

## SYMBOLIC COMPUTING IN PROBABILISTIC AND STOCHASTIC ANALYSIS

MARCIN KAMIŃSKI<sup>a</sup>

<sup>a</sup>Faculty of Civil Engineering, Architecture and Environmental Engineering  
Łódź University of Technology, Al. Politechniki 6, 90-924 Łódź, Poland  
e-mail: Marcin.Kaminski@p.lodz.pl

The main aim is to present recent developments in applications of symbolic computing in probabilistic and stochastic analysis, and this is done using the example of the well-known MAPLE system. The key theoretical methods discussed are (i) analytical derivations, (ii) the classical Monte-Carlo simulation approach, (iii) the stochastic perturbation technique, as well as (iv) some semi-analytical approaches. It is demonstrated in particular how to engage the basic symbolic tools implemented in any system to derive the basic equations for the stochastic perturbation technique and how to make an efficient implementation of the semi-analytical methods using an automatic differentiation and integration provided by the computer algebra program itself. The second important illustration is probabilistic extension of the finite element and finite difference methods coded in MAPLE, showing how to solve boundary value problems with random parameters in the environment of symbolic computing. The response function method belongs to the third group, where interference of classical deterministic software with the non-linear fitting numerical techniques available in various symbolic environments is displayed. We recover in this context the probabilistic structural response in engineering systems and show how to solve partial differential equations including Gaussian randomness in their coefficients.

**Keywords:** probabilistic analysis, stochastic computer methods, symbolic computation.

### 1. Introduction

Probabilistic and stochastic computational analysis in engineering is important considering determination of reliability measures like reliability indices or failure probability (Sobczyk and Spencer, 1992), as well as both time-independent and dependent stochastic durability predictions (Melchers, 2002; Van Noortwijk and Frangopol, 2004). Some state of the art papers and analysis have been reported in the literature in the last years (Hurtado and Barbat, 1998; Grigoriu, 2000; Schueller, 2007; Chakraverty, 2014). On the other hand, the well-known applications of a similar nature are available in both economics (Wiggins, 1987) and, traditionally, in physics (Binder and Heermann, 1997). Since the enormous progress of all discrete computational techniques like the finite difference (To and Kiu, 1994) or finite element methods (Zienkiewicz and Taylor, 2005), the boundary element method (Burczyński, 1995), the finite volume method, the discrete element method or some meshless alternatives, one needs to explore the opportunity to expand them all towards uncertainty analysis in symbolic computing environments. The most

convenient idea would be to provide the general purpose codes in such platforms like MAPLE, MATHEMATICA, MATHCAD or MAXIMA, for instance, to automatically randomize most of the existing deterministic codes through direct intervention or external interoperability with the engineering systems like ABAQUS, ADINA, ANSYS or BEASY.

It is known that symbolic analysis has been employed in recent years mainly in the artificial intelligence area (Shachter *et al.*, 1990; Kwiatkowska *et al.*, 2004; 2007; López *et al.*, 2006) with no well documented applications in engineering analysis. The proposed idea is discussed in this paper in the context of the MAPLE scripts serving for determination of the fundamental probabilistic characteristics of several resulting functions of input random variables in the statistical, probabilistic and stochastic context (Moller and Beer, 2004; Spanos and Ghanem, 1991). This is done thanks to the Monte-Carlo simulation (Binder and Heermann, 1997; Brandt, 1999), analytical symbolic calculus as well as the stochastic perturbation technique (Kamiński, 2013), implemented in a dual way thanks to the direct differentiation method

(DDM) as well as the response function method (RFM). Numerical illustrations included here are focused on the single input random variable distributed according to the Gaussian distribution, but multi-variable analysis is also available. A very important and somewhat different issue is a scientific visualization of the resulting randomness in any problem—as a function of the input uncertainty dispersion level, with respect to the perturbation parameter, the total number of random trials or the perturbation method order.

## 2. Mathematical foundations

**2.1. Probabilistic and statistical definitions.** The following definitions are employed concerning the random variable  $B$ , its probability density function (PDF)  $p_B(x)$  and its number of realizations (assuming the same probabilities of their occurrence) equal to  $M$  (Feller, 1965; Spanos and Ghanem, 1991):

(a) expected values,

$$E[B] = \int_{-\infty}^{\infty} B p_B(x) dx, \tag{1}$$

$$\hat{E}[B] = \frac{1}{M} \sum_{i=1}^M B^{(i)};$$

(b) variances,

$$\text{Var}(B) = \int_{-\infty}^{+\infty} (B - E[B])^2 p_B(x) dx, \tag{2}$$

$$\widehat{\text{Var}}(B) = \frac{1}{M-1} \sum_{i=1}^M (B^{(i)} - E[B])^2;$$

(c) standard deviation,

$$\sigma_B = \sqrt{\text{Var}(B)}; \tag{3}$$

(d) coefficient of variation,

$$\alpha(B) = \frac{\sqrt{\text{Var}(B)}}{E[B]}; \tag{4}$$

(e)  $k$ -th central probabilistic moments (for  $k > 2$ ),

$$\mu_k(B) = \int_{-\infty}^{+\infty} (B - E[B])^k p_B(x) dx, \tag{5}$$

$$\hat{\mu}_k(B) = \frac{1}{M} \sum_{i=1}^M (B^{(i)} - E[B])^k;$$

(f) coefficient of skewness,

$$\beta(B) = \frac{\mu_3(B)}{\sigma_B^3} = \frac{E[B^3] - 3\mu(B)\sigma_B^2 - \mu^3(B)}{\sigma_B^3}; \tag{6}$$

(g) coefficient of flatness and kurtosis,

$$\gamma(B) = \frac{\mu_4(B)}{\sigma_B^4}, \quad \kappa(B) = \gamma(B) - 3; \tag{7}$$

where the dashed quantities denote the additional statistical estimators.

As is known, analytical derivations may result in extremely long expansions, especially for higher probabilistic moments like the third or the fourth one, so that, alternatively, they may be carried out using the complex exponential characteristic functions of  $B$  itself. A very interesting property (unimplemented in the symbolic computing environments) is probabilistic entropy known from information theory, having its deterministic counterpart in classical thermodynamics. Probabilistic entropy illustrates the uncertainty of occurrence of some random event at the next moment, so that the entropy equal to 0 accompanies a probability equal to 1 (or 0) for any random experiment showing no randomness at all. If a countable set of random events has  $n$  elements associated with the probabilities  $p_i$  for  $i = 1, \dots, n$ , then the entropy in this space is defined uniquely by the following sum (Shannon, 1948):

$$H = - \sum_{i=1}^n p_i \log_r(p_i), \tag{8}$$

where the logarithm basis  $r$  is the entropy unit; computational information theory is naturally based on bits, where  $r = 2$ . This discrete definition restricts the entropy values to non-negative real numbers, where  $H$  reaches maximum, i.e., for two elements, random space with both events having the same probability  $-1$  (like a bit of entropy per a single throw with the geometrically regular coin).

Although some computational applications show that consequences of the definition (8) may be of the paramount importance in data compression algorithms, its usage in the composite materials area, together with other structural engineering applications, demands continuous distributions, so that we propose (Kamiński, 2013)

$$h_B = - \int_{-\infty}^{+\infty} p_B(x) \log(p_B(x)) dx. \tag{9}$$

We may study entropy variations in different boundary value problems with random parameters from this formula to check if the given problem amplifies or damps an input uncertainty level. Let us note that this problem is of special importance in the composite materials area and homogenization theory, where the total number of uncertainty sources is huge (periodicity, material properties, interface defects location and size, etc.). Substituting the Gaussian probability density

function into Eqn. (9),

$$p_B(x) = \frac{1}{\sigma_B \sqrt{2\pi}} \exp\left(-\frac{(x - E[B])^2}{2\sigma_B^2}\right), \quad (10)$$

one determines this entropy as

$$h_B = \frac{1}{2} \log\left(2\pi e \sigma_B^2\right). \quad (11)$$

Once we need to analyse the truncated Gaussian random fields, then an integration process must be bounded using three-sigma rule, for instance.

**2.2. Stochastic perturbation technique.** The main philosophy of this method (Nayfeh, 2000) is to expand all the structural or system parameters and the response functions in the initial deterministic problem (heat conductivity or capacity, temperature, as well as its gradient) using a given order Taylor series with random coefficients. It was provided in the literature using the first order method (Sakata *et al.*, 2008), the second order technique (Kleiber and Hien, 1992), the third order approach (Peng *et al.*, 1998) and general usually higher order extension (Kamiński, 2013). It is done with the following representation of the random function  $\mathbf{u}(B)$  around its mean value (Kamiński, 2013):

$$\mathbf{u}(B) = \mathbf{u}^0(B^0) + \dots + \frac{\varepsilon^n}{n!} \frac{\partial^n \mathbf{u}(B)}{\partial B^n} \Big|_{B=B^0} \Delta B^n, \quad (12)$$

where  $\varepsilon$  is a given small perturbation (taken usually as equal to 1), while the  $n$ -th order variation is given as follows:

$$\varepsilon^n \Delta B^n = (\delta B)^n = \varepsilon^n (B - B^0)^n. \quad (13)$$

The expected values are exactly given with the use of the 10th order expansion as

$$E[\mathbf{u}(B)] = \mathbf{u}^0(B^0) + \dots + \frac{\varepsilon^{10}}{10!} \frac{\partial^{10} \mathbf{u}(B)}{\partial B^{10}} \mu_{10}(B) \quad (14)$$

for any natural  $m$  with  $\mu_{2m}(B)$  being central probabilistic moment of the  $2m$ -th order. Quite similarly, one may derive analytical relations for higher probabilistic moments taking into account that

$$\mu_p(B) = \begin{cases} 0, & p = 2k + 1, \\ \{\sigma_B\}^p (p-1)!!, & p = 2k \end{cases} \quad (15)$$

for any natural  $k \geq 1$ . As one may suppose, the higher order moments we need to compute, the higher order perturbations need to be included into all formulas, so that the complexity of the computational model increases

non-proportionally, together with the precision and the size of the output information needed.

The crucial numerical issue here may be determination of partial derivatives of the state function  $\mathbf{u}(B)$  with respect to the input parameter  $B$ , and, depending on the case study we can apply an analytical technique, fully implemented in the computer algebra system, a semi-analytical approach—a combination of the symbolic calculus with some other discrete technique implementation software or, finally, just a typical numerical solution provided entirely by this solver itself. Nevertheless, let us consider the following equations system:

$$\mathbf{L}(B)\mathbf{u}(B) = \mathbf{f}(B), \quad (16)$$

where  $\mathbf{L}(B)$  usually represents the main system matrix,  $\mathbf{f}(B)$  includes the boundary conditions imposed onto the system, while  $\mathbf{u}(B)$  is the structural response. According to the main philosophy proposed above, we provide the  $n$ -th order expansion at both of sides of this statement, getting

$$\begin{aligned} & \left\{ \varepsilon^0 \mathbf{L}^0(B^0) + \sum_{j=1}^n \varepsilon^j \frac{\partial^j \mathbf{L}(B)}{\partial B^j} (\Delta B)^j \right\} \\ & \times \left\{ \varepsilon^0 \mathbf{u}^0(B^0) + \sum_{j=1}^n \varepsilon^j \frac{\partial^j \mathbf{u}(B)}{\partial B^j} (\Delta B)^j \right\} \\ & = \mathbf{f}^0(B^0) + \sum_{j=1}^n \varepsilon^j \frac{\partial^j \mathbf{f}(B)}{\partial B^j} (\Delta B)^j. \end{aligned} \quad (17)$$

After multiplication at the L.H.S. in Eqn. (17), we collect the components of the same order of the perturbation parameter  $\varepsilon$ , arriving at the increasing order hierarchical equilibrium equations

$$\begin{cases} \varepsilon^0 : \mathbf{L}^0(B^0)\mathbf{u}^0(B^0) = \mathbf{f}^0(B^0), \\ \varepsilon^1 : \frac{\partial \mathbf{L}(B)}{\partial B} \mathbf{u}^0(B) + \mathbf{L}^0(B) \frac{\partial \mathbf{u}(B)}{\partial B} = \frac{\partial \mathbf{f}(B)}{\partial B}, \\ \vdots \\ \varepsilon^n : \sum_{j=0}^n \binom{n}{j} \frac{\partial^{n-j} \mathbf{L}(B)}{\partial B^{n-j}} \frac{\partial^j \mathbf{u}(B)}{\partial B^j} = \frac{\partial^n \mathbf{f}(B)}{\partial B^n}. \end{cases} \quad (18)$$

Further, leaving the highest order derivative of the solution  $\mathbf{u}(B)$  on the L.H.S. only and using simple algebra, one gets zeroth, first, second and, finally,  $n$ -th order equations as

$$\begin{cases} \mathbf{u}^0(B^0) = \{\mathbf{L}^0(B^0)\}^{-1} \mathbf{f}^0(B^0), \\ \frac{\partial \mathbf{u}(B)}{\partial B} = \{\mathbf{L}^0(B^0)\}^{-1} \left\{ \frac{\partial \mathbf{f}(B)}{\partial B} - \frac{\partial \mathbf{L}(B)}{\partial B} \mathbf{u}^0(B) \right\}, \\ \vdots \end{cases} \quad (19)$$

A primary objective is the real function  $u$  of the following polynomial form:

$$u(B; n) = \sum_{j=0}^n D^{(j)} B^j, \quad (20)$$

where  $D^j$ ,  $j = 1, \dots, n$ , are some real parameters and  $B$  is the given Gaussian random variable. The key issue is determination of the basic probabilistic moments and characteristics of this power series expansion with respect to the expectation and standard deviation of  $B$  itself. Let us note that physical and engineering applications are usually focused on the expectations, variances, standard deviations, coefficients of variation, third and fourth probabilistic moments, as well as the skewness and kurtosis of the structural random response.

**2.3. Least squares method.** Further, we provide a mathematical basis for the least squares method (LSM) adjacent to the fourth order tensor in both the non-weighted (NLSM) and the weighted (WLSM) version (Bjorck, 1996). We use a polynomial approximation of the  $s$ -th order (indexed by  $j$  here) through  $n$  numerical tests of the specific problem solved around the mean value of the given input random parameter  $B$ . We look for the following polynomial approximation:

$$u(B) \cong D^j B^j = f(\mathbf{D}, B), \quad j = 1, \dots, s, \quad s < n, \quad (21)$$

because approximation is applied as it guarantees automatic zeroing of partial derivatives with respect to  $B$  of an order higher than this polynomial.

We introduce residuals at each trial point to get an algebraic condition for this expansion coefficients, i.e.,

$$r_{(i)} = u^{(i)} - f(\mathbf{D}, B_i), \quad i = 1, \dots, n. \quad (22)$$

This is done by a minimization of the functional of weighted residuals

$$S = \sum_{i=1}^n w_{ii} r_{(i)}^2, \quad i = 1, \dots, n, \quad (23)$$

so that

$$\frac{\partial S}{\partial D^{(j)}} = -2 \sum_{i=1}^n w_{ii} r_{(i)} \frac{\partial f(\mathbf{D}, B_i)}{\partial D^{(j)}}. \quad (24)$$

Further, we adopt the following notation:

$$\mathbf{J} = J_{ij} = \frac{\partial f(\mathbf{D}, B_i)}{\partial D^{(j)}}, \quad (25)$$

and we form the modified equations as

$$\sum_{i=1}^n \sum_{j=1}^s J_{ij} w_{ii} J_{ij} D^{(j)} = \sum_{i=1}^n J_{ij} w_{ii} u^{(i)}, \quad (26)$$

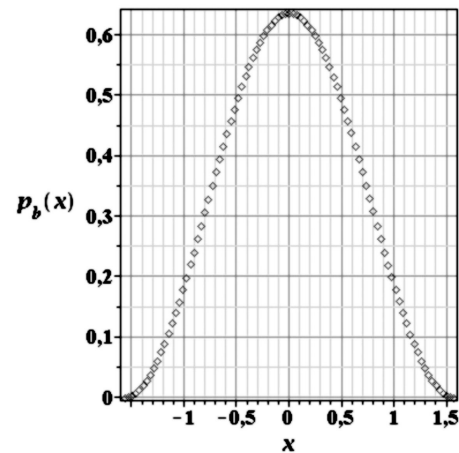


Fig. 1. Cosine-based probability density function.

obtaining the normal equations

$$((\mathbf{J})^T \mathbf{w} \mathbf{J}) \mathbf{D} = (\mathbf{J})^T \mathbf{w} \mathbf{u}. \quad (27)$$

This equations system (with the dimensions  $n \times s$ ) is solved symbolically in MAPLE.

### 3. Computational illustrations

**3.1. Analytical symbolic derivations.** One of the most important aspects of symbolic computing in the probabilistic aspect—in both applications and teaching of mathematics and theoretical engineering subjects like structural reliability—is the opportunity to derive analytical formulas describing basic probabilistic characteristics for various distributions. A fundamental linear transform for the binomial distribution is shown below for probabilistic characteristics up to the fourth order.

---

**Algorithm 1.** Analytical derivation of the basic probabilistic characteristics for a linear transform  $C$  of the binomial distribution.

---

```

> restart: with(Statistics);
B := RandomVariable(Binomial(n,p)); C := a * B + b :
{The binomial variable is the number of successes
in a set of n Bernoulli trials (p—probability of the
success)}
> E := ExpectedValue(C); CoV :=Variation(C);
D2 :=Variance(C); S :=Skewness(C);
K :=Kurtosis(C);

```

---

The same may be done using just numerical integration with a variety of specific algorithms because most of the random variables in engineering applications have their own upper and lower bounds. A slightly

non-standard application of the computer algebra system together with its visualization procedures is the development of new probability density functions (see Fig. 1).

**3.2. Technical aspects of the crude Monte-Carlo simulation.** The next important problem is determination of the accuracy of the simulation method, which means defining the number of random trials decisive for an efficient Monte-Carlo simulation of the given boundary value problem. Some perspective may be given by the following numerical error analysis, where we study initial numerical discrepancy for the Gaussian random variable and, in particular, for its probabilistic characteristics up to the fourth order for  $10^4$  random samples generated in MAPLE (see Algorithm 2). This very simple verification shows that the same, rather significant, number of random trials returns a 1.0% error in expectations, only a 1.5% error for the variance, but 2.2% and an almost 4% error level for the skewness and kurtosis. This justifies the general idea that really precise determination of higher order statistics needs many more random trials than estimation of expectations and variances.

**Algorithm 2.** A priori computational error analysis in the Monte-Carlo simulation.

```
> restart: with(Statistics):
> B := RandomVariable(Normal(10,1));
A := Sample(B, 10000):
> Bootstrap(Mean,A,replications=10000,
output=['value','standarderror']);
> Mean(B);
> Bootstrap(Variance,A,replications=10000,
output=['value','standarderror']);
> Variance(B);
> Bootstrap(Skewness,A,replications=10000,
output=['value','standarderror']);
> Skewness(B);
> Bootstrap(Kurtosis,A,replications=10000,
output=['value','standarderror']);
> Kurtosis(B);
```

The second, purely computational, issue is the import of data from the additional acquisition software and their further processing in the symbolic computing program. It is illustrated in the script contained in Algorithm 3. We take several measurements of the strength of the given material and transfer them to MS EXCEL. We need to plot a histogram and try to fit some probability density function, and this is to make a conversion into the MAPLE format; some basic estimators are calculated from these random trials, too. As documented in Fig. 2, mathematical fitting with some traditional analytical form is not so easy as the distribution finally appears to be of a bimodal character. This practically means that statistical estimators

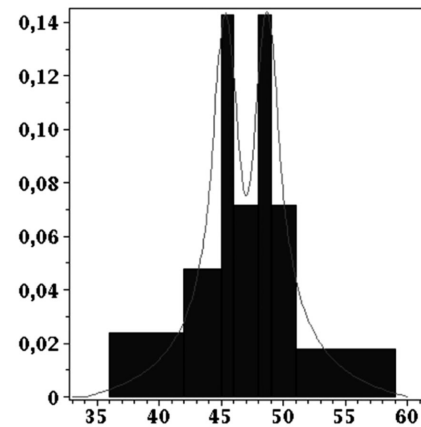


Fig. 2. Bimodal distribution histogram.

are taken from two different populations, and this conclusion may be more important than their numerical values; verification of the admissibility condition for the proposed PDF (for both bounded and unbounded integrals) is also quite straightforward, as in the previous case.

**Algorithm 3.** Statistics using interoperability of MAPLE and EXCEL.

```
> restart: with(stats): with(plots): with (describe):
with(plottools): with(stats[statplots]): with(Statistics):
{Experimental data are taken from the following file
and converted to MAPLE}
> wyniki:=ExcelTools:
-Import("D : \\Statistics\\data.xls", "data");
> lista1:=convert(data,list): lista2:=sort(lista1):
{Histogram of the estimated distribution}
> p1:=histogram(lista2,color=blue): display(p1);
{Data analysis}
> EX:=evalf(describe[mean](lista2));
> M:=evalf(describe[mode](lista2));
> S:=evalf(describe[standarddeviation](lista2));
> COV:=evalf(describe[coefficientofvariation](lista2));
> SK:=evalf(describe[skewness](lista2));
> K:=evalf(describe[kurtosis](lista2));
{We try to fit some function as the PDF}
> f:=proc(x) options operator,arrow;
> piecewise(x;34,0,x;44.5,(-1)*.16/(x-45.8)-.16/11.8,
x;49.5,(1/29)*sin((3/5)*Pi*(x-44.5))+84/767,x;60,.16/
(x-48.2)-0.16/11.8,0) end proc;
> p2:=plot(f(x),x=33..60,color=red): display({p1,p2});
{We check the basic integral condition of the PDF}
> c1:=Int(f(x),x=-infinity..infinity);
```

**3.3. Stochastic finite difference method.** The next case study shows how efficient symbolic computing is

in the implementation of the stochastic finite difference method in 1D structural analysis of the beams resting on an elastic foundation with some random design parameters (cf. Fig. 3).

Let us consider for this purpose the following ordinary fourth order differential equation for the linear elastic isotropic and statistically homogeneous beam exposed to the transversally distributed load  $q(x)$ :

$$\frac{d^2}{dx^2} \left( e(x)J(x) \frac{d^2 v(x)}{dx^2} \right) = -kv(x) + q(x). \quad (28)$$

It fulfills classical boundary conditions applicable in engineering theories of elastic beams, where  $e(x)$  [Pa] denotes the Young modulus of the beam,  $J(x)$  [m<sup>4</sup>] is the inertia moment of this structure,  $k$  [Pa] is the compliance coefficient of the subsoil and  $v(x)$  [m] denotes the deflection function of the beam.

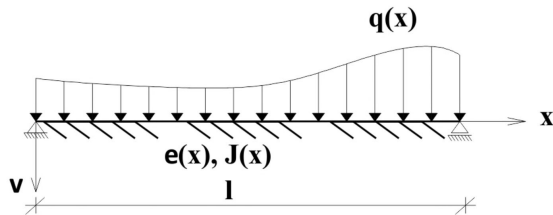


Fig. 3. Elastic beam on the Winkler foundation.

Let us also note that we further neglect the influence of the longitudinal and transversal forces as well as reduce the analysis to small deformations, which significantly simplifies the final form of the differential equilibrium equation; let us divide the entire domain of the beam length  $l$  [m] into  $n$  equidistant sub-domains. The following approximation of the spatial derivative of the function  $v(x)$  with central differences for the fourth derivative is employed:

$$\begin{aligned} \left( \frac{d^4 v}{dx^4} \right)_i &\cong \left( \frac{\Delta^4 v}{\Delta x^4} \right)_i \\ &= \frac{v_{i+2} - 4v_{i+1} + 6v_i - 4v_{i-1} + v_{i-2}}{\Delta x^4}. \end{aligned} \quad (29)$$

The particular case of the bending stiffness  $e(x)J(x) = const. = eJ$  enables us to transform this formula into a definitely simpler form,

$$\begin{aligned} v_{i-2} - 4v_{i-1} + \left( 6 + \frac{k\Delta x^4}{eJ} \right) v_i \\ - 4v_{i+1} + v_{i+2} = \frac{q_i \Delta x^4}{eJ}. \end{aligned} \quad (30)$$

The case study deals with a cantilever I-beam made of stainless steel  $e = 205$  GPa, having a constant

cross-sectional area, an inertia moment equal to  $J = 1.71 \times 10^{-6}$  m<sup>4</sup> (I 100), and resting on the elastic foundation characterized by the expected value of the compliance coefficient  $E[k] = 5 \times 10^7$  N/m<sup>2</sup>, while its standard deviation equals  $\sigma(k) = 0.30E[k]$ ; the external distributed load is constant along this beam,  $q = 10$  kN/m, for  $l = 3$  m, with its boundary conditions ( $y(x = 0) = 0$  and  $y'(x = 0) = 0$ ) discretised as follows:

$$\begin{aligned} v_0 = 0, \quad v_{-1} = v_1, \quad v_{n+1} = 2v_n - v_{n-1}, \\ v_{n+2} = 4v_n - 4v_{n-1} + v_{n-2}, \end{aligned} \quad (31)$$

where the nodes indexed with the numbers  $n+2$  and  $n+1$  are fictitious. The computational grid is partitioned into  $n = 25$  equidistant intervals and  $n$  is treated here as an additional parameter of this solution.

The computer code programmed in the MAPLE system allows for an automatic generation and computation of the first four probabilistic moments, all according to the 10th order stochastic perturbation approach. This code enables a computation of the expected values (Fig. 5(a)) and coefficients of variation (Fig. 5(b)) of the vertical displacements for this beam, and this is done in a semi-analytical manner; the resulting discrete displacements are analytical functions of the design parameters and, when differentiated ten times with respect to the parameter  $k$ , they form expectations and coefficients of variation of the structural response. This type of solution procedure is unavailable without the symbolic environment and enables relatively easy verification of the stochastic sensitivity of any system subjected to external forcing. Once we have a non-Gaussian input, one can modify the perturbation-based equations of equilibrium towards the computation of higher order statistics, too. As the solution is done via matrices with an analytically defined stiffness, there is no need to form and to solve (up to the tenth order) equilibrium equations similar to Eqns.(18) or (19).

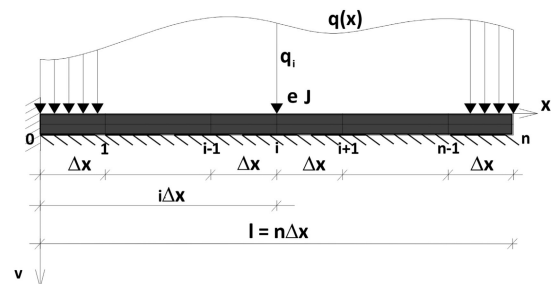


Fig. 4. SFDM discretisation of the elastic beam on the Winkler foundation.

**3.4. Stochastic finite element method.** Computational implementation of the generalised perturbation technique in conjunction with the finite element analysis is performed in the MAPLE symbolic environment. It enables (a) relatively easy extension of the technique to any perturbation order, (b) convergence analysis for particular probabilistic moments, (c) integration with FORTRAN codes for FEM programs, as well as (d) efficient visualization of the results. This code is tested using the example of 10 finite elements discretisation (1D bar elements) of the simple tension of a prismatic beam with a constant cross-section. The following data

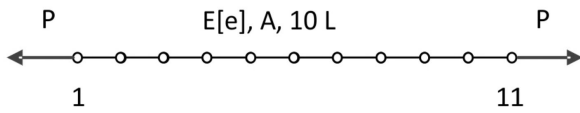


Fig. 5. Numerical discretisation of the SFEM test.

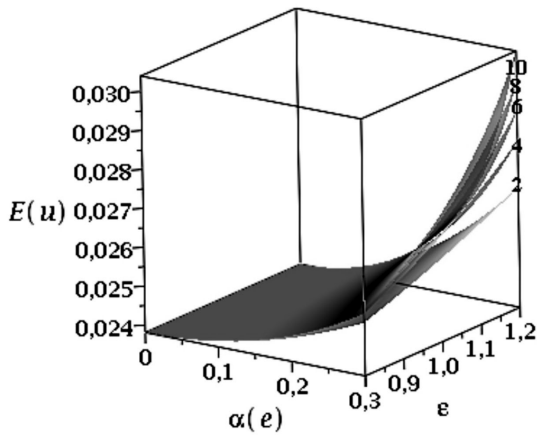


Fig. 6. Expectations of random displacements of the elastic bar under tension.

have been adopted:  $E[e] = 200$  GPa (Gaussian Young modulus),  $A = 10^{-4} m^2$  (cross-sectional area),  $L = 0,1$  m,  $P = 10^2$  kN (concentrated force applied), where  $L$  stands for each finite element length. The zeroth order stiffness matrix of a single finite element is given here as

$$\mathbf{K}_{2 \times 2}^0 = \frac{e^0 A}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \quad (32)$$

This test problem is a good illustration of probabilistic convergence because of an infinite number of non-zero partial derivatives with respect to a random input parameter, so that each new perturbation order introduces new components into the probabilistic output. Next, it is

**Algorithm 4.** SFDM solution for the elastic beam on the Winkler foundation.

```
> restart: with(plots): with(plottools): n:=25:
> for i from 1 by 1 to n do eqn[i] := v[i-2] - 4*v[i-1] + (6 +
k*(dx)^4/(E*J))*v[i] - 4*v[i+1] + v[i+2] = (dx)^4/(E*J)*q
end do:
> eqn[1] := subs(v[-1] = v[1], v[0] = 0, v[n+1] = 2*v[n] -
v[n-1], v[n+2] = 4*v[n] - 4*v[n-1] + v[n-2], eqn[1]) :
> eqn[2] := subs(v[-1] = v[1], v[0] = 0, v[n+1] = 2*v[n] -
v[n-1], v[n+2] = 4*v[n] - 4*v[n-1] + v[n-2], eqn[2]) :
> eqn[n-1] := subs(v[-1] = v[1], v[0] = 0, v[n+1] =
2*v[n] - v[n-1], v[n+2] = 4*v[n] - 4*v[n-1] + v[n-2],
eqn[n-1]) :
> eqn[n] := subs(v[-1] = v[1], v[0] = 0, v[n+1] =
2*v[n] - v[n-1], v[n+2] = 4*v[n] - 4*v[n-1] + v[n-2],
eqn[n]) :
> for i from 1 by 1 to n do eqn[i] end do:
> q := 10^4 : J := 1.71 * 10^(-6) : E := 20.5 * 10^10 : l :=
3.0 : dx := l/n :
> eqnsys:=seq(eqn[j],j=1..n): varlist:=seq(v[k],k=1..n):
> solution:=solve([eqnsys],[varlist]): assign(%):
> for i from 0 by 1 to n do l[i]:=i*dx end do:
> for i from 1 by 1 to n do
M1v[i] := v[i] + .5 * sk^2 * (diff(v[i], k$2)) + .25 * sk^4 *
(diff(v[i], k$4)) + .167 * sk^6 * (diff(v[i], k$6)) + .125 * sk^8 *
(diff(v[i], k$8)) + .1 * sk^10 * (diff(v[i], k$10)) end do:
> for i from 1 by 1 to n do M2v[i] := (25.2 *
(diff(v[i], k$5))^2 + 42. * (diff(v[i], k$4)) * (diff(v[i],
k$6)) + 24. * (diff(v[i], k$3)) * (diff(v[i], k$7)) + 9. *
(diff(v[i], k$2))
*(diff(v[i], k$8)) + 2. * (diff(v[i], k) * (diff(v[i], k$9))) *
sk^10 + (7. * (diff(v[i],
k$2)) * (diff(v[i], k$6)) + 8.75 * (diff(v[i], k$4))^2 + 14. *
(diff(v[i], k$3))
*(diff(v[i], k$5)) + 2. * (diff(v[i], k) * (diff(v[i], k$7))) *
sk^8 + (3.333 * (diff(v[i],
k$3))^2 + 2. * (diff(v[i], k) * (diff(v[i], k$5)) + 5. *
(diff(v[i], k$2)) * (diff(v[i],
k$4))) * sk^6 + (2. * (diff(v[i], k) * (diff(v[i], k$3)) + 1.5 *
(diff(v[i],
k$2))^2) * sk^4 + sk^2 * (diff(v[i], k))^2) end do:
> for i from 1 by 1 to n do stdv[i]:=sqrt(M2v[i]) end do:
> for i from 1 by 1 to n do alfav[i]:=stdv[i]/M1v[i] end do:
> k := 5 * 10^7 : sk := 0.30 * 5 * 10^7 :
> M1v[0]:=0: M2v[0]:=0: alfav[0]:=0:
> points1 := (seq([l[m], M1v[m]], m = 0..n)) :
pointplot(points1, axes = boxed, axis = [gridlines =
[colour = green, majorlines = 2]],
axesfont = [Times, kold, 12], title =
'expectation', titlefont = [Times, bold, 12]);
> points6 := (seq([l[m], alfav[m]], m = 0..n)) :
pointplot(points6, axes = boxed, axis
= [gridlines = [colour = green, majorlines =
2]], axesfont = [Times, kold, 12], title =
'alfav', titlefont = [Times, bold, 12]);
```

**Algorithm 5.** SFEM algorithm.

```

> restart : with(linalg) : with(plots) :
with(plottools) :
> #deterministicproblemsolution, 2 -
nodedlinearfiniteelementswith2degreesoffreedom
> C := matrix([[1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
0, 0, 0], [-1, 2, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, -1, 2, -1, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, -1, 2, -1, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, -1, 2, -1, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, -1, 2, -1, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, -1, 2, -1, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, -1, 2, -1, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, -1, 2, -1, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, -1, 2, -1, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 2, -1],
[0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 1]])
: kk := Y * A/L : K := kk * C : evalm(K) :
> ff := vector([-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]) : pp :=
P : rhs0 := pp * ff :
> #randomquantitydefinition
> b := Y : soln :=
evalm(leastsqrs(K, rhs0, 'optimize')) : assign(q =
soln) :
> #1stordersolutiontotheproblem
> K1 := diff(kk, b) * C : evalm(K1) : f1 :=
diff(pp, b) * ff : rhs1 := f1 - multiply(K1, q) :
evalm(rhs1) :
> soln1 := evalm(leastsqrs(K, rhs1, 'optimize')) :
assign(q1 = soln1) :
{}
> #10thordersolutiontotheproblem
> K10 := diff(kk, b$10) * C : evalm(K10) :
f10 := diff(pp, b$10) * ff : rhs10 :=
f10 - 10 * multiply(K1, q9) : evalm(rhs10) :
q10 := evalm(leastsqrs(K, rhs10, 'optimize')) :
> #InputprobabilisticdatafortheGaussianvariableb
> Y := 210e9 : A := 1E - 4 : L := 1.0E - 1 : sigb :=
alfab * Y : P := 10E5 : mi2b := sigb2 : mi4b :=
3 * sigb4 : mi6b := 1 * 3 * 5 * sigb6 : mi8b :=
1 * 3 * 5 * 7 * sigb8 : mi10b := 1 * 3 * 5 * 7 * 9 * sigb10 :
{Computations of probabilistic moments for the output}
{Expected values in various orders of the perturbation
analysis}
> E2q := evalm(q + 1/(2!) * eps2 * q2 * mi2b) : E4q :=
evalm(E2q + 1/(4!) * eps4 * q4 * mi4b) :
> E6q := evalm(E4q + 1/(6!) * eps6 * q6 * mi6b) :
E8q := evalm(E6q + 1/(8!) * eps8 * q8 * mi8b) :
> E10q := evalm(E8q + 1/(10!) * eps10 * q10 * mi10b) :

```

observed that the exact analytical solution for this problem consists of an inversion of the Gaussian input quantity, which means that the integral equivalent to the expected value does not converge and its value can be obtained by numerical integration only. Contrary to the SFDM experiment, now the solution is provided as a function of two variables—an input coefficient of variation as well as the perturbation parameter. This is a unique opportunity to verify numerically the influence of this parameter choice

and to show the polynomial character of the probabilistic response with respect to this specific variable. As one may notice in Fig. 6, the code is based on the least squares solution to the equilibrium problem, and the full DDM scheme is implemented to solve up to the 10th order perturbation-based equations; therefore, symbolic partial differentiation is applied now to the system matrices and vectors.

The expected values of displacements computed according to the 2nd, 4th, 6th, 8th and 10th order perturbation approach at the tensioned edge of a beam are plotted in Fig. 7 in the parametric space of the perturbation parameter and input coefficient of variation. As is intuitively clear, the larger the perturbation parameter, the larger the resulting expectation—the same remains true for the interrelation of the observed expectation versus input coefficient of variation. It is hard to state which parameter of these two is more influential, but it is apparent that the differences between neighbouring orders in the perturbation approach systematically decrease, which confirms numerically at least probabilistic convergence of this method; for full verification, the reader is referred to the work of Kamiński (2013).

**3.5. Probabilistic response determination with the WLSM.**

The last computational illustration shows the interoperability of the FEM system ABAQUS with MAPLE to use the series of boundary value problems of linear elasticity to solve a stochastic problem in the area of composite materials. We determine the effective elastic characteristics (Kamiński, 2005) of the polymeric matrix filled with rubber particles uniformly distributed in space.

This is done through equating deformation energies for the original and effective homogeneous material contained in the same volume, i.e., into the representative volume element containing here a single particle and surrounding matrix. Probabilistic calculation is provided on the basis of a series of numerical experiments leading to the determination of response functions of effective characteristics versus material characteristics of the composite (subjected to some initial uncertainty). We solve for the basic probabilistic characteristics of the homogenized tensor with the use of the weighted least squares method implemented in the MAPLE system (into the curve fitting library). Determination of the effective material tensor needs the strain energy of the heterogeneous medium that equals

$$U^{(\alpha)} = \frac{1}{2} \int_{\Omega} C_{ijkl}^{(\alpha)} \varepsilon_{ij}^{(\alpha)} \varepsilon_{kl}^{(\alpha)} d\Omega, \quad (33)$$

where  $\alpha$  indices the series of numerical experiments for the needs of the WLSM. The homogenized medium is a linear and isotropic one and accumulates the same amount



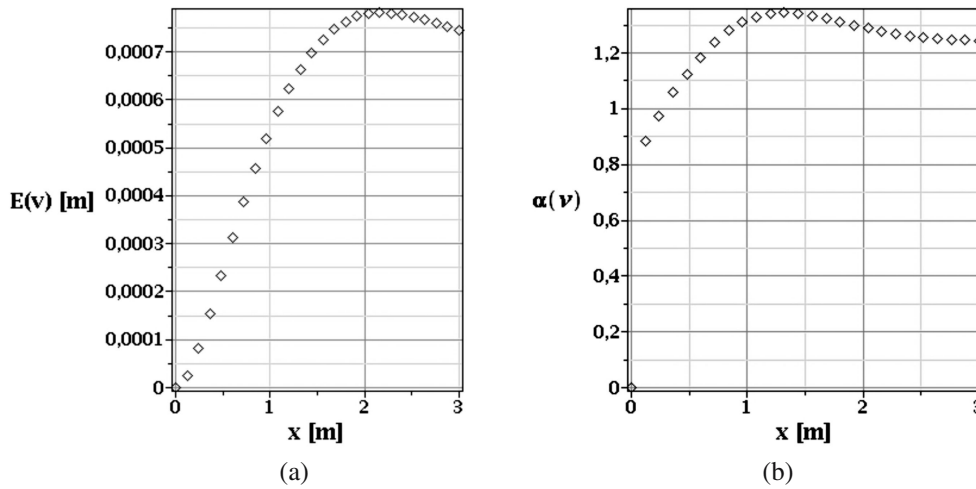


Fig. 7. Expectations (a) and coefficients of variation (b) of displacements of the elastic beam.

of energy having effective elastic characteristics series  $C_{ijkl}^{(eff)(\alpha)}$ , so that we compare this against the energy stored in the homogenized medium,

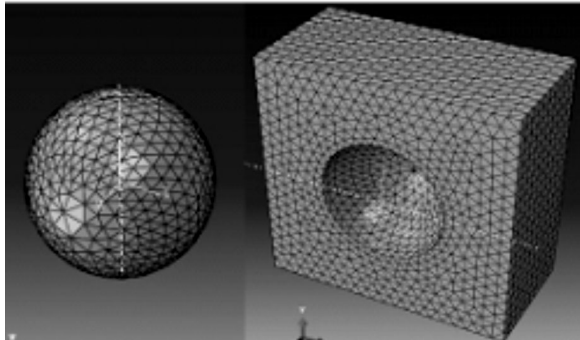


Fig. 8. ABAQUS discretization of the RVE for the particle-filled polymer.

$$\begin{aligned}
 U^{(\alpha)} &= \frac{1}{2} \int_{\Omega} C_{ijkl}^{(\alpha)} \varepsilon_{ij}^{(\alpha)} \varepsilon_{kl}^{(\alpha)} d\Omega \\
 &= U^{\text{hom}(\alpha)} \\
 &= \frac{1}{2} \int_{\Omega} C_{ijkl}^{(eff)(\alpha)} \varepsilon_{ij}^{h(\alpha)} \varepsilon_{kl}^{h(\alpha)} d\Omega,
 \end{aligned}
 \tag{34}$$

where  $\varepsilon_{ij}^h(\alpha)$  denotes the strain tensor adjacent to the homogenized equivalent medium. Numerical experiments with the homogenization method are completed in the FEM system ABAQUS, where a 3D model of the RVE is created, and where the series of the cell problems are solved for the series of the elastic energies in unidirectional and biaxial tension. Elastic parameters of the rubber particle are taken as  $E^{(p)} = 1.0$  MPa,  $\nu^{(p)} = 0.4888$ , and for the polymer matrix as  $E^{(m)} = 4.0$  GPa,  $\nu^{(m)} = 0.34$ . Spatial discretisation of the RVE domain

with tetrahedral elements C3D4 is given schematically for the full particle and half of the matrix in Fig. 8 (61,265 for the matrix and 3,829 elements for the particle); the entire cell has unit dimensions in all directions, while the particle has a diameter equal to 0.5 (effectively, a particle volume fraction of 6.5%).

According to the well documented better stability of the weighted least squares method (WLSM) (Kamiński, 2013), we propose the Dirac distribution of the weights around the mean value of the internal energy and of the effective tensor components to determine their probabilistic moments. This is done in the symbolic environment of the MAPLE program using various approximating polynomial orders—we determine analogous series of the effective elasticity tensor, whose probabilistic characteristics are computed by the weighting scheme  $[1, 1, 1, 1, 1, 6, 1, 1, 1, 1, 1]$  (6 stands for the input expectation) using 11 point representations of the Young modulus for the matrix and for the particle around their mean values (uncorrelated Gaussian variables). The resulting probabilistic characteristics of internal energy in unidirectional tension of the RVE are presented in Figs. 9(a)–12(b) (expectations, coefficients of variation, skewness and kurtosis). The output probabilistic characteristics are plotted as functions of the approximating polynomial order as well as the random dispersion of the given uncertainty source, where the left series corresponds to the random Young modulus of the matrix, while the right one to the uncertainty of the same modulus of the particle filler. Randomization of the particle elastic modulus results in almost constant expectations of the deformation energy (Fig. 9), independent of the initial uncertainty level (with some small fluctuations depending on the approximating polynomial order). Figure 9(b) shows

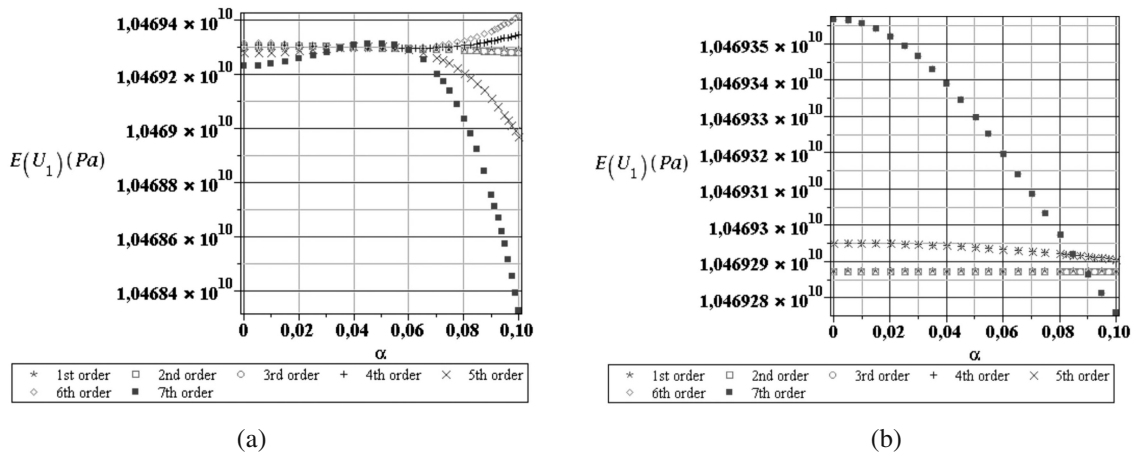


Fig. 9. Expected values of the internal energy versus Young modulus uncertainty of the matrix (a), of the particle (b).

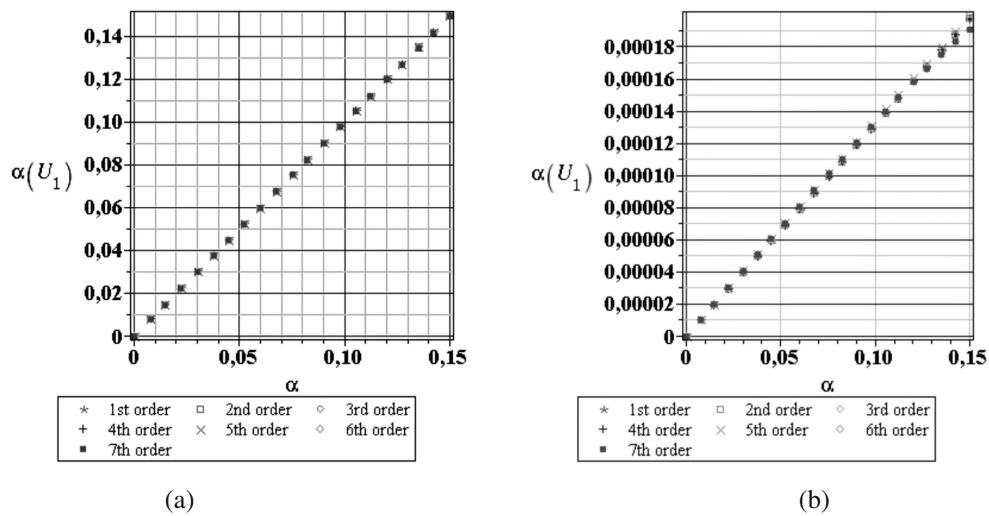


Fig. 10. Coefficients of variation of the internal energy versus Young modulus uncertainty of the matrix (a), of the particle (b).

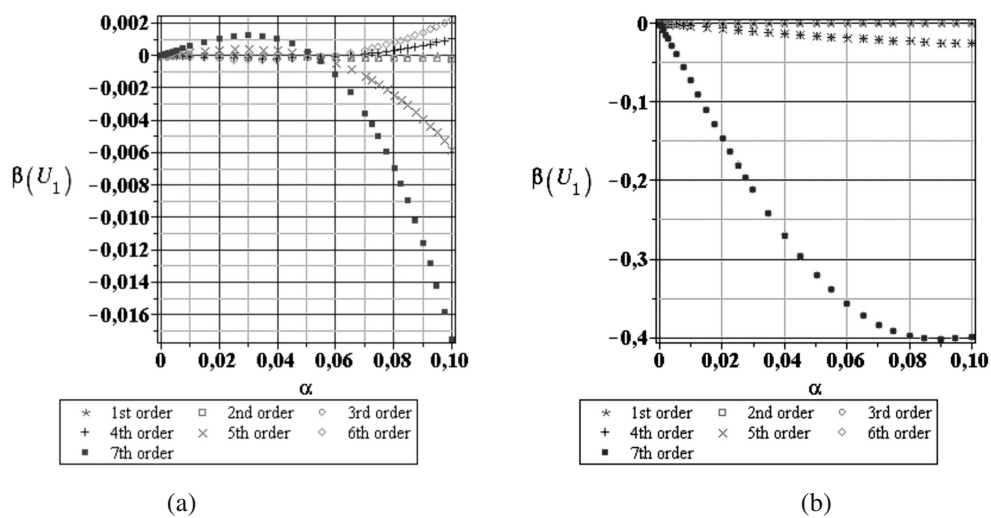


Fig. 11. Skewness of the internal energy versus Young modulus uncertainty of the matrix (a), of the particle (b).

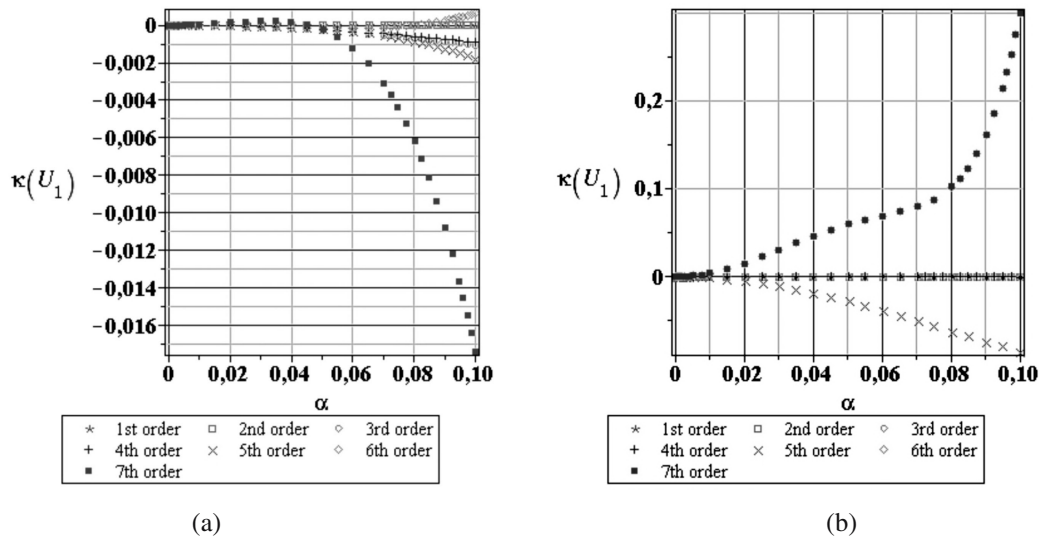


Fig. 12. Kurtosis of the internal energy versus Young modulus uncertainty of the matrix (a), of the particle (b).

that it results also in almost non-random variations in this energy, whose probabilistic distribution, considering Figs. 11 and 12, is rather distant from the Gaussian one. This is due to the fact that volume ratio of the particle into the RVE is very small. The situation changes significantly, while the matrix elastic modulus is an uncertainty source in the homogenization problem. Although the expectations behave in a quite similar way and are also almost constant, the coefficient of variation remains linearly dependent upon the input one and keeps its input values (no probabilistic entropy loss at all during the homogenization), which means that the homogenization process has no influence on the uncertainty level of the composite material itself. Finally, one can notice in Figs. 11(a)–12(a) that the resulting probability distribution of the deformation energy can be treated as the Gaussian one with a relatively small error. As is known, this significantly simplifies any further numerical experiments as the first two probabilistic moments determine the entire random distribution (effective characteristics depend linearly upon this energy). This is of paramount importance when randomization is provided by the Monte-Carlo simulation scheme because of the numerical errors displayed in the first numerical experiment.

#### 4. Concluding remarks

Computer algebra systems are very efficient in the implementation of statistical, probabilistic and stochastic techniques in the context of analytical, semi-analytical, discrete (simulation) as well as the stochastic perturbation technique. This is due to internal statistical functions as well as additional visualization procedures but thanks to the symbolic integration, differentiation, Taylor

expansions as well as the least squares approximation for some data series. There is no doubt that further implementations are necessary to explore the usage of probabilistic entropies computations as well as PDF recognition schemes (available in MAPLE in a limited range as the kernel density analysis). An interoperability with academic source codes relevant to discrete methods like the FDM or FEM as well as fast data interchange with the existing statistical packages may be crucial in the nearest future for the next significant milestones towards stochastic computations developments. The particular applications shown in the paper may be extended on the interval or stratified sampling with MAPLE or, in the framework of response functions recovery, towards a full Taylor–Newton–Gauss algorithm based on any real function continuous function fitting into the given data series (like exponential, power series or harmonic signals combination). It should be underlined that most of the MAPLE scripts discussed above are available online through the Maple Application Center (at <http://www.maplesoft.com/applications/index.aspx/>).

It is apparent from the above that the stochastic perturbation technique may deserve separate implementation into the existing computer algebra system, because it may (i) extend the capabilities of other commercial systems and, on the other hand, (ii) enable automatic randomization of almost all engineering and applied science problems that can be solved right now in the deterministic context including especially non-Gaussian random variables (Falsone, 2005). As was shown, symbolic implementation of the stochastic perturbation method returns very concise computer codes for both direct differentiation of the fundamental

governing equation of the given problem as well as for least squares recovery of the analytical response function through several deterministic solutions. There are of course some particular issues that could be resolved in symbolic computing environments (like for MAPLE at least), such as the programming of legends for 3D graphs and an increase in their resolution or some new, more detailed statistical commands.

### Acknowledgment

Computer modeling of the last numerical illustration has been carried out using the system ABAQUS operating on the MARS cluster (IBM BladeCenterH with 5365,76 GFlops) in the Academic Computer Center CYFRONET in Cracow, Poland.

### References

- Binder, K. and Heermann, D. (1997). *Monte Carlo Simulation in Statistical Physics*, Springer Verlag, Berlin/Heidelberg.
- Bjorck, A. (1996). *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA.
- Brandt, S. (1999). *Data Analysis Statistical and Computational Methods for Scientists and Engineers*, Springer-Verlag, New York, NY.
- Burczyński, T. (1995). Boundary element method in stochastic shape design sensitivity analysis and identification of uncertain elastic solids, *Engineering Analysis with Boundary Elements* **15**(2): 151–160.
- Chakraverty, S. (2014). *Mathematics of Uncertainty Modeling in the Analysis of Engineering and Science Problems*, IGI Global, Hershey.
- Falsone, G. (2005). An extension of the Kazakov relationship for non-Gaussian random variables and its use in the non-linear stochastic dynamics, *Probabilistic Engineering Mechanics* **20**(1): 45–56.
- Feller, W. (1965). *An Introduction to Probability Theory and Its Applications*, Wiley, New York, NY.
- Grigoriu, M. (2000). Stochastic mechanics, *International Journal of Solids and Structures* **37**(1–2): 228–248.
- Hurtado, J. and Barbat, A. (1998). Monte-Carlo techniques in computational stochastic mechanics, *Archives of Computer Methods in Engineering* **37**(1): 3–30.
- Kamiński, M. (2005). *Computational Mechanics of Composite Material*, Springer-Verlag, London/New York, NY.
- Kamiński, M. (2013). *The Stochastic Perturbation Method for Computational Mechanics*, Wiley, Chichester.
- Kleiber, M. and Hien, T. (1992). *The Stochastic Finite Element Method*, Wiley, Chichester.
- Kwiatkowska, M., Norman, G., Sproston, J. and Wang, F. (2007). Symbolic model checking for probabilistic timed automata, *Information and Computation* **205**(7): 1027–1077.
- Kwiatkowska, M., Parker, D., Zhang, Y. and Mehmood, R. (2004). Dual-processor parallelisation of symbolic probabilistic model checking, *12th International Symposium Modeling, Analysis and Simulation of Computer and Telecommunication Systems MASCOTS'04, Volendam, The Netherlands*, pp. 123–130.
- López, N., Nunez, M. and Rodriguez, I. (2006). Specification, testing and implementation relations for symbolic-probabilistic system, *Theoretical Computer Science* **353**(1–3): 228–248.
- Melchers, R. (2002). *Structural Reliability Analysis and Prediction*, Wiley, Chichester.
- Moller, B. and Beer, M. (2004). *Fuzzy Randomness. Uncertainty in Civil Engineering and Computational Mechanics*, Springer-Verlag, Berlin/Heidelberg.
- Nayfeh, A.H. (2000). *Perturbation Method*, Wiley-VCH Verlag GmbH, Weinheim.
- Peng, X., Geng, L., Liyan, W., Liu, G. and Lam, K. (1998). A stochastic finite element method for fatigue reliability analysis of gear teeth subjected to bending, *Computational Mechanics* **21**(3): 253–261.
- Sakata, S., Ashida, F., Kojima, T. and Zako, M. (2008). Three-dimensional stochastic analysis using a perturbation-based homogenization method for elastic properties of composite material considering microscopic uncertainty, *International Journal of Solids and Structures* **45**(3–4): 894–907.
- Schueller, G. (2007). On the treatment of uncertainties in structural mechanics and analysis, *Computers and Structures* **85**(5–6): 235–243.
- Shachter, R.D., D'Ambrosio, B. and Del Favero, B. (1990). Symbolic probabilistic inference in belief networks, *Proceedings of the 8th National Conference on Artificial Intelligence AAAI-90, Boston, MA, USA*, pp. 126–131.
- Shannon, C. (1948). A mathematical theory of communication, *The Bell System Technical Journal* **27**(4): 623–656.
- Sobczyk, K. and Spencer, B. (1992). *Random Fatigue: From Data to Theory*, Academic Press, Boston, MA.
- Spanos, P. and Ghanem, R. (1991). *Stochastic Finite Elements. A Spectral Approach*, Springer-Verlag, Berlin/Heidelberg.
- To, C. and Kiu, M. (1994). Random responses of discretized beams and plates by the stochastic central difference method with time co-ordinate transformation, *Computers and Structures* **53**(3): 727–738.
- Van Noortwijk, J. and Frangopol, D. (2004). Two probabilistic life-cycle maintenance models for deteriorating civil infrastructures, *Probabilistic Engineering Mechanics* **19**(4): 345–359.
- Wiggins, J. (1987). Option values under stochastic volatility. Theory and empirical evidence, *Journal of Financial Economics* **19**(2): 351–372.
- Zienkiewicz, O. and Taylor, R. (2005). *Finite Element Method for Solid and Structural Mechanics*, Elsevier, Amsterdam.



**Marcin Kamiński** (born in 1969 in Toruń, Poland) received his M.Sc. and Ph.D. degrees in civil engineering from the Technical University of Łódź in 1994 and 1997, and the D.Sc. degree in 2006 from the Wrocław University of Technology in Poland. He held a postdoctoral fellowship of the Foundation for Polish Science at the George Brown School of Engineering, Rice University in Houston, USA, in 1999–2000. He has authored two monographs, *Computational Mechanics of Composite Materials* (Springer, 2005) and *The Stochastic Perturbation Method for Computational Mechanics* (Wiley, 2013), as well as more than 250 publications in international journals and conference proceedings. He received the John Argyris Award in Computational Mechanics from ECCOMAS in 2001 and the J.T. Oden Faculty Fellowship at the ICES in The University of Texas at Austin in 2004. He was a visiting professor at Politecnico di Milano in Italy (2013) and the Institute of Polymer Research in Dresden, Germany (2009–2015). His main scientific interest is in probabilistic computational mechanics, reliability analysis, and durability prediction of steel and composite structures as well as homogenization methods. He is a member of the IACM, IASS, GAMM and the New York Academy of Science.

Received: 17 October 2013

Revised: 25 July 2014

Re-revised: 8 November 2014