple left-hand sides, since only the coefficient matrix K changes in every simulation. Due to the fact that the solution process has to start from the beginning, a new stiffness matrix needs to be formed at each simulation. Such solution can be performed either with a standard direct method based on Cholesky factorization or with iterative methods.

In order to alleviate the incapability of direct schemes to exploit the proximity of the resulting systems of equations, iterative solvers have been proposed which are customized to the particular properties of the equilibrium equations arising in the context of MC methods. The resulting near-by problems can be effectively solved using the preconditioned conjugate gradient (PCG), algorithm equipped with a preconditioner following the rationale of incomplete Cholesky preconditioning. This solution procedure consists of utilizing the deterministic K_0 stiffness matrix as its preconditioner throughout the entire simulation process for the solution of the near-by problems.

The PCG algorithm, when solving a linear system of the form $\mathbf{Ax} = \mathbf{b}$ with a preconditioner $\tilde{\mathbf{A}}$, is depicted in Table 1 for iteration k.

- Initialization phase: $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0$ $\mathbf{z}^0 = \tilde{\mathbf{A}}^{-1}\mathbf{r}^0$, $\mathbf{p}^0 = \mathbf{z}^0$, $\mathbf{q}^0 = \mathbf{A}\mathbf{p}^0$, $\eta^0 = \frac{\mathbf{p}^{0^T}\mathbf{r}^0}{\mathbf{p}^{0^T}\mathbf{q}^0}$,
- Repeat for $k = 1, 2...$ until convergence:

Solution estimate		$\mathbf{x}^k = \mathbf{x}^{k-1} + \eta^{k-1}\mathbf{p}^{k-1}$
Residual vector		$\mathbf{r}^k = \mathbf{r}^{k-1} - \eta^{k-1}\mathbf{q}^{k-1}$
Preconditioned residual vector		$\mathbf{z}^k = \tilde{\mathbf{A}}^{-1}\mathbf{r}^k$
Search vector	Using re-orthogonalization	$\mathbf{p}^k = \mathbf{z}^k - \sum_{i=0}^{k-1} \frac{\mathbf{z}^{k^T}\mathbf{q}^i}{\mathbf{p}^{k^T}\mathbf{q}^i}\mathbf{p}^i$
A matrix product vector		$\mathbf{q}^k = \mathbf{A}\mathbf{p}^k$
η estimation	Using re-orthogonalization	$\eta^k = \frac{\mathbf{p}^{k^T}\mathbf{r}^k}{\mathbf{p}^{k^T}\mathbf{q}^k}$

Table 9.7: The PCG algorithm

The PCG algorithm equipped with a preconditioner following the rationale of incomplete Cholesky preconditioning features an error matrix $\mathbf{E}_i$. This matrix is dependent on the discarded elements of the lower triangular matrix produced by the in-

complete Cholesky factorization procedure. Considering the near-by problems of the form,

$$(\mathbf{K}_0 + \Delta\mathbf{K}_i)\mathbf{u}_i = \mathbf{f}\,, \quad i = 1, ..., n_{sim} \tag{9.8}$$

if matrix $\mathbf{E}_i$ is taken as $\Delta\mathbf{K}_i$, the preconditioning matrix becomes the initial matrix $\tilde{\mathbf{A}} = \mathbf{K}_0$. The PCG algorithm equipped with the latter preconditioner throughout the entire solution process constitutes the PCG method for the solution of the n_{sim} near-by problems of eq. (9.8). With the preconditioning matrix $\tilde{\mathbf{A}} = \mathbf{K}_0$ remaining the same during the successive Monte Carlo simulations, the repeated solutions required for the evaluation of the preconditioned residual vector $\mathbf{z}^k = \tilde{\mathbf{A}}^{-1}\mathbf{r}^k$ can be treated as problems with multiple right-hand sides, since this vector needs to be evaluated at each PCG iteration k of each simulation i.

9.4.2 Solution of the augmented SSFEM systems of equations

The augmented systems that are generated when using SSFEM are suitable candidates for iterative solvers since they are flexible enough to be custom tailored to their particular architecture of the augmented systems. A number of solution procedures for solving the augmented linear equations of the SSFEM method has been proposed addressing small to medium problems. However, as the problem size grows, such a solution can become quite challenging due to the enormous memory and computational resources required. Solution techniques are based on either Gauss-Jacobi or preconditioned conjugate gradient (PCG) iterative solvers for addressing this problem.

The augmented systems that are generated from the application of SSFEM involve large coefficient matrices that feature a block form. When dealing with Gaussian stochastic fields, each block of the diagonal is comprised of the deterministic stiffness matrix $\mathbf{K}_0$ scaled by an integer. Consider the preconditioning matrix for the case of Gaussian distribution of the form

$$\tilde{\mathbf{A}} = \begin{bmatrix} a_1\mathbf{K}_0 & 0 & \cdots & 0 \\ 0 & a_2\mathbf{K}_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_n\mathbf{K}_0 \end{bmatrix} \tag{9.9}$$

where a_n are the coefficients as calculated from the PC bases. For each evaluation of the preconditioned residual vector, the same $\mathbf{K}_0$ matrix needs to be "inverted" n times, as in the case of the MC-PCG-Skyline method. This matrix "inversion" is implemented as the solution of n linear systems. Since matrix $\tilde{\mathbf{A}}$ is block diagonal, the solution process can be pipelined as the successive solution of n linear systems with multiple right-hand sides. In log-normal stochastic fields, each block of the diagonal is comprised either of the deterministic stiffness matrix $\mathbf{K}_0$ scaled by an integer or of a linear combination

of the deterministic stiffness matrix $\mathbf{K}_0$ and stochastic matrices $\mathbf{K}_1$ to $\mathbf{K}_n$. Matrices $\mathbf{K}_1$ to $\mathbf{K}_n$ may differ up to orders of magnitude when compared to $\mathbf{K}_0$, so matrix $\boldsymbol{K}$ is also considered block-dominant.

9.5 Computational merits of the proposed methodology

The enhancement of the computational performance of the proposed approach due to the aforementioned increase in the sparsity of the coefficient matrix is demonstrated in Table 9.8. In the computational cost of the proposed approach however, the cost required for the calculation of the VRF must also be considered. Since this calculation is used in order to obtain an a priori rough error estimation, the **VRF** can be computed with a relatively sparse discretization in the frequency domain. Therefore, a value of $N = 5$ for the number of points in the wave domain is expected to be sufficient for the purposes of this estimation, leading to an additional cost of 50-100 MCS, which of course should be added to the cost of the adaptive sparse PC solution.

In this Table it can be seen that a 75% reduction of the total computing time was achieved with the sparse SSFEM approach compared to the full SSFEM method, even with the addition of the extra 100 MCS required for the calculation of the **VRF**. The sparse SSFEM solution was achieved with 32 PCG iterations instead of 40 required for the full problem.

	Factorization	**Direct solution** (1 MCS)	**SSFEM**	
			Full (40 iter.)	**Sparse** (32 iter. + 100 MCS)
Computing time (sec)	0.021	0.0004871	23.437	6.137

Table 9.8: Computational performance of classical SSFEM (Full problem) vs adaptive SSFEM (Sparse problem)

It should be noted here that the computational efficiency of the proposed approach is expected to be more pronounced in systems with a large number of dofs as well as in cases in which a sensitivity analysis is required, with respect to various probabilistic parameters describing the random fields involved, in which the **VRF** remains the same, since it depends only on deterministic parameters of the system.

Conclusions

In this chapter a methodology is described to construct an adaptive sparse polynomial chaos (PC) expansion of the response of stochastic systems whose input parameters are modeled with random fields. The proposed methodology utilizes the concept of variability response function (**VRF**) in order to compute an a priori inexpensive estimation of the spatial distribution of the second-order error of the response as a function of the number of terms used in the truncated KL series representation of the random fields involved in the problem. As a result a spatial adaptation of the number of terms used for describing the random field is achieved in order to obtain a uniform error distribution, leading to a significant reduction of the number of PC coefficients as and a corresponding increase in the sparsity of the corresponding deterministic matrix.

The benefits of the aforementioned sparsity increase in the computational efficiency of PC are significant especially if a preconditioned conjugate gradient (PCG) solver is used with a block diagonal preconditioner . In the computational cost of the proposed approach however, the cost required for the calculation of the VRF must also be considered. Since this calculation is used in order to obtain an a priori rough error estimation, the VRF can be computed with a relatively sparse discretization in the frequency domain. Therefore, a value of $N = 5$ in is expected to be sufficient for the purposes of this estimation, leading to an additional cost of 50-100 MCS, which of course should be added to the cost of the adaptive sparse PC solution. The computational efficiency of the proposed approach will be more pronounced in systems with a large number of dofs as well as in cases in which a sensitivity analysis is required, with respect to various probabilistic parameters describing the random fields involved, in which the VRF remains the same, since it depends only on deterministic parameters of the system.

10

Summary - Innovation of thesis

This thesis presented mathematical and numerical concepts regarding the modeling of uncertain quantities in structural systems and provided a detailed review of methods for intrusive and non-intrusive finite element structural reliability assessment. A summary of the main outcomes of this work throughout its chapters is given bellow.

Chapter 2 first provided an introduction to the basics of the theory of probability. The second part of the chapter focused on random variables and reviewed its second order properties, discussed about stationarity and ergodicity and presented some basic distribution models. The final part of this chapter gave an overview on the representation of quantities that vary in time (processes) or space (fields) by introducing the mathematical notion of stochastic processes (or random fields).The cases of Gaussian and non-Gaussian random processes were described in detail.For the case of Gaussian random fields the Spectral representation method and the Karhunen-Loève expansion were presented analytically while for the non-Gaussian random fields the translation theory was presented.

Chapter 3 introduced the finite element method

Chapter 4 introduced the use of neural networks in the framework of structural engineering and mainly in reliability analysis. First, a review of its basic principles and the engineering approach is provided, followed by a discreption of the architectures of neural network. Subsequently, two major technique categories were described for training a neural network, namely the Global adaptive techniques and the Local adaptive techniques. For the last category two methods were presented, the Quickpror method and the Rprop method.

In chapter 5 of the thesis the simulation methods used in the context of reliability analysis were described. In the first part of the chapter the fundamental concepts of reliability analysis were discussed. The second part of the chapter provided an exhausting review of simulation methods with a focus on Monte Carlo simulation, Latin hypercube and Subset simulation.

10. SUMMARY - INNOVATION OF THESIS

In chapter 6 a new methodology for computing the probability of structural failure by incorporating ANN in the framework of the SS method was presented. The basic concept of this method is to progressively train the ANN at each SS level in a sequence of moving ranges (windows) defined by the upper and lower bounds of the samples generated by the Markov Chains of the modified Metropolis algorithm. Two alternative training strategies are implemented for the ANN training procedure and they were described analytically in this chapter. Numerical examples demonstrated the efficiency of the proposed methodology.

In chapter 7 of the thesis, a description of the stochastic finite element method (SFEM) was provided. Monte Carlo simulation and its various forms, depending on the method used for the representation of the random field (Spectral representation and Karhunen-Loève expansion), was presented. Next, the variability response functions were introduced and a discussion on their implementation was made based on results reported in the literature. Subsequently, the description of the general formulation of the spectral stochastic finite element method (SSFEM) was performed for both Gaussian and non-Gaussian random fields. Finally, a comparison of the Monte Carlo method and SSFEM in large-scale structural systems was made. The comparison was based on the computation of the second order moments of the response field as well as on a reliability analysis. In order to set an objective basis for assessing the performance of these methods, a parametric study was conducted, regarding different values for standard deviation σ_E and correlation length b. A numerical test was presented demonstrating the efficiency of the proposed implementations on a problem with different stochastic characteristics. Especially, for the computation of the second moments of the response field, the following procedure was followed: A series of Monte Carlo analyses was carried out with a fixed value of M, in order to estimate the necessary number of simulations for a convergence error of less than 1% for each value of σ_E and b examined. Assuming that the convergence behavior remained invariant for increasing M, another series of Monte Carlo analyses was carried out, in order to estimate the appropriate order of the KL expansion for a prescribed convergence error. The same procedure was carried out next performing SSFEM analyses, in order to estimate the appropriate order of the PC expansion required for convergence to the corresponding MC results. Finally, a reliability analysis was performed, in which the order of the PC expansion was being modified in order to reach a prescribed convergence error in the estimation of the probability of failure, compared to the corresponding MC results.

In chapter 8 a novel idea was presented for computing the response variability of stochastic structural systems, whose material properties are described as random fields, by incorporating ANN in the framework of MC simulation. The basic idea of the proposed methodology is to reduce the size of the input data of the ANN configuration and facilitate its training for predicting the response of a system in the framework of finite element based stochastic structural analysis. Thus, the numerical model can be accurately and efficiently substituted by the trained ANN to predict the stochastic

response with significantly less computational effort than the conventional MC simulation. The methodology developed is based on the multi–dimensional spectral representation method for the modeling of the involved random quantities. The proposed methodology was applied to Gaussian and as well as to non-Gaussian random fields. Three numerical examples, concerning 2D and 3D dimensional structures and also Gaussian and non-Gaussian random fields demonstrated the efficiency of this method.

Chapter 9 presented also a new methodology for constructing an adaptive sparse polynomial chaos (PC) expansion of the response of stochastic systems whose input parameters are modeled with random fields. The proposed methodology utilizes the concept of VRF in order to compute an a priori low cost estimation of the spatial distribution of the second-order error of the response as a function of the number of terms used in the truncated Karhunen-Loève (KL) series representation of the random field involved in the problem. This way the sensitivity of the response variance to the spectral content (correlation structure) of the random input is taken into account through a spatial variation of the truncated KL terms. The criterion for selecting the number of KL terms at different parts of the structure is the uniformity of the spatial distribution of the second order error. The methodology was also presented next through numerical examples.

10. SUMMARY - INNOVATION OF THESIS

Bibliography

[1] S. Adhikari. **A reduced spectral function approach for the stochastic finite element analysis**. *Computer Methods in Applied Mechanics and Engineering*, **200(21–22)**:1804–1821, 2011. 4

[2] I. Aleksander and H. Morton. *Neurons and Symbols, the Stuff that Mind is Made of*. 1993. 43

[3] I. Aleksander and H. Morton. *An Introduction to Neural Computing*. 1995. 43

[4] D.J. Amit. *Modeling Brain Function, the World of Attractor Neural Networks*. 1989. 43

[5] M. Anders and M. Hori. **Stochastic finite element method for elasto-plastic body**. *Int. J. Num. Meth. Eng.*, **46(11)**:1897–1916, 1999. 90

[6] M. Anders and M. Hori. **Three-dimensional stochastic finite element method for elasto-plasticbody**. *Int. J. Num. Meth. Eng.*, **51**:449–478, 2001. 3, 90

[7] J.A. Anderson. *An Introduction to Neural Networks*. 1995. 43

[8] J.H. Argyris. **Energy theorems and structural analysis-Part 1 (Oct.-ŰNov.)**. *Aircraft Engineering*, **26**, 1954. 29

[9] J.H. Argyris. **Energy theorems and structural analysis-Part 2 (Feb.–May)**. *Aircraft Engineering*, **27**, 1955. 29

[10] SR. Arwade and G. Deodatis. **Variability response functions for effective material properties.** *Probabilistic Engineering Mechanics*, **26**:174–181, 2010. 4, 88

[11] K.E. Atkinson. **The Numerical Solution of Integral Equations of the Second Kind**. *Cambridge, UK: Cambridge press*, 1997. 26

[12] S.K. Au and J.L. Beck. **A new adaptive importance sampling scheme.** *Struct. Saf.*, **21**:135–158, 1999. 3, 61, 65

[13] S.K. Au and J.L. Beck. **Estimation of small failure probabilities in high dimensions by subset simulation.** *Probab. Eng. Mech.*, **16 (4)**:263–277, 2001. 3, 61, 63, 65

[14] S.K. Au, C. Papadimitriou, and J.L. Beck. **Reliability of uncertain dynamical systems with multiple design points**. *Struct. Saf.*, **21**:113–133, 1999. 3

[15] I. Babuska and P. Chatzipantelidis. **On solving elliptic stochastic partial differential equations**. *Comp. Meth. Appl. Mech. Eng.*, **191 (37–38)**:4093–4122, 2002. 90

[16] R. Beale and T. Jackson. *Neural Computing, an Introduction*. 1990. 43

[17] M. Berveiller, B. Sudret, and M. Lemaire. **Stochastic finite element: a non-intrusive approach by regression.** *Eur. J. Comput. Mech.*, **15**:81–92, 2006. 1

[18] G. Blatman and B Sudret. **An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis**. *Prob. Eng. Mech.*, **25(2)**:183–197, 2010. 4

[19] G. Blatman and B Sudret. **Adaptive sparse polynomial chaos expansion based on least angle regression**. *Journal of Computational Physics.*, **230(6)**:2345–2367, 2011. 4

[20] P. Bocchini and G. Deodatis. **Critical review and latest developments of a class of simulation algorithms for strongly non-Gaussian random fields**. *Probabilistic Engineering Mechanics*, **23**:393–407, 2008. 22

[21] N.K. Bose and P. Liang. *Neural Networks Foundamentals with Graphs, Algorithms and Applications*. 1996. 43

[22] C.G. Bucher. **Adaptive sampling – an iterative fast Monte Carlo procedure**. *Struct. Saf.*, **5(2)**:119–126, 1988. 3

[23] C.G. Bucher and T. Most. **A comparison of approximate response functions in structural reliability analysis.** *Probab. Engrg. Mech.*, **23 (2–3)**:154–163, 2008. 3

[24] C.G. Bucher and P.D. Spanos. **Adaptive sampling, an iterative fast Monte Carlo procedure.** *Struct. Saf.*, **5**:119–126, 1995. 3

[25] R. Callan. *The Essence of Neural Networks*. 1999. 43

[26] D.C. Charmpis and M. Papadrakakis. **Improving the computational efficiency in finite element analysis of shells with uncertain properties.** *Comput. Methods Appl. Mech. Engrg.*, **194**:1447–1478, 2005. 2

[27] D.C. Charmpis, M. Papadrakakis, and G.S. Stefanou. **A domain decomposition solution for the stochastic finite element analysis of shells.** In *H.A. Mang, F.G. Rammerstorfer, J. Eberhardsteiner (Eds.), Proceedings of the Fifth World Congress on Computational Mechanics (WCCM V)*, Vienna University of Technology, Austria, 2002. 2

[28] N.Z. Chen and C. Guedes Soares. **Spectral stochastic finite element analysis for laminated composite plates**. *Comput. Methods Appl. Mech. Engrg.*, **197**:4830–4839, 2008. 3

[29] J. Cheng and Q.S. Li. **Reliability analysis of structures using artificial neural network based genetic algorithms.** *Comput. Methods Appl. Mech. Engrg.*, **197 (45–48)**:3742–3750, 2008. 3

[30] M. Chester. *Neural Networks, a Tutorial*. 1993. 43

[31] D.B. Chung, M.A. Gutierrez, and R. de Borst. **Object-oriented stochastic finite element analysis of fibre metal laminates**. *Comput. Methods Appl. Mech. Engrg.*, **194**:1427–1446, 2005. 3

[32] D.B. Chung, M.A. Guttierez, LL Graham-Brady, and Lingen F-J. **Efficient Numerical Strategies for spectral stochastic finite element models.** *Int. J. Numer. Meth. Engng.*, **64**:1334–1349, 2005. 3

[33] C.A. Cornell. **A normative second-moment reliability theory for structural design.** *Solid Mechanics Division,University of Waterloo, Waterloo, Ontario, Canada*, 1969. 58

[34] J.E. Dayhoff. *Neural Network Architectures, an Introduction*. 1990. 43

[35] M. Deb, I. Babuska, and M. J. Oden. **Solution of stochastic partial differential equations using Galerkin finite element techniques**. *Comp. Meth. Appl. Mech. Eng.*, **190 (48)**:6359–6372, 2001. 90

[36] J. Deng. **Structural reliability analysis for implicit performance function using radial basis function network.** *International Journal of Solids and Structures*, **43**:3255–3291, 2006. 3

[37] G. Deodatis. **The weighted integral method, I : Stochastic stiffness matrix**. *J. Eng. Mech*, **117**:1851–1864, 1991. 86

[38] G. Deodatis, L. Graham, and R. Micaletti. **A hierarchy of upper bounds on the response of stochastic systems with large variation of their properties: random variable case.** *Prob. Engrg. Mech.*, **18 (4)**:349–364, 2003. 4, 88

[39] G. Deodatis and R. Micaletti. **Simulation of highly skewed non-Gaussian stochastic processes**. *Engineering Mechanics*, **127**:1284–1295, 2001. 22

[40] G. Deodatis and M. Shinozuka. **Bounds of response variability of stochastic systems.** *J. Engrg. Mech.*, **115 (1)**:2543–2563, 1989. 88

[41] G. Deodatis and M. Shinozuka. **Stochastic Finite Element Analysis of seismic soil-structure interaction**. *The weighted integral method, II : response variability and reliability*, **117(8)**:1865–1877, 1991. 86

[42] A. Der Kiureghian and T. Dakessian. **Multiple design points in first and second order reliability.** *Struct. Saf.*, **20**:37–49, 1998. 26

[43] A. Der Kiureghian and J.B. Ke. **The stochastic finite element method in structural reliability**. *Proc 11th IFIP WG7.5 Conference*, **3(2)**:83–91, 1988. 86

[44] A. Der Kiureghian and R.L. Taylor. **Numerical methods in structural reliability**. *Proc. 4th Int. Conf. on Appl. of Statistics and Probability in Soil and Structural Engineering*, pages 769–784, 1983. 86

[45] C. Desceliers, R. Ghanem, and C. Soize. **Polynomial chaos representation of a stochastic preconditioner**. *Int. J. Numer. Methods Engrg.*, **64 (5)**:618–634, 2005. 3

[46] V. Dotsenko. *An Introduction to the Theory of Spin Glasses and Neural Networks*. 1994. 43

[47] S. Fahlman. **An Empirical Study of Learning Speed in Back-Propagation Networks.** Carnegie Mellon, 1988. 51, 52

[48] D.M. Frangopol and K. Maute. **Advances in Probabilistic Mechanics and Structural Reliability**. *Comput. Struct.*, 2004. 85

[49] P. Fraunfelder, C. Schwab, and R.A. Todor. **Finite elements for elliptic problems with stochastic coefficients**. *Comput. Methods Appl. Mech. Engrg.*, **194**:205–228, 2005. 3

[50] R. Ghanem. **Probabilistic characterization of transport in heterogeneous media**. *Comp. Met. Appl. Mech. Eng.*, **158**:199–220, 1998. 90

[51] R. GHANEM. **Ingredients for a general purpose stochastic finite elements implementation**. *Comp. Meth. Appl. Mech. Eng.*, **168**:19–34, 1999a. 90

[52] R. GHANEM. **Stochastic finite elements with multiple random non-Gaussian properties**. *J. Eng. Mech. ASCE*, **1**:26–40, 1999b. 90, 94

[53] R. GHANEM AND R.M. KRUGER. **Numerical solution of spectral stochastic finite element systems.** *Comput. Methods Appl. Mech. Engrg.*, **129**:289–303, 1996. 2, 3

[54] R. GHANEM AND P.D. SPANOS. **Polynomial chaos in stochastic finite elements**. *J. Appl. Mech. ASME*, **57**:197–202, 1990. 1, 3, 90

[55] R. GHANEM AND P.D. SPANOS. *Stochastic finite elements - A spectral approach*. 1991. 1, 86, 90

[56] D. GHIOCEL AND R. GHANEM. **Stochastic Finite Element Analysis of seismic soil-structure interaction**. *J. Eng. Mech*, **128**:66–77, 2002. 90

[57] D. GHOSH, P. AVERY, AND C. FARHAT. **A method to solve spectral stochastic finite element problems for large-scale systems.** *Int. J. Numer. Meth. Engng.*, **00**:1–6, 2008. 3, 4

[58] D.G. GIOVANIS AND V. PAPADOPOULOS. **Adaptivity of spectral Galerkin stochastic finite elements using variability response functions**. *Under review*, 2013. 4, 127

[59] D.G. GIOVANIS AND V. PAPADOPOULOS. **Neural Network assisted stochastic analysis of structures**. *Under review*, 2013. 107

[60] H.M. GOMES AND A.M. AWRUCH. **Comparison of response surface and neural network with other methods for structural reliability analysis**. *Struct. Saf.*, **26**:49–67, 2004. 3

[61] L. GRAHAM AND G. DEODATIS. **Weighted integral method and variability response functions for stochastic plate bending problems**. *Struct. Saf*, **20**:165–188, 1998. 88

[62] L. GRAHAM, K. GURLEY, AND F. MASTERS. **Non-Gaussian simulation of local material properties based on a moving-window technique**. *Probabilistic Engineering Mechanics*, **18**:223–234, 2003. 22

[63] M. GRIGORIU. **On the spectral representation method in simulation.** *Probabilistic Engineering Mechanics*, **8 (2)**:75–90, 1993. 22

[64] M. GRIGORIU. **Evaluation of Karhunen-Loève , spectral and sampling representations for stochastic processes.** *J. Eng. Mech.*, **132**:179–189, 2006. 25, 86

[65] M. GRIGORIU. **Evaluation of Karhunen-Loève, spectral and sampling representations for stochastic processes.** *J. Engrg. Mech. (ASCE)*, **132**:179–189, 2006. 25, 86

[66] M. GRIGORIU. **Existence and construction of translation models for stationary non-Gaussian processes**. *Probabilistic Engineering Mechanics*, **24**:545–551, 2009. 22

[67] K. GURLEY AND A. KAREEM. **A conditional simulation of non-normal velocity/pressure fields**. *Wind Engineering and Industrial Aerodynamics*, **77–78**:39–51, 1998. 22

[68] K. GURLEY, M. TOGNARELLI, AND A. KAREEM. **Analysis of simulation tools for wind engineering**. *Probabilistic Engineering Mechanics*, **12**:9–31, 1997. 22

[69] A.M. HASOFER AND N.C. LIND. **Exact and invariant second-moment code format.** *J. Engrg.Mechanics*, **100**:111–121, 1974. 59

[70] S. HAYKIN. *Neural Networks: A Comprehensive Foundation, Second Edition*. 1999. 43

[71] J. HERTZ, A. KROGH, AND R.G. PALMER. *Introduction to the Theory of Neural Computation*. 1991. 43

[72] T. HISADA AND S. NAKAGIRI. **Role of stochastic finite element method in structural safety and reliability**. *Proc. 4th Int. Conf. Struct. Safety and Reliability*, **ICOSSARŠ85**:385–395, 1985. 86

[73] S.P. HUANG, S.T. QUEK, AND K.K. PHOON. **Convergence study of the truncated Karhunen-Loève expansion for simulation of stochastic processes.** *Int. J. Numer. Methods Engrg.*, **52**:1029–1043, 2001. 25, 86

[74] E.J. HURTADO AND D.A. ALVAREZ. **Neural-network-based reliability analysis: a comparative study**. *Comput. Methods Appl. Mech. Engrg.*, **191**:113–132, 2001. 3

[75] J.E. HURTADO. **Neural networks in stochastic mechanics**. *Archieves of Computational Methods in Engineering*, **8 (3)**:303–342, 2001. 3

[76] J.E. HURTADO. **Analysis of one-dimensional stochastic finite elements using neural networks.** *Archives of Computational Methods in Engineering.*, **5**:3–30, 2002. 3

[77] J.E. HURTADO AND D.A. ALVAREZ. **Neural network–based reliability analysis: A comparative study.** *Comput. Methods Appl. Mech. Engrg.*, **191 (1–2)**:113–132, 2001. 3

[78] J. J. ZHANG AND B. ELLINGWOOD. **Orthogonal series expansions of random fields in reliability analysis**. *J. Engrg. Mech. (ASCE)*, **120**:2660–2677, 1994. 22

[79] A. KEESE. **A review of recent developments in the numerical solution of stochastic partial differential equations (Stochastic Finite Elements)**. *Working Report 2003-06, Technical University, Braunschweig, Germany*, 2003. 3

[80] A. KEESE AND H. MATTHIES. **Efficient solvers for non linear stochastic problems**. *H. Mang, ,F. Rammerstorfer, , J. Eberhardsteiner (eds)*, **Proc. 5th World Cong. Comp. Mech.**, 2002. 90

[81] A. KEESE AND H.G. MATTHIES. **Hierarchical parallelisation for the solution of stochastic finite element equations**. *Computers and Structures*, **83**:1033–1047, 2005. 3

[82] T. KHANNA. *Foundations of Neural Networks*. 1990. 43

[83] M. KLEIBER. **Computational stochastic mechanics**. *Comp. Meth. Appl. Mech. Eng.*, 1997. 85

[84] M. KLEIBER AND T.D. HIEN. **The stochastic finite element methods - Basic perturbation technique and computational implementation**. *J. Wiley and sons, Chichester*, 1992. 86

[85] A. KOLMOGOROV. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Julius Springer, 1993. 9

[86] N.D. LAGAROS, A.TH. GARAVELAS, AND M. PAPADRAKAKIS. **Innovative seismic design optimization with reliability constraints.** *Comput. Methods Appl. Mech. Engrg.*, **198 (1)**:28–41, 2008. 3

[87] N.D. LAGAROS AND M. PAPADRAKAKIS. **Improving the condition of the jacobian in neural network training.** *Adv. in Engng Soft.*, **35 (1)**:9–25, 2004. 3, 43, 61, 107, 109

[88] N.D. Lagaros and M. Papadrakakis. **Learning improvement of neural networks used in structural optimization.** *Adv. Engrg.Software*, **35**:9–25, 2004. 51

[89] N.D. Lagaros, M. Papadrakakis, M. Fragiadakis, G. Stefanou, and Y. Tsompanakis. **Neural network aided stochastic computations and earthquake engineering.** *Computer-Assisted Mechanics and Engineering Sciences*, **14**:251–257, 2007. 3

[90] C.F. Li, Y.T. Feng, and D.R.J. Owen. **Explicit solution to the stochastic system of linear algebraic equations** $({}_1A_1 + {}_2A_2 + \cdots + {}_mA_m)x = b$. *Comput. Meth. Appl. Mech. Eng.*, **195 (44–47)**:6560–6576, 2006. 3

[91] W.K. Liu, T. Belytschko, and A. Mani. **Probabilistic finite elements for non linear structural dynamics**. *Comp. Meth. App. Mech. Eng.*, **56**:61–86, 1986a. 86

[92] W.K. Liu, T. Belytschko, and A. Mani. **Random field finite elements**. *Int. J. Num. Meth. Eng.*, **23**:1831–1845, 1986b. 86

[93] M. Loeve. *Probability Theory*. 1977. 25, 86

[94] G. M. Miranda and Deodatis. **Generalized Variability Response Functions for Beam Structures with Stochastic Parameters**. *Journal of Engineering Mechanics*, **1**:340, 2012. 4

[95] D.J.C MacKay. **A practical Bayesian framework for backprop networks.** *Neural Comput.*, **4**:448–472, 1992. 43, 52

[96] F. Masters and KR. Gurley. **Non-Gaussian simulation: cumulative distribution function map-based spectral correction**. *Engineering Mechanics*, **129**:1418–1428, 2003. 22

[97] F. Masters, KR. Gurley, and GA. Kopp. **Multivariate stochastic simulation of wind pressure over low-rise structures through linear model interpolation**. *Wind Engineering and Industrial Aerodynamics*, **98**:226–35, 2010. 22

[98] G. Matthies, C. Brenner, C.G. Bucher, and C. Guedes Soares. **Uncertainties in probabilistic numerical analysis of structures and solids - Stochastic finite elements**. *Struct. Safety*, **19**:283–336, 1997. 85

[99] H.G. Matthies and A. Keese. **Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations**. *Comput. Methods Appl. Mech. Engrg.*, **194**:1295–1331, 2005. 3

[100] M.D. McKay, R.J. Beckman, and W.J. Conove. **A comparison of three methods for selecting values of input variables in the analysis of output from a computer code.** *Technometrics.*, **21 (2)**:239–245, 1979. 61, 107, 109

[101] R. E. Melchers. **Importance sampling in structural systems.** *Struct. Saf.*, **6**:3–10, 1989. 3

[102] R.E Melchers. *Structural Reliability Analysis and Prediction (Second Edition).* John Wiley & Sons,Chichester, 1999. 56

[103] E. Mesbahi and Y. Pu. **Application of ANN-based response surface method to prediction of ultimate strength of stiffened panels.** *Journal of Structural Engineering*, **134 (10)**:1649–1656, 2008. 3

[104] N. Metropolis, A.W Rosenbluth, M.N Rosenbluth, and A.H Teller. **Equations of state calculations by fast computing machines.** *J. Chem. Phys.*, **21 (6)**:1087–1092, 1953. 43, 63

[105] Pellissetti M.F. **Parallel processing in structural reliability**. *Structural Engineering and Mechanics*, **32 (1)**:95–126, 2009. 2

[106] M. Minsky and S. Papert. **Perceptrons: An Introduction to Computational Geometry**. *MIT Press, Cambridge, MA*, 1969. 47

[107] M. Miranda and G. Deodatis. **On the response variability of beams with large stochastic variations of system parameters**. 2006. 4, 88

[108] T. Most and C. Bucher. **Probabilistic analysis of concrete cracking using neural networks and random fields**. *Prob. Engrg. Mech.*, **22**:219–229, 2007. 3

[109] B. Mueller and J. Reinhardt. *Neural Networks, an Introduction*. 1991. 43

[110] J. Nie and B.R. Ellingwood. **A new directional simulation method for system reliability. Part II: Application of neural networks**. *Prob. Engrg. Mech.*, **19 (4)**:437–447, 2004. 3

[111] A. Olsson. **On Latin hypercube sampling for structural reliability analysis**. *Struct. Saf.*, **25(1)**:47–68, 2003. 61, 107, 109

[112] H.M. Panayirci. **Efficient solution of Galerkin-based polynomial chaos expansion systems**. *Adv. Engrg. Software*, **41**:1277–1286, 2010. 3

[113] V. Papadopoulos and G. Deodatis. **Response variability of stochastic frame structures using evolutionary field theory**. *Comput. Methods Appl. Mech. Eng.*, **195(9–12)**:1050–1074, 2006. 4, 88

[114] V. Papadopoulos, G. Deodatis, and M. Papadrakakis. **Flexibility-based upper bounds on the response variability of simple beams**. *Comput. Methods Appl. Mech. Eng.*, **194(41–43)**:1385–1404, 2005. 4, 88

[115] V. Papadopoulos, D.G. Giovanis, N.D. Lagaros, and M. Papadrakakis. **Accelerated Subset Simulation with neural networks for reliability analysis**. *Comput. Methods, Appl. Mech. Engrg.*, **223-224**:70–80, 2012. 67

[116] V. Papadopoulos and M. Papadrakakis. **Stochastic finite element-based reliability analysis of space frames.** *Probabilistic Engineering Mechanics*, **13 (1)**:53–65, 1998. 2

[117] V. Papadopoulos, M. Papadrakakis, and G. Deodatis. **Analysis of mean response and response variability of stochastic finite element systems**. *Comput. Methods Appl. Mech. Eng.*, **195(41–43)**:5454–5471, 2006. 4, 88, 89

[118] M. Papadrakakis and G. Babilis. **Solution techniques for the p-version of the adaptive finite element method.** *International Journal for Numerical Methods in Engineering.*, **37**:1413–1431, 1994. 4

[119] M. Papadrakakis, G. Babilis, and P. Braouzi. **Efficiency of refinement procedures for the p–version of the adaptive finite element method**. *Engineering Computations*, **1**:98–118, 1997. 4

[120] M. Papadrakakis, N.D. Lagaros, and Y. Tsompanakis. **Structural optimization using evolution strategies and neural networks.** *Comput. Methods Appl. Mech. Engrg.*, **156**:309–333, 1998. 3

[121] M. Papadrakakis and V. Papadopoulos. **Robust and efficient methods for stochastic finite element analysis using Monte Carlo simulation.** *Comput. Methods Appl. Mech. Engrg.*, **134**:325–340, 1996. 1, 2

[122] M. Papadrakakis, V. Papadopoulos, and N.D. Lagaros. **Structural reliability analysis of elastic-plastic structures using neural networks and Monte Carlo simulation.** *Comp. Meth. Appl. Mech. Eng.*, **136**:145–163, 1996. 3

[123] A. Papoulis and S.U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill series in electrical engineering: Communications and signal processing. Tata McGraw-Hill, 2002. 9

[124] M.F. Pellissetti and R. Ghanem. **Iterative solution of systems of linear equations arising in the context of stochastic finite elements.** *Adv. Engrg. Software*, **31**:607–616, 2000. 2, 3

[125] K.K. Phoon, H.W. Huang, and S.T. 2005 Quek. **Simulation of strongly non-Gaussian processes using Karhunen-Loève expansion**. *Probabilistic Engineering Mechanics*, **20(2)**:188–198, 2005. 22

[126] R. Popescu, G. Deodatis, and J. Prevost. **Simulation of homogenous non-Gaussian stochastic vector fields**. *Probabilistic Engineering Mechanics*, **13**:1–13, 1998. 22

[127] H.J. Pradlwarter and G.I. Shueller. **On advanced Monte Carlo simulation procedures in stochastic structural dynamics.** *International Journal of Non-Linear Mechanics*, **32 (4)**:735–744, 1997. 1

[128] B. Puig, F. Poirion, and C. Soize. **Non-Gaussian simulation using Hermite polynomials expansion: convergences and algorithms.** *Prob. Engrg. Mech.*, **17**:253–264, 2002. 22

[129] M. Riedmiller. **Advanced supervised learning in multi-layer perceptrons from back-propagation to adaptive learning algorithms**. *Int. Journal of Computer Standards and Interfaces*, **16**:265–278, 1994. 43

[130] M. Riedmiller and H. Braun. **A direct adaptive method for faster back-propagation learning: the RPROP algorithm.** pages 586–591, San Francisco, 1993. 3, 51, 52, 68

[131] W. Ritz. **Über eine Neue Metode zur Lösung gewisser Variations probleme der Matematischen Physik**. *J. Reine Angew. Math.*, **135**:1–61, 1909. 29

[132] S. Sakamoto and R. Ghanem. **Polynomial chaos decomposition for the simulation of non-Gaussian nonstationary stochastic processes**. *Engineering Mechanics*, **Proceedings of the ASCE**:128(2):190Ű–201, 2002. 22

[133] W. Schiffmann, M. Joost, and R. Werner. **Optimization of the back-propagation algorithm for training multi-layer perceptrons.** *Technical Report, University of Koblenz, Institute of Physics*, 1993. 51

[134] G. Schueller. **A state-of-the-art report on computational stochastic mechanics**. *Prob. Eng. Mech.,IASSAR report*, 1997. 85

[135] Y. Shi and G. Deodatis. **A novel approach for simulation of non-Gaussian fields: applications in estimating wire strengths from experimental data**. *Safety and reliability of engineering systems and structures. Proceedings of the ninth international conference on structural safety and reliability*, 2005. 22

[136] Y. Shi and R. Deodatis, G.and Betti. **Recent developments in simulation of non- Gaussian processes/fields with application in suspension bridge cable strength estimation**. *Proceedings of the ninth ASCE specialty conference on probabilistic mechanics and structural reliability*, 2004. 22

[137] M.D. Shields, G. Deodatis, and P. Bocchini. **A simple and efficient methodology to approximate a general non-Gaussian stationary stochastic process by a translation process**. *Probabilistic Engineering Mechanics*, **26**:511–519, 2011. 22

[138] M. Shinozuka. **Structural response variability**. *J. Eng. Mech.*, **113(6)**:825–842, 1987. 4, 88, 89

[139] M. Shinozuka and G. Deodatis. **Simulation of stochastic processes by spectral representation**. *Appl. Mech. Rev.*, **44(4)**:191–203, 1991. 22, 86, 89

[140] M. Shinozuka and G. Deodatis. **Simulation of multi-dimensional Gaussian stochastic fields by spectral representation.** *Applied Mechanics Review*, **49**:29–53, 1996. 24, 107

[141] M. Shinozuka and G. Deodatis. **Response variability of SFE systems.** *J. Engrg. Mech. (ASCE)*, **114**:499–519, 1998. 88

[142] P.D. Spanos and R. Ghanem. **Stochastic finite element expansion for random media.** *J. Eng. Mech., ASCE*, **115 (5)**:1035–1053, 1989. 1, 26

[143] G. Stavroulakis, D.G. Giovanis, M. Papadrakakis, and V. Papadopoulos. **A new perspective on the solution of uncertainty quantification and reliability analysis of large-scale problems**. *Comput. Methods Appl. Mech. Engrg.*, **http://dx.doi.org/10.1016/j.cma.2014.03.009**, 2014. In press. 96

[144] G. Stefanou. **The stochastic finite element method: past, present and future.** *Comput.Methods Appl. Mech. Eng.*, **198**:1031–1051, 2009. 85

[145] B. Sudret and A. der Kiureghian. **Stochastic finite element methods and reliability: a state-of-the-art report.** *Rep. No. UCB/SEMM-2000/08, University of California at Berkeley, USA*, **24**:619–644, 2000. 22

[146] B. Sudret and A. Der Kiureghian. **Stochastic finite elements and reliability : A state-of-the-art report**. *Technical Report no UCB/SEMM-2000/08, university of California, Berkeley,173 pages*, 2000. 3, 85

[147] B. Sudret and A. Der Kiureghian. **Comparison of finite element reliability methods**. *Prob. Eng.Mech.*, **17**:337–348, 2002. 86

[148] T. TAKADA. **Weighted integral method in multidimensional stochastic finite element analysis**. *Prob. Eng. Mech.*, **5(4)**:158–166, 1990a. 86

[149] T. TAKADA. **Weighted integral method in stochastic finite element analysis**. *Prob. Eng. Mech.*, **5(3)**:146–156, 1990b. 86

[150] B. VAN DEN NIEUWENHOF AND J.P. COYOTTE. **Modal approaches for the stochastic finite element analysis of structures with material and geometric uncertainties**. *Comp. Meth. Appl.Mech.Eng*, **192(33–34)**:3705–3729, 2003. 90

[151] F.J. WALL AND G. DEODATIS. **Variability response functions of stochastic plane stress/strain problems**. *J. Eng. Mech.*, **120(9)**:1963–1982, 1994. 4, 88

[152] P.D. WASSERMAN. *Neural Computing Ôheory and Practice*. 1989. 43

[153] P.D. WASSERMAN. *Advanced Methods in Neural Computing*. 1993. 43

[154] N. WIENER. **The homogeneous chaos**. *Am. J. Math.*, **60**:897–936, 1938. 3

[155] M.M.R. WILLIAMS. **Polynomial chaos functions and stochastic differential equations**. *Ann. Nucl. Energy*, **33**:774–785, 2006. 3

[156] D. XIU AND G. E KARNIADAKIS. **Stochastic Finite Element Analysis of seismic soil-structure interaction**. *Int. J. Heat Mass Transfer*, **46**:4681–4693,, 2003. 90

[157] D. XIU AND G.E. KARNIADAKIS. **The Wiener-Askey polynomial chaos for stochastic differential equations.** *SIAM J. Sci. Comput.*, **24**:619–644, 2002. 1

[158] F. YAMAZAKI AND M. SHINOZUKA. **Digital generation of non-Gaussian stochastic fields**. *Engineering Mechanics*, **114**:1183–1197, 1988. 22

[159] F. ZENG AND N.E. WIBERG. **Adaptive hŰ-p procedures for high accuracy finite element analysis of two–dimensional linear elastic problems**. *Computers and Structures*, **42**:869–86, 1992. 4

[160] C. ZIENKIEWICZ, J. CAGO, AND D.W. KELLY. **The hierarchical concept in finite element analysis**. *Computers and Structures*, **16**:63–5, 1983. 4

[161] C.O. ZIENKIEWICZ AND Y.K. CHEUNG. *The Finite Element Method in Structural and Continuum Mechanics*. 1967. 29

-