



HAL
open science

NUMERICAL SOLUTIONS OF LINEAR MIXED BOUNDARY VALUE PROBLEMS USING STOCHASTIC REPRESENTATIONS

Jean-Paul Morillon

► **To cite this version:**

Jean-Paul Morillon. NUMERICAL SOLUTIONS OF LINEAR MIXED BOUNDARY VALUE PROBLEMS USING STOCHASTIC REPRESENTATIONS. *International Journal for Numerical Methods in Engineering*, 1997, 40 (3), pp.387-405. 10.1002/(SICI)1097-0207(19970215)40:33.0.CO;2-D . hal-00967554

HAL Id: hal-00967554

<https://hal.science/hal-00967554>

Submitted on 29 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Numerical solutions of linear mixed boundary value problems using stochastic representations

Jean-Paul MORILLON

Email: Jean-Paul.Morillon@univ-reunion.fr

(Revised manuscript)

SUMMARY

The application of It's formula induces some probabilistic representations of solutions of deterministic linear problems with boundary conditions of Dirichlet, Neumann, Fourier, and mixed types. These representations are used to establish some easily implementable algorithms which compute an approximate solution by means of simulation of reflected random walks. They do not require selected configurations at the neighborhood of the domain boundary, nor a discretization mesh. The associated simulation methods are obtained and applied to problems for each class of boundary conditions. Numerical experiments with distributed source in two- or three-dimensional geometries, and computational results with estimation of error, are reported.

1. INTRODUCTION

Numerical solution of boundary value problems can be performed by the usual finite difference or finite element methods and adequate space mesh. Monte Carlo methods can also be relevant; they lead, after equations discretization, to specific treatments related to domain local geometry (see, for instance, References 1–4).

This work is also investigating another method in applying Markov process theory which provides integral representations of the solution to stationary deterministic linear boundary value problems. In particular, the Dirichlet problem is known to have an integral representation⁵ which is used to obtain an algorithm which computes the solution by implementing a random walk simulation.⁶

More specifically, a stationary deterministic linear boundary value problem is to be solved with boundary condition of Neumann, Fourier or mixed types. We show, in particular, using examples, that the boundary condition treatments can be reduced to the counting of absorptions and reflections on the boundaries.

Section 2 is devoted to the stochastic representations of solutions with boundary conditions of different types. In Section 3, we present the approaches of these representations by the realizations of random processes, and we establish the corresponding computational algorithms. In Section 4, numerical simulations of random walks provide results associated with two- or three-dimensional geometries. These methods are discussed in the last section.

2. REPRESENTATION OF SOLUTIONS

This section is devoted to the stochastic representations of solutions to linear boundary value problems by means of stochastic integrals. The Dirichlet, Neumann, Fourier, and mixed boundary conditions are presented.

Let us consider an open bounded region G in \mathbb{R}^d (the dimension d being fixed). Let us denote $x \in \mathbb{R}^d$ the space variable and n the unit inward normal defined on the boundary ∂G .

We will show how, from stochastic differential equations (SDE) (for these SDE, see for instance References 7–8, and references therein), we can represent solutions to some boundary value problems. We will use the expectation of integrals, i.e. functionals of trajectories and random variables which are solutions of SDE. The application of It's formula associated with SDE leads to a representation of solutions.

From a functional point of view, the obtained representations have only been established according to regularity hypotheses on geometry of domain G , and on partial differential equations (cf. references mentioned in the sequel of this section). Nevertheless, we are using here the stochastic representations even when data are not necessarily smooth: for instance, the boundary ∂G of domain can show corners or edges (this procedure previously proved to be effective in References 1, 2 and 4 for the discretized equations and in Reference 6 for the Dirichlet problem).

2.1. Dirichlet problem

Let us consider the following Dirichlet problem:

$$\begin{cases} -\frac{1}{2} \Delta u = f & \text{in } G, \\ u = g_1 & \text{on } \partial G, \end{cases} \quad (1)$$

where the function u , defined from G into \mathbb{R} , is to be determined,

and the data are the functions:

$$\begin{aligned} f &: G \longrightarrow \mathbb{R}, \\ g_1 &: \partial G \longrightarrow \mathbb{R}. \end{aligned}$$

Let us introduce the Markov process:

$$X_t^x = x + W_t, \quad t \geq 0,$$

where W_t is the standard Wiener process with values in \mathbb{R}^d so that: $X_0^x = x$.

Let \overline{G} be the complementary set of G in \mathbb{R}^d . The hitting time τ of \overline{G} is defined by:

$$\tau = \inf \{t > 0 \mid X_t^x \in \overline{G}\}.$$

The application of It's formula to $u(X_t^x)$ leads to the following representation of the solution to (1) in the form:

$$u(x) = E[Y], \quad x \in G, \quad (2)$$

with the random variable:

$$Y = \int_0^\tau f(X_t^x) dt + g_1(X_\tau^x). \quad (3)$$

Let us mention that functional results to this problem and its associated representation can be found, for instance, in References 5, 9–10.

2.2. Neumann problem

Let us consider the Neumann problem, using similar notation:

$$\begin{cases} -\frac{1}{2} \Delta u = f & \text{in } G, \\ \frac{\partial u}{\partial n} = g_2 & \text{on } \partial G, \end{cases} \quad (4)$$

The solutions to (4) are defined up to an additive constant. Moreover, the application of divergence theorem leads to the following compatibility condition on the data:

$$2 \int_G f(y) dy - \int_{\partial G} g_2(\gamma) d\gamma = 0.$$

Let us introduce the Markov process defined in G by the Laplace operator within the domain, or by reflection on the boundary in the direction of the inward normal:

$$X_t^x = x + W_t + \int_0^t \mathbb{1}_{\partial G}(X_s^x) n(X_s^x) d\xi_s^x, \quad t \geq 0, \quad (5)$$

where $\mathbb{1}_{\partial G}$ is the indicator of ∂G , and ξ_t^x is a non-decreasing process which increases only when t hits the set $\{s > 0 \mid X_s^x \in \overline{G}\}$, so that: $\xi_0^x = 0$.

The application of It's formula to $u(X_t^x)$ leads to the representation of the solution to (4) as follows:

$$u(x) = E[Y], \quad x \in G, \quad (6)$$

with the following random variable:

$$Y = \int_0^{+\infty} f(X_t^x) dt - \int_0^{+\infty} g_2(X_t^x) \mathbb{1}_{\partial G}(X_t^x) d\xi_t^x + C, \quad (7)$$

where C is an arbitrary additive constant.

Some functional results regarding SDE (5) with reflection on ∂G and Neumann problem (4), are given respectively in References 11 and 5.

Let us mention the modified Neumann problem:

$$\begin{cases} -\frac{1}{2} \Delta u + \lambda u = f & \text{in } G, \\ \frac{\partial u}{\partial n} = g_2 & \text{on } \partial G, \end{cases} \quad (8)$$

where the function:

$$\lambda : G \longrightarrow \mathbb{R}$$

is given with $\lambda \geq \lambda_0 > 0$ ($\lambda_0 \in \mathbb{R}$).

The application of It's formula to:

$$u(X_t^x) \exp\left(-\int_0^t \lambda(X_s^x) ds\right)$$

leads to the following representation of the solution to (8):

$$\begin{aligned} u(x) &= E\left[\int_0^{+\infty} f(X_t^x) \exp\left(-\int_0^t \lambda(X_s^x) ds\right) dt\right] \\ &\quad - E\left[\int_0^{+\infty} g_2(X_t^x) \exp\left(-\int_0^t \lambda(X_s^x) ds\right) \mathbb{1}_{\partial G}(X_t^x) d\xi_t^x\right]. \end{aligned}$$

Concerning problems close to (8), some functional results can be found in References 5, 2 and 12. For instance, in this latter work, λ is a real number, and the solution to problem (8) approximates to a solution of (4) when λ tends to zero.

2.3. Problem with Fourier boundary condition

Let us consider the following problem with Fourier boundary condition, using analogous notation:

$$\begin{cases} -\frac{1}{2} \Delta u = f & \text{in } G, \\ \frac{\partial u}{\partial n} - \mu u = g_3 & \text{on } \partial G, \end{cases} \quad (9)$$

where μ is a strictly positive constant.

Let us introduce the Markov process solution of SDE (5). The application of It's formula to:

$$u(X_t^x) \exp\left(-\mu \int_0^t \mathbb{1}_{\partial G}(X_s^x) d\xi_s^x\right)$$

leads to the representation of the solution to (9) in the form:

$$u(x) = E[Y], \quad x \in G, \quad (10)$$

with the random variable:

$$\begin{aligned} Y &= \int_0^{+\infty} f(X_t^x) \exp\left(-\mu \int_0^t \mathbb{1}_{\partial G}(X_s^x) d\xi_s^x\right) dt \\ &\quad - \int_0^{+\infty} g_3(X_t^x) \exp\left(-\mu \int_0^t \mathbb{1}_{\partial G}(X_s^x) d\xi_s^x\right) \mathbb{1}_{\partial G}(X_t^x) d\xi_t^x. \end{aligned} \quad (11)$$

Reference 5 gives functional results to problems (9) modified in (12):

$$\begin{cases} -\frac{1}{2} \Delta u + \lambda u = f & (G) \\ \frac{\partial u}{\partial n} - \mu u = g_3 & (\partial G) \end{cases} \quad (12)$$

where the function λ is given from G into \mathbb{R} , with $\lambda \geq \lambda_0 > 0$.

The application of It's formula to:

$$u(X_t^x) \exp\left(-\int_0^t \lambda(X_s^x) ds - \mu \int_0^t \mathbb{1}_{\partial G}(X_s^x) d\xi_s^x\right)$$

leads to the representation of the solution to (12) in the form:

$$\begin{aligned} u(x) &= E \left[\int_0^{+\infty} f(X_t^x) \exp\left(-\int_0^t \lambda(X_s^x) ds - \mu \int_0^t \mathbb{1}_{\partial G}(X_s^x) d\xi_s^x\right) dt \right] \\ &\quad - E \left[\int_0^{+\infty} g_3(X_t^x) \exp\left(-\int_0^t \lambda(X_s^x) ds - \mu \int_0^t \mathbb{1}_{\partial G}(X_s^x) d\xi_s^x\right) \mathbb{1}_{\partial G}(X_t^x) d\xi_t^x \right]. \end{aligned}$$

2.4. Problem with mixed boundary conditions

The partition $\partial G = \Gamma_D \cup \Gamma_N \cup \Gamma_F$ is considered, where the parts Γ_D , Γ_N , and Γ_F correspond respectively to the Dirichlet, Neumann and Fourier boundary conditions. In order to avoid solutions defined up to an additive constant (see Section 2.2), we assume that the interior of Γ_D is not empty.

Let us now consider the following problem with mixed boundary conditions, and written down with the above notation:

$$\left\{ \begin{array}{ll} -\frac{1}{2} \Delta u = f & (G) \\ u = g_1 & (\Gamma_D) \\ \frac{\partial u}{\partial n} = g_2 & (\Gamma_N) \\ \frac{\partial u}{\partial n} - \mu u = g_3 & (\Gamma_F) \end{array} \right. \quad (13)$$

where μ is a strictly positive constant.

Let us introduce the Markov process:

$$X_t^x = x + W_t + \int_0^t \mathbb{1}_{\Gamma_N \cup \Gamma_F}(X_s^x) n(X_s^x) d\xi_s^x, \quad t \geq 0,$$

where $\mathbb{1}_{\Gamma_N \cup \Gamma_F}$ is the indicator of the set $\Gamma_N \cup \Gamma_F$, and where ξ_t^x is a non-decreasing process which only increases when t hits the set $\{s > 0 \mid X_s^x \in \overline{G} \setminus \Gamma_D\}$, so that $\xi_0^x = 0$.

The hitting time τ of Γ_D is defined by:

$$\tau = \inf \{t > 0 \mid X_t^x \in \Gamma_D\}.$$

The application of It's formula to:

$$u(X_t^x) \exp\left(-\mu \int_0^t \mathbb{1}_{\Gamma_F}(X_s^x) d\xi_s^x\right)$$

leads to the representation of the solution to (13) in the form:

$$u(x) = E[Y] \quad (14)$$

with the random variable:

$$\begin{aligned} Y &= \int_0^\tau f(X_t^x) \exp\left(-\mu \int_0^t \mathbb{1}_{\Gamma_F}(X_s^x) d\xi_s^x\right) dt \\ &+ g_1(X_\tau^x) \exp\left(-\mu \int_0^\tau \mathbb{1}_{\Gamma_F}(X_s^x) d\xi_s^x\right) \mathbb{1}_{\Gamma_D}(X_\tau^x) \\ &- \int_0^\tau g_2(X_t^x) \exp\left(-\mu \int_0^t \mathbb{1}_{\Gamma_F}(X_s^x) d\xi_s^x\right) \mathbb{1}_{\Gamma_N}(X_t^x) d\xi_t^x \\ &- \int_0^\tau g_3(X_t^x) \exp\left(-\mu \int_0^t \mathbb{1}_{\Gamma_F}(X_s^x) d\xi_s^x\right) \mathbb{1}_{\Gamma_F}(X_t^x) d\xi_t^x. \end{aligned} \quad (15)$$

The problem (13) can be written in the following condensed form, using appropriate values of coefficients $\bar{\nu}$ and $\bar{\mu}$:

$$\begin{cases} -\frac{1}{2} \Delta u = f & (G) \\ \bar{\nu} \frac{\partial u}{\partial n} - \bar{\mu} u = \bar{g} & (\partial G) \end{cases} \quad (16)$$

Let us note the analogy between problems (16) and (9). Nevertheless, as for the functional resolution, some difficulties occur since the coefficients $\bar{\nu}$ and $\bar{\mu}$ are discontinuous. Some partial results on this subject can be found in Reference 13.

3. RESOLUTION ALGORITHMS

The main purpose of this section is to give an approach of the representations of Section 2 by means of realization of random processes, in order to compute the solutions to boundary value problems.

3.1. Dirichlet problem

The solution u to problem (1) is represented as the mean of a random variable Y involving functions of the standard Wiener process.⁶

The representation (2–3) shows that it is sufficient to compute NT approximated values of Y , i.e. Y_1, \dots, Y_{NT} , in order to obtain an estimation of $u(x)$:

$$u(x) = \frac{1}{NT} \sum_{n=0}^{NT} Y_n.$$

Each value Y_n results from a simulation of a Wiener process as follows. The space step $h > 0$ being fixed, the process X_t^x is simulated by the sequence of random vectors $X_0, X_1, \dots, X_k, \dots$ defined by means of the recurrent formulae:

$$\begin{aligned} X_0 &= x \in G \text{ (being the initialization), and} \\ X_{k+1} &= X_k + h D_k, \quad k \in \mathbb{N}, \end{aligned}$$

where D_k is a random vector so that, if e_i , $i = 1, \dots, d$, is the canonical basis of \mathbb{R}^d , $D_k = \sum_{i=1}^d D_{k,i} e_i$ verifies:

$$\text{Prob}(D_k = e_i) = \text{Prob}(D_k = -e_i) = \frac{1}{2d}, \quad i = 1, \dots, d.$$

For instance, in the three-dimensional case, we generate finite sequences of random numbers $U_0, \dots, U_k, \dots, U_N$, in order to simulate a random variable U , uniformly distributed in the interval $[0, 1]$ by means of pseudo-random number generators. Then we write, for all k :

$$\frac{i}{6} \leq U_k \leq \frac{i+1}{6} \implies D_k = (-1)^i e_j, \quad i = 0, \dots, 5,$$

with

$$j = 1 + \left[\frac{i}{2} \right]$$

where the brackets $[]$ represent the integer part.

In the two-dimensional case, for each time interval, the fixed space step h is stepping along one of the two axes of an orthonormal basis. This random axis is chosen with probability $1/2$, either in the positive or negative direction, being also chosen with probability $1/2$ (see Figure 1 for the stepping and Figure 2 for the basic step algorithm, written in pseudo-Pascal, where the function `random` gives a real number in $[0, 1[$).

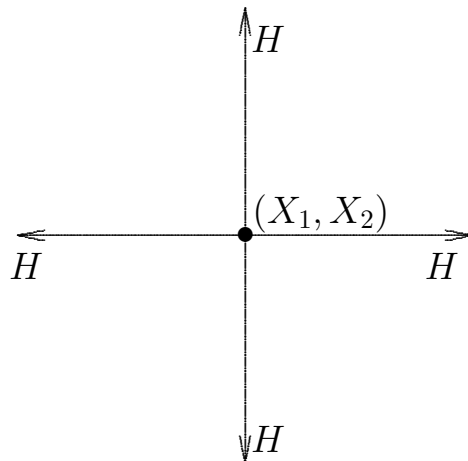


Figure 1. Basic steps of a random walk in the plane, with probability $1/4$ each

```

U1 := random;      U2 := random;
if  U1 < 0.5  then
  if  U2 < 0.5  then X2 := X2 + H
  else X2 := X2 - H
else
  if  U2 < 0.5  then X1 := X1 + H
  else X1 := X1 - H;

```

Figure 2. Algorithm of basic step

Each simulation gives a realization X_k , $0 \leq k \leq N$, of a random walk in G so that:

$$X_0 = x \in G, \dots, X_{N-1} \in G, \quad \text{and} \quad X_N \notin G,$$

which trajectory is the polygonal line connecting successively the points X_k and X_{k+1} , $k = 0, \dots, N - 1$.

The random process X_t^x is discretized in time by the Euler method; in particular, if h is the fixed stepsize of a simulated random walk, the space and time increments are defined by:

$$\Delta X_t^x = h \quad \text{and} \quad \Delta t = h^2/d.$$

For each simulated random walk X_k , we compute the score Y_N :

$$Y_N = \Delta t \cdot \sum_{k=1}^{N-1} f(X_k) + g_1(X_N).$$

Then this procedure is iterated NT times in order to obtain a sample of Y , and an estimation of its mean, i.e. $u(x)$.

The integer NT corresponds to the number of simulated random walks, and consequently to the number of absorptions by \bar{G} . This integer NT is the maximum index of the main loop of the associated program. The second (and latter) loop follows, step by step, each random walk by adding up successively the values of the source function f , then those of the absorption effect. Finally the score of each corresponding Y_N is added up.

Let us consider a domain G in \mathbb{R}^2 and a point $(X1D, X2D)$ given in G . Then a possible algorithm $A1$ of the Dirichlet problem can be written in the form:

```

YM := 0.0;
for i:=1 to NT do
begin
  X1:= X1D;    X2:= X2D;    YA:= 0.0;
  while INTER(X1,X2) = true do
  begin
    YA:= YA + F(X1,X2);
    X1:= X1 + random basic step;
    X2:= X2 + random basic step;
  end;
  YM:= YM + H * H * YA/2 + G1(X1,X2);
end;

```

Figure 3. Dirichlet problem loop

with the procedure `INTER` which checks the absorption of the random walk in progress, and which takes either the value *true* or *false* whether point $(X1, X2)$ belongs to G or not. The function `F` defines the effect of the source for each reached point of the walk inside G , and `G1` defines the absorption effect for the last point.

3.2. Neumann problem

The solution u to problem (4) is represented as the mean of a random variable Y involving functions of a reflected random walk. The representation (6–7) is used and $u(x)$ is computed as the mean of NT approximated values of Y . Each value Y_N results from a simulation of a random walk reflected by the domain boundary and limited by a fixed number NS of steps. Each simulated random walk has the same behaviour in the interior as in the Dirichlet case (see Section 3.1), and is instantaneously reflected by the boundary. Since the simulated process happens to leave the domain, its trajectory resumes to the interior last position; this procedure goes on until the number of steps reaches the value NS . Each random walk is then finite.

In the two-dimensional case, the boundary reflections is reflecting along one of the two axes of an orthonormal basis (see Figure 4 for the reflection step and Figure 5 for the reflection algorithm).

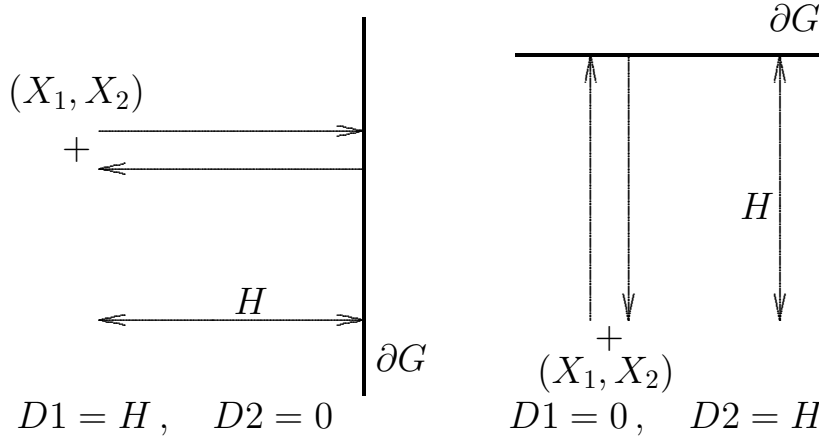


Figure 4. Horizontal and vertical reflections back to the previous position

Procedure Reflection(X_1, X_2)

{ (X_1, X_2) is the position hitting by the random walk }

{ (D_1, D_2) is a shift generating reflection }

if $D_1 \neq 0$ then $X_1 := X_1 - D_1$

if $D_2 \neq 0$ then $X_2 := X_2 - D_2$

Figure 5. Reflection algorithm

The random processes X_t^x and ξ_t^x are discretized in time by the Euler method; in particular, if h is the fixed stepsize of a simulated reflected random walk, then we set the following increments:

$$\Delta X_t^x = h, \quad \Delta t = h^2/d \quad \text{and} \quad \Delta \xi_t^x = h.$$

For each simulated random walk $(X_k; 0 \leq k \leq NS)$, we compute the following score:

$$Y_n = \Delta t \cdot \sum_{k=1}^{NS} f(X_k) - \Delta \xi \cdot \sum_{k=1}^{NS} g_2(X_k) \mathbf{1}_{\partial G}(X_k).$$

This procedure is iterated NT times in order to obtain an estimation of $u(x)$, which is the mean of the NT scores.

The integer NT corresponds to the number of simulated reflected random walks, i.e. the maximum index of the program main loop. The latter loop follows, step by step, each random walk by adding up successively the values of the source function f . For each reflection, a counter COUNT is incremented, and the

values of the effects of the Neumann boundary condition and the source function are stored.

The extension of algorithm *A1* to the Neumann problem is easily implemented:

loop (3) “while INTER(X1,X2) = true do ... end” has only to be replaced by the loop of the step counter. A possible algorithm *A2* of the Neumann problem admits the following interior loop:

```

while COUNT < NS do
  begin
    YA := YA + F(X); { Source effect }
    Basic step Procedure : X := X + D;
    COUNT := COUNT + 1
    INTER( X, IMMO );
    if IMMO = true then
      begin
        Procedure Reflection(X);
        YAN := YAN + G2(X);
                { Neumann condition effect }
        YA := YA + d * F(X);
                { Source effect on the boundary }
      end;
    end;
  end;
YM := YM + H * H * YA / d - H * YAN;

```

Figure 6. Neumann problem loop

with the procedure `INTER` checking the reflection of the random walk in progress. The logical variable `IMMO` takes the value *true* when the random walk hits the boundary.

3.3. Problem with Fourier boundary condition

The solution u to the problem (9) is represented as the mean of the random variable Y , which is a function of a reflected random walk. So as to estimate $u(x)$ by the representation (10–12), we proceed as for the Neumann problem (see Section 3.2). We simulate NT reflected random walks at the boundary, and fix the maximum number of reflections NR . Let NP be the number of

steps of a NR reflections random walk. The random processes X_t^x and ξ_t^x are discretized according to the Euler scheme.

For each simulated reflected random walk ($X_k; 0 \leq k \leq NP$), the following score is computed:

$$Y_n = \Delta t \cdot \sum_{k=1}^{NP} f(X_k) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=1}^k \mathbb{1}_{\partial G}(X_l)\right) - \Delta\xi \cdot \sum_{k=1}^{NP} g_3(X_k) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=1}^k \mathbb{1}_{\partial G}(X_l)\right) \mathbb{1}_{\partial G}(X_k).$$

In order to estimate $u(x)$, we calculate the arithmetic mean of the NT scores. The number NR is the number of the reflections occurring in the NT simulated random walks. For each reflection, a counter **COUNT** is incremented, and the values of the effects of the Fourier boundary condition and the source function are stored.

As for the Neumann problem, the algorithm of the problem with Fourier boundary condition requires two embedded loops checked at each step. The main loop index increases up to the number NT of random walks, the latter loop index up to the number NR of reflections.

The extension of the $A2$ Algorithm to the Fourier case is easily implemented: loop (6) has only to be replaced by the loop of the reflection counter. A possible algorithm $A3$ of the problem with Fourier boundary condition admits the following interior loop:

```

while COUNT < NR do
  begin
    YA := YA + F(X) * EXP(-MU*COUNT*H);
          { Source effect }
    Basic step Procedure : X := X + D;
    INTER( X, IMMO );
    if IMMO = true then
      begin
        Procedure Reflection(X);
        COUNT := COUNT + 1;
        YAF := YAF + G3(X) * EXP(-MU*COUNT*H);
              { Fourier condition effect }
        YA := YA + d * F(X) * EXP(-MU*COUNT*H);
              { Source effect on the boundary }
      end
    end
  end

```

```

end;
end
YM := YM + H * H * YA / d - H * YAF;

```

Figure 7. Problem loop with Fourier boundary condition

where MU is a strictly positive constant.

The stochastic integral:

$$\int_0^{+\infty} f(X_t^x) \exp\left(-\mu \int_0^t \mathbb{1}_{\partial G}(X_s^x) d\xi_s^x\right) dt$$

is discretized in the form of a series, the first term of which is:

$$\Delta t \cdot \sum_{k=1}^{+\infty} f(X_k) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=1}^k \mathbb{1}_{\partial G}(X_l)\right)$$

and can be written in the form:

$$\Delta t \cdot \sum_{k=1}^{NP} f(X_k) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=1}^k \mathbb{1}_{\partial G}(X_l)\right) + R$$

where NR is the number of reflections of the discrete process X_k , NP the step number of the random walk, and R the series:

$$R = \Delta t \cdot \sum_{k=NP+1}^{+\infty} f(X_k) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=1}^k \mathbb{1}_{\partial G}(X_l)\right).$$

Since the function f is bounded and continuous, and the constant μ is strictly positive, the remaining term:

$$R = \Delta t \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=1}^{NP} \mathbb{1}_{\partial G}(X_l)\right) \cdot \sum_{k=NP+1}^{+\infty} f(X_k) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=NP+1}^k \mathbb{1}_{\partial G}(X_l)\right),$$

$$R = \Delta t \cdot \exp(-\Delta\xi \cdot \mu \cdot NR) \cdot \sum_{k=NP+1}^{+\infty} f(X_k) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=NP+1}^k \mathbb{1}_{\partial G}(X_l)\right)$$

converges to zero as NR tends to infinity. Indeed, since Δt and $\Delta\xi$ are fixed positive constants, the exponential term:

$$\exp(-\Delta\xi \cdot \mu \cdot NR)$$

tends to zero as NR tends to infinity, and the last factor is the limit of a convergent series.

The second integral of Y is also discretized in the form of a series which can be treated similarly. We then deduce the convergence of the series as N tends to infinity.

3.4. Problem with mixed boundary conditions

The solution u to problem (13) is represented by the mean of a random variable Y , function of a reflected, then absorbed random walk. The representation (14–16) is used. As for the problems with Dirichlet, Neumann or Fourier boundary conditions (see Sections 3.1, 3.2, 3.3), we estimate $u(x)$ by simulating NT random walks, first reflected by the boundary parts Γ_N and Γ_F , then absorbed by Γ_D .

The number of steps of such random walks with NR reflections is denoted N . The random processes X_t^x and ξ_t^x are discretized in time by the Euler method. For each simulated random walk $(X_k; 0 \leq k \leq N)$, we compute the following score :

$$\begin{aligned}
Y_n = & \Delta t \cdot \sum_{k=1}^N f(X_k) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=1}^k \mathbb{1}_{\Gamma_F}(X_l)\right) \\
& + g_1(X_N) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=1}^N \mathbb{1}_{\Gamma_F}(X_l)\right) \mathbb{1}_{\Gamma_D}(X_N) \\
& - \Delta\xi \cdot \sum_{k=1}^{N-1} g_2(X_k) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=1}^k \mathbb{1}_{\Gamma_F}(X_l)\right) \mathbb{1}_{\Gamma_N}(X_k) \\
& - \Delta\xi \cdot \sum_{k=1}^{N-1} g_3(X_k) \cdot \exp\left(-\Delta\xi \cdot \mu \cdot \sum_{l=1}^k \mathbb{1}_{\Gamma_F}(X_l)\right) \mathbb{1}_{\Gamma_F}(X_k) .
\end{aligned}$$

We estimate $u(x)$ by computing the mean of these NT scores. At each reflection on the boundary Γ_F , the reflection counter `COUNT` increases.

The extension of algorithms $A1$, $A2$ and $A3$ to the mixed case is easily implemented: any of the loop (3), (6) or (7) has only to be replaced by another loop which follows the trajectory of a random walk, step by step, until absorption. A possible algorithm $A4$ to the problem with mixed boundary conditions admits the following interior loop:

```

while INSIDE = true do begin
  YA := YA + F(X) * EXP(-MU*COUNT*H);
  Basic step Procedure : X := X + D;
  INTER( X, INSIDE, IMMO_N, IMMO_F );
  if IMMO_N = true then begin
    Procedure Reflection(X);
    YAN := YAN + G2(X) * EXP(-MU*COUNT*H);
    YA := YA + d * F(X) * EXP(-MU*COUNT*H);
  end; { if }
  if IMMO_F = true then begin
    Procedure Reflection(X);
    COUNT := COUNT + 1;
    YAF := YAF + G3(X) * EXP(-MU*COUNT*H);
    YA := YA + d * F(X) * EXP(-MU*COUNT*H);
  end; { if }
end; { while }
YM := YM + H*H*YA/d + G1(X)*EXP(-MU*COUNT*H) - H*YAN - H*YAF;

```

Figure 8. Problem loop with mixed boundary conditions

where MU is a strictly positive constant.

The computer program includes the functions F , $G1$, $G2$ and $G3$ given by problem (13), the procedures `Basic step` and `Reflection` previously described in Sections 3.1, 3.2 and 3.3, as well as the modified procedure `INTER`:

```

Procedure INTER( X, INSIDE, IMMO_N, IMMO_F );
begin
  INSIDE := true; IMMO_N := false; IMMO_F := false;

  if (X ∈  $\Gamma_D$ ) then INSIDE := false;
  if (X ∈  $\Gamma_N$ ) then IMMO_N := true;
  if (X ∈  $\Gamma_F$ ) then IMMO_F := true;
end;

```

Figure 9. Procedure `INTER` of the problem with mixed boundary conditions

As for the previous algorithms, algorithm $A4$ includes two loops:

- the main loop index increases up to the number of random walks until the fixed maximum NT ;
- the second loop generates a random walk reflected by Γ_N or Γ_F , and then absorbed by Γ_D .

When the trajectory hits Γ_N or Γ_F , one reflection occurs; if it hits Γ_F , counter `COUNT` increases. When an absorption point is hit, a position test stops the simulated random walk.

4. NUMERICAL EXPERIMENTS

The validity and suitability of the present approach, i.e. the stochastic representations in Section 2 and their approximate solutions in Section 3, are investigated here by considering and evaluating a set of boundary value problems. The results are compared with analytical solutions found in the literature.

The algorithms described above have been implemented. The programs, written in Fortran or Pascal, have been run on Sun Spark work stations or compatible PC. We have used some generators of pseudo-random numbers: `random` of Turbo-Pascal, and `ran`, `r_addran()` of Fortran 77.

4.1. Dirichlet problem

Let us consider problem (1) in the ring G defined by:

$$G = \left\{ (x, y) \in \mathbb{R}^2 \mid 1 < \sqrt{x^2 + y^2} < 3 \right\}$$

(cf. figure 10),

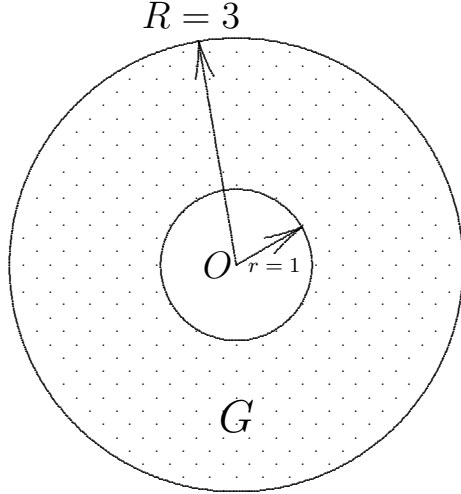


Figure 10. Circular ring G with $\mathcal{C}(O; r = 1) \cup \mathcal{C}(O; R = 3)$ boundary

with the following data:

$$\begin{aligned} f &\equiv 0 && \text{in } G, \\ g_1 &\equiv 4 && \text{on } \mathcal{C}(0; 1) = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}, \\ g_1 &\equiv 6 && \text{on } \mathcal{C}(0; 3) = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 9\}. \end{aligned}$$

This Dirichlet problem admits a unique solution written in the form:

$$u(x, y) = 4 + 2 \frac{\ln \sqrt{x^2 + y^2}}{\ln 3}.$$

For instance, if we use Algorithm A1, we obtain the approximate solution u_c , given in Table I, with the following data:

Point:	$(x, y) = (2; 0),$
Number of simulated random walks:	$NT = 10^6,$
Exact value:	$u(2; 0) = 5.26186.$

Table I. Dirichlet problem
Approximate solution $u_c(2; 0)$

4.2. Neumann problem

The following examples show two problems defined in the previous ring G (cf. Figure 10), with some homogeneous and non-homogeneous Neumann conditions.

Stepsize h	Mean value u_c	Relative error $ u_c - u /u$
0.1	5.27250	$2.0 \cdot 10^{-3}$
0.05	5.27258	$2.0 \cdot 10^{-3}$
0.025	5.26732	$1.0 \cdot 10^{-3}$
0.005	5.26307	$0.2 \cdot 10^{-3}$

4.2.1. Homogeneous Neumann boundary condition.

Let us consider problem (4) in the ring G with the functions f and g_2 defined in G and ∂G respectively by:

$$f(x, y) = \frac{4}{\sqrt{x^2 + y^2}} - 2, \quad \text{and} \quad g_2 \equiv 0.$$

This homogeneous Neumann problem has an exact solution, defined up to an arbitrary additive constant C , which can be written in the form:

$$u(x, y) = x^2 + y^2 - 8\sqrt{x^2 + y^2} + 6 \ln \sqrt{x^2 + y^2} + C.$$

Let us denote the approximate increment between two points of G by δu_c , and the exact increment of a solution by δu_{ex} . Some results obtained with $h = 0.025$, $NT = 10^5$, and $NP = 8\,000$, by means of Algorithm A2, are given in Table II.

Table II. Neumann homogeneous problem — Approximate increments

xPoint	yPoint	Mean value δu_c	Exact value δu_{ex}	Relative error $\frac{ \delta u_c - \delta u_{\text{ex}} }{ \delta u_{\text{ex}} }$
1.2	0.0	-0.780009	-0.775046	$6.4 \cdot 10^{-3}$
2.0	0.0			
1.2	0.0	-1.32062	-1.31621	$3.4 \cdot 10^{-3}$
2.8	0.0			
2.0	0.0	-0.540608	-0.541165	$1.0 \cdot 10^{-3}$
2.8	0.0			

For these data, the average number of reflections \overline{NR} is also obtained: $\overline{NR} = 59$ at the point (1.2; 0.0), 53 at the point (2.0; 0.0), and 72 at the point (2.8; 0.0).

4.2.2. Nonhomogeneous Neumann boundary conditions.

Let us consider again problem (4), with the functions f and g_2 defined in G and ∂G respectively by:

$$f \equiv -2 \quad \text{and} \quad \begin{cases} g_2 \equiv 2 & \text{on } \mathcal{C}(0; 1), \\ g_2 \equiv -6 & \text{on } \mathcal{C}(0; 3). \end{cases}$$

This nonhomogeneous Neumann problem has an exact solution, defined up to an additive constant C , which can be written in the form:

$$u(x, y) = x^2 + y^2 + C .$$

Some results obtained with $h = 0.025$, $NT = 20\,000$, and $NP = 8\,000$ by means of Algorithm A2, are given in Table III.

Table III. Neumann nonhomogeneous problem — Approximate increments

xPoint	yPoint	Mean value	Exact value	Relative error
X1	X2	δu_c	δu_{ex}	$\frac{ \delta u_c - \delta u_{\text{ex}} }{ \delta u_{\text{ex}} }$
1.2	0.0	2.868	2.560	$12.0 \cdot 10^{-2}$
2.0	0.0			
1.2	0.0	7.153	6.400	$11.8 \cdot 10^{-2}$
2.8	0.0			
2.0	0.0	4.285	3.840	$11.6 \cdot 10^{-2}$
2.8	0.0			

These above examples of Neumann problems are analogous to that of Reference 14 in the Dirichlet case with discretized equations.

4.3. Problems with Fourier boundary conditions

The following example is a problem with Fourier condition on the boundary of a square. The second example is an extension to a cube.

4.3.1. Two-dimensional case.

Let us consider problem (9), originating from Reference 15, on the square $G =]0, 1[\times]0, 1[$, with the data:

$$\begin{aligned} \forall (x, y) \in G, \quad f(x, y) &= x(1-x) + y(1-y), \\ \forall (x, y) \in \partial G, \quad g_3(x, y) &= \begin{cases} x(1-x) & \text{if } (0 \leq x \leq 1, y = 0 \text{ or } y = 1), \\ y(1-y) & \text{if } (x = 0 \text{ or } x = 1, 0 \leq y \leq 1), \end{cases} \\ \mu &\equiv 1. \end{aligned}$$

This problem admits a unique solution written in the form:

$$u(x, y) = x(1-x)y(1-y).$$

Using Algorithm A3, Tables IV–V show the results at the interior point (0.5; 0.5), and boundary point (1.0; 0.5) respectively.

- Interior point: $(0.5; 0.5) \in G$ — Exact value: $u = 0.0625$ — Results obtained with $h = 0.025$ and $NT = 10^5$.

Table IV. Problem with Fourier boundary conditions
Approximate solution $u_c(0.5; 0.5)$

Number of reflections	Mean value	Relative error
NR	u_c	$ (u_c - u)/u $
200	0.062713	$3.4 \cdot 10^{-3}$
1000	0.062525	$0.4 \cdot 10^{-3}$

- Boundary point: $(1.0; 0.5) \in \partial G$ — Exact value: $u = 0$ — Stepsize: $h = 0.005$.

Table V. Problem with Fourier boundary conditions
Approximate solution $u_c(1.0; 0.5)$

Sample size: $NT = 10^5$		Number of reflections: $NR = 5\,000$	
Number of reflections NR	Mean value u_c	Sample size NT	Mean value u_c
200	$-5.24 \cdot 10^{-4}$	10^4	$-2.77 \cdot 10^{-4}$
1000	$-2.27 \cdot 10^{-4}$	10^5	$-1.58 \cdot 10^{-4}$
5000	$-1.58 \cdot 10^{-4}$	$2 \cdot 10^5$	$0.89 \cdot 10^{-4}$

4.3.2. Three-dimensional case.

Let us consider problem (9), in the cube $G =]0; 1[\times]0; 1[\times]0; 1[$, with the following data:

$$\begin{aligned} \forall (x, y, z) \in G, \quad f(x, y) &= x(1-x) + y(1-y) + z(1-z), \\ \forall (x, y, z) \in \partial G, \quad g_3(x, y) &= \begin{cases} x(1-x)y(1-y) & \text{if } z = 0 \text{ or } z = 1, \\ y(1-y)z(1-z) & \text{if } x = 0 \text{ or } x = 1, \\ z(1-z)x(1-x) & \text{if } y = 0 \text{ or } y = 1, \end{cases} \\ \mu &\equiv 1. \end{aligned}$$

This problem admits the unique solution:

$$u(x, y, z) = x(1-x)y(1-y)z(1-z).$$

Using Algorithm A3, the results at point (1.0; 0.5; 0.5) belonging to ∂G , are shown in Table VI, with the following data:

$$\begin{aligned} \text{Stepsize:} \quad h &= 0.005, \\ \text{Sample size:} \quad NT &= 10^4, \\ \text{Exact value:} \quad u &= 0. \end{aligned}$$

Table VI. Problem with Fourier boundary conditions
Approximate solution $u_c(1.0; 0.5; 0.5)$

Number of reflections NR	Mean value u_c
200	$-1.42 \cdot 10^{-4}$
1000	$-0.31 \cdot 10^{-4}$

4.4. Problems with mixed boundary conditions

Let us consider problem (13), originating from Reference 16, in the rectangle $G =]0; 1[\times]0; 0.6[$ (cf. Figure 11), with the

following data:

$$\begin{aligned}
 f &\equiv 0 && \text{in } G, \\
 g_1 &\equiv 100 && \text{on } \Gamma_D = \{(x, y) \in \partial G \mid x = 1\}, \\
 g_2 &\equiv 0 && \text{on } \Gamma_N = \{(x, y) \in \partial G \mid x = 0\}, \\
 g_3 &\equiv 0 && \text{on } \Gamma_F = \partial G \setminus (\Gamma_D \cup \Gamma_N), \\
 \mu &= 750/52.
 \end{aligned}$$

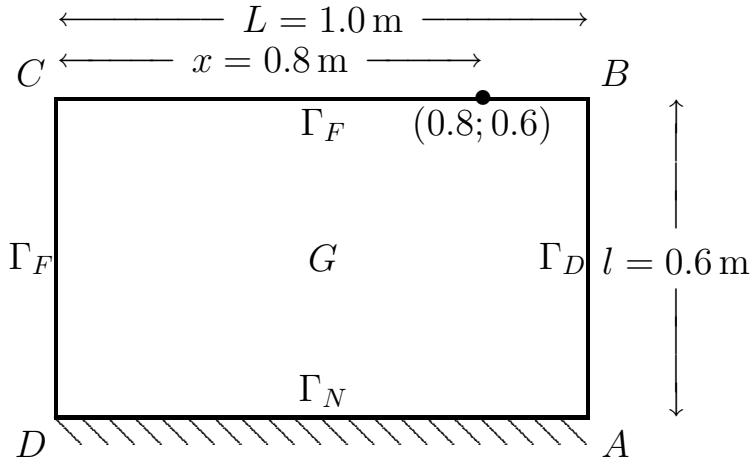


Figure 11. Rectangular domain $G =]0; 1[\times]0; 0.6[$ for the problem with mixed boundary conditions

The value of u at point $(0.8; 0.6)$ is known:¹⁶

$$u(0.8; 0.6) = 18.3.$$

Using Algorithm A4, some results are shown in Table VII.

Table VII. Problem with mixed boundary conditions
Approximate solution $u_c(0.8; 0.6)$

Stepsize h	Sample size NT	Mean value u_c	Relative error $ (u_c - u)/u $
0.005	10^4	18.677	$2.1 \cdot 10^{-2}$
0.002	$2 \cdot 10^4$	18.229	$0.4 \cdot 10^{-2}$

For a fixed sample size $NT = 10^5$, we have obtained the results of Table VIII.

Table VIII. Problem with mixed boundary conditions
 Approximate solution $u_c(0.8; 0.6)$ for different steps

Stepsize of the walk h	Average of reflections* \overline{NR}	Mean value u_c	Relative error $ (u_c - u)/u $
0.1	7	34.145	0.866
0.05	14	25.333	0.384
0.025	30	19.370	0.058
0.005	135	18.485	0.010

* \overline{NR} is the average number of reflections.

It can be noted that the convergence towards the known value is the more effective since the average number of reflections increases (i.e. \overline{NR} is large enough).

5. CONCLUDING REMARKS

We have presented stochastic methods for the numerical approximation of some linear boundary value problems. The representations are deduced from the application of the It's formula for the Dirichlet and Neumann problems with smooth data, and for the problems with Fourier or mixed conditions with non smooth boundary. These representations have given some easily implementable algorithms. The programs are short, easy to write and check step by step.

The extension to an upper dimension problem, in particular the three-dimensional case, only requires a minor modification of the basic step (two lines are only to be added), and a few additional lines in the procedure INTER to locate the boundary. We do not have to store in memory any mesh of the region G , nor deal with the corresponding numbering-lists.

The empirical variance:

$$\sigma_c^2 = \frac{1}{NT} \sum_{n=1}^{NT} (Y_n - u_c)^2$$

where Y_n is the score obtained from the n -th random walk, and u_c the mean value of the NT scores, has been computed:

- for the Dirichlet problem (4.1), σ_c is about 0.15;
- for the problems (4.3.1)–(4.3.2) with Fourier boundary conditions, σ_c is about 0.08 in the two-dimensional case and 0.018 in the three-dimensional case;
- for the problems (4.2.1)–(4.2.2) with homogeneous and non-homogeneous Neumann boundary conditions, σ_c is about 1.6 and 8.2 respectively;
- for the problem (4.4) with mixed boundary conditions, σ_c is about 26;

As for the classical Monte Carlo schemes, this stochastic method admits an expected rate of convergence of about $1/\sqrt{NT}$. The problems with Neumann or mixed boundary conditions request a larger number NT of evaluations. The essential properties of Monte Carlo methods are maintained; in particular, this stochastic method of the solution approximation does not depend on the choice of the point.

Lastly, the random walks are simulated simply by means of repeated Bernoulli's elementary events.

REFERENCES

1. A. Haji-Sheikh, *Monte Carlo methods*, in W.J. Minkowczyk et al. (eds.), *Handbook of Numerical Heat Transfer*, Chapter 16, John Wiley & Sons, Inc., New York, 1988.
2. M. Cessenat, R. Dautray, G. Ledanois, P.-L. Lions, É. Pardoux, and R. Sentis, *Méthodes probabilistes pour les équations de la physique*, Éditions Eyrolles (Série Synthèses), Paris, 1989.
3. G. Marshall, 'Monte Carlo methods for the solution of nonlinear partial differential equations', *Computer Physics Communications*, **56**, 51–61 (1989).
4. H. J. Kushner and P. G. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*, Springer-Verlag, New York, 1992.
5. M. I. Freidlin, *Functional Integration and Partial Differential Equations*, Volume 109 of *Annals of Mathematics Studies*, Princeton University Press, Princeton, New Jersey, 1985.
6. J. E. Souza de Cursi, 'Numerical methods for linear boundary value problems based on Feynman-Kac representations', *Mathematics and Computers in Simulation*, **36**, 1–16 (1994).
7. É. Pardoux and D. Talay, 'Discretization and simulation of stochastic differential equations', *Acta Applicandae Mathematicae*, **3**, 23–47 (1985).
8. P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, volume 23 of *Applications of Mathematics*, Springer-Verlag, New York, 1992.
9. A. Bensoussan and J.-L. Lions, *Contrôle impulsionnel et inéquations quasi-variationnelles*, volume 11 of *Méthodes Mathématiques de l'informatique*, Dunod, Paris, 1982.
10. I. Karatzas and S. E. Shreve, *Brownian Motion and Stochastic Calculus*, Springer-Verlag, New York, 1991.
11. P. L. Lions and A. S. Sznitman, 'Stochastic differential equations with reflecting boundary conditions', *Comm. Pure Appl. Math.*, **37**, 511–537 (1984).
12. M. Pogu and J. E. Souza de Cursi, *Résolution du problème de Neumann — Relation entre équation variationnelle et représentation stochastique — Technique sans maillage*, Research Report, Service de Mathématiques, École Centrale de Nantes, Nantes, 1994.
13. Y. Qingji, 'Solving boundary value problems by means of stochastic calculus', *Mathematica Applicata*, **4**, n°4, 86–97 (1991).
14. D. Zwillinger, *Handbook of Differential Equations*, Academic Press, Inc., Boston, second edition, 1992.
15. P. L. George and K. Leeuwijn, *Bibliothèque thermique MODULEF*, Fiches techniques "Éléments finis" 100, INRIA, Rocquencourt, 1989.

16. B. Muller, D. Steer, and M. Vidrascu, *MODULEF: Quelques exemples de tests de thermique*, Fiches techniques “Éléments finis” 87, INRIA, Rocquencourt, 1988.