doi:10.2195/LJ_Not_Ref_e_Plehn_072005

Approximation and Computation of Random Variables using Finite Elements

Dr.-Ing. Helge Plehn Prof. Dr.-Ing Rainer Bruns

Machine Elements and Technical Logistics,
Helmut-Schmidt-University,
University of the Federal Armed Forces, Hamburg

A method is introduced with which continuous random variables can be represented and linked with one another. The distribution functions of the random variables are approximated using a Finite Element approach in a finite interval $[t_{min}; t_{max}]$. As an example, two stochastically independent random variables are added by numerical computation of the convolution integral of their probability density functions.

Es wird ein Verfahren vorgestellt, mit dem stetige Zufallsgrößen rechnerunterstützt dargestellt und miteinander verknüpft werden können. Die Verteilungsfunktionen der Zufallsgrößen werden mit einem Finite-Elemente-Ansatz in einem endlichen Intervall $[t_{min}; t_{max}]$ approximiert. Die Addition zweier Zufallsgrößen wird durch numerische Berechnung des Faltungsintegrals durchgeführt.

Introduction

Increasing requirements on accuracy mean that it is often no longer sufficient to map technical systems by means of deterministic models. Computer-aided methods for solving stochastic models will therefore become more important in the future. The statistical precision of the widely-used Monte Carlo Method is often very low [4]. On the other hand, the analytical approach may fail due to limited computability, especially when random variables are described by means of complicated analytical functions. Although combining algorithms are known for some analytical functions [8], freedom in modelling is restricted by parameterization.

The method introduced in [3] is developed in order to represent continuous random variables and to link them with one another. The distribution functions are approximated using the Finite Elements Method. The combining algorithms are based on the

numerical computation of convolution integrals. A detailled description of different combining algorithms, eg. for sums, differences, products and quotients, can be found in [6].

For instance, the waiting and service times in a material flow system are considered, and the sum of these gives the throughput time. The random variables are approximated in the interval $[t_{min}; t_{max}]$, with $t_{min} \geq 0$. Other areas of application might include manufacturing tolerances or strengths of machine elements as well as reliabilities of technical systems. At the outset there will be an explanation of approximating the distribution function with the Finite Element Method, and this is followed by an examination of the errors. After this the addition of two independent stochastic variables will be dealt with as an example for combinations. This is followed by a consideration of the required memory capacity and computing time. In conclusion the results will be summarized and discussed.

Representation of stochastic variables

Continuous random variables are characterized by their density or distribution functions, with one of these two functions being sufficient to uniquely describe a random variable. To achieve a flexible, computer-oriented presentation of random variables, these functions have to undergo discretization. It is then intended that the discretized distribution or density function be used among other functions for numerically calculating random numbers for any desired random distributions and combinations of random variables.

Finite Element approach

Discretization of the distribution function of a random variable is conducted using the Finite Element Method, which has been successfully employed in other fields [2]. In this method the distribution function is approximated by piecewise defined quadratic approximation functions. By adapting elementation to the course of the function, the quantity of data can be kept small despite a high accuracy of approximation. As the function is defined for all intermediate values and the associated inverse function is numerically easy to form, the Finite Element Method is outstandingly suited for generating random numbers for the distribution being considered.

The three parameters of the quadratic approximation functions are defined by the values of the function in the two corner nodes and that in the mid-side node. In contrast to the general isoparametric second order approximation functions, in the meshes used here, the mid-side nodes are located precisely in the exact centre between the two corner nodes. The position of the mid-side node is therefore specified by the two corner nodes. Consequently, for N elements the distribution can be represented by a $3 \cdot (N+1)$ matrix (see Table 1).

The range of values for distribution functions is limited by definition to the interval $[0; 1]$. The approximation is valid for arguments within the interval $[t_{min}; t_{max}]$. Outside of this $F(t \leq t_{min}) = 0$ and $F(t \geq t_{max}) = 1$ apply. A distribution function for which the number of arguments is not limited is restricted by an upper and lower boundary

Table 1: The $3 \cdot (N+1)$-matrix for the Finite Element approximation of a distribution

node coordinates	function values corner nodes	function values mid-side nodes
$t_0 = t_{min}$	$F(t_0) = 0$	$F(\frac{t_0+t_1}{2})$
t_1	$F(t_1)$	$F(\frac{t_1+t_2}{2})$
t_2	$F(t_2)$	$F(\frac{t_2+t_3}{2})$
$\vdots$	$\vdots$	$\vdots$
t_{N-1}	$F(t_{N-1})$	$F(\frac{t_{N-1}+t_N}{2})$
$t_N = t_{max}$	$F(t_N) = 1$	1

in such a way that the resulting error of the approximation remains sufficiently small. Infinite corner elements provide a way of avoiding this restriction; however, in technical systems only random variables with a finite definition interval occur. As this examples deals with the distribution of times, only arguments with $t \geq 0$ are meaningful.

Figure 1 shows the approximation of the distribution function for a Gaussian distribution with 15 elements on the left. The expected value, μ, is 100 and the standard deviation, σ, is equal to 15. The corner nodes of the elementation are indicated by vertical black lines, the lines of the mid-side nodes are dotted. The different element widths $t_{i+1} - t_i$ can be clearly recognized. The right-hand side of Figure 1 shows the density function of the same approximation. As the density function is a derivation of the distribution function, the approximation functions of the density function are linear. Discontinuities in the corner nodes (continuous lines) may occur in the FE approach selected.

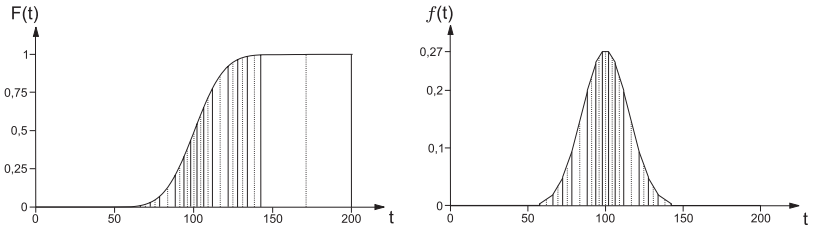

Figure 1: Finite Element approximation of a Gaussian distribution. Probability distribution function (left) and probability density function (right).

The quality of the approximation is influenced by three factors. The first factor is the number of elements. As the number of elements, N, increases, more precise mapping becomes possible while, at the same time, the quantity of data required for the approximation increases. The second factor is the computing input required to adapt

the elementation to the course of the function. This mesh optimization is dealt with in the following section. The last factor of influence is the approximation criterion used for determining the function values for the corner and mid-side nodes. This is dealt with in Section .

Mesh optimization

A decisive factor in reducing the computing input and the memory is optimization of the FE mesh. In this way, despite the high accuracy of approximation, the quantity of data and hence also the computing input for combinations can be kept small. There are established algorithms in the FEM for mesh generation and optimization. In adaptive networking [9], the FE program examines the results created with a given mesh. At all the sites where these results are questionable according to predefined criteria, the mesh is refined interactively until an acceptable result exists. The mesh improvements may take place via a finer division of the mesh or by means of higher-value elements. However, the majority of the procedures presented in recent years refer to mesh refinement.

One possibility of mesh refinement would therefore be to simply subdivide elements of insufficient approximation quality into several elements. The approximation quality could then be redefined and the mesh refined iteratively until a sufficiently good result has been achieved.

One further way of obtaining effective meshes with an acceptable computing input is the concept of hierarchical networks. The advantage lies in the fact that the entire structure does not have to be reanalysed, but instead the information from the original coarse mesh can be used for the finer mesh sections.

The approach used here first considers one-dimensional random variables. Function values of a random distribution function are assigned to every value within a permissible interval. As a result, the FE mesh extends in only one dimension. This makes a very high degree of freedom possible in mesh generation. The idea underlying the following mesh optimization consists of predefining an upper limit for the deviation between the distribution function and its approximation which is valid for all elements. Here either the number of the elements may be constant or a maximum total error is defined, which is obtained with a minimum number of elements. An algorithm for the mesh optimization of analytically presentable distribution functions is now presented in which the number of elements is constant. This is followed by an explanation of how this algorithm can be applied to those functions that are not known analytically but pointwise.

Analytical functions

In order to consider different random variables in a uniform manner, it is first necessary to approximate analytical distribution functions of random variables by means of FE distribution functions.

As already described, the error has to be equally large for all elements. We first consider an element with the corner nodes t_o and t_u and a mid-side node of $t_m = \frac{t_o+t_u}{2}$. To estimate the error in approximation, the density function is developed into a Taylor

series around the mid-side node:

$$f(t) = f(t_m) + \left[\frac{df}{dt}\right]_{t_m} (t - t_m) + \frac{1}{2}\left[\frac{d^2 f}{dt^2}\right]_{t_m} (t - t_m)^2 + \dots \tag{1}$$

The approximation functions of the FE distribution function are parabolic. Consequently, the approximation functions of the density function are linear. The constant and linear term of the Taylor series is therefore contained in the approximation. Consequently, the error $e(t)$ is:

$$e(t) = \frac{1}{2}\left[\frac{d^2 f}{dt^2}\right]_{t_m} (t - t_m)^2 + \dots \tag{2}$$

with all higher Terms of the Taylor expansion being ignored. The quadratic term of the Taylor expansion is greatest at the corner nodes. The maximum error $e_{max}(t)$ is therefore:

$$e_{max}(t) = |e(t_u)| = |e(t_o)| = \frac{1}{2}\left|\left[\frac{d^2 f}{dt^2}\right]_{t_m}\right| \left(\frac{L}{2}\right)^2 \tag{3}$$

$L = |t_o - t_u|$ is the length of the element being considered. Equation (3) can be transformed to:

$$\sqrt{8 \cdot e_{max}} = \sqrt{\left|\left[\frac{d^2 f}{dt^2}\right]_{t_m}\right|} \cdot L \tag{4}$$

Let e_{max} now be the permissible upper limit of the error, which is identical for all elements. Then $\sqrt{8 \cdot e_{max}}$ is equally large for all elements. The following then applies:

$$k = \sqrt{8 \cdot e_{max}} = \sqrt{\left|\left[\frac{d^2 f}{dt^2}\right]_{t_m}\right|} \cdot L = const \tag{5}$$

However, the mid-side node t_m of the element is not known a priori. The root term in the position t_m in (5) is therefore approximated using the arithmetic mean of this term over the entire element $[t_u; t_o]$.

$$\sqrt{\left|\left[\frac{d^2 f}{dt^2}\right]_{t_m}\right|} \approx \overline{\sqrt{\left|\left[\frac{d^2 f}{dt^2}\right]\right|}} = \frac{1}{L} \cdot \int_{t_u}^{t_o} \sqrt{\left|\left[\frac{d^2 f}{dt^2}\right]\right|}\, dt \tag{6}$$

If this approximation is inserted into (5), it follows that:

$$k = \int_{t_u}^{t_o} \sqrt{\left|\left[\frac{d^2 f}{dt^2}\right]\right|}\, dt = const \tag{7}$$

Let the antiderivative of the integral in equation (7) be $\delta_f(t)$:

$$\delta_f(t) = \int_{t_{min}}^{t} \sqrt{\left|\left[\frac{d^2 f}{d\xi^2}\right]\right|}\, d\xi \tag{8}$$

It then follows that:

$$k = \delta_f(t_o) - \delta_f(t_u) = const \tag{9}$$

The function $\delta_f(t)$ increases monotonously and can be interpreted as the accumulated error of the approximation. In Figure 2 $\delta_f(t)$ is plotted for the Gaussian distribution, which has already served as an example in Section . The ordinate is divided into $N = 15$ equidistant sections of height k. Assigning the abscissae yields the corner nodes of the approximation. (The corner nodes may be compared with those of Figure 1.)

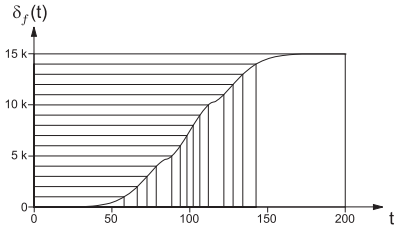

Figure 2: Error function $\delta_f(t)$ for a Gaussian distribution with $N = 15$ elements

If Equation (9) is summed for all N elements, this yields:

$$\sum_{i=1}^{N} [\delta_f(t_{o,i}) - \delta_f(t_{u,i})] = N \cdot k \tag{10}$$

The upper corner node of an element is the same as the lower corner node of the following element. Hence it follows that $t_{o,i} = t_{u,i+1}$ and $\delta_f(t_{o,i}) - \delta_f(t_{u,i+1}) = 0$. Therefore all of the corner nodes in the sum in Equation (10) are eliminated apart from the first and last. It follows that:

$$\delta_f(t_{max}) - \delta_f(t_{min}) = N \cdot k \tag{11}$$

Here $\delta_f(t_{min}) = 0$. The constant k can now be calculated:

$$k = \frac{\delta_f(t_{max})}{N} \tag{12}$$

The lower corner node of the first element and the upper corner node of the last element are defined as $t_0 = t_{u,1} = t_{min}$ und $t_N = t_{o,N} = t_{max}$ by the interval $[t_{min}; t_{max}]$. From all the other corner nodes it is known that the difference of their function values $\delta_f(t_{o,i}) - \delta_f(t_{u,i})$ is constant. As the function $\delta_f(t)$ rises continuously and monotonously, the corner nodes can be calculated with the aid of the inverse function δ_f^{-1}:

$$t_i = t_{o,i} = t_{u,i+1} = \delta_f^{-1}(i \cdot k) \tag{13}$$

doi:10.2195/LJ_Not_Ref_e_Plehn_072005

Where $i \in [0; N]$. Owing to the approximations used and the restrictive assumptions this mesh does not represent an optimum but it is very good and can be set up in relatively little time without iterations.

Functions identified by points

If the distribution network is to be generated for a random variable that is the result of combining other random variables, then in general no analytical expression for the distribution is known. However, the function value can be calculated at any desired position. A discrete frequency distribution is also obtained for random variables measured in practice. In the case of such functions, the method just described has to be adapted because the two-fold derivation of the density function cannot be determined analytically. If the function value is known for a sufficient number of equidistant points $t_i \in [0; M]$, it is possible to perform a numerical two-fold derivation [1]. Asymmetrical differential quotients are used for the corner points (t_0, t_M) and symmetrical ones for the remaining points.

$$\left.\frac{\Delta^2 f(t)}{(\Delta t)^2}\right|_{t_0} = \frac{2 \cdot f(t_0) - 5 \cdot f(t_1) + 4 \cdot f(t_2) - f(t_3)}{(\Delta t)^2} \tag{14}$$

$$\left.\frac{\Delta^2 f(t)}{(\Delta t)^2}\right|_{t_i} = \frac{f(t_{i-1}) - 2 \cdot f(t_i) + f(t_{i+1})}{(\Delta t)^2} \tag{15}$$

$$\left.\frac{\Delta^2 f(t)}{(\Delta t)^2}\right|_{t_M} = \frac{2 \cdot f(t_M) - 5 \cdot f(t_{M-1}) + 4 \cdot f(t_{M-2}) - f(t_{M-3})}{(\Delta t)^2} \tag{16}$$

Subsequent integration of the values over the roots yields the error function $\delta_f(t)$. The trapezoidal formula is used for numerically calculating the integral of the individual sections. The accumulated error function δ_f from (8) is then calculated as follows:

$$\delta_f(t_0) = 0 \tag{17}$$

$$\delta_f(t_i) = \delta_f(t_{i-1}) + \frac{\Delta t}{2}\left(\sqrt{\left|\left.\frac{\Delta^2 f(t)}{(\Delta t)^2}\right|_{t_{i-1}}\right|} + \sqrt{\left|\left.\frac{\Delta^2 f(t)}{(\Delta t)^2}\right|_{t_i}\right|}\right) \tag{18}$$

If instead of the density function only the distribution function is known, the second order quotient of differences from the density function in (18) has to be substituted by the third order quotient of differences for the distribution function. Instead of equations (14) to (16) the symmetrical and asymmetrical third order quotients of differences are then calculated according to [1].

After the pointwise calculation of the error function, $\delta_f(t)$ is interpolated linearly in the gaps. The inverse function $\delta_f^{-1}(t)$ can then be formed. The mesh can be derived from this according to ((12) and (13).

Approximation criterion

If the node coordinates are defined, the function values of the corner and mid-side nodes have to be selected so that the approximation is as accurate as possible. The method of least squares is normally used for this, which minimizes the integral over the square of the error e^2:

$$e_{tot} = \int_{t_{min}}^{t_{max}} e^2 dt \rightarrow min! \tag{19}$$

The $2 \cdot N - 1$ unknown function values can be determined from (19). However, for this purpose a linear equation system with $2 \cdot N - 1$ unknowns has to be solved, the coefficients of which are integrals over the definition interval $[t_{min}; t_{max}]$ that have to be calculated by means of a suitable numerical integration.

In order to avoid this computing input, a collocation method for determining the values of the node function was used initially. Here the deviation between the approximation and the function to be transformed is minimized at the nodes.

As the number of degrees of freedom in this FE approach is equal to the number of nodes, the collocation process imposes an exact agreement between the function and its approximation in the nodal point. The node function values are thus simply given from the values of the function to be approximated at the nodes. However, in the regions between the nodes it is in principle possible for deviations of any size to occur. In particular, it is possible that the course of the FE approximation within an element may not be monotonous. The node function values determined by the collocation process may therefore have to be altered again so that the monotony restriction is observed. If the parabola determined by the three nodes of an element has an extremum within the element, the function value of the mid-side node has to be varied in such a way that the vertex of the parabola coincides with a corner node. However, with good elementation this case occurs very rarely, and the error resulting from this is small.

In the collocation process the mean square error is larger than in the least squares method, but the quality of the approximation can be improved significantly more easily by raising the number of elements.

Error examination

The quality of the approximation depends on the number of elements. A Gaussian distribution was approximated with various numbers of elements, the mesh optimization described above was used for this. The result is shown in Figure 3. The mean quadratic deviation $\overline{e^2}$ between the FE approximation and the exact distribution function of the Gaussian distribution decreases continuously as the number of elements increases. However, the number of elements should not be too great due to the greater computing input required. The numerical errors also increase with the number of elements. As with every model, the following principle applies here: as good as necessary, as little as possible. An approximation suitable for most applications can be achieved with as few as 20 elements.

doi:10.2195/LJ_Not_Ref_e_Plehn_072005

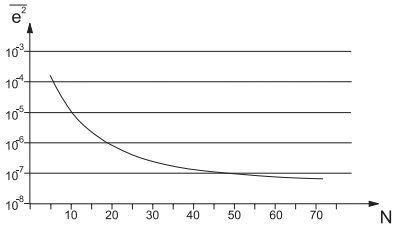

Figure 3: Mean quadratic error $\overline{e^2}$ over the number of elements N

Addition of random variables

Let the two independent random variables for a material flow system be service time, t_1, and waiting time, t_2, with the distribution functions of $F_1(t_1)$ and $F_2(t_2)$ and density functions of $f_1(t_1)$ und $f_2(t_2)$. The sum

$$t = t_1 + t_2 \tag{20}$$

corresponds to the throughput time, also called cycle time or sojourn time for the said material flow system. Let the density function of the throughput time be $h(t)$. According to [8] the density function of a sum of independent random variables can be calculated as the convolution of the addends density functions $f_1(t_1)$ and $f_2(t_2)$:

$$h(t) = f_1(t_1) * f_2(t_2) = \int_{-\infty}^{\infty} f_1(\tau) \cdot f_2(t-\tau) d\tau \tag{21}$$

The distribution function $H(t)$ satisfies the equation:

$$H(t) = \int_{-\infty}^{\infty} f_1(\tau) \cdot F_2(t-\tau) d\tau = \int_{-\infty}^{\infty} \frac{d}{d\tau} F_1(\tau) \cdot F_2(t-\tau) d\tau \tag{22}$$

The convolution integral is computed pointwise. The structure of the FE approximation is exploited for this purpose to reduce the computing time. The mesh of the result is adapted as described in Section .

The integrals for the calculation of differences, products and quotients [5, 7] are listed in Table 2. A detailed description of these and more combining algorithms can be found in [6].

Pointwise calculation of the convolution integral

First of all Figures 4, 5 and 6 will be used to illustrate how the convolution integral of two density functions is calculated.

Figure 4 shows the density function $f_1(t)$ of the service time on the left and that of the waiting time $f_2(t)$ on the right. The number of elements in both functions is very small for greater clarity. $f_1(t)$ is the approximation of a Gaussian distribution, $f_2(t)$ is that of

Table 2: Combinations and related calculations of the density functions

Combination	calculation of the density function
$z = y + x$	$f_Z(z) = \int_{-\infty}^{\infty} f_X(x) f_Y(z - x) dx$
$z = y - x$	$f_Z(z) = \int_{-\infty}^{\infty} f_X(x) f_Y(z + x) dx$
$z = y \cdot x$	$f_Z(z) = \int_{-\infty}^{\infty} f_X(x) f_Y(\frac{z}{x}) \frac{1}{\lvert x \rvert} dx$
$z = \frac{y}{x}$	$f_Z(z) = \int_{-\infty}^{\infty} f_X(x) f_Y(xz) \lvert x \rvert dx$

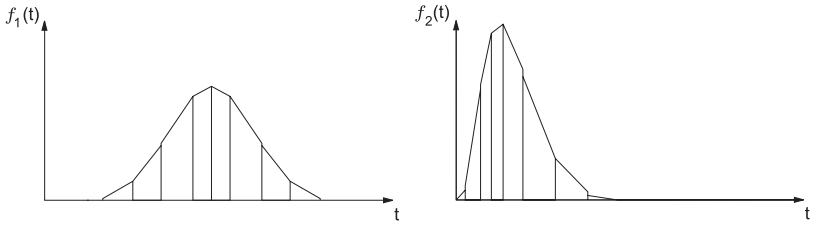

Figure 4: Approximation of two density functions

a k-Erlang distribution. The corner nodes are shown by vertical lines. The distribution functions are approximated element-wise by second order polynomials, for this reason the approximation functions of the density functions are linear. The inconsistencies at some corner nodes can be clearly recognized.

According to (21) one of the two functions (here f_2) is mirrored on the ordinate and shifted by t. Figure 5 shows the two density functions together. In this the abscissa, τ, the ordinate of the first density function, $f_1(\tau)$, and the ordinate of the second density function, $f_2(t - \tau)$, form a three-dimensional coordinate system.

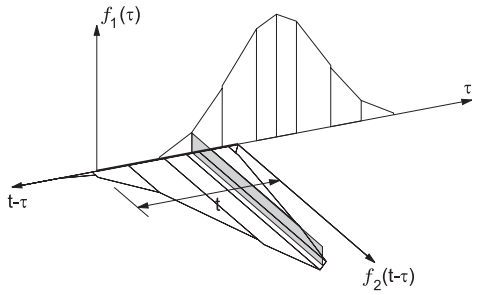

Figure 5: Orthogonal representation of two density functions to illustrate the integrand of the convolution integral

The bold section, t, corresponds to the overlap of the two functions. The integration

interval can be reduced from $[-\infty;\infty]$ to the range in which neither of the two factors of the integrand disappears in (21). The transparent rectangle in Figure 5 has sides of length $f_1(\tau_i)$ and $f_2(t-\tau_i)$. The area of the rectangle thus corresponds to the value of the integrand from (21) for a specific τ_i. When integrating this rectangle over the integration interval, one obtains the volume shown in Figure 6. The value of this volume thus equals $h(t)$.

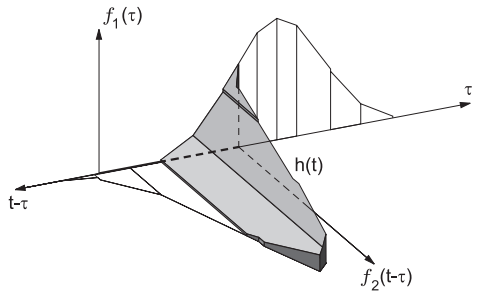

Figure 6: Geometric representation of the convolution of two density functions, the volume is equivalent to the value of the convolution integral at position t

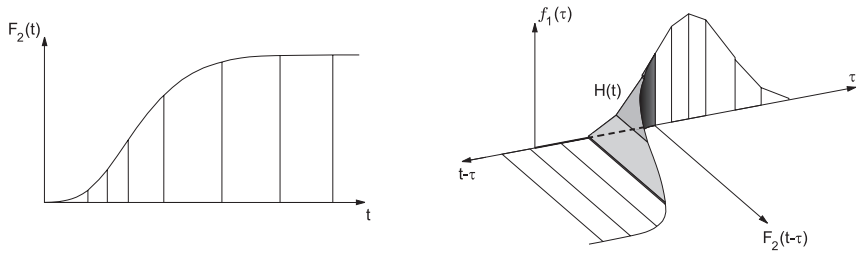

Figure 7: Left: approximation of a k-Erlang distribution function; right: geometric representation of the convolution

Replacing the density function, $f_2(t-\tau)$, in Figure 6 by the distribution function, $F_2(t-\tau)$, yields Figure 7. The distribution function, $F_2(t-\tau)$, is shown on the left, and on the right next to it is the volume corresponding to the convolution integral (22). The value of the volume $H(t)$ is equivalent to the probability that the sum of the service time, t_1, and waiting time, t_2, is less than or equal to a certain throughput time t.

The convolution of the two distribution functions $F_1(t)$ and $F_2(t)$ are now to be calculated for the position t. The convolution integral is given by (22):

$$H(t) = \int_{-\infty}^{\infty} \frac{d}{d\tau} F_1(\tau) \cdot F_2(t-\tau) d\tau \tag{23}$$

For $\tau < t_{1,min}$, $F_1 = 0$ applies, and for $\tau > t - t_{2,min}$, $F_2 = 0$. Furthermore, $F_1(\tau > t_{1,max}) = 1$, which therefore means that $f_1(\tau > t_{1,max}) = 0$. The integration interval can be reduced to $[\tau_{1,min}, \tau_{max}]$:

$$\tau_{min} = t_{1,min} \quad (24)$$
$$\tau_{max} = min\{t_{1,max}, t - t_{2,min}\} \quad (25)$$

Equation (23) can thus be simplified to:

$$H(t) = \int_{\tau_{min}}^{\tau_{max}} f_1(\tau) \cdot F_2(t-\tau) d\tau \quad (26)$$

As explained above, the function F_1 is defined by N_1 elements, the approximation functions in each case are second order polynomials. Let k_1 be used to number the elements continuously ($k_1 \in [1; N_1]$). If the argument t is found within the element k_1, then $t_{u,k_1} \leq t < t_{o,k_1}$, with t_{u,k_1} and t_{o,k_1} being the lower and upper corner nodes[1] of the k_1th element. In general, other polynomial coefficients apply for each element, which have the index k_1. This applies in analogy for the function F_2, where the elements are denoted by the index k_2. In general the following applies for the functions within an element:

$$F_1(t)_{k_1} = a_{1,k_1} \cdot t^2 + b_{1,k_1} \cdot t + c_{1,k_1} \quad (27)$$
$$F_2(t)_{k_2} = a_{2,k_2} \cdot t^2 + b_{2,k_2} \cdot t + c_{2,k_2} \quad (28)$$

Hence the following applies for the two factors of the convolution integral in (26):

$$f_1(\tau)_{k_1} = 2 \cdot a_{1,k_1} \cdot \tau + b_{1,k_1} \quad (29)$$
$$F_2(t-\tau)_{k_2} = a_{2,k_2} \cdot (t-\tau)^2 + b_{2,k_2} \cdot (t-\tau) + c_{2,k_2} \quad (30)$$

The integration interval from (26) is divided into M subintervals for which the polynomial coefficients of the two functions are always constant. The points at which the integration interval is divided are corner nodes of F_1 or of F_2. The M subintervals are denoted by the index i ($i \in [1; M]$). Elements k_1 and k_2 of the FE approximations F_1 and F_2 are uniquely assigned to each subinterval. It therefore follows that:

$$H(t) = \sum_{i=1}^{M} \int_{t_{i-1}}^{t_i} f_1(\tau)_i \cdot F_2(t-\tau)_i \, d\tau \quad (31)$$

The division of the integration interval into $M = 8$ subintervals is shown in Figure (8) for the above example. Table 3 contains the assignment of the elements k_1 and k_2 of the approximations to the subintervals i.

[1] In Figures 4 to 7 the corner nodes are indicated by parallel lines to the ordinate.

doi:10.2195/LJ_Not_Ref_e_Plehn_072005

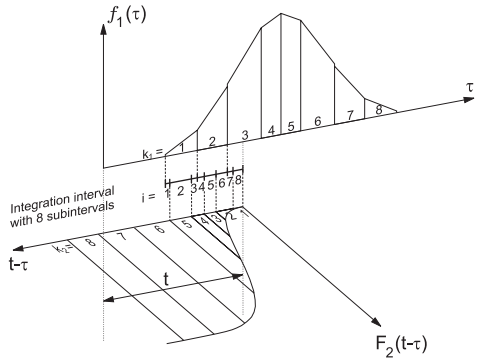

Figure 8: Division of the integration interval into $M = 8$ subintervals each with constant polynomial coefficients of the FE approximations F_1 and F_2.

Table 3: Assignment of the elements k_1 and k_2 to the subintervals i

i	1	2	3	4	5	6	7	8
k_1	1	1	1	2	2	2	3	3
k_2	6	5	4	4	3	2	2	1

In general the integration interval $[t_{1,min}; min\{t_{1,max}, t - t_{2,min}\}]$ from (22) is divided into M intervals $[t_{i-1}; t_i]$. The set for the new interval limits is:

$$\begin{aligned} \{t_i \mid i \in [0; M]\} &= \{t_0 = t_{1,min}\} \\ &\bigcup \{t_{o,k_1} \mid t_{o,k_1} \in [t_0; t_M],\ k_1 \in [1; N_1]\} \\ &\bigcup \{t - t_{u,k_2} \mid t - t_{u,k_2} \in [t_0; t_M],\ k_2 \in [1; N_2]\} \\ &\bigcup \{t_M = min\{t_{1,max}; t - t_{2,min}\}\} \end{aligned} \tag{32}$$

Where $t_{i-1} < t_i$ applies for all i. Equations (29) and (30) are now inserted into (31). During this process the polynomial coefficients are given new indexes[2].

$$\begin{aligned} H(t) &= \sum_{i=1}^{M} \int_{t_{i-1}}^{t_i} \Big[(2 \cdot a_{1,i} \cdot \tau + b_{1,i}) \\ &\quad \cdot \Big(a_{2,i} \cdot (t - \tau)^2 + b_{2,i} \cdot (t - \tau) + c_{2,i} \Big) \Big] d\tau \end{aligned} \tag{33}$$

[2]Summing takes place over $i = 1$ to M. However, $i \in [0; M]$ applies for the indices of the interval limits. The reason for this is that M adjacent intervals are naturally described by $M + 1$ interval limits.

$$
\begin{aligned}
= \sum_{i=1}^{M} & \left[\frac{1}{2}a_{1,i}a_{2,i}\tau^4 + \left(\frac{1}{3}b_{1,i}a_{2,i} - \frac{2}{3}a_{1,i}b_{2,i} - \frac{4}{3}a_{1,i}a_{2,i}t\right)\tau^3\right. \\
& + \left(a_{1,i}a_{2,i}t^2 + a_{1,i}b_{2,i}t - b_{1,i}a_{2,i}t + a_{1,i}c_{2,i} - \frac{1}{2}b_{1,i}b_{2,i}\right)\tau^2 \\
& \left. + \Big(b_{1,i}a_{2,i}t^2 + b_{1,i}b_{2,i}t + b_{1,i}c_{2,i}\Big)\tau\right]_{t_{i-1}}^{t_i} \qquad (34)
\end{aligned}
$$

As the polynomial coefficients of both functions are known for all M integration intervals $[t_{i-1}; t_i]$, the convolution integral $H(t)$ for the point t is determined with (34).

Mesh Generation

The approximation intervals of the distribution functions $F_1(t_1)$ and $F_2(t_2)$ are $[t_{1,min}; t_{1,max}]$ and $[t_{2,min}; t_{2,max}]$. Let the approximation interval of the distribution function for the sum $H(t)$ be $[t_{min}; t_{max}]$. The interval limits are given directly by the sum relationship $t = t_1 + t_2$:

$$t_{min} = t_{1,min} + t_{2,min} \qquad (35)$$

$$t_{max} = t_{1,max} + t_{2,max} \qquad (36)$$

If N is the number of the elements, the convolution integral is calculated at $10 \cdot N + 1$ equidistant reference points in the interval $[t_{min}; t_{max}]$. The error function $\delta_f(t)$ is determined by numerical differentiation from this point set as described in Section . The inverse function $\delta_f^{-1}(t)$ is then used to calculated the $N + 1$ corner nodes. The convolution integral has to be solved again for each of the function values at the corner and mid-side nodes. Monotony is guaranteed, as explained in Section . In accordance with Table 1 the matrix from the node coordinates and function values then yields the Finite Element approximation of the convolution for the two functions.

Hardware requirements

The algorithms presented were converted into software with Microsoft Visual Basic 6. When considering the computing time, it should be noted that code optimizations or possibly the choice of another language could considerably reduce the time required. The intention was merely to demonstrate the functional method.

Memory requirements

When Table 1 is considered, it can be seen that the $3 \cdot (N + 1)$ matrix contains a total of $3 \cdot N - 1$ variable elements. If these are mapped by variables with double accuracy (double type), this gives a memory requirement of $8 \cdot (3 \cdot N - 1) = 24 \cdot N - 8$ bytes. As a rule, a sufficient approximation of the probability distribution can be obtained with as

few as 20 elements. With 50 elements the mean square error is $< 10^{-7}$. In the latter case this yields a memory requirement of 1192 bytes for each stochastic variable. Compared to the capacity of today's computers the required memory is very low.

Computing time

The computing time also depends upon the number of elements. In the mesh optimization presented the convolution integral is calculated at $12 \cdot (N-1)$ points. If the functions to be convoluted are also approximated by N elements, then for each convolution integral the sum is formed over a maximum of $2 \cdot N$ summands according to equation (34). Here the coefficients $a_{j,i}$, $b_{j,i}$ and $c_{j,i}$ are to be calculated from the node parameters. A standard PC only requires approximately 0.7 s to calculate a convolution in which the number of elements is $N = 20$, including mesh optimization. In contrast, a convolution calculation with $N = 100$ elements takes as much as 12 s. Consequently, the computing time required for more complex models is not inconsiderable. The quality of the approximation for individual distributions should not be selected too high. $N = 20$ elements should in general be sufficient. The time required for a simulation experiment with this method is less than that of conventional, discretely stochastic simulations. However, the advantage primarily lies in the fact that in several simulation experiments only the submodels in which the distributions are changed have to be recalculated.

Summary

The approximation and combination of stochastic variables using the FE method was presented. Random variables with any possible distribution functions can be transformed and combined. The quality of the approximation can be varied by the number of elements and thus be adapted to the requirements. An algorithm for mesh generation was developed to adapt the elementation to the course of the distribution function. Addition was shown to illustrate the combination of random variables. The algorithm used for this is based on numerical calculation of the convolution integral. Other combinations such as subtraction, multiplication or division can be performed in an analogous manner. The hardware requirements were estimated. The quantity of data generated can be saved without any difficulty. The computing time required increases rapidly as the number of elements rises. Nevertheless, the development of increasingly powerful computers enables models of a higher complexity to be modelled and computed within an acceptable time.

References

[1] Anderson D A, Tannehill J C, Pletcher R H (1984) *Computational Fluid Mechanics and Heat Transfer.* Hemisphere Publishing Corporation, New York

[2] Bathe K-J (1995) *Finite Element Procedures.* Prentice Hall, Englewood Cliffs

[3] Bruns R (1999) *Finite-Elemente-Approximation von Verteilungsfunktionen stetiger Zufallsgrößen.* Z. angew. Math. Mech. **79** : 795-796

[4] Freiberger W, Grenander U (1971) *A Short Course in Computational Probability and Statistics.* Springer-Verlag, Berlin Heidelberg New York

[5] Jondral F, Wiesler A (2000) *Grundlagen der Wahrscheinlichkeitsrechnung und stochastischer Prozesse für Ingenieure.* Teubner, Stuttgart

[6] Plehn H (2003) *Eine Methode zur Berechnung Stochastischer Modelle mit Finiten Elementen.* Shaker Verlag, Aachen

[7] Schürger K (1998) *Wahrscheinlichkeitstheorie.* Oldenbourg, München

[8] Springer M D (1979) *The Algebra of Random Variables.* John Wiley & Sons, New York

[9] Steinbuch R (1998) *Finite Elemente - Ein Einstieg.* Springer-Verlag, Berlin Heidelberg New York

