Hybrid Methods in Engineering, vol. 3, pp. 1–9, 2001

SIMULATION OF INFILTRATION IN POROUS MEDIUM BY LAPLACE TRANSFORM TECHNIQUE AND FINITE DIFFERENCE METHOD

E. Wendland
Depto. de Hidráulica e Saneamento, Universidade de São Paulo, Caixa Postal 359, São Carlos–SP–Brasil, 13 560-970; [current address: Dept. de Hidráulica d Saneamento, Univ. de São Paulo, Caixa Postal 359, São Carlos, Brasil 13 560-970, ew@sc.usp.br]

M. T. Vilhena
Instituto de Matemática Aplicada, Universidade Federal do Rio Grande do Sul

A hybrid method using the Laplace transform and the finite difference method has been considered to determine the wetting profile in a soil submitted to an infiltration process. This scheme has been applied to the case of a semi-infinite column of sand, satisfying the constant flux condition at the surface. The numerical results and the experimental water content profiles are in good agreement.

KEYWORDS: hybrid method, finite difference, Laplace transform, porous medium

INTRODUCTION

Water movement through an unsaturated, nonswelling soil is mathematically described by Richards' equation [1]. The difficulties in solving this equation are due not only to the nonlinearity involved, but also to time-dependent boundary conditions and to the fitting of the experimental data for the physical parameters. In view of this, several different techniques of numerical analysis have been presented for solving transient water-movement problems, such as finite difference, finite element [2], boundary element [3] methods, etc.

These methods, however, present practical limitations, in the sense that it is often necessary to take very small time steps to avoid undesirable numerical oscillations in the solution. In this work we present a hybrid method using the Laplace transform (with numerical inversion) to remove the time dependency of the numerical scheme and the finite difference method (FDM) to approximate the space derivatives in the resulting transformed equation.

This approach was applied to the case of infiltration in an initially air-dried semi-infinite column satisfying the constant flux condition at the surface. The simulations exhibited a close approximation to experimental data given in the literature [4].

THEORY

The differential equation governing the one-dimensional movement of water in an unsaturated vertical soil column is given as

NOMENCLATURE

A	adjustment constant	P	load vector
A_k	weights for the summation	p_k	roots for the summation
c_1, c_2	adjustment constants	t	time variable, s
K	hydraulic conductivity, m/s	z	space variable, m
K_s	saturated hydraulic conductivity, m/s	α	adjustment constants
K_i^+	conductivity in the discrete equation, m/s	θ	volumetric water content
C	specific water capacity, 1/m	θ_s	saturated water content
L	Laplace transform operator	θ_r	residual water content
L^{-1}	inversion for Laplace transform operator	ψ	soil water tension, m
R	mobility matrix	$\overline{\psi}$	soil water tension in the Laplace space

$$C(\psi)\frac{\partial \psi}{\partial t} = \frac{\partial}{\partial z}\left[K(\psi)\frac{\partial \psi}{\partial z}\right] + \frac{\partial K(\psi)}{\partial z} \tag{1}$$

where ψ is the soil water tension, $K(\psi)$ the hydraulic conductivity, $C(\psi) = d\theta/d\psi$ the specific water capacity, and θ the volumetric water content.

We consider the following boundary conditions

$$q = q_0 \qquad t > 0\,,\ z = 0 \tag{2}$$

$$\psi = \psi_\infty \qquad t > 0\,,\ z \to \infty \tag{3}$$

and the initial condition

$$\psi = f(z) \qquad t = 0\,,\ z \geq 0 \tag{4}$$

To remove the time dependence from the governing Eq. (1), we apply the Laplace transform technique [5] defined as

$$\overline{\psi}(s) = \mathsf{L}\{\psi(t)\} = \int_0^\infty e^{-st}\psi(t)\,dt \tag{5}$$

with the inversion formula

$$\psi(t) = \mathsf{L}^{-1}\{\overline{\psi}(s)\} = \frac{1}{2\pi i}\int_{v-i\infty}^{v+i\infty} e^{st}\overline{\psi}(s)\,ds \tag{6}$$

where $s = v + iw$, and v, $w \in R$.

This inversion will be approximated by the Gaussian quadrature formula proposed by Stroud and Secrest [6]

$$\psi(t) = \mathcal{L}^{-1}\{\overline{\psi}(s)\} = \sum_{k=1}^{n} A_k \frac{p_k}{t} \overline{\psi}\left(\frac{p_k}{t}\right) \tag{7}$$

with the values of A_k (weights) and p_k (roots) given in the appendix for the summation of $n = 2$, 4, and 8 quadrature points.

Initially considering the parameters C and K as time independent (invariable within a time step) and applying the Laplace transform technique to Eq. (1), we obtain

$$C\left(s\overline{\psi} - f(z)\right) = \frac{\partial}{\partial z}\left(K\frac{\partial \overline{\psi}}{\partial z}\right) - \frac{1}{s}\frac{\partial K}{\partial z} \tag{8}$$

Discretizing the space variable in intervals Δz ($z_i = i\Delta z$, $i = 0, 1, \ldots, n$) and applying the finite difference method for Eq. (8) results in

$$C_i\left(s\overline{\psi}_i - f_i\right) = \frac{1}{\Delta z^2}\left[K_i^{+}\left(\overline{\psi}_{i+1} - \overline{\psi}_i\right) - K_i^{-}\left(\overline{\psi}_i - \overline{\psi}_{i-1}\right)\right] - \frac{1}{s}\left(\frac{K_{i+1} - K_i}{\Delta z}\right) \tag{9}$$

In Eq. (9), the diffusive term has been approximated by the central difference scheme and the advective term by upwind approximation. The expressions for the linearized terms C and K appearing in Eq. (9) are given in the appendix. The continuity of the function between the boundary ($z = 0$) and the domain ($z > 0$) was assured by the simultaneous discretization of Eq. (8) and the transformed Eq. (2) at the surface by using a fictive point above it.

Equation (9) can be rewritten in a matrix form

$$[R(s)]\{\overline{\psi}\} = \{P\} \tag{10}$$

where $[R(s)]$ is an $N \times N$ tridiagonal matrix.

The unknown variable ψ can now be determined at a specific time by solving Eq. (10) n times ($s = s_k$, $k = 1, \ldots, n$), where n is the number of quadrature points used.

Now we take into account the variation of the parameters C and K by discretizing the time variable ($t_j = j\Delta t$, $j = 0, \ldots, T$) and assuming these parameters to be constant over each time interval so that we can apply the previously discussed method.

Experimental Data

The proposed method has been applied to the case of sand, for which Haverkamp et al. [4] performed measurements. The initial and boundary conditions of the experiment were

$$
\begin{aligned}
f(z) &= -61.5\,\text{cm} \qquad t = 0\,,\ z \geq 0 \\
q_0 &= 13.69\ \text{cm/h} \qquad t > 0,\ z = 0 \\
\psi_\infty &= -61.5\,\text{cm} \qquad t > 0\,,\ z \to \infty
\end{aligned} \tag{11}
$$

The relationships for the hydraulic conductivity, water content, and soil water tension have been fitted by Haverkamp et al. [4]; the result is the expression

$$K(\psi) = K_s \frac{A}{A + |\psi|^{c_1}} \tag{12}$$

in which

K_s = 34 cm/h (saturated soil conductivity)
A = 1.175 $\times 10^6$ (adjustment constant)
c_1 = 4.74 (adjustment constant)

and

$$\theta(\psi) = \frac{\alpha(\theta_s - \theta_r)}{\alpha + |\psi|^{c_2}} + \theta_s \tag{13}$$

in which

θ_s = 0.287 (saturated water content)
θ_r = 0.075 (residual water content)
α = 1.611 $\times$ 10^6 (adjustment constant)
c_2 = 3.96 (adjustment constant)

RESULTS

The simulation with the hybrid method has been done for n = 2, 4, and 8 and Δz = 1 cm.

Figure 1 shows the computed water content profiles simulated with n = 2 and time intervals of 5 s in comparison with experimental data.

The computed and measured water content profiles agree very well. A closer approximation can be seen particularly in the region of advance of the wetting front. In Table 1, the mass balance after 2880 s (0.8 h) of simulation for n = 2, 4, and 8 is shown. The mass balance is given by the percentile difference between the water in the column after the simulation and the water theoretically infiltrated through the surface, normalized by the theoretical value.

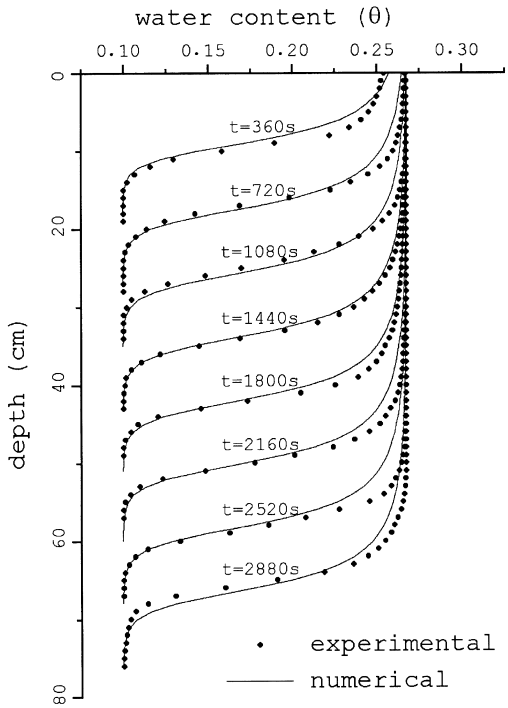

Fig. 1 Comparison between measured water-content profiles and results computed by Laplace Transform technique with $n = 2$ and $\Delta t = 5$ s.

Table 1 shows that the hybrid method exhibits good performance in terms of mass balance. It should be noted that the dependence on the number of points chosen for the inversion formula is irrelevant.

Figures 2 and 3 present the water content profiles computed with $n = 2$ and time intervals of 10 s and 20 s, respectively. The numerical results are also in good agreement with the experimental ones.

The mass balance after 2880 s of infiltration, presented in Table 2, shows the influence of the adopted time step in the results obtained by simulation. It can be seen that for greater time intervals, we get worse results for the mass balance for any value of n because of the strong nonlinear dependence of C and K on ψ.

Table 1. Mass Balance (%) after 2880 s of Infiltration Simulated with Time Steps of 5 s

	Points of Gauss quadrature		
	2	4	8
Error %	0.374	0.373	0.373

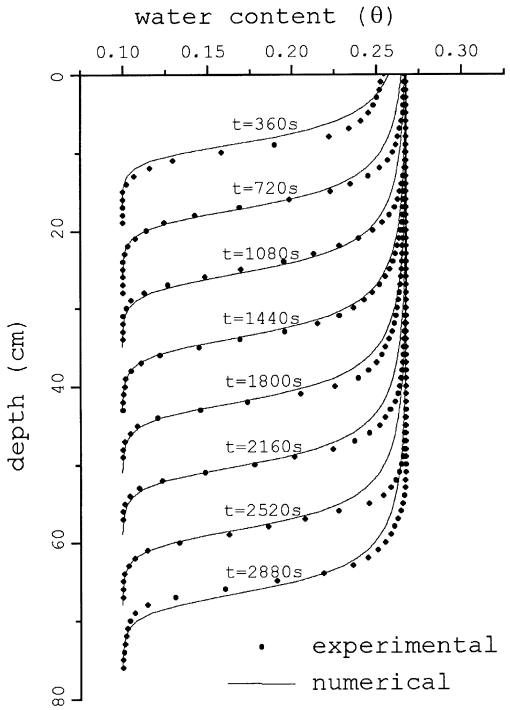

Fig. 2 Comparison between measured water-content profiles and results computed by Laplace Transform Technique with $n = 2$ and $\Delta t = 10$ s.

Table 2. Mass Balance after 2880 s of Infiltration Simulated with Time Steps of 10 s and 20 s

Intervals	Points of Gauss quadrature		
	2	4	8
10 s	0.635	0.671	0.694
20 s	1.052	1.049	1.049

CONCLUSION

This article presents a hybrid method involving the Laplace transform and the FDM for simulation of transient, nonlinear, diffusive problems. To demonstrate the reliability of this scheme, the numerical results obtained are compared with experimental data provided in the literature.

In the literature, it is commonly established that the Laplace transform should not be applied to nonlinear partial deferential equations (PDEs). With this work, we show that this statement is not necessarily correct. Close agreement between the computed and experimental water-content profiles shows that this approach is reliable and efficient despite the nonlinearity of the governing equation.

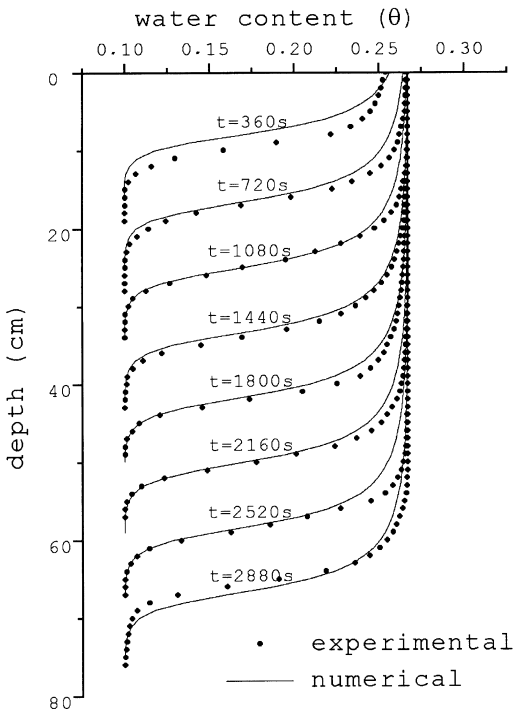

Fig. 3 Comparison between measured water-content profiles and results computed by Laplace Transform Technique with $n = 2$ and $\Delta t = 20$ s.

The major advantage of this method lies in the fact that there is no time-step limitation due to numerical instability, because the proposed approximated solution has a continuous dependence on time. However, the time interval has to be determined by taking into account the dependence of C and K on ψ to reach the desired accuracy. Consequently, this hybrid method is more appropriate for soils with a smooth variation of conductivity.

Comparison of the proposed method with existing numerical methods to confirm its superiority is an important step in its popularization, and we intend to address it in future works. The superiority of the proposed method will appear in cases where the changes in the nonlinear term are smooth, thereby allowing larger time steps than would be possible for the traditional time-marching schemes (e.g., explicit and implicit finite differences) schemes.

APPENDIX

1. Expressions for the linearized terms of Eq. (7).

$$f_i = f(z_i) = f(i\Delta z)$$

$$C_i = C(f_i)$$

$$K_{i+1} = K(f_{i+1})$$

$$K_i = K(f_i)$$

$$K^+ = \frac{K(f_{i+1}) + K(f_i)}{2}$$

$$K^- = \frac{K(f_i) + K(f_{i-1})}{2}$$

2. Weights and roots for the Gaussian quadrature in complex space, from Stroud and Secrest [6].

(a) Weights

	Re A_k	Im A_k
n = 2		
	0.500000000000000	1.414213562373095
n = 4		
	1.201377135377056	12.15505645082921
	–.7013771353770559	–2.839866120892252
	1.201377135377056	–12.15505645082921
	–.7013771353770559	2.839866120892252
n = 8		
	–39.79528773006955	1338.78390214722
	70.02048005466929	–641.9332467967155
	–34.2426429574756	127.8087478207435
	4.517450632875858	–6.142474951003654
	–39.79528773006955	–1338.78390214722
	70.02048005466929	641.9332467967155
	–34.2426429574756	–127.8087478207435
	4.517450632875858	6.142474951003654

(b) Roots

	Re p_k	Im p_k
n = 2		
	2.000000000000000	1.414213562373095
n = 4		
	4.787193103128466	1.567476416895208
	3.212806896871534	4.773087433276642
	4.787193103128466	–1.567476416895208
	3.212806896871534	–4.773087433276642
n = 8		
	10.16944600665751	1.649201796822228
	9.40637121369074	4.969217287623294
	7.738688146830548	8.370879306237983

4.685494632821197	12.01057859981379
10.16944600665751	–1.649201796822228
9.40637121369074	–4.969217287623294
7.738688146830548	–8.370879306237983
4.685494632821197	–12.01057859981379

ACKNOWLEDGMENTS

The authors thank an anonymous reviewer, whose constructive remarks certainly contributed to an improved article. The first author thanks FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) for financial support through grant 99/07835-1, which partially financed this publication.

REFERENCES

1. Philip, J. R. (1969). Theory of infiltration. *Advances in Hydroscience, 5,* 217–297.
2. Abriola, L. M. (1986). Finite element solution of the unsaturated flow equation using hierarchic basis functions. *Proceedings of the 6th International Conference on Finite Elements in Water Resources* (pp. 125–133)*,* Lisbon. Berlin: Springer-Verlag.
3. Pullan, A. J. (1988). Boundary element solutions of quasilinearised time-dependent infiltration. *Applied Mathematical Modeling*, *12,* 9–17.
4. Haverkamp, R., Vauclin, M., Touma, J., Wierenga, P. J. & Vachaud, G. (1977). A comparison of numerical simulations models for one-dimensional infiltration. *Soil Science Society of America Journal*, *41*, 285–294.
5. Bellman, R. E. & Roth, R. S. (1984). *The Laplace transform.* Singapore: World Scientific Publishing.
6. Stroud, A. H. & Secrest, D. (1966). *Gaussian quadrature formulas* (pp. 307–313). Englewood Cliffs, NJ: Prentice-Hall.

Hybrid Methods in Engineering, vol. 3, pp. 11–23, 2001

STUDY OF DIFFERENT TURBULENCE CLOSURE MODELS SIMULATING A NEUTRAL WIND TUNNEL FLOW EXPERIMENT

E. Ferrero
Dipartimento di Scienze e Tecnologie Avanzate, Università del Piemonte Orientale, Corso T. Borsalino 54, 15100, Alessandria, Italy

S. Trini Castelli & D. Anfossi
Istituto di Cosmogeofisica del CNR, Corso Fiume 4, 10133, Torino, Italy

S. Finardi
Arianet s.r.l., Viale Elvezia 42, 20052 Monza (MI), Italy

E. Di Lisi
Dipartimento di Fisica Generale, Università di Torino, via Pietro Giuria 1, 10125, Torino, Italy

In this work, three different turbulence closure models, based both on a prognostic transport-diffusion equation and a diagnostic equation, are implemented in the fluid-dynamical model RAMS to simulate a neutral flow in a wind tunnel. The first closure model (E-l) solves the equation for turbulent kinetic energy only, whereas in the second (E-ε) and third (E-ε-l) models both the turbulent kinetic energy equation and the one for its dissipation rate ε are solved. In the E-l model, ε is provided by a diagnostic equation as a function of E and the mixing length l. The E-ε and E-ε-l models differ in the definition of the diffusion coefficient for the momentum K_m. E-ε closure prescribes K_m by means of the prognostic values of E and ε, whereas the E-ε-l model uses the mixing length concept. Different values for the empirical constants defining the mixing length are adopted. The turbulence models are tested against observed data on flat terrain. An example of application over a two-dimensional valley (U.S. Environmental Protection Agency–RUSVAL experiments) is also shown. Mean wind and turbulent kinetic energy vertical profiles both measured and calculated are shown and compared.

INTRODUCTION

Key parameters in all the atmospheric numerical models at the regional or local scale are the diffusion coefficients of momentum, heat, and energy. The fact that the last two coefficients are generally considered proportional to the first stresses the importance of the diffusion coefficient of momentum and of evaluating the accuracy of a numerical model for it.

In previous works [10, 11], we discussed the results obtained when simulating the flow over a two-dimensional valley by using the circulation model RAMS [7] and different turbulence closure models. The main result we found is the inability of the 2.5 approximation of the Mellor and Yamada model [6] to accurately reproduce the flow and turbulence fields inside the valley. A simpler diagnostic K model [9] or more complex E-l and E-ε models, in which K_m is calculated by solving prognostic equations and accounting

for the mixing length concept, gave better results. Whereas E-l closure solves the turbulent kinetic energy (TKE) prognostic equation only, the E-ε model takes into account even the equation for the TKE dissipation rate ε. Here we consider both a standard version of the E-ε model, where the prognostic values of E and ε are used to calculate the momentum diffusion coefficient K_m, and what we define as an E-ε-l model, where a diagnostic equation based on the mixing length concept is introduced.

The transport and diffusion equations, both for TKE and ε, depend on a number of constants, which are to be determined by considering the physical parameters of the particular case studied. Most values found in the literature are given for engineering flows—see, for instance, Detering and Etling [2] and Rodi [8]—and have been tested in many studies. On the contrary, in the case of atmospheric boundary layer (ABL) simulations, the values are not yet well assessed.

In this article we present a sensitivity analysis of the constants defining the different turbulence closures and compare the values that we have estimated with some sets of values widely adopted in the literature. This analysis is carried out by considering the flow characteristics in flat terrain. The resulting best set is then used to compare the different closure models in the case of complex terrain.

The models are tested against the wind tunnel experiment RUSVAL [1,4] carried out by the U.S. Environment Protection Agency (EPA) Laboratory, where a simulated flow, both over flat terrain and over gentle topography, was reproduced and accurately measured. It is worth noting that in the experiment, turbulence developed analogously as in a real ABL. We estimated the tunnel Reynolds number value as $\mathrm{Re}_{\mathrm{TUNNEL}} \cong 2.5 \cdot 10^5$, whereas for the actual boundary layer we obtained $\mathrm{Re}_{\mathrm{ABL}} \cong 1.5 \cdot 10^8$. The critical Reynolds number defining the turbulent regime in a wind tunnel is generally estimated to be on the order of 10^3. This allows us to conclude that in both cases there is a regime of fully developed turbulence.

The section entitled Turbulence Models describes the turbulence models applied in this study. A brief description of the flow model is given as well. The Wind Tunnel Experiment presents the laboratory experiments and the measurement database used for the comparison and discusses the choice of scales and similarity criteria. Results and Discussion is devoted to the discussion of the results, and Conclusions provides some concluding remarks.

TURBULENCE MODELS

The three turbulence models were coupled to the RAMS circulation models. All models solve a transport-diffusion equation for the TKE, and two of them solve a transport-diffusion equation for ε as well. Except for E-ε closure, a parameterized mixing length is also used for calculating the diffusion coefficient.

The E-l Model

The prognostic equation for the TKE in a neutrally stratified flow in the standard E-l closure model that we adopted reads as:

$$\frac{dE}{dt} = \frac{\partial}{\partial x_j} K_E \frac{\partial E}{\partial x_j} + P - \varepsilon \tag{1}$$

where $P = -\overline{u_i' u_j'} \frac{\partial \overline{u_i}}{\partial x_j}$ is the shear production, with $\overline{u_i' u_j'} = -K_m \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right) + \frac{2}{3} E \delta_{ij}$

The dissipation rate is defined by the Kolmogorov relationship:

$$\varepsilon = \frac{c_\varepsilon E^{3/2}}{l_d} \tag{2}$$

where l_d is the dissipation length and the diffusion coefficient K_m is given by the Prandtl-Kolmogorov hypothesis:

$$K_m = c_\mu E^{1/2} l_m \tag{3}$$

where l_m is the mixing length. It is generally assumed that $l_d = l_m = l$.

The mixing length l adopted in RAMS simulations is that of Blackadar:

$$l = \frac{kz}{1 + kz/l_\infty} \tag{4}$$

where its asymptotic value l_∞, is estimated according to Ying [12]. Following the scaling procedure adopted in this work, described in The Wind Tunnel Experiment, the Ying value is l_∞ =201.6 m.

Eddy viscosity coefficient for TKE is defined as:

$$K_E = \alpha_e K_m \tag{5}$$

where α_e is an empirical constant.

The E-ε and E-ε-l models

In these closures, a prognostic transport-diffusion equation for the rate of TKE dissipation ε is joined to Eq. (1):

$$\frac{d\varepsilon}{dt} = \frac{\partial}{\partial x_j} \left(\alpha_\varepsilon K_m \frac{\partial \varepsilon}{\partial x_j} \right) + \frac{\varepsilon}{E} \left(c_{1\varepsilon} P - c_{2\varepsilon} \varepsilon \right) \tag{6}$$

The relationship between the constants reads as:

$$c_{1\varepsilon} = c_{2\varepsilon} - k^2 \frac{\alpha_\varepsilon}{c_\mu^2} \tag{7}$$

where k is the von Karman constant and α_ε is a constant here taken equal to 0.77; see, for example, Detering and Etling [2]. The empirical constants c_μ, $c_{1\varepsilon}$, and $c_{2\varepsilon}$ are to be determined.

The E-ε-l and E-ε models calculate K_m in different ways. In the first model, the same Eq. (3) of the E-l model is used with the Blackadar mixing length. In the second model, K_m is defined as [2]:

$$K_m = c_k \frac{E^2}{\varepsilon} \quad \text{where} \quad c_k = c_\mu^4 \tag{8}$$

THE WIND TUNNEL EXPERIMENT

The wind tunnel experiment we chose to test the numerical model is the EPA-RUSVAL experiment carried out in by the U.S. Environment Protection Agency Laboratory [4]. A neutral flow either on flat terrain or over a two-dimensional valley was simulated. The following characteristics of the vertical wind profiles were measured: roughness length z_0 = $0.16 \cdot 10^{-3}$ m, friction velocity u_* = 0.19 m/s, and free stream velocity u_∞ = 4 m/s.

The standard deviation of the three components of the wind fluctuation in the case of the flat terrain were $\sigma_u = 2.5u_*$, $\sigma_v = 1.2u_*$, and $\sigma_w = 1.8u_*$, respectively. The maximum valley depth was H = 0.117 m and the valley width was $2a$, where a = 0.936 m. Hence, the aspect ratio a/H was 8. Other experiments, not accounted for in this work, were performed with the aspect ratio equal to 5 and 3.

A rich data set was measured during the wind tunnel experiment. Vertical profiles of mean wind velocity and Reynolds stress components σ_u, σ_v, σ_w, and $\overline{u'w'}$ were measured along the x axis (longitudinal direction of the tunnel) at 5 different locations in the flat-terrain experiment and at 15 locations upwind, inside, and downwind of the valley (referred to by means of the ratio x/a). It is worth noting that the flow was essentially two-dimensional because of the symmetry with respect to the x–z plane. The velocity components along the y axis (crosswind direction) were assumed to be zero.

Because we are interested in assessing the accuracy of a closure model for ABL simulation and we used RAMS as our circulation model, the wind tunnel experiment was scaled in the numerical simulation to the dimension of real atmosphere. A scaling factor (600:1) for lengths was chosen because the authors of the EPA experiments, referring to the RUSHIL simulation [3] performed in the same tunnel, determined that the wind tunnel ABL (1 m) corresponded to a 600-m actual neutral ABL. In this way, the maximum depth of the valley was H = 70.2 m and the roughness length z_0 = 0.096 m. The scaled values of the five measuring locations in flat terrain are x = –645 m, –105 m, 285 m, 2295 m, and 4215 m.

Accounting for the similarity theory of the neutral surface layer, we adopted a logarithmic law, in which the parameters u_* and z_0 were determined by fitting the measured profile:

$$\frac{u}{u_*} = \frac{1}{k}\ln\left(\frac{z}{z_0}\right) \tag{9}$$

where k is the von Karman constant ($k = 0.4$). Because of the nondimensional form of this profile, no scaling on the velocities had to be done. Consequently, the time scale was the same as that for the lengths.

The horizontal simulation domain was 6000×500 m^2 with grid size $\Delta x = \Delta y = 50$ m. In the vertical direction, a stretched grid was used to improve the resolution in the lower layers. The minimum vertical grid size was 10 m and the maximum 60 m.

RESULTS AND DISCUSSION

The basic aim of this work was to search for a set of constants defining the different closures able to provide good performances in simulating a neutral ABL. We performed many simulations with the three turbulence models and different values of these constants, selected from the literature or calculated by us in previous works. Here we produce a selection of the results obtained, summarized in the following tables. Table 1 shows the values considered for the E-l closure. In the first simulation (El_1), the two empirical constants are based on the calculation by Ying [12] applied to the RUSHIL case, whereas in the second simulation (El_2) the empirical constants are the values calculated by our group in a previous work [11]. Table 2 refers to simulations with the E-ε and E-ε-l closures. In both cases, we consider two tests. The first test concerns a set of standard constants evaluated from laboratory data [5] and widely used in engineering flow problems (Eε_1 and Eεl_1), compared with a similar set (Eε_2 and Eεl_2) partially calculated by our group [11]. The second test adopts the set of constants proposed by Detering and Etling [2] for ABL calculations (Eε_3 and Eεl_3). Even in this case, they were compared with a similar set where some values are substituted by the constants that we estimated (Eε_4 and Eεl_4).

In the case of the flat-terrain experiment, the first inflow measurement locations, at the beginning of the tunnel, showed a large variability in the observed profiles, whereas the profiles at the last two locations, which were in the middle of the tunnel and quite far from each other, show qualitatively the same behavior. This means that probably in the experiment the flow needed to cover a certain distance to reach its equilibrium and to let

Table 1. E-l Closure Simulations, Set of Constants

Simulation	Study	Empirical constants
El_1	Ying [12]	$c_\varepsilon = 0.17$
		$c_\mu = c_\varepsilon^{1/3} = 0.55$
El_2	Trini Castelli et al. [11]	$c_\varepsilon = 0.08$
		$c_\mu = 0.42$

Table 2. E-ε and E-εl Closure Simulations, Set of Constants

Simulation	Study	Empirical constants
Eε_1	Launder-Spalding [5]	$c_\mu = 0.55$
Eεl_1		$c_{1\varepsilon} = 1.44$, $c_{2\varepsilon} = 1.92$ $\alpha_e = 1.0$, $\alpha_\varepsilon = 0.77$
Eε_2	Trini Castelli et al. [11]	$c_\mu = 0.42$, $c_{1\varepsilon} = 1.22$
Eεl_2	Launder and Spalding [5]	$c_{2\varepsilon} = 1.92$, $\alpha_e = 1.0$, $\alpha_\varepsilon = 0.77$
Eε_3	Detering and Etling [2]	$c_\mu = 0.40$
Eεl_3		$c_{1\varepsilon} = 1.13$, $c_{2\varepsilon} = 1.90$ $\alpha_e = 1.35$, $\alpha_\varepsilon = 0.77$
Eε_4	Trini Castelli et al. [11]	$c_\mu = 0.42$, $c_{1\varepsilon} = 1.22$
Eεl_4	Detering and Etling [2]	$c_{2\varepsilon} = 1.90$, $\alpha_e = 1.35$, $\alpha_\varepsilon = 0.77$

turbulence develop fully. Hence, the last measurement points are to be considered more meaningful when comparing simulation results with observed data. For this reason, we present the results obtained at the two last locations.

In Fig. 1, the normalized horizontal wind velocity component and TKE vertical profiles obtained on flat terrain by the El_1 (dotted line) and El_2 (solid line) simulations are depicted. Although there are no differences regarding the mean wind in the two simulations, a significant improvement is found for the TKE profile using the second set of values (El_2). This result can probably be related to the adopted c_μ—that is, $c_\mu = u_* / E^{1/2}$ [11]—calculated by using experimental values for u_* and surface layer TKE.

Figure 2 shows the results of the comparison between the Eε_1 and Eε_2 simulations, based on the laboratory set of constants. Even in this case, the agreement with the observed data is better when adopting the values that take into account surface layer parameters in Eε_2, because the TKE maximum value doubles with respect to Eε_1 and there is no underestimation in the lower layer. We recall that $c_{1\varepsilon}$ depends upon c_μ, per Eq. 7. Differences in the wind horizontal component profiles are negligible.

The Eε_3 and Eε_4 simulations, based on the ABL set of constants, give equivalent results, as can be seen in Fig. 3. This is the result of the very small differences in the values of the constants (see Table 2). The quality of the results is comparable to the best simulation (Eε_2) in Fig. 2 at the lower layer, whereas in the middle levels a slightly worse agreement with measured data is observed. A small improvement in the mean wind is shown by Eε_4 at $x = 4215$ m with respect to the previous measurement location ($x = 2295$ m).

Figure 4 is the same as Fig. 2 except for E-ε-l closure. Even in this case there is an evident improvement for the Eεl_2 simulation with respect to Eεl_1, but an underestimation of TKE values appears at both the higher and lower levels.

A better result is found in both simulations Eεl_3 and Eεl_4 by using the ABL sets in the E-ε-l closure, both for the TKE and mean wind profiles, as shown in Fig. 5. As seen in Fig. 3, the simulations have very similar agreements, probably because of the similar values of the constants. A comparison of Figs. 1–5 shows that best results of the different closures are of comparable quality.

Taking into account the best results found in the case of the flat-terrain experiment, we selected the sets of constants and the closure models to perform the simulations in the case of the valley V8. Specifically, we chose the set El_2 for the E-l closure, Eε_2 for the E-ε closure, and Eεl_4 for the E-ε-l closure. In Figs. 6–8, the plots correspond to 5 differ-

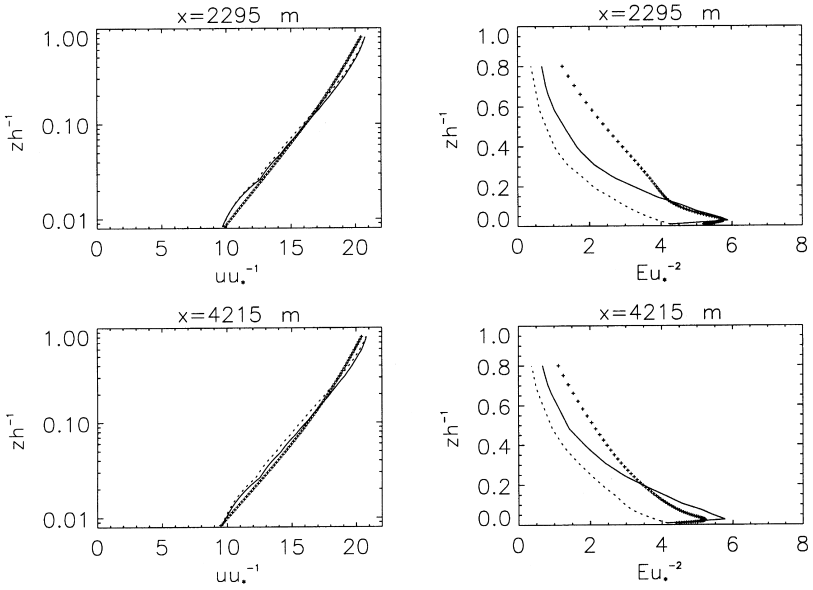

Fig. 1 Comparison of El_1 (dotted line) and El_2 (solid line) simulations (ref. Table 1) against EPA-RUSVAL measured data (crosses) on flat terrain: *u* and TKE normalized profiles.

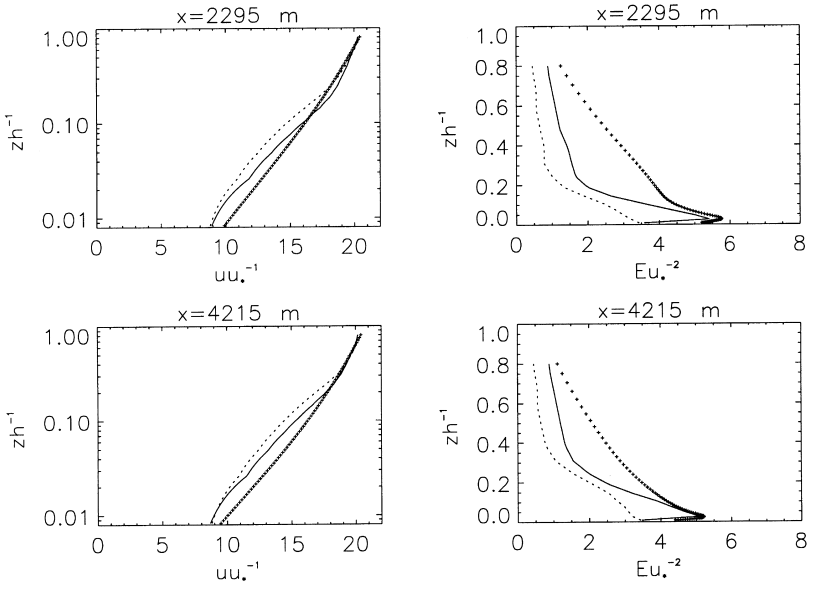

Fig. 2 Comparison of Eε_1 (dotted line) and Eε_2 (solid line) simulations (ref. Table 2) against EPA-RUSVAL measured data (crosses) on flat terrain: *u* and TKE normalized profiles.

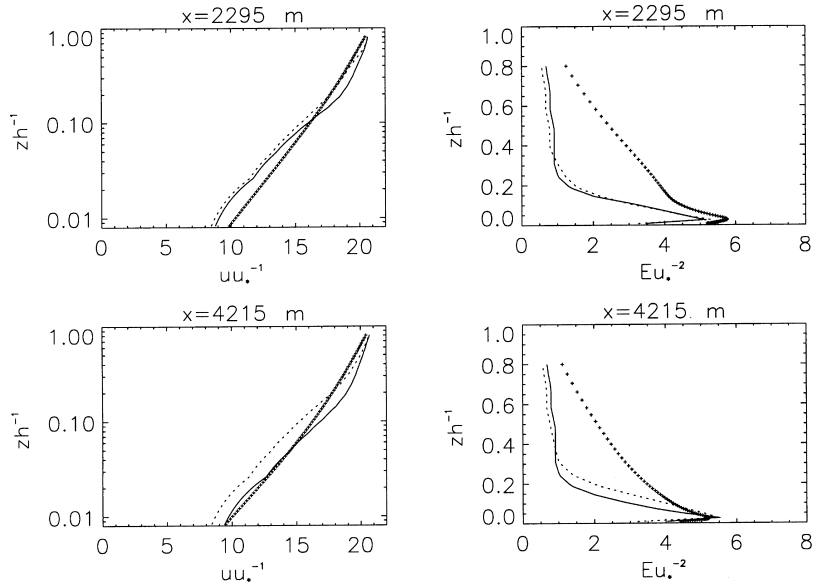

Fig. 3 Comparison of Eε_3 (dotted line) and Eε_4 (solid line) simulations (ref. Table 2) against EPA-RUSVAL measured data (crosses) on flat terrain: *u* and TKE normalized profiles.

ent positions, from upwind to downwind of the valley, normalized by the valley's half width *a* ($x/a = -2$, $x/a = -0.5$, $x/a = 0$, $x/a = 0.5$, and $x/a = 2$).

In Figs. 6 and 7, respectively, the vertical profiles of normalized horizontal and vertical wind velocity components are plotted versus the normalized height. The agreement of all the simulations with the different closures can be considered good at all the locations.

In Fig. 8, normalized TKE vertical profiles are depicted at the same five locations. As a general result, the best agreement is obtained with the E-l model. From a more detailed perspective, at the higher levels the underestimation of the measured values is less than that found with the other two models. This underestimation, shown by all the closures, may be related to the fact that all the constants have been estimated on the basis of surface-layer parameters and evaluated for flat terrain. Some adjustments of the constants are probably needed to account for the characteristics of the whole boundary layer.

At the valley bottom ($x/a = 0$), the TKE maximum is better captured by the E-l and E-ε closures than by the E-ε-l closure. At the second point ($x/a = -0.5$), all three models show a maximum smaller than the measured one, whereas at the third point ($x/a = 0.5$) an underestimation occurs.

The E-ε closure seems to give slightly better results than the E-ε-l closure, but the former needs a much longer simulation time to attain its equilibrium. This is probably related to a small initial instability of the diffusion coefficient defined as proportional to the ratio between TKE and its dissipation rate. These last two quantities, resulting from prognostic equations, can experience some fluctuations.

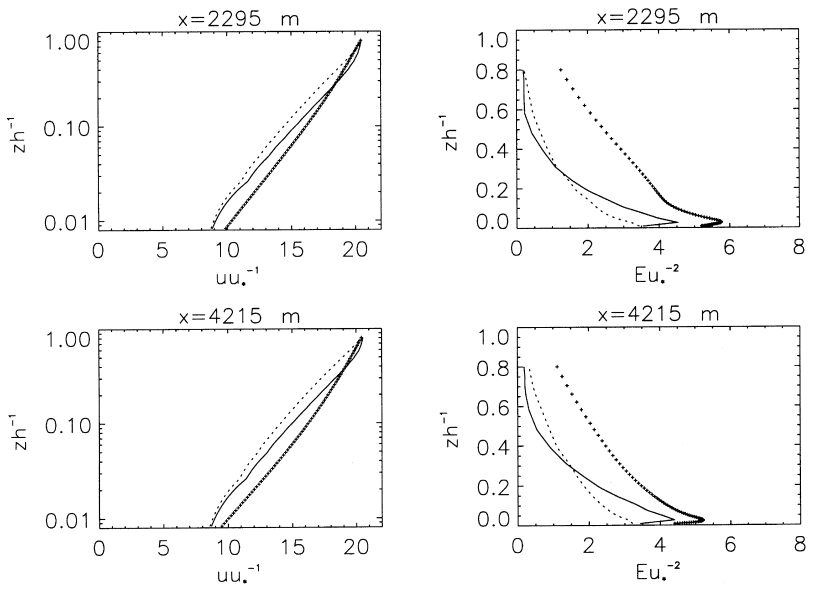

Fig. 4 Comparison of Eεl_1 (dotted line) and Eεl_2 (solid line) simulations (ref. Table 3) against EPA-RUSVAL measured data (crosses) on flat terrain: *u* and TKE normalized profiles.

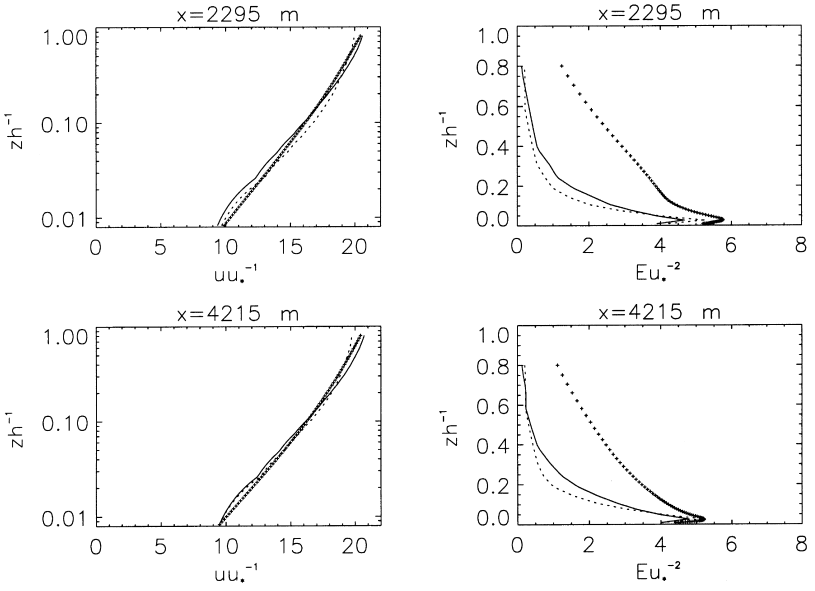

Fig. 5 Comparison of Eεl_3 (dotted line) and Eεl_4 (solid line) simulations (ref. Table 3) against EPA-RUSVAL measured data (crosses) on flat terrain: *u* and TKE normalized profiles.

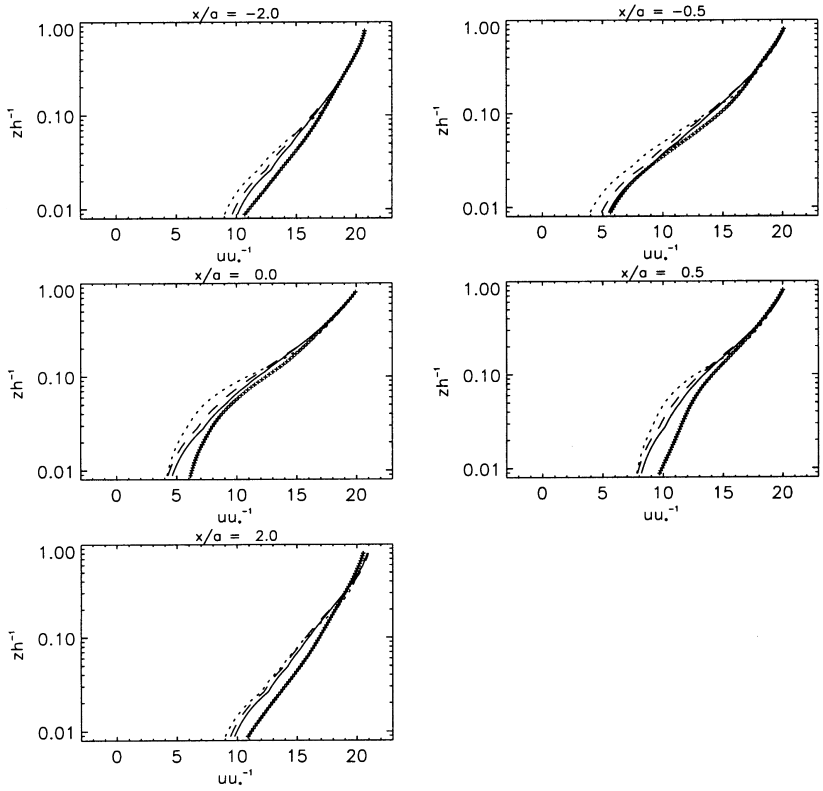

Fig. 6 Comparison of El_2 (solid line), Eε_2 (dotted line), and Eεl_4 (dashed line) simulations against EPA-RUSVAL measured data (crosses) over the valley V8: *u* normalized profiles.

CONCLUSIONS

In this work, a sensitivity analysis compared three different turbulence closure schemes implemented and applied in the flow model RAMS. The three closure models were based on a prognostic equation for TKE and, respectively, a diagnostic equation for the mixing length (E-l model), a prognostic equation for the dissipation rate (E-ε model), or on both equations (E-ε-l model).

The influence of the empirical constants defining the closures on the ability of the models to simulate flow and turbulence fields was evaluated. To investigate the accuracy of these values, we selected two relative sets of constants widely used in literature applications—one for laboratory experiments and the other for the ABL. For all the closures, we compared the results obtained by using the chosen sets with those obtained by using new values of the constants, which we prescribed in previous works.

The data set adopted for the comparison was collected during the EPA-RUSVAL wind tunnel experiment, where a neutral flow on flat terrain and over a two-dimensional

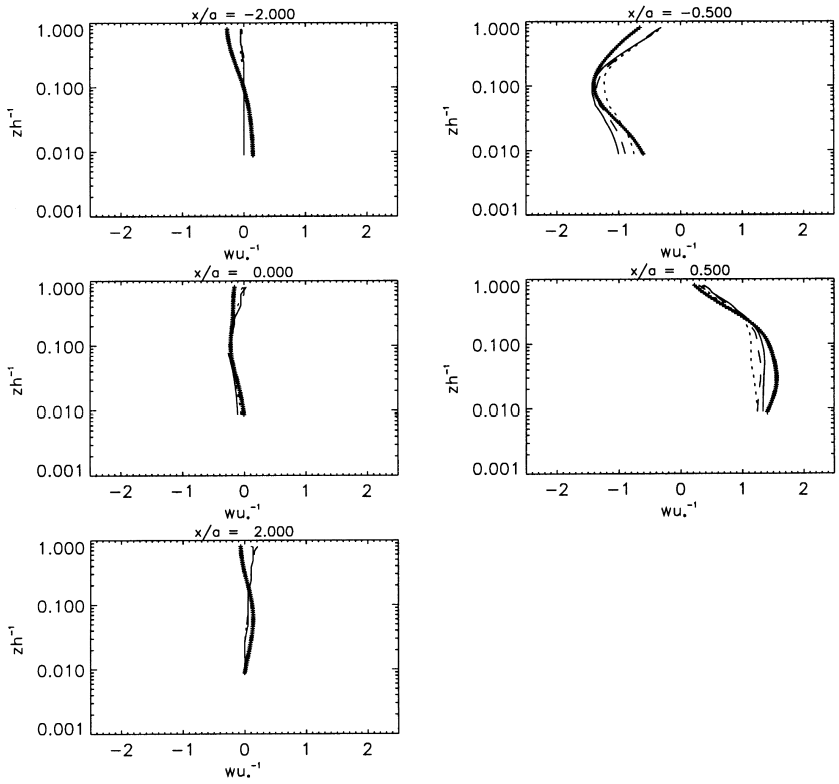

Fig. 7 Comparison of El_2 (solid line), Eε_2 (dotted line), and Eεl_4 (dashed line) simulations against EPA-RUSVAL measured data (crosses) over the valley V8: *u* normalized profiles.

valley was simulated. In the simulations with the fluid-dynamical model RAMS, the tunnel boundary layer was reported to the scale of an actual boundary layer.

Our modified set of constants was estimated according to the similarity theory and surface layer concepts for the neutral stratification and taking into account the turbulence measured in the wind tunnel. The comparison was made in terms of vertical profiles of the horizontal velocity and turbulent kinetic energy.

In the case of flat terrain, the analysis showed that simulations with our set gave much better results with respect to the laboratory set and comparable results in the case of the atmospheric set, in which the values were more similar to ours. All the closures showed good agreement with the observed flow. Turbulence profiles in the surface layer were also well reproduced, whereas some discrepancies appeared at the upper layers. In general, the E-l model gave better results than the E-ε-l and E-ε closures. This result was confirmed when the sets providing the best simulations in the flat-terrain analysis were used in the case of the valley.

Always keeping in mind the schematic characteristics of the simulated experiment, a simpler model, like the E-l, which is defined by two constants only, seemed to be more robust than more refined models, where more empirical constants have to be defined and

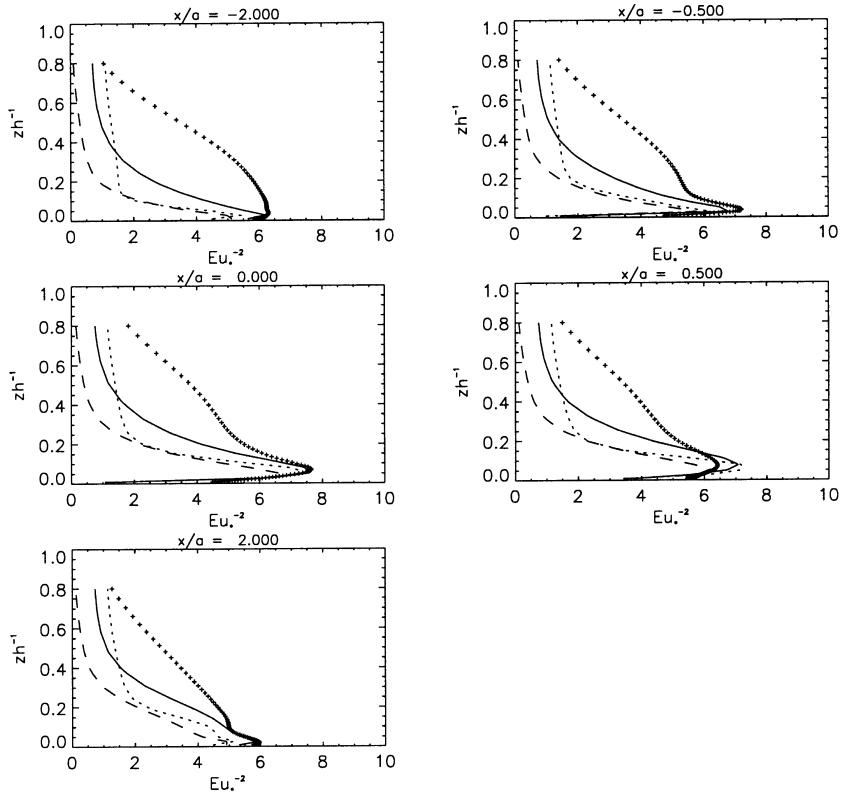

Fig. 8 Comparison of El_2 (solid line), Eε_2 (dotted line), and Eεl_4 (dashed line) simulations against EPA-RUSVAL measured data (crosses) over the valley V8: TKE normalized profiles.

estimated. As a consequence, the increased complexity of the closure models might affect their reliability. Moreover, the difficulty of expressing several terms appearing in the TKE dissipation rate ε differential equation is a very crucial topic in turbulence modeling and at the present is still an open problem.

This analysis shows that estimating the constants by means of known physical quantities of the surface layer was successful in improving the quality of the simulations because it adapts the closure to the specific case. On the basis of this conclusion, and looking at the discrepancies between observed and predicted profiles at the higher levels, further improvement of this method should be possible by considering the physical characteristics of the whole boundary layer.

REFERENCES

1. Busuoli, M., Trombetti, F., & Tampieri, F. (1993). Data set for studies of flow and dispersion in complex terrain. II. The "RUSVAL" wind tunnel experiment (flow data). Technical Paper No. 3, FIS-TP-93/1.

2. Detering, H. W., & Etling, D. (1985). Application of the E-ε turbulence model to the atmospheric boundary layer. *Boundary-Layer Meteorol., 33,* 113–133.
3. Khurshudyan, L. H., Snyder, W. H., & Nekrasov, I. V. (1981). Flow and dispersion of pollutants within two-dimensional hills. U.S. Environmental Protection Agency Report 6000/4-81/067, Research Triangle Park, NC.
4. Khurshudyan, L. H., Snyder, W. H., Nekrasov, I. V., Lawson, R. E., Thompson, R. S, & Schiermeier, F. A. (1990). Flow and dispersion of pollutants within two-dimensional valleys, Summary report on joint Soviet–American study. U.S. Environmental Protection Agency Report 600/3-90/025, Research Triangle Park, NC.
5. Launder, B. E., & Spalding, D. B. (1974). The numericalcomputations of turbulent flows. *Comp. Meth. in Appl. Mech. and Eng.*, *3*, 269–289.
6. Mellor, G. L., & Yamada, T. (1982). Development of a turbulence closure model for geophysical fluid problems. *Rev. Geopys. and Space Physics*, *20*, 851–875.
7. Pielke, R. A., Cotton, W. R., Walko, R. L., Tremback, C. J., Lyons, W. A., Grasso, L. D., Nicholls, M. E., Moran, M. D., Wesley, D. A., Lee, T. J., & Copeland, J. H. (1992). A comprehensive meteorological modeling system—RAMS. *Meteorology and Atmospheric Physics*, *49*, 69–91.
8. Rodi, W. (1980). Turbulence models and their application in hydraulics—a state of the art review, Report. Institut fur Hydromechanik and University of Karlsruhe, Karlsruhe, Germany.
9. Smagorinsky, J. (1963). General circulation experiments with the primitive equations. *Mon. Weath. Rev.*, *91*(3), 99–164.
10. Trini Castelli, S, Ferrero, E., & Anfossi, D. (1997). Comparison between different turbulence closures in a flow model applied to a schematic 2-D valley in a wind tunnel experiment. In Solari G. (Ed.), *Proceedings of Second European and African Conference on Wind Engineering*, Genova, Italy, *1*, 317–324.
11. Trini Castelli, S., Ferrero, E., Anfossi, D., & Ying, R. (1999). Comparison of turbulence closure models over a schematic valley in a neutral boundary layer, *Proceeding of the 13th Symposium on Boundary Layers and Turbulence—79th AMS Annual Meeting*, 601–604.
12. Ying, R. (1992). Research program aiming at establishing the turbulence parameters necessary to achieve a fluidodynamics code. Progress Report No. 2, NASA Goddard Institute for Space Studies.

Hybrid Methods in Engineering, vol. 3, pp. 25–51, 2001

STOCHASTIC THIRD-ORDER HYBRID STRESS-BASED FINITE ELEMENT METHOD

Marcin Kamiński
Division of Mechanics of Materials, Technical University of Łódź, Al. Politechniki 6, 93-590 Łódź, Poland; marcin@kmm-lx.p.lodz.pl; http://kmm.p.lodz.pl/Marcin_Kaminski

This article discuses the application of the second-order perturbation, third probabilistic moment approach in the hybrid stress-based finite element analysis. The approach is demonstrated through the example of the linear elastic heterogeneous medium, where the additional stochastic finite element method discretization is based on the Airy, Prandtl, and Goursat stress functions. The numerical examples shown in the article illustrate the probabilistic stress state in some engineering structures with randomly varying material and geometrical parameters. The results obtained in the computational experiments can be directly used in the third-order reliability method analyses of various engineering problems having any closed-form mathematical or approximative numerical solutions.

KEYWORDS: stress-based finite element method, second-order third moment stochastic perturbation method, Airy and Prandtl functions, Monte Carlo simulation, stochastic finite element method

INTRODUCTION

The equilibrium problems of solids and structures with random parameters or under random excitations have been studied very extensively by numerous authors in the context of probabilistic static and dynamic response in both linear and nonlinear ranges taking into account the formulation and verification of various reliability criteria [9, 14, 18]. The technique is to use the Monte Carlo simulation (MCS) technique, stochastic spectral approaches, stochastic weighted residuals methods, or, alternatively, stochastic perturbation techniques [10, 12, 15]. Because of the computational time savings in comparison with simulation methods, especially in the case of large-scale engineering systems discrete modeling, and taking into account the opportunity of spatial discretization of random probabilistic characteristics fields, the second-order perturbation, second probabilistic moment method is used with its numerical implementation, called the stochastic finite element method (SFEM). The method has been successfully applied in the displacement-based finite element method [23] in elastostatic [21], elastodynamic analyses as well as in inelastic and transient heat transfer problems [12] for both homogeneous and heterogeneous media. Its extension to the random equilibrium stress analysis is discussed in this article.

The main motivation behind the extension of the traditional stochastic second-order, second-moment (SOSM) approach to the third probabilistic moment method (SOTM) is

The author would like to acknowledge the financial support of the Foundation for Polish Science for his postdoctoral research at Rice University in Houston, Texas

the general lack of a skewness parameter in the first approach. This lack practically eliminates most probabilistic engineering problem solutions where nonsymmetric random variables or processes appear. Considering the fact that the same perturbation order is used in both methods, zero-, first-, and second-order solutions of the original problem are exactly the same; however, the formulas for the second- and third-order probabilistic moments are decisively extended in comparison with the corresponding SOSM equations. Therefore, by implementing the methodology described here, one can use the results of stochastic structural SOSM-based computations or those obtained for heat transfer in randomly uncertain media, finally, to determine the third-order coefficients without substantial modifications of the numerical approach. The application of the Weibull probability density function (PDF)-based SOTM in conjunction with the stress-based finite element method [22] may be especially valuable in stochastic reliability analysis where Weibull variables as well as stress-based reliability limit functions are frequently used.

As is demonstrated in numerical analysis, the third-order probabilistic moment method is more accurate than the SOSM method and is more relevant for application in Weibull-based reliability analysis. At the same time, the output results depend on the individual interrelations between probabilistic characteristics up to the third order. The limitations of this approach must be precisely determined with respect to all these coefficient values. Because many reliability criteria are defined in terms of allowable and actual stresses, further implementations of and numerical experiments with the SOTM methodology may be a very promising area of modern computational engineering.

GOVERNING EQUATIONS

Deterministic Stress Equilibrium Problem

Let us consider the set $\Omega \subset \Re^3$ bounded by a regular and sufficiently smooth boundary $\partial\Omega$ and assume that there is a heterogeneous linear elastic medium in Ω built from n homogeneous and coherent components. The compliance tensor is introduced as a function of engineering constants, that is, Young's moduli and Poisson's ratios of the structural components

$$c_{ijkl}(\mathbf{x}) = \chi_a(\mathbf{x})\, c^{(a)}_{ijkl} \quad \text{for } i, j, k, l = 1,2;\ a = 1, \ldots, n \tag{1}$$

with

$$\chi_a(\mathbf{x}) = \begin{cases} 1; \mathbf{x} \in \Omega_a \\ 0; \mathbf{x} \in \Omega / \Omega_a \end{cases} \tag{2}$$

and

$$c^{(a)}_{ijkl} = \frac{1}{E_a}\left((1+\nu_a)\delta_{ik}\delta_{jl} - \nu_a(1+\kappa\nu_a)\delta_{ij}\delta_{kl}\right) \qquad \kappa = 0, 1 \tag{3}$$

for the plane stress/strain analysis.

Next, the following equilibrium problem is considered

$$\sigma_{ij,j} = 0; \quad \mathbf{x} \in \Omega \tag{4}$$

$$\varepsilon_{ij} = c_{ijkl}\sigma_{kl}; \quad \mathbf{x} \in \Omega \tag{5}$$

$$\varepsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right); \quad \mathbf{x} \in \Omega \tag{6}$$

$$u_i = \hat{u}_i; \quad \mathbf{x} \in \partial\Omega_u \tag{7}$$

$$\sigma_{ij} n_j = \hat{t}_i; \quad \mathbf{x} \in \partial\Omega_\sigma \tag{8}$$

where

$$\partial\Omega_\sigma \cup \partial\Omega_u = \partial\Omega \tag{9}$$

$$\partial\Omega_\sigma \cap \partial\Omega_u = \varnothing \tag{10}$$

Let us introduce the statically admissible stresses space S_0 such that

$$S_0 = \left\{ \boldsymbol{\sigma} \in \left[L^2(\Omega) \right]^9 : \sigma_{ij} = \sigma_{ji}; \sigma_{ij,j} = 0 : \mathbf{x} \in \Omega; \sigma_{ij} n_j = 0 : \mathbf{x} \in \partial\Omega_\sigma \right\} \tag{11}$$

Multiplying Eq. (6) by $\delta\sigma_{ij}$ and integrating over Ω, we arrive at

$$\int_\Omega \left(\varepsilon_{ij} - \tfrac{1}{2}\left(u_{i,j} + u_{j,i}\right) \right) \delta\sigma_{ij} d\Omega = 0 \tag{12}$$

The complementary energy principle is obtained for any $\delta\sigma_{ij} \in S_0$

$$\int_\Omega \varepsilon_{ij} \delta\sigma_{ij} d\Omega - \int_{\partial\Omega_u} \hat{u}_i \delta\sigma_{ij} n_j d(\partial\Omega) = 0 \tag{13}$$

or, alternatively

$$\Sigma(\sigma) = \tfrac{1}{2} \int_\Omega c_{ijkl}\sigma_{ij}\sigma_{kl} d\Omega - \int_{\partial\Omega_u} \hat{u}_i \sigma_{ij} n_j d(\partial\Omega) \tag{14}$$

which, after minimization, leads to the real stress field being a solution for the equilibrium problem, Eqs. (4)–(10).

The preceding equations make possible the solving of the boundary problem with deterministic or random coefficients by using the second-order perturbation, second probabilistic moment method as is shown in the next paragraph. It should be emphasized that the method is quite general in character and makes possible the randomization of any differential or algebraic equations with respect to perturbations up to the second order of the problem parameters as well as probabilistic moments up to the second order of all the state variables.

An analogous formulation may be proposed for torsion of a linear, isotropic, and homogeneous medium formulated in terms of a warping function φ

$$\frac{1}{G}\left(\frac{\partial^2\varphi}{\partial x^2}+\frac{\partial^2\varphi}{\partial y^2}\right)=Q(x,y) \tag{15}$$

where $Q(x, y)$ represents any external load. The stress tensor components can be introduced as

$$\sigma_{13}=\frac{\partial\varphi}{\partial x_2} \qquad \sigma_{23}=-\frac{\partial\varphi}{\partial x_1} \tag{16}$$

and the constitutive relation can be rewritten as

$$\frac{\partial\varphi}{\partial x_i}=G\theta\left(\frac{\partial\psi}{\partial x_j}-x_i\right) \tag{17}$$

in the case of an externally applied twisting moment $Q(x, y) = -2\theta$ where Ψ is a warping function. Finally, the complementary energy necessary to finite element discretization can be expressed as

$$\Sigma(\varphi)=\int_{\Omega}\frac{1}{2G}\left[\left(\frac{\partial\varphi}{\partial x_1}\right)^2+\left(\frac{\partial\varphi}{\partial x_2}\right)^2\right]d\Omega-\int_{\partial\Omega_u}Q\varphi d\Omega \tag{18}$$

As is known, the torsion problem as a so-called field problem is equivalent to various related physical phenomena characterized by the same Laplace partial differential equation as the heat conduction analysis, for example [8].

Second-Order Perturbation Third Probabilistic Moment Approach

Let us denote random variables of the problem in the form of a $\{b^r(\mathbf{x};\omega)\}$ and its probability density as $g(b^r)$ and $g(b^r, b^s)$ respectively, where $r, s = 1, 2, \ldots, R$. The expected

values, cross-covariances, and skewness of this vector components are calculated as follows [7, 20]

$$E[b^r] = \int_{-\infty}^{+\infty} b^r g(b^r) db^r \tag{19}$$

$$Cov(b^r, b^s) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (b^r - E[b^r])(b^s - E[b^s]) g(b^r, b^s) db^r db^s \tag{20}$$

$$S(u_i) = \frac{1}{\sigma^3(u_i)} \int_{-\infty}^{+\infty} \left(u_i - E[u_i]\right)^3 p_R(b) db \tag{21}$$

In further applications, the Weibull distribution is used with its probability density function

$$p_R = \begin{cases} \dfrac{\beta}{\lambda}\left(\dfrac{x-\bar{x}}{\lambda}\right)^{\beta-1} \exp\left[-\left(\dfrac{x-\bar{x}}{\lambda}\right)^{\beta}\right]; & x > \bar{x} \\ 0, & x < \bar{x} \end{cases} \tag{22}$$

where β is the Weibull shape parameter, λ denotes the scale parameter, and $\bar{x}$ is the location parameter, which indicates the smallest value of the random variable x for which the probability density function is positive. Starting from this definition, the Weibull PDF has been used for general mechanical applications where many random variables must be nonnegative (Young modulus, some geometrical parameters, for example) and, especially, for composite failure and fatigue modeling. Let us note that if discrete representation of random variable $b(x; \omega)$ is used, then statistical estimators may be applied to approximate any order probabilistic moments of this variable [1].

Next, the following stochastic Taylor series expansion is applied [15–17]

$$F(\boldsymbol{\xi};\omega) = F^0(\boldsymbol{\xi};\omega) + \theta F'^r(\boldsymbol{\xi};\omega)\Delta b^r + \frac{1}{2}\theta^2 F'^{rs}(\boldsymbol{\xi};\omega)\Delta b^r \Delta b^s \tag{23}$$

where θ is a given small perturbation, $\theta\Delta b^{r1}$ denotes the first-order variation of Δb^{r1} about its expected value $E[b^{r1}]$, and $F^{(n)}(x; \omega)$ represents the nth-order partial derivatives with respect to the random variables evaluated at their expected values. To obtain either SOSM or SOTM model (or both) for the stress-based FEM, Eq. (13) is rewritten in the following form

$$\int_{\Omega} c_{ijkl}\sigma_{kl}\delta\sigma_{ij} d\Omega = \int_{\partial\Omega_u} \hat{u}_i \delta\sigma_{ij} n_j d(\partial\Omega) \tag{24}$$

Therefore, up to the second-order variational statements are obtained in conjunction with Eq. (23), as:

- Zero-order (ε^0 terms, one equation)

$$\int_{\Omega} c^0_{ijkl}\sigma^0_{kl}\delta\sigma_{ij}d\Omega = \int_{\partial\Omega_u} \hat{u}^0_i \delta\sigma_{ij}n_j d(\partial\Omega) \tag{25}$$

- First-order (ε^1 terms, R equations)

$$\int_{\Omega} c^0_{ijkl}\sigma'^r_{kl}\delta\sigma_{ij}d\Omega = \int_{\partial\Omega_u} \hat{u}'^r_i \delta\sigma_{ij}n_j d(\partial\Omega) - \int_{\Omega} c'^r_{ijkl}\sigma^0_{kl}\delta\sigma_{ij}d\Omega \tag{26}$$

- Second-order (ε^2 terms, one equation)

$$\int_{\Omega} c^0_{ijkl}\sigma'^{rs}_{kl}S^{rs}_b\delta\sigma_{ij}d\Omega = \int_{\partial\Omega_u} \hat{u}'^{rs}_i S^{rs}_b\delta\sigma_{ij}n_j d(\partial\Omega) - \int_{\Omega} \left(2c'^r_{ijkl}\sigma^s_{kl} + c'^{rs}_{ijkl}\sigma^0_{kl}\right) S^{rs}_b\delta\sigma_{ij}d\Omega \tag{27}$$

Note that the second-order equation is obtained here by multiplying the R-variate probability density function $p_R(b_1, b_2, \ldots, b_R)$ by the ε^2-terms and integrating over the domain of the random field variables $b(x_k)$. It holds that

$$\begin{aligned}
&\int_{-\infty}^{+\infty}\left[\int_{\partial\Omega_u} \varepsilon^2\hat{u}'^{rs}_i\left[b^0(x_k);x_k\right]\Delta b_r(x_k)\Delta b_s(x_k)\delta\sigma_{ij}n_j d(\partial\Omega)\right] p_R\left(b(x_k)\right)db \\
&= \varepsilon^2\int_{\partial\Omega_u} \hat{u}'^{rs}_i\left[b^0(x_k);x_k\right]\left[\int_{-\infty}^{+\infty}\Delta b_r(x_k)\Delta b_s(x_k)\,p_R\left(b(x_k)\right)db\right]\delta\sigma_{ij}n_j d(\partial\Omega) \\
&= \varepsilon^2\int_{\partial\Omega_u} \hat{u}'^{rs}_i\left[b^0(x_k);x_k\right]S^{rs}_b\delta\sigma_{ij}n_j d(\partial\Omega)
\end{aligned} \tag{28}$$

Next, Eq. (25) is solved for σ^0_{kl}; then Eq. (26) is solved for the first-order terms of σ_{kl}; and, finally, Eq. (27) is solved for $\sigma^{(2)}_{kl}$. The probabilistic third-moment characterization of the equilibrium state functions starts from the expected value of the stress tensor components

$$E\left[\sigma_{kl}\left[b(x_k);x_k\right]\right] = \int_{-\infty}^{+\infty}\sigma_{kl}\left[b(x_k);x_k\right]p_R\left(b(x_k)\right)db$$

$$= \int_{-\infty}^{+\infty} \left\{ \sigma_{kl}^{0}\left[b(x_k);x_k\right] + \sigma_{kl}^{\prime r}\left[b(x_k);x_k\right]\Delta b_r(x_k) \right.$$

$$\left. + \frac{1}{2}\sigma_{kl}^{\prime rs}\left[b(x_k);x_k\right]\Delta b_r(x_k)\Delta b_s(x_k)\right\} p_R\left(b(x_k)\right) db \tag{29}$$

which finally gives

$$E\left[\sigma_{kl}\right] = \sigma_{kl}^{0} + \frac{1}{2}\sigma_{kl}^{\prime rs}\, S_b^{rs} \tag{30}$$

where $\varepsilon = 1$ is applied; further improvements of the perturbation method with respect to the perturbation order and parameter are shown in Elishakoff et al. [5].

The next two probabilistic moments are obtained in a quite similar manner as

$$Var\left(\sigma_{kl}\right) = \left(\sigma_{kl}^{0}\right)^2 + \sum_{i=1}^{n}\left[\left(\frac{\partial \sigma_{kl}}{\partial b_i}\right)^2 + \sigma_{kl}^{0}\frac{\partial^2 \sigma_{kl}}{\partial b_i^2}\right]\sigma^2\left(b_i\right)$$

$$+ \sum_{i=1}^{n}\left(\frac{\partial \sigma_{kl}}{\partial b_i}\frac{\partial^2 \sigma_{kl}}{\partial b_i^2}\right) S\left(b_i\right)\sigma^3\left(b_i\right) - E^2\left[\sigma_{kl}\right] \tag{31}$$

and

$$S\left(\sigma_{kl}\right) = \left\{\left(\sigma_{kl}^{0}\right)^3 + \frac{3}{2}\sum_{i=1}^{n}\left[2\sigma_{kl}^{0}\left(\frac{\partial \sigma_{kl}}{\partial b_i}\right)^2 + \left(\sigma_{kl}^{0}\right)^2\frac{\partial^2 \sigma_{kl}}{\partial b_i^2}\right]\sigma^2\left(b_i\right)\right.$$

$$\left. + \sum_{i=1}^{n}\left[\left(\frac{\partial \sigma_{kl}}{\partial b_i}\right)^3 + 3\sigma_{kl}^{0}\frac{\partial \sigma_{kl}}{\partial b_i}\frac{\partial^2 \sigma_{kl}}{\partial b_i^2}\right] S\left(b_i\right)\sigma^3\left(b_i\right) - E^3[\sigma_{kl}] - 3E[\sigma_{kl}]\sigma^2(\sigma_{kl})\right\}\frac{1}{\sigma^3(\sigma_{kl})} \tag{32}$$

To determine the first two probabilistic moments for the strain tensor components, the second-order perturbations are inserted into the constitutive relation (1) as

$$\varepsilon_{ij} = c_{ijkl}\sigma_{kl}$$

$$= \left(c_{ijkl}^{0}\left[b(x_k);x_k\right] + c_{ijkl}^{\prime r}\left[b(x_k);x_k\right]\Delta b_r(x_k) + \frac{1}{2}c_{ijkl}^{\prime rs}\left[b(x_k);x_k\right]\Delta b_r(x_k)\Delta b_s(x_k)\right)$$

$$\times\left(\sigma_{kl}^{0}\left[b(x_k);x_k\right] + \sigma_{kl}^{\prime r}\left[b(x_k);x_k\right]\Delta b_r(x_k) + \frac{1}{2}\sigma_{kl}^{\prime rs}\left[b(x_k);x_k\right]\Delta b_r(x_k)\Delta b_s(x_k)\right) \tag{33}$$

Weibull Second-Order, Third-Moment Reliability Analysis

Having computed the first three probabilistic moments of contact stresses (expected values, standard deviations, and skewness coefficients), the random field of limit function $g(z; \omega)$ must be proposed. Usually, it can be introduced as a difference between allowable and actual stresses $\sigma_z(z; \omega)$ in the specimen as

$$g(z;\omega) = \sigma_{\text{all}}(\omega) - \sigma_z(z;\omega) \tag{34}$$

Let us underline that allowable stresses are most frequently analyzed as random variables, whereas actual stresses are random fields, and that is why the computational analysis is carried out here for the specific value of the vertical coordinate z. The random allowable stresses $\sigma_{\text{all}}(\omega)$ are specified by the use of the first three probabilistic moments $E[\sigma_{\text{all}}(\omega)]$, $Var[\sigma_{\text{all}}(\omega)]$, and $S[\sigma_{\text{all}}(\omega)]$.

The corresponding probabilistic characteristics of the limit function are calculated as

$$E[g] = g^0 + \frac{1}{2}\sum_{i=1}^{n}\left(\frac{\partial^2 g}{\partial b_i^2}\right)\sigma^2(b_i) \tag{35}$$

$$\sigma^2(g) = (g^0)^2 + \sum_{i=1}^{n}\left[\left(\frac{\partial g}{\partial b_i}\right)^2 + g^0\frac{\partial^2 g}{\partial b_i^2}\right]\sigma^2(b_i)$$

$$+\sum_{i=1}^{n}\left(\frac{\partial g}{\partial b_i}\frac{\partial^2 g}{\partial b_i^2}\right)S(b_i)\sigma^3(b_i) - E^2[g] \tag{36}$$

and

$$S(g) = \left\{(g^0)^3 + \frac{3}{2}\sum_{i=1}^{n}\left[2g^0\left(\frac{\partial g}{\partial b_i}\right)^2 + (g^0)^2\frac{\partial^2 g}{\partial b_i^2}\right]\sigma^2(b_i)\right.$$

$$\left.+\sum_{i=1}^{n}\left[\left(\frac{\partial g}{\partial b_i}\right)^3 + 3g^0\frac{\partial g}{\partial b_i}\frac{\partial^2 g}{\partial b_i^2}\right]S(b_i)\sigma^3(b_i) - E^3[g] - 3E[g]\sigma^2(g)\right\}\frac{1}{\sigma^3(g)} \tag{37}$$

Inserting the limit state function g from Eq. (34) into Eqs. (35)–(37) and assuming that the random variable of allowable stresses and the random field of actual stresses are uncorrelated, we obtain

$$E[g] = \sigma_{all}^0 - \sigma_z^0 - \frac{1}{2}\sum_{i=1}^{n}\left(\frac{\partial^2 \sigma_z}{\partial b_i^2}\right)\sigma^2(b_i) \tag{38}$$

$$\sigma^2(g) = \left(\sigma_{\text{all}}^0 - \sigma_z^0\right)^2 + \sum_{i=1}^{n}\left[\left(\frac{\partial \sigma_z}{\partial b_i}\right)^2 - \left(\sigma_{\text{all}}^0 - \sigma_z^0\right)\frac{\partial^2 \sigma_z}{\partial b_i^2}\right]\sigma^2(b_i)$$

$$+\sum_{i=1}^{n}\left(\frac{\partial \sigma_z}{\partial b_i}\frac{\partial^2 \sigma_z}{\partial b_i^2}\right)S(b_i)\sigma^3(b_i) - E^2\left[\sigma_{\text{all}} - \sigma_z\right] \tag{39}$$

and, finally

$$S(g) = \left\{\left(\sigma_{\text{all}}^0 - \sigma_z^0\right)^3 + \frac{3}{2}\sum_{i=1}^{n}\left[2\left(\sigma_{\text{all}}^0 - \sigma_z^0\right)\left(\frac{\partial \sigma_z}{\partial b_i}\right)^2 + \left(\sigma_{\text{all}}^0 - \sigma_z^0\right)^2\frac{\partial^2 \sigma_z}{\partial b_i^2}\right]\sigma^2(b_i)\right.$$

$$+\sum_{i=1}^{n}\left[\left(\frac{\partial g}{\partial b_i}\right)^3 + 3g^0\frac{\partial g}{\partial b_i}\frac{\partial^2 g}{\partial b_i^2}\right]S(b_i)\sigma^3(b_i)$$

$$\left. - E^3\left[\sigma_{\text{all}}^0 - \sigma_z^0\right] - 3E\left[\sigma_{\text{all}}^0 - \sigma_z^0\right]\sigma^2\left(\sigma_{\text{all}}^0 - \sigma_z^0\right)\right\}\frac{1}{\sigma^3\left(\sigma_{\text{all}}^0 - \sigma_z^0\right)} \tag{40}$$

Comparing the second-order, second-moment approach with the second-order, third-moment approach, it is apparent that the expected values are described by exactly the same equation, whereas standard deviations (or variances) have some extra components connected with skewness of the analyzed PDF; third-order parameter of the probabilistic output is introduced in the SOTM-based analysis. Finally, the parameters $x(g)$, $\lambda(g)$, and $\beta(g)$ of the equivalent Weibull PDF are derived from the following system of equations

$$E[g] = \lambda(g)\Gamma\left(1 + \frac{1}{\beta(g)}\right) + x(g) \tag{41}$$

$$\sigma^2(g) = \lambda^2(g)\left[\Gamma\left(1 + \frac{2}{\beta(g)}\right) - \Gamma^2\left(1 + \frac{1}{\beta(g)}\right)\right] \tag{42}$$

$$S(g) = \lambda^3(g)\left[\Gamma\left(1 + \frac{3}{\beta(g)}\right) - 3\Gamma\left(1 + \frac{2}{\beta(g)}\right)\Gamma\left(1 + \frac{1}{\beta(g)}\right)\right]\frac{1}{\sigma^2(g)}$$

$$+2\Gamma^3\left(1 + \frac{1}{\beta(g)}\right)\frac{1}{\sigma^2(g)} \tag{43}$$

where the gamma function is defined as

$$\Gamma(x)=\begin{cases}\int_0^{\infty} e^{-t}t^{x-1}dt \quad (\text{for} \quad x>0)\\ \lim\limits_{n\to\infty}\dfrac{n!n^{x-1}}{x(x+1)(x+2)...(x+n-1)} \quad (\text{for any } x\in\Re)\end{cases} \tag{44}$$

It should be mentioned that the symbolic computations approach is the most effective method of solving these equations and computing the equivalent Weibull distribution probabilistic parameters. Finally, the structural reliability index R [9, 14, 18] of a limit function g is calculated from the following formula

$$R(g)=\exp\left[-\left(-\frac{x(g)}{\beta(g)}\right)^{\lambda(g)}\right] \tag{45}$$

The values of this index should behave like the classical probability function—not less than 0 and not greater than 1.

STRESS-BASED FINITE ELEMENT METHOD

Deterministic Approach

The following approximation of the stress tensor components by the use of Airy functions is applied in the case of two-dimensional problems to discretize variational statement (13)

$$\boldsymbol{\sigma}=\begin{bmatrix}\dfrac{\partial^2}{\partial y^2}\\ \dfrac{\partial^2}{\partial x^2}\\ -\dfrac{\partial^2}{\partial x\partial y}\end{bmatrix}\mathbf{F}=\partial^2\mathbf{F} \tag{46}$$

where function $F(x, y)$ may be represented as follows

$$F\left(x,y\right)=\sum_{p,q\le\alpha} d_{pq}x^p y^q \tag{47}$$

with the value of parameter α depending on the stress-based finite element type being used. Next, the interpolation function N is introduced with the degrees of freedom vector **a**, which makes possible the introduction of the following representation

$$\boldsymbol{\sigma} = \partial^2 \mathbf{N}\,\mathbf{a} = \begin{bmatrix} \dfrac{\partial^2 N_1}{\partial y^2} & \dfrac{\partial^2 N_2}{\partial y^2} & \cdots \\ \dfrac{\partial^2 N_1}{\partial x^2} & \dfrac{\partial^2 N_2}{\partial x^2} & \cdots \\ -\dfrac{\partial^2 N_1}{\partial x \partial y} & -\dfrac{\partial^2 N_2}{\partial x \partial y} & \cdots \end{bmatrix} \mathbf{a} = \mathbf{H}\,\mathbf{a} \tag{48}$$

Defining the prescribed displacements vector $\hat{u}_i$ for $\mathbf{x} \in \partial\Omega_u$ and the matrix

$$\mathbf{n} = \begin{bmatrix} n_x & 0 & n_x \\ 0 & n_y & n_y \end{bmatrix} \tag{49}$$

where n_x and n_y denote the components of the unity vector normal to $\partial\Omega$ and directed externally to Ω, Eq. (13) may be rewritten as

$$\delta\mathbf{a}^T \int_{\Omega} \mathbf{H}^T \mathbf{a}\, d\Omega - \delta\mathbf{a}^T \int_{\partial\Omega_u} \mathbf{H}^T \mathbf{n}^T \hat{\mathbf{u}} d(\partial\Omega) = 0 \tag{50}$$

and hence

$$\int_{\Omega} \mathbf{H}^T \mathbf{a} d\Omega - \mathbf{F} = 0 \tag{51}$$

where

$$\mathbf{F} = \int_{\partial\Omega_u} \mathbf{H}^T \mathbf{n}^T \hat{\mathbf{u}} d(\partial\Omega) \tag{52}$$

Finally, it can be written that

$$\mathbf{K}\,\mathbf{a} = \mathbf{F} \tag{53}$$

where $\mathbf{K}$ is a system compliance matrix defined as

$$\mathbf{K} = \int_{\Omega} \mathbf{H}^T \mathbf{c} \mathbf{H} d\Omega \tag{54}$$

Note that in the general case the stress tensor components may be rewritten as

$$\sigma_{ij} = \varepsilon_{ipr}\varepsilon_{jqs}\Phi_{rs,pq} \tag{55}$$

where Φ_{rs} denotes the Maxwell-Morera function components or, using the Goursat function, we have

$$\Phi(x, y) = \mathrm{Re}\left[z\varphi(z) + \chi(z)\right] \tag{56}$$

Therefore, the stress tensor components are defined analogously as for the classical Airy function, so that one can obtain these stresses as

$$\begin{cases} \sigma_x + \sigma_y = 4\,\mathrm{Re}\left[\varphi'(z)\right] \\ \sigma_x + \sigma_y + 2i\tau_{xy} = 2\left[z\varphi''(z) + \lambda''(z)\right] \\ 2G\left(u + iv\right) = \left(\dfrac{3-\nu}{1+\nu}\right)\varphi(z) - z\varphi'(z) - \chi'(z) \end{cases} \tag{57}$$

where $\varphi(z)$ and $\chi(z)$ are taken as

$$\varphi(z) = \sum_{n=0}^{\infty} A_n z^n \qquad \chi(z) = \sum_{n=0}^{\infty} B_n z^n$$

$$A_n = a_n + ib_n \qquad B_n = c_n + id_n \tag{58}$$

Finally, the Hu-Washizu principle [2, 13] is applied in this case to obtain the variational formulation of the equilibrium problem.

The FEM discretization of the torsion problem outlined above—compare Eqs. (15)–(18)—is accomplished by using the constant stress triangular finite elements, where the following discretization for φ is applied

$$\varphi = N_1\varphi_1 + N_2\varphi_2 + N_3\varphi_3 = \mathbf{N}\,\mathbf{a} \tag{59}$$

Therefore, the complementary energy can be represented as

$$\Sigma(\varphi) = \frac{1}{2}\iint_{\Omega} \mathbf{a}^T\mathbf{H}^T d\,\mathbf{H}\mathbf{a}\,d\Omega - \iint_{\Omega} 2\theta\mathbf{H}\mathbf{a}\,d\Omega \tag{60}$$

Minimization of the functional $\Sigma(\varphi)$ crucial for the FEM formulation gives

$$\iint_{\Omega} \mathbf{H}^T d\,\mathbf{H}\,d\Omega\mathbf{a} = \iint_{\Omega} 2\theta\mathbf{N}^T d\Omega \tag{61}$$

where

$$\mathbf{K} = \frac{1}{G}\iint_{\Omega} \mathbf{B}^T\mathbf{B}\, d\Omega \tag{62}$$

for a homogeneous region characterized by Kirchhoff modulus G and an external load vector given in terms of the twisting angle 2θ as

$$\mathbf{F} = 2\theta\iint_{\Omega} \mathbf{N}^T d\Omega \tag{63}$$

Stochastic Stress-Based Finite Elements

Let us consider a space discretization of Ω by a typical finite element mesh to introduce the matrix equations for the second-order and second-moment stochastic analysis. First, the input vector of random variables $b_r(\mathbf{x})$ is discretized in terms of some point values by using the following spatial representation

$$b_r(\mathbf{x}) = \varphi_{r\alpha}(\mathbf{x})\, b_\alpha \qquad r = 1, \ldots, R; \quad \alpha = 1, \ldots, N \tag{64}$$

where α is the shape function for the αth node, N is the number of nodal points in the mesh, and $b_{r\alpha}$ is the matrix of random parameter nodal values. Then, the expected values and cross-covariances are interpolated as

$$E\left[b_r(\mathbf{x})\right] = b_r^0(\mathbf{x}) = \varphi_{r\rho}(\mathbf{x})\, b_\rho^0 \tag{65}$$

$$Cov\left(b_r(\mathbf{x}); b_s(\mathbf{x})\right) = S_b^{rs} = \varphi_{r\rho}(\mathbf{x})\,\varphi_{s\sigma}(\mathbf{x})\, S_b^{\rho\sigma} \tag{66}$$

where

$$\Delta b_r(\mathbf{x}) = \varphi_{r\rho}(\mathbf{x})\,\Delta b_\rho \tag{67}$$

$$\Delta b_\rho = b_\rho - b_\rho^0 \tag{68}$$

b_ρ^0 and $S_b^{\rho\sigma}$ are the random value vector and covariance matrix of the vector, respectively. It should be emphasized that the node spatial discretization of random fields can be introduced as equivalent to the nodal points of the original mesh. Alternatively, the random fields spatial discretization may be carried out by using the additional averaging method (the random variable is defined as the spatial average of the random field over the finite element domain), the midpoint method (the random variable is defined as the value of the random field at the centroid of the element), and, finally, the series expansion method,

where the random field is modeled as the series of shape functions with random coefficients and any field discretization [16]. Next, all material properties and the state variables are expanded by using the same shape functions, as illustrated through the example of the compliance tensor

$$c_{ijkl}\left[b_r(\mathbf{x});\mathbf{x}\right]=c\left[\varphi_{r\rho}(\mathbf{x})\,b_\rho;\mathbf{x}\right]=\varphi_\alpha(\mathbf{x})\,c_{ijkl\alpha}\left(b_\rho\right)\qquad \rho=1,\ldots,R \tag{69}$$

Next, the Taylor series expansion for the nodal random variables is employed in the form of

$$c_{ijkl\alpha}\left(b_\rho\right)=c^0_{ijkl\alpha}\left(b^0_\rho\right)+\varepsilon c'^{\rho}_{ijkl\alpha}\left(b^0_\rho\right)\Delta b_\rho+\frac{1}{2}\varepsilon^2 c'^{\rho\sigma}_{ijkl\alpha}\left(b^0_\rho\right)\Delta b_\rho\Delta b_\sigma \tag{70}$$

and, from both of these equations, it holds that:

$$c_{ijkl\alpha}\left[b_r(\mathbf{x});\mathbf{x}\right]=\varphi_\alpha(\mathbf{x})\left(c^0_{ijkl\alpha}\left(b^0_\rho\right)+\varepsilon c'^{\rho}_{ijkl\alpha}\left(b^0_\rho\right)\Delta b_\rho+\frac{1}{2}\varepsilon^2 c'^{\rho\sigma}_{ijkl\alpha}\left(b^0_\rho\right)\Delta b_\rho\Delta b_\sigma\right) \tag{71}$$

The perturbations up to the second order of the compliance tensor are equal to

$$c^0_{ijkl\alpha}\left[b_r(\mathbf{x});\mathbf{x}\right]=\varphi_\alpha(\mathbf{x})\,c^0_{ijkl\alpha}\left(b^0_\rho\right) \tag{72}$$

$$c'^{r}_{ijkl\alpha}\left[b_r(\mathbf{x});\mathbf{x}\right]\varphi_{r\rho}(\mathbf{x})=\varphi_\alpha(\mathbf{x})\,c'^{\rho}_{ijkl\alpha}\left(b^0_\rho\right) \tag{73}$$

$$c'^{rs}_{ijkl\alpha}\left[b_r(\mathbf{x});\mathbf{x}\right]\varphi_{r\rho}(\mathbf{x})\,\varphi_{s\sigma}(\mathbf{x})=\varphi_\alpha(\mathbf{x})\,c'^{\rho\sigma}_{ijkl\alpha}\left(b^0_\rho\right) \tag{74}$$

Applying the preceding finite element approximations into zero-, first-, and second-order variational statements, the following hierarchical equilibrium equations are obtained:

- Zero-order (ε^0 terms, one system of N linear simultaneous algebraic equations for $a^0_\alpha(b^0_\rho)$, $\alpha=1,\ldots,N$)

$$K^0_{\alpha\beta}\left(b^0_\rho\right)a^0_\beta\left(b^0_\rho\right)=F^0_\alpha\left(b^0_\rho\right) \tag{75}$$

- First-order (ε^1 terms, R systems of N linear simultaneous algebraic equations for $a'^{\rho}_\alpha(b^0_\rho)$, $\rho=1,\ldots,R$; $\alpha=1,\ldots,N$)

$$K^0_{\alpha\beta}\left(b^0_\rho\right)a'^{\rho}_\beta\left(b^0_\rho\right)=F'^{\rho}_\alpha\left(b^0_\rho\right)-K'^{\rho}_{\alpha\beta}\left(b^0_\rho\right)a^0_\beta\left(b^0_\rho\right) \tag{76}$$

- First-order (ε^2 terms, one system of N linear simultaneous algebraic equations for $a_\alpha^{(2)}(b_\rho^0)$)

$$K_{\alpha\beta}^0\left(b_\rho^0\right)a_\beta^{(2)}\left(b_\rho^0\right)=\left[\ F_\alpha'^{\rho\sigma}\left(b_\rho^0\right)-2K_{\alpha\beta}'^{\rho}\left(b_\rho^0\right)a_\beta'^{\sigma}\left(b_\rho^0\right)-K_{\alpha\beta}'^{\rho\sigma}\left(b_\rho^0\right)a_\beta^0\left(b_\rho^0\right)\right]S_b^{\rho\sigma} \tag{77}$$

with

$$a_\beta^{(2)}\left(b_\rho^0\right)=a_\beta'^{\rho\sigma}\left(b_\rho^0\right)S_b^{\rho\sigma} \tag{78}$$

Solving these equations for the zero-, first-, and second-order stress tensor fields and applying the extension

$$a_\alpha\left(b_\rho\right)=a_\alpha^0\left(b_\rho^0\right)+\varepsilon a_\alpha'^{\rho}\left(b_\rho^0\right)\Delta b_\rho+\frac{1}{2}\varepsilon^2 a_\alpha'^{\rho\sigma}\left(b_\rho^0\right)\Delta b_\rho\Delta b_\sigma \tag{79}$$

the expected values and cross-covariances can be calculated as

$$E\left[a_\alpha\right]=a_\alpha^0+\frac{1}{2}a_\alpha'^{\rho\sigma}S_b^{\rho\sigma} \tag{80}$$

$$S_a^{\alpha\beta}=a_\alpha'^{\rho}a_\beta'^{\sigma}S_b^{\rho\sigma} \tag{81}$$

Using analogous methodology, the expected values of strain tensor components are derived as follows

$$E\left[\varepsilon_{ij}\left(\mathbf{x}\right)\right]=c_{ijkl}^0\left(\mathbf{x}\right)H_{kl\alpha}\left(\mathbf{x}\right)a_\alpha^0$$

$$+\frac{1}{2}\left[c_{ijkl}'^{rs}\left(\mathbf{x}\right)H_{kl\alpha}\left(\mathbf{x}\right)a_\alpha^0+2c_{ijkl}'^{r}\left(\mathbf{x}\right)H_{kl\alpha}\left(\mathbf{x}\right)a_\alpha'^{s}+c_{ijkl}^0\left(\mathbf{x}\right)H_{kl\alpha}\left(\mathbf{x}\right)a_\alpha'^{rs}\right]S_b^{rs} \tag{82}$$

At the same time, the first-order strain tensor components cross-covariance has the following form

$$Cov\left(\varepsilon_{ij}\left(\mathbf{x}\right);\varepsilon_{kl}\left(\mathbf{x}\right)\right)$$

$$=\left[c_{ijmn}'^{r}\left(\mathbf{x}\right)c_{ij\bar{m}\bar{n}}'^{s}\left(\mathbf{x}\right)H_{mn\alpha}\left(\mathbf{x}\right)a_\alpha^0H_{\bar{m}\bar{n}\beta}\left(\mathbf{x}\right)a_\beta^0\right.$$

$$+c_{ijmn}'^{r}\left(\mathbf{x}\right)c_{ij\bar{m}\bar{n}}^0\left(\mathbf{x}\right)H_{mn\alpha}\left(\mathbf{x}\right)a_\alpha^0H_{\bar{m}\bar{n}\beta}\left(\mathbf{x}\right)a_\beta'^{s}$$

$$+c_{ijmn}^0\left(\mathbf{x}\right)c_{ij\bar{m}\bar{n}}'^{r}\left(\mathbf{x}\right)H_{mn\alpha}\left(\mathbf{x}\right)a_\alpha'^{s}H_{\bar{m}\bar{n}\beta}\left(\mathbf{x}\right)a_\beta^0$$

$$\left.+c_{ijmn}^0\left(\mathbf{x}\right)c_{ij\bar{m}\bar{n}}^0\left(\mathbf{x}\right)H_{mn\alpha}\left(\mathbf{x}\right)a_\alpha'^{r}H_{\bar{m}\bar{n}\beta}\left(\mathbf{x}\right)a_\beta'^{s}\right]S_b^{rs} \tag{83}$$

Starting from the equations posed previously, the computational implementation of the stress-based SFEM based on the Hsieh-Clough-Tocher triangular or the Bogner-Fox-Schmit rectangular finite elements can be done. Further considerations dealing with extension of the presented method to stochastic nonlinear statics or dynamics may be carried out by using the corresponding displacement-based SFEM models.

NUMERICAL ILLUSTRATIONS

Cantilever Beam Solution By Second-Order Airy Functions

The general capabilities of the probabilistic second-order and third-moment approach are displayed through the classical example of a homogeneous steel cantilever beam with the unit thickness loaded by the transverse force *P*. Starting from the classical Airy functions theory, the stress tensor components can be expressed as

$$\sigma_x = -\frac{3}{2}\frac{P}{c^3}xy \qquad \sigma_y = 0 \qquad \tau_{xy} = -\frac{3P}{4c}\left(1-\frac{y^2}{c^2}\right) \tag{84}$$

where 2c is the beam height. The solution of this problem can be obtained alternatively by using Goursat complex functions following analysis presented by Kawai [13]. The first three probabilistic moments of the stress components are determined here by using the perturbation methodology implemented in the mathematical package MAPLE [3] analogously to those obtained for the beams with spatially varying stochastic stiffness [6]. The beam height is treated here as an input random variable with the following parameters: $E[c]$ = 0.10 m, $\sigma(c)$ = 0.01 m, $S(c)$ = 0.10, whereas the other design variables are taken as L = 0.50 m, P = 10 kN, E = 209 GPa, and υ = 0.3. The results of the analysis are presented in Figs. 1–4 (probabilistic moments of the longitudinal stresses: expected values, variances in second-moment approaches, variances in third-moment approaches, and skewness) and in Figs. 5–8 (the additional moments of transverse stresses). It should be emphasized that the MAPLE visualization is very useful because the input random variable induces various random fields at the output.

As is shown in Figs. 1 and 4, the expected values of stresses have shapes relatively similar to and well known from deterministic analyses. Especially valuable is a comparison of the variances resulting from the SOSM and SOTM approaches. In both cases (longitudinal and transverse stresses), the second-order probabilistic characteristics are a little greater for the SOSM computations than in a third-moment analysis. This is a promising result, taking into account a comparison of the perturbation methodology with the MCS results—the coefficients of variation of the random output are greater for the SOSM technique than those resulting from MCS analysis. Therefore, if the SOTM gives results between those obtained by these two methodologies, it is clear that by increasing the perturbation order and probabilistic moment in the analyzed approach, the final results will be closer to the exact result obtained by the simulation.

Finally, as is known, the third-order probabilistic moments can be interpreted as a measure of the nonsymmetry of the output probability density function. Hence, compar-

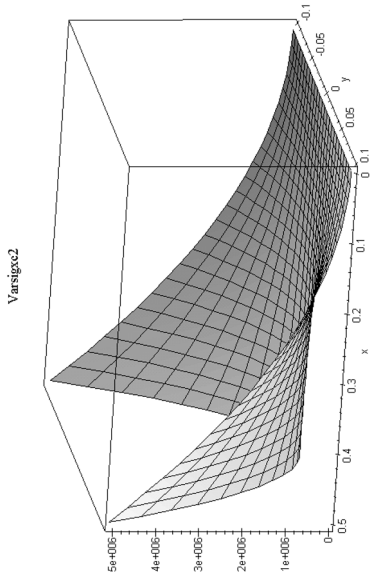

Fig. 2 Variances of σ_x in second-moment method.

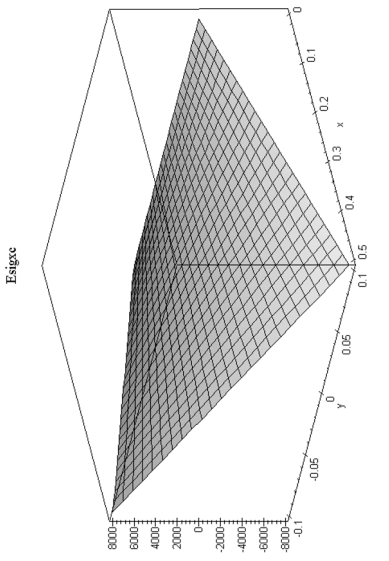

Fig. 1 Expected values of σ_x.

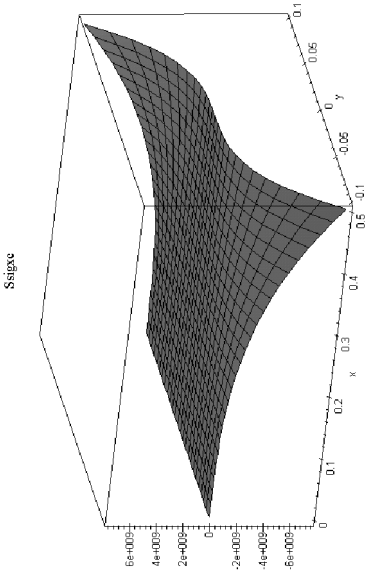

Fig. 4 Skewnesses of σ_x.

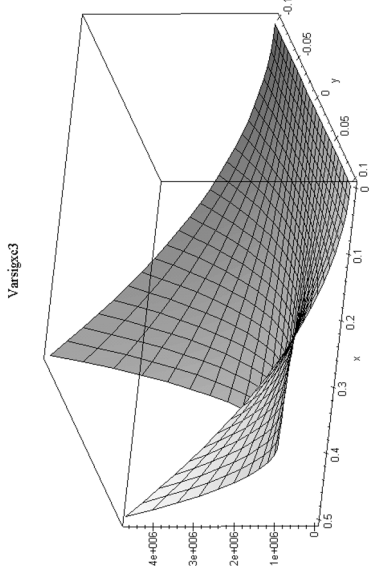

Fig. 3 Variances of σ_x in third-moment method.

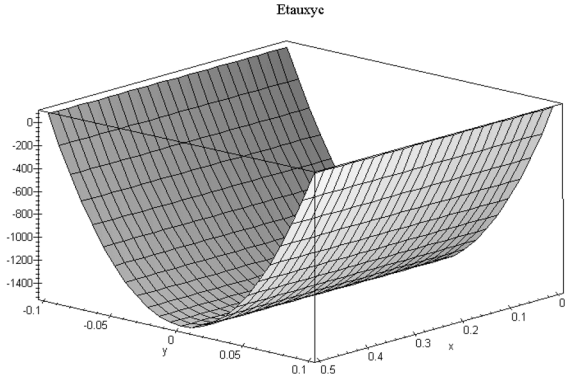

Fig. 5 Expected values of τ_{xy}.

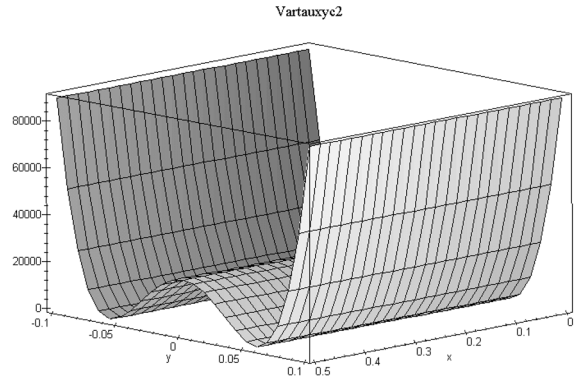

Fig. 6 Variances of τ_{xy} in second-moment method.

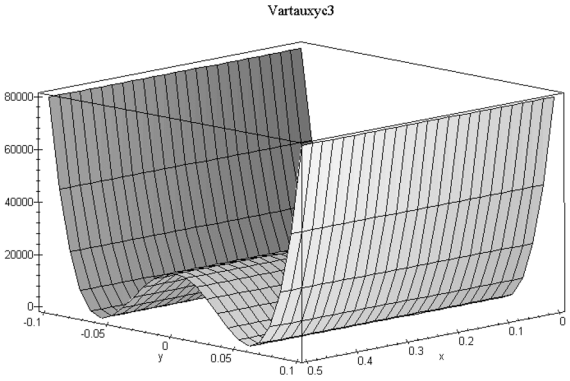

Fig. 7 Variances of τ_{xy} in third-moment method.

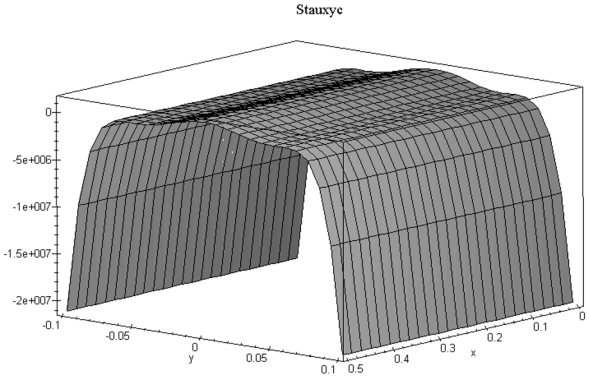

Fig. 8 Skewnesses of τ_{xy}.

ing third-order moments with second-order probabilistic characteristics, it is clear that both surfaces have their extreme values in the same regions; however, they can have different signs (variances maxima may correspond to the skewness maxima and minima at the same time). Because third-order moments characterize the lack of symmetry, they determine the error of assumption made frequently in second-moment analyses, which is that the random process at the output can be of the Gaussian type. Clearly, for the regions of a beam essential to its reliability, the stress components cannot be treated as Gaussian random field types, and, therefore, the following inequality [16]

$$E[\sigma_{kl}] - 3\sigma(\sigma_{kl}) \le \sigma_{kl} \le E[\sigma_{kl}] + 3\sigma(\sigma_{kl}) \tag{85}$$

can no longer be valid; this result is important because of numerous previous analyses published by Liu et al. [16] as well as the general approach to the stochastic reliability analysis.

Deterministic Torsion of a Rectangular Composite Beam

The main idea of the next computational experiment is to determine the convergency of the stress-based finite element method for heterogeneous media. The experiment is carried out by using the example of a quarter of the square periodicity cell with centrally located square reinforcement. The volume ratio of the reinforcing element is 25% of the entire element, and the composite is subjected to torsion in the central point of a cell. The composite parameters are taken as follows: G_1 = 2.0E7, G_2 = 1.0E7; and the representative volume element (RVE) length is taken as a = 4.0. The results of computations are collected for discretization of 2^n (n = 3, 4, 5, 6, 7) elements in Figs. 9–13, and the stresses scale is shown for all graphs in Fig. 14. The boundary conditions for the Prandtl function are taken as equal to 0 on the external boundaries of the RVE and are constant at the interface between composite

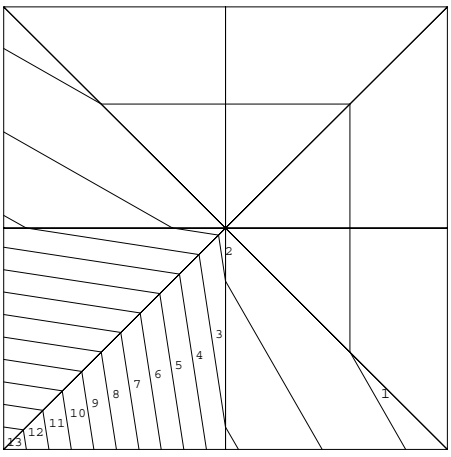

Fig. 9 Stresses in 8-element analysis.

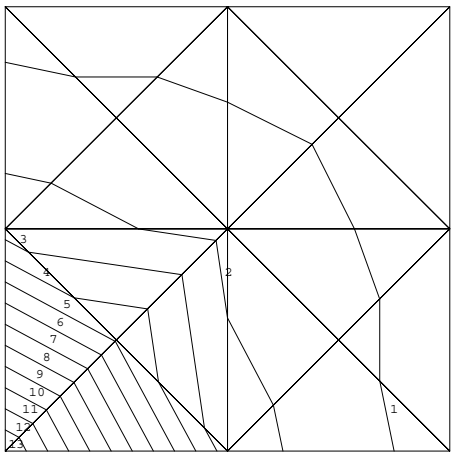

Fig. 10 Stresses in 16-element analysis.

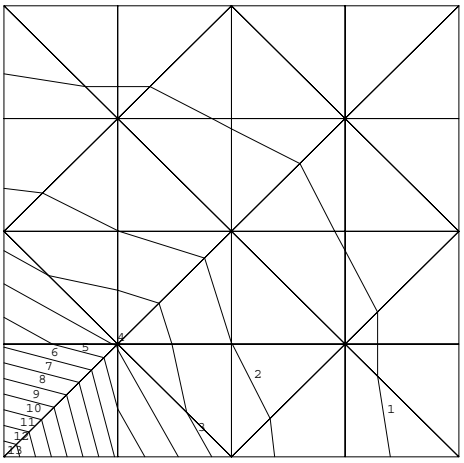

Fig. 11 Stresses in 32-element analysis.

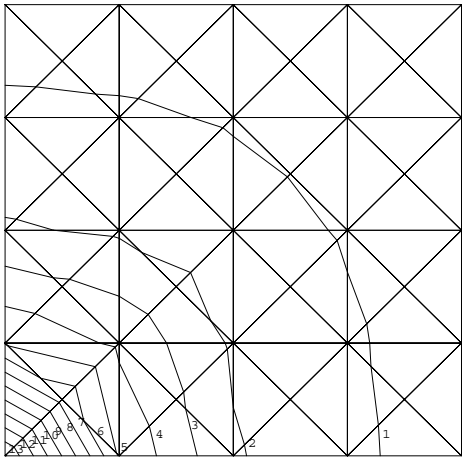

Fig. 12 Stresses in 64-element analysis.

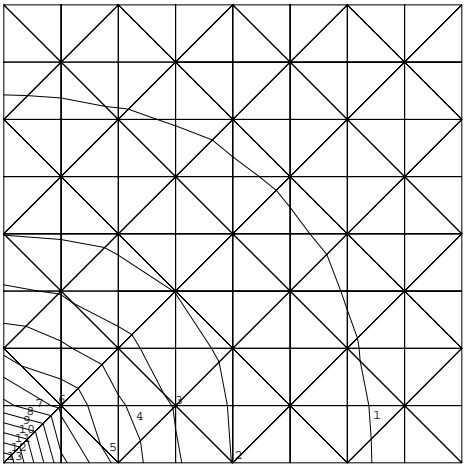

Fig. 13 Stresses in 128-element analysis.

1: 4.00E-02	2: 1.20E-01	3: 2.00E-01	4: 2.80E-01	5: 3.60E-01
6: 4.40E-01	7: 5.20E-01	8: 6.00E-01	9: 6.80E-01	10: 7.60E-01

Fig. 14 Stress scale in computational experiments.

components, which is achieved by using Lagrange multiplier-based linear boundary stress finite elements.

What we can observe in all these experiments is that the stress scale is exactly the same for all cases; the only one general difference is that the smoothness of the high-den-

Table 1. Convergence Test for Stress-Based Finite Element Method in Heterogeneous Media

FE number	Nodes number	$\langle\sigma_x\rangle$	$\langle\sigma_y\rangle$	Strain energy	Twisting moment
$8 = 2^3$	9	–0.17803	0.17803	0.98485E-08	0.11919E+01
$16 = 2^4$	13	–0.16500	0.16500	0.80833E-08	0.10067E+01
$32 = 2^5$	25	–0.14295	0.14295	0.71943E-08	0.85351E+00
$64 = 2^6$	41	–0.12969	0.12969	0.63417E-08	0.76101E+00
$128 = 2^7$	81	–0.11636	0.11636	0.57212E-08	0.67892E+00

sity mesh problem is decisively greater. More significant variations of these solutions are presented in Table 1 for the case of the overall stresses on the RVE area as well as for the total strain energy of the composite.

Probabilistic Patch Test for Rectangular Homogeneous Beam

The stochastic finite element method implementation aspects are explained in detail through the example of torsion of a square bar shown in Fig. 15 and presented in a deterministic context by Desai [4]. The discretization through the use of the constant strain triangle (CST) finite elements of a cross-section quarter is presented in Fig. 15.

The Kirchhoff modulus is defined by its expected value $E[G]$ and variance $Var(G)$; the twisting angle is taken as $\theta = 1$. To illustrate the accuracy of the second-order approach, the variances of φ_1 and φ_5 are compared with corresponding values computed by the use of the MCS technique and corresponding maximum-likelihood statistical estimators [2]. The results for $E[G] = 10.0$ and 10^3, which are the total number of the MCS random samples, are presented in Fig. 16 (denoted by y_1 and y_5) as a function of $Var(G)$ in the range corresponding to the coefficient of variation from the interval (0.0; 0.5).

According to the results of previous computational experiments, the coefficients of variation obtained for the MCS are generally greater than those computed in the SOSM

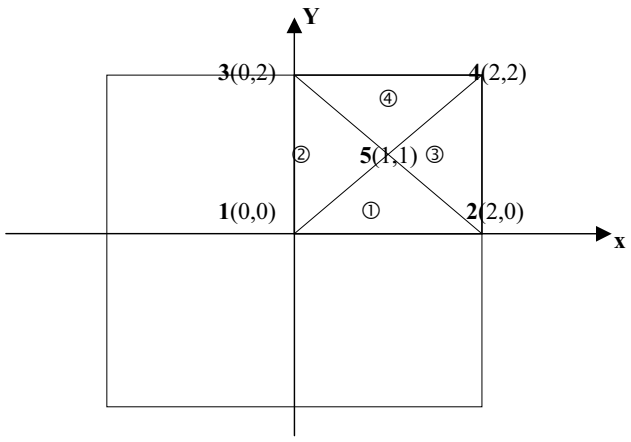

Fig. 15 Cross-section of the bar under torsion.

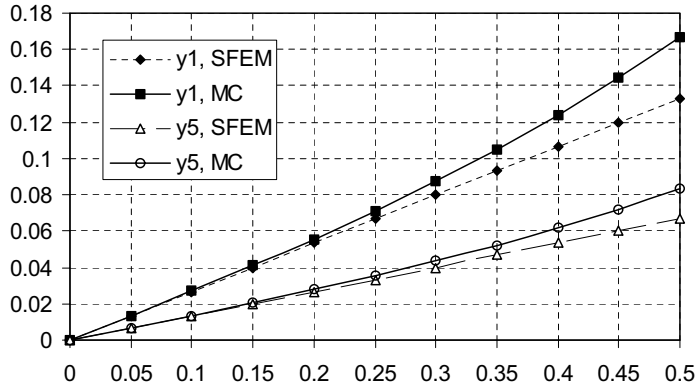

Fig. 16 Comparison of the SOSM and MCS.

perturbation analysis. These differences are negligible for $\alpha(G) \leq 0.2$; however, because these coefficients tend to 0.5, the differences can be as big as 25%. Considering that, the SOSM method may be used in further computations with the restriction on its usage with respect to the coefficient of input random variables. As was mentioned previously, the SOTM technique gives more accurate results; however, the accuracy of stochastic computations depends on the interrelations between probabilistic moments up to the third order. Further demonstration of the SFEM equations solution is presented in Figs. 17–22.

The analyzed probabilistic moments (i.e., expected values—Figs. 17, 20; standard deviations—Figs. 18 and 21; skewness—Figs. 19 and 22) are shown on the vertical axes of these graphs. The standard deviations of Kirchhoff moduli (in percents) are presented on the horizontal axes together with the skewness parameter of the output φ_1 and φ_5. As can be observed, all the moments depend mainly on the second-order probabilistic moments. Third-order probabilistic characteristics do not influence expected values at all. The influence of input skewness is visible in the standard deviations of the output functions but by about one order less than in the case of input standard deviations. The contribution of third-order probabilistic characteristics is decisive only for third-order moments of the analyzed φ_i.

As can be determined from these pictures, the ranges of output probabilistic moments is almost the same as for the input ones. The only exception is obtained for third-order probabilistic characteristics where input positive and negative skewness parameters result in negative output values only, which can have absolute values three times greater than the original parameters.

Weibull Third-Order Reliability Method (W-TORM) in Contact Problem Analysis

Computational experiments are conducted by using the symbolic computations system MAPLE, where the W-SOTM reliability analysis for the contact problem [19] has been implemented. The set of input data is given in Table 2 for the contact problem

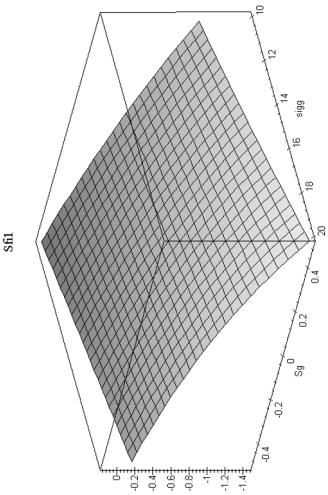

Fig. 19 Skewnesses of φ_1.

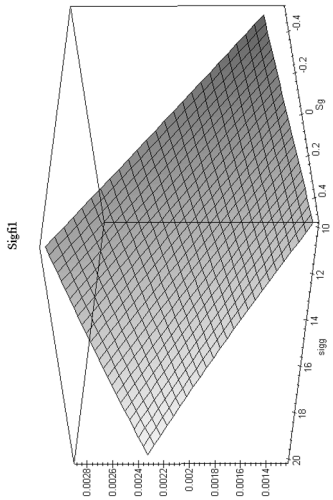

Fig. 18 Standard deviations of φ_1.

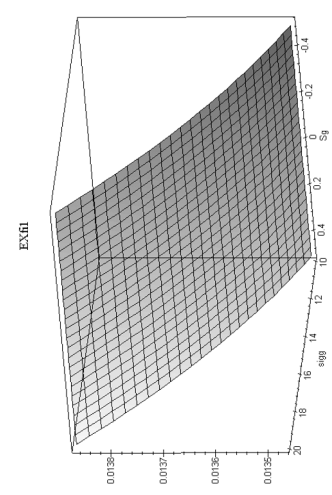

Fig. 17 Expected values of φ_1.

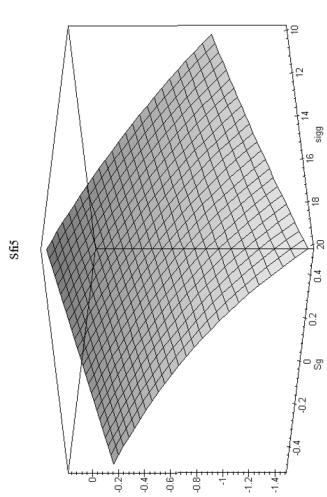

Fig. 22 Skewnesses of φ_5.

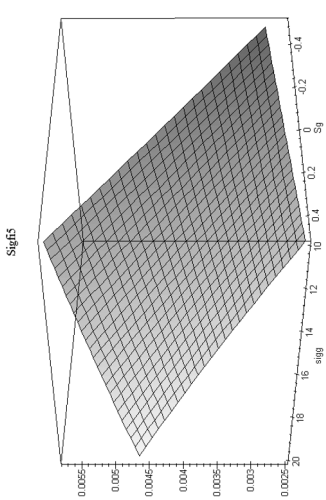

Fig. 21 Standard deviations of φ_5.

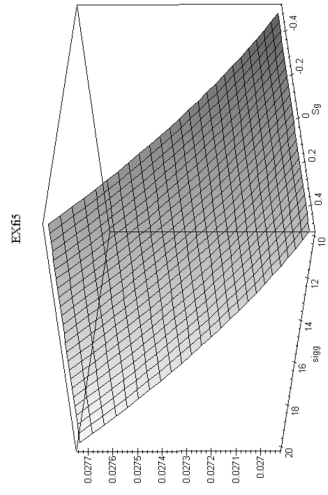

Fig. 20 Expected values of φ_5.

Table 2. Probabilistic Input and Output for Reliability Study

Parameter	Value
E_2	2.0E6
ν_1	0.3
ν_2	0.2
R_2	1.8
P	5.0E2
z	–0.018
$\sigma(E_2)$	0.2E6
$S(E_2)$	0.0
$\sigma(R_2)$	0.018
$S(R_2)$	0.0
σ_{all}	–4.0E5
α	10.0
β	1.01
$E[g]$	–211378.33
$\sigma(g)$	38213.61838 (α = 0.18)
$S(g)$	5.158577

shown in Figs. 23–24, where the following description is adopted: $\alpha = E_1/E_2$ and $\beta = R_1/R_2$. The Weibull PDF of the limit function is determined together with its probabilistic moments up to the third order (compare Table 2) obtained by the symbolic computational solution of the nonlinear integral equation system, Eqs. (41)–(43). The PDF of a limit function is presented in Fig. 25; probabilistic vertical stresses are shown on the horizontal axis, and the probability density function on the vertical axis.

First, it is clear that even for a relatively small input coefficient of variation for input parameters (not greater than 0.1), the randomness level of the output function is about 18% of the relevant expected value; that is why the proposed third-order approach is more

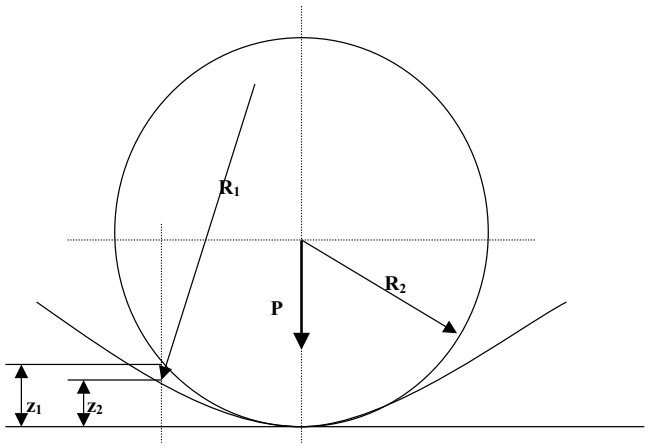

Fig. 23 Contact problem geometry.

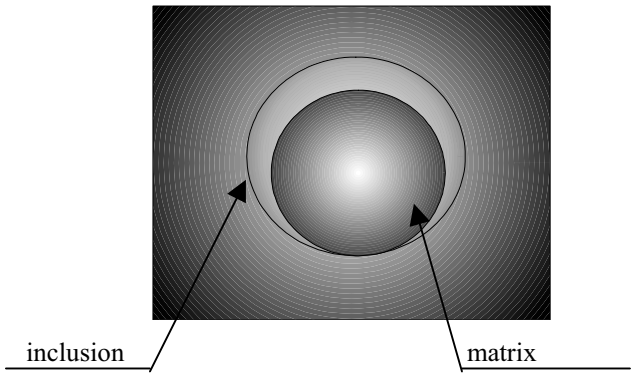

Fig. 24 Three-dimensional view of particle-reinforced composite in contact.

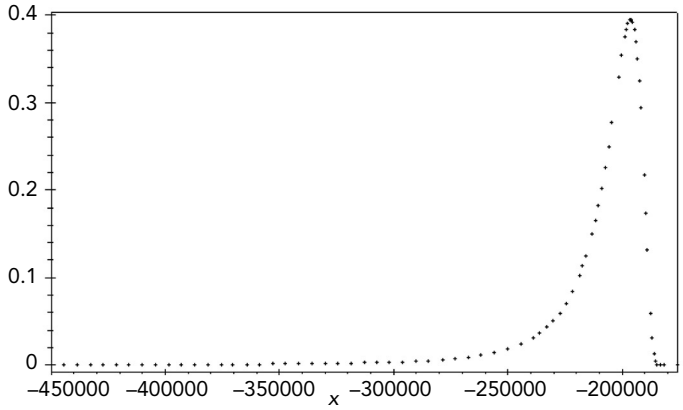

Fig. 25 Weibull limit probability density function.

accurate for the analyzed contact problem. Furthermore, we observe that even for input skewnesses equal to 0, the corresponding third-order probabilistic characteristics differ from 0, which reflects the differences in algebraic combinations of lower order characteristics. In further analysis it is necessary to verify the sensitivity (in both the deterministic and stochastic contexts) of output Weibull PDF probabilistic moments with respect to all input mechanical parameters and their random characteristics; the cross-correlation function of contact stresses $Cov[\sigma(z_1), \sigma(z_2)]$ can be symbolically computed by using the same program or, alternatively, by using the systems MATHCAD or MATLAB.

CONCLUDING REMARKS

The stochastic second-order, third-moment, and stress-based finite element method is a very efficient numerical tool in general analysis of random heterogeneous media. As demonstrated previously in various engineering problems, stochastic third-order stress analysis

is very suitable in third-order reliability method studies worked out very recently. The third-order approach gives more accurate results than the well-known second-order second-moment methodology; thus, it is recommended for further computational implementations of perturbation-based stochastic methods. Furthermore, thanks to the third probabilistic moment analysis, the validity of some inequalities proposed in the second-order approach can be verified and corrected. Finally, we observe that the symbolic computations packages such as MAPLE are very convenient for stochastic perturbation analysis if only a closed-form deterministic solution is theoretically available.

Starting from the SOSM and SOTM equations for the stress-based SFEM, we can build up, using the classical Hsieh-Clough-Tocher triangular or the Bogner-Fox-Schmit rectangular finite elements, the corresponding general computer program. Because the SOSM extension of the deterministic finite element method does not need any intervention within the finite elements subroutines, the existing stress-based computer programs can be used for the second-order stochastic implementation proposed. On the other hand, by analogy to the considerations presented in this article, the second-order perturbation second probabilistic moment approach to torsion or related field problems can be introduced by using the stress method proposed previously or the so-called flux method; both SOSM formulations may be useful in stochastic homogenization of composites [11].

Taking into account the computational experiments presented in this article, it is observed that the more popular displacement-based finite element method can be recommended for computation of the displacement field probabilistic moments. The expected values, cross-covariances, and third probabilistic moments of stresses are more efficiently computed by using the proposed approach, which is very useful in reliability analysis, as demonstrated here. Analogously to the displacement-based finite element method, the maximum value of any input coefficient of variation need not be larger than 0.2; however, significant time savings are obtained with comparison to the MCS, whereas spatial discretization of random fields is possible contrary to the stochastic spectral techniques [9]. Because the SOTM method gives smaller values of the variances than the SOSM-based computer programs, this criterion is not so restrictive for approaches based on higher probabilistic moments.

REFERENCES

1. Bendat, J. S., & Piersol, A. G. (1971). *Random data: analysis and measurement procedures*. New York: Wiley.
2. Borkowski, A. (Ed.) (1977). *Duality and complementarity in mechanics of solids*. Wroctaw: Ossolineum.
3. Char, B. W., et al. (1992). *First leaves: a tutorial introduction to Maple V*. Harrisburg, VA: Springer-Verlag.
4. Desai, C. S. (1979). *Elementary finite element method*. Englewood Cliffs, NJ: Prentice-Hall.
5. Elishakoff, I., Ren, Y. J., & Shinozuka, M. (1995). Improved finite element method for stochastic problems. *Chaos, Solutions & Fractals*, *5*(5), 833–846.
6. Elishakoff, I., Ren, Y. J., & Shinozuka, M. (1995). Some exact solutions for bending of beams with spatially stochastic stiffness. *Int. Journal of Solids and Structures*, *32*, 2315–2327.
7. Feller, W. (1970). *An introduction to probability theory and applications*. New York: Wiley.
8. Fraejis de Veubeke, B., & Hogge, M. (1972). Dual analysis for heat conduction problems by finite elements. *International Journal on Numerical Methods in Engineering*, *5*, 65–82.

9. Ghanem, R. G., & Spanos, P. D. (1991). *Stochastic finite elements: a spectral approach*. Ann Arbor, MI: Springer-Verlag.
10. Grigoriu, M. (2000). Stochastic mechanics. *International Journal of Solids and Structures*, *37*, 197–214.
11. Kamiński, M. (2000). Homogenized properties of *n*-components composite materials. *International Journal of Engineering Sciences*, *38*(4), 405–427.
12. Kamiński, M., & Hien, T. D. (1999). Stochastic finite element modeling of transient heat transfer in layered composites. *International Communications for Heat and Mass Transfer*, *26*(6), 791–800.
13. Kawai, T. (2000). The force method revisited. *International Journal of Numerical Methods in Engineering*, *47*, 275–286.
14. der Kiureghian, A., & Jyh, B. K. (1988). The stochastic finite element method in structural reliability. *Probabilistic Engineering Mechanics*, *3*(2), 83–91.
15. Kleiber, M., & Hien, T. D. (1992). *The stochastic finite element method*. Chicheter, UK: Wiley.
16. Liu, W. K., Belytschko, T., & Mani, A. (1986). Probabilistic finite elements for nonlinear structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, *56*, 61–81.
17. Nayfeh, A. H. (1973). *Perturbation methods*. New York: Wiley.
18. Thoft-Christensen, P., & Baker, M. J. (1982). Structural reliability theory and its applications. New York: Springer-Verlag.
19. Timoshenko, S., & Goodier, J. N. (1951). Theory of elasticity. New York: McGraw-Hill.
20. Vanmarcke, E. H. (1983). *Random fields. Analysis and synthesis*. Cambridge, MA: MIT Press.
21. Vanmarcke, E. H., & Grigoriu, M. (1983). Stochastic finite element analysis of simple beams. *Journal of Engineering Mechanics ASCE*, *109*(5), 1203–1214.
22. Więckowski, Z. (1999). Stress-based finite element analysis of plane plasticity problems. *International Journal on Numerical Methods in Engineering*, *44*, 1505–1525.
23. Zienkiewicz, O. C., & Taylor, R. C. (1991). *The finite element method*. McGraw-Hill.

Hybrid Methods in Engineering, vol. 3, pp. 53–65, 2001

A NEW NUMERICAL TECHNIQUE FOR TRANSPORTATION OF AIRBORNE PARTICLES

S. D. Wright
The School of the Environment, University of Leeds, Leeds, LS2 9JT, England

L. Elliott & D. B. Ingham
Department of Applied Mathematics, University of Leeds, Leeds, LS2 9JT, England

It is great of importance to be able to predict the trajectories of airborne particles that are released from exhaust systems into the atmospheric boundary layer. These particles may disperse under favorable weather conditions, but under certain weather conditions, or as a result of the local topography, potentially hazardous levels of contamination may occur. In this article, the differential equations that govern the transport of these particles within the atmospheric boundary layer are considered and a numerical algorithm is devised to solve them efficiently. It will be shown that, unlike the transportation of particles over small scales, the efficient integration of the equations of motion is nontrivial without making certain simplifying assumptions.

INTRODUCTION

Many authors have investigated the motion of particles within the atmosphere. Particle trajectory models include those of Thompson [14] and Joynt and Blackman [2], who developed random walk models based on particles with uncorrelated velocities at successive time steps (namely, a Weiner process). The Langevin equation has also been well used in calculating particle trajectories that allow correlations between the particle velocities at successive time steps. Numerous authors have adopted this approach, including, for example, Ley [3], Van Dop [17], and Monti and Leuzzi [4]. Thompson [11, 12, 13] showed how the statistical properties of the atmosphere could be incorporated in a rigorous mathematical manner, and Sawford [6], Rodean [5], and Wilson and Sawford [19] have reviewed this work.

Underwood [16] has proposed an alternative approach. He represented *elements of pollution* as particles and envisaged that they experience *collisions* with the turbulent eddies governed by the Lagrangian time scale. At each collision, all memory of the prevailing turbulent velocity is lost, whereas between collisions, perfect correlation is retained. Underwood [15] showed that this method gave good results compared with the results of Taylor [10] for the root mean square displacement of particles within homogeneous turbulence and for modeling dispersion within a neutral atmosphere.

In the present work, the pollutant of interest is of a *particulate* nature; hence, the use of a particle-tracking model is not aimed at representing a gaseous plume.

GOVERNING EQUATIONS

The drag force acting on a body, which is moving relative to the fluid, is the net result of the distribution and magnitude of the local pressure field and shearing forces over the sur-

face of the body. For very slow flow, Stokes [9] calculated the drag force on a spherical particle to be

$$\underline{F}_D = 3\mu\pi d\left(\underline{u} - \underline{u}_p\right) \tag{1}$$

where d is the diameter of the particle, μ is the viscosity of the fluid, $\underline{u}$ the velocity of the fluid, and $\underline{u}_p$ the velocity of the particle. Equation (1) is valid only when the particle Reynolds number is much less than unity, that is

$$\mathrm{Re}_p = \frac{\rho d\left|\underline{u} - \underline{u}_p\right|}{\mu} \ll 1 \tag{2}$$

where ρ is the density of the fluid. The domain where Eq. (2) is valid is widely known as the Stokes regime. In general, Eq. (1) is written in terms of a *drag coefficient*, C_D, which is defined as

$$C_D = \frac{\left|\underline{F}_D\right| / A}{\tfrac{1}{2}\rho\left|\underline{u} - \underline{u}_p\right|^2} \tag{3}$$

where A is the projected area of the particle in its direction of travel. Thus, for the Stokes regime, Eq. (1) can be written in the form

$$\underline{F}_D = \frac{C_D\rho\pi d^2\left|\underline{u} - \underline{u}_p\right|\left(\underline{u} - \underline{u}_p\right)}{8} \tag{4}$$

and

$$C_D = \frac{24}{\mathrm{Re}_p} \tag{5}$$

Many authors have modified the drag coefficient, either through theoretical calculations or experimental investigations, so that it is valid over a reasonably large range of particle Reynolds numbers. Hence, the drag coefficient that will be used in this study was that given and also employed by Shuen et al. [8], namely

$$C_D = \frac{24}{\mathrm{Re}_p}\left(1 + \frac{\mathrm{Re}_p^{2/3}}{6}\right) \tag{6}$$

Given the modified drag coefficient, Eq. (4) is valid for all values of the particle Reynolds number; for example, see Vincent [18]. At small values of Re_p the drag coefficient, C_D, is

inversely proportional to Re_p, whereas at large values of Re_p the presence of a separated turbulent wake causes a drag crisis in C_D. In vector notation, the governing equation of motion of an airborne particle of mass m is given by

$$m\frac{d\underline{u}_p}{dt} = \underline{F}_D + \underline{C} + m\underline{g} \tag{7}$$

where $\underline{F}_D$ is the drag force on the particle, as given in Eq. (4), $\underline{C}$ is the Coriolis force, and $m\underline{g}$ is the gravitational force. Thus, assuming that the particle is spherical, the equations of motion that govern the transport of particles are given by

$$\frac{du_{1p}}{dt} = \frac{3C_D\rho}{4\rho_p d}\left|\underline{u}-\underline{u}_p\right|\left(u_1-u_{1p}\right)+fu_{2p}-fu_{3p}\cos(\alpha)\cot(\varphi) \tag{8a}$$

$$\frac{du_{2p}}{dt} = \frac{3C_D\rho}{4\rho_p d}\left|\underline{u}-\underline{u}_p\right|\left(u_2-u_{2p}\right)-fu_{1p}+fu_{3p}\sin(\alpha)\cot(\varphi) \tag{8b}$$

$$\frac{du_{3p}}{dt} = \frac{3C_D\rho}{4\rho_p d}\left|\underline{u}-\underline{u}_p\right|\left(u_3-u_{3p}\right)+f\cot(\varphi)\left(u_{1p}\cos(\alpha)-u_{2p}\sin(\alpha)\right)-g \tag{8c}$$

where ρ_p is the particle density, u_{1p}, u_{2p}, and u_{3p} are the velocity components in the x_1, x_2, and x_3 directions, respectively, α is the angle the x_1 axis makes with the east–west direction, φ the latitude, and f the Coriolis parameter.

Further, on nondimensionalizing Eqs. (8a–8c), using the scalings

$$\begin{aligned} x_1 &= L\tilde{x}_1 \quad x_2 = L\tilde{x}_2 \quad x_3 = H\tilde{x}_3 \quad \rho = M\tilde{\rho}/L^3 \\ u_1 &= G\tilde{u} \quad u_2 = G\tilde{u}_2 \quad u_3 = W\tilde{u}_3 \quad t = L/G\,\tilde{t}\sqrt{2} \end{aligned} \tag{9}$$

where G is a horizontal velocity scaling, L and H are suitable horizontal and vertical length scales, and $W = GH/L$ is a vertical velocity scale, Eqs. (8a)–(8c) become

$$\frac{d^2\tilde{x}_{1p}}{d\tilde{t}^2} = \frac{1}{\mathrm{St}}\left(1+\frac{\mathrm{Re}_p^{2/3}}{6}\right)\left(\tilde{u}_1-\tilde{u}_{1p}\right)+\frac{\tilde{u}_{2p}-Q\tilde{u}_{3p}\cot(\varphi)\cos(\alpha)}{\mathrm{Ro}} \tag{10a}$$

$$\frac{d^2\tilde{x}_{2p}}{d\tilde{t}^2} = \frac{1}{\mathrm{St}}\left(1+\frac{\mathrm{Re}_p^{2/3}}{6}\right)\left(\tilde{u}_2-\tilde{u}_{2p}\right)-\frac{\tilde{u}_{1p}-Q\tilde{u}_{3p}\cot(\varphi)\sin(\alpha)}{\mathrm{Ro}} \tag{10b}$$

$$\frac{d^2\tilde{x}_{3p}}{d\tilde{t}^2} = \frac{1}{\mathrm{St}}\left(1+\frac{\mathrm{Re}_p^{2/3}}{6}\right)\left(\tilde{u}_3-\tilde{u}_{3p}\right)+\frac{\cot(\varphi)}{Q\mathrm{Ro}}\left(\tilde{u}_{1p}\cos(\alpha)-\tilde{u}_{2p}\sin(\alpha)\right)-\frac{1}{Q^2\mathrm{Fr}} \tag{10c}$$

where St = $\rho_p d^2 G/18\mu L$ is the Stokes number, Fr = G^2/gH is the Froude number, Ro = G/fL is the Rossby number, $Q = H/L$, and $(\tilde{x}_{1p}, \tilde{x}_{2p}, \tilde{x}_{3p})$ is the nondimensional position vector of the particle. The Stokes number represents the ability of a particle to follow the streamlines, and the extent by which the airflow and particle trajectory patterns differ is largely due to the magnitude of the Stokes number. In general, the larger the Stokes number of an airborne particle, the more the particle will diverge from the streamlines of the fluid. Conversely, for a very small Stokes number, a particle is likely to follow the streamlines of the fluid very closely.

THE SOLUTION OF THE GOVERNING DIFFERENTIAL EQUATIONS

If Eqs. (10a)–(10c) are to be solved, then numerical techniques must be used to integrate the governing equations of motion with respect to time. One standard technique that could be used is an explicit method, such as the fourth-order Runge-Kutta method. For a particle traveling in a fluid at rest, with the Coriolis force neglected, then for such a numerical scheme to be stable for Eqs. (10a)–(10c), the time step Δt needs to satisfy the inequality $\Delta t < 2.785 \cdot \mathrm{St}$ For small-scale flows, for example $L = 1$ m, with a particle of diameter $d = 100$ μm, density of $\rho_p = 1000$ kg/m, velocity scale $G = 20$ m/s and the viscosity for air given as $\mu = 1.3 \cdot 10^{-5}$ kg/m·s, then the previously mentioned constraint becomes $\Delta t < 8.4 \cdot 10^{-2}$ This is an acceptable time-step for small-scale flows because the usual duration of the simulation is quite small; hence, a reasonable number of time-steps is required. However, within the atmosphere, the length scale L of a typical size of a hill or town is much larger, say $L = 8000$ m—see Wright et al. [20]—and this length scale has to be used in both the calculation of the fluid flow and in the particle trajectory simulation. Thus, for the explicit Runge-Kutta scheme to be stable, $\Delta t < 1.05 \cdot 10^{-5}$. The maximum time step also decreases as the square of the particle diameter; thus, the smaller the particle diameter, the smaller the maximum time step that can be taken.

The explicit Runge-Kutta scheme described previously is of little use for solving the governing differential Eqs. (10a)–(10c) in their present form because the length scale L used in calculating the fluid flow needs to be introduced in the evaluation of the Stokes number. This makes the equations stiff and is the reason why such a small time step is required in the stability criterion. Physically, a small time scale is associated with the particle because of its small relaxation time.

Butcher (1964) considered the accuracy of an n stage implicit scheme as compared with an n stage explicit scheme. The general two-stage implicit scheme he considered was as follows

$$\underline{y}_{n+1} = \underline{y}_n + \Delta t\left\{b_1 \underline{k}_1 + b_2 \underline{k}_2\right\} \tag{11a}$$

$$\underline{k}_1 = \underline{g}(t_n + \alpha_1 \Delta t,\ \underline{y}_n + \beta_{11}\underline{k}_1 + \beta_{12}\underline{k}_2) \tag{11b}$$

$$\underline{k}_2 = \underline{g}(t_n + \alpha_2 \Delta t,\ \underline{y}_n + \beta_{21}\underline{k}_1 + \beta_{22}\underline{k}_2) \tag{11c}$$

where $\underline{y}_n$ is an approximation to the solution of the differential equation (10). By expanding Eq. (11) in a Taylor series and comparing with the Taylor series expansion of $\underline{y}_{n+1}$ in

powers of $\underline{y}_n$, it is easy to show that all the terms up to and including the terms in Δt^4 can be made to match by choosing

$$\begin{aligned} b_1 &= b_2 = \frac{1}{2} & \beta_{22} &= \beta_{11} = \frac{1}{4} \\ \alpha_1 &= \frac{1}{2} - \frac{\sqrt{3}}{6} & \beta_{12} &= \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \alpha_2 &= \frac{1}{2} + \frac{\sqrt{3}}{6} & \beta_{21} &= \frac{1}{4} + \frac{\sqrt{3}}{6} \end{aligned} \tag{12}$$

thereby producing the numerical scheme of Eq. (11) that is fourth-order accurate. This numerical scheme is A-stable. Defining n to be the number of time steps, Dahlquist (1963) then defined an A-stable scheme to be one in which "all the solutions tend to zero, as $n \to \infty$, when the method is applied with a fixed positive $\Delta\tilde{t}$ to any differential equation of the form $dy/dt = \lambda y$ with $\Re e(\lambda) < 0$." The implicit Runge-Kutta method given by Eqs. (11a)–(11c) is stable, although not necessarily accurate for arbitrary large values of $\Delta\tilde{t}$, when applied to Eqs. (10a)–(10c). However, because the scheme is fourth-order accurate, time steps much larger than that given by the explicit scheme can be used without a significant loss of accuracy.

Although the implicit Runge-Kutta method in Eq. (11) is A-stable, this stability arises from the need to solve a system of nonlinear algebraic equations at each time step because Eqs. (11b) and (11c), in general, are nonlinear in $\underline{k}_1$ and $\underline{k}_2$, respectively. For the system of O.D.Es Eqs. (10a)–(10c), 12 nonlinear algebraic equations in 12 unknowns must be solved at each time step; that is $\underline{k}_1$ and $\underline{k}_2$ each have 6 unknown components. Unless this can be achieved efficiently, then the extra work required to accomplish this task outweighs the benefits gained from the method being A-stable, and the explicit Runge-Kutta scheme may just as well be employed with a small $\Delta\tilde{t}$.

One method for solving a system of equations of the form

$$F(x) = 0 \tag{13}$$

is the Newton method. This method has the advantage that it gives quadratic convergence of Eq. (13) from the initial guess $\underline{x}_o$ to the solution $\underline{x}_s$. The solution to this system, if one exists, can be found iteratively from an initial guess $\underline{x}_o$ by first solving the linear system of equations for $\underline{x}$

$$J(\underline{x}_o)\underline{x} = -F(\underline{x}_o) \tag{14}$$

where $J(\underline{x})$ is the Jacobian of $\underline{F}(\underline{x})$ and further setting $\underline{x}_1 = \underline{x}_o + \underline{x}$ and then iterating until convergence; see, for example, Burden and Faires (1989). However, a significant weakness of this method is that for a system of n equations, then at every iteration $(n^2 + n)$ evaluations of the vector $\underline{F}(\underline{x})$ are required. Further, the evaluation of the Jacobian matrix, if $\underline{F}$ is complicated, and the subsequent solution of the system of linear equations, if n is large, for each Newton iteration is computationally expensive. Solving a system of linear equations involves $O(n^3)$ operations; thus, for just one iteration $O(n^3)$ arithmetic operations are

required. Hence, a more efficient method of solving Eq. (14) is sought by introducing an approximation matrix to the Jacobian $J(\underline{x}_1)$, given that $\underline{x}_1$ has been obtained by the Newton method. An approximation to the Jacobian, $J(\underline{x}_1)$ is given by

$$A(\underline{x}_1 - \underline{x}_o) = \underline{F}(\underline{x}_1) - \underline{F}(\underline{x}_o) \tag{15}$$

where A is an $n \times n$ matrix. In general, Eq. (15) does not uniquely define the matrix, A, because no information is available on how the matrix A operates on vectors that are orthogonal to $\underline{x}_1 - \underline{x}_o$. Hence, it is also required that

$$A\underline{z} = J(\underline{x}_o)\underline{z} \quad \text{whenever} \quad (\underline{x}_1 - \underline{x}_o)^t \underline{z} = 0 \tag{16}$$

This specifies that all vectors that are orthogonal to $(\underline{x}_1 - \underline{x}_o)$ are unaffected by the update of the Jacobian matrix, J, to the approximate matrix A. Dennis and Moré (1977) showed that Eqs. (15) and (16) uniquely define the matrix A to be

$$A_1 = J(\underline{x}_o) + \frac{\left[\underline{F}(\underline{x}_1) - \underline{F}(\underline{x}_o) - J(\underline{x}_o)(\underline{x}_1 - \underline{x}_o)\right](\underline{x}_1 - \underline{x}_o)^t}{\|\underline{x}_1 - \underline{x}_o\|^2} \tag{17}$$

and this matrix can then be used in the Newton method in place of $J(\underline{x}_1)$ to determine $\underline{x}_2$. Hence, in general, an iterative scheme for obtaining $\underline{x}_{k+1}$ is given by:

$$\underline{x}_{k+1} = \underline{x}_k - A_k^{-1}(\underline{x}_k)\underline{F}(\underline{x}_k) \tag{18}$$

$$A_i = A_{i-1} + \frac{\underline{p}_i - A_{i-1}\underline{s}_i}{\|\underline{s}_i\|_2^2}\underline{s}_i^t \tag{19}$$

where $s_i = \underline{x}_i + \underline{x}_{i-1}$ and $\underline{p}_i = \underline{F}(\underline{x}_i) - \underline{F}(\underline{x}_{i-1})$. This reduces the number of scalar function evaluations of $\underline{F}$ from $(n^2 + n)$ to n per iteration. However, this method still requires the inversion of the matrix A at every iteration; thus, it is still far too computationally expensive. A significant improvement in the calculation of the inverse of the matrix A_i can be achieved by considering the matrix inversion formula of Sherman and Morrison [7], namely

$$(A + \underline{x}\underline{y}^t)^{-1} \equiv A^{-1} - \frac{A^{-1}\underline{x}\underline{y}^t A^{-1}}{1 + \underline{y}^t A^{-1}\underline{x}} \tag{20}$$

Defining the quantities A, $\underline{x}$, and $\underline{y}$ that are present in Eq. (20) as follows:

$$A = A_{i-1} \tag{21}$$

$$\underline{x} = \frac{\underline{p}_i - A_{i-1}\underline{s}_i}{\|\underline{s}_i\|^2} \tag{22}$$

$$\underline{y} = \underline{s}_i \tag{23}$$

and using Eq. (19), along with identity Eq. (20), yields, after some algebraic manipulation

$$A_i^{-1} = A_{i-1}^{-1} + \frac{(\underline{s}_i - A_{i-1}^{-1}\underline{p}_i)\underline{s}_i^t A_{i-1}^{-1}}{\underline{s}_i^t A_{i-1}^{-1}\underline{p}_i} \tag{24}$$

which eliminates the need to solve the linear system of equations at each iteration. Instead, the iterative method in Eq. (19) can be used, with the matrix A_i^{-1} at each iteration being updated from the previous iteration by using the formula of Eq. (24). Hence, the number of scalar function evaluations has been reduced from $(n^2 + n)$ to n, and the number of arithmetic evaluations required to calculate the inverse Jacobian matrix (in the form of solving a linear system) has been reduced from $O(n^3)$ to $O(n^2)$ per iteration. This is because Eq. (24) involves matrix multiplication only and is computationally more efficient than solving a system of linear equations to invert the matrix A_i. As a consequence, the quadratic convergence of the Newton method has been reduced to superlinear convergence, which is an acceptable trade-off in most applications. This approach uses the stability of the implicit method in Eqs. (11a)–(11c) to overcome the inherent stiffness of the governing equations (10a)–(10c), which arises because of the different characteristic time scales that govern the motion of the fluid and the particle. For the fluid, the characteristic time scale is given by

$$\tau_F = \frac{L}{G} \tag{25}$$

whereas the characteristic time scale of the particle is the relaxation time, namely

$$\tau_P = \frac{d^2 \rho_p}{18\mu} \frac{24}{C_D \operatorname{Re}_p} \tag{26}$$

See Vincent [18]. The ratio of these two time scales, in the limit of the particle Reynolds number tending to zero, yields:

$$\lim_{\operatorname{Re}_p \to 0} \left(\frac{\tau_p}{\tau_F} \right) \to \mathrm{St} \tag{27}$$

and hence this ratio determines how closely the particle follows the streamline. For small values of τ_p/τ_F, the particle will follow the streamlines closely because the Stokes number

will be small, whereas for large values of τ_p/τ_F the particle is relatively slow to respond to flow distortions and the particle tends not to follow the fluid movement. For reasonably small particles, the particle relaxation time is at least two orders of magnitude smaller than τ_F. Therefore, the Stokes number is small and the governing equations of motion are stiff.

PARTICLES IN LOCAL EQUILIBRIUM IN THE ATMOSPHERE

To overcome the stiffness in the governing equations of motion (10a)–(10c), an implicit scheme has been derived. This scheme allows the use of a time step of arbitrary size and hence larger than the particle relaxation time.

Another approach that can overcome the inherent stiffness in the governing Eqs. (10a)–(10c) is to assume that the particle is in constant local equilibrium with its surroundings. This approximation neglects the acceleration of the particle, because the acceleration occurs over a time scale equivalent to the relaxation time, which can be quite small for small particles. This approximation is equivalent to the limit of particle relaxation time and hence corresponds to the Stokes number tending to zero.

In general, the smaller the particle, the smaller both the Stokes number and the particle relaxation time and thus the more accurate the approximation. Using this approximation, the governing equations of motion (10a)–(10c) reduce to, on also neglecting the Coriolis force

$$\frac{d\tilde{x}_{1p}}{d\tilde{t}} = \tilde{u}_1 \tag{28a}$$

$$\frac{d\tilde{x}_{2p}}{d\tilde{t}} = \tilde{u}_2 \tag{28b}$$

$$\frac{d\tilde{x}_{3p}}{d\tilde{t}} = \tilde{u}_3 - \frac{1}{Q^2 \mathrm{Fr}} \frac{6\mathrm{St}}{(6 + \mathrm{Re}_p^{2/3})} \tag{28c}$$

and hence the term *local equilibrium* is used to indicate that the particle velocity is equal to the fluid velocity at all positions in space and time, except for an extra sedimentation velocity imposed on the particle in the x_3 direction as a result of the effects of gravity. For the solution of Eqs. (28a)–(28c), only one initial condition is required: the particle's starting position. No information about the particle's initial velocity can be specified because it is assumed to have the velocity of the fluid at the position from which it is released into the atmospheric boundary layer. Before solving Eqs. (28a)–(28c), a method is required to calculate an approximation to the particle Reynolds number. Using Eq. (2) and assuming that the particle is in local equilibrium results in

$$\mathrm{Re}_p = \frac{\rho d w_s}{\mu} \tag{29}$$

where w_s is the sedimentation velocity of the particle and is given by

$$w_s = \tau_p g = \frac{d^2 \rho_p}{18\mu} \frac{6g}{6 + \mathrm{Re}_p^{2/3}} \tag{30}$$

Hence, on combining Eqs. (29) and (30), a nonlinear equation for the particle Reynolds number is given by

$$\mathrm{Re}_p = \frac{1}{3\mu^2} \frac{g\rho\rho_p d^3}{\left(6 + \mathrm{Re}_p^{2/3}\right)} \tag{31}$$

and this can be solved by an iterative method. To solve the equations of motion (28a)–(28c), the stiffness of the original system of Eqs. (10a)–(10c) has been removed. Therefore, an explicit Runge-Kutta scheme can be implemented.

The physical interpretation of the stability of the implicit Runge-Kutta scheme in Eqs. (11a)–(11c) for time steps larger than the relaxation time can now be given; that is, for large values of $\Delta\tilde{t}$, the effects of the sharp acceleration of the particle over time scales of the order of the relaxation time become *smoothed* by the scheme. This gives stability; thus, as $\Delta\tilde{t} \to \infty$, Eqs. (10a)–(10c), neglecting the Coriolis force, reduce to Eqs. (28a)–(28c). This is because in the limit of $\Delta\tilde{t} \to \infty$ the implicit Runge-Kutta scheme of Eqs. (11a)–(11c) effectively implements the conditions of local equilibrium on the full governing Eqs. (10a)–(10c). Hence, in the limit $\Delta\tilde{t} \to \infty$, the explicit Runge-Kutta scheme, along with the equations of motion (28a)–(28c), is equivalent to the implicit Runge-Kutta scheme of Eqs. (11a)–(11c), along with the equations of motion (10a)–(10c), with the Coriolis force neglected.

COMPARISON OF THE FULL EQUATIONS AND THOSE ASSUMING LOCAL EQUILIBRIUM

To analyze the theory presented previously, the motion of particles released into an Ekman boundary layer over flat topography are considered. The effects of increasing the integration time step, in addition to approximating the full equations by assuming that the particles are in local equilibrium with the fluid flow, are investigated.

For a constant kinematic eddy viscosity ν, where $\nu = \nu_o + n_t$, with molecular viscosity, ν_o, and eddy viscosity, ν_t, Ekman [1] was able to present a solution to the Navier-Stokes equations, subject to the no-slip boundary condition at the ground and the geostrophic boundary condition at a large distance from the ground, namely:

$$u_1 = u_{1g} - e^{-ax_3}\left[u_{1g}\cos(ax_3) + u_{2g}\sin(ax_3)\right] \tag{32a}$$

$$u_2 = u_{2g} + e^{-ax_3}\left[u_{1g}\sin(ax_3) - u_{2g}\cos(ax_3)\right] \tag{32b}$$

where $a = \sqrt{1/2\mathrm{Rov}}$ and (u_1, u_2) and (u_{1g}, u_{2g}) are the horizontal fluid and geostrophic wind velocities, respectively. Hence, Eqs. (32a) and (32b) are taken for the fluid flow components in the numerical solution of the particle trajectory Eqs. (10a)–(10c).

Table 1 shows, for a range of particle sizes, the value of the particle Reynolds number, Re_p, assuming local equilibrium; the Stokes number, St; the nondimensional parameter, a; and the Froude number, Fr. For all the calculations that follow, for a given particle diameter, the parameters shown in Table 1 are employed. In addition, because there are no natural length scales for the Ekman boundary layer over flat topography, these have been taken to be unity.

Table 2 shows the particle landing position (x_1, x_2) obtained from solving the full governing Eqs. (10a)–(10c) for successively smaller time steps of a 50 μm particle released from a height $x_3 = 100$ m into the mean Ekman boundary layer over flat topography. As the time step decreases, the landing position of the particle converges to a constant value; however, Table 2 also indicates that a relatively large time step can be taken without a significant loss in accuracy, thereby justifying the use of an implicit scheme.

Table 1. Value of Particle Reynolds Number, Re_p; Stokes number, St; Parameter, a; and Froude number, Fr, for a Range of Particle Sizes

Particle diameter (μm)	Particle Reynolds number, Re_p	Stokes number, St	$a = \sqrt{\frac{1}{2\mathrm{Rov}}}$	Froude number, Fr
1000	334.7	$3.63 \cdot 10^{0}$	0.008	0.102
750	193.8	$2.04 \cdot 10^{0}$	0.008	0.102
500	87.96	$9.08 \cdot 10^{-1}$	0.008	0.102
250	20.87	$2.27 \cdot 10^{-1}$	0.008	0.102
100	2.34	$3.63 \cdot 10^{-2}$	0.008	0.102
75	1.08	$2.04 \cdot 10^{-2}$	0.008	0.102
50	0.35	$9.08 \cdot 10^{-3}$	0.008	0.102

Table 2. Particle Landing Position for Particle Released at Height of $X_3 = 100$ m into Mean Ekman Boundary Layer over Flat Topography with Parameters Given in Table 1 for Successively Smaller Time Steps

Time step (s)	Particle landing position (x_1, x_2) (m)	
500.0	8919.6633	5470.2812
200.0	8943.9460	5495.2361
50.00	8989.8770	5546.3087
30.00	8993.0531	5549.4626
10.00	9005.6433	5555.7229
5.000	9005.7698	5555.8492
1.000	9005.7908	5555.8702
0.500	9005.7913	5555.8708
0.100	9005.795	5555.8709
0.050	9005.7915	5555.8709
0.010	9005.7915	5555.8709

To investigate the effects of assuming that the particles are in local equilibrium with the fluid flow, particles of various sizes have been released from a height of x_3 = 100 m into the mean Ekman boundary layer over flat topography and the landing positions of the particles calculated by using both the full and approximated equations of motion, that is, Eqs. (10a)–(10c) and (28a)–(28c), respectively.

Table 3 shows the landing positions of particles of various sizes with the parameter a = 0.008, calculated by using the fully implicit scheme and a small time step. Hence, the difference in the solutions, calculated by using the full Eqs. (10a)–(10c) compared with the approximate Eqs. (28a)–(28c), is solely due to the approximations made in obtaining Eqs. (28a)–(28c). Table 3 shows that there is a very small relative error between the two solutions for particles of small diameter but an increasing relative error for particles of larger diameter. This is as one would intuitively expect given that the larger the diameter of the particle, the larger the Stokes number and, thus, the more the particles deviate from the fluid streamlines. The results presented in Table 3 suggest that for small particles it is valid to assume that the particles are in local equilibrium with the surrounding fluid.

Table 3. Particle Landing Position for Particles of Various Sizes Released at Height of X_3 = 100 m into Mean Ekman Boundary Layer over Flat Topography with Parameters Given in Table 1, Calculated Using Full Equations (10a)–(10c) and Approximate Equations (28a)–(28c)

Particle diameter (μm)	Particle landing position (x_{F1}, x_{F2}) (m)		Particle landing position (x_{L1}, x_{L2}) (m)		Error $\lvert \underline{x}_F - \underline{x}_L \rvert$ (m)	Relative error % $\frac{\lvert \underline{x}_F - \underline{x}_L \rvert}{\lvert \underline{x}_L \rvert}$
1000	226.83	133.62	187.46	115.71	42.97	19.5
750	276.97	165.56	242.79	149.86	37.36	13.0
500	384.72	233.14	356.62	220.12	30.77	7.34
250	772.43	473.64	751.48	463.85	22.97	2.60
100	2701.8	1665.0	2684.1	1656.7	19.37	0.61
75	4356.1	2686.1	4338.0	2677.6	19.83	0.39
50	9005.7	5555.8	8986.1	5546.6	21.61	0.20

Table 4. Particle Landing Position for Particles of Various Sizes Released at Height of X_3 = 100 m into Turbulent Ekman Boundary Layer over Flat Topography with Parameters Given in Table 1, Calculated Using Full Equations (10a)–(10c) and Approximate Equations (28a)–(28c)

Particle diameter (μm)	Particle landing position (x_{F1}, x_{F2}) (m)		Particle landing position (x_{L1}, x_{L2}) (m)		Error $\lvert \underline{x}_F - \underline{x}_L \rvert$ (m)	Relative error % $\frac{\lvert \underline{x}_F - \underline{x}_L \rvert}{\lvert \underline{x}_L \rvert}$
1000	232.46	137.00	188.09	115.98	48.86	22.1
750	281.89	166.80	239.78	147.33	46.11	16.4
500	392.55	242.12	341.86	211.90	59.00	14.7
250	819.47	512.22	748.34	471.29	82.01	9.27
100	2667.1	1618.8	2579.7	1571.1	99.52	3.29
75	4165.1	2513.6	4046.7	2450.1	134.2	2.84
50	8908.0	5298.9	8822.8	5256.1	95.08	0.93

To further investigate the approximations made in deriving Eqs. (28a)–(28c), the calculations were performed once more, but this time a turbulent fluctuation was added to the mean fluid flow. It was assumed that the eddy sizes are proportional to the mixing length, λ_m, with the magnitude of the turbulent fluctuations being 10% of the mean fluid flow. Table 4 shows the landing positions of particles of various sizes, calculated by using the fully implicit Runge-Kutta scheme and a small time step.

Again a similar pattern occurs; that is, for particles of smaller diameter, the relative difference between the two solutions is small, but an increasing relative error exists for particles of larger diameter. The relative error between the two solutions for a given particle size is larger than for the mean fluid flow field. This is easy to understand because every time a particle enters a turbulent eddy, it will take a finite time for its velocity to adjust to the new fluid velocity. Hence, a small error is encountered by assuming that the particle instantaneously adjusts to this new velocity—that is, that the particle is in local equilibrium with the surrounding fluid. This adjustment time is of the order of magnitude of the relaxation time of the particle; thus, the larger the particle the larger the error.

The preceding discussion suggests that for large particles, the full equations must be solved, but because of the relatively large Stokes numbers, this can be accomplished by implementing an explicit Runge-Kutta scheme. For small particles, small relative errors are introduced by assuming the particles are in local equilibrium with the fluid flow, thus still allowing the approximate Eqs. (28a)–(28c) to be solved by using an explicit Runge-Kutta scheme.

However, for highly turbulent fluid flows and for medium sized particles—that is, particles of diameter d of about $O(200\ \mu m)$—the relative error introduced by using Eqs. (28a)–(28c) may be unacceptably large. Therefore, the implicit Runge-Kutta scheme has to be implemented to solve Eqs. (10a)–(10c).

CONCLUSIONS

In this article, the numerical techniques needed to solve the equations governing the transport of particles within the atmospheric boundary layer have been studied. The accuracy of the numerical techniques presented was investigated by using Eqs. (32a) and (32b) to represent the fluid flow within the atmospheric boundary layer. It was shown that as the time step was reduced, the landing position of the particles converged and that an accurate solution could be obtained by using a relatively large time step, thereby justifying the use of the implicit Runge-Kutta scheme. For small particles, the relative difference in calculating their trajectories from the full governing Eqs. (10a)–(10c) and the approximate governing Eqs. (28a)–(28c) was small, with this difference being slightly larger for turbulent fluid flow. Hence, for small particles, the approximate Eqs. (28a)–(28c) should be solved by using an explicit time-stepping scheme.

For large particles, the full equations must be solved; however, because of the larger Stokes number, an explicit time-stepping scheme can be employed for the integration.on the other hand, for intermediate sized particles, say $O(200\ \mu m)$, in turbulent flow, the full Eqs. (10a)–(10c) need to be solved. This can be efficiently accomplished only by using an implicit numerical algorithm, as presented in this article. It is envisioned that these techniques will have a broad range of applicability across the engineering field.

REFERENCES

1. Ekman, V. W. (1905). On the influence of the Earth's rotation on ocean currents. *Arkiv. Mat. Astron. Fysik, 2*, 1–53.
2. Joynt, R. C., & Blackman, D. R. (1976). A numerical model of pollutant transport. *Atmos. Environ., 10,* 433–442.
3. Ley, A. J. (1982). A random walk simulation of two-dimensional turbulent diffusion in the neutral surface layer. *Atmos. Environ*., *16*, 2799–2808.
4. Monti, P., & Leuzzi, G. (1996). A closure to derive a three-dimensional well-mixed trajectory-model for non-Gaussion, inhomogeneous turbulence. *Boundary-Layer Meteorol*., *80,* 311–331.
5. Rodean, H. C. (1995). Turbulent-diffusion as a stochastic Lagrangian process. *Environmetrics*, *6*, 659–663.
6. Sawford, B. L. (1985). Lagrangian statistical simulation of concentration mean and fluctuation fields. *J. Clim. Appl. Meteorol*., *24*, 1152–1166.
7. Sherman, J., & Morrison, W. J. (1949). Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. *Ann. Math. Statist*., *20*, 621–629.
8. Sheuen, J. S., Chen, L. D., & Faeth, G. M. (1983). Evaluation of a stochastic model of particle dispersion in a turbulent round jet. *AIChE*, *29*, 167–170.
9. Stokes, G. G. (1851). On the effect of the internal friction of fluids on the motion of a pendulum. *Trans. Camb. Phil. Soc*., *9*, 8–106.
10. Taylor, G. I. (1921). Diffusion by continuous movements. *Proc. Lon. Math. Soc*., *20*, 196.
11. Thompson, D. J. (1990). A stochastic model for the motion of particle pairs in isotropic high-Reynolds-number turbulence, and its application to the problem of concentration variance. *J. Fluid Mech.*, *210*, 113–153.
12. Thompson, D. J. (1984). A random walk modelling of diffusion in inhomogeneous turbulence. *Quart. J. R. Met. Soc*., 110, 1107–1120.
13. Thompson, D. J. (1987). Criteria for the selection of stochastic models of particle trajectories in turbulent flows. *J. Fluid Mech.*, *180*, 529–556.
14. Thompson, R. (1971). Numerical calculation of turbulent diffusion. *Quart. J. Roy. Met. Soc*., *97*, 1107–1120.
15. Underwood, B. Y. (1990). Gravitational settling of particles dispersing from an elevated point source in the neutral surface layer of the atmosphere SRD R516. SRD. AEA Technology.
16. Underwood, B. Y. (1991). Deposition velocity and the collision model of atmospheric dispersion-I. Framework and application to cases with constant turbulent velocity scale. *Atmos. Environ*., *25A*, 2749–2759.
17. Van Dop, H. (1992). Buoyant plume rise in a Lagrangian framework. *Atmos. Environ*., *26A*, 1335–1346.
18. Vincent, J. H. (1989). *Aerosol sampling: science and practice.* Chichester, England: Wiley.
19. Wilson, J. D., & Sawford, B. L. (1995). Review of Lagrangian stochastic models for trajectories in the turbulent atmosphere, *Boundary-Layer Meterol.*, *78*, 192–210.
20. Wright, S. D., Elliott, L., & Ingham, D. B. (2001). The dispersion of airborne pollutants. *Hybrid Methods in Engineering*, *2,* 131-147.

Hybrid Methods in Engineering, vol. 3, pp. 67–73, 2001

A LINEAR PROGRAMMING MODEL FOR BOTTLENECK IDENTIFICATION IN CELLULAR MANUFACTURING

L. G. Azevedo Filho
Department of Mechanical Engineering, School of Engineering, Federal University of Rio de Janeiro, Brazil

R. Y. Qassim
Program of Mechanical Engineering, COPPE, Federal University of Rio de Janeiro, Brazil

The objective of this work is to demonstrate the error in assuming that exceptional elements (EEs) constitute bottlenecks in cellular manufacturing. This is achieved by developing a linear programming model and applying it to an example from industry.

INTRODUCTION

For more than two decades, there has been a transition ongoing in manufacturing from process-based to product-based fabrication and assembly of parts. Process-based manufacturing involves a *functional* layout of machines; for example, milling machines are grouped separately from drilling machines, and so forth. On the other hand, product-based manufacturing involves a *cellular* layout of machines; for example, cylindrical part types are manufactured together in a group of machines including, drilling, milling, finishing, etc. The principles and methods of cellular manufacturing have been established for some time; see, for example, Refs. [1–3].

Be that as it may, there exist a number of open questions related to the configuration and operation of manufacturing cells. One of these questions concerns the so-called exceptional elements (EEs), which are machines or parts that are *shared* by more than one cell. Vannelli and Kumar [4] and Kumar and Vannelli [5] were the first to model this problem. More recently, Kenn and Wei [6] and Shafer et al. [7] have contributed to the discussion on how to deal with EEs in cellular manufacturing. In all the work completed so far, it has been implicitly assumed that EE machines constitute bottlenecks and have to be treated as such. In this article, we show that this assumption is incorrect; that is, EE machines may or may not be bottlenecks. Towards this end, we employ the theory of constraints (TOC) as a framework for analysis and develop a linear programming model to identify bottlenecks in a cellular manufacturing system. An industrial example of the fabrication and assembly of mechanical precision components is used to illustrate the application of the model.

One of the authors (RYQ) wishes to thank CNPq for the award of a research scholarship. The other author (LGAF) wishes to thank the Department of Mechanical Engineering, School of Engineering, Federal University of Rio de Janeiro, Brazil, for financial support of his final-year undergraduate project, which constitutes part of the work reported in this article.

THEORY OF CONSTRAINTS

In the last decade, the TOC has been increasingly implemented as a manufacturing management technique in industry. The TOC is centered on the concept of a bottleneck—technically referred to as a capacity-constrained resource (CCR). In a manufacturing system, a CCR is defined as any machine or operation for which the processing capacity is less than or equal to the demand placed upon it. Naturally, the first and most important step in the implementation of the TOC consists of correctly identifying CCRs in the manufacturing system under consideration. It has been shown that linear programming provides an elegant and efficient tool for this purpose [8, 9]. For the reader interested in TOC concepts, Rahman [10] presents an up-to-date review.

CCR IDENTIFICATION MODEL IN CELLULAR MANUFACTURING

Consider a set of manufacturing cells that share a number of machines. These are the EEs. Now we introduce the following notation:

i	Index for cells; $i \in I$
j	Index for machines; $j \in J_i$
k	Index for part families; $k \in K$
m	Index for part types; $m \in M_K$
D_m	Demand for part type m
F_m	Unit profit margin of part type m
P_{mj}	Unit processing time of part type m in machine j
Q_m	Production quantity of part type m
R_j	Available time of machine j
S_j	Idle time of machine j

To identify CCRs in the cellular manufacturing system presented previously, the following linear programming (LP) model may be formulated:

Maximize
$$\sum_{m \in M} F_m Q_m \tag{1}$$

subject to
$$\sum_{m \in M} P_{mj} Q_m + S_j = R_j \quad \forall j \in J_i, i \in I \tag{2}$$

$$Q_m \le D_m \qquad \forall m \in M_k, k \in K \tag{3}$$

$$Q_m \ge 0 \qquad \forall m \in M_k, k \in K \tag{4}$$

The objective function (1) represents total profit of the cellular manufacturing system under consideration. The constraints (2) represent machine capacity limitations. The constraints (3) ensure that production does not exceed demand for any part type. The constraints (4) ensure nonnegativity of the production quantity of any part type.

Upon solving the LP model, any machine with $S_j = 0$ constitutes a CCR. Furthermore, if a CCR belongs to more than one cell, that CCR is also an EE.

INDUSTRIAL EXAMPLE

The LP model presented previously has been applied to a fabrication/assembly system for mechanical precision components. The system under consideration, which has been considered as part of a study in an industrial company performed by the authors, consists of two cells and two part families. Each cell consists of a number of machines and manual operations. For modeling purposes, machines and manual operations are considered to be equivalent. With a part type, we associate a unit processing time. With a machine, we associate an idle time and an available time. Information related to part families and machine cells is presented in Tables 1–3. The number of EEs is four; that is, machines 34, 45, 46, and 58.

Table 1. Part Type Demands and Profit Margins

Part type	Minimum demand	Profit Margin	Part type	Minimum demand	Profit Margin
A1	100	25.11	B1	10	36.8
A2	100	25.11	B2	10	36.8
A3	100	25.11	B3	10	36.8
A4	100	25.11	B4	10	36.8
A5	100	25.11	B5	10	36.8
A6	100	25.11	B6	10	36.8
A7	100	25.11			
A8	100	25.11			

Observations:
Maximum demanded of part type A family = 10000.
Maximum demanded of part type B family = 450.

Table 2. Part Type Processing Times

Part type	Minimum/ operation	Processing time	Part type	Minimum/ operation	Processing time
A1-A8	1	0.12	A1-A8	11	0.15
B1-B6	2	0.42	B1-B6	12	0.001
A1-A8	3	0.68	B1-B6	13	0.69
A1-A8	4	0.10	B1-B6	14	0.23
B1-B6	5	0.03	B1-B6	15	1.11
A1-A8	6	0.02	B1-B6	16	0.02
B1-B6	7	0.12	A1-A8	17	1.40
A1-A8	8	0.04	B1-B6	18	4.29
B1-B6	9	0.19	B1-B6	19	0.32
B1-B6	10	0.17	B1-B6	20	0.23

Table 2. Continued

Part type	Minimum/ operation	Processing time	Part type	Minimum/ operation	Processing time
B1-B6	21	0.11	B4	45	0.0575
B1-B6	22	0.40	B5	45	0.0475
A1-A8	23	0.37	B6	45	0.04
A1-A8	24	0.65	A1	46	0.15
B1-B6	25	0.28	A2	46	0.1575
B1-B6	26	0.97	A3	46	0.165
B1-B6	27	0.14	A4	46	0.1725
B1-B6	28	0.07	A5	46	0.1425
B1-B6	29	0.10	A6	46	0.12
B1-B6	30	0.16	A7	46	0.135
B1-B6	31	0.24	A8	46	0.1275
B1-B6	32	0.10	B1	46	0.17
A1-A8	33	0.28	B2	46	0.175
A1	34	0.12	B3	46	0.183
A2	34	0.126	B4	46	0.192
A3	34	0.132	B5	46	0.158
A4	34	0.138	B6	46	0.133
A5	34	0.114	A1-A8	47	0.09
A6	34	0.096	B1-B6	48	0.88
A7	34	0.108	B1-B6	49	0.20
A8	34	0.102	B1-B6	50	0.11
B1	34	0.001	B1-B6	51	0.001
B2	34	0.015	B1-B6	52	0.18
B3	34	0.16	B1-B6	53	0.02
B4	34	0.016	A1-A8	54	0.76
B5	34	0.013	A1-A8	55	0.02
B6	34	0.011	B1-B6	56	0.28
A1-A8	35	0.12	B1-B6	57	0.60
B1-B6	36	0.47	A1	58	0.12
B1-B6	37	0.43	A2	58	0.126
B1-B6	38	0.58	A3	58	0.132
B1-B6	39	0.30	A4	58	0.138
B1-B6	40	7.5	A5	58	0.114
B1-B6	41	8.57	A6	58	0.096
B1-B6	42	0.23	A7	58	0.108
B1-B6	43	0.16	A8	58	0.102
B1-B6	44	2.73	B1	58	0.12
A1	45	0.12	B2	58	0.123
A2	45	0.126	B3	58	0.129
A3	45	0.132	B4	58	0.135
A4	45	0.138	B5	58	0.11
A5	45	0.114	B6	58	0.094
A6	45	0.096	B1	59	10.0
A7	45	0.108	B2	59	10.5
A8	45	0.102	B3	59	11.0
B1	45	0.05	B4	59	11.5
B2	45	0.0525	B5	59	9.5
B3	45	0.055	B6	59	8.0

Table 2. Continued

Part type	Minimum/ operation	Processing time	Part type	Minimum/ operation	Processing time
A1	60	0.77	A1-A8	73	0.83
A2	60	0.808	A1-A8	74	0.83
A3	60	0.846	A1-A8	75	0.08
A4	60	0.886	A1-A8	76	0.48
A5	60	0.731	B1-B6	105	1.0
A6	60	0.615	B1-B6	106	1.0
A7	60	0.692	B1-B6	107	0.7
A8	60	0.654	B1-B6	108	0.67
A1-A8	61	0.15	B1-B6	109	0.3
A1-A8	62	0.15	B1-B6	110	0.74
A1-A8	63	0.38	B1-B6	111	0.42
A1-A8	64	0.55	B1-B6	112	0.56
A1-A8	65	0.48	B1-B6	113	0.32
A1-A8	66	0.78	B1-B6	114	0.38
A1-A8	67	0.52	B1-B6	115	0.09
A1-A8	68	0.47	B1-B6	116	0.24
A1-A8	69	0.41	B1-B6	117	0.1
A1-A8	70	0.21	B1-B6	118	0.1
A1-A8	71	0.83	B1-B6	119	0.05
A1-A8	72	0.83			

Table 3. Machine/Operation Available Time

Machine/ operation	Available time	Machine/ operation	Available time	Machine/ operation	Available time
1	8320	25	9600	49	7735
2	9600	26	9600	50	8400
3	7460	27	9600	51	9600
4	8400	28	9600	52	9600
5	9600	29	8400	53	9535
6	8400	30	9600	54	9600
7	9600	31	9600	55	7330
8	8345	32	8400	56	8400
9	9600	33	9600	57	8400
10	9600	34	9575	58	8225
11	8105	35	9600	59	8400
12	9600	36	9600	60	7895
13	9600	37	9600	61	8095
14	9600	38	9600	62	8045
15	9600	39	9600	63	8065
16	8155	40	9600	64	8215
17	9600	41	9600	65	8255
18	9600	42	8400	66	9460
19	8400	43	9600	67	9235
20	8400	44	9600	68	8165
21	9600	45	9410	69	9540
22	8280	46	9400	70	9180
23	9595	47	7755	71–119	9600
24	9600	48	9600		

We solved the CCR identification LP model for this industrial example by using MATHEMATICA [11]. Results summarized in Table 4 show that there is only one CCR—machine number 60. Thus, the CCR in this case is *not* an EE.

Table 4. Machine/Operation Idle Time

Machine/ operation	Idle time	Machine/ operation	Idle time	Machine/ operation	Idle time
1	7120	41	5744	81	600
2	9411	42	8296	82	8400
3	660	43	9528	83	8400
4	6400	44	8372	84	2000
5	9586	45	8190	85	9456
6	8200	46	7827	86	9470
7	9546	47	6855	87	9595
8	7945	48	9204	88	9578
9	9515	49	7645	89	9524
10	9524	50	8351	90	9497
11	6605	51	9600	91	9465
12	9600	52	9519	92	9497
13	9290	53	9526	93	9420
14	9492	54	2000	94	9515
15	9101	55	7130	95	9465
16	7955	56	8274	96	9411
17	8970	57	8130	97	9546
18	7670	58	6973	98	9595
19	8256	59	3895	99	9578
20	8296	60	211	100	9519
21	9550	61	6595	101	9082
22	4280	62	6545	102	9555
23	5895	63	4265	103	9559
24	9308	64	2715	104	9559
25	9474	65	3455	105	9150
26	9164	66	1660	106	9150
27	9537	67	4035	107	9285
28	9569	68	3465	108	9299
29	8355	69	5440	109	9465
30	9528	70	7080	110	9267
31	9492	71	1300	111	9389
32	7400	72	1300	112	9456
33	9474	73	1300	113	9456
34	8373	74	1300	114	9078
35	8400	75	8800	115	9559
36	9389	76	4800	116	9492
37	9406	77	9300	117	9510
38	9339	78	8800	118	9510
39	9465	79	6600	119	9578
40	6225	80	8400		

CONCLUSIONS

The assumption that a bottleneck and an EE are necessarily one and the same has been shown to be invalid by applying a linear programming model to an industrial example.

REFERENCES

1. Chase, R. B., & Aquilano, N. J. (1995). *Production and operations management: manufacturing and services*, 7th ed. Irwin.
2. Wemmerlöv, U., & Hyar, N. L. (1989). Cellular manufacturing in U.S. industry: a survey of users. *International Journal of Production Research*, *27,* 1511–1530.
3. Marsh, R. F., Shafer, S. M., & Meredith, J. R. (1999). A comparison of cellular manufacturing research presumptions with practice. *International Journal of Production Research*, *37*, 3119–3136.
4. Vanneli, A., & Kumar, K. R. (1986). A method for finding bottleneck cells for grouping part-machine families. *International Journal of Production Research*, *24*, 387–400.
5. Kumar, K. R., & Vanneli, A. (1987). Strategic Subcontracting for efficient disaggregated manufacturing. *International Journal of Production Research*, *12*, 1715–1728.
6. Kern, G. M., & Wei, J. C. (1991). The cost of eliminating EEs in group technology cell formation. *International Journal of Production Research*, *29*, 1535–1547.
7. Shafer, S. M., Kern, G. M., & Wei, J. C. (1992). A mathematical programming approach for dealing with EEs in cellular manufacturing. *International Journal of Production Research*, *30*, 1029–1036.
8. Luebbe, R., & Firch, B. (1992). Theory of constraints and linear programming: a comparison. *International Journal of Production Research*, *30*, 1471–1478.
9. Plenert, G. (1993). Optimizing theory of constraints when multiple constrained resources exist. *European Journal of Operational Research*, *70*, 126–133.
10. Rahman, S. (1998). Theory of constraints: a review of the philosophy and its applications. *International Journal of Operations and Production Management*, *18*, 336–355.
11. Wolfram, S. (1999). *The mathematica book*, 4th ed. Wolfram Media/Cambridge University Press.

Hybrid Methods in Engineering, vol. 3, pp. 75–87, 2001

SOLVING FLEXIBLE FLOW SHOP PROBLEMS BY COMBINING LPT AND GUPTA SCHEDULING ALGORITHMS

Tzung-Pei Hong
Department of Information Management, I-Shou University, Kaohsiung, 84008, *Taiwan, R.O.C.; tphong@isu.edu.tw*

Chan-Lon Wang & Shyue-Liang Wang
Department of Information Management, I-Shou University, Kaohsiung, 84008, *Taiwan, R.O.C; phd9004@cs.nchu.edu.tw, slwang@isu.edu.tw*

Scheduling is an important process widely used in manufacturing, production, management, computer science, and so on. Appropriate scheduling not only reduces manufacturing costs but also reduces possibilities for violating due dates. Finding good schedules for given sets of jobs can thus help factory supervisors effectively control job flows and provide solutions for job sequencing. In simple flow shop problems, each machine operation center includes just one machine. If at least one machine center includes more than one machine, the scheduling problem becomes a flexible flow shop problem. Flexible flow shops can thus be thought of as generalizations of simple flow shops. In the past, Sriskandarajah and Sethi [6] proposed a heuristic algorithm for solving flexible flow shop problems for two machine centers. In this article, we extend their algorithm to solve flexible flow shop problems for more than two machine centers. The heuristic Gupta algorithm is adopted as the kernel for achieving this purpose. Because this problem is a nondeterministic-polynomial (NP)-complete problem, optimal solutions seem unnecessary, especially when the number of jobs is large.

KEYWORDS: scheduling, flexible flow shop, LPT scheduling, Gupta scheduling

INTRODUCTION

In simple flow shop problems, each machine center has just one machine. If at least one machine center has more than one machine, the problem is called a flexible flow shop problem [2]. Flexible flow shops are thus generalizations of simple flow shops. Scheduling jobs in flexible flow shops is considered a non-deterministic-polynomial (NP)-complete problem [1, 5].

In the past, Sriskandarajah and Sethi [6] proposed a heuristic algorithm for solving flexible flow shop problems for two machine centers. In this article, we extend their algorithm to solve flexible flow shop problems for more than two machine centers. A scheduling algorithm, which combines the longest processing time (LPT) and Gupta algorithms, is proposed for flexible flow shops with more than two machine centers. The LPT method is first used to assign jobs to each machine group (flow shop). The Gupta algorithm is then used to deal with job sequencing. Experimental results show that our proposed method can efficiently solve large-sized scheduling problems, although the obtained

The authors would like to thank the anonymous referees for their very constructive comments.

makespans may be a little larger than the optimal solutions. The method is thus more suitable for real applications.

The remainder of this article is organized as follows. First, we present the assumptions and notation used in this article. Next, we propose an algorithm for a flexible flow shop with more than two machine centers and give an example to illustrate the proposed scheduling algorithm. Then we describe experiments for verifying the performance of the proposed heuristic algorithm. Finally, we give our conclusions.

ASSUMPTIONS AND NOTATION

The assumptions and notation used in this article are presented in this section.

Assumptions

- Jobs are not preemptive.
- Each job has m ($m > 2$) tasks with processing times executed, respectively, on each of m machine centers.
- Each machine center has the same number of homogeneous machines.

Notation

n	The number of jobs
m	The number of tasks in each job
mc_i	The ith machine center, $i = 1, 2, \ldots, m$
p	The number of machines in each machine center
F_i	The ith allocated machine group (flow shop), $i = 1$ to p
F_{ji}	The jth machine of the flow shop F_i, $j = 1$ to m
f_i	The completion time of the ith flow shop
cf_{ij}	The completion time of the ith machine in the jth flow shop
T_{ij}	The ith task for the jth job, $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$
t_{ij}	The execution time of T_{ij}
tt_j	The total execution time of the jth job
Q_{Fi}	The group of jobs executed on the ith flow shop
ff	The final completion time of the whole schedule

THE FLEXIBLE FLOW SHOP ALGORITHM FOR MORE THAN TWO MACHINE CENTERS

Sriskandarajah and Sethi [6] have proposed a scheduling algorithm for two machine centers. Here, we generalize that algorithm to solve flow shop problems of more than two machine centers. The flexible flow shop algorithm for more than two machine centers is based on the LPT [4, 5] and the Gupta [3], [5] algorithms to manage job scheduling. The flexible flow shop algorithm is decomposed into three parts. The first part forms the machine groups, each of which contains a machine from each center. The second part uses the LPT method to assign jobs to each machine group (flow shop). The third part deals with job sequencing and timing using the Gupta algorithm. The proposed flexible flow shop algorithm is stated next.

A Gupta-Based Flexible Flow Shop Algorithm

Input: A set of n jobs, each having m ($m > 2$) tasks, to be executed respectively on each of m machines centers with p homogenous machines

Output: A schedule with a completion time

Part 1: Forming the Machine Groups

Step 1: Form p machine groups, each of which contains one machine from each machine center. Each machine group can be thought of as a simple flow shop $F_1, F_2, \ldots, F_p$.

Step 2: Initialize the completion time of each flow shop $f_1, f_2, \ldots, f_p$ to zero.

Step 3: For each job J_j, $1 \leq j \leq n$, find its total execution time $tt_j = t_{1j} + t_{2j} + \ldots + t_{mj}$.

Part 2: Assigning Jobs to Machine Groups

Step 4: Sort the jobs in descending order of processing time tt_j; if any two jobs have the same tt_j values, sort them in an arbitrary order.

Step 5: Find the flow shop F_i with the minimum processing time f_i among all the flow shops; if two flow shops have the same minimum f_i value, choose one arbitrarily.

Step 6: Assign the first job J_j in the sorted list to the chosen flow shop F_i that has the minimum completion time f_i among all m flow shops.

Step 7: Add the total time tt_j of job J_j to the required total time of the chosen flow shop, F_i; that is:

$$f_i = f_i + tt_j$$

Step 8: Remove job J_j from the job list.

Step 9: Repeat Steps 5 through 8 until the job list is empty.

After Step 9, jobs are clustered into p groups and are allocated to the p machine groups (flow shops).

Part 3: Dealing with Job Sequencing in Each Flow Shop

Step 10: For the ith group of jobs (i = 1 to p), form the subgroup of jobs U_i that take less time on the first machine than on the last, such that $U_i = \{k \mid t_{1k} < t_{mk}, J_k \in$ the ith group of jobs$\}$.

Step 11: For the ith group of jobs (i = 1 to p), form the subgroup of jobs V_i that take less time on the last machine than on the first or that take equal time on both machines, such that $V_i = \{j \mid t_{mj} \leq t_{1j}, J_j \in$ the ith group of jobs$\}$.

Step 12: For each job J_h in U_i, find the minimum of $(t_{kh} + t_{(k+1)h})$ for k = 1 to $m - 1$; restated, set:

$$\pi_h = \min_{k=1}^{(m-1)} \left(t_{kh} + t_{(k+1)h} \right)$$

Step 13: For each job J_r in V_i, find the minimum of $(t_{kr} + t_{(k+1)r})$ for k = 1 to $m - 1$; restated, set:

$$\pi_r = \min_{k=1}^{(m-1)} \left(t_{kr} + t_{(k+1)r} \right)$$

Step 14: Sort the jobs in U_i in ascending order of π_p's; if two or more jobs have the same value of π_p, sort them in an arbitrary order.
Step 15: Sort the jobs in V_i in descending order of π_r's; if two or more jobs have the same value of π_r, sort them in an arbitrary order.
Step 16: For each flow shop F_i, schedule the jobs in the sorted order of U_i and then in the sorted order of V_i; denote the job list scheduled in F_i as Q_{F_i}.
Step 17: In each flow shop F_i, set the initial completion time of the machines cf_{gi} (g = 1 to m, i = 1 to p) to zero.
Step 18: For each flow shop F_i, assign the first job J_j in Q_{F_i} to the machines such that J_{1j} is assigned to F_{1i}, J_{2j} is assigned to F_{2i}, . . . , J_{mj} is assigned to F_{mi}.
Step 19: Add the processing time t_{1j} to the completion time of the first machine cf_{1i}; that is:

$$cf_{1i} = cf_{1i} + t_{1j}$$

Step 20: Set $cf_{(k+1)i} = \max(cf_{ki}, cf_{(k+1)i}) + t_{(k+1)j}$, for k = 1 to $(m - 1)$.
Step 21: Remove job J_j from Q_{F_i}.
Step 22: Repeat Steps 19 through 21 until Q_{F_i} is empty.
Step 23: Set the final completion time of each flow shop f_i = the completion time of the mth machine cf_{mi}.
Step 24: Find the maximum final completion time ff among the completion time of all the flow shops.

After Step 24, scheduling is finished and a total completion time ff has been found.

AN EXAMPLE

Assume that eight jobs, J_1 to J_8, each having three tasks (J_{1j}, J_{2j}, J_{3j}), are to be scheduled for execution in three operations. Each operation is executed by a machine at the corresponding machine center. Each machine center includes three homogeneous machines. Assume the execution times of these jobs are as listed in Table 1. The algorithm then proceeds in three parts.

Part 1: Forming the Machine Groups

Step 1: Form three machine groups, F_1, F_2, F_3, each of which is thought of as a three-machine flow shop. Without lose of generality, we may assume the flow shops are constructed as follows:

$$F_1 \rightarrow \{m_{11} + m_{12} + m_{13}\}$$
$$F_2 \rightarrow \{m_{21} + m_{22} + m_{23}\}$$
$$F_3 \rightarrow \{m_{31} + m_{32} + m_{33}\}$$

where m_{ij} is the ith machine in the jth center.

Table 1. Processing Times for the Eight Jobs

Job	Execution time		
	t_{1j}	t_{2j}	t_{3j}
J_1	8	12	4
J_2	6	8	9
J_3	12	14	17
J_4	17	11	5
J_5	4	6	11
J_6	11	9	8
J_7	9	8	8
J_8	2	8	3

Step 2: Initialize $f_1 = f_2 = f_3 = 0$, where f_i is the initial completion time of F_i.
Step 3: For each job J_j, $j = 1$ to 8, find its total execution time $tt_j = t_{1j} + t_{2j} + t_{3j}$. For example, the total processing time of Job 1 is calculated as:

$$tt_1 = t_{11} + t_{21} + t_{31} = 8 + 12 + 4 = 24$$

The total processing times of the other jobs can be similarly found; the results are listed in Table 2.

Part 2: Assigning Jobs to Machine Groups

Step 4: Sort the jobs J_1 to J_8 in descending order of the total processing time (tt_j). The following results are obtained:

$$\text{Job list} = \{J_3, J_4, J_6, J_7, J_1, J_2, J_5, J_8\}$$

Step 5: Find the minimum f_i among all the flow shops F_1 to F_3. Because the total processing time of all the flow shops is equal to zero, any one of them can be chosen arbitrarily. Without lose of generality, assume F_1 is chosen.
Step 6: Assign the first job J_3 in the sorted list to the chosen flow shop F_1.

Table 2. Total Processing Times of the Eight Jobs

Job_j	Total processing time tt_j
J_1	24
J_2	23
J_3	43
J_4	33
J_5	21
J_6	28
J_7	25
J_8	13

Table 3. Flow Shops with Allocated Jobs and Total Processing Time

Flow shop$_i$	Allocated jobs	Total processing time
F_1	J_3	43
F_2	Empty	0
F_3	Empty	0

Step 7: Add the total processing time tt_3 of job J_3 to the required total time of the chosen flow shop F_1. Thus:

$$f_1 = f_1 + tt_3 = 0 + 43 = 43$$

After Step 7, the results of allocating jobs to flow shops are shown in Table 3.

Step 8: Remove the job J_3 from the job list. After J_3 is removed, the job list is then as follows:

$$\text{job list} = \{J_4, J_6, J_7, J_1, J_2, J_5, J_8\}$$

Step 9: Repeat Steps 5 through 8 until the job list is empty.

After Step 9, jobs are clustered into three groups and are respectively allocated to the three flow shops. Results are shown in Table 4.

Part 3: Dealing with Job Sequencing in Each Flow Shop

Step 10: Form the three subgroups of jobs U_i that take less time on the first machine than on the last. Results are as follows:

$$U_1 = \{J_2, J_3\}$$
$$U_2 = \{J_8\}$$
$$U_3 = \{J_5\}$$

Step 11: Form the three subgroups of jobs V_i that take less time on the last machine than on the first or that take equal time on both machines. Results are as follows:

Table 4. The Job Set of Each Flow Shop

Flow shop$_i$	Jobs allocated
F_1	J_3, J_2
F_2	J_4, J_1, J_8
F_3	J_6, J_7, J_5

$$V_1 = \{\}$$
$$V_2 = \{J_1, J_4\}$$
$$V_3 = \{J_6, J_7\}$$

Step 12: For each job J_h in U_i, find the value π_h as:

$$\pi_h = \min_{k=1}^{(m-1)} (t_{kh} + t_{(k+1)h})$$
$$= \min((t_{1h} + t_{2h}), (t_{2h} + t_{3h}))$$

Therefore, in U_1, $\pi_2 = \min\{14, 17\} = 14$, and $\pi_3 = \min\{26, 31\} = 26$. Similarly, in U_2, $\pi_8 = \min\{10, 11\} = 10$, and in U_3, $\pi_5 = \min\{10, 17\} = 10$.

Step 13: Similarly, in V_2, $\pi_1 = \min\{20, 16\} = 16$ and $\pi_4 = \min\{28, 16\} = 16$; in V_3, $\pi_6 = \min\{20, 17\} = 17$ and $\pi_7 = \min\{17, 16\} = 16$.
Step 14: Sort the jobs in U as:

$$U_1 = \{J_2, J_3\}$$
$$U_2 = \{J_8\}$$
$$U_3 = \{J_5\}$$

Step 15: Sort the jobs in V as:

$$V_1 = \{\}$$
$$V_2 = \{J_4, J_1\}$$
$$V_3 = \{J_6, J_7\}$$

Step 16: For each flow shop F_i, schedule the jobs in the sorted order of U_i and then in the sorted order of V_i. The scheduled flow shops Q_{F_1}, Q_{F_2}, Q_{F_3} are formed as follows:

Flow shop 1: $Q_{F_1} = \{J_2, J_3\}$
Flow shop 2: $Q_{F_2} = \{J_8, J_4, J_1\}$
Flow shop 3: $Q_{F_3} = \{J_5, J_6, J_7\}$

Step 17: In each flow shop F_i, set the initial completion time of each machine (cf_{ji}) to zero.
Step 18: For each flow shop F_1, F_2, and F_3, assign the first jobs J_2, J_8, J_5 in Q_{F_1}, Q_{F_2}, Q_{F_3} to the machines, respectively.
Step 19: Add the processing time t_{12}, t_{18}, t_{15} to the completion time of the machine cf_{11}, cf_{12}, cf_{13}. Results are as follows:

Flow shop 1: cf_{11} $cf_{11} + t_{12} = 0 + 6 = 6$

Flow shop 2: $cf_{12} = cf_{12} + t_{18} = 0 + 2 = 2$

Flow shop 3: $cf_{13} = cf_{13} + t_{15} = 0 + 4 = 4$

Step 20: Calculate the completion time of Jobs J_2, J_8, J_5 on other machines. As mentioned previously, in flow shop problems, jobs are processed by series of machine centers in exactly the same order. That is, one machine center does the first operation on each job, another does the second, and so on. However, the processing start time of the second operation is the maximum between the current completion time of the first operation and the last completion time of the second operation. Results are as follows:

Flow shop 1:

$cf_{21} = \max(cf_{11}, cf_{21}) + t_{22} = 6 + 8 = 14$

$cf_{31} = \max(cf_{21}, cf_{31}) + t_{32} = 14 + 9 = 23$

Flow shop 2:

$cf_{22} = \max(cf_{12}, cf_{22}) + t_{28} = 2 + 8 = 10$

$cf_{32} = \max(cf_{22}, cf_{32}) + t_{38} = 10 + 3 = 13$

Flow shop 3:

$cf_{23} = \max(cf_{13}, cf_{23}) + t_{25} = 4 + 6 = 10$

$cf_{33} = \max(cf_{23}, cf_{33}) + t_{35} = 10 + 11 = 21$

Table 5. Completion Time of Each Machine

Flow shop 1:

Job_j	cf_{11}	cf_{21}	cf_{31}
J_2	6	14	23
J_3	18	32	49

Flow shop 2:

Job_j	cf_{12}	cf_{22}	cf_{32}
J_8	2	10	13
J_4	19	30	35
J_1	27	42	46

Flow shop 3:

Job_j	cf_{13}	cf_{23}	cf_{33}
J_5	4	10	21
J_6	15	26	34
J_7	24	32	40

Table 6. Completion time of each flow shop

Flow shop$_i$	Completion time (f_i)
F_1	49
F_2	46
F_3	40

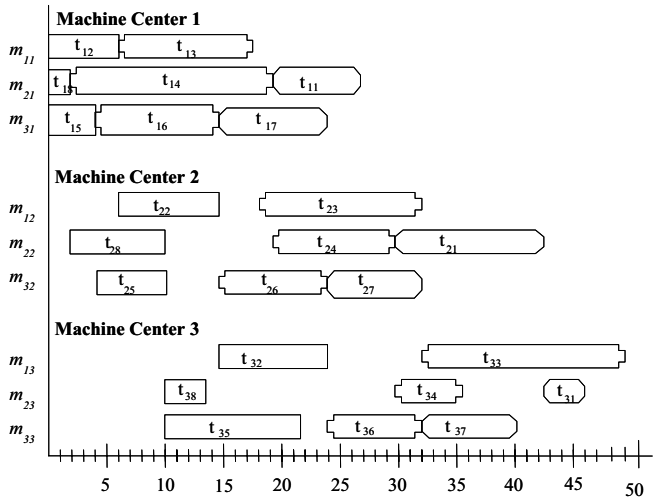

Fig. 1 The final scheduling result in the example.

Step 21: Remove J_2, J_8, and J_5 from Q_{F1}, Q_{F2} and Q_{F3}, respectively.
Step 22: Repeat Steps 10 through 21 for the other jobs. The results are then as shown in Table 5.
Step 23: Set the final completion time of each flow shop f_i = the completion time of the mth machine cf_{mi}. Thus, $f_1 = cf_{31}$, $f_2 = cf_{32}$, and $f_3 = cf_{33}$.
The results are as shown in Table 6.
Step 24: Find the maximal final completion time ff among the completion time of all the flow shops. We can thus get

$$ff = 49$$

ff is then output as the final total completion time, and the resulting schedule is shown in Fig. 1.

EXPERIMENTS

This section describes experiments conducted to show the performance of the proposed heuristic algorithm. The experiments were implemented by using Visual C++ on a Pen-

tium III 500 PC. Also presented are experiments made to compare the time required by the proposed heuristic algorithm with that required by the dynamic programming approach. Five sets of problems were tested, respectively, for 3 to 7 jobs. Each job had three tasks and each machine center had two homogeneous machines. The execution time of each task was randomly generated in the range of 5 to 50. Each set of problems was executed for 20 tests and the makespans and computation times were measured. The dynamic programming approach did not work for more than seven jobs in our environments because of the large amount of computation time.

The dynamic programming approach considered all possible combinations and used pruning techniques to increase its efficiency. The makespans obtained from the dynamic

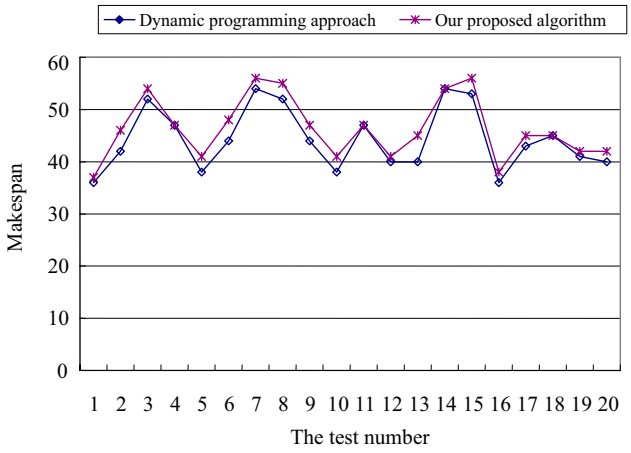

Fig. 2 Makespans of 20 tests for three jobs.

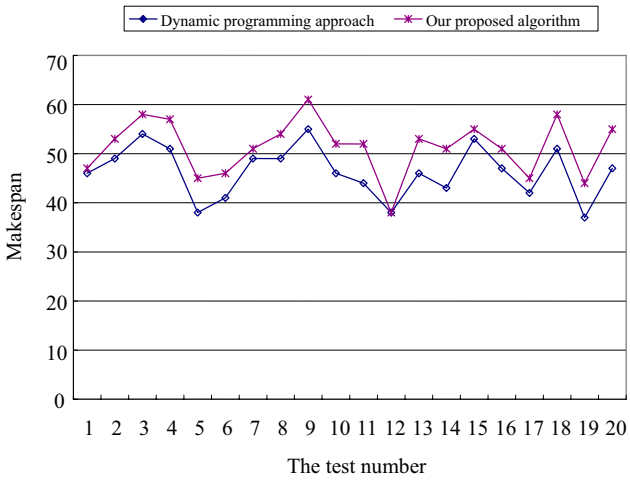

Fig. 3 Makespans of 20 tests for four jobs.

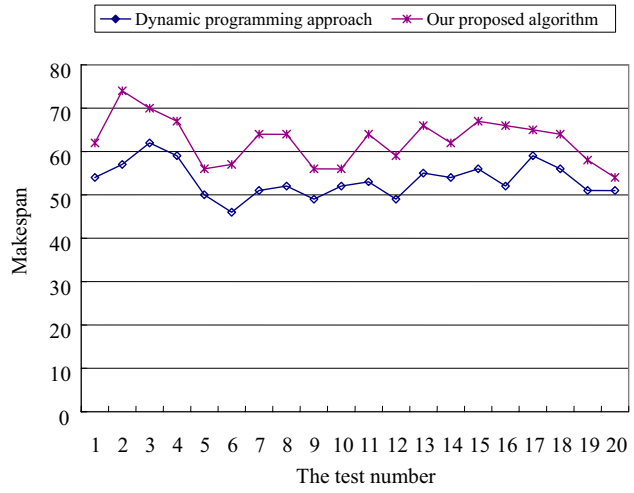

Fig. 4 Makespans of 20 tests for five jobs.

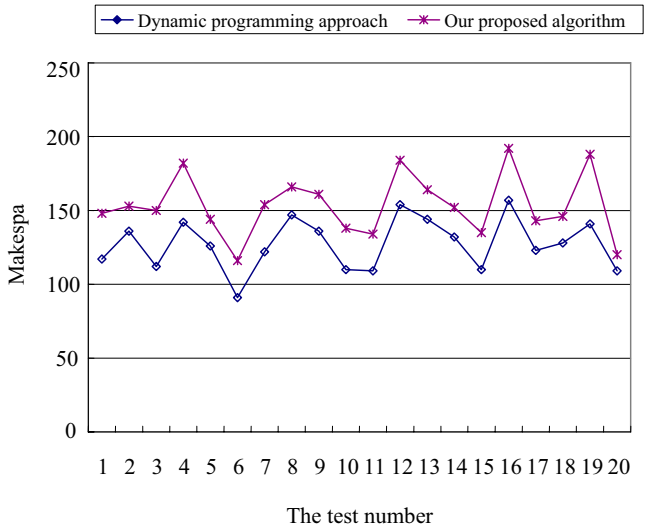

Fig. 5 Makespans of 20 tests for six jobs.

programming approach were optimal. The makespans for problems of three to seven jobs obtained by our proposed method and by the dynamic programming approach are shown, respectively, in Figs. 2 to 6.

The average execution times for problems of three to seven jobs are shown in Fig. 7. The average execution times for problems of 8 to 12 jobs by our proposed algorithm are also shown there.

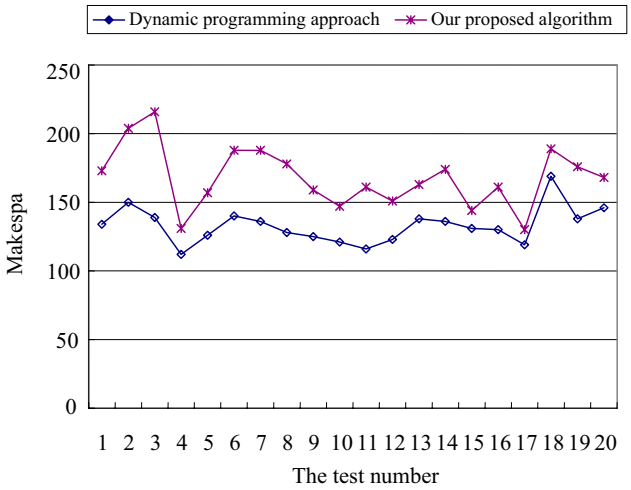

Fig. 6 Makespans of 20 tests for seven jobs.

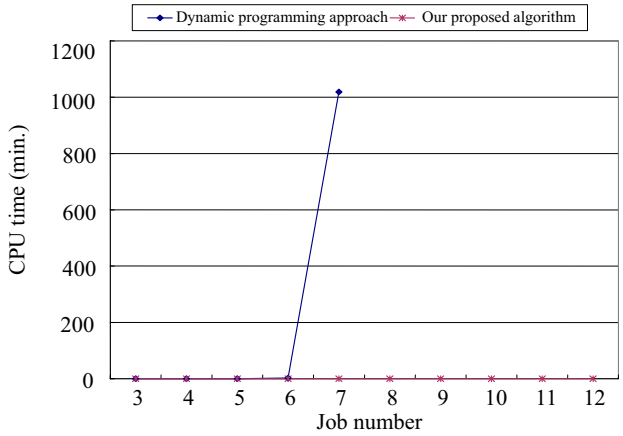

Fig. 7 The average CPU times for processing different numbers of jobs.

From the preceding figures, it is easily seen that our proposed algorithm obtained slightly larger makespans than did the dynamic programming approach. The computational time needed by the dynamic programming approach was, however, much larger than that needed by our proposed algorithm, especially when the job number was large. Actually, because the flexible flow shop problem is an NP-hard problem, the dynamic programming approach can work only for a small number of jobs. Our proposed algorithm can solve this problem, thus being more suitable for real applications.

CONCLUSION

Appropriate scheduling can not only reduce manufacturing costs but also reduce the possibility of violating due dates. Finding good schedules for given sets of jobs can thus help factory supervisors control job flows and provide for nearly optimal job sequencing.

Scheduling jobs in flexible flow shops has long been known as an NP-complete problem. In this article, we have proposed a heuristic scheduling algorithm for scheduling jobs in flexible flow shops with more than two machine centers. The scheduling results can help system managers have broader views of scheduling, so that they may conduct appropriate analyses. In the future, we will consider other task constraints, such as setup times, due dates, and priorities.

REFERENCES

1. Chung, S. C., & Liao, D. Y. (1992). Scheduling flexible flow shops with no setup effects. *The 1992 IEEE International. Conference on Robotics and Automation*, 1179–1184.
2. Dudek, R. A., Panwalkar, S. S., & Smith, M. L. (1992). The lessons of flowshop scheduling research. *Operations Research*, *40,* 7–13.
3. Hong, T. P., & Chuang, T. N. (1996). A fuzzy Gupta scheduling algorithm. *The Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, 320–324.
4. Hong, T. P., Huang, C. M., & Yu, K. M. (1998). LPT scheduling for fuzzy tasks, *Fuzzy Sets and Systems, 97*, 277–286.
5. Morton, T. E., & Pentico, W. E. (1993). *Heuristic scheduling systems with applications to production systems and project management*. New York: Wiley.
6. Sriskandarajah, C., & Sethi, S. P. (1989). Scheduling algorithms for flexible flow shops: worst and average case performance, *European Journal of Operational Research*, *43*, 143–160.

Hybrid Methods in Engineering, vol. 3, pp. 89–94, 2001

CONVERGENCE ACCELERATION OF INTEGRAL TRANSFORM SOLUTIONS

M. D. Mikhailov & R. M. Cotta
Mechanical Engineering Department—EE/COPPE/UFRJ, Universidade Federal do Rio de Janeiro, Cidade Universitária, CP 68.503, Rio de Janeiro, RJ, 21945-970, Brasil; mikhailov@lttc.coppe.ufrj.br, cotta@lttc.coppe.ufrj.br

A number of techniques, intended to accelerate convergence of eigenseries solution, are investigated. The test problem describes transient temperature distribution along fins of constant profile. The solutions are presented in a form that permitted to work with time independent functions. For any solution are used as many terms as necessary to obtain 5 significant digits. It is shown that the splitting-up solution offers ideal convergence in contrast to the slowly convergent classical integral transform solution. The convergence of single filter solution, double filter solution, integral balance solution, and combined filtering and integral balance solution are acceptable for practical applications. It is demonstrated that the Shanks transformation may considerably accelerate the convergence of eigenseries solution.

INTRODUCTION

The generalized integral transform technique (GITT) is well documented [1–3] and widely used to solve linear as well as non-linear diffusion and convection-diffusion models. A major aspect in the practical implementation of GITT is the need for improving the convergence behavior of the resulting eigenfunction series by splitting up the original problem [4, 5]. However, the expressions so developed are limited to linear problems with time-dependent source functions represented by exponentials and q-order polynomials. For this reason different filtering schemes were employed in the related literature. The corresponding references are given in [2], where a number of convergence acceleration schemes are investigated. All such convergence acceleration efforts are made in order to increment the rate of convergence by extracting partial sums of the series in a closed form expression.

Another possible way, only mentioned in [2], is to employ special techniques to directly enhance the summation itself, such as Shanks transformations [6], Lanczos σ factor [7] or other manipulations of the series [8].

This article is prepared by using the *Mathematica* software system [9]. The goal is twofold: to investigate again all acceleration schemes used in [2] and to apply the Shanks transformation [6].

As a test the following solutions of the transient fin problem are used:

- splitting-up solution,
- classical integral transform solution,
- single filter solution,
- double filter solution,
- integral balance solution,
- combined filtering and integral balance solution.

The above solutions are rewritten in an unified form that permitted to investigate the convergence of time independent functions. In contrast to [2] here we use for any solution as many terms as necessary in the expansions to obtain the results with 5 converged significant digits.

It is then demonstrated that the Shanks transformation accelerates considerably the convergence of eigenseries solutions.

TEST PROBLEM

As a test case we select the same dimensionless problem as in [2, pp. 51–58, pp. 359–340].

The problem describes the temperature distribution along a fin of constant profile

$$\frac{\partial\theta[\xi,\tau]}{\partial\tau}=\frac{\partial^2\theta[\xi,\tau]}{\partial\xi^2}-\mathrm{M}^2\theta[\varepsilon,\tau] \tag{1}$$

The boundary conditions express the base's temperature decay following a shutdown operation of the associated heat transfer equipment

$$\theta[0,\tau]=E^{-b\tau} \tag{2}$$

$$\frac{\partial\theta[1,\tau]}{\partial\xi}=0 \tag{3}$$

As an initial condition the steady-state temperature profile for constant temperature at the fin's base is adopted

$$\theta[\xi,0]=\cosh[\mathrm{M}(1-\xi)]\,\mathrm{sech}[\mathrm{M},\,\xi] \tag{4}$$

The different convergence improving strategies, proposed in [2], give different solutions written in the unified form

$$\theta[\xi,\tau]=æ^{-b\tau}f[\xi]-2b æ^{-\mathrm{M}^2\tau}\sum_{i=1}^{n}\frac{\lambda_i\sin[\xi\lambda_i]}{(\lambda_i^2+\mathrm{M}^2-b)(\lambda_i^2+\mathrm{M}^2)}æ^{-\lambda_i^2\tau} \tag{5}$$

where the eigenvalues are given by

$$\lambda_i=\left(1-\frac{1}{2}\right)\pi \tag{6}$$

The classical integral transform solution [2, p. 53] coincides with Eq. (5) when

$$f[\xi] = 2\sum_{i=1}^{n} \frac{\lambda_i \sin[\xi \lambda_i]}{\lambda_i^2 + \mathrm{M}^2 - b} \tag{7}$$

Since the eigenvalue problem is homogeneous, the solution (7) cannot reproduce the non-homogeneous boundary condition (2). This fact markedly affects the convergence behavior of the solution, particularly at points approaching $\xi = 0$.

The single filter solution [2, p. 55] gives

$$f[\xi] = \cosh[\mathrm{M}(1-\xi)]\,\mathrm{sech}[\mathrm{M}] + 2b\sum_{i=1}^{n} \frac{\lambda_i \sin[\xi \lambda_i]}{(\lambda_i^2 + \mathrm{M}^2 - b)(\lambda_i^2 + \mathrm{M}^2)} \tag{8}$$

The double filter solution [2, p. 57] yields

$$\begin{aligned} f[\xi] &= \cosh[\mathrm{M}(1-\xi)]\,\mathrm{sech}[\mathrm{M}] \\ &+ \frac{b}{2\mathrm{M}}((\xi - \mathrm{sech}[\mathrm{M}]^2)\sinh[\mathrm{M}\xi] - \xi\cosh[\mathrm{M}\xi]\tanh[\mathrm{M}]) \\ &+ 2b^2\sum_{i=1}^{n} \frac{\lambda_i \sin[\xi \lambda_i]}{(\lambda_i^2 + \mathrm{M}^2 - b)(\lambda_i^2 + \mathrm{M}^2)^2} \end{aligned} \tag{9}$$

The integral balance solution [2, p. 58] gives

$$f[\xi] = 1 + 2(b - \mathrm{M}^2)\sum_{i=1}^{n} \frac{\sin[\xi \lambda_i]}{(\lambda_i^2 + \mathrm{M}^2 - b)\lambda_i} \tag{10}$$

The combined filtering and integral balance solution [2, p. 58] yields

$$\begin{aligned} f[\xi] &= \left(1 - \frac{b}{\mathrm{M}^2}\right)\cosh[\mathrm{M}(1-\xi)]\sec[\mathrm{M}] + \frac{b}{\mathrm{M}^2} \\ &+ 2b(b - \mathrm{M}^2)\sum_{i=1}^{n} \frac{\lambda_i \sin[\xi \lambda_i]}{(\lambda_i^2 + \mathrm{M}^2 - b)\lambda_i^2(\lambda_i^2 + \mathrm{M}^2)} \end{aligned} \tag{11}$$

The splitting-up solution [2, p. 340] gives:

$$f[\xi] = \cos[\sqrt{b - \mathrm{M}^2}\,(1-\xi)]\sec[\sqrt{b - \mathrm{M}^2}\,] \tag{12}$$

SEQUENCE SUM

Let S be a sequence (n = 1, 2, . . .) of numbers or functions, for example

$$S = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9\} \tag{13a}$$

Let k be a positive integer and

$$\Delta A_n = A_{n+1} - A_n \tag{13b}$$

The following non-linear operator gives a new sequence for $n = k + 1, k + 2, \ldots$ called the kth transform of S [6]

$$\varepsilon_k[n, S] = \frac{\mathrm{Det}\begin{pmatrix} A_{n-k} & \cdots & A_{n-1} & A_n \\ \Delta A_{n-k} & \cdots & \Delta A_{n-1} & \Delta A_n \\ \Delta A_{n-k+1} & \cdots & \Delta A_n & \Delta A_{n+1} \\ \cdot & \cdots & \cdot & \cdot \\ \Delta A_{n-1} & \cdots & \Delta A_{k+n} & \Delta A_{n+k-1} \end{pmatrix}}{\mathrm{Det}\begin{pmatrix} 1 & \cdots & 1 & 1 \\ \Delta A_{n-k} & \cdots & \Delta A_{n-1} & \Delta A_n \\ \Delta A_{n-k+1} & \cdots & \Delta A_n & \Delta A_{n+1} \\ \cdot & \cdots & \cdot & \cdot \\ \Delta A_{n-1} & \cdots & \Delta A_{k+n} & \Delta A_{n+k-1} \end{pmatrix}} \tag{14}$$

where $n > k$, $n + k \leq$ Length[S], and $k <$ Length[S]/2. Here Length[S] gives the number of elements in the sequence S.

The Shanks transformation (14) is useful in finding a sum of a slowly convergent sequence using a limited number of partial sums. The best prediction, called Shanks sum, is given when n = Ceiling[Length[S]/2] and $k = n - 1$. Here Ceiling[x] gives the smallest integer greater or equal to x. Our Mathematica rule computing Shanks sum is given in [10].

RESULTS

We select the same numerical values for a comparative convergence behavior used in [2, 11], with M = 0.1, τ = 0.1, and b = 10.

The splitting-up solution (12) is shown in Table 1.

The results for $\theta[\xi, \tau]$ are the same as those given in the "Exact" column of [2, Table 3.1]. The sum in Eq. (5) is the fastest convergent since not more than 3 terms are necessary to obtain 5 correct significant digits.

The excellent convergence behavior is because the splitting-up scheme introduced in [4] gives $f[\xi]$ in closed form. Since the splitting-up approach is applicable only when the source terms of the problem are exponentials or polynomial functions, other alternative filters are proposed in [2].

Table 1. The Solution (5) Using (12) for M = 0.1, τ = 0.1, and b = 10

ξ	$\theta[\xi, \tau]$	n	$f[\xi]$
0	0.36788	0	1
0.2	0.67699	3	0.81805
0.4	0.85531	2	0.31996
0.6	0.94196	3	–0.30180
0.8	0.97670	2	–0.80691
1.0	0.98541	3	–1.00018

In contrast to [2], were $\theta[\xi, \tau]$ is computed, we here directly investigate the filters given by Eqs. (8)–(11).

The following table gives the number of terms used in the expansions to obtain the last column of the Table 1 above.

At ξ = 0 all sums in Eqs. (7)–(11) become zero. Therefore Eq. (7) gives wrong results, $f[0] = 0$, while all other filters give the correct result, $f[0] = 1$.

The best convergence performance is, as expected, that obtained through the splitting-up procedure, Eq. (12), which is not shown in Table 2 since it is represented by an exact formula.

The worst performance is demonstrated by the formal solution, Eq. (7), without any convergence acceleration scheme. Even using 10^5 terms, the last significant digit requested, as in Table 1, is not fully recovered.

All the other convergence enhanced expressions, Eqs. (8)–(11) are acceptable for practical application.

Table 3 gives $f[\xi]$ obtained from Eq. (7) by Shanks transformation formula and the number of partial sums used.

Taking into account the very slow convergence of the solution (7), Shanks summation gives astonishing accurate results.

Table 2. Number of Terms Required in the Filters to Obtain 5 Converged Digits

ξ	Eq. (7)	Eq. (8)	Eq. (9)	Eq. (10)	Eq. (11)
0.2	10^5	116	17	116	17
0.4	10^5	81	13	81	12
0.6	10^5	76	13	76	13
0.8	10^5	51	11	51	12
1.0	10^5	22	11	51	11

Table 3. Shanks Sums of Eq. (7) and the Number of Partial Sums Used

ξ	$f[\xi]$	n
0.2	0.81801	19
0.4	0.31999	15
0.6	–0.30179	13
0.8	–0.80691	11
1.0	–1.00018	9

Table 4. Shanks Sums of Eq. (8) and the Number of Partial Sums Used

ξ	$f[\xi]$	n
0.2	0.81805	19
0.4	0.31996	15
0.6	–0.30180	13
0.8	–0.80691	9
1.0	–1.00018	7

Table 4 gives $f[\xi]$ obtained from Eq. (8) by Shanks transformation formula and the number of partial sums used.

Again Shanks transformation gives accurate results, providing the expected convergence acceleration.

CONCLUSIONS

The splitting-up procedure [4] offers an essentially ideal convergence behavior. It should be remembered though that this approach is not applicable to more involved problems.

The classical integral transform technique can readily solve non-homogeneous problems, but the obtained solutions in series form may have slow convergence. However, the convergence acceleration schemes described in [2] and here reviewed are acceptable for practical applications.

The Shanks transformation recently implemented in *Mathematica* [10] is very promising for the combined use with the generalized integral transform technique (GITT) [1–3], because it is universal and easily applicable to many different problems, whether linear or nonlinear.

REFERENCES

1. Cotta, R. M. (1993). *Integral transforms in computational heat and fluid flow.* Boca Raton, FL: CRC Press.
2. Cotta, R. M., & Mikhailov, M. D. (1997). *Heat conduction—lumped analysis, integral transforms, symbolic computation*. New York: Wiley.
3. Cotta, R. M. (Ed.) (1998). *The integral transform method in thermal and fluids science and engineering*. New York, NY: Begell House.
4. Mikhailov, M. D. (1977). Splitting-up of heat conduction problems. *Lett. Heat & Mass Transfer*, *4*, 163–166.
5. Mikhailov, M. D., & Ozisik, M. N. (1984). *Unified analysis and solutions of heat and mass diffusion*. New York: Wiley.
6. Shanks, D. (1955). Non-linear transformation of divergent and slowly convergent sequences. *J. of Mathematics and Physics*, *34*, 1–42.
7. Lanczos, C. (1967). *Applied analysis*. London: Pitman & Sons.
8. Tolstov, G. P. (1962). *Fourier series*. Englewood Cliffs, NJ: Prentice-Hall.
9. Wolfram, S. (1996). *The mathematica book*, 3rd ed. Wolfram Media/Cambridge University Press.
10. Mikhailov, M. D., & Cotta, R. M. *Integral transform method with mathematica*. (in preparation).
11. Almeida, A. R., & Cotta, R. M. (1996). A comparison of convergence acceleration schemes for eigenfunction expansions of partial differential equations. *Int. J. Num. Meth. Heat & Fluid Flow*, *6*(6), 85–97.

Hybrid Methods in Engineering, vol. 3, pp. 95–109, 2001

A HYBRID SPECTRAL NODAL METHOD FOR ONE-SPEED DISCRETE ORDINATES EIGENVALUE PROBLEMS IN TWO-DIMENSIONAL CARTESIAN GEOMETRY

Hermes Alves Filho & Ricardo C. Barros
Instituto Politécnico—IPRJ, Universidade do Estado do Rio de Janeiro, UERJ, Caixa Postal 97282, 28601-970, Nova Friburgo, RJ, Brazil; halves@iprj.uerj.br, ricardob@iprj.uerj.br

Fernando Carvalho da Silva
Programa de Engenharia Nuclear—COPPE, Universidade Federal do Rio de Janeiro—UFRJ, Caixa Postal 68509, 21 945-970, Rio de Janeiro, RJ, Brazil; fernando@lmn.con.ufrj.br

We describe a hybrid spectral nodal method applied to one-speed S_N eigenvalue problems in X, Y-geometry for nuclear reactor global calculations. To solve the transverse-integrated S_N nodal equations, we generalize the spectral diamond (SD) method that we developed for numerically solving slab-geometry S_N eigenvalue problems with no spatial truncation error. In the present generalization, we approximate the transverse leakage through the edges of each spatial node by constants, so we call our method the SD-constant nodal (SD-CN) method, which we use in the fuel regions of the nuclear reactor core. In the nonmultiplying regions, for example, reflector and baffle, we use the spectral Green's function-constant nodal (SGF-CN) method; hence the hybrid characteristic of our method. To converge the numerical solution for each S_N fixed source problem (inner iterations) in each outer iteration (power method), we use the one-node block inversion (NBI) scheme. We show numerical results for two typical model problems to illustrate the method's accuracy in coarse-mesh calculations and to justify the hybrid characteristic of the numerical algorithm.

KEYWORDS: hybrid method, spectral nodal method, discrete ordinates, neutron transport, nuclear reactor physics

INTRODUCTION

Deterministic computational modeling of physical phenomena generally requires three major steps: (1) We need to choose a mathematical model that usually consists of a set of differential equations supposedly representing approximately the physical phenomenon we want to analyze; (2) we need to choose a convergent and accurate numerical scheme to approximate the mathematical model we have chosen in Step (1); and (3) we need to write an efficient algorithm to solve on a digital computer the discretized equations that we are left with by applying the numerical scheme that we have chosen in Step (2).

This work was developed by the Deterministic Computational Neutronics (DCN) research team, sponsored by CNPq and FAPERJ—Brazil. The authors acknowledge the fruitful help provided by Marcos Pimenta de Abreu (IPRJ/UERJ) José Humberto Zani (IPRJ/UERJ—DCN), Todd Wareing (Los Alamos National Laboratory), and Yousry Azmy (Oak Ridge National Laboratory) during the development of this work.

Before embarking on details associated with the computational modeling described in this article, getting a glimpse of the overall picture would be helpful. Nuclear reactor global calculations (eigenvalue problems) provide the effective multiplication factor (dominant eigenvalue k) and the stationary neutron flux distribution (fundamental mode) at selected times during the lifetime of the core. The effective multiplication factor is required to establish the nuclear composition and configuration that satisfy criticality and control requirements. The steady-state flux distribution must be known to calculate reactor rates and power distributions that are needed for the thermal, mechanical, and shielding design of the nuclear reactor, as well as for evaluating refueling requirements [1].

In this article, we present a deterministic computational modeling of neutron transport for nuclear reactor global calculations. We use the discrete ordinates (S_N) formulation [2] of the steady-state, one-speed, first-order form of the neutron transport equation in X, Y-geometry as the mathematical model (compare with Step 1, noted previously). As with the numerical method (compare with Step 2, noted previously) practical limitations on computer storage and execution time have motivated the development of coarse-mesh methods for S_N problems. Therefore, we describe a hybrid spectral nodal method for coarse-mesh nuclear reactor S_N calculations. To solve the transverse-integrated S_N nodal equations, we generalize the spectral diamond (SD) method that we developed for numerically solving slab-geometry S_N eigenvalue problems with no spatial truncation error [3]. In the present generalization, we approximate the transverse leakage through the edges of each spatial node by constants, so we call our method the SD-constant nodal (SD-CN) method, which we use in the fuel regions of the nuclear reactor core [4]. In the nonmultiplying regions, for example, reflector and baffle, we use the spectral Green's function-constant nodal (SGF-CN) method [5]; hence the hybrid characteristic of the present SD-SGF-CN method. To converge the numerical solution for each S_N "fixed source" problem (inner iterations) in each outer iteration (power method), we use the one-node block inversion (NBI) iterative scheme (compare with Step 3, noted previously) [5].

At this point, we present an outline of the remainder of this article. In the next section we present the mathematical model. Furthermore, we describe the SD-CN method that we use in the multiplying regions of the domain and the SGF-CN method that we use in the nonmultiplying regions of the domain to build the present hybrid spectral nodal method. Then we describe the NBI iterative scheme and, in conclusion, present some numerical results for two typical problems with a brief discussion.

MATHEMATICAL MODEL

Let us consider the S_N equations in a rectangular domain D of width X and height Y with isotropic scattering

$$\mu_m \frac{\partial}{\partial x}\psi_m(x,y) + \eta_m \frac{\partial}{\partial y}\psi_m(x,y) + \sigma_T(x,y)\psi_m(x,y)$$

$$= [\sigma_{S0}(x,y) + \frac{\nu}{k}\sigma_F(x,y)]\sum_{n=1}^{M}\psi_n(x,y)\omega_n \qquad (1)$$

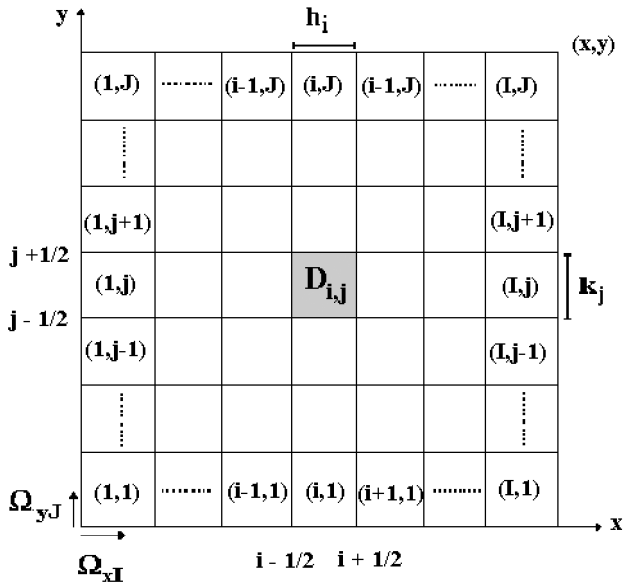

Fig. 1 Spatial grid $\Omega_{xI} \times \Omega_{yJ}$.

where

$$(x, y) \in D \qquad m = 1{:}M \qquad \text{with} \qquad M = N(N+2)/2$$

N order of the angular quadrature set
ψ_m angular flux
σ_T total macroscopic cross section
σ_{S0} zero-th order component of the differential scattering cross section
σ_f fission macroscopic cross section
k eigenvalue
ν average number of neutrons emitted in each fission event

At this point, let us consider an arbitrary spatial grid $\Omega_{xI} \times \Omega_{yJ}$ on D, where each spatial cell is termed node $D_{i,j}$ of width h_i ($i = 1{:}I$) and height k_j ($j = 1{:}J$), as shown in Fig. 1. Each node $D_{i,j}$ has constant cross sections $\sigma_{Ti,j}$, $\sigma_{S0i,j}$, $\sigma_{Fi,j}$.

To obtain the "one-dimensional" transverse-integrated S_N nodal equations, we follow the standard procedure [6]. By applying the transverse integration operator

$$\frac{1}{k_j} \int_{y_{j-1/2}}^{y_{j+1/2}} (\circ) dy$$

to Eq. (1), we obtain the "one-dimensional" transverse-integrated S_N nodal equation for the x direction

$$\mu_m \frac{d}{dx}\tilde{\psi}_{m,j}(x)+\sigma_{Ti,j}\tilde{\psi}_{m,j}(x)=\tilde{S}_{m,j}(x)-\frac{\eta_m}{k_j}[\psi_m(x,y_{j+1/2})-\psi_m(x,y_{j-1/2})] \quad (2)$$

Here we have defined the zero-th order y-moment of the angular flux

$$\tilde{\psi}_{m,j}(x)\equiv\frac{1}{k_j}\int_{y_{j-1/2}}^{y_{j+1/2}}\psi_m(x,y)dy \quad (3)$$

and similarly, the zero-th order y-moment of the source

$$\tilde{S}_{m,j}(x)\equiv[\sigma_{S0i,j}+\frac{\nu}{k}\sigma_{Fi,j}]\sum_{n=1}^{M}\tilde{\psi}_{n,j}(x)\omega_n \quad (4)$$

Now we apply the transverse integration operator

$$\frac{1}{h_i}\int_{x_{i-1/2}}^{x_{i+1/2}}(\circ)dx$$

to Eq. (1), and we obtain the "one-dimensional" transverse-integrated S_N nodal equation for the y direction

$$\eta_m \frac{d}{dy}\hat{\psi}_{m,i}(y)+\sigma_{Ti,j}\hat{\psi}_{m,i}(y)=\hat{S}_{m,i}(y)+Q_{i,j}-\frac{\mu_m}{h_i}[\psi_m(x_{i+1/2},y)-\psi_m(x_{i-1/2},y)] \quad (5)$$

where we have defined the zero-th order x-moment of the angular flux

$$\hat{\psi}_{m,i}(y)\equiv\frac{1}{h_i}\int_{x_{i-1/2}}^{x_{i+1/2}}\psi_m(x,y)dx \quad (6)$$

and the zero-th order x-moment of the source

$$\hat{S}_{m,i}(y) \equiv [\sigma_{S0i,j} + \frac{\nu}{k}\sigma_{Fi,j}]\sum_{n=1}^{M} \hat{\psi}_{n,i}(y)\omega_n \tag{7}$$

Equations (2) and (5) are exact; thus far, we have made no approximations. However, for each spatial node $D_{i,j}$ and together with the node boundary conditions, that is, using estimations or prescribed values for the node incident quantities

$$\begin{cases} \tilde{\psi}_{m,j}(x_{i+1/2}) \equiv \tilde{\psi}_{m,i+1/2,j} & \mu_m < 0 \\ \tilde{\psi}_{m,j}(x_{i-1/2}) \equiv \tilde{\psi}_{m,i-1/2,j} & \mu_m > 0 \end{cases} \tag{8}$$

and

$$\begin{cases} \hat{\psi}_{m,i}(y_{j+1/2}) \equiv \hat{\psi}_{m,i,j+1/2} & \eta_m < 0 \\ \hat{\psi}_{m,i}(y_{j-1/2}) \equiv \hat{\psi}_{m,i,j-1/2} & \eta_m > 0 \end{cases} \tag{9}$$

Eqs. (2) and (5) form a system of $2M$ ordinary differential equations in $6M$ unknown quantities. Therefore, to obtain a unique solution, besides the node boundary conditions, given by Eqs. (8) and (9), we need to consider approximations that relate the angular flux $\psi_m(x, y)$ to its zero-th order x- and y-moments in the transverse leakage terms. In this article we consider constant approximations for the transverse leakage terms. That is

$$\frac{\eta_m}{k_j}[\psi_m(x, y_{j+1/2}) - \psi_m(x, y_{j-1/2})] \approx \frac{\eta_m}{k_j}(\hat{\psi}_{m,i,j+1/2} - \hat{\psi}_{m,i,j-1/2}) \equiv \hat{L}_{m,i,j} \tag{10}$$

and

$$\frac{\mu_m}{h_i}[\psi_m(x_{i+1/2}, y) - \psi_m(x_{i-1/2}, y)] \approx \frac{\mu_m}{h_i}(\tilde{\psi}_{m,i+1/2,j} - \tilde{\psi}_{m,i-1/2,j}) \equiv \tilde{L}_{m,i,j} \tag{11}$$

The general solutions of Eqs. (2) and (5) in $D_{i,j}$ with flat approximations for the transverse leakage terms are given by

$$\tilde{\psi}_{m,j}(x) = \tilde{\psi}^{p}_{m,j}(x) + \tilde{\psi}^{h}_{m,j}(x) \qquad x \in D_{i,j} \tag{12}$$

and

$$\hat{\psi}_{m,i}(y) = \hat{\psi}^{p}_{m,i}(y) + \hat{\psi}^{h}_{m,i}(y) \qquad y \in D_{i,j} \tag{13}$$

Here the superscript p indicates the particular solution that depends upon the approximations we make in the transverse leakage terms. The superscript h indicates the homogeneous component of the solution, which satisfies the homogenous equation associated with Eqs. (2) and (5). By substituting Eq. (10) into Eq. (2), we seek a spatially constant particular solution to the resulting equation. Therefore, we obtain

$$\tilde{\psi}^{p}_{m,i,j} = -\frac{c_{0i,j}}{\sigma_{Ti,j}(1-c_{0i,j})}\sum_{n=1}^{M}\hat{L}_{n,i,j}\omega_n - \frac{\hat{L}_{m,i,j}}{\sigma_{Ti,j}} \tag{14}$$

where

$$c_{0i,j} = \frac{\sigma_{0i,j} + \frac{\nu}{k}\sigma_{Fi,j}}{\sigma_{Ti,j}} \qquad m = 1:M$$

A similar expression is obtained for the particular solution of Eq. (5). To obtain the homogeneous components of the general solutions, we perform a spectral analysis of the homogeneous equations associated with Eqs. (2) and (5). This spectral analysis is described in detail in Ref. [7]. Combining the particular solution (14) with the homogeneous solution that we obtain from the spectral analysis

$$\tilde{\psi}^{h}_{m,j}(x) = \sum_{k=1}^{M}\beta_k\tilde{\psi}^{h}_{m,j}(x,\vartheta_k) = \sum_{k=1}^{M}\beta_k a^{x}_{m}(\vartheta_k)\exp(-\sigma_{Ti,j}x/\vartheta_k) \qquad m = 1:M \tag{15}$$

where β_k, $k = 1{:}M$ are arbitrary constants, we use Eq. (12) to write down the general solution of Eq. (2). To find the general solution in the y variable, given by Eq. (13), we proceed similarly.

THE HYBRID NUMERICAL SCHEME

Let us integrate the one-speed X, Y-geometry S_N equations with isotropic scattering (1) over an arbitrary node $D_{i,j}$. The result is the familiar spatial balance equation

$$\frac{\mu_m}{h_i}(\tilde{\psi}_{m,i+1/2,j} - \tilde{\psi}_{m,i-1/2,j}) + \frac{\eta_m}{k_j}(\hat{\psi}_{m,i,j+1/2} - \hat{\psi}_{m,i,j-1/2}) + \sigma_{Ti,j}\overline{\psi}_{m,i,j}$$
$$= (\sigma_{soi,j} + \frac{\nu\sigma_{fi,j}}{k})\sum_{n=1}^{M}\overline{\psi}_{n,i,j}\omega_n \qquad m = 1:M \tag{16}$$

where the node-average angular flux is defined as

$$\bar{\Psi}_{m,i,j} \equiv \frac{1}{h_i k_j} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \psi_m(x, y) dx dy \tag{17}$$

The balance equation (16) combined with appropriate continuity conditions on the node edges and boundary conditions imposed on the outer boundaries of the domain form an underdetermined system; therefore we need auxiliary equations to obtain the same number of equations as unknowns. In the nonmultiplying regions of the domain, for example, reflector and baffle, we use the SGF auxiliary equations [5]. In the fuel regions, we use the nonstandard SD auxiliary equations. Further, we describe these two types of auxiliary equations.

The SGF Auxiliary Equation

In the nonmultiplying regions of the domain, for example, reflector and baffle, we use the SGF auxiliary equation, which we define as

$$\bar{\Psi}_{m,i,j} = \sum_{\mu_n > 0} \theta_{m,n} \tilde{\Psi}_{n,i-1/2,j} + \sum_{\mu_n < 0} \theta_{m,n} \tilde{\Psi}_{n,i+1/2,j} + \hat{G}_{m,i,j} \tag{18a}$$

and

$$\bar{\Psi}_{m,i,j} = \sum_{\eta_n > 0} \gamma_{m,n} \hat{\Psi}_{n,i,j_{-1/2}} + \sum_{\eta_n < 0} \gamma_{m,n} \hat{\Psi}_{n,i,j_{+1/2}} + \tilde{G}_{m,i,j} \tag{18b}$$

where $m = 1{:}M$. In Eq. (18a), $\theta_{m,n}$ and $\hat{G}_{m,i,j}$ are to be determined by requiring that the general solution of Eq. (2) with flat leakage approximation, given by Eq. (12) combined with Eqs. (14) and (15), where $\sigma_{Fi,j} = 0$, have node-average and node-edge average angular fluxes that, for all values of β_k, satisfy Eq. (18a). Following a similar procedure, we obtain $\gamma_{m,n}$ and $\tilde{G}_{m,i,j}$. We note that for nonmultiplying regions, $\sigma_{Fi,j} = 0$, and hence $c_{o,i,j} < 1$, namely, Eq. (14); therefore, the M^2 values of $\theta_{m,n}(\gamma_{m,n})$ in Eqs. (18a,18b) are fixed and depend only upon the cross sections and width h_i (height k_j) of node $D_{i,j}$. This good feature of the SGF auxiliary Eqs. (18a, 18b) can be used to determine S_N albedo matrices to substitute approximately the nonmultiplying media around nuclear reactor cores in global calculations. The implementation of this idea must, however, await future work.

The SD Auxiliary Equation

In the fuel regions of the domain, we use the SD auxiliary equations, which we write as

$$\bar{\psi}_{m,i,j} = \frac{1}{2}\sum_{n=1}^{M} \theta_{m,n}(\tilde{\psi}_{n,i-1/2,j} + \tilde{\psi}_{n,i+1/2,j}) + \hat{G}_{m,i,j} \tag{19a}$$

and

$$\bar{\psi}_{m,i,j} = \frac{1}{2}\sum_{n=1}^{M} \gamma_{m,n}(\hat{\psi}_{n,i,j-1/2} + \hat{\psi}_{n,i,j+1/2}) + \tilde{G}_{m,i,j} \tag{19b}$$

where $m = 1{:}M$. Here the quantities $\theta_{m,n}$, $\gamma_{m,n}$ and the expressions for $\tilde{G}_{m,i,j}$ and $\hat{G}_{m,i,j}$ are to be determined by following similar steps as for the SGF auxiliary equations. It is important to point out, however, that for each estimate of the dominant eigenvalue k in the outer iterations, we need to calculate the M^2 quantities $\theta_{m,n}$ and $\gamma_{m,n}$ by solving linear systems that we obtain for the cases $c_{o,i,j} < 1$ or $c_{o,i,j} > 1$.

At this point we remark that, unlike the SGF auxiliary equations (18a, 18b), which relate the node-average angular flux in a given direction to the node-edge average angular fluxes in the incoming directions, the SD auxiliary equations (19a, 19b) relate the node-average angular flux in a given direction to the node-edge average angular fluxes in all directions, including the exiting directions. This characteristic of the SD auxiliary equations is convenient for active zones because it indicates that an interior neutron source (due to fission) exists inside node $D_{i,j}$.

ITERATIVE ALGORITHM

The power method is traditionally used to converge the dominant solution of S_N eigenvalue problems [8]. For each estimate of the dominant eigenvalue k in the outer iterations, we solve a "fixed-source" S_N problem by using inner iterations. The one-node block inversion (NBI) scheme iterates on the node-edge average angular fluxes by performing node-block inversions. That is, the NBI scheme uses the best estimates available for the incoming node-edge average angular fluxes to evaluate the exiting fluxes, which constitute the incoming fluxes for the adjacent nodes in the directions of the transport sweeps. To illustrate this concept, let us consider Fig. 2, representing an arbitrary spatial node $D_{i,j}$ with the node-edge average angular fluxes. Each arrow in Fig. 2 represents $N(N+2)/8$ directions in each quadrant. For the sweep indicated (from SW to NE), the outward arrows (thicker arrows) in the north and east of $D_{i,j}$ represent the outgoing node-edge average angular fluxes that we need to calculate, because they form the incoming node-edge average angular fluxes for the adjacent nodes in this sweep. The inward arrows (medium arrows) represent the incoming node-edge average angular fluxes, which are known, or at least we use the best estimates available for them. Finally, the outward arrows (thinner arrows) in the south and west of $D_{i,j}$ represent the node-edge average angular fluxes, which can be calculated but which are not needed for this sweep. They represent the outgoing quantities that are needed for the NE to SW sweep. An analogous convention is followed for the NW to SE and SE to NW transport sweeps. As we see, we need to iterate only on the node-edge average angular fluxes. Therefore, unlike the

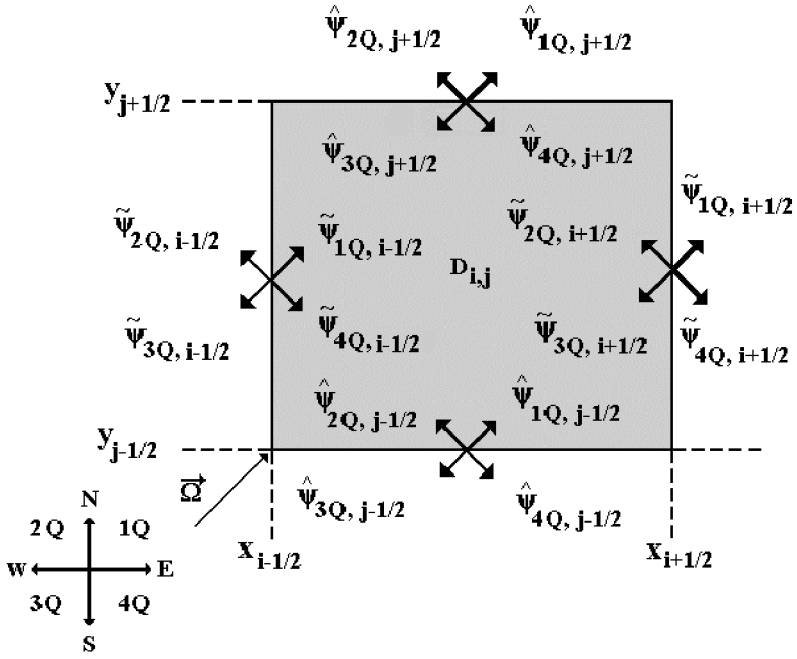

Fig. 2 SW to NE transport sweep.

conventional source iteration (SI) scheme [2], the NBI scheme needs to store the node-edge average fluxes in all directions. For coarse-mesh calculations, however, this storage requirement is very much alleviated, apart from the fact that the convergence rate of the NBI scheme is higher than the convergence rate of the SI scheme for coarse spatial grids. We are now working on different versions of the NBI iterative scheme, looking forward to parallelizing the algorithm and to simplifying the numerical scheme of the SGF-LN method, which we plan to develop soon [11].

NUMERICAL RESULTS AND DISCUSSION

In this section, we perform two numerical experiments to illustrate the accuracy of the present hybrid SD-SGF-CN method for coarse-mesh eigenvalue S_N calculations. Figure 3 shows 1/4 of a two-dimensional x, y nuclear reactor core cut perpendicular to its z-axis; its material zones have the one-energy group data listed in Table 1.

Table 2 shows the dominant eigenvalue (effective multiplication factor, k_{eff}) generated by the conventional constant-constant nodal (CCN) method [9], the conventional linear nodal (LN) method [6] and the present SD-SGF-CN method with the S_4 level symmetric angular quadrature set [2] on various spatial grids. The reference result has been generated by the traditional diamond difference (DD) method [2] on a very fine grid, composed of 64 spatial cells per region in each spatial direction (x and y).

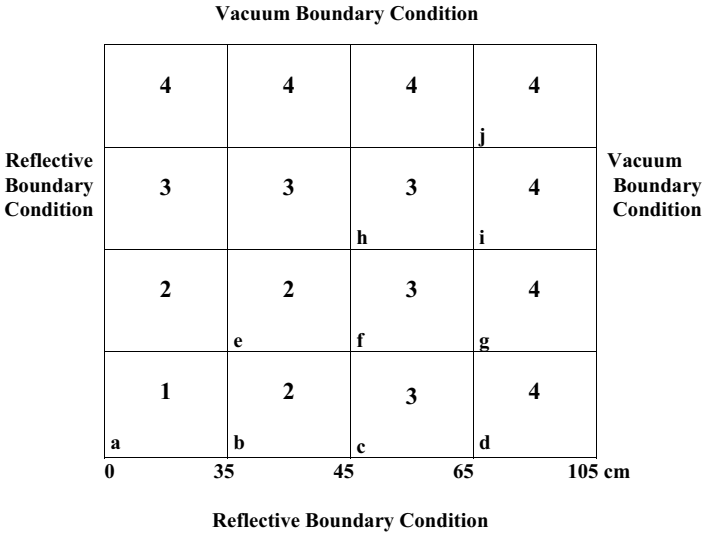

Fig. 3 Model Problem No. 1 (1/8—symmetric domain).

As we see, the SD-SGF-CN method is more accurate than the CCN method, although in both methods, we consider flat approximations for the transverse leakage terms and we deal only with zero-th order spatial moments of the transverse-integrated S_N equations. On the other hand, the SD-SGF-CN method generated numerical results for the effective multiplication factor that were about as accurate as the LN results, except for very coarse grids, in which case the SD-SGF-CN method generated more accurate results, although in the LN method we consider linear approximations for the transverse leakage terms and deal with zero-th and first order spatial moments of the transverse-integrated S_N equations. At this point, we remark that the companion nonhybrid SGF-CN method showed numerical instability for the coarsest spatial grid listed in Table 2 and simply did not converge the problem. Furthermore, Fig. 4 shows the relative deviations (%) of the average power density distribution within the domain represented in Fig. 3, as generated by the CCN, the LN and the SD-SGF-CN methods with respect to the fine-grid reference results generated by the DD method.

According to Fig. 4 (spatial grid composed of 1 node per region), the maximum deviation turned out to be 34.58% in region h, as generated by the CCN method. On the

Table 1. Nuclear Data for Model Problem No. 1

Zone number	σ_T	σ_{s0}	$\nu\sigma_F$
1	2.22589E-1[a]	2.20563E-1	2.83283E-3
2	2.16566E-1	2.10697E-1	1.04347E-2
3	3.01439E-1	2.96069E-1	5.13036E-4
4	2.52250E-1	2.50794E-1	0.0

[a]Read as $2.22589 \cdot 10^{-1}$.

Table 2. Numerical Results for Model Problem No. 1

Spatial grid Γ_n[a]	Numerical method	(k_{eff})	Relative deviation (%)[g]	Total number of transport sweeps
Γ_2	CCN[b]	0.62134357	35.42	247
	LN[c]	0.93844318	2.46	998
	SGF-CN[d]	————	————	————
	SD-SGF-CN[e]	0.96865121	0.67	607
Γ_3	CCN	0.77330886	19.63	510
	LN	0.95868309	0.37	1010
	SGF-CN	0.96395583	0.18	1002
	SD-SGF-CN	0.96395579	0.18	1028
Γ_4	CCN	0.88168998	8.37	1025
	LN	0.96182392	0.04	1007
	SGF-CN	0.96273336	0.06	1605
	SD-SGF-CN	0.96273334	0.06	1662
Γ_5	CCN	0.93553854	2.77	1801
	LN	0.96217276	0.00	1013
	SGF-CN	0.96235350	0.02	2272
	SD-SGF-CN	0.96235346	0.02	2336
Γ_6	CCN	0.95479778	0.77	2653
	LN	0.96220192	0.00	1016
	SGF-CN	0.96224373	0.00	2922
	SD-SGF-CN	0.96224372	0.00	2940
Γ_7	CCN	0.96029527	0.19	9666
	LN	0.96220398	0.00	1019
	SGF-CN	0.96221464	0.00	3433
	SD-SGF-CN	0.96221464	0.00	3442
Γ_8	DD[f]	0.96220092		

[a] $2^n/4$ spatial nodes per region in each spatial direction.
[b] Constant-constant nodal method.
[c] Linear nodal method.
[d] Spectral Green's function-constant nodal method.
[e] Spectral diamond-spectral Green's function-constant nodal method.
[f] Diamond difference method.
[g] Relative deviation with respect to the DD fine-mesh solution.

same spatial grid, the maximum deviation generated by the LN method was 11.54% in region h and the maximum deviation generated by the hybrid SD-SGF-CN method was 8.61% in region e. As with the efficiency of the proposed SD-SGF-CN method for coarse-mesh calculations, let us consider the following gedankenexperiment. In performing nuclear reactor global calculations, let us suppose that the user considers that relative deviations in the effective multiplication factor greater than 5%, with respect to the reference result, are not acceptable. In this case, according to Table 2, the CCN method satisfies the user's demand on a spatial grid composed of 8×8 nodes per region, whereas the SD-SGF-CN method satisfies the same condition on a spatial grid composed of 1×1 node per region. Therefore, considering that we ran both methods on the same machine with the NBI scheme in the inner iterations, it is fair to compare the CPU execution times. The CCN code generated the dominant eigenvalue within 2.77% in 3484 seconds. In contrast, the SD-SGF-CN code generated the dominant eigenvalue within 0.67% in

			j
	***CCN **LN *SD-SGF-CN	*** 34.58 ** 11.54 * 3.27 **h**	**i**
	8.80 3.98 8.61 **e**	23.19 10.59 3.88 **f**	**g**
4.22 0.67 0.03 **a**	0.74 0.38 0.92 **b**	12.91 5.47 0.54 **c**	**d**

Fig. 4 Relative deviations (%) in the average power density distribution generated on a spatial grid composed of 1 node per region for model problem No. 1.

43 seconds. Because the LN code has been implemented with a different iterative scheme, that is, the SI scheme, we do not think it is fair to compare execution time of the LN code with the execution times of the CCN and SD-SGF-CN codes.

Our second model problem consists of a 60×60 cm heterogeneous domain, composed of three different material zones, namely, Fig. 5. Table 3 gives the nuclear data for the three zones: two active zones (Pu^{239}) and one reflector (H_2O) around the core [12].

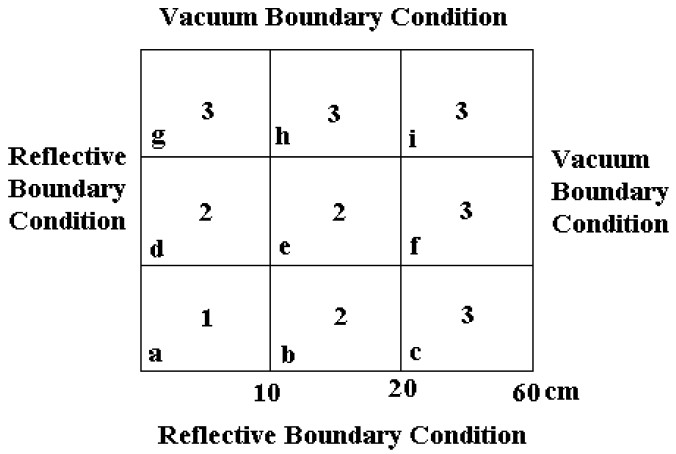

Fig. 5 Model problem No. 2 (critical system, $k_{\text{eff}} = 1.0$).

Table 3. Nuclear Data for Model Problem No. 2

Zone number	σ_T	σ_{s0}	$\nu\sigma_F$[b]
1	3.26400E-1[a]	2.25216E-1	0.11491
2	3.26400E-1	2.25216E-1	0.10070
3	3.26400E-1	2.93760E-1	0.0

[a] $3.26400 \cdot 10^{-1}$.
[b] The numerical values given in Ref. [12] have been divided by 2.3008 to obtain a critical system ($k_{\text{eff}} = 1.0$).

Table 4 lists the dominant eigenvalue (k_{eff}) generated by both the nonhybrid SGF-CN method, for those cases for which it converged, and the hybrid SD-SGF-CN method with the S_4 level symmetric angular quadrature set on various spatial grids. As with the first model problem, the reference result has been generated by the traditional DD method on a spatial grid composed of 64 spatial cells per region in each spatial direction.

Again, the nonhybrid SGF-CN method showed some numerical instability and did not converge for the spatial grids composed of 1×1 and 8×8 nodes per region. However, upon convergence, the SGF-CN and the hybrid SD-SGF-CN methods generated the same results within acceptable deviations. We note that the hybrid SD-SGF-CN method did not show numerical instability and hence converged model problems No. 1 and No. 2 on *all* the spatial grids considered.

Moreover, we consider the following gedankenexperiment. In performing criticality calculations, let us suppose that the research team demands that the relative deviations in the average power density distribution cannot exceed 2% with respect to the reference results. Figure 6 shows that the critical system represented in Figure 5 has maximum relative deviation in the average power density equal to 1.82% in region b with respect to the reference fine-grid DD results. Therefore, the research team's demand is fulfilled by using a spatial grid composed of only 1 node per region in the proposed hybrid SD-SGF-CN method.

Before closing, we list a number of general conclusions and suggestions for future work: (1) the hybrid characteristic of the SD-SGF-CN method seems to be necessary, because the companion nonhybrid SGF-CN method showed numerical instability in coarse-mesh calculations. This drawback of the SGF class of methods for S_N eigenvalue problems had already been reported in Ref. [3] for slab geometry calculations. On the other hand, using the SGF-CN method in the nonmultiplying regions of the domain is convenient because the SGF auxiliary equations are much simpler than the SD auxiliary equations for the calculations of S_N albedo matrices—a topic that we intend to work on in near future; (2) the SD-SGF-CN method, as described in this article, is much more accurate than the conventional CCN method for coarse-mesh calculations, although in both methods we consider flat leakage approximations; (3) the SD-SGF-CN method generated numerical results for the first test problem that were, roughly speaking, as accurate as the LN results. However, this good feature seems to be problem dependent, and for future work, we suggest developing the companion SD-SGF-LN method and comparing it more fairly with the LN method; (4) we also suggest extending the present one-speed SD-SGF-CN method to multigroup S_N problems; this should be quite straightforward [3]. Moreover, the extension to three-dimensional S_N problems with linearly anisotropic scattering should not create any substantial theoretical difficulties.

Table 4. Numerical Results for Model Problem No. 2

Spatial grid Γ_n[a]	Numerical method	(k_{eff})[e]	Relative deviation (%)[g]
Γ_2	SGF-CN[b]	———— (—)	———— (—)
	SD-SGF-CN[c]	1.00382542 (150)[f]	0.39
Γ_3	SGF-CN	1.00142263 (970)	0.15
	SD-SGF-CN	1.00142255 (245)	0.15
Γ_4	SGF-CN	1.00048251 (255)	0.05
	SD-SGF-CN	1.00048253 (285)	0.05
Γ_5	SGF-CN	———— (—)	———— (—)
	SD-SGF-CN	1.00015363 (270)	0.02
Γ_6	SGF-CN	1.00005179 (270)	0.01
	SD-SGF-CN	1.00005177 (270)	0.01
Γ_7	SGF-CN	1.00002322 (275)	0.00
	SD-SGF-CN	1.00002322 (275)	0.00
Γ_8	DD[d]	0.99994868	

[a] $2^n/4$ spatial nodes per region in each spatial direction.
[b] Spectral Green's function-constant nodal method.
[c] Spectral diamond-spectral Green's function-constant nodal method.
[d] Diamond difference method.
[e] Effective multiplication factor with fixed number of inner iterations (5) in each outer iteration.
[f] Total transport sweeps.
[g] Relative deviation with respect to the DD fine-mesh solution.

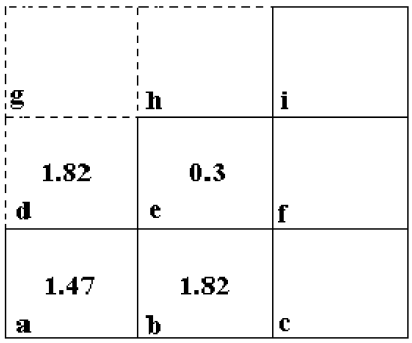

Fig. 6 Relative deviation (%) in the average power density distribution (W/cm^3) generated on a spatial grid composed of 1 node per region for model problem No. 2.

REFERENCES

1. Weisman, J. (1977). *Elements of nuclear reactor design.* Elsevier.
2. Lewis, E. E., & Miller, W. F. Jr. (1984). *Computational methods of neutron transport.* New York: Wiley.
3. de Abreu, M. P., Alves Filho, H. & Barros, R. C. (1996). *Transport Theory and Statistical Physics*, *25*(1), 61.
4. Alves Filho, H., da Silva, F. C., & Barros, R. C. (1998). *Int. Conf. on the Physics of Nuc. Sci. and Tech.*, Long Island, NY, Proceedings Vol. 1, 515.
5. Barros, R. C., & Larsen, E. W. (1992). *Nuclear Science and Engineering*, *111*, 34.
6. Walters, W. F. (1986). *Progress in Nuclear Energy*, *18*, 21.
7. Barros, R. C., da Silva, F. C., & Alves Filho, H. (1999). *Progress in Nuclear Energy, 35*(3–4), 293.
8. Fladmark, G. E. (1972). IAEA-numerical reactor calculations. Vienna, Jan. 17–21. Proc. IAEA-SM-154/20.
9. Walters, W. F., & O'Dell, R. D. (1981). *ANS/ENS Int. Top. Meeting on Adv. in Math. Meth. for Solution of Nuc. Eng. Prob.*, *1*, 114.
10. Barros, R. C., da Silva, F. C., & Alves Filho, H. (1998). *Progress in Nuclear Energy*, *33*(1–2), 117.
11. Zani, J. H. (1999). *Personal communication*.
12. Sood, A., Forster, R. A., & Parsons, D. K. (1999). Analytical benchmark test set for criticality code verification, LA-13511.