1.7. — metody stochastyczne
3.3.1. — liny, pręty, belki
3.7. — wzajemne oddziaływania ciała stałego i płynu, flater
5.13.7. — metody obliczeniowe

P. Kazimierczyk, P. Hagedorn

ON OPTIMAL PLACEMENT OF SPACERS IN BUNDLED CONDUCTORS

3/1994

WARSZAWA 1994

ISSN 0208-5658

Praca wpłynęła do Redakcji dnia 8 listopada 1993 r.

N a p r a w a c h r ę k o p i s u

Instytut Podstawowych Problemów Techniki PAN
Nakład 100 egz. Ark.wyd. 6,70 Ark.druk 8,40
Oddano do drukarni w styczniu 1994 r.

Wydawnictwo Spółdzielcze sp. z o.o.
Warszawa, ul.Jasna 1

"On Optimal Placement of Spacers in Bundled Conductors"

by*:

PIOTR KAZIMIERCZYK and PETER HAGEDORN

* Most of this work was done when the first author, affiliated by the Institute of Fundamental Technological Research Polish Academy of Sciences, was visiting the second author in his Institut für Mechanik, Technische Hochschule Darmstadt, as a Research Fellow of the Alexander von Humboldt Foundation.

Abstract

The paper is devoted to the optimization of the placement of spacer-dampers in bundled conductors of overhead transmission lines. The main purpose of the spacers' use is to align individual conductors in a bundle and to assure the adequate spacing between them. At the same time, however, the spacers may provide damping, which is desirable for minimizing the wind-induced vibrations. Such vibrations, via fatigue of conductors, lead to major damage and maintenance costs (possibly exceeding the costs resulting from such dangerous, but rare, catastrophic phenomena like galloping or short-circuit occurrences). Fatigue occurs mainly at the tension and suspension clamps, and also at the spacer clamps. At these pionts the forces applied to the conductor can, in a first approximation, be assumed as concentrated forces.

In the present paper aeolian excitation due to the wind is modelled as a stationary time-and-space-dependent random field, whose spatial and temporal spectra account for yearly distributions of wind speed and turbulence. This assumption allows calculation of the mean square (MSQ) values of bending stresses at the above-mentioned critical points. Fatigue accumulation theories for stationary processes provide a direct injective relation between the lifetime of a specimen and the MSQ value of loading; thus, to maximize the conductor's lifetime one should optimize the spacer-dampers as well as their placement, with regard to the maximum of the MSQ bending stresses along the span. In this paper we only deal with the optimization of the placement. The maximum of the MSQ of the bending stresses is the cost function in the paper (optimal spacer placement minimizes the cost function). The vibration of conductors is modelled by the wave equation, in this paper.

Although simple models were chosen for both the excitation and the optimality function will depend on a significant number of parameters. The complexity of the problem is magnified by the fact that a large number of eigenmodes (about 500) of each individual conductor lie in the range of technical importance and call for consideration in the study. The presence of spacer-dampers results, at least formally, in the coupling of individual modes, thus factoring the dimension of the problem. Despite this complexity, the method presented allows one to study the most important issues and dependencies, although the computation times are long. The results presented concern the simplest bundle: two conductors vibrating in the same vertical plane. However, also more complex bundles can be treated by means of the method presented, as the computational power of the available machines increases. This is of interest with regard to design, as our preliminary results indicate that the placement of the spacer-dampers may influence the fatigue tremendously: the order of magnitude of the bending stresses may vary with a relatively small variation of the placement.

Introduction

Vibration is one of the critical factors limiting the lifetime of the conductors of overhead transmission lines. Among several mechanisms of vibration build-up the vortex shedding excitation appears to be most important in this respect. It leads to the highest damage costs on the one hand, and, unlike other dangerous but rare phenomena (galloping or short circuit occurrences, say), is almost always present. Fatigue of conductors due to aeolian excitation should therefore be accounted during the design of a transmission line. It can be considered an element of reliability analysis, to a great extent independent of other elements accounting for rare catastrophic phenomena. Precise tools for such a reliability analysis and the optimal design of conductors in transmission lines are not well developed, yet.

For the case of a single conductor the aeolian vibrations are frequently damped by means of Stockbridge dampers as described for instance in [1,2]. In high-voltage overhead transmission lines, however, due to electrical reasons, the conductors are often assembled in the form of bundles. In bundled conductors it is usually no longer possible to obtain adequate damping by means of Stockbridge dampers only. Recently, *spacer-dampers* have been produced which not only provide adequate spacing, but also should assure sufficient damping. The performance of spacer-dampers and their optimal tuning to the conductors were studied recently in [4]. The optimization of the spacers' placement, which is the subject of the present paper, was not addressed in [4].

In this paper the aeolian vibrations of a vertical bundle of two conductors are considered. It is assumed that the conductors are connected through linearly elastic spacers having linear (viscous) damping properties. The mass of the spacer -dampers is neglected. Because of the main purpose of applying the spacers, their distribution along the conductors can only slightly vary from a uniform (equidistant) distribution. However, due to resonance properties, even small changes of their placement, with respect to the uniform one, are important. Therefore, the optimization of the location of spacers around the equidistant positions is considered.

The model employed in this paper is based on the results of a series of previous studies [1-4]. The conductors' vertical vibrations are described by the wave equation, and the bending strains at the clamps are evaluated from the differences of slopes on both sides of the clamps, via singular perturbation. The technically important vibrations are in the frequency range of 5 - 50 Hz approximately. This means that about 500 eigenmodes of each individual conductor should be taken into account. The large number of modes leads to a considerable computational complexity, as the presence of spacer-dampers excludes decoupling of the individual modes.

Numerical experiments indicate that the coupling is negligible for those values of parameters which are of practical importance. To perform the experiments, however, the solution accounting for the coupling had first to be constructed.

Despite the complexity an efficient iterative algorithm has been found for solving the eigenvalue problem associated with the undamped system (purely elastic spacers). The algorithm can be used for any number of spacers. Using the eigenfunctions of the associated undamped problem the equations for the modal expansion of the full equation can be found. These equations form an infinite system of coupled linear second order ordinary differential equations with constant coefficients, with Fourier temporal coefficients of the loading on the right-hand side.

In our analysis the infinite system of equations for the temporal Fourier coefficients of the vertical displacement of a conductor is replaced by a finite system accounting for the first M modes of vibrations, where M was taken up to $M = 500$. Mathematically, this is equivalent to the assumption that the damping coefficients are infinite for higher order modes. The results of calculations performed for realistic values of parameters provide additional confirmation of the validity of this assumption; calculations with 15 significant digits show the differences between results obtained for $M = 100$ and $M = 500$ only atthe 5-th to 10-th digit.

The system of equations for the temporal Fourier coefficients of the vertical displacements of the conductors, together with the eigenforms for an undamped related problem, can be used for calculating the mean square values of the bending strains in the conductor. For large values of M, only an approximate solution can be arrived at within a reasonable CPU-time. Two methods were used to approximate the solution: a method based on the Green's-function, and the spectral method. Since the first one leads to the less efficient algorithm it was only used for small values of M, and in this case both methods show good agreement. All numerical results presented in this paper were obtained by using the spectral method.

The idea behind the first method was to approximate the system's Green's function. The terms representing the damping forces (dependent on the solution) are grouped on the right-hand sides of the equations for normal coordinates. Integral equations can then be derived for the damping terms. Due to the boundedness of the integral kernels and the contracting properties, an iteration leads to the convergence of consecutive approximations. Using the resulting approximate expressions for the damping terms, the global solution of the governing equation can be approximated in integral form, where the (known) integral kernel is an approximation of the Green's function, and the operator acts on the external random field, modelling the wind excitation.

Using the linearity of this representation with respect to the random external field, the second moments of the bending strains at the "nodes" (i.e. at the conductor clamps and at the spacer clamps) can be expressed as the values of certain integral operators on the correlation function of the loading field. The assumed form of this correlation function accounts for the span-wise correlation of the vortex shedding excitation mechanism as well

as for the distribution of the intensity of this excitation, reflected by the yearly distribution of the mean wind speeds.

The second method to approximate the MSQ bending strains is based on the spectral representation for the forces at the points of interest. Since the system is linear, the spectra of the forces are linearly related to the spectrum of the loading. To retrieve them from the assumed loading spectrum one has to approximate an inverse Fourier transform of a quadratic expression containing the inverses of a complex $M \times M$ transfer matrix of the system of equations for temporal Fourier coefficients of conductor displacements. Two orders of approximation were utilized. The zero order approximation is equivalent to neglecting the coupling between different modes of vibrations. The first order approximation is identical to the zero order one (that is, all the first order terms vanish but those of the zero order). The second order approximation takes the coupling into account. A remarkable result of this study is the observation that the coupling is not significant for practical values of the parameters. This allows utilization of the zero order approximation requiring the M^2 order algorithm instead of the M^4 order algorithm which is necessary for the second order approximation (the Green's function first order approach also leads to an M^4 order algorithm).

In the above described way we calculated the mean square bending strains at all spacer clamps as well as at the suspension clamps. The penalty function is defined as the largest of these mean square values. The placement of the spacers is considered optimal if it minimizes this penalty function.

The rationale behind this approach is that, on a short-time scale, the nature of (instantaneous) aeolian excitation is well described by a narrow-band Gaussian process, so that the Palmgren-Miner theory of accumulated damage permits to express the lifetime at each clamp directly in terms of the above-mentioned mean-square values. On the other hand, the shortest of the conductor lifetimes at the spacers' clamps is the bundle lifetime.

The results presented are restricted to the case of the vertical bundle of two conductors aligned with the help of a few spacers. The algorithm is, however, capable of dealing with an arbitrary number of spacers, and the method can be generalized to the case of more general bundles of conductors (not only to pairs of conductors).

A glance at the results suffices to draw two fundamental conclusions. First, that the nature of the problem is extremely involved, and no simple argument can predict the results. The method proposed provides a tool which, at least quantitatively, reveals the influence of the most important parameters on the bending strains in the conductors and, consequently, on their fatigue. Second, that the variations of the bending strains are so sensitive to small changes in the spacing, that the careful analysis of the placement problem may considerably influence the bundle lifetime. Also with respect to the construction of the spacers some conjectures can be formulated even from the few numerical experiments presented in this paper. For example, the dangerous peaks of bending strains are strongly related to the *structural damping parameter*. This is particularly important where the spacers are distributed regularly. There, their clamps should damp rotation of the clamp with respect to the spacer body, rather than only stretching or elongation of the spacer.

The ratio between stiffness and damping of the spacers can also be optimized by means of the algorithms developed. Although the models for conductors, spacers, and excitations are all far too simple to reflect *all* aspects of the real systems, at least some of the dependencies revealed by the present study should be taken into account in designing conductor bundles.

1 The Model

A mathematical model to be used for our analysis has to describe the conductors of an overhead power line, the spacers, as well as the interaction between a vibrating bundle and the moving mass of air. A conductor itself has a complex structure, it is formed by a steel core and many aluminium wires. Similarly, the spacers are not at all easy to describe. Even the simplest design consist of several bodies interconnected by means of visco-elastic-plastie joints. Interaction of the structure with the air is also a complicated phenomenon. Thus, it seems to be unlikely that any detailed model describing the vibrations under consideration, may alow for a computationally efficient calculation of optimal spacer positions. On the other hand, however precise the optimization of the spacers' location may be in theory, the installation of the spacers on the real bundle can only be accomplished within limited precision. The precision of mounting is of the order of one thousandth of a conductor span. This enhances the sensitivity of a particular solution on the spacers' positions, rather than the precision of modelling.

In most existing publications the model of a conductor is assumed in the form of the linear beam equation with constant coefficients (in particular with constant tension, thus the effect of the sag and gravitation are neglected). The wave equation approximation is utilized at least equally frequently. The quality of this approximation was recognized as satisfactory for many purposes already in the fifties (cf. [5]). As we are mostly interested in the bending strains at the clamps, the wave equation approximation serves as an appropriate model (see [1], where the method of singular perturbations was used to establish the formulae for bending strains at these points).

The spacer-dampers available on the market have different forms and properties. Our method applies to a wide range of them, but most of the numerical results concern a simple spacer consisting of a straight main body with a head attached at each end through a layer of vulcanized rubber. The spacers are attached to the conductors by clamps embracing the conductors. Usually, the joint is quite flexible with respect to small angular displacements of a conductor with respect to the spacer's body, and we assume that only forces parallel

to the spacer's body are transmitted. This kind of a spacer is relatively stiff, and the rubber present in the heads provides only a small amount of damping. Within the range of typical forces, the spacer will be assumed as linearly elastic, and the damping viscous. Other spacers, especially the z-shaped spacer-dampers, are much more flexible and provide higher damping.

The mass of a spacer of the type being considered is of the order of one thousandth of a single conductor mass. Therefore, for small amplitudes of vibration in the frequency range of 5 - 50 Hz, the inertia forces resulting from accelerating the spacers' masses are negligible compared to the damping and elastic forces. For this reason the masses of the spacers are neglected in what follows.

Summing up, we assume the model of a vertical bundle of two conductors (refer to the Figure 1(a)) in the following form:

$$\rho W^1_{tt} = TW^1_{xx} + \sum_{i=1}^{N} i\delta(x - x_1)F_i + z_1, \tag{1.1}$$

$$\rho W^2_{tt} = TW^2_{xx} - \sum_{i=1}^{N} i\delta(x - x_1)F_i + z_2, \tag{1.2}$$

$$F_i = \left[(W^2 - W^1)K + (W^2_t - W^1_t)C\right]_{x=x_i}, \tag{1.3}$$

$$W^j|_{x\in\{0,L\}} \equiv 0, \quad W^j_t = W^j|_{t=t_o} \equiv 0; \quad j = 1,2; \quad t_o \to -\infty. \tag{1.4}$$

Here $W^j = W^j(x,t)$ is the vertical displacement of the j-th conductor with respect to the static equilibrium position, $F_i = F_i(t)$ is the vertical force between the conductors and the i-th spacer acting on the conductor, ρ is the conductor's mass per unit length, T is the conductor tension, K is the stiffness of a spacer, C is its damping constant. $z_j = z_j(x,t)$ are the random fields of external loads resulting from the interaction of the conductors with air. For the purpose of our study, in which the mean-square values of forces at points $\{0, x_1, \ldots, x_N, L\}$ are sought, only the second-order characteristics of the random fields (z_1, z_2) are necessary. Therefore we content ourselves with the assumption that the random fields z_1, z_2 are such that the field

$$\tilde{z} = z_1 - z_2 \tag{1.5}$$

is a stationary, centered random field whose correlation function factorizes to

$$E\tilde{z}(x,t)\tilde{z}(x + \Delta x, t + \Delta t) = \tilde{\sigma}^2 \tilde{K}_x(\Delta x)\tilde{K}_t(\Delta t), \tag{1.6}$$

where

$$\tilde{K}_x(\cdot) = \exp\{-\tilde{\alpha}|\cdot|\}\cos(\tilde{\mu}\cdot), \tag{1.7}$$

$$\tilde{K}_t(\cdot) = \exp\left\{-\tilde{\lambda}|\cdot|\right\}\cos(\tilde{\nu}\cdot). \tag{1.8}$$

The local (in space and time sense) nature of aeolian excitation is rather that of a narrow-band quasi-stationary process at each point of the conductor. The dominant frequency is close to that assigned to a particular wind speed and the Reynold's number by the Strouhal relation. To some extent it may be modified by the lock-in mechanism ([6, 7]) . As to the spatial correlation not much is known, but that it is quite strong within the range of some 10 conductor diameters. This information is not sufficient for modelling overhead power lines, as a typical conductor span is about 10^4 times larger!

The parameters $\tilde{\mu}, \tilde{\nu}$ can be interpreted as the yearly characteristics of typical temporal and spatial frequencies of the air vortices along an conductor. It should be mentioned that the proper values of the parameters $\tilde{\mu}, \tilde{\nu}, \tilde{\alpha}$, and $\tilde{\lambda}$ for a particular location of the span, may strongly depend on the local terrain configuration and roughness.

Note that in particular power line locations different types of correlation functions may better describe the properties of the aeolian excitation $\tilde{z}$. The method employed in the following can also be applied for other correlation functions different from (1.6), only some formulae would have to be changed.

Our choice of the type of covariance function originated from four reasons. The first reason was the heuristic model described above. The second was the outcome of experiments performed on a real structure (see [3]; no spatial correlation was estimated). The third was mathematical convenience: the spectral densities (the Fourier transforms of these correlation functions) are rational functions. This property is helpful in obtaining closed formulae for variances of strains as inverse transforms of products of these spectra with the transfer function of the linear system of differential equations with constant coefficients for the temporal Fourier coordinates of the solution to the governing equations.

The fourth reason behind selecting the correlation functions in the form (1.6), (1.7) is that they are relatively simple, yet depend on a few parameters, which can be given an useful physical interpretation. Thus, varying the parameters one can gain some insight into the dependence of the mean-square forces at the clamps on the characteristic turbulence and the characteristic Strouhal frequency.

Different tensions, T, in equations (1.1), (1.2) could also be assumed. In this case, the dimension of the problem would be raised to the second power, as the equations could not be decoupled, and the calculation of the mean-square forces would require the analysis of a system of both equations.

Under the present assumption, subtracting (adding) equation (1.2) from (to) equation (1.1) and defining:

$$u = W^1 - W^2, \qquad v = W^1 + W^2, \tag{1.9}$$

leads to the system of equations

$$\rho u_{tt} = T u_{xx} + 2 \sum_{i=1}^{N} \delta(x - x_i) F_i + \tilde{z}, \tag{1.10}$$

$$\rho v_{tt} = T v_{tt} + z_1 + z_2, \tag{1.11}$$

which is decoupled, because (1.3) can be rewritten as

$$F_i = [-Ku - Cu_t]|_{x=x_i}. \tag{1.12}$$

The system (1.10), (1.12), together with uniform boundary and initial conditions following from (1.4), constitutes the basic model. For notational convenience let us introduce the dimensionless spatial variable

$$y = \frac{x}{L}. \tag{1.13}$$

Define also

$$\begin{array}{lllll} y_i := x_i/L, & s := \frac{t}{L}\sqrt{\frac{T}{\rho}}, & s_0 := \frac{t_0}{L}\sqrt{\frac{T}{\rho}}, & \kappa := 2K/T, & \beta := \frac{2C}{T\rho}, \\ \sigma := \tilde{\sigma}\frac{L}{\sqrt{T}}, & \alpha := \tilde{\alpha}/L, & \mu := \tilde{\mu}/L, & \lambda := \frac{\tilde{\lambda}}{L}\sqrt{\frac{\rho}{T}}, & \nu := \frac{\tilde{\nu}}{L}\sqrt{\frac{\rho}{T}}, \end{array} \tag{1.14}$$

and

$$w = w(y,s) := u\left(L\,y, L\,s\sqrt{\frac{\rho}{T}}\right). \tag{1.15}$$

With this notation the basic equations take the following form:

$$\begin{array}{l} w_{ss} - w_{yy} + \sum_{i=1}^{N} \delta(y - y_i)[\kappa w + \beta w_s] = z, \\ w(0,s) = w(1,s) = w(y,s_0) = w_s(y,s_0) \equiv 0, \qquad s \geq s_0, \quad y \in <0,1>. \end{array} \tag{1.16}$$

Note that while replacing the spatial variable x by $y = \frac{x}{L}$, the Dirac-delta distribution $\delta(x - x_i)$ is to be replaced by $L \cdot \delta(y - y_i)$.

We recognize (1.16) to be the model of a string attached to the ground through linearly elastic springs and viscous dampers (see Figure 1(b)).

Our main object of interest are the mean-square values of the bending forces in the conductor. Bearing in mind the beam equation, one realizes that the bending strains assume their maxima at the points of the maximal curvature. These can be identified with the conductor ends and the points at which the concentrated forces are applied. Making use of the singular perturbation method, it was shown in [1] that, at these points, the bending strains can be approximated by the values of the vertical forces resulting from the solution of the wave equation, obtained from the original beam equation by letting the bending stiffness of the beam go to zero. In other words, our original task is equivalent to the one connected with the forces F_i themselves, and with the reaction forces at both conductor ends. The forces F_i can be recognized as the following functionals on the solution $w(y,s)$:

$$F_i(t) = K\,w(y_i,s) + C\,L\sqrt{\frac{\rho}{T}}\,w_s(y_i,s).$$

For computational reasons, however, it is more useful to employ formulae given in [1], which are also valid for the end-points of the string. Therefore, we identify the bending strains of interest with the forces $F_i(t)$ $(i = 0, 1, 2, \ldots, N+1)$ calculated as

$$\begin{aligned} F_i(t) &= T\left(u(x_i^-, t) - u(x_i^+, t)\right) \\ &= LT\left(w_y(y_i^-, s) - w_y(y_i^+, s)\right), \end{aligned} \tag{1.17}$$

where, by definition, $w_y(0^-, s) = w_y(1^+, s) \equiv 0$. Obviously, the analysis of the forces (1.17) is equivalent to that of the variables (1.15)

$$\phi_i(s) = w_y(y_i^-, s) - w_y(y_i^+, s). \tag{1.18}$$

The functions ϕ_i (we will call them *strains*, although due to the transformation of variables they have no dimension of strain) are linear functionals of the past of the external excitation

$$z(y, s) := \frac{L}{\sqrt{T}}\, \tilde{z}\left(L\, y, L\, s\sqrt{\frac{\rho}{T}}\right).$$

We have assumed that z is a stationary random field with a finite variance σ^2. Consequently, we observe that if the damping present in the system is large enough, then all stochastic processes ϕ_i will be second order processes with finite, time dependent variances $\sigma_i^2(s - s_0)$. Letting s_0 tend to minus infinity we arrive at the stationary values of the variances δ_i^2. However, the amount of damping in the system depends on the positions of the spacers. It is clear that, for some spacer locations, certain modes of vibration can not be damped in our model. This would lead to mean-square instabilities of our solution. In physical systems, however, damping is always present. There exists internal friction in the conductors. When the amplitudes of vibration are large, the air itself damps the vibration. Our simplified model is not capable to describe well such phenomena. However, it can be improved by simple means.

First, let us recall what was said about the damping properties of the spacers. We have assumed that the higher frequencies are completely damped. This means that, although we use a viscous damping model for spacers, reflected by the term βw_s in equation (1.16), we have another linear operator in mind, rather than simple time differentiation: we think of a linear operator whose Laplace transform is equal to a differentiator within a finite band, and is equal to zero outside this band.

Second, most of the damping mechanisms omitted until now in the model (1.16) can be, in first approximation, modelled by an additional term in the following form: $D_{str} = d_{str} \cdot w_s(y, s)$, where d_{str} is a constant, which is small in comparison with the damping constant β describing the effectiveness of spacers. This constant may be given different interpretations. We would like to interpret it as a sum of a structural damping constant characterizing the rate of internal energy dissipation in the conductor and the friction between the conductor and the air (as pointed out above, this kind of friction is important when the vibration amplitudes are large; in this case the velocities of the conductor

segments significantly exceed the velocities of the air vortices; accordingly, the component D_{str} is negligible for small velocities and becomes important for large velocities).

Another kind of damping might have been introduced into the model, namely the damping of rotational displacements at the spacers' clamps. This kind of an interaction between spacers and the conductor, modelled as a string, would require a term of the form $\delta(y - y_i)\frac{d}{dt}(w_y(y_i^-) + w_y(y_i^+))$. It will be seen in section 3 that, mathematically, such a term would add to the one just proposed in the equations for the temporal Fourier coefficients of the solution, w. This means that the structural damping (playing such an important role in assuring stability of the system), can be controlled by introducing the rotational damping in spacers' clamps.

After we have introduced the damping D_{str}, which appears along the entire span of the string and disappears only at the nodal points, we realize that the response of the string to a zero-mean excitation z is stable in the mean-square sense. The magnitude of the stationary variances of the strains is finite and depends on the ratios of the intensity σ and damping constants β and d_{str} as well as on the positions of the spacers. Let us note that this kind of stability does not guarantee that the level of vibrations must necessarily remain within the safe level for each individual realization of the stochastic response process under consideration, even if the stationary variances σ_i remain well below any prescribed bounds. It would be of interest to investigate the so-called technical stability, requiring that, with probability one, a sample-path of the response of the system, starting from certain restriction of the safe region will remain within the safe region. For such analysis, however, additional assumptions would be necessary, concerning the nature of the excitation process. Since all we would like to assume is that z is a stationary field of second order we content ourselves with the mean-square stability assuring the existence of these values, which are the main object of our study: the stationary variances of strains.

Summing up our modelling consideration, we formulate our optimization problem as follows.

Given the initial value problem

$$\begin{aligned} &w_{ss} - w_{yy} + \sum_{i=1}^{N} \delta(y - y_i)\,[\kappa w + \beta w_s] + d_{str} w_s = z, \\ &w(0,s) = w(1,s) = w(y,s_0) = w_s(y,s_0) \equiv 0, \end{aligned} \tag{1.19}$$

for $s \geq s_0, \quad 0 < y < 1$, where the external excitation z is a stationary, centered, random field with the correlation function

$$\begin{aligned} Ez(y + \Delta y, s + \Delta s)z(y,s) &= \sigma^2 K_y(\Delta y) K_s(\Delta s), \\ K_y(\cdot) &= \exp\{-\alpha|\cdot|\}\cos(\mu\cdot), \\ K_s(\cdot) &= \exp\{-\lambda|\cdot|\}\cos(\nu\cdot), \end{aligned} \tag{1.20}$$

find the placement $Y := \{y_1, y_2, \ldots, y_N\}$ (y_i is a coordinate of the point at which i-th of N spacers is to be mounted) such that $y_0 = 0 < y_1 < y_2 < \ldots < y_N < 1 = y_{N+1}$ and

the following cost function, $Q(y)$, is minimized:

$$Q(Y) := \max_{i=0,1,2,\ldots,N+1} Q_i^2. \tag{1.21}$$

Here

$$Q_i^2 := \lim_{s_0 \to -\infty} E[w_y(y_i^-, 0) - w_y(y_i^+, 0)]^2 \tag{1.22}$$

(recall that, by definition, $w_x(0^-, s) = w_x(1^+, s) \equiv 0$).

The rationale behind the above choice of the cost-function is as follows. The expected lifetime at any given point y_i is a strictly decreasing function of the mean square value, $Q_i^2 = E\phi_i^2$, of the bending strain at this point. But the conductor's lifetime is the shortest of the lifetimes at the critical points y_i. Thus, the point at which the bending strains assume the largest mean-square value is the most susceptible to the fatigue damage and the mean-square value of strains at this point is the indicator of the conductor's lifetime.

The cost function Q as well as each of the mean-square strains Q_i^2, are not only the functions of Y, but also depend on all the parameters contained in the model. The methods developed in what follows enable us to investigate these dependencies (not only the relation $Y \to Q(Y)$, which is the main object of interest).

2 Eigenvalue problem for an associated spatial operator

On the way to the variances Q_i^2 the following relations will be utilized: $\{\sigma, K_y, K_s\} \leftrightarrow z \leftrightarrow w \leftrightarrow \phi_i \leftrightarrow Q_i^2$. The first relation is given by the formulae (1.20). The third one is nothing but the equation (1.18). Two different ways to use the fourth relation (1.22) will be discussed in chapters 4 and 5, respectively. The second relation, $z \leftrightarrow w$, is the subject of this and the next sections. The idea is to employ the well-known method of separation of variables, and the associated modal technique. In this chapter we prepare the tools to realize this plan. By solving the eigenvalue problem for an spatial differential operator appearing in the equation (1.19) we construct an orthonormal family of functions. These functions will be used in the next chapter, as the modal coefficients in the Fourier expansion of the solution, w, to the entire initial problem (1.19).

Let us consider the eigenvectors of the following differential operator

$$B := \frac{d^2}{dy^2} - \sum_{i=1}^{N} \kappa\delta(y - y_i). \tag{2.1}$$

The operator B is self-adjoint on the space of piece-wise analytical, continuous functions of argument $y \in (0;1)$, satisfying homogeneous boundary conditions at points 0 and 1. Therefore, the family of its eigenfunctions constitutes a suitable basis in which the solution to the equation (1.19) can be sought after.

To find the eigenvalues of the operator B we make use of the concept of a Green's function (cf. [8]). That is, at first, we look for a solution to the following problem (see Figure 2).

$$\frac{d^2}{dy^2}v + \lambda v = 0, \qquad y \in (0;l) \cup (l;1), \tag{2.2}$$

$$v(0) = v(1) = 0 \tag{2.3}$$

$$v(l^-) = v(l^+) \tag{2.4}$$

$$\frac{d}{dy}v(l^-) - \frac{d}{dy}v(l^+) = 1 \tag{2.5}$$

It can be verified that the solution is given by the following function $g : (0;1)\times(0;1) \to \mathbb{R}$, ($q = \sqrt{\lambda}$ below).

$$g(l,y) = \frac{1}{q \sin q} \begin{cases} \sin(q - ql) \sin qy, & \text{for} \quad y \le l; \\ \sin ql \sin(q - qy), & \text{for} \quad y > l; \end{cases} \tag{2.6}$$

The function g is the Green's function for the operator $\frac{d^2}{dy^2} + \lambda$, with homogeneous boundary conditions. For any function f, the solution of the following inhomogeneous equation (compare with eq. (2.2))

$$\frac{d^2}{dy^2}v + \lambda v = f,$$

with boundary conditions (2.3), satisfies the identity:

$$v(y) = \int_0^1 g(y,l)\, f(l)\, dl. \tag{2.7}$$

Let v be an eigenfunction of the operator B, associated with the eigenvalue $-\lambda$. Then $Bv = -\lambda v$, that is

$$\frac{d^2}{dy^2} + \lambda v = \kappa \sum_{i=1}^{N} \delta(y - y_i)v. \tag{2.8}$$

As we consider B on the space of functions satisfying conditions (2.3), we can make use of the identity (2.7) with $f(l) = \sum_{i=1}^{N} \delta(l - y_i)v(l)$. Thus, we obtain

$$\begin{aligned} v(y) &= \int_0^1 g(y,l) \sum_{i=1}^{N} \delta(l - y_i)v(l)dl \\ &= \kappa \sum_{i=1}^{N} g(y, y_i)v(y_i), \qquad (2.9) \\ &= \frac{\kappa}{q \sin q} \left[\sum_{\{i|y_i \le y\}} \sin(q - qy) \sin qy_i \, v(y_i) + \sum_{\{i|y_i > y\}} \sin(q - qy_i) \sin qy \, v(y_i) \right] \end{aligned}$$

which is a representation of an eigenfunction in terms of its values at points $y_1, \ldots, y_N$. Equality (2.9) must hold in particular for $y = y_i, i = 1, \ldots, N$. Thus, from (2.6) and (2.9) we obtain the N conditions:

$$\begin{aligned} \frac{1}{\kappa} q v(y_i) = \frac{1}{\sin q} \Bigg[& \sum_{\{j|y_j \le y_i\}} \sin(q - qy_i) \sin qy_j \, v(y_j) \\ & + \sum_{\{j|y_j > y_i\}} \sin qy_i \sin(q - qy_j) v(y_j) \Bigg]. \end{aligned} \qquad (2.10)$$

Denoting:

$$\mathbf{V} = [v_1, v_2, \ldots, v_N]^T = [v(y_1), v(y_2), \ldots, v(y_N)]^T,$$

and defining:

$$\mathbf{I}_N = \{\delta_{ij}\}_{i,j=1,\ldots,N}, \qquad \mathbf{A}_N = \{a_{ij}\}_{i,u=1,\ldots,N},$$

where

$$a_{ij} = \begin{cases} \sin qy_j \, \sin(q - qy_i), & \text{for} \quad j \le i; \\ \sin qy_i, \, \sin(q - qy_j), & \text{for} \quad j > i; \end{cases}$$

we can rewrite equality (2.10) in the matrix form:

$$(\mathbf{A}_N - \frac{q}{\kappa} \sin q \, \mathbf{I}_N) \, \mathbf{V} = \mathbf{0}. \qquad (2.11)$$

Clearly, equation (2.11) may have a non-trivial solution only if

$$\mathrm{Det}\,(\mathbf{A}_N - \frac{q}{\kappa} \sin q \cdot \mathbf{I}_N) = 0. \qquad (2.12)$$

From this requirement we can derive the eigenvalues of the operator B. There are two qualitatively different families of solutions. One family is associated with those values of q, for which: $\sin q = 0$, and $\sin qy_i = 0$, $i = 1, \ldots, N$. The second family is obtained for q such that $\sin q \neq 0$.

Note that if $\sin q = 0$, but $\sin q_i \neq 0$ for at least one $i \in \{1, 2, \ldots, N\}$, than the solution to (2.11) does not define an eigenvector of the operator B. This is because (2.11) is not fully equivalent to (2.10): it results from (2.10) by multiplying both of its sides by $\sin q$. Therefore, if $\sin q = 0$ then (2.11) might still hold true, while (2.10) would become meaningless if one of $\sin q y_i \neq 0$. Simultaneously, the problem (2.2)–(2.5) has no solution for $\sin q = 0$. In other words, for q such that $\sin q = 0$, no non-zero concentrated forces can act at the string (2.2) to (2.5), admitting any solution of this auxiliary static problem. Consequently, for $q = k\pi$ there exists no eigenfunction of the operator B, taking non-zero values at the points where the spacers are mounted. Indeed, if the displacement at the spacer's clamp is non-zero, then the concentrated force cannot disappear at this point as, in the static case, the only part of the force coming from the spacer is the elastic force proportional to the displacement. Therefore, the non-trivial solutions to (2.11), which are associated to the values $q = k\pi$ have to be disregarded as eigenvalues for the operator B.

For such $q = k\pi$, however, that not only $\sin q = 0$ but also all $\sin y_i = 0$, a non-trivial solution to the equation (2.8) exists and is equal to the function $v(y) = \sin y$. Clearly, at all points y_i the displacement is equal to zero and no concentrated reaction forces appear. Note that for this type of eigenmode no damping is provided by the spacers.

If $\sin q \neq 0$ all elements of the matrix $\mathbf{A}$ are, in general, different from zero. Numerical calculation of the determinant (2.12), and the usage of general numerical procedures for finding eigensolutions $\mathbf{V}$ from (2.11) would be quite time-consuming, especially for a large number of spacers. The problem is critical, as we require a large number of eigenvalues (up to 500) and, given no analytical formula for the determinant (2.12), the only way to find its roots is the numerical search. This way requires frequent computations of the determinant (2.12). In order to minimize this part of calculations, a specific algorithm was developed to find the eigenvalues and eigenvectors of the matrix $\mathbf{A}$, using its particular form.

To shorten the notation let us define:

$$
\begin{array}{ccc}
s_i^a = \sin q y_i, & s_i^b = \sin(q - q y_i), & s = \sin q, \\
c_i^a = \cos q y_i, & c_i^b = \cos(q - q y_i), & c = \cos q, \\
d_1 = s_1^a, & r_1^1 = s_1^a d_1, & r_1^2 = c_1^a d_1, \\
\Sigma_1 = s_1^b d_1, & d_k = s_k^a + \dfrac{\kappa}{q}(c_k^a r_{k-1}^1 - s_i^a r_{k-1}^2), & \Sigma_k = \Sigma_{k-1} + s_k^b d_k, \\
r_k^1 = r_{k-1}^1 + s_k^q d_k, & r_k^2 = r_{k-1}^2 + c_k^a d_k, & D_k = \Sigma_k - \dfrac{q}{\kappa} \cdot \sin q.
\end{array}
\tag{2.13}
$$

Theorem 1. *For any natural number N, if $\sin q \neq 0$, then*

$$
Det\left(\mathbf{A}_N - \frac{q}{\kappa}\sin q \cdot \mathbf{I}_N\right) = \left(-q\frac{\sin q}{\kappa}\right)^{N-1} D_N. \tag{2.14}
$$

If $D_N = 0$, then the vector $\mathbf{V} = [d_1, d_2, \ldots, d_N]^T$ satisfies the equation (2.11).

PROOF: From the definition of matrix $\mathbf{A}$, determinant (2.14) has the following form:

$$\begin{vmatrix} s_1^a s_1^b - \frac{sq}{\kappa} & s_1^a s_2^b & s_1^a s_3^b & \cdots & s_1^a s_N^b \\ s_1^a s_2^b & s_2^a s_2^b - \frac{sq}{\kappa} & s_2^a s_3^b & \cdots & s_2^a s_N^b \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_1^a s_N^b & s_2^a s_N^b & s_3^a s_N^b & \cdots & s_N^a s_N^b - \frac{sq}{\kappa} \end{vmatrix}.$$

Assume that $s_1^a \neq 0$. In the opposite case the first row and the first columns of the determinant would be equal to zero, with the exception of the diagonal element, which would be equal to the factor $-\frac{sq}{\kappa}$, appearing on the right side of equality (2.14). Therefore, the determinant would be equal to the product of the factor and a sub-determinant resulting from the original one by 'striking out' the first column and the first row. It can be verified that the inductive definitions of symbols (2.13) remain valid, and to arrive at both theses it would suffice to prove them for the remaining part of the determinant. So, the task would be reduced to the initial one, with s_2^a in place of s_1^a. If s_2^a were equal to zero, this procedure might be continued until $s_k^a \neq 0$ would be found. Then, an appropriate renumeration would again reduce the task to the initial one. In the case, where all $s_i^a = 0$, $i = 1, 2, \ldots, N$, the determinant would obviously be equal to $(-\frac{qs}{\kappa})^N$, as the matrix $\mathbf{A}$ would disappear. But also all $\Sigma_k (k = 1, \ldots, N)$ would vanish, so thesis (2.14) would hold true. Thus, our assumption $s_1^a \neq 0$ does not affect the generality of the proof.

Let us perform the following operations on rows of the determinant. For each $k = 2, 3, \ldots, N$, subtract the first row divided by s_1^a from the k-th row, and multiply it by s_k^a. Then, divide each of these rows by $\frac{q \sin q}{\kappa}$. Making use of the identity

$$s_i^a s_k^b - s_k^a s_i^b = s \cdot \sin q(y_i - y_k),$$

and denoting $s_{kl} = s\frac{\kappa}{q} \sin q(y_l - y_k)$, we can rewrite the determinant as follows:

$$\left(\frac{q \sin q}{\kappa}\right)^{N-1} \begin{vmatrix} s_1^a s_1^b - \frac{qs}{\kappa} & s_1^a s_2^b & s_1^a s_3^b & \cdots & s_1^a s_N^b \\ s_{21} + \frac{s_2^a}{s_1^a} & -1 & 0 & \cdots & 0 \\ s_{31} + \frac{s_3^a}{s_1^a} & s_{32} & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{N1} + \frac{s_N^a}{s_1^a} & s_{N2} & s_{N3} & \cdots & -1 \end{vmatrix}.$$

Now, we use the values -1 on the diagonal to eliminate all elements below them, starting from the third row and ending on the N-th row. Finally, again using the diagonal elements of the rows 2 up to N, we eliminate the elements in columns $2, 3, \ldots, N$ in the first row. The remaining non-zero elements are only those on the diagonal and in the first column. The first-column-elements in rows $k = 2, 3, \ldots, N$ we denote by $\tilde{d}_k$. Since the determinant becomes triangular, its value is equal to the product of $(-\frac{qs}{\kappa})^{N-1}$ and the value of the element in the first row and first column. It remains to show that this element takes the value D_N.

Let us analyze what elements of the first column. Clearly, the element in the second row remains equal to $s_{21} + \frac{s_2^a}{s_1^a} = \tilde{d}_2$. The element in the third row, $\tilde{d}_3$, is given by $\tilde{d}_3 = s_{31} + \frac{s_3^a}{s_1^a} + s_{32}\tilde{d}_2$. Similarly, $\tilde{d}_k = s_{k1} + \frac{s_k^a}{s_1^a} + \sum_{j=2}^{k-1} s_{kj}\tilde{d}_j$, for $k = 3, \ldots, N$.

Finally, the first element is equal to $s_1^a s_1^b - \frac{q^s}{\kappa} + \sum_{j=2}^{N} s_1^a s_j^b \tilde{d}_j =: \overline{D}_N$. Denote $\tilde{d}_1 = 1$, $\overline{d}_k = s_1^a \tilde{d}$. We shall show that $\overline{D}_N = D_N$, and $\overline{d}_k = d_k$. Obviously, $\tilde{d}_1 = s_1^a = d_1$. For $\overline{d}_k$ we have:

$$
\begin{aligned}
\overline{d}_k = s_1^a \tilde{d}_k &= s_1^a (s_{k1} + \frac{s_k^a}{s_1^a} + \sum_{j=1}^{k-1} s_{kj} \cdot \tilde{d}_j) \\
&= s_1^a s_{k1} + s_k^a + \sum_{j=2}^{k-1} s_{kj} (s_1^a \tilde{d}_j) \\
&= s_k^a + s_{k1}\overline{d}_1 + \sum_{j=2}^{k-1} s_{kj}\overline{d}_j \\
&= s_k^a + \sum_{j=1}^{k-1} s_{kj}\overline{d}_j \\
&= s_k^a + \frac{\kappa}{q} \sum_{j=1}^{k-1} \overline{d}_j \sin q(y_j - y_k) \\
&= s_k^a + \frac{\kappa}{q} \sum_{j=1}^{k-1} \overline{d}_j (s_j^a c_k^a - s_k^a c_j^a) \\
&= s_k^a + \frac{\kappa}{q} (c_k^a \sum_{j=1}^{k-1} \overline{d}_j s_j^a - s_k^a \sum_{j=1}^{k-1} \overline{d}_j c_j^a) \\
&= s_k^a + \frac{\kappa}{q} (c_k^a r_{k-1}^1 - s_k^a r_{k-1}^2) = d_k
\end{aligned}
$$

From the definition (2.13) $\Sigma_k = \sum_{j=1}^{k} s_k^b d_k$, and $\overline{d}_k = d_k$. Therefore,

$$
d_N = \sum_{j=1}^{N} s_k^b d_k - \frac{q}{\kappa} \sin q = \overline{D}_N,
$$

and equality (2.14) holds true.

Now, suppose that $D_N = 0$. To transform the determinant we have been performing only operations on its rows. Applying the same operations to the rows of the matrix equation (2.11), one arrives at the following, equivalent equation, which can easily be solved for the vector $\mathbf{V}$:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ \tilde{d}_2 & -1 & 0 & 0 & \cdots & 0 \\ \tilde{d}_3 & 0 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{d}_N & 0 & 0 & 0 & \cdots & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_N \end{bmatrix} = \mathbf{0}$$

Letting $v_1 = s_1^a = d_1$, we calculate $v_k = s_1^a \tilde{d}_K = d_k$, for $k = 2, 3, \ldots, N$. This proves the theorem.

Note that the iterative definitions (2.13) permit us to calculate the determinant and the eigenvector $\mathbf{V}$ iteratively, that is, the procedure is of order (N^1), although the matrix $\mathbf{A}_N$ may not be sparse: all its elements may be different from zero !

Having calculated the values v_k we can use the formula (2.9) to represent the eigenform corresponding to the vector $\mathbf{V} = [v_1, \ldots, v_N]^T$. The eigenforms obtained this way are, in general, not normalized. To find one root q of the determinant D_N, the value of function $D_N(q)$ has to be calculated many times. Normalization is being performed only after a root of the determinant has already been found. This means, that normalization is performed much more seldom than the calculation of the determinant. Therefore, one might expect that its importance for the calculation time is not that critical, the more, that the number of spacers, N, is not large in our application. However, employing nonnormalized eigenvectors would multiply calculations of the mean square forces, which, for the number of modes not exceeding 100, say, are more time-consuming than solving the eigenvalue problem. Therefore, also the normalization process was optimized. The resulting optimization algorithm may also be useful for other problems, where the number of concentrated forces acting on a string is larger than in the present application. The (N^1)-order algorithm for normalization is as follows.

Let $v(y)$ be an eigenvector represented by its nodal points via equality (2.9). Then, the normalized eigenvector $V(y)$ is obtained from $v(y)$ by dividing the latter one by a normalizing constant, C_{nor} :

$$V(y) = \frac{1}{C_{nor}} v(y), \tag{2.15}$$

where the normalizing constant is defined as:

$$C_{nor} = \int_0^1 v^2(y) dy. \tag{2.16}$$

To calculate the constant the following quantities can be determined iteratively:

$$
\begin{aligned}
&d_i^a = d_i \cdot s_i^a, && d_i^b = d_i \cdot s_i^b,\\
&C_i = s_i^b c_i^a - c_i^b s_i^a,\\
&C_i^a = s_i^a c_i^a, && C_i^b = s_i^b c_i^b,\\
&S_1^1 = (d_1^b)2(qy_1 - C_1^a) + (d_1^a)2(q - qy_1 - C_1^b),\\
&S_1^2 = S_1^3 = 0,\\
&r_1^1 = d_1^b(qy_1 - C_1^a), && r_1^2 = d_1^a qy_1\\
&r_1^3 = d_1^a, && r_1^4 = d_1^a C_1,\\
&S_i^1 = S_{i-1}^1 + (d_i^b)^2(qy_i - C_i^a) + (d_i^a)^2(yq - qy_i - C_i^b)\\
&S_i^2 = S_{i-1}^2 + d_i^b\left[2r_{i-1}^1 + 2c(r_{i-1}^2 - qy_i r_{i-1}^3) - C_i r_{i-1}^3 + r_{i-1}^4\right]\\
&S_i^3 = S_i^3 + d_i^a(q - qy_i - C_i^t)r_{i-1}^3\\
&r_i^1 = r_{i-1}^1 + d_i^b(qy_i - C_i^a), && r_i^2 = r_{i-1}^1 + d_i^a qy_i,\\
&r_i^3 = r_{i-1}^3 + d_i^a, && r_i^4 = r_{i-1}^4 + d_i^a C_i.
\end{aligned}
\tag{2.17}
$$

Theorem 2. $C_{nor} = \frac{1}{2q}(S_N^1 + S_N^2 + 2S_N^3)$

PROOF: From (2.9), (2.16) and Theorem 1 we derive:

$$
\begin{aligned}
C_{nor} &= \int_0^1 \left[\sum_{y_i \le y} d_i \sin qy_i \sin(q - qy) + \sum_{y_j > y} d_j \sin qy \sin(q - qy_j)\right]^2 dy\\
&= \sum_{i=1}^{N} d_i^2 \left[(s_i^b)^2 \int_0^{y_i} \sin^2 qy\, dy + (c_i^a)^2 \int_{y_i}^1 \sin^2(q - qy) dy\right]\\
&\quad + 2\sum_{i=2}^{N}\sum_{j=1}^{i-1} d_i d_j \left[s_i^b s_j^b \int_0^{y_j} \sin^2 qy\, dy + s_i^b s_j^a \int_{y_j}^{y_i} \sin qy \sin q(1-y)\, dy\right.\\
&\quad \left. + s_i^a s_j^a \int_{y_i}^1 \sin^2(q - qy)\, dy\right].
\end{aligned}
\tag{2.18}
$$

Using the induction principle, definitions (2.16), (2.17), and substituting the following closed formulae for all types of integrals in (2.17):

$$
\begin{aligned}
\int_0^{y_i} \sin^2 qy\, dy &= \frac{1}{2q}(qy_i - s_i^a c_i^a),\\
\int_{y_j}^{y_i} \sin qy \sin(q - qy)\, dy &= \frac{1}{2q}\left[q(y_j - y_i)c - \frac{1}{2}(s_i^b c_i^a - c_i^b s_i^a - s_j^b c_j^a + c_j^b s_j^a)\right]\\
\int_{y_i}^1 \sin^2 q(1-y)\, dy &= \frac{1}{2q}(q - qy_i - s_i^b c_j^b),
\end{aligned}
$$

and, finally, grouping respective expressions under the summation signs, one arrives at the thesis.

Normalizing the eigenforms corresponding to the eigenvalues of the form $q = k\pi$ (if such eigenvalues do exist for the particular value of κ and positions $(y_1, \ldots, y_N)$ does not constitute any difficulty, as $\int_0^1 \sin^2 k\pi y dy = \frac{1}{2}$. Let us also remind that the eigenforms corresponding to different eigenvalues are mutually orthogonal, as the operator B is self-adjoint.

The search for zeros of the characteristic determinant is performed numerically by scanning the half-axis $q \in (0, \infty)$ in the growing direction with a scanning step being automatically adjusted. Once the zero has been embraced, the root is being successfully approximated, up to the prescribed precision. Usually, the achievable precision is of order $10^{-30} - 10^{-40}$.

3 Approximate solution to the governing equation

Having calculated the sequence of eigenvalues (q_k), and the respective normalized eigenfunctions, $V_k(y)$, represented by the nodal values

$$\mathbf{V}^k = \left[V_1^k, \ldots, V_N^k\right]^T := \frac{1}{C_{nor}^{(k)}} \left[v_1^k, \ldots, v_N^k\right]^T, \quad k = 1, 2, \ldots, M,$$

we approximate the solution, w, of the Cauchy problem,

$$\begin{aligned} &w_{ss} - w_{yy} + \sum_{i=1}^{N} \delta(y - y_i)\left[\kappa w + \beta w_s\right] + d_{str} w_s = z, \\ &w(0, s) = w(1, s) = w(y, s_0) = w_0(y, s_0) \equiv 0, \quad s \geq s_0,\ 0 \leq y \leq 1, \end{aligned} \tag{3.1}$$

by the series

$$\sum_{k=1}^{M} \zeta_k(s)\, V_k(y) =: w^M(s, y). \tag{3.2}$$

Substituting w^M into equation (3.1) in place of w, multiplying both sides by the function $V_j(y)$ and integrating both sides with respect to the spatial variable, y, we obtain the following system of equations for the temporal Fourier coefficients, ζ:

$$\ddot{\zeta}_j + \sum_{k=1}^{M} \tilde{\rho}_{jk}\dot{\zeta}_k + d_{str}\dot{\zeta}_j + q_j^2\zeta_j = z_j, \qquad \zeta_j(s_0) = \dot{\zeta}_j(s_0) = 0, \tag{3.3}$$

where

$$z_j(s) = \int_0^1 V_j(y)\, z(s,y)\, ds, \tag{3.4}$$

$$\tilde{\rho}_{jk} = \beta \sum_{i=1}^{N} v_i^j v_i^k. \tag{3.5}$$

In arriving at (3.3) the orthonormality of the eigenforms $V^k(y)$ was utilized:

$$\int_0^1 V^k(y)\, V^j(y)\, dy = \delta_{jk},$$

(here δ_{jk} is the Kronecker delta).

Note the way in which the parameter d_{str} enters the system (3.3). Typically, the amount of damping provided by spacers to damp the mode j, $\beta\rho_{jj}\dot{\zeta}_j$, is much bigger than the structural damping, $d_{str}\dot{\zeta}_j$. For those eigenforms, however, which are associated with the eigenform having poles at the spacers' clamps, that is for which all $v_i = 0, i = 1, \ldots, N$, the damping provided by the spacers is equal to zero. In this case the only damping in the system is just the structural damping $d_{str}\dot{\zeta}_j$. Therefore, it becomes critical with regard to the level of vibration and, consequently, the lifetime of the conductor.

As usual, when the method of separation of variables is used to solve a wave equation, the question of the convergence of the series (3.2) arises. The presence of damping complicates the analysis in the present case. Letting M go to infinity we arrive at an infinite system of coupled equations for the temporal coefficients, ζ_j, instead of the finite system (3.3). The structure of the damping operator, corresponding to the coefficients ρ_{jk}, is, in general, not easy to analyze, as the mutual relations between the nodal values of the eigenforms V^j depend on many factors and are difficult to predict or estimate. On the other hand, however, as it was pointed out in the introduction, there are strong indications to use the finite approximation (3.2). Firstly, although in first approximation we use a viscous model for damping in the spacers, in practice, the higher-frequency mechanical vibrations are damped to the extent allowing one to neglect their influence on the reaction forces at clamps. Secondly, the overwhelming majority of the energy imparted into a bundle by the aeolian excitation is related with the frequencies between $5 - 60Hz$, which corresponds with the first hundred eigenforms. Thus, provided the coupling with

the higher-frequency modes is not significant, one might restrict the considerations to this region of the spectrum. Actually, the lack of such coupling was observed during all calculations (see discussion of the results in the latter chapters).

Even if restricted to the first M modes, the solution sought in the form (3.2) is not easy to obtain. At least formally, the damping matrix

$$\mathbf{D} = \{\tilde{\rho}_{jk} + d_{str}\delta_{jk}\}\,, \tag{3.6}$$

has most elements different from zero. Therefore, as the number of modes, M, is large, only an approximate solution can be achieved. In what follows we discuss three methods of approximating the mean-square strains, all of which are based upon the assumption that the diagonal elements of matrix $\mathbf{D}$ dominate the non-diagonal ones. Again, this assumption was verified during the calculations performed for realistic values of parameters. In all cases the magnitude of non-diagonal elements was by several orders less than that of the diagonal ones.

For notational convenience let us define:

$$\rho_{jk} = \frac{1}{2}(\tilde{\rho}_{jj} + d_{str})\delta_{jk} + \tilde{\rho}_{jk}(1 - \delta_{jk}). \tag{3.7}$$

This denotation does not express any distinction between the two mechanisms of damping, which is so important with respect to the spacers' placement. However, after we had discussed this issue so thoroughly, we can, hopefully use this denotation bearing the preceding discussion in mind.

One way to approximate the solution of (3.3) is as follows. Let us rearrange the damping terms:

$$\ddot{\zeta}_j + 2\rho_{jj}\dot{\zeta}_j + q_j^2\zeta_j = j_j - \varepsilon g_j, \tag{3.8}$$

where

$$g_j = \sum_{k\neq j} \overline{\rho}_{jk}\dot{\zeta}_k, \qquad \text{and} \qquad \overline{\rho}_{jk} = \frac{1}{\varepsilon}\rho_{jk}, \quad j \neq k. \tag{3.9}$$

The parameter ε above, can be selected as the maximum over j of the ratios of terms $\sum_{k\neq j}\rho_{jk}\dot{\zeta}_k$, and the external terms, f_j. It follows that ε is a small parameter, taking, in practical situations, the values of the order $10^{-5} - 10^{-10}$. This result, however, was difficult to predict. Therefore, although we had assumed from the beginning that ε is small, neglecting the non-diagonal terms had not been justified before the experimental part of the study had been accomplished. The conjecture that the coupling terms are negligible is one of the important results, but at this point of analysis, the coupling has to be taken into account in order to facilitate the comparison.

Denote

$$\Delta_k = \sqrt{q_k^2 - \rho_{kk}^2} \tag{3.10}$$

(for big amounts of damping, and for the first few eigenfrequencies, the expression under the square root sign is negative and, consequently, Δ_k is imaginary).

Using the homogeneous initial conditions, we can express the solution, ζ_k, in the following form.

$$\zeta_k(s) = \frac{1}{\Delta_k} \int_{s_0}^{s} e^{-\rho_{kk}(s-\tau)} \sin \Delta_k(s-\tau)(f_k(\tau) - \varepsilon g_k(\tau))\, d\tau \tag{3.11}$$

Differentiating this expression with respect to time we obtain:

$$\dot{\zeta}_k(s) = \int_{s_0}^{s} e^{-\rho_{kk}(s-\tau)} \left[\cos \Delta_k(s-\tau) - \frac{\rho_{kk}}{\Delta_{kk}} \sin \Delta_k(s-\tau)\right] (f_k(\tau) - \varepsilon g_k(\tau))\, d\tau.$$

This expression can be substituted into the definition of the function g_j. The resulting formula, in turn, can be substituted into (3.11), yielding the following approximation of the solution of system (3.3) in terms of external excitation:

$$\begin{aligned} \zeta_k = &\int_{s_0}^{s} f_k(\tau)\varphi_{kk}(s-\tau)\, d\tau \\ &+ \varepsilon \sum_{j \neq k} \int_{s_0}^{s} f_j(\tau)\tilde{\varphi}_{kj}(s-\tau) d\tau \\ &+ \varepsilon^2 \sum_{j \neq k} \int_{s_0}^{s} g_j(\tau)\tilde{\varphi}_{kj}(s-\tau) d\tau. \end{aligned} \tag{3.12}$$

Note that the sum of integral kernels from the first two lines of (3.12) can be regarded as an approximation of the Green's function for the system (3.3). The kernels, φ_{kj}, have the following forms:

$$\begin{aligned} &\varphi_{kk}(s) = \frac{1}{\Delta_k} e^{-\rho_{kk}s} \sin \Delta_k s, \qquad \tilde{\varphi}_{kj} = \frac{1}{\varepsilon}\varphi_{kj}, \\ &\varphi_{kj}(s-\tau) = \rho_{kj} e^{\rho_{jj}s - \rho_{kk}\tau} \int_{\tau}^{s} e^{-(\rho_{jj}-\rho_{kk})r} \sin \Delta_k(s-r) \\ &\qquad \times \left[\cos \Delta_j(r-\tau) - \frac{\rho_{jj}}{\Delta_j} \sin \Delta_j(r-\tau)\right] dr. \end{aligned} \tag{3.13}$$

Due to the assumption concerning the parameter ε, the kernels are of the same order of magnitude. This means that the order of magnitude of consecutive terms is determined by the power to which the small parameter ε is raised in the particular term. Of course the procedure can be repeatedly iterated, yielding better and better approximations of the solution. However, the terms obtained even by the first order approximation are rather complicated. It will be seen in the next section that the calculation of the mean-square strains based on the above first order approximation is practically impossible for large

numbers of modes. Therefore, the above method of approximation was used only as a reference method, for a small number of modes, M , and under the additional assumption that damping is very small. In this case the term $\frac{\rho_{ii}}{\Delta_j}$ can be neglected, and the formulae simplify significantly.

A somewhat more efficient numerical method is based on the spectral approach (chapter V). This method might be related to the approximation of the solution of the system (3.3) by means of the Laplace-transform approach. However, since the method of calculation of mean-square values does not really require solving the equation (3.3), we omit discussing the latter problem.

4 Calculation of mean-square forces using Green's function approximation

Let us remind that the forces at the edge points, $0 = y_0 < y_1 < \ldots < y_N < y_{N+1} = 1$, are calculated as

$$F_i = w_y^M|_{y=y_i^-} - w_y^M|_{y=y_i^+}.$$

Using (3.2) and (1.17) we obtain

$$F_i(s) = \sum_{k=1}^{M} \zeta_k(s) p_k^i, \tag{4.1}$$

where

$$\begin{aligned} p_k^0 &= -q_k \sum_{j=1}^{N} \sin(q_k - q_k y_j)\, V_j^k. \\ p_k^i &= q_k \sin q_k\ V_i^k, \qquad i = 1, 2, \ldots, N, \\ p_k^{N+1} &= q_k \sum_{j=1}^{N} \sin q_k y_j\ V_j^k, \end{aligned} \tag{4.2}$$

Using formula (4.1) and approximation (3.12), one can represent the forces as follows:

$$F_i(s) = \sum_{k=1}^{M} \zeta_k(s) p_k^i = \sum_{k=1}^{M} \sum_{n=1}^{M} p_k^i \int_{s_0}^{s} d\tau \int_0^1 V^n(y) \varphi_{kn}(s-\tau) z(y,\tau) dy.$$

Thus, the mean-square forces can be expressed as:

$$EF_i^2(s) = \sum_{j,n=1}^{M} R_{jn} \sum_{k,m=1}^{M} V_i^k V_i^m Q_{jknm}, \tag{4.3}$$

where

$$R_{jn} = \int_0^1 dx \int_0^1 V^j(x) V^n(y) K_y(y-x) dy, \tag{4.4}$$

and

$$Q_{jknm} = \int_{s_0}^{s} d\tau \int_{s_0}^{s} \varphi_{kn}(s-\tau) \varphi_{mj}(s-t) K_s(t-\tau) d\tau. \tag{4.5}$$

Since we are interested in *stationary* values of mean-square forces, we let s_0 tend to $-\infty$ in (4.5), and set $s = 0$ for simplicity. As already mentioned before, representation (4.3) is not very practical from the computational point of view. Therefore, it was used only for small values of M (up to 30) and for small damping ($\rho_{jj} \ll \Delta_j$). For such damping:

$$\varphi_{kn}(-\tau) = a_{kn} e^{\rho_{kk}\tau} \sin(\Delta_k \tau - \xi_{kn}^k) + b_{kn} e^{\rho_{nn}\tau} \sin(\Delta_n \tau - \xi_{kn}^n), \tag{4.6}$$

where

$$a_{kn} = \begin{cases} -\dfrac{1}{\Delta_k} & \text{for } k = n; \\ \dfrac{\rho_{kn}}{\Delta_k} \sqrt{\dfrac{(\rho_{kk}-\rho_{nn})^2 + \Delta_k^2}{((\rho_{kk}-\rho_{nn})^2 + \Delta_k^2 + \Delta_n^2)^2 - 4\Delta_k^2\Delta_n^2}} & \text{for } k \neq n; \end{cases}$$

$$b_{kn} = \begin{cases} 0 & \text{for } k = n; \\ \rho_{kn} \dfrac{1}{\sqrt{((\rho_{kk}-\rho_{nn})^2 + \Delta_k^2 + \Delta_n^2)^2 - 4\Delta_k^2\Delta_n^2}} & \text{for } k \neq n; \end{cases}$$

$$\xi_{kn}^k = \begin{cases} 0 & \text{for } k = n; \\ \text{arc tg} \dfrac{\Delta_k \left[(\rho_{kk}-\rho_{nn})^2 + \Delta_k^2 - \Delta_n^2\right]}{(\rho_{kk}-\rho_{nn})\left[(\rho_{kk}-\rho_{nn})^2 + \Delta_k^2 + \Delta_n^2\right]} & \text{for } k \neq n; \end{cases}$$

$$\xi_{kn}^n = \begin{cases} 0 & \text{for } k = n; \\ \text{arc tg} \dfrac{(\rho_{kk}-\rho_{nn})^2 + \Delta_k^2 - \Delta_n^2}{2\Delta_n(\rho_{kk}-\rho_{nn})} & \text{for } k \neq n; \end{cases}$$

Representation (4.6) (resulting from (3.13) by performing the analytical calculation of convolution integrals), though looking complicated, is actually quite compact, as it has the same form for all combinations k, n, and it has a minimal number of summands of the form of a product of exponential and trigonometric functions. This reduction is achieved by introducing the phase shifts ξ which can be calculated before the summations in the formula (4.3). This policy is consequently employed while deriving the formulae for Q_{knmj} .

Let us define

$$\psi(\alpha,\beta)=\begin{cases}\text{arc sin}\frac{\alpha}{\sqrt{\alpha^2+\beta^2}}, & \text{for}\quad \beta\geq 0,\\ \pi-\text{arc sin}\frac{\alpha}{\sqrt{\alpha^2+\beta^2}}, & \text{for}\quad \beta<0,\end{cases}$$

$$\begin{aligned}
\psi_{ki}^{\pm} &= \psi(\Delta_k\pm\Delta_i,\rho_{kk}+\rho_{ii}),\\
z_{ki}^{\pm} &= \left(\sqrt{(\Delta_k\pm\Delta_i)^2+(\rho_{kk}+\rho_{ii})^2}\right)^{-1},\\
\psi_i^{+\pm} &= \psi(\Delta_i+\nu,\rho_{ii}\pm\lambda),\\
\psi_i^{-\pm} &= \psi(\Delta_i-\nu,\rho_{ii}\pm\lambda),\\
z_i^{+\pm} &= (\sqrt{(\Delta_i+\nu)^2+(\rho_{ii}\pm\lambda)^2})^{-1},\\
z_i^{-\pm} &= (\sqrt{(\Delta_i-\nu)^2+(\rho_{ii}\pm\lambda)^2})^{-1}.
\end{aligned}\tag{4.7}$$

Using this notation we can express the inner integrtion in (4.5) as follows:

$$\begin{aligned}
&\int_{-\infty}^{0}\varphi_{ij}(-\tau)\,K_s(s-\tau)\,d\tau\\
&\quad=\frac{1}{2}a_{ij}\left\{e^{\lambda s}[z_i^{+-}\sin(-\xi_{ij}^i-\nu s-\psi_i^{+-})+z_i^{--}\sin(-\xi_{ij}^i+\nu s-\psi_i^{--})]\right.\\
&\qquad+e^{\rho_{ii}s}[z_i^{++}\sin(\Delta_i s-\xi_{ij}^i-\psi_i^{++})+z_i^{-+}\sin(\Delta_i s-\xi_{ij}^i-\psi_i^{-+})\\
&\qquad\left.-z_i^{-+}\sin(\Delta_i s-\xi_{ij}^i-\psi_i^{+-})-z_i^{--}\sin(\Delta_i s-\xi_{ij}^i-\psi_i^{--})]\right\}\\
&\quad+\frac{1}{2}b_{ij}\left\{e^{\lambda s}[z_j^{+-}\sin(-\xi_{ij}^j-\nu s-\psi_j^{+-})+z_j^{--}\sin(-\xi_{ij}^j+\nu s-\psi_j^{--})]\right.\\
&\qquad+e^{\rho_{jj}s}[z_j^{++}\sin(\Delta_j s-\xi_{ij}^j-\psi_j^{++})+z_j^{-+}\sin(\Delta_j s-\xi_{ij}^j-\psi_j^{-+})\\
&\qquad\left.-z_j^{+-}\sin(\Delta_i s-\xi_{ij}^j-\psi_j^{+-})-z_j^{--}\sin(\Delta_j s-\xi_{ij}-\psi_j^{--})]\right\}.
\end{aligned}$$

After performing the outer integration in (4.5) the formulae complicate further. Therefore, we introduce the abbreviation:

$$\begin{aligned}
\varphi^{ki}(\xi_1,\xi_2)=&z_i^{+-}(z_k^{++}\cos(-\xi_1+\xi_2+\psi_i^{+-}+\psi_k^{++})-z_k^{-+}\cos(-\xi_1-\xi_2-\psi_i^{+-}-\psi_k^{-+}))\\
&+z_i^{--}(z_k^{-+}\cos(-\xi_1+\xi_2+\psi_i^{--}-\psi_k^{-+})-z_k^{++}\cos(-\xi_1-\xi_2-\psi_i^{--}-\psi_k^{++}))\\
&+z_i^{++}(z_{ki}^{-}\cos(-\xi_1+\xi_2+\psi_i^{++}-\psi_{ki}^{-})-z_{ki}^{+}\cos(-\xi_1-\xi_2-\psi_i^{++}-\psi_{ki}^{+}))\\
&+z_i^{-+}(z_{ki}^{-}\cos(-\xi_1+\xi_2+\psi_i^{-+}+\psi_{ki}^{-})-z_{ki}^{+}\cos(-\xi_1-\xi_2-\psi_i^{-+}-\psi_{ki}^{+}))\\
&-z_i^{+-}(z_{ki}^{-}\cos(-\xi_1+\xi_2+\psi_i^{+-}+\psi_{ki}^{-})-z_{ki}^{+}\cos(-\xi_1-\xi_2-\psi_i^{+-}-\psi_{ki}^{+}))\\
&-z_i^{--}(z_{ki}^{-}\cos(-\xi_1+\xi_2+\psi_i^{--}+\psi_{ki}^{-})-z_{ki}^{+}\cos(-\xi_1-\xi_2-\psi_i^{--}-\psi_{ki}^{+})).
\end{aligned}$$

Now we are ready to give the formula for Q_{knij} :

$$\begin{aligned}
4Q_{knij}=&\,a_{kn}a_{ij}\varphi^{ki}(\xi_{kn}^k,\xi_{ij}^i)+b_{kn}a_{ij}\varphi^{ni}(\xi_{kn}^n,\xi_{ij}^i)\\
&+a_{kn}b_{ij}\varphi^{kj}(\xi_{kn}^k,\xi_{ij}^i)+b_{kn}b_{ij}\varphi^{nj}(\xi_{kn}^n,\xi_{ij}^i).
\end{aligned}\tag{4.8}$$

Note that the consequent utilization of angle shifts ξ and ψ results in that the additions of the angles calculated before the summation substitute the calculation of multiple cosine values and the multiplication of the results. This increases the speed of computation several times. Note also that the presence of the z-type factors in the above formulae enhances the significance of the structural damping. Indeed, because d_{str} adds to ρ_{kk} the constants $z_{ki}^{\pm}$ are well defined, even if the spacers are positioned at the nodal points of the eigenforms. For $d_{str} \to 0$ some of z_{ki}-s may diverge to infinity, resulting in a blow-up of MSQ forces.

Let us proceed to the calculation of the spatial Fourier coefficients R_{jn}. Although definition (4.3) is quite simple, the task is rather complicated because the eigenforms $V^k(y)$ are defined via different formulae on different subintervals of the domain $(0;1)$. We divide the region of integration $\{(x,y)|\, 0 < x < 1, 0 < y < 1\}$ into $(N+1)(N+2)$ disjoint subsets:

$$
\begin{aligned}
D_i^x &= \{(x,y)|\, y_i < x < y_{i+1}, y_i < y < x\}, && i = 0,1,\ldots,N,\\
D_i^y &= \{(x,y)|\, y_i < y < y_{i+1}, y_i < x < y\}, && i = 0,1,\ldots,N,\\
D_{in} &= \{(x,y)|\, y_i < x < y_{i+1}, y_n < y < y_{n+1}\}, && i,n = 0,1,\ldots,N; \quad i \neq n.
\end{aligned}
$$

Define:

$$
\overline{S}_i^j = \sum_{a=i}^{N} v_a^j \sin(q_j - q_j y_a), \qquad \overline{S}_{N+1}^j = 0,
$$

$$
\overline{S}_i^j = \sum_{a=1}^{i} v_a^j \sin q_j y_a, \qquad \overline{S}_0^j = 0.
$$

From definition (2.9), on the interval $y_i < y < y_{i+1} \quad (i = 0,1,\ldots,N)$, the function V^j can be represented as

$$
\begin{aligned}
V^j(y) &= \overline{S}_{i+1}^j \sin q_j y + s_i^j \sin(q_j - q_j y)\\
&= z_i^j \sin(q_i y + \psi_i^j),
\end{aligned}
\tag{4.9}
$$

where

$$
z_i^j = \sqrt{(\overline{S}_{i+1}^j - S_i^j \cos q_j)^2 + (S_i^j \sin q_j)^2},
$$

and the angle shift ψ_i^j is defined by the following two conditions:

$$
\begin{aligned}
\cos\psi_i^j &= \overline{S}_{i+1}^j - S_i^j \cos q_j,\\
\sin\psi_i^j &= S_i^j \sin q_j.
\end{aligned}
$$

For $i > n$, the contribution to the integral

$$
R_{jk} = \int_0^1 dy \int_0^1 V^j(x) V^k(y) K_y(x-y) dx,
$$

resulting from the integration over the subdomain D_{in} is equal to

$$\int_{y_i}^{y_{i+1}} dx \int_{y_n}^{y_{n+1}} dy \left[e^{-\alpha|x-y|} \cos\mu(x-y)\, z_i^j\, z_n^k\, \sin(q_j x + \psi_i^j)\, \sin(q_k y + \psi_n^k) \right]$$
$$= z_i^j z_n^k \left\{ \int_{y_i}^{y_{i+1}} e^{-\alpha x} \cos\mu x\, \sin(q_j x + \psi_i^j)\, dx \int_{y_n}^{y_{n+1}} e^{\alpha y} \cos\mu y\, \sin(q_k y + \psi_n^k)\, dy \right. \quad (4.10)$$
$$\left. + \int_{y_i}^{y_{i+1}} e^{-\alpha x} \sin\mu x\, \sin(q_j x + \psi_i^j)\, dx \int_{y_n}^{y_{n+1}} e^{\alpha y} \sin\mu y\, \sin(q_k y + \psi_n^k)\, dy \right\}.$$

For $i < n$ the above formula requires only the change of signs in the exponents under the integrals. Similarly, the contribution related to the subdomain D_i^x can be calculated as follows.

$$z_i^k z_i^j \left\{ \int_{y_i}^{y_{i+1}} e^{-\alpha x} \cos\mu x \sin(q_k x + \psi_i^k) [\int_{y_i}^{x} e^{\alpha y} \cos\mu y\, \sin(q_j y + \psi_i^j)\, dy]\, dx \right.$$
$$\left. + \int_{y_i}^{y_{i+1}} e^{-\alpha x} \sin\mu x \sin(q_k x + \psi_i^k) [\int_{y_i}^{x} e^{\alpha y} \sin\mu y\, \sin(q_j y + \psi_i^j)\, dy]\, dx \right\}$$
$$= \frac{z_i^k z_i^j}{4} \left\{ \int_{y_i}^{y_{i+1}} [\sin((\mu+q_k)x + \psi_i^k) - \sin((\mu-q_k)x - \psi_i^k)] \right.$$
$$\times [z_-^j \cos((\mu-q_j)x - \psi_i^j + \psi_-^j) - z_+^j \cos((\mu+q_j)x + \psi_i^j + \psi_+^j)]\, dx$$
$$- \left[z_+^j \cos((\mu+q_j)y_i + \psi_i^j + \psi_+^j) - z_-^j \cos((\mu-q_j)y_i - \psi_i^j + \psi_-^j) \right]$$
$$\times \left[e^{-\alpha(y_{i+1}-y_i)} \left(z_+^k \cos((\mu+q_k)y_{i+1} + \psi_i^k - \psi_+^k) \right.\right.$$
$$\left. - z_-^k \cos((\mu-q_k)y_{i+1} - \psi_i^k - \psi_-^k) \right)$$
$$\left. - \left(z_+^k \cos((\mu+q_k)y_i + \psi_i^k - \psi_+^k) - z_-^k \cos((\mu-q_k)y_i - \psi_i^k - \psi_-^k) \right) \right]$$
$$+ \int_{y_i}^{y_{i+1}} [\cos((\mu-q_k)x - \psi_i^k) - \cos((\mu+q_k)x + \psi_i^k)]$$
$$\times [z_-^j \sin((\mu-q_j)x - \psi_i^j + \psi_-^j) - z_+^j \sin((\mu+q_j)x + \psi_i^j + \psi_+^j)]\, dx$$
$$- \left[z_-^j \sin((\mu-q_j)y_i - \psi_i^j - \psi_-^j) - z_+^j \sin((\mu+q_i)y_i + \psi_i^j + \psi_+^j) \right]$$
$$\times \left[e^{-\alpha(y_{i+1}-y_i)} \left(z_-^k \sin((\mu-q_k)y_{i+1} - \psi_i^k - \psi_-^k) \right.\right.$$
$$\left. - z_+^k \sin((\mu+q_k)y_{i+1} + \psi_i^k - \psi_+^k) \right)$$
$$\left. \left. - \left(z_-^k \sin((\mu-q_k)y_i - \psi_i^k - \psi_-^k) - z_+^k \sin((\mu+q_k)y_i + \psi_i^k - \psi_+^k) \right) \right] \right\}. \quad (4.11)$$

In a similar way as before, the multipliers $z_\pm^{\cdot}$ and phase shifts $\psi_\pm^{\cdot}$ are defined by the formulae:

$$\sqrt{\alpha^2 + (\mu \pm q_\cdot)^2} =: z_\pm^{\cdot}, \qquad \alpha = z_\pm^{\cdot} \sin\psi_\pm^{\cdot}, \qquad \mu \pm q_\cdot = z_\pm^{\cdot} \cos\psi_\pm^{\cdot}.$$

Most terms in the above lengthy formula are explicit. The integrals can also be calculated analytically. The result is a combination of 64 trigonometric expressions. A careful analysis

of relations between their arguments allows the reduction of similar addends. Finally, the considered part of the contribution to R_{jk} reduces to the following component:

$$z_-^j \left(c_i(q_k + q_j;\, \psi_k^k + \psi_k^j - \psi_-^j) - c_i(q_j - q_k;\, \psi_k^j - \psi_i^k - \psi_-^j) \right) + z_+^j \left(c_i(-q_k - q_j;\, -\psi_i^k - \psi_k^j - \psi_+^j) - c_i(q_k - q_j;\, \psi_i^k - \psi_i^j - \psi_+^j) \right), \qquad (4.12)$$

where

$$c_i(a;b) = \begin{cases} (y_{i+1} - y_i)\sin b, & a = 0, \\ (\cos(ay_{i+1} + b) - \cos(ay_i + b))/a, & a \neq 0. \end{cases}$$

Substitution of (4.12) in place of integrals in (4.11), yields the contribution to R_{kj} from the subdomain D_i^x. Similarly, the contribution from D_i^y can be calculated. We do not provide explicit expressions for integrals in the representation (4.10) for the contribution from D_{in} because the integrals are particular cases of those entering (4.11). Summing up the contributions from all subdomains D_i^x, D_i^y, D_{in} we obtain the value $R_{kj} = R_{jk}$.

It is worth noting that matrix $R = \{R_{kj}\}$ is not only symmetric and real, but it is positive definite as well, as it can be considered a Gramm matrix for the system of eigenforms, V^k with the product defined by (4.4). (Because K_s is a covariance function, (4.4) is a well-defined scalar product). The eigenforms, V^k, are mutually orthogonal with respect to a different inner product:

$$\int_0^1 V^k(y)\, V^j(y)\, dy.$$

Since these two inner products are different, the orthogonality with respect to the product (4.4) ist not, in general, preserved. Therefore, matrix R is not necessarily diagonal. This means that the excitations

$$z_j(s) = \int_0^1 V^j(y)\, z(y,s)\, dy$$

related to different modes of vibration may, in general, be statistically correlated with each other. This is natural, as all those excitation processes result from one random field. On the other hand, however, a stochastic process can be represented as a combination of statistically independent random harmonics or other, more general random functions. The well-known spectral representations, or orthogonal representations, like the Karhunen-Loeve representation for stochastic processes, say, exemplify that. It is therefore of interest, to know the degree of the correlation between different $z_j(s)$. The calculations which we have conducted for realistic values of parameters, show very weak correlations. For N almost equidistantly distributed spacers, the eigenforms and eigenvalues usually form groups of N eigenvalues which differ only slightly within a group, and which are essentially different from those of other groups. Processes related to the eigenforms of one group happen to be little correlated with each other, whereas the correlation between the excitation processes z_j, z_k related to different groups of eigenfrequencies are usually almost uncorrelated (the correlation is of the order 10^{-6}—10^{-10}).

As already mentioned, the above approach is of relatively little use because of the complication of formulae (4.3) and (4.8). The matrix R, however, will also be utilized in the spectral approach presented in the next chapter.

5 Spectral approach in approximation of the mean-square strains

Relations (3.3) and (4.1) can be rewritten as follows:

$$\mathcal{L}^T \mathcal{X} = \mathcal{Z}, \tag{5.1}$$

$$F_i = (P^i)^T \mathcal{X}. \tag{5.2}$$

Here

$$\mathcal{Z} = [z_1, z_2, \ldots, z_M]$$

(recall that $z_j(s) = \int_0^1 v^j(y) z(y,s) dy$) is the vector-valued stochastic process of external excitation),

$$\mathcal{L}(\frac{d}{dt}) = I_M \frac{d^2}{dt^2} + D\frac{d}{dt} + diag\,\{q_i^2\} \tag{5.3}$$

is a second-order matrix-valued differential operator,

$$\mathcal{X} = [\zeta_1, \zeta_2, \ldots, \zeta_M]^T$$

is a vector of temporal Fourier coefficients of the solution, w, of the governing wave equation; at the same time, however, it can be considered as the response of the linear second-order system (5.1) to the stochastic excitation $\mathcal{Z}$.

Finally,

$$(P^i)^T = [p_1^i, p_2^i, \ldots, p_M^i]$$

(see (4.2)).

Summing up, the force F_i is a result of the action of two consecutive linear operations on the stationary stochastic process $\mathcal{Z}$. It is well-known that in this case the transfer-functions of the linear operations relate the spectra of the input and the output processes. Given the spectrum of a stationary stochastic process of second order, one can calculate

the variance using the known reciprocal relation. Denote the covariance function of the process F_i by K_{F_i}, that is, define

$$K_{F_i}(\Delta s) = \mathbf{E}\, F_i(s) F_i(s + \Delta s),$$

where $\mathbf{E}$ is, as usual, the expected value operator. The following relations hold true:

$$S_{F_i} = \int_{-\infty}^{\infty} e^{-i\omega s} K_{F_i}(s) ds$$
$$K_{F_i} = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega s} S_{F_i}(\omega) d\omega.$$

Since the stationary mean-square value $\sigma_i^2 = K_{F_i}(0)$, one has

$$\sigma_i^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{F_i}(\omega) d\omega. \tag{5.4}$$

To use formula (5.4) we need to express the spectrum S_{F_i} in terms of known quantities. What we know is the covariance function of excitation process $\mathcal{Z}$, and the relations (5.1) and (5.2), transforming the excitation into the forces F_i. First, we derive the expression for the spectral density matrix, $S_Z = \{S_Z^{kj}\}$, of the vector of the modal excitation processes, $\mathcal{Z}$, appearing in (5.1):

$$\begin{aligned}
S_Z^{kj}(\omega) &= \int_{-\infty}^{\infty} e^{-i\omega s} \mathbf{E}[z_k(\tau) z_j(\tau + s)]\, ds \\
&= \int_{-\infty}^{\infty} e^{-i\omega s} \left[\int_0^1 V^k(x) z(x, \tau)\, dx \int_0^1 V^j(y) z(y, \tau + s)\, dy \right] ds \\
&= \int_0^1 dx \int_0^1 dy \left[V^k(x) V^j(y) \int_{-\infty}^{\infty} e^{-i\omega s} \mathbf{E}[z(x, \tau)\, z(y, \tau + s)]\, ds \right] \\
&= \int_0^1 dx \int_0^1 dy \left[V^k(x) V^j(y) \int_{-\infty}^{\infty} e^{-i\omega s} K_y(x - y)\, K_s(s)\, ds \right] \\
&= \int_0^1 dx \int_0^1 V^k(x) V^j(y) K_y(x - y) dy \int_{-\infty}^{\infty} e^{-i\omega s} K_s(s) ds \\
&= R_{kj} S_z(\omega).
\end{aligned} \tag{5.5}$$

In R_{jk} we recognize the matrix (4.4), whose elements can be calculated by the methods explained in the previous chapter. $S_z(\omega)$ in (5.4) is defined as the Fourier transform of the temporal factor, K_s, of the correlation function of the external excitation.

Since we assume this function to be of the form (1.20), $S_z(\omega)$ can be calculated as follows:

$$\begin{aligned} S_z(\omega) &= \int_{-\infty}^{\infty} e^{-i\omega\tau} K_s(\tau) d\tau \\ &= \int_{-\infty}^{\infty} e^{-i\omega\tau} e^{-\lambda|\tau|} \cos\nu\tau \, d\tau \\ &= 2Re \int_0^{\infty} e^{-\omega\tau} \frac{1}{2} \left[e^{-\lambda\tau - i\nu\tau} + e^{-\lambda + i\nu\tau} \right] d\tau \\ &= 2\lambda \frac{\lambda^2 + \omega^2 + \nu^2}{(\lambda^2 + \omega^2 + \nu^2)^2 - 4\omega^2\nu^2}. \end{aligned} \tag{5.6}$$

Combining results of chapter 4 with equations (5.5) and (5.6) we obtain the spectrum $S_{\mathcal{Z}}$ of the input to the relation (5.2). Thus, to use (5.4) for the sought variances, it remains to characterize the relation between the input spectrum $S_{\mathcal{Z}}$ and the output spectrum S_{F_i}. From the definition of a correlation function, and from its relation with a power spectral density function one easily obtains that, for linearly dependent (through equation (5.2)) processes F_i and $\mathcal{X}$, the following equality holds true.

$$S_{F_i}(\omega) = (P^i)^T S_{\mathcal{X}}(\omega) P^i \tag{5.7}$$

Next, we relate the spectrum of the response, $\mathcal{X}$, to the spectrum of the input, $\mathcal{Z}$. For scalar processes the respective relation appears in almost all textbooks on stochastic dynamical systems, but for the vector-valued processes it is less known, so we indicate the main points of derivation:

$$\begin{aligned} S_{\mathcal{Z}}(\omega) &= \int_{-\infty}^{\infty} e^{-i\omega\tau} \mathbf{E}\left[\mathcal{Z}(s) \mathcal{Z}^T(s+\tau) \right] d\tau \\ &= \int_{-\infty}^{\infty} e^{-i\omega\tau} \mathbf{E}\left[\left(\mathcal{L}^T(\frac{d}{ds}) \mathcal{X} \right)(s) \left(\mathcal{L}^T(\frac{d}{dr}) \mathcal{X} \right)^T (r)_{|r=s+\tau} \right] d\tau \\ &= \int_{-\infty}^{\infty} e^{-i\omega\tau} \mathcal{L}^T(\frac{d}{ds}) \left(\mathcal{L}^T(\frac{d}{dr}) \mathbf{E}\left[\mathcal{X}(s) \mathcal{X}^T(r) \right]_{|r=s+\tau} \right)^T d\tau \\ &= \int_{-\infty}^{\infty} e^{-i\omega\tau} \mathcal{L}^T(\frac{d}{ds}) \left(\mathcal{L}^T(\frac{d}{dr}) K_{\mathcal{X}}(r-s)_{|r=s+\tau} \right)^T d\tau \\ &= \int_{-\infty}^{\infty} e^{-i\omega\tau} \overline{\mathcal{L}}^T(\frac{d}{d\tau}) \left(\mathcal{L}^T(\frac{d}{d\tau}) K_{\mathcal{X}}(\tau) \right)^T d\tau \\ &= \overline{\mathcal{L}}^T(i\omega) \left(\int_{-\infty}^{\infty} e^{-i\omega\tau} K_{\mathcal{X}}(\tau) \, d\tau \right) \mathcal{L}(i\omega) \\ &= \overline{\mathcal{L}}^T(i\omega) S_{\mathcal{X}}(\omega) \mathcal{L}(i\omega), \end{aligned}$$

where

$$\overline{\mathcal{L}}^T(\frac{d}{d\tau}) = I_M \frac{d^2}{d\tau^2} - D\frac{d}{d\tau} + diag\left\{q_i^2\right\}.$$

The above relation converts easily:

$$S_X(\omega) = \left(\overline{\mathcal{L}}^T(i\omega)\right)^{-1} S_Z(\omega)\mathcal{L}^{-1}(i\omega). \tag{5.8}$$

Combining (5.4), (5.5), (5.7) and (5.8) we obtain:

$$\sigma_i^2 = 2\pi \int_{-\infty}^{\infty} (P^i)^T(\overline{\mathcal{L}}^T(i\omega))^{-1} R S_z(\omega)\mathcal{L}^{-1}(i\omega)P^i d\omega. \tag{5.9}$$

Because of the size of the matrices $\overline{\mathcal{L}}$ and $\mathcal{L}$, direct application of formula (5.9) is practically impossible. To approximate (5.9) we use the fact that non-diagonal elements of the damping matrix are much smaller than the diagonal ones. Let us decompose the matrix $\mathcal{L}(i\omega)$ into a sum of its diagonal and non-diagonal parts

$$\mathcal{L}(i\omega) = diag\left\{a_k(\omega)\right\} + i\left\{b_{kj}(\omega)\right\} = A + iB,$$

where $a_k = (q_k^2 - \omega^2) + i\omega\rho_{kk}$, $b_{kj} = \omega\rho_{kj}$ for $k \neq j$, and $b_{kk} = 0$. Let $x = \max_{k,j} |b_{kj}|/|a_k|$. As we know, $x \leq 1$ (in fact, any case where x is comparable with 1 can be considered pathological; normally x is much less than 1). Note that $\overline{\mathcal{L}}(i\omega) = \overline{(\mathcal{L}(i\omega))} = \overline{A} - iB$ (B is real, symmetric). Denote also $\tilde{B} = \frac{1}{x}B$.

Consider a function

$$h(x) := P^T(\overline{A} - ix\tilde{B})^{-1} S_Z (A + ix\tilde{B})^{-1} P. \tag{5.10}$$

For small x, the function h can be approximated by the first few elements of its Mac Laurin expansion:

$$\begin{aligned} h(x) &\approx h(0) + h'(0)x + \frac{1}{2}h''(0)x^2 =: \tilde{h}(x) \\ &= P^T\overline{A}^{-1} S_Z A^{-1} P \\ &+ P^T i(\overline{A}B\overline{A}^{-1} S_Z A^{-1} S_Z A^{-1} BA^{-1})P \\ &+ P^T\left[\overline{A}^{-1} B\overline{A}^{-1} S_Z A^{-1} - \left(\overline{A}^{-1} B\overline{A}^{-1} B\overline{A}^{-1} S_Z A^{-1} + \overline{A}^{-1} S_Z A^{-1} BA^{-1} BA^{-1}\right)\right] P. \end{aligned} \tag{5.11}$$

The first element in expansion (5.11) will be called the "zero-order approximation", the sum of the first and second elements will be addressed to as the "first-order approximation", and the total sum is the "second-order approximation".

Matrix A is diagonal. Matrix B is real and symmetric. Matrix S_Z is real, symmetric, and positive definite. These properties allow us to eliminate part of the operations

necessary to compute the value $\tilde{h}(x)$. In particular, any matrix can be decomposed as follows:

$$C = \frac{1}{2}(C + C^T) + \frac{1}{2}(C - C^T).$$

The first addend is obviously a symmetric matrix, whereas the second one is an antisymmetric matrix. For any antisymmetric matrix $G = -G^T$ the following equality holds true: $P^T G P = 0$. Thus, to compute $P^T C P$ it suffices to compute $P^T H P$, where $H = \frac{1}{2}(C + C^T)$. The antisymmetric part of the matrix C can be neglected while computing $P^T C P$. Now, computing $P^T H P$ (for a symmetric matrix H) requires less operations than computing $P^T C P$ for a general matrix C:

$$P^T H P = \sum_{k=1}^{M} p_k^2 h_{kk} + \sum_{j=2}^{M} \sum_{k=1}^{j-1} p_j p_k h_{kj},$$

while

$$P^T C P = \sum_{k=1}^{M} p_k^2 c_{kk} + \sum_{j=1}^{M} \sum_{k=1;k\neq j}^{M} p_j p_k c_{kj}.$$

Moreover, what we are really interested in, is not the value (5.10) for each particular ω, but its integral over $\omega \in (-\infty, \infty)$. Thus we can further eliminate the antisymmetric (odd as a function of argument ω) part of each element of the matrix H.

Actually, we can show that an even part (in the sense of even function) of any element of a symmetric part (in the sense of matrix transposition) of the matrix

$$\overline{A}^{-1} B \overline{A}^{-1} S_Z A^{-1} - \overline{A}^{-1} S_Z S^{-1} B A^{-1}$$

vanishes (the calculations are easy but tedious and are omitted in this report). An important consequence of this fact is that the zero-order approximation (5.11) is identical with the first-order approximation.

Note that matrix D does not appear in the zero-order approximation. In other words, the "cross-modal-damping" terms ρ_{kj}, responsible for the mechanical coupling and transfer of energy between modes, are not taken into account while using the zero-order approximation. It is difficult to derive a practical estimate of the influence of the coupling on the mean-square forces. Therefore, to be able to study this influence, the use of the second-order approximation is necessary, as it takes the coupling into account and enables us to compare results obtained with and without coupling.

It should be stressed that, although the *mechanical* coupling (the one resulting from the terms ρ_{kj}) is neglected in the zero-order approximation, the resulting mean-square forces estimates are not equal to a sum of mean-squares of forces which would result from considering each mode separately. This is so, because there exists another coupling mechanism accounted for, even in the zero-order approximation (5.11). It is the *statistical correlation* between the portions of external excitation which apply to the individual modes.

Empirical laws established in the field of aeolian excitations show, that, in the short time-range usually there is one dominant frequency of vibrations which is being excited. The random fluctuations, however, result always in some spread of excitation energy, and some its portion is always being imparted through the modes corresponding to the frequencies, neighbouring the dominant one. Although our model of excitation is very simple, it reflects the above spread of energy, even when zero-order approximation is employed.

To make use of the approximation (5.11) we have to substitute it into formula (5.9) and perform the integration. Since all factors of the integrands are rational functions, we can express the integrals analytically in terms of the integrands' residua

$$\int_{-\infty}^{\infty} \mathcal{Z}(z)\,dz = 2\pi i \sum_{Im^2 k>0} Res_{z=z_u} \mathcal{Z} \quad \left(= -2\pi i \sum_{Im^2 k<0} Res_{z=z_k} \mathcal{Z} \right),$$

for functions $\mathcal{Z}$ analytical except, possibly, at the points z_k such that $Im\, z_k \neq 0$, and such that $\lim_{|z|\to\infty} z^2 \mathcal{Z}(z) = 0$. As the integrands are already factorized in (5.11), the algorithm is easy to derive. In what follows we provide all elements necessary for the computation of mean-square forces but we do not present the expanded, explicit formulae because they are long and hardly readable. A careful reader is encouraged to check the relations that were effectively employed in calculations, by analysing the listings of FORTRAN codes enclosed in the Appendices 1 and 2 (subroutines FUJN and FUKINJ).

Using the zero-order approximation we obtain

$$\sigma_i^2 \approx \sigma_i^2(0) = (P^i)^T C^0 P^i = \sum_{k=1}^{M} p_k^i \left(2 \sum_{j=1}^{k-1} p_j^i C_{kj}^0 + p_k^i C_{kk}^0 \right), \tag{5.12}$$

where

$$C_{kj}^0 = R_{kj} \int_{-\infty}^{\infty} S_z(\omega) \frac{1}{a_k(\omega)} \frac{1}{a_j(\omega)} d\omega. \tag{5.13}$$

Factor $S_z(\omega)$ can be expressed as follows (cf. (5.6)):

$$S(\omega) = 2\lambda \frac{\omega^2 + \lambda^2 + \nu^2}{((\omega-\nu)^2 + \lambda^2)((\omega+\nu)^2 + \lambda^2)}.$$

Thus, it has the poles at the points $s_-^{\pm} := -\nu \pm i\lambda$ and $s_+^{\pm} := \nu \pm i\lambda$. It is easy to check that

$$Res_{\omega=s_{\pm}^+} S_z(\omega) = \frac{1}{2i}. \qquad \text{and} \qquad Res_{\omega=s_{\pm}^-} S_z(\omega) = -\frac{1}{2i}.$$

Similarly, the direct calculation proves that

$$Z_k(\omega) := a_k^{-1}(\omega) = -\frac{1}{\omega - r_k^+} \frac{1}{\omega - r_k^-}$$

$$\bar{Z}_k(\omega) := \bar{a}_k^{-1}(\omega) = -\frac{1}{\omega - \bar{r}_k^+} \frac{1}{\omega - \bar{r}_k^-},$$

where:

$$r_k^{\pm} := \pm\Delta_k + i\rho_{kk}, \qquad \tilde{r}_k^{\pm} := \Delta_k - i\rho_{kk},$$

so that

$$Res_{\omega=r_k^+} Z_k(\omega) = Res_{\omega=\tilde{r}_k^+} \tilde{Z}_k(\omega) = -\frac{1}{2\Delta_k},$$
$$Res_{\omega=r_k^-} Z_k(\omega) = Res_{\omega=\tilde{r}_k^-} \tilde{Z}_k(\omega) = -\frac{1}{2\Delta_k}.$$

Remark 1: Recall that $\Delta_k = \sqrt{q_k^2 - \rho_{kk}^2}$ can either be purely real or purely imaginary. From the definition, $\tilde{r}_k^{\pm} = \overline{r}_k^{\pm}$, if Δ_k is real, and $\tilde{r}_k^{\pm} = \overline{r}_k^{\pm}$, if Δ_k is real.

Remark 2: $S_z(\overline{\omega}) = \overline{S_z(\omega)}$

Remark 3: $Z_k(\overline{\omega}) = \overline{\tilde{Z}_k(\omega)}$

To calculate the integral (5.13) we make use of one more obvious property: if a function h is analytic at $\omega = \omega_0$, then $Res_{\omega=\omega_0} g(\omega)h(\omega) = h(\omega_0) Res_{\omega=\omega_0} g(\omega)$.

Using the above we can derive:

$$C_{kj}^0 = 2\pi i R_{kj} \left\{ \frac{1}{2i} \left[\tilde{Z}_k(s_-^+) Z_j(s_-^+) + \tilde{Z}_k(s_+^+) Z_j(s_+^+) \right] + \frac{1}{2\Delta_j} \left[S_z(r_j^-) \tilde{Z}_k(r_j^-) - S_z(r_j^+) \tilde{Z}_k(r_j^+) \right] \right\}. \tag{5.14}$$

In fact, it is easy to show that the imaginary part of (5.14) vanishes, so that only the real part can be taken into account in the calculations.

The second-order correction $\sigma_n^2(2)$ can be calculated in a similar way ($\sigma_n^2 \approx \sigma_n^2(0) + \sigma_n^2(2)$). Here are the formulae:

$$\sigma_n^2(2) = (P^n)^T \left\{ \overline{A}^{-1} B \overline{A}^{-1} S_Z A^{-1} B A^{-1} - \left(\overline{A}^{-1} B \overline{A}^{-1} B \overline{A}^{-1} S_Z A^{-1} - \overline{A}^{-1} S_Z A^{-1} B A^{-1} B A^{-1} \right) \right\} P^n$$
$$= \sum_{i=1}^{M} p_i^n \sum_{l=1}^{M} p_l^n \sum_{j=1}^{M} \left[\rho_{ij}^* \sum_{k=1}^{M} R_{jk} \rho_{kl}^* S_{ijkl}^1 - R_{ij} \sum_{k=1}^{M} \rho_{jk}^* (S_{ijkl}^z + S_{ijkl}^3) \right], \tag{5.15}$$

where $\rho_{jn}^* = \rho_{jn}(1 - \delta_{jk})$, (δ_{jk} is the Kronecker delta here), and

$$S_{ijkl}^1 = -2\pi i \sum_{r \in \{s_-^+, s_+^+, r_k^+, r_k^-, r_l^+, r_l^-\}} Res_{\omega=r} S_z(\omega) \omega^2 \tilde{Z}_i(\omega) \tilde{Z}_j(\omega) Z_k(\omega) Z_l(\omega)$$

$$S_{ijkl}^2 = -2\pi i \sum_{r \in \{s_-^+, s_+^+, r_i^-, r_l^+\}} Res_{\omega=r} S_z(\omega) \omega^2 Z_i(\omega) \tilde{Z}_j(\omega) \tilde{Z}_k(\omega) \tilde{Z}_l(\omega)$$

$$S_{ijkl}^3 = +2\pi i \sum_{r \in \{s_-^-, s_+^-, \tilde{r}_i^+, \tilde{r}_i^-\}} Res_{\omega=r} S_z(\omega) \omega^2 \tilde{Z}_i(\omega) Z_j(\omega) Z_k(\omega) Z_l(\omega).$$

6 Results

Numerical calculations were performed for several configurations of parameters. One particular configuration is listed below. In what follows, the values in the list will be referred to, as the "standard values". For any particular result only those values of parameters will be mentioned which are different from the standard values.

Conductor tension, T:	30,000 [N]
Span length, L:	300 [m]
Conductor mass density, ρ :	2 [kg/m]
Spacer stiffness, K:	50,000 [N/m]
Spacer damping constant, β :	1,000 [Ns/m]
Structural damping constant, d_{str} :	1 [Ns/m]
Intensity of excitation, σ_z :	1 [N/m]
Excitation correlation length, $\frac{1}{\alpha}$:	30 [m]
Excitation correlation time, $\frac{1}{\lambda}$:	0.3 [s]
Excitation correlation wave length, $1/\mu$:	3 [m]
Excitation correlation wave period, $1/\nu$:	2 [s]

6.1. Convergence of results

The solution of the governing equation was obtained using the Fourier method of separation of variables. Consequently, we approximate the solution by a finite Fourier series expansion with respect to the spatial eigenforms of an associated problem. The estimates of the mean-square forces take also the form of finite Fourier series (cf.(5.12), (5.15)). For the convergence of results, as the number of accounted modes goes to infinity, a sufficiently fast convergence of the Fourier coefficients of these series is necessary.

In particular, for the approximation (5.12) to converge as M tends to infinity, it is necessary that C^0_{jk} , the global Fourier coefficients (see (5.13)), tend to zero sufficiently fast, as j, k tend to infinity. Analyzing the formulae for C^0_{jk} we notice that C^0_{jk} is a product of the temporal and spatial Fourier coefficients (the latter can be identified with

R_{jk}). Further analysis shows that both temporal and spatial Fourier coefficients are of the magnitude of $(q_j q_k)^{-1}$. Since the order of growth of q_j is linear with respect to j, and $\sum_{k=1}^{\infty}(\frac{r_k'}{q_k})^2 < \infty$, this suffices for the convergence of the series.

The ratios R_{jk}/q_k^2 and $C_{kk}^0/(R_{kk}q_k^2)$ reflect the degree of the excitation of a particular mode. If all modes were excited to the same degree, these ratios would be constant as functions of the mode number. In Figures 3, 4, and 5 the ratios of $R_{kk}/q_k^2, C_{kk}^0/(R_{kk}q_k^2)$ and C_{kk}^0/q_{kk}^4 are plotted for the standard set of parameters and for one spacer placed at the position 0.411333. The choice of the position was purely random. The general shape of the spectra does not vary significantly with the change of the position. Only some isolated peaks strongly depend on the position, which is natural, as the spacer's placement influences the resonance characteristic of the system. The overall shape, in turn, can be related to the loading spectrum.

It can be seen that, not only the rate of convergence is preserved, but also the bandwidth of the *relative* spectra is finite. Even if the relative spectra presented were of infinite bandwidth (remaining bounded though), the Fourier series would still converge. The addition brought to the series by the mode of number k, would be of the order $1/k^2$. This means that the remainder of the series related to the modes of the numbers larger than 500 would be negligible (as assumed and discussed in the introduction).

Localization of spectra enhances the above effect. In most cases, the approximation obtained for $M = 50$ was indistinguishable from that obtained for $M = 500$. This is illustrated by several further results. Figures 6, 7, 8, and 9 represent the mean-square forces for different values of M in function of the position of one of two spacers, where position of the second spacer is fixed, and is equal to $100m$ (that is, the second spacer is placed at one third of the span). Consecutive results were obtained using 10, 50, 100, and 500 modes, respectively (damping and structural damping for these calculations were, in all four cases, equal $10[Ns/m]$ and $0.001[Ns/m]$, resp.). It is apparent that already the approximation obtained with 50 modes is practically indistinguishable from that obtained by the use of 500 modes. Similar convergence was observed also for a larger numbers of spacers. Figures 10 and 11 illustrate the differences between approximations obtained with the help of 20 and 100 modes, respectively, in case of 5 spacers. The positions of spacers 1, 2, 4, and 5 were 50, 100, 200, and 250 m, respectively (1/6, 2/6, 4/6, 5/6 of the conductor's length). The position of the third spacer varied around the middle point. In this study, the damping constants were 10 and $10^{-3}[Ns/m]$. It can be noticed that already with 20 modes a qualitatively good approximation is obtained. Approximations with more than 100 modes are not presented as they are indistinguishable form that for 100 modes.

Another aspect of the precision of estimation is that connected with the method used. In the Figures 12 and 13 the results of calculations performed with the zero-order and second-order spectral method are presented. For these calculations the damping constants were 1 and $0.01[Ns/m]$, respectively. The results are practically identical. Differences cannot be seen on the graphs as they appear in the fifth, or sixth digit only.

The basic numerical experiment, giving insight into the following results, is illustrated on Fig. 14. It represents the forces resulting from the first mode of vibrations in dependence on the position of a single spacer. The most important issues influencing the results are well represented at this graph. Basically, there are two components of the msq-curves plotted on all the Figures. One component is connected with the lack of damping, appearing whenever the spacer is placed at zeros of a mode-shape function (an eigenmode of the associated problem). Since there is no displacement at these points, no damping is provided by the spacer. This results in a great level of vibration and large forces at the conductor clamps, what is reflected by the peaks apearing in the Figures. The heights of the peaks are dependent on the structural damping constant.

The remaining part of the graph reflects the effectiveness of damping provided by the spacer. This part will be called the "regular part". The level of the regular part depends on the spacer damping constant.

The two parts, the peaks and the regular part, are independent in the sense that the magnitude of the structural damping influences the regular part very slightly, while the spacer damping has no effect on the peaks. Therefore, we can consider all further results presented, as combinations of the two types of results, and one can easily imagine what would be the change of the pattern if the proportion between the two damping constants was changed.

In Figures 15 and 16 the regular part of solution is plotted for one spacer (both Figures show the same curve, in two different ranges: Figure 15 along the entire span, Figure 16 in the neighbourhood of the middle of the span (the damping constants were $\beta = 1$, and $d_{str} = 0.01$). The peaks do not show up, because the spacer damping is too small to damp the vibrations, and the relatively stiff spacer causes big strains at its clamp not only when the spacer is placed at "resonant" positions, but at almost any position.

Consecutive graphs illustrate the mutual independence of the peaks and the regular part, as well as the dependence of levels of forces on the magnitude of damping constants.

The curves in Figures 17, 18, and 19 are almost identical, because the ratio of the spacer damping constant to the structural damping constant is the same for all these Figures. However, the level of forces varies. The magnitudes of damping constants are $(1, 10^{-4}), (10, 10^{-3}), (100, 10^{-2})$, respectively. The magnitudes of mean-square forces are $10^6, 10^5$ and 10^4 , respectively.

To observe the effect of the change of the damping ratio, $\frac{\beta}{d_{str}}$, on the build-up of peaks, it is worth to compare the results presented in Figures 20, 21, 19, 22, and 23 (in this succession!). The damping ratios are $10^2, 10^3, 10^4, 10^5$, and 10^7 , respectively. The spacer damping constants were equal to $1, 10, 100, 1000$, and $100[Ns/m]$, respectively. The latter case $(\beta = 1000. d_{st} = 10^{-4})$ is also shown in Figure 24, where the regular part (invisible in Figure 23) is well seen. The results confirm that the level of the regular part is conversely proportional to the spacer damping, and the heigth of the peaks is conversely

proportional to the structural damping. It is worth noting that the shape of the regular part seen in the graph 24 is essentially the same as that in the graph 20.

There are many other parameters whose influence on the msq-forces can be studied with the help of the method presented. For instance, Figure 25 shows an entirely different pattern. Here the damping constants are again 1 and 0.01 , respectively, but the damper is much softer that that represented in graph 15. Therefore, the forces at the spacer clamps are much lower than in the case of the stiff spacer. Note, however, that the level of forces at the conductor clamps is, in general, higher. Note also that, if the spacer is soft, then its position does not have a big influence on the forces at the clamps.

Another kind of parameters are the parameters of loading. Graph 27 represents the results obtained for the excitation with much smaller correlation. The damping constants are 1 and 0.01 , respectively. Therefore, only the regular part is visible (for smaller structural damping the peaks would appear exactly at the local minima of the regular part). Note, that the curves at Figure 26 are strikingly regular. Indeed, as the excitation tends toward the white noise, the response approaches the static Green's function for the clamped string. Note also that the local minima at one third and one half of the span result from the fact that these are the locations of the maxima of the first and second eigenmodes at the free string. Therefore, the spacer damping has the strongest effect on those most significant modes.

Studies performed for one spacer are important, although it is rather unusual to use only one spacer in practice. The importance of these studies follows from the fact that the results can be relatively easily interpreted, and the mechanisms revealed by these studies apply also to a bigger number of spacers.

6.3. Results for two spacers

In Figure 27 one finds the basic pattern playing the same role for two spacers as the graph 14 in the case of one spacer; the surface represents the envelope of mean-square forces versus two spacers locations. Again, the regular parts, and the peak parts can be observed. The values of the parameters were $\beta = 10, d_{str} = 0.01$.

Some of the results for two spacers have already been mentioned. These are the results represented in Figures 6 through 9, showing the msq-strains at both conductor clamps, and at the spacers' clamps, as functions of the second spacer's position, if the first spacer is mounted exactly at one third of the span. The resonance peaks are easily seen and they appear, of course, if the second spacer is mounted at two thirds of the span. It is interesting to know the sensitivity of the result, with respect to the position of the first spacer. In Figure 28 the forces are represented in the case, where the first spacer is positioned at 33.33% of the span, that is approximately 1 cm away from the one third of the span length. The peak still occurs, but its height is reduced from $80,000[N^2]$ to $30,000[N^2]$. The results shown in Figure 29 are obtained for the first spacer located at

33.30% of the conductor's span, i.e., approximately 10 cm away from one third. The peak disappears. Note, however, that the level of the regular part is slightly higher than in Figures 9 and 28. The effect is even more evident from Figure 30, where the first spacer is already about 1 m apart from one third of the conductor's span. We see the trade-off between the two mechanisms: maximization of damping effects at the maxima of certain eigenmodes, and minimization of damping of those eigenmodes which have zeroes at the same points. The first mechanism leads to local minima at the locations like $(1/3, 2/3)$, whereas with respect to the second mechanism this placement is of the "resonance" type, and leads to the build-up of almost undamped vibrations.

It is also well seen from Figures 31 and 32, representing the envelope of msq-forces (the largest of the four msq-forces: at both ends and at both spacer clamps) versus both spacer locations. In Figure 31 the orthotropic view of this value is shown, while in Figure 32 the topographic representation is plotted. The symmetry of the system is well reflected. It is also apparent that the peaks are situated at the local minima of the regular part. The results in Figures 31, 32 were obtained for damping constants $\beta = 10[Ns/m]$ and $d_{str} = 0.001[Ns/m]$.

The following three graphs illustrate the effect of other parameters on the shape of the envelope of msq-forces (recall that it is just this envelope which is the cost-function; see introduction). All three of them show the results obtained with $\beta = 10[Ns/m], d_{str} = 0.003[Ns/m]$, and with 50 modes taken into account.

In Figure 33 one can see the effect of the correlation constants' magnitudes on the level of mean-square forces. The correlation time is assumed only $0.01[s]$ and the correlation length only $1[m]$. One can conclude that such an irregular, "strongly random" kind of loading is not as damaging as the more regular (more correlated) loadings considered previously.

An even stronger effect is seen in Figure 34, where the excitation wave-length and the wave-period were assumed very small: $0.3[m]$ and $0.02[m]$, respectively. The response level is incomparable with that seen in Figure 31, for instance. This result may, however, be somewhat misleading, because also in this case only the 50 first eigenfrequencies were taken into account, whereas the loading energy is imparted through much shorter wave-lengths. Anyway, it is interesting to note that, as in other cases, the minimal values are assumed when the spacers are located symmetrically with respect to the conductor's length.

This conjecture is even more evident from Figure 35 related to the case, where much softer spacers were used: their stiffness was only $100[Ns/m]$. At first glance it might be astonishing that the use of softer spacers increases the level of strains. Indeed, compared to the Figure 31 the level is about four times higher! The reason is that the envelope shown in Figure 35 lies usually far above the curve representing the forces at the spacers' clamps. The softer the spacers are, the smaller are the forces at their clamps, of course. However, if the spacers are too soft, then the relation between spacers stiffness, damping properties, and the conductor tension, leads to the decrease of their damping efficiency, and consequently to the build-up of large vibrations and big strains at the conductor clamps. The same effect has already been observed in Figure 25, in the case of one spacer.

Similar case studies can be accomplished for other combinations of parameters, as well as for higher numbers of spacers. However, as the number of spacers grows, the situation becomes more and more involved, and many effects interfere with each other. Thus, only few results concerning the cases of three, and of five spacers will be reported.

6.4. Results for three spacers

By means of Figures 36, 37, 38 we wish to illustrate once more the dual nature of the damping phenomenon: the trade-off between looking for the local minima of the regular part of the cost-function and the danger connected with peaks' occurrences. The results represented at the above-mentioned Figures were calculated with $\beta = 1000[Ns/m]$, $d_{str} = 0.01[Ns/m]$. The strains are shown as functions of the position of the middle spacer. The side spacers were located exactly at one fourth, and at three fourths of the span, for Figures 36, 37, and were slightly (approximately 1.5 m) shifted toward the center, in the case of Figure 38.

The peaks in the graph 36 dominate the regular part, which stresses the importance of the analysis for the conductor lifetime. To show the regular part of msq-curves Figure 37 is provided. Comparing Figures 37 and 38 one notices that at the equidistant points (related to the uniform distribution of spacers along the span) the resonant peaks do not appear. At other places, however, the level of regular parts is higher.

In this connection, the relation between the spacer damping constant and the structural damping constant is extremely important. If the structural damping constant is large enough compared to the spacer damping, the peaks do not dominate the regular part. This is illustrated by Figures 39, 40, where $\beta = 10, d_{str} = 0.01$. Here, the forces are shown in dependence of the first ("left") spacer's position.

In the case shown in Figure 39, the middle spacer was placed exactly in the middle of the span, and the third ("right") spacer, exactly at three fourths of the conductor length. Therefore, a peak appears if the first spacer is placed at one fourth. However, the height of the peak is small, compared to the level of the mean-square forces at left clamp, for instance. There are many interesting issues seen from this Figure, like the low level of forces at the right clamp for almost all positions of the left spacer (except for the one fourth), or a big variation of the force at the left clamp. Let us, however, concentrate on the problem of avoiding the resonance peaks.

The case represented in Figure 40 differs from the previously considered in that both the middle spacer and the right spacer were shifted to the right by 1.5 m. The peak vanishes, but the level of all forces increases. This strongly suggests that the spacers should be constructed in such a way, that the *structural* damping constant would be increased. As already mentioned, it can be achieved by using the possibility to damp the angular vibrations at the spacer clamp.

Numerical experiments for five spacers were performed with $\beta = 10, d_{str} = 10^{-3}$, so that the peaks can be well distinguished from regular parts. Some of results were already discussed (Figures 10, 11). In Figures 41, 42, 43, and 44 one can see the results obtained by the use of 100 modes, when the middle spacer's position was the varying one, while the remaining spacers were shifted to the right from their uniform placements (1/6, 2/6, 4/6, 5/6) by ca 1 mm, 1 cm, 1 dm, and 1 m, respectively. As it was to be expected, if the shift is as small as 1 mm (Figure 41) then the results are hardly distinguishable form those represented in Figure 11. However, already as small a shift as 1 cm (Figure 44) is large enough to remove the peak from the picture! Recall that in case of two spacers such a shift had much smaller consequences. In the present case, however, the symmetry is being broken by four spacers, not one, as it was in the previous case. Figures 43 and 44 show how sensitive the system is to further shifts. Further breaking of the symmetry leads to the growth of forces, especially at the end, from which the spacers are more distant.

To give the feeling for the complication of the system with five spacers we provide also two graphs, representing the msq-forces as functions of the first (Figure 45) and of the second spacer's position (Figure 46). In both cases the remaining, fixed, four spacers were placed one centimeter to the right from their uniform positions. As in the case of the dependence on the middle spacer's position (Figure 42) the peaks do not appear. Nevertheless, the relations are not at all easy to interpret. One observation, which seems to be well documented, is that a shift of a particular spacer affects mostly the forces at this spacer's clamps, as well as at the neighbouring clamps. The forces at more distant points are less sensitive to such shifts. Another important observation is that the amount of variation is big. One meter of a difference in positioning leads to the differences of msq-forces which are of the order of 50% of the average level of the regular part. This enhances importance of mounting precision, as well as the significance of the bundle design.

7 Concluding Remarks

An effective method was developed for studying dependencies of mean-square forces at conductor and spacer clamps, on the spacers' positions, as well as on other parameters entering the model. Several conjectures have been derived. The seemingly most important conclusion expressed and illustrated in several parts of the text, can be summarized as

follows. There is a trade-off between damping efficiency of spacers, when they are placed at the maxima of certain eigenmodes, and their inefficiency of damping of those eigenmodes assuming zeros at those points. A remedy can be sought in increasing the efficiency of damping of angular displacements at the spacer clamps.

References

1. Hagedorn, P. *Ein einfaches Rechnenmodell zur Berechnung winderregter Schwingungen an Hochspannungsleitungen mit Dämpfern*, Ingenieur-Archiv, vol. 49, pp. 161-177, 1980.

2. Schmidt, J.T. and Hagedorn P. *A stochastic model for wind induced vibrations and damage accumulation in the cables of overhead transmission lines*, Proc. 1-st Int. Symp. on Probabilistic Methods Applied to Electric Power Systems, Toronto, Canada, 1986, Pergamon Press,pp. 485-494.

3. Kraus, M. and Hagedorn, P. *Aeolian vibrations: wind energy input evaluated from measurements on an energized transmission line*, IEEE, 90 SM 346-7 PWRD, 1990.

4. Hagedorn, P. and Kraus, M. *On the performance of spacer dampers in bundled conductors*, European Trans. on Electrical Power Engg., in print, 1993.

5. Bolotin, V.V. et al *On vibrations of overhead transmission lines and the fight with them*, (in Russian) Trudy Moskovskogo Ordena Lenina Energeticeskogo Instituta, vol. XXXII, pp. 21-53, 1959.

6. Bishop, R.E.D., and A.Y. Hassan, "The Lift and Drag Forces on a Circular Cylinder in a Flowing Field", *Proc. Roy. Soc. (London) Ser. A*, vol. 277, pp. 51-75, 1964.

7. Koopman, G.H., "The Vortex Wakes of Vibrating Cyllinders at low Reynolds Numbers", *J. Fluid Mech.* vol. 28, pp. 501-512, 1967.

8. Inman, D.J., *Vibration with Control, Measurement and Stability*, Prentice Hall, Inc., 352 p., 1989.

List of Figures

22. MSQ-forces vs spacer's location; spacer damping 1000; damping ratio 10^4 .

23. MSQ-forces vs spacer's location; spacer damping 1000; damping ratio 10^5 .

24. MSQ-forces vs spacer's location; regular part; spacer damping 1000; damping ratio 10^7 .

25. MSQ-forces vs spacer's location; the case of a soft damper.

26. MSQ-forces vs spacer's location; the case of a soft damper.

27. MSQ-strains vs spacers' location; one mode by two spacers.

28. MSQ-forces vs 2-nd spacer's position; first spacer at 33.33 % of the span.

29. MSQ-forces vs 2-nd spacer's position; first spacer at 33.3 % of the span.

30. MSQ-forces vs 2-nd spacer's position; first spacer at 33 % of the span.

31. MSQ-forces vs spacers' locations; orthotropic view.

32. MSQ-forces vs spacers' locations; topographical representation.

33. MSQ-forces vs spacers' locations; week (weak?) correlation case.

34. MSQ-forces vs spacers' locations; high-frequency excitation.

35. MSQ-forces vs spacers' locations; soft spacers case.

36. MSQ-forces vs 2-nd spacer's location; domination of resonance peaks.

37. MSQ-forces vs 2-nd spacer's location; regular part.

38. MSQ-forces vs 2-nd spacer's location; regular part free of peaks by non-uniform placement.

39. MSQ-forces vs 1-st spacer's location; uniform placement.

40. MSQ-forces vs 1-st spacer's location; non-uniform placement.

41. MSQ-forces vs 3-rd spacer's location; 1 mm shift from uniform placement.

42. MSQ-forces vs 3-rd spacer's location; 1 cm shift from uniform placement.

43. MSQ-forces vs 3-rd spacer's location; 1 dm shift from uniform placement.

44. MSQ-forces vs 3-rd spacer's location; 1 m shift from uniform placement.

45. MSQ-forces vs 1-st spacer's location; 1 cm shift from uniform placement.

46. MSQ-forces vs 2-nd spacer's location; 1 cm shift from uniform placement.

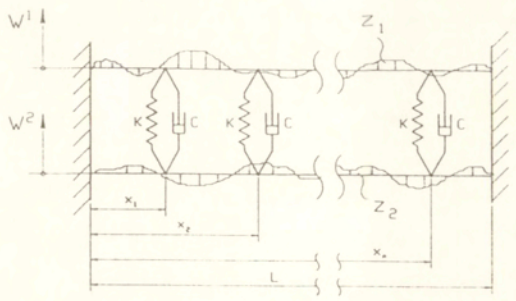

Fig. 1(a)

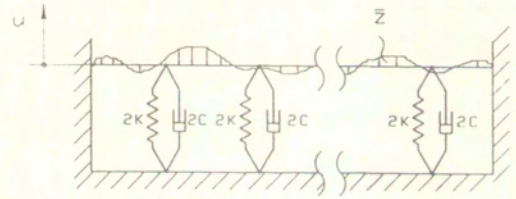

Fig. 1(b)

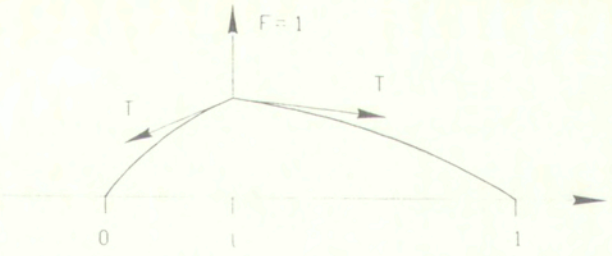

Fig. 2

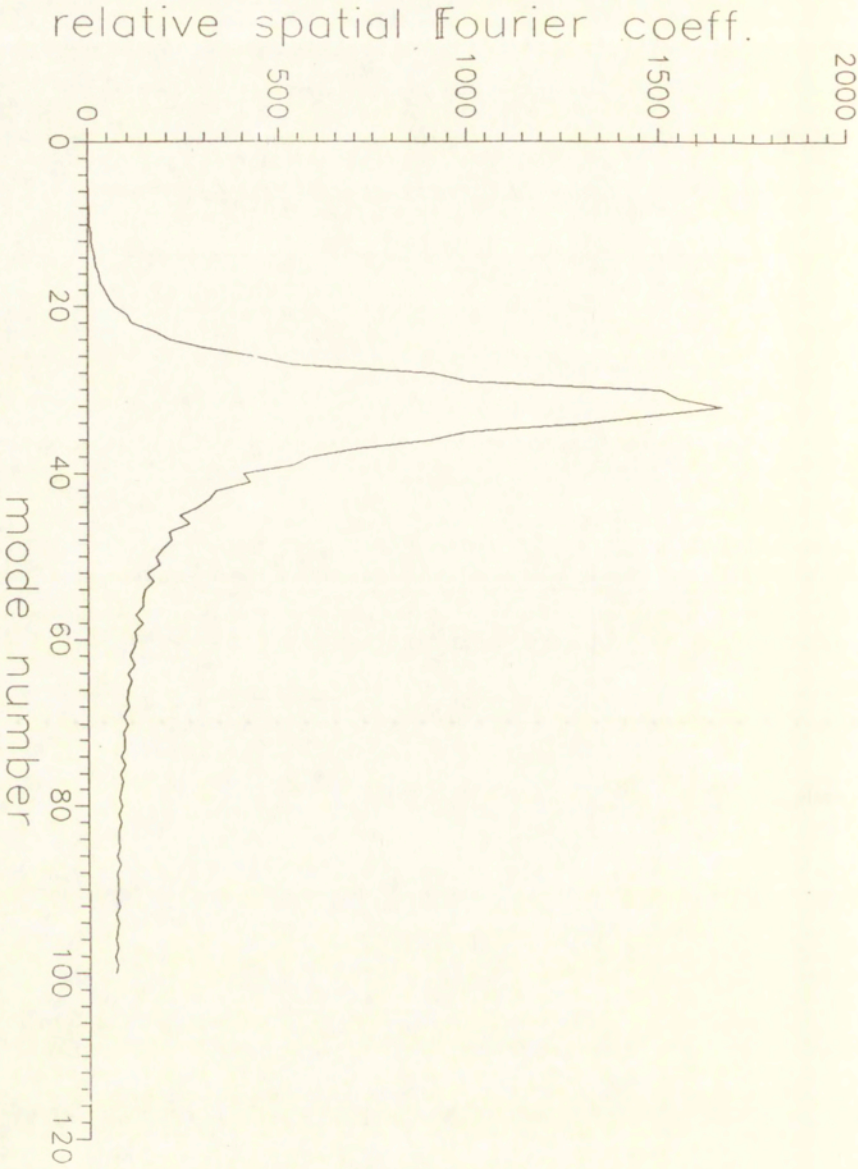

Fig. 3

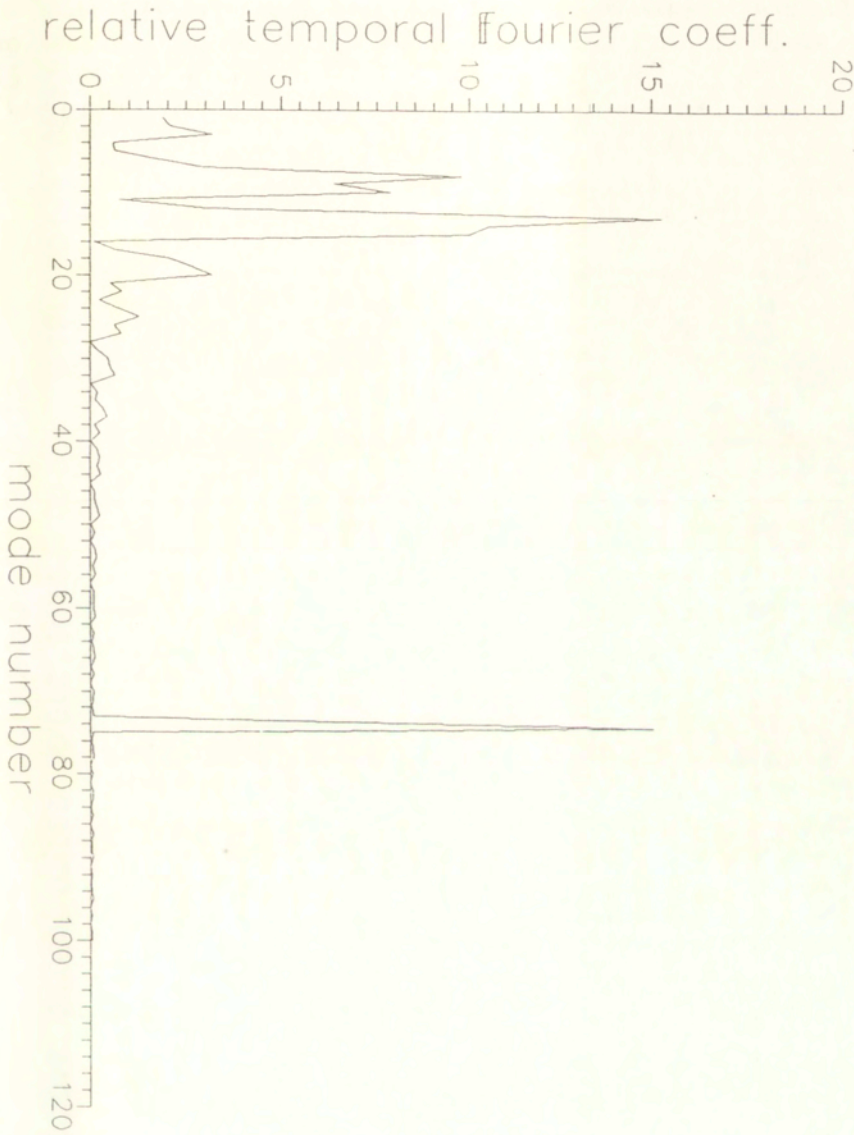

Fig. 4

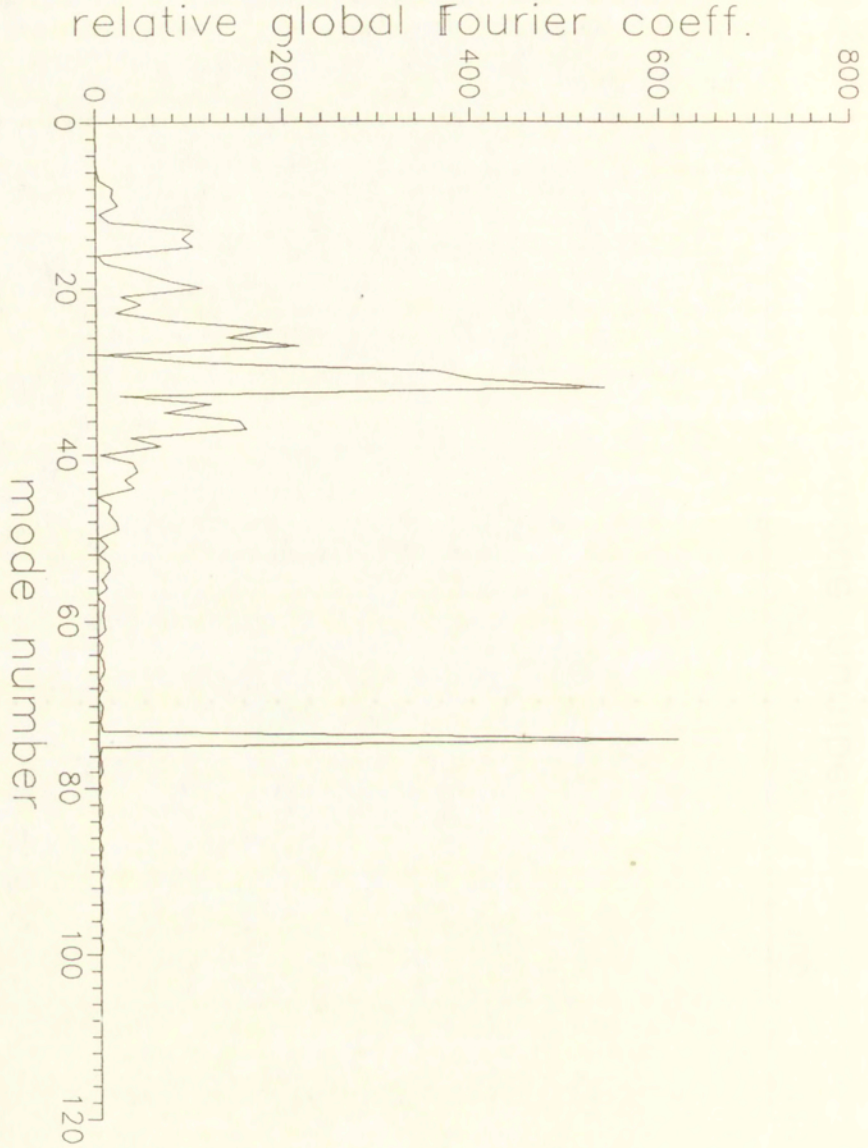

Fig. 5

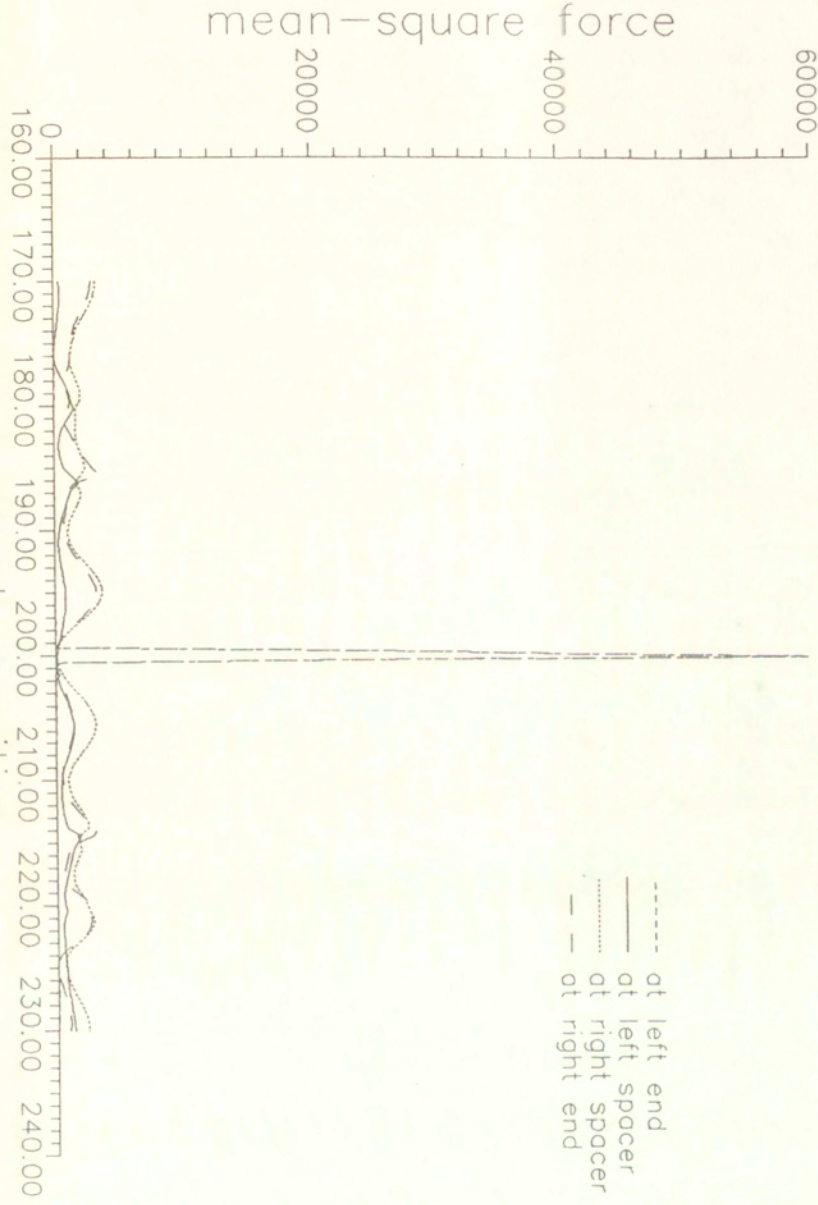

Fig. 6

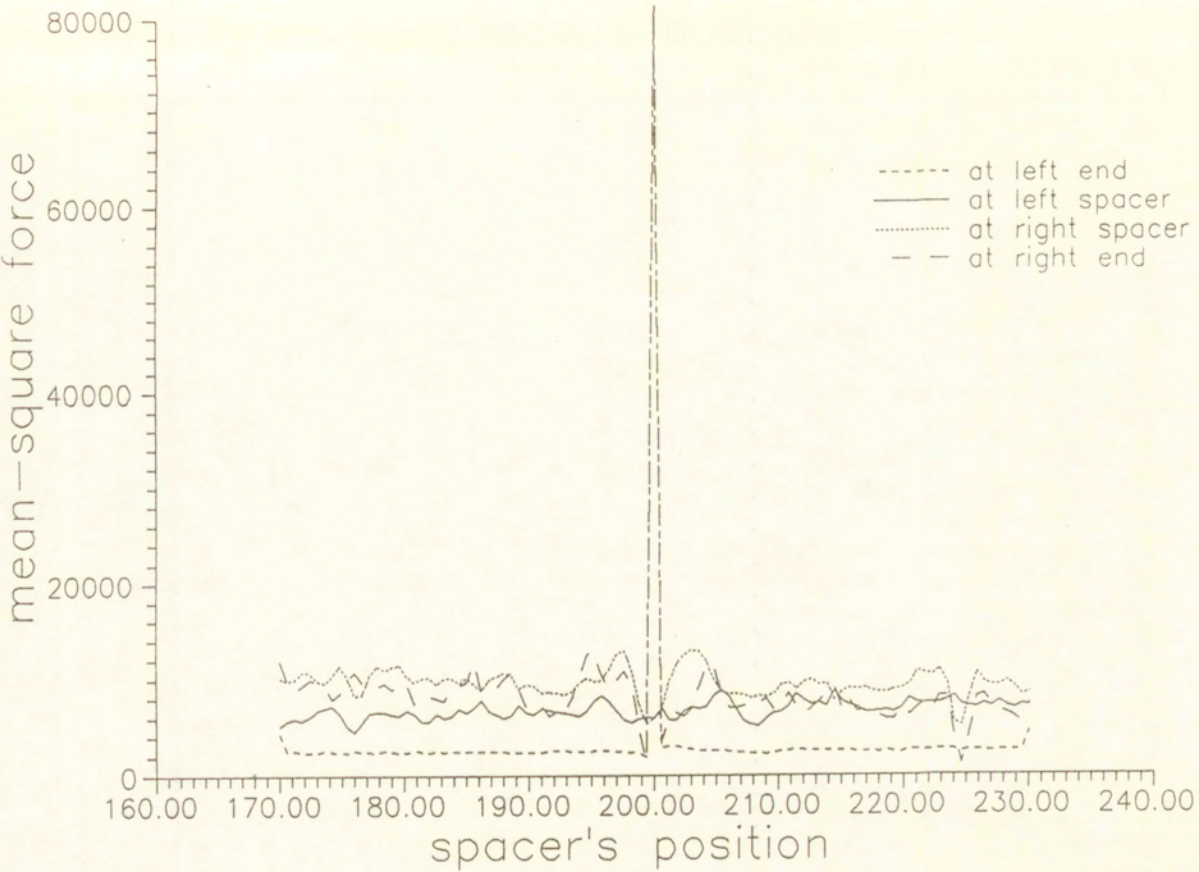

Fig. 7

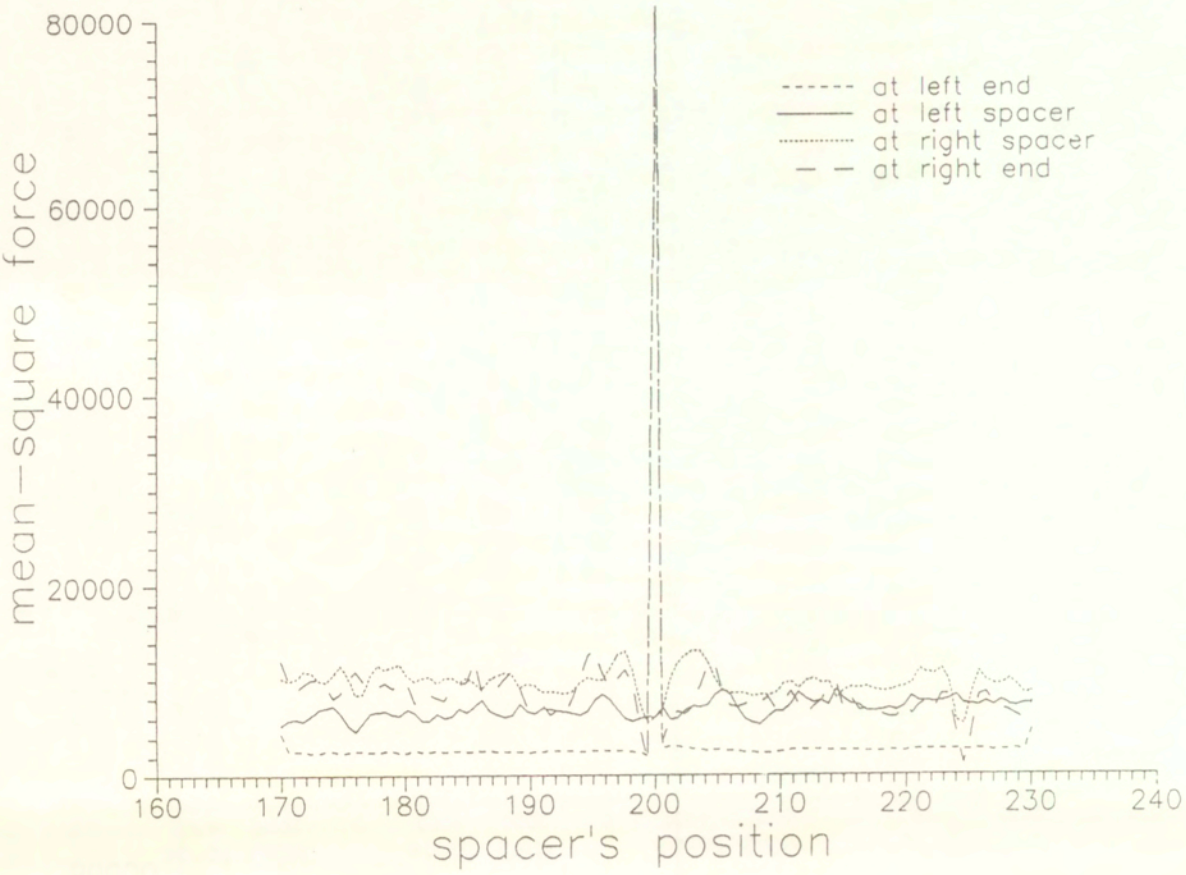

Fig. 8

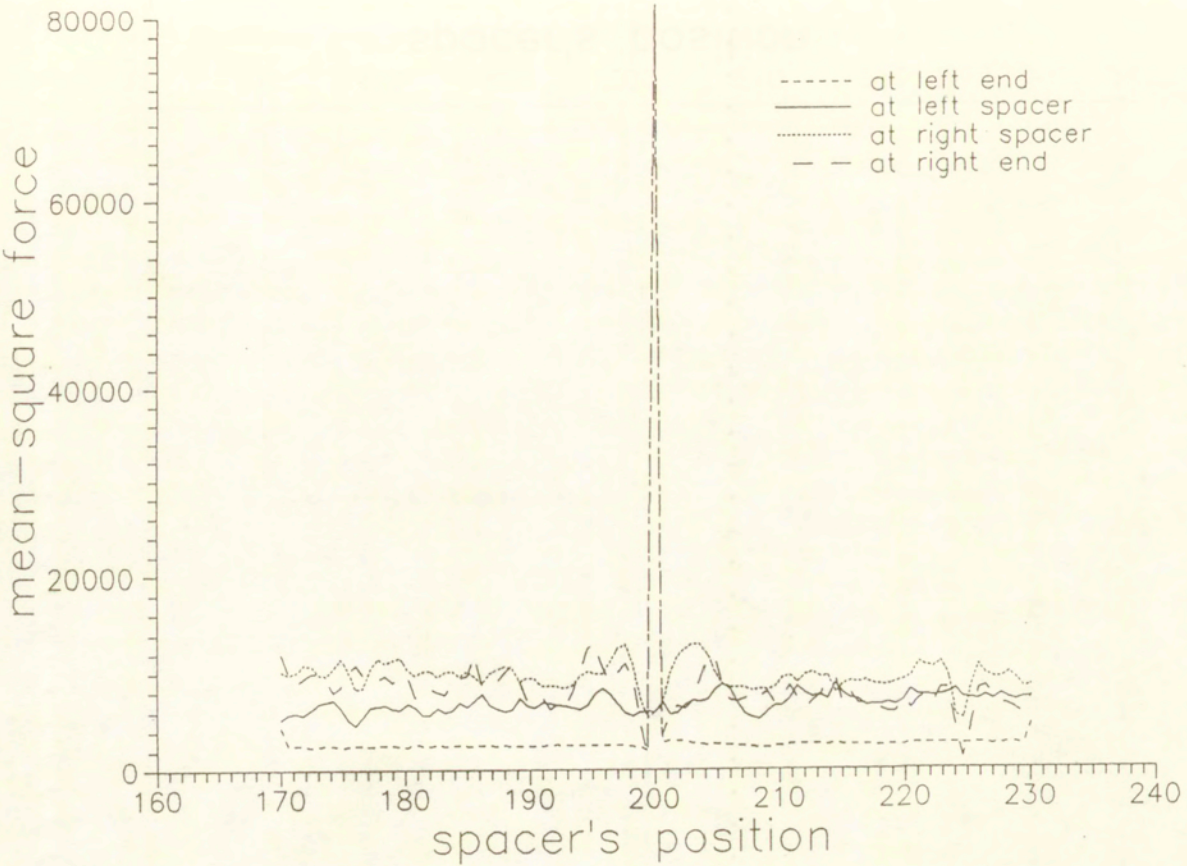

Fig. 9

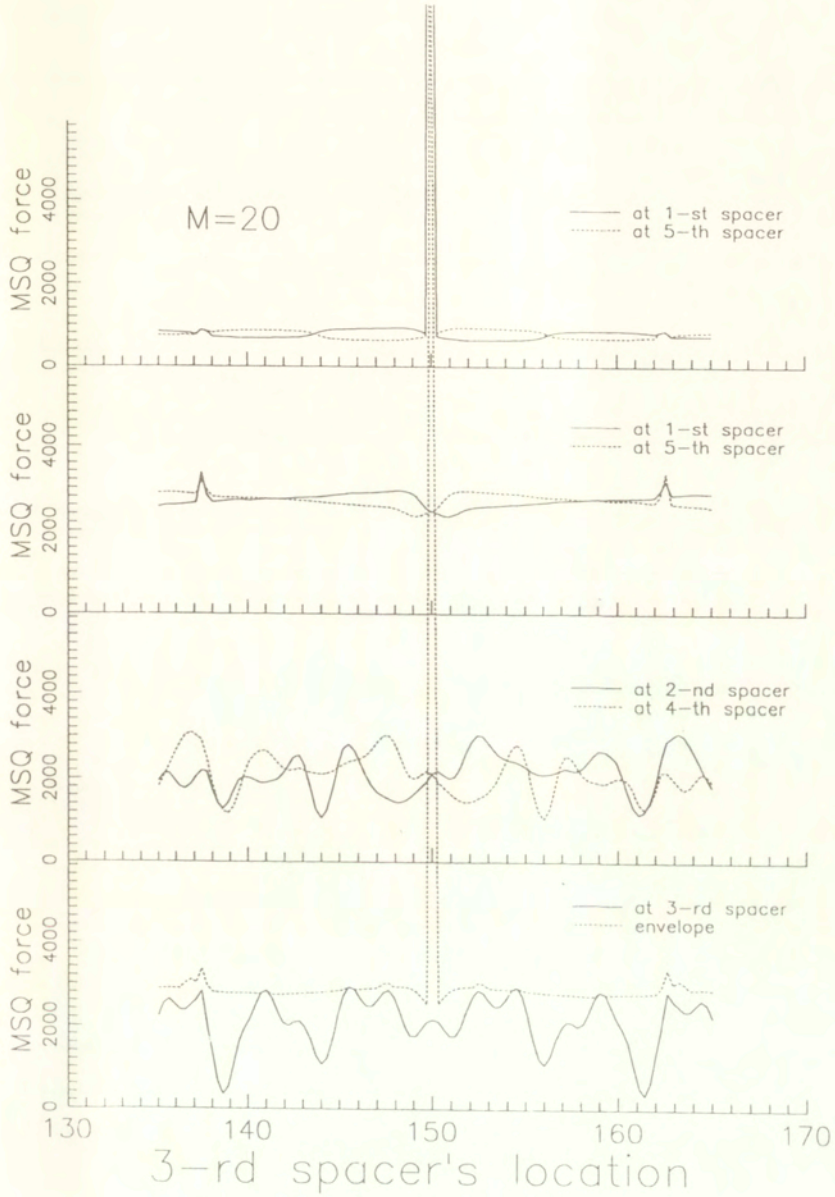

Fig. 10

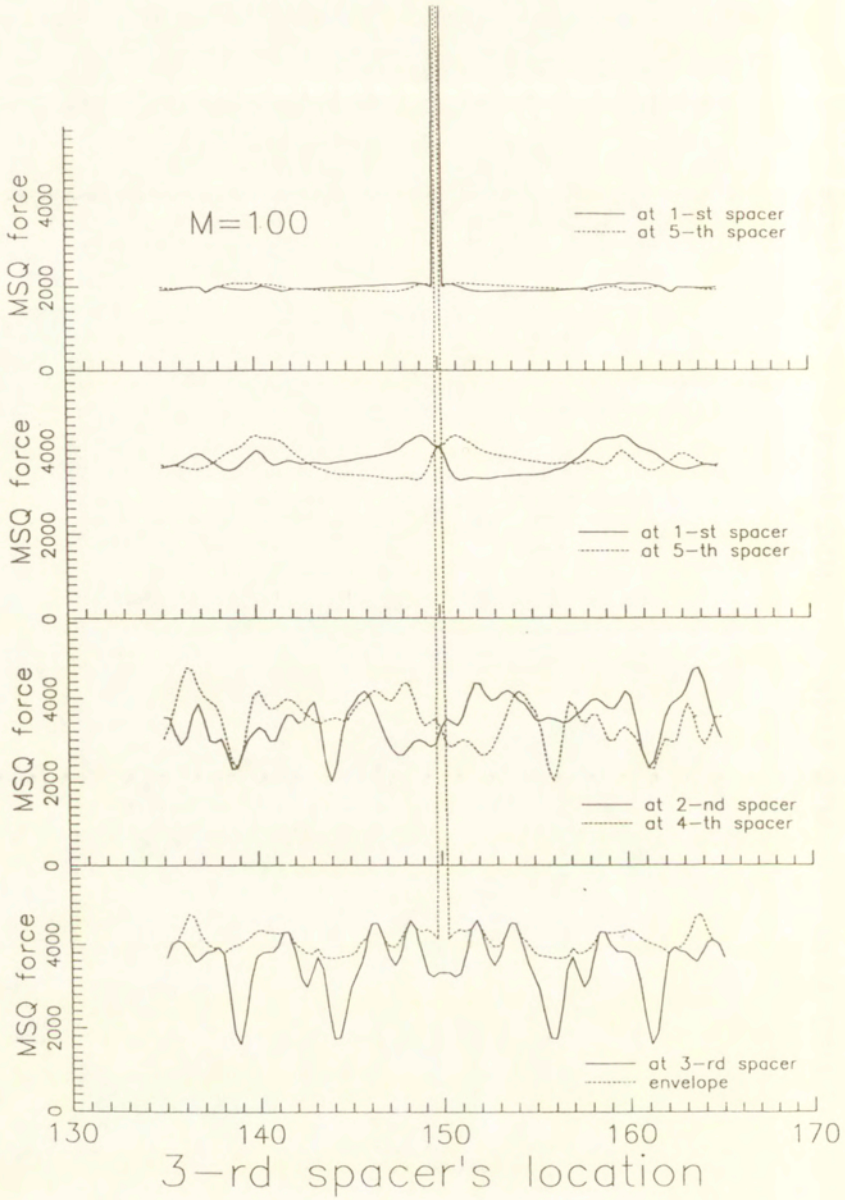

Fig. 11.

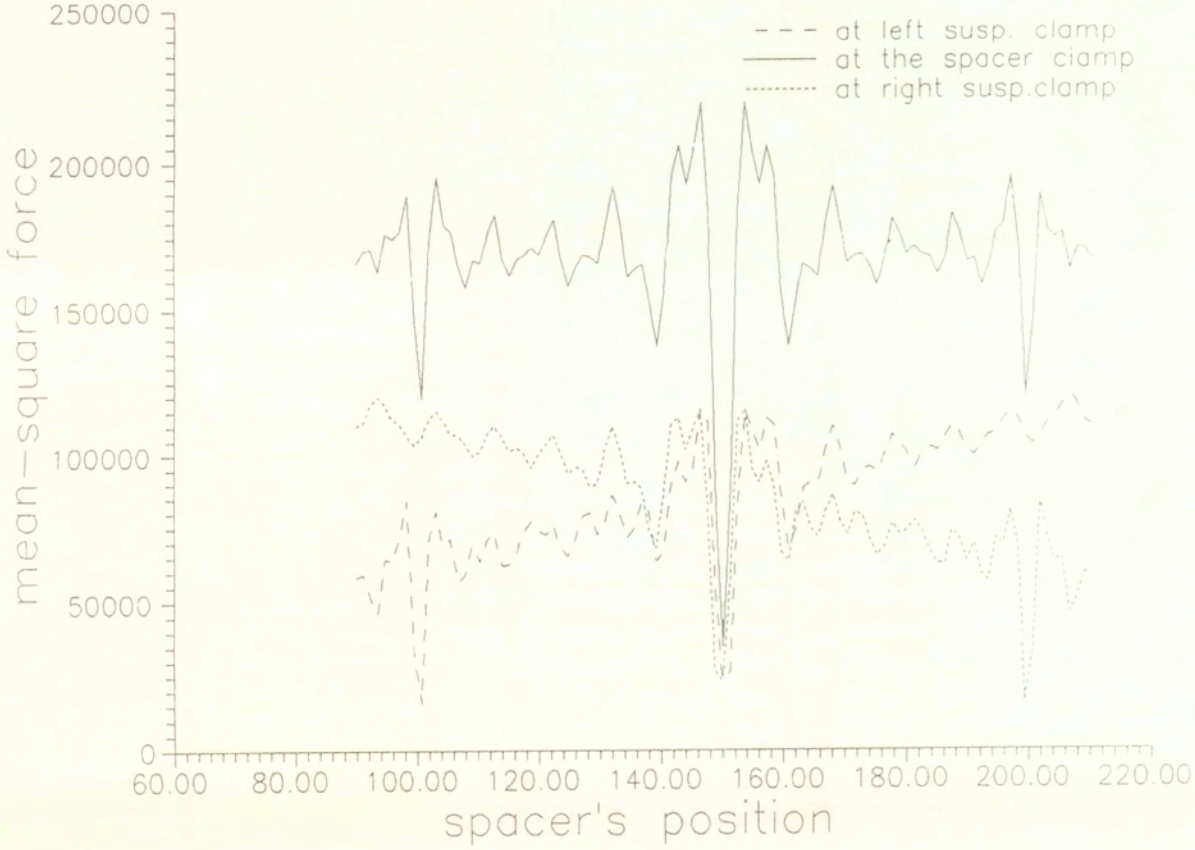

Fig. 12

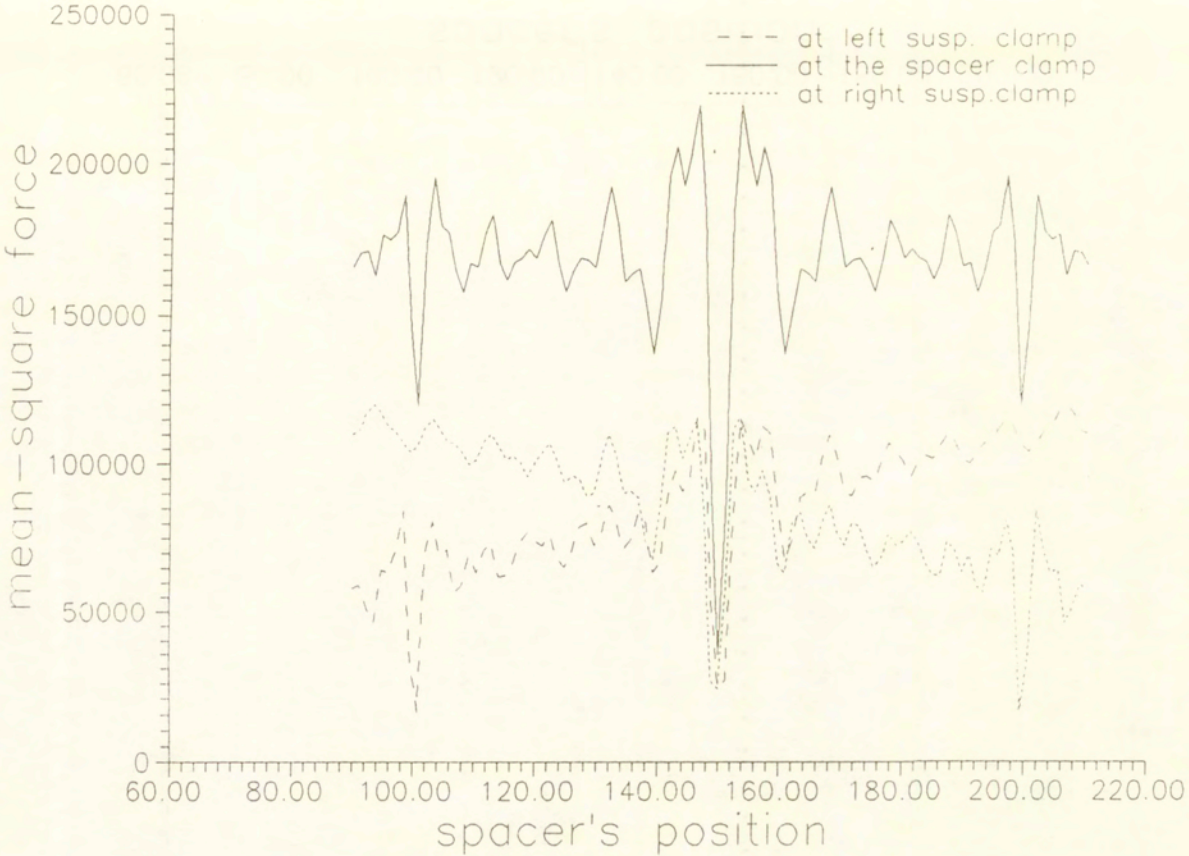

Fig. 13

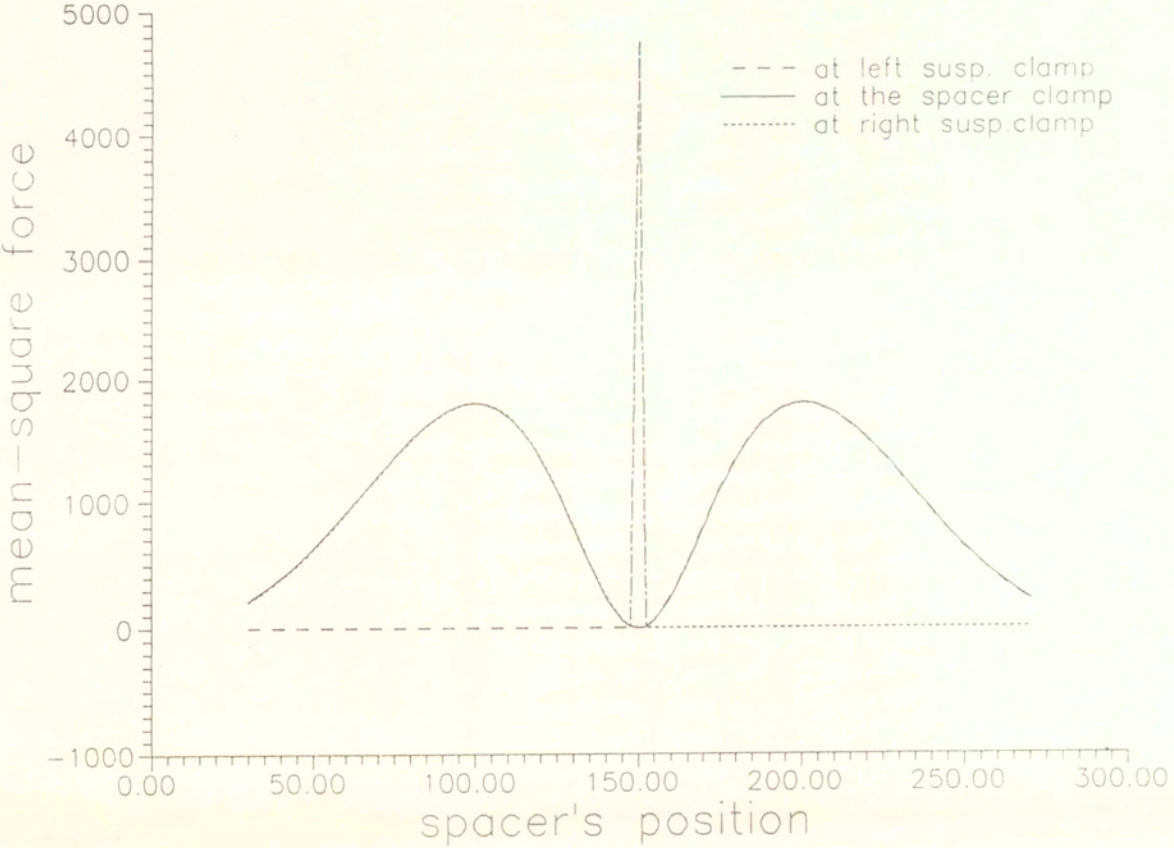

Fig. 14

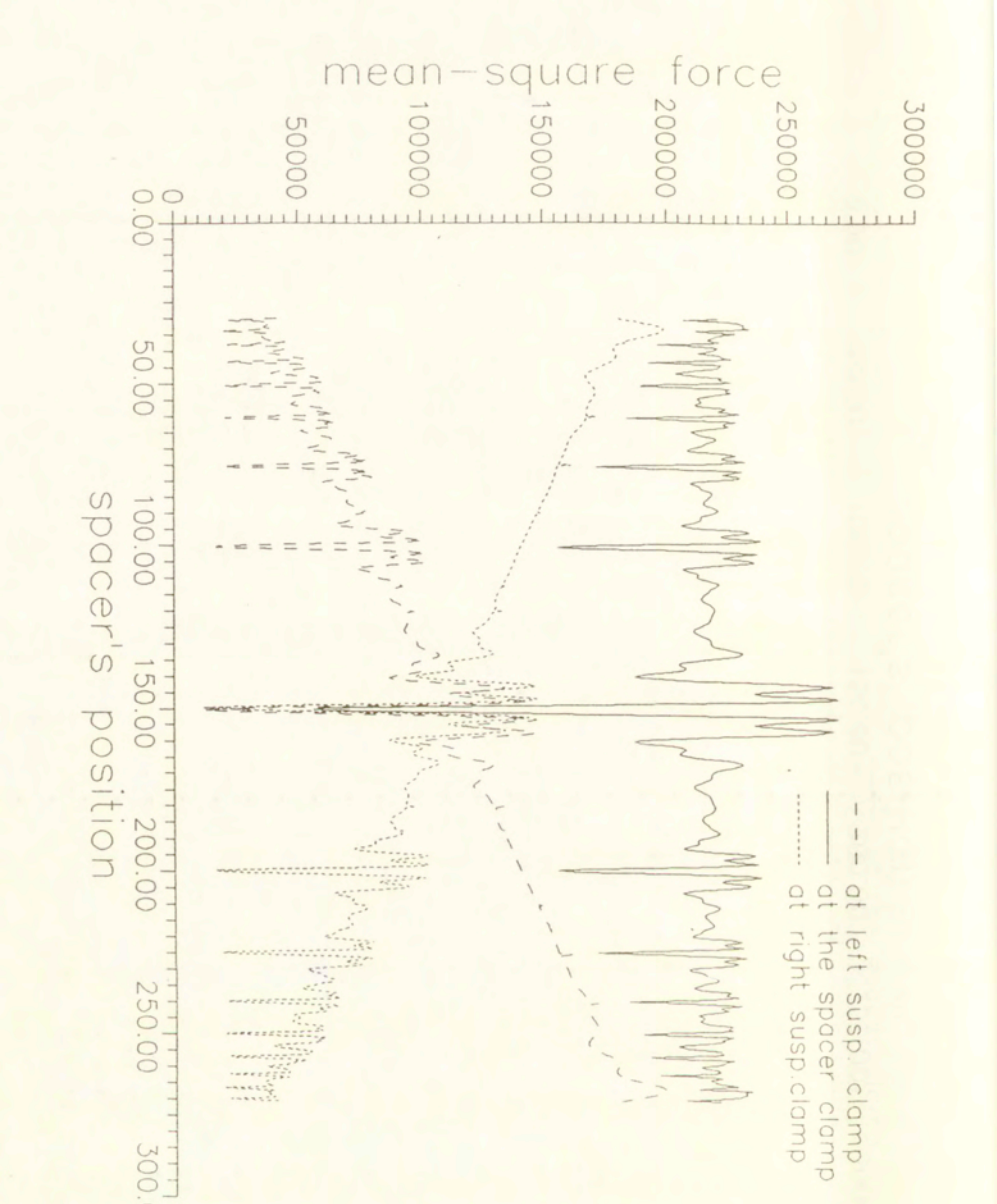

Fig. 15

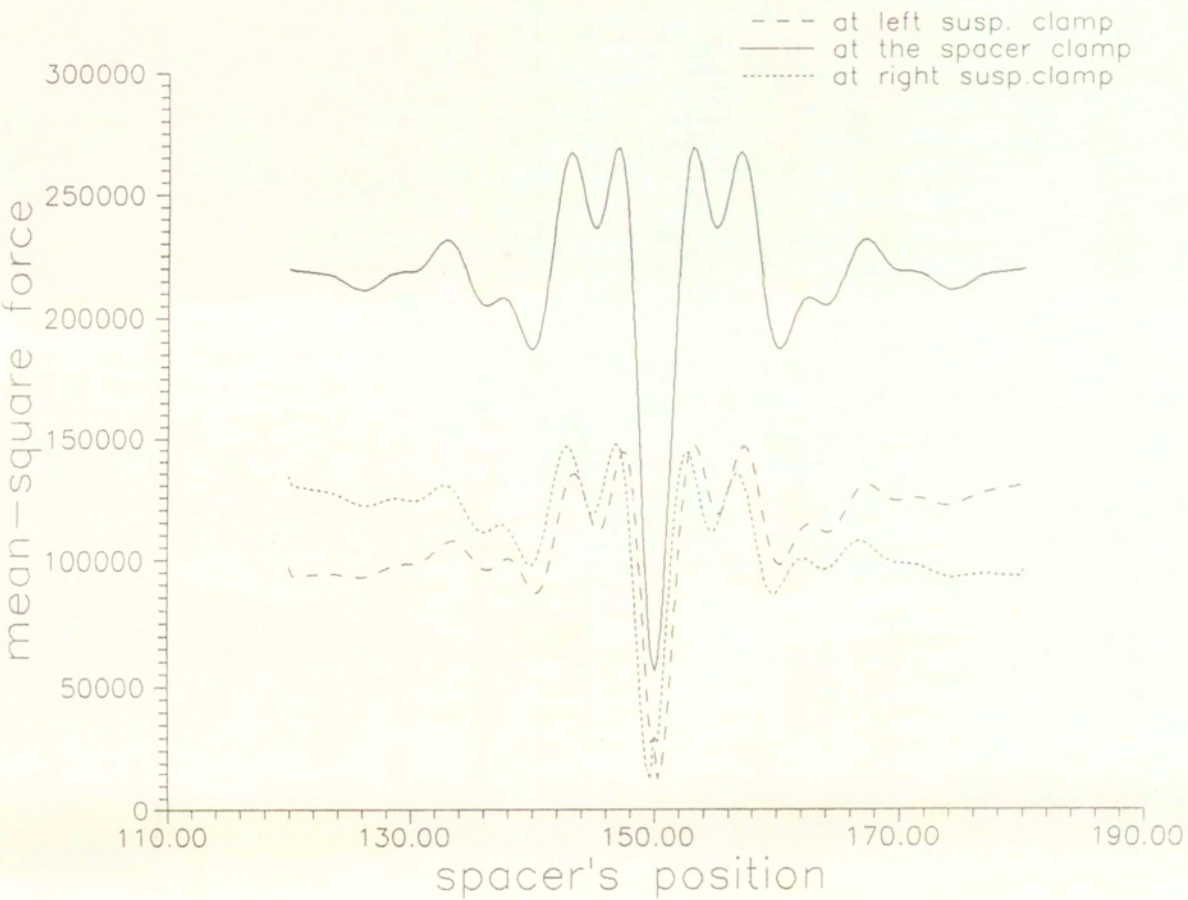

Fig. 16

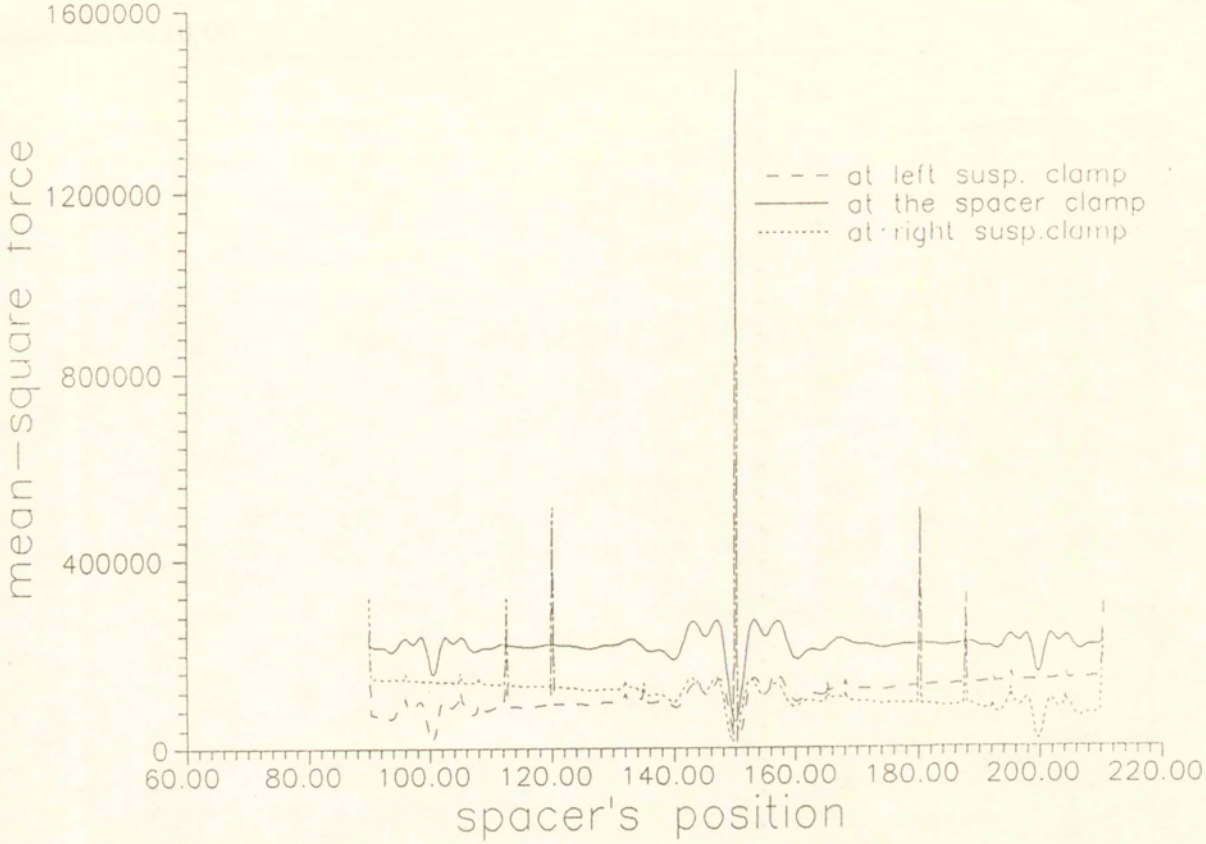

Fig. 17

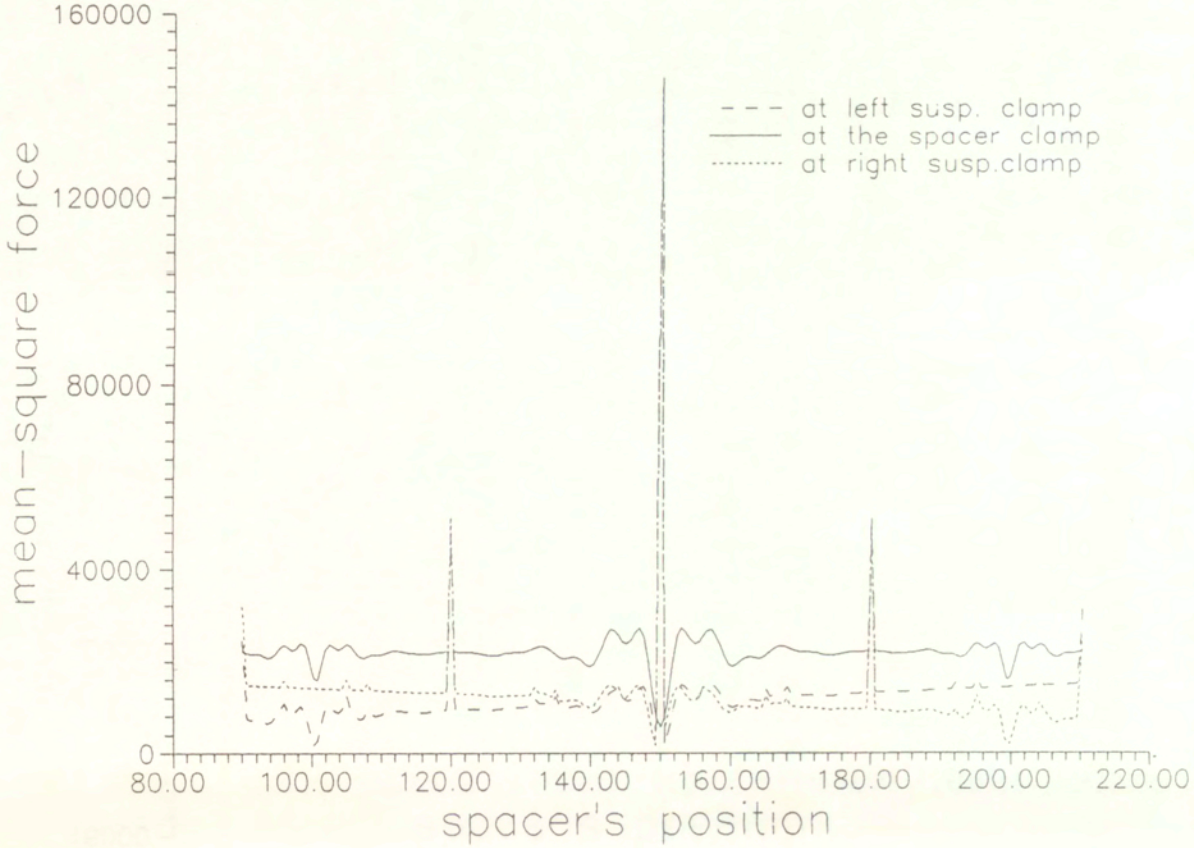

Fig. 18

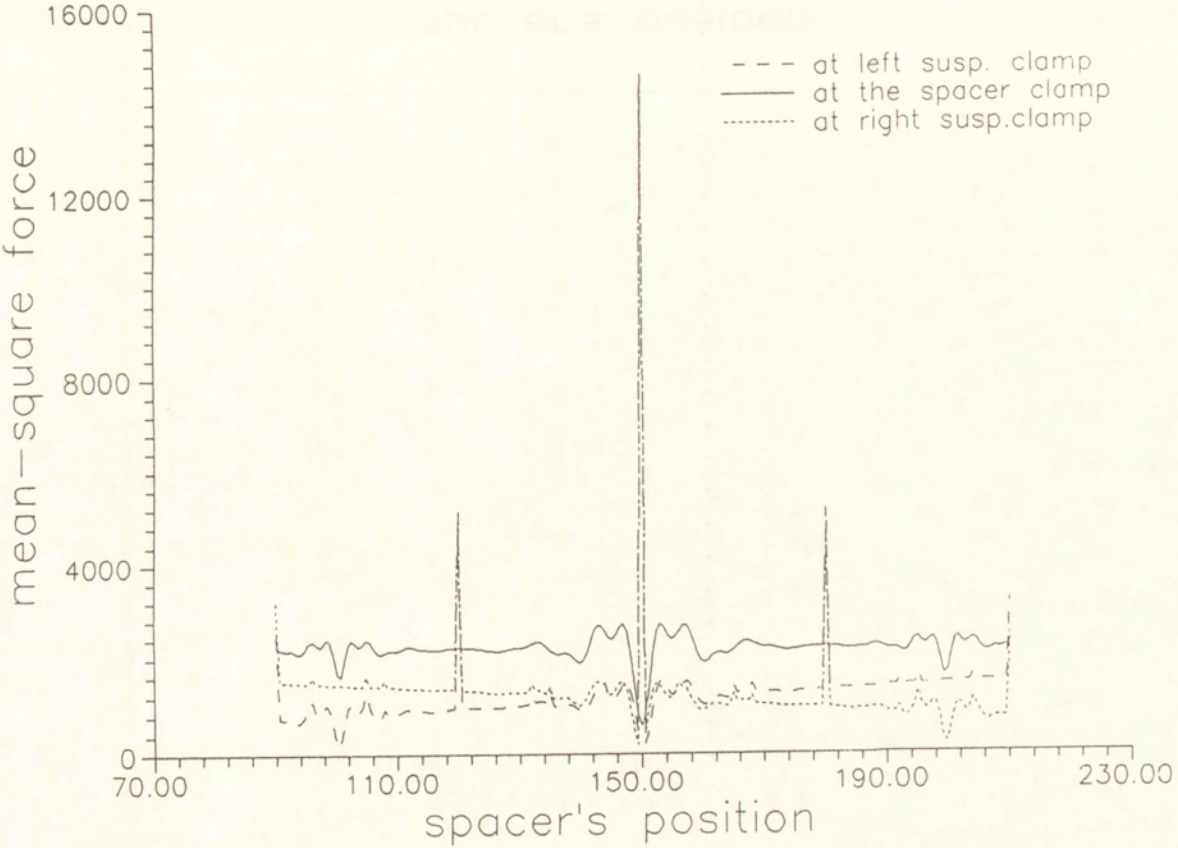

Fig. 19

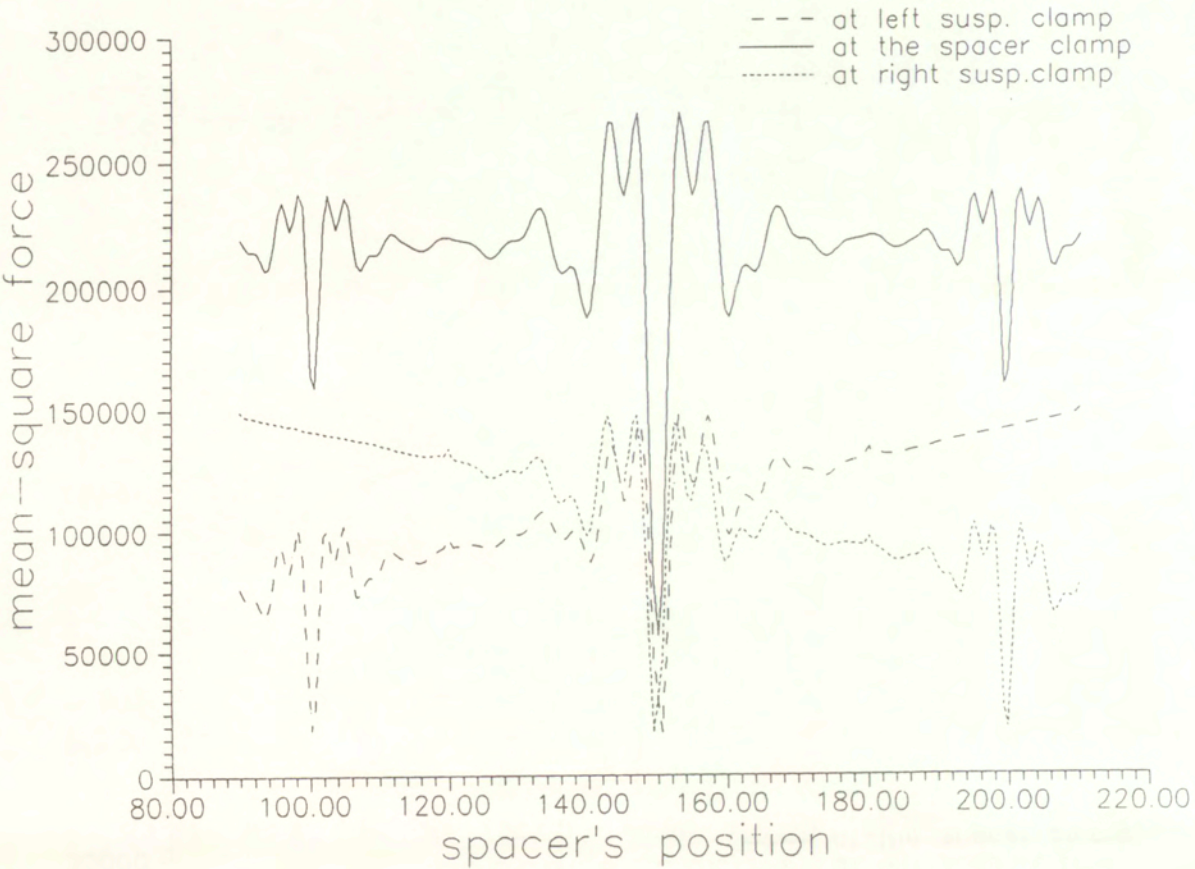

Fig. 20

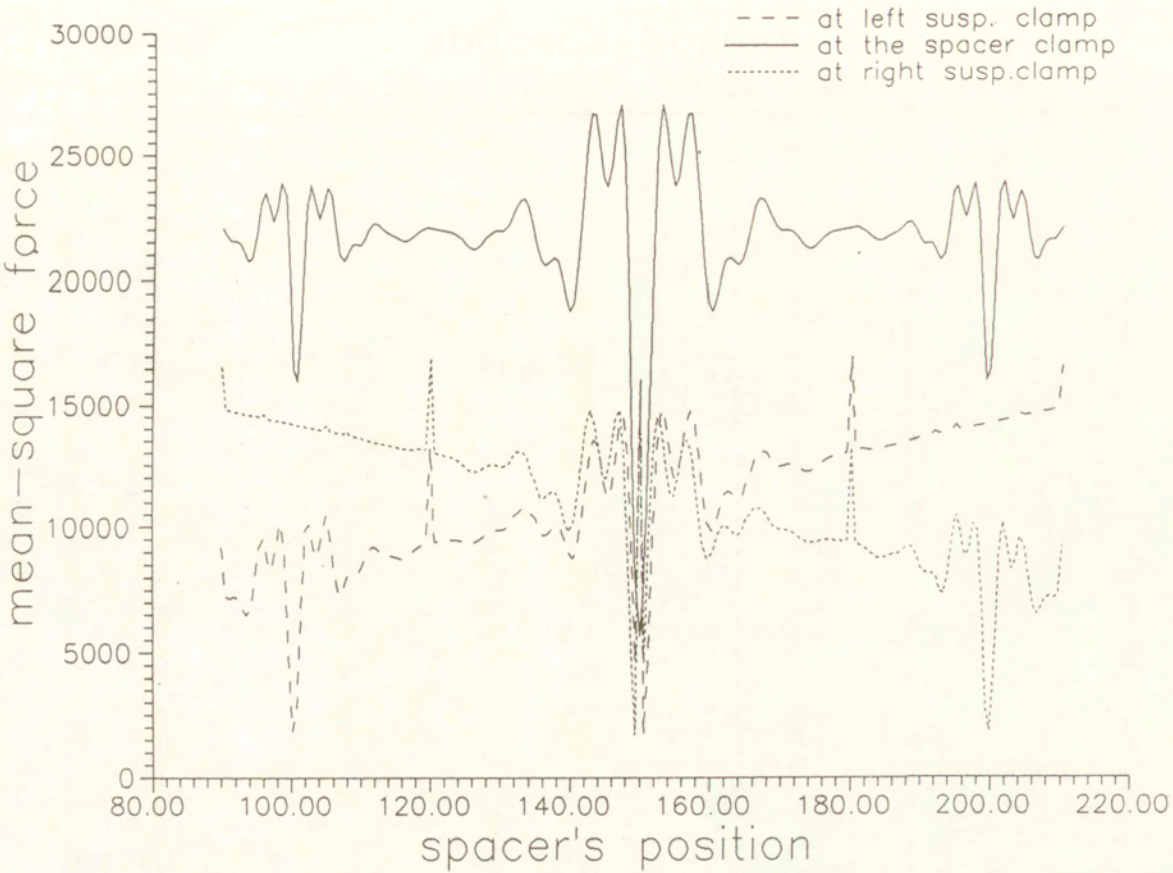

Fig. 21

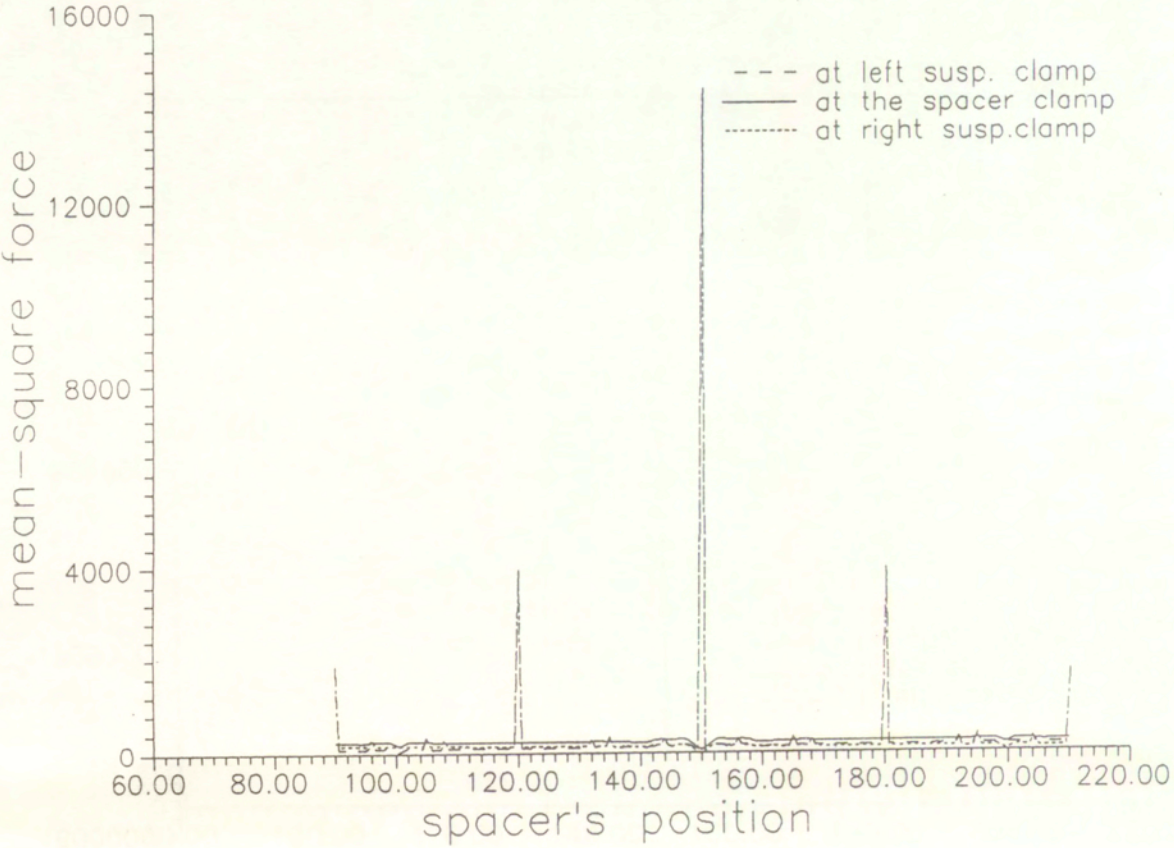

Fig 22

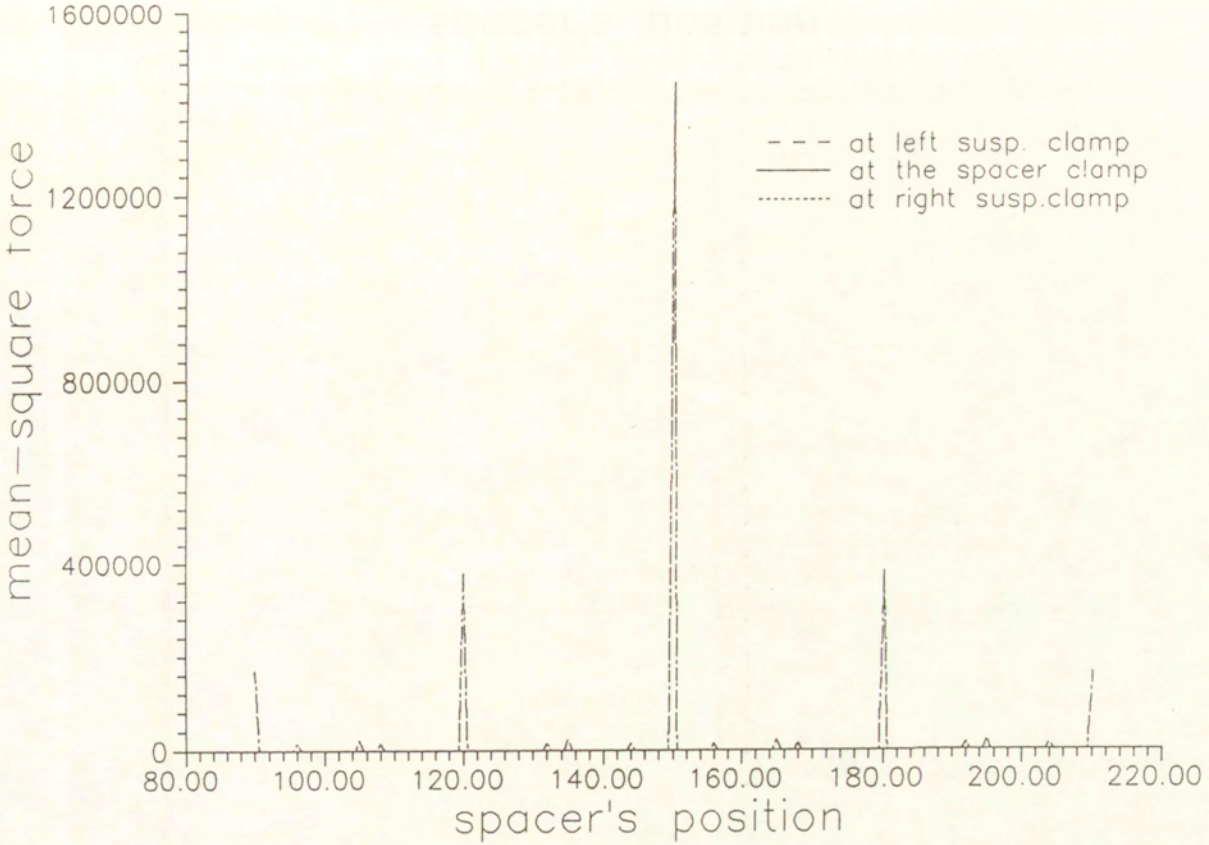

Fig. 23

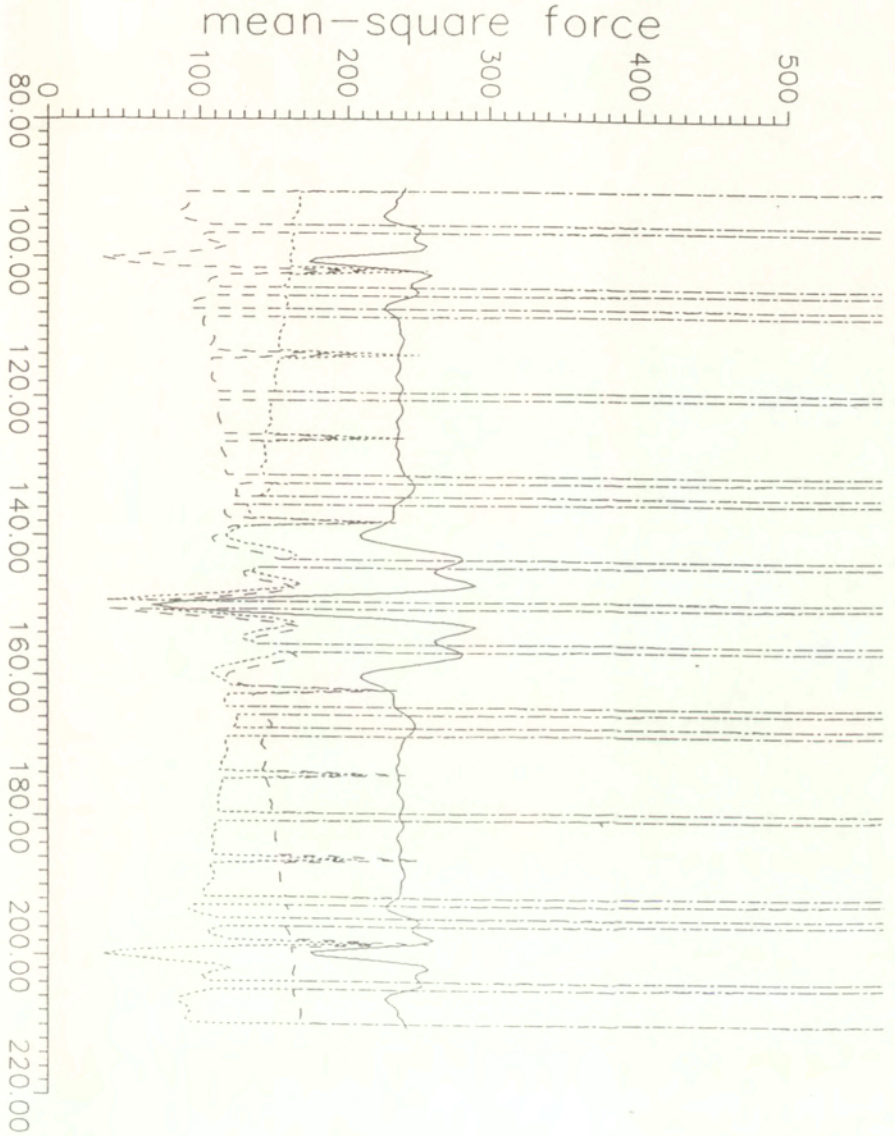

Fig. 24

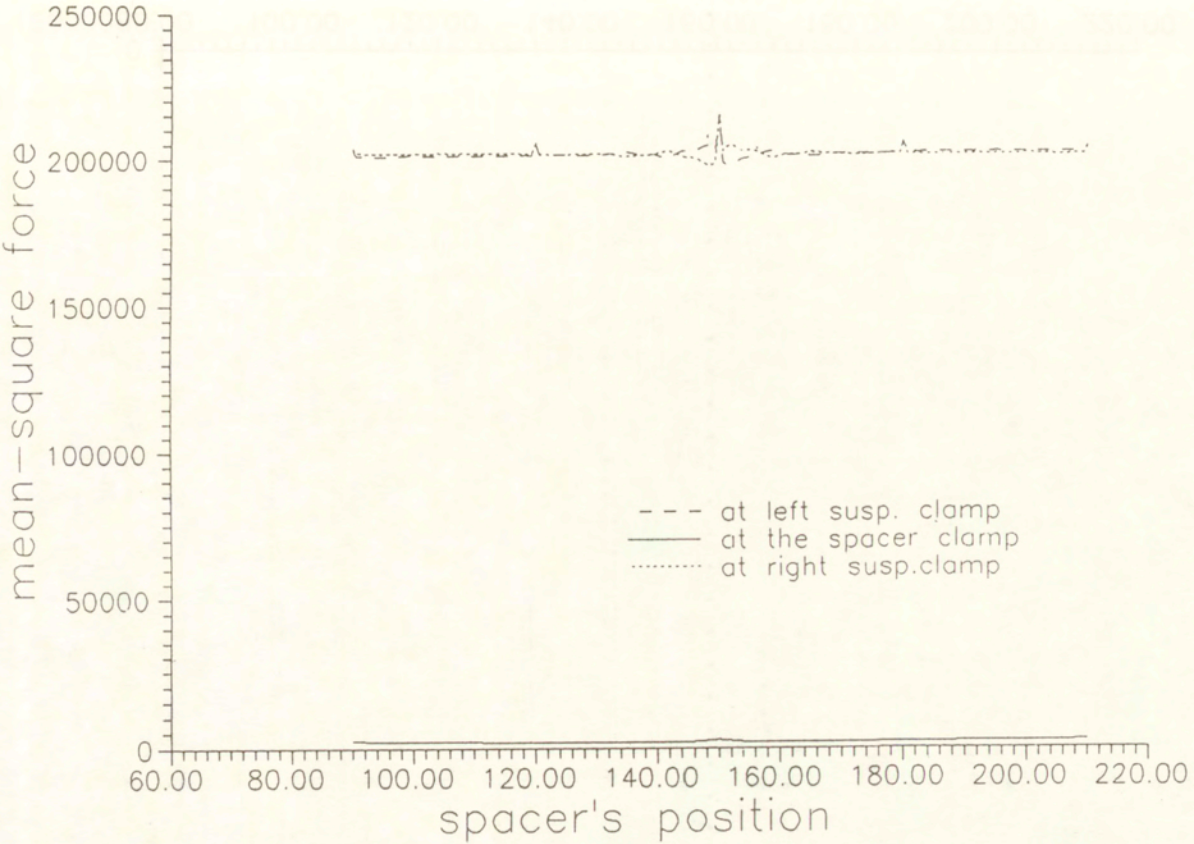

Fig. 25

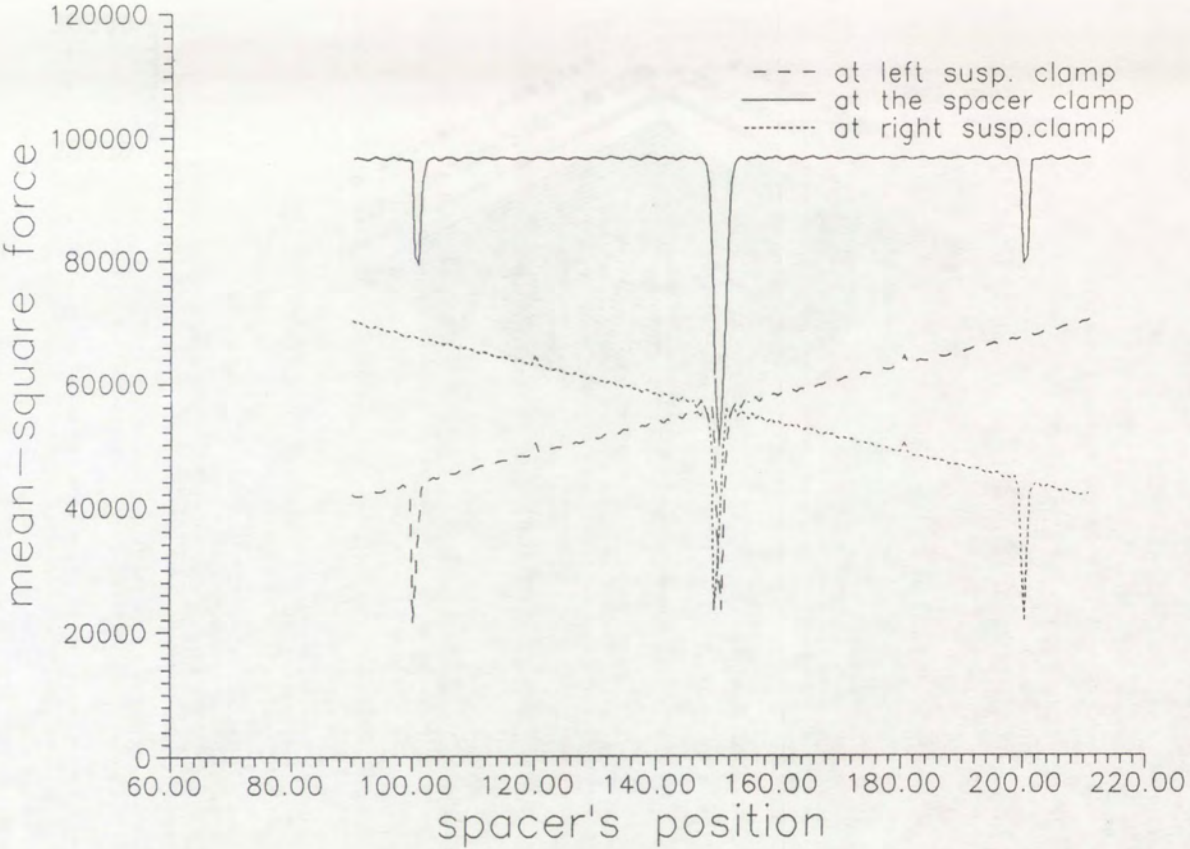

Fig. 26

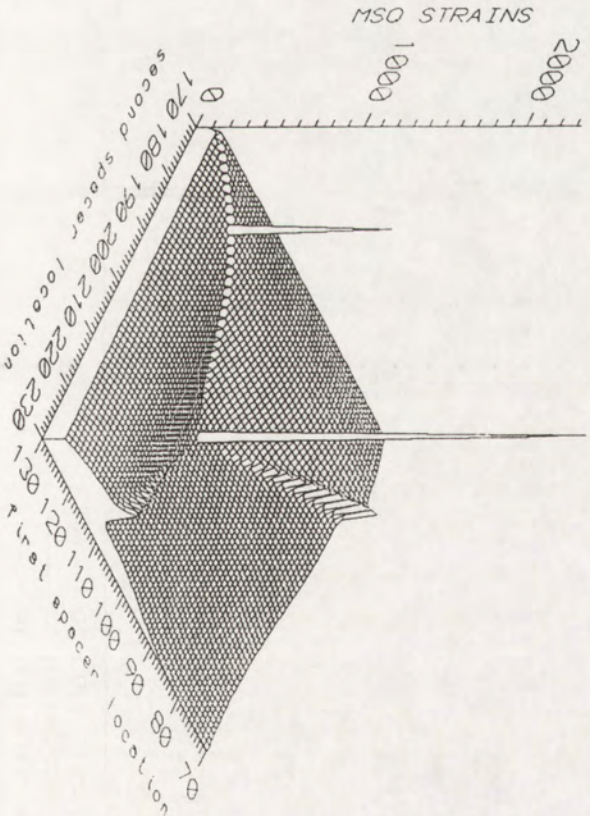

Fig. 27

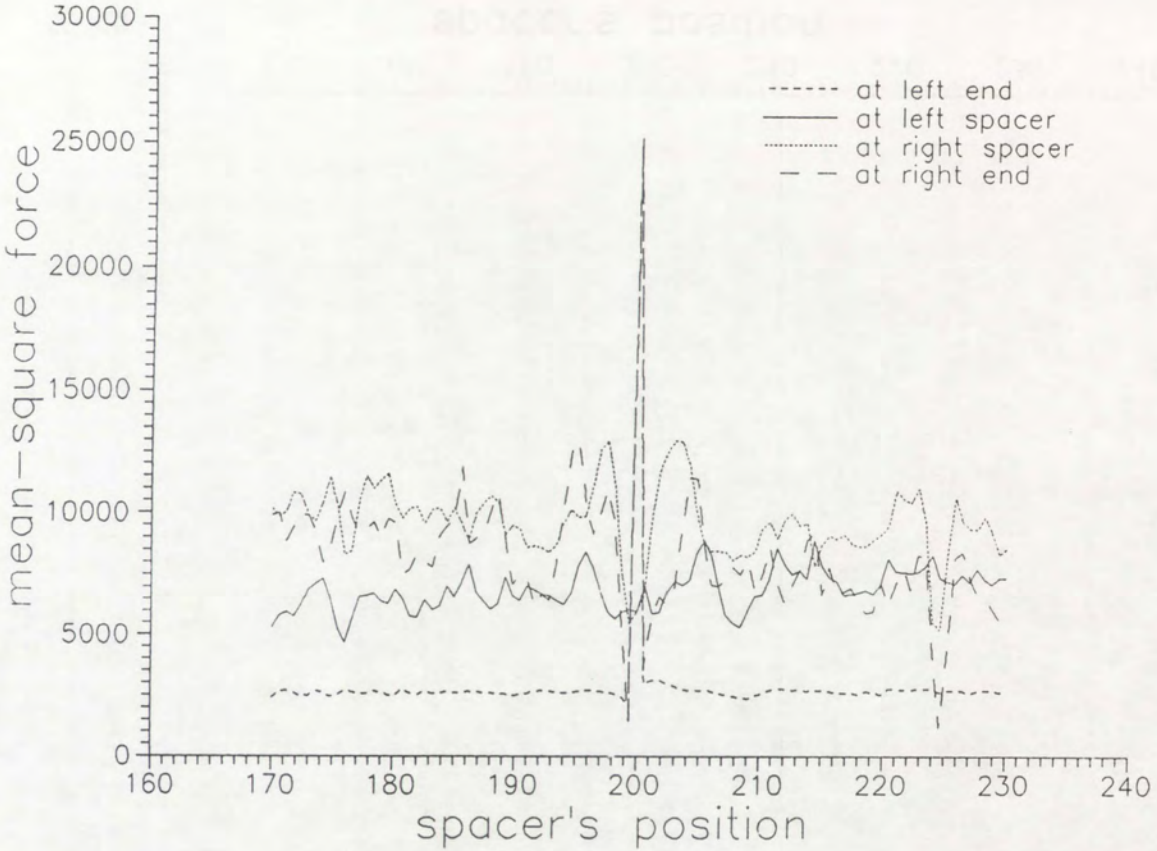

Fig. 28

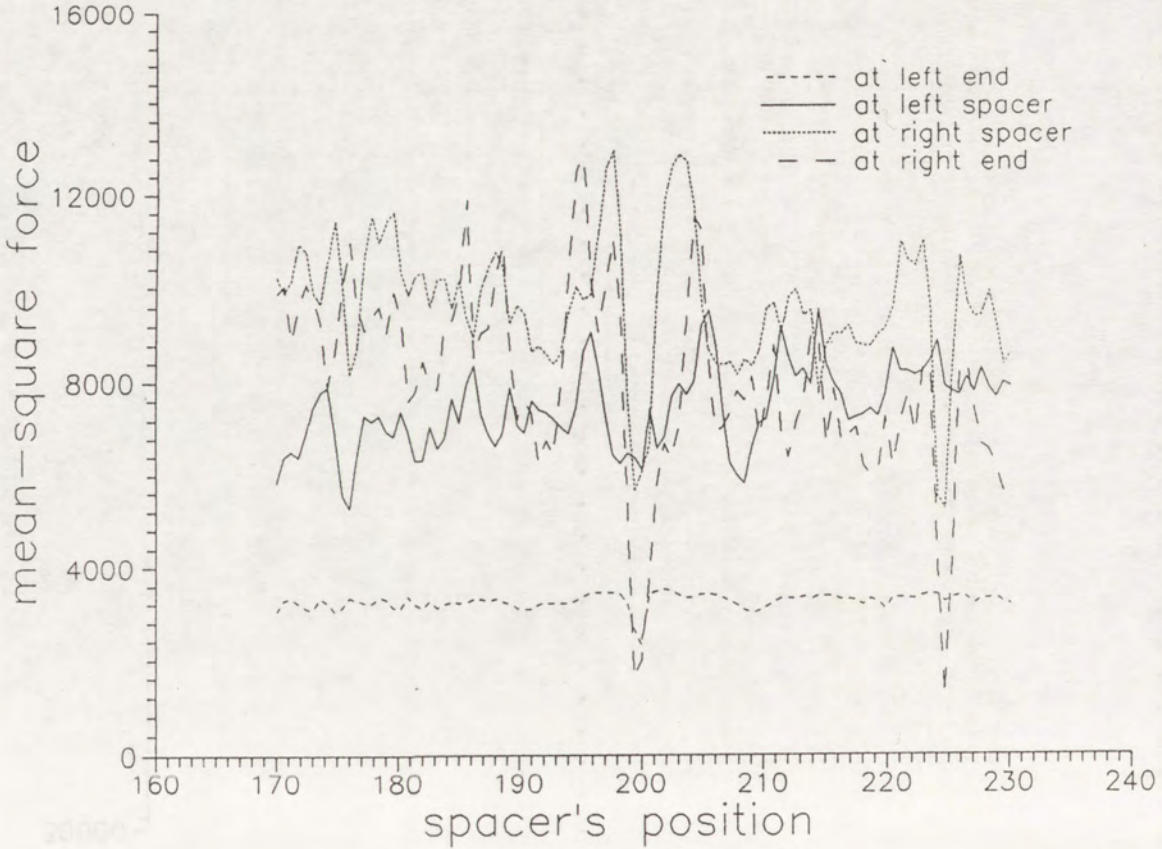

Fig. 29

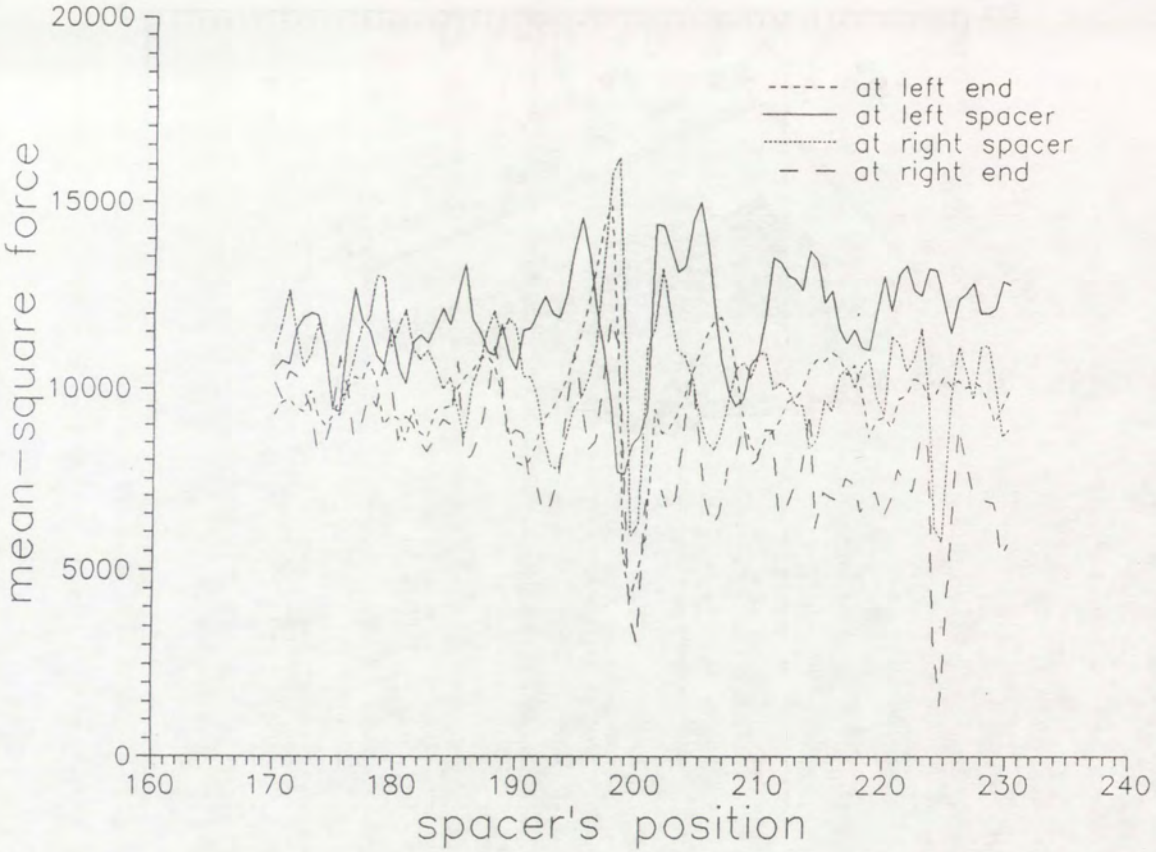

Fig 30

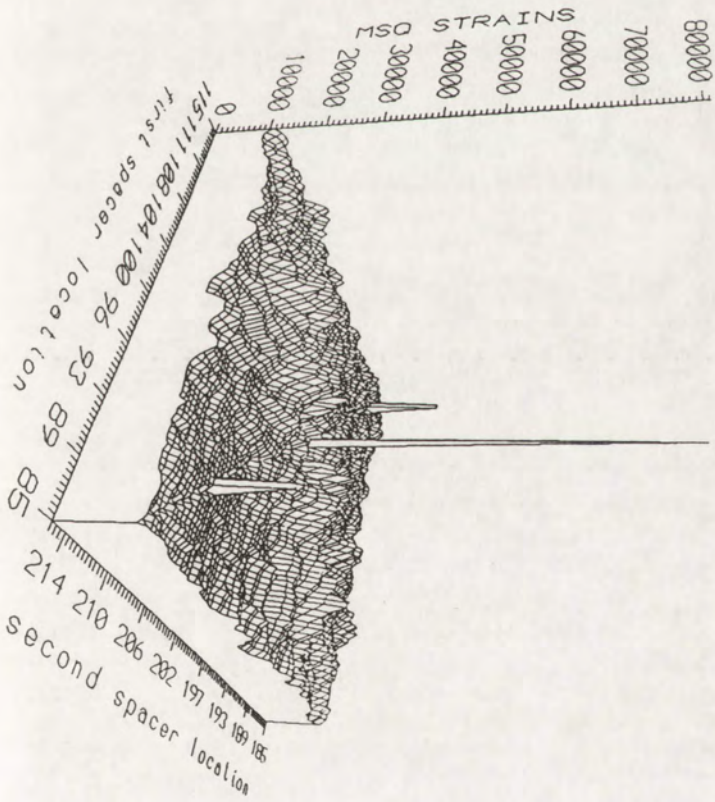

Fig. 31

Fig. 32

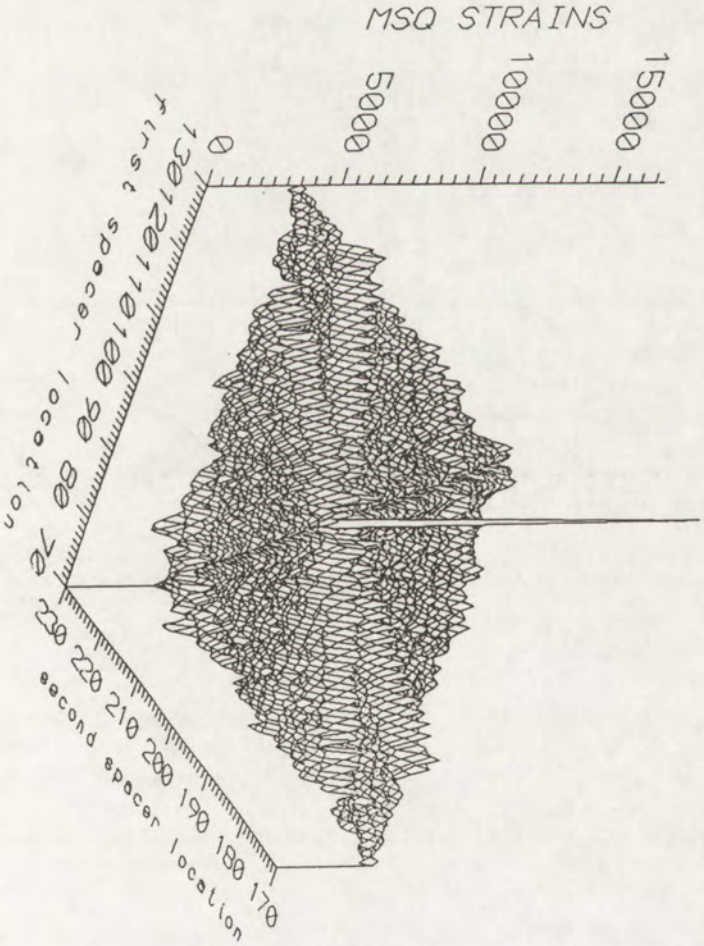

Fig. 33

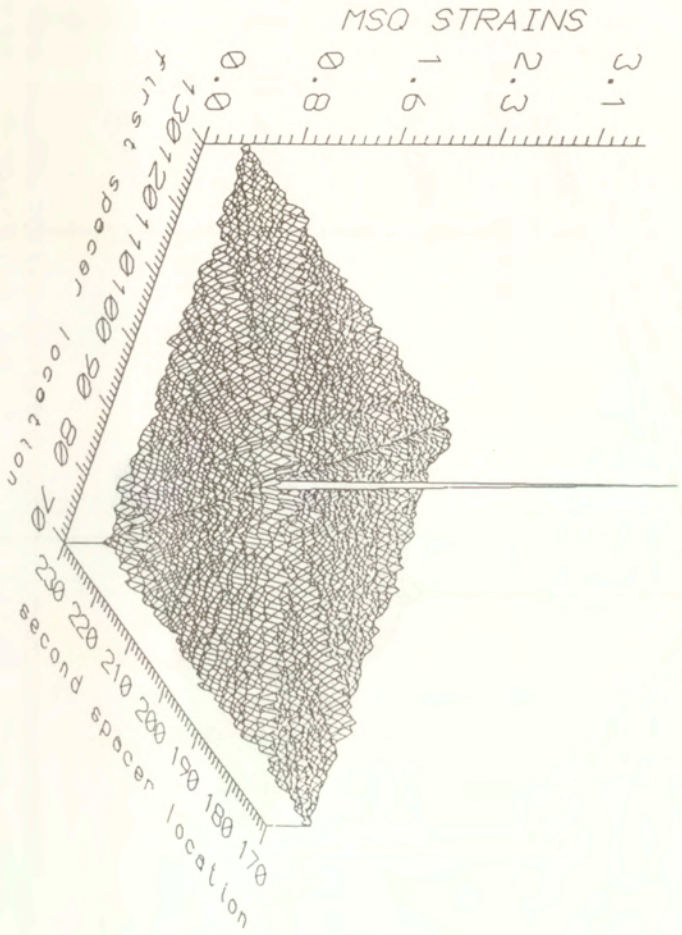

Fig. 34

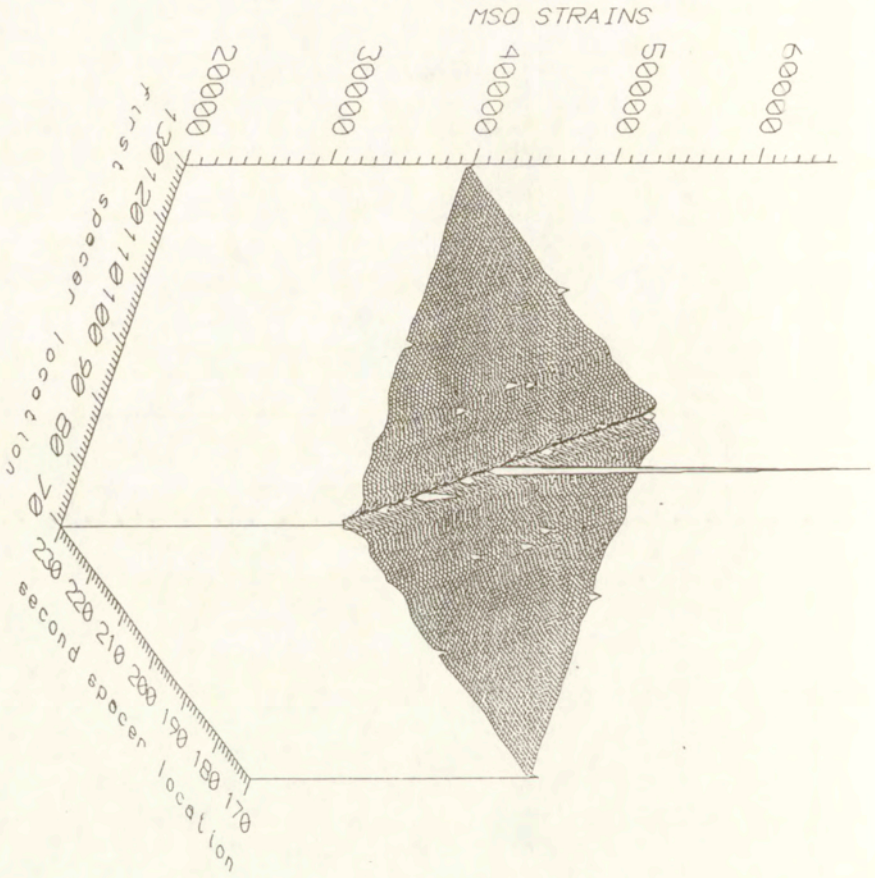

Fig. 35

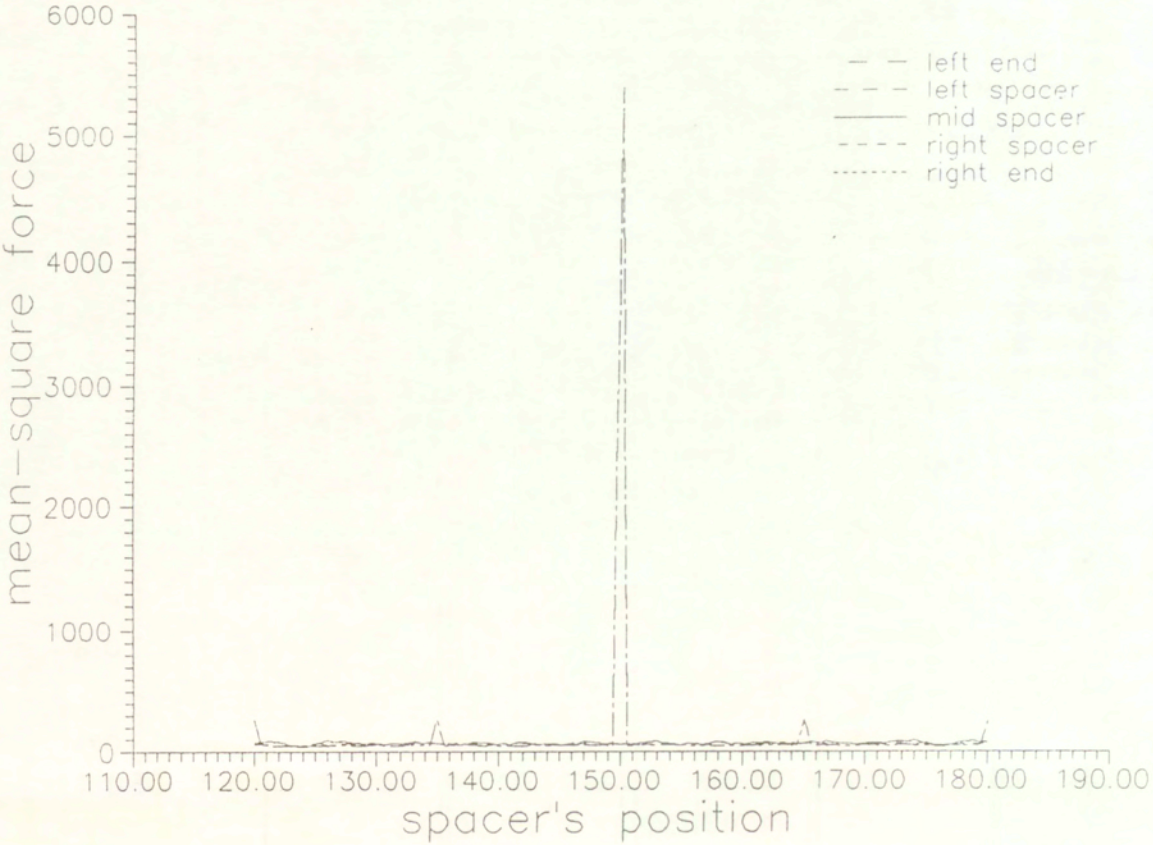

Fig. 36

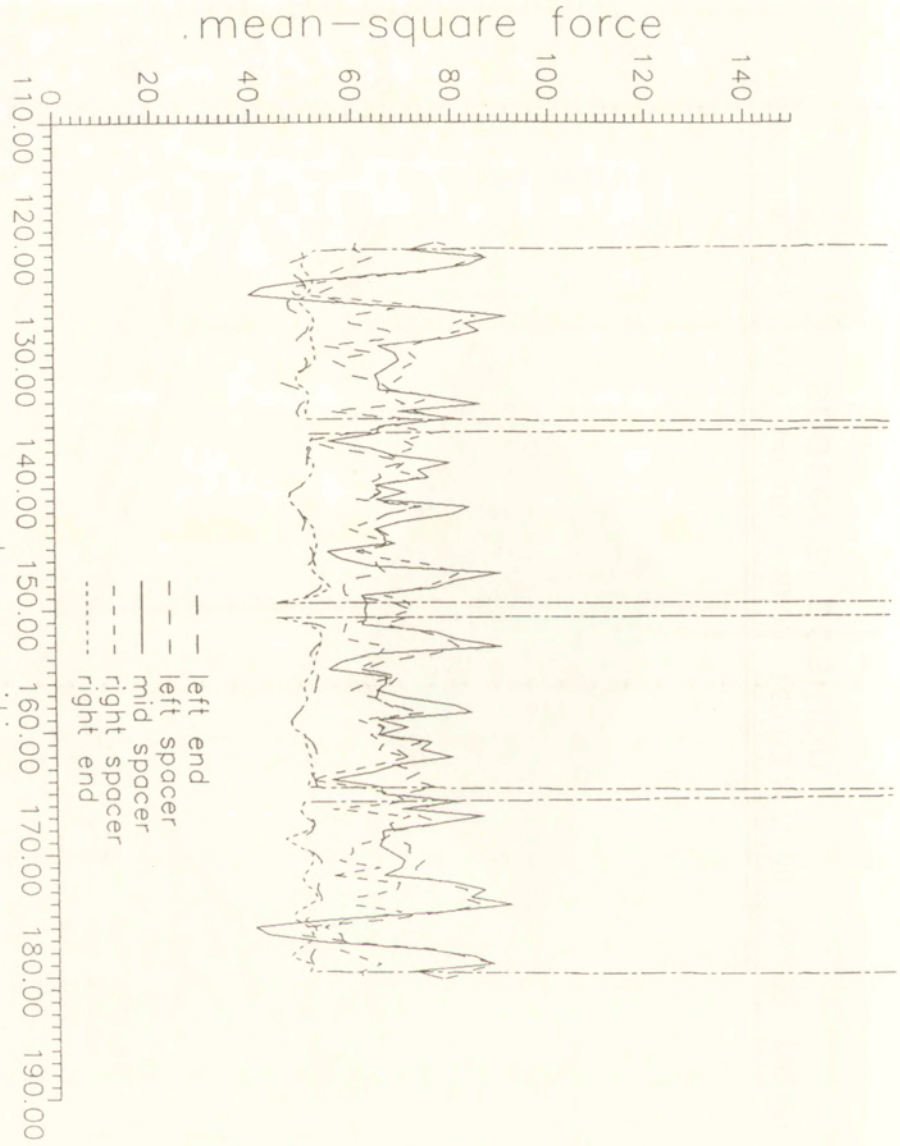

Fig. 37

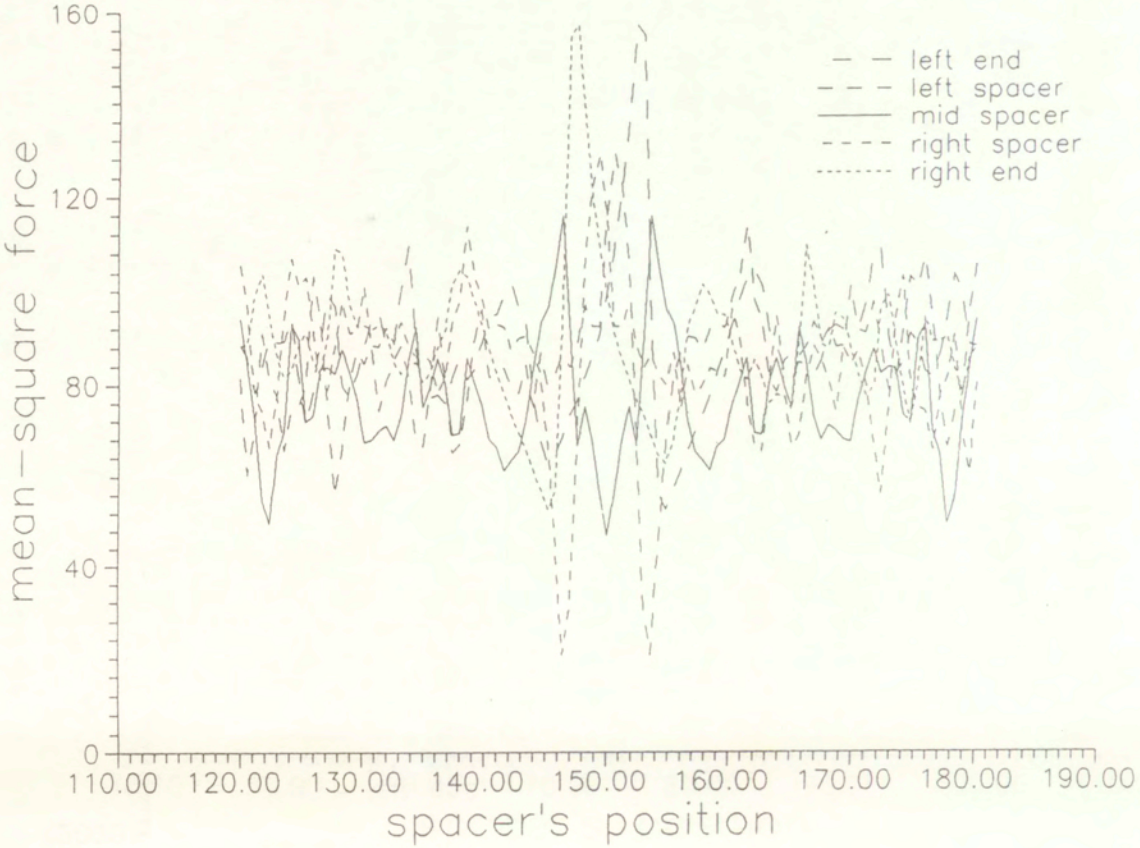

Fig. 38

Fig. 39

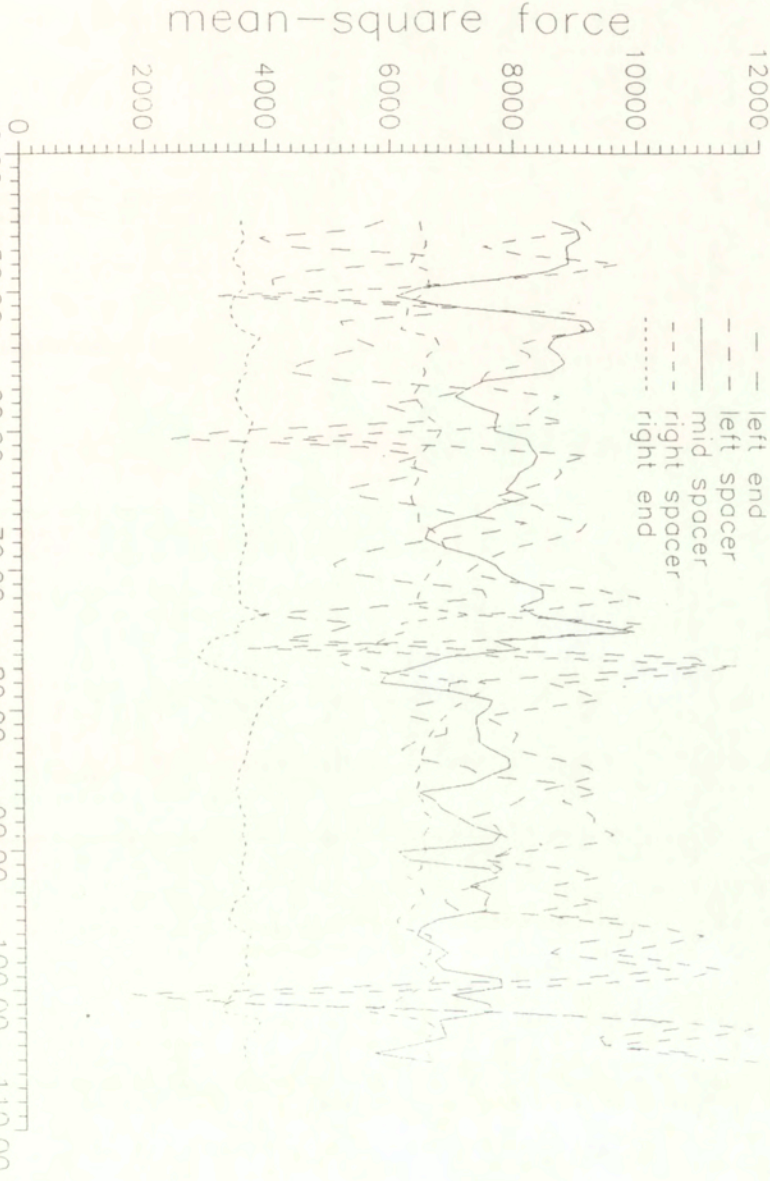

Fig. 40

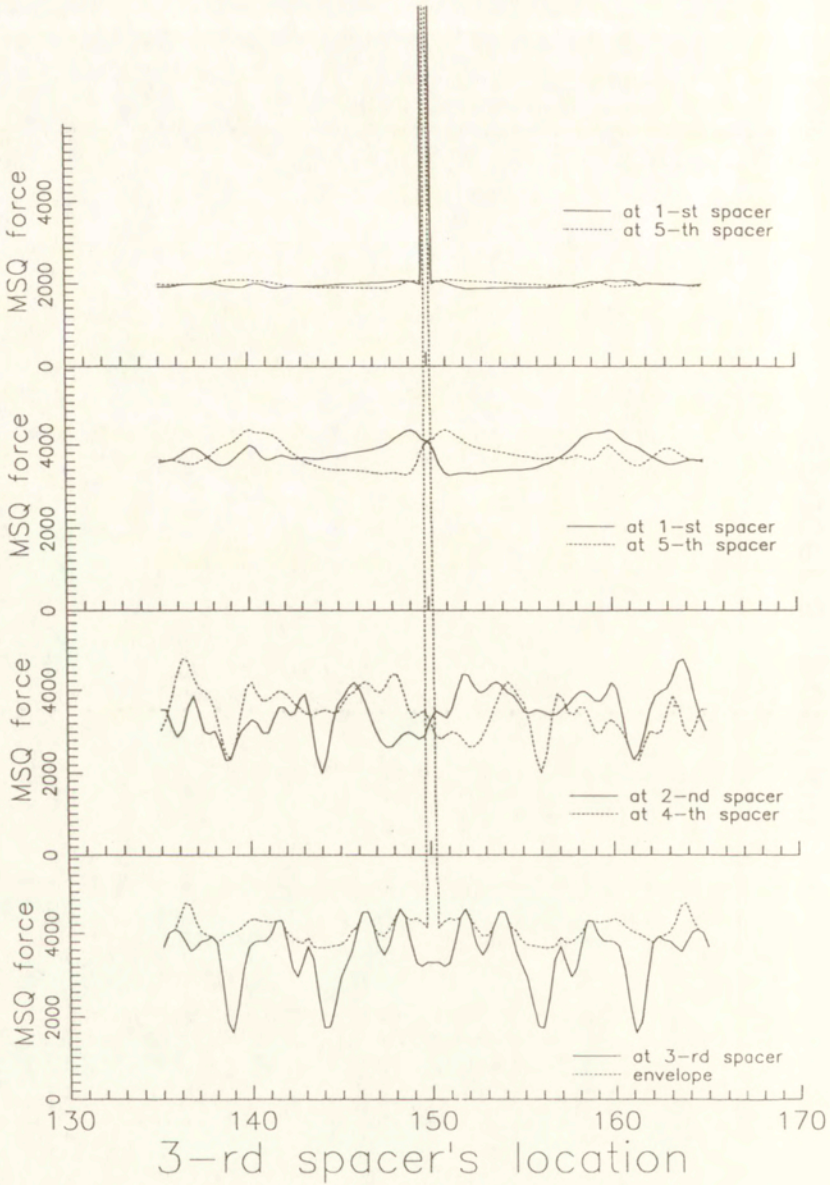

Fig. 41

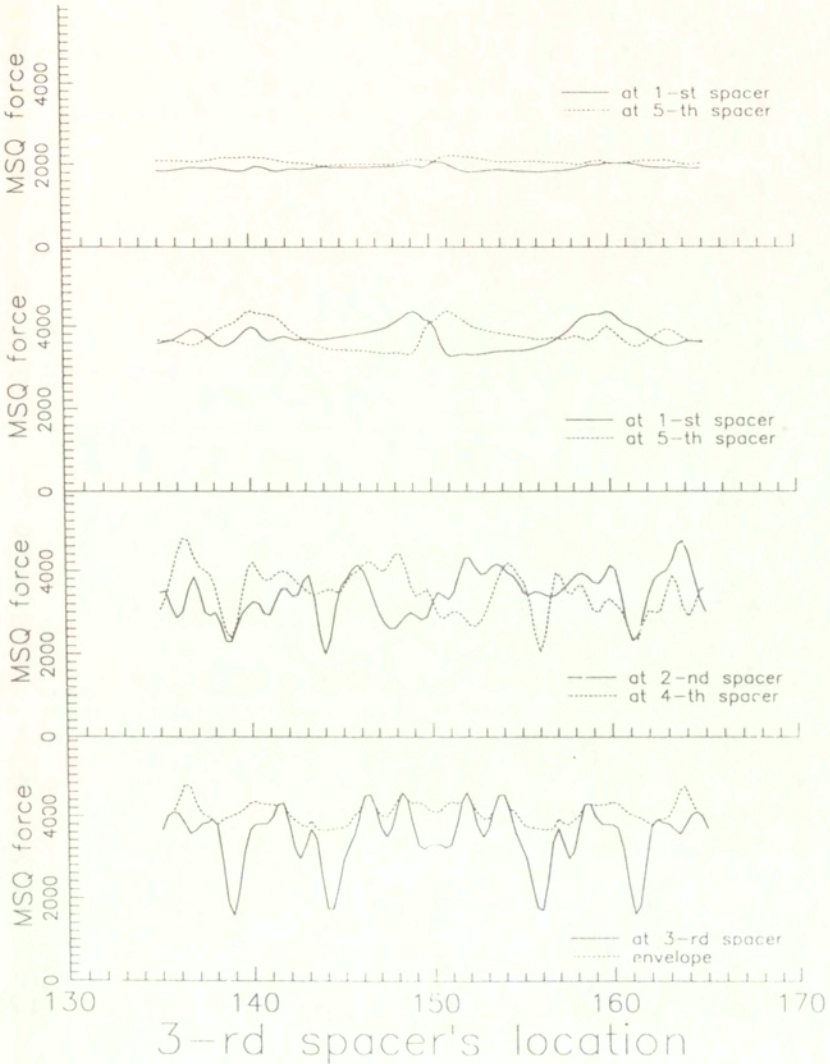

Fig. 42

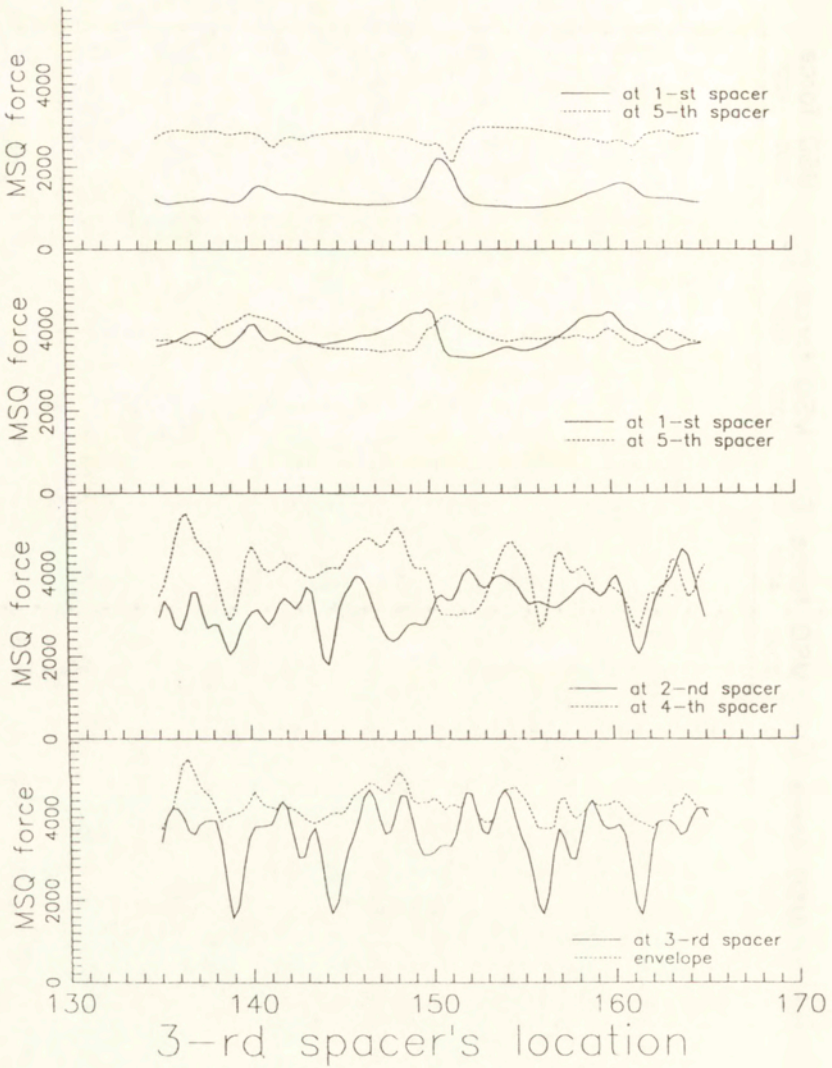

Fig. 43

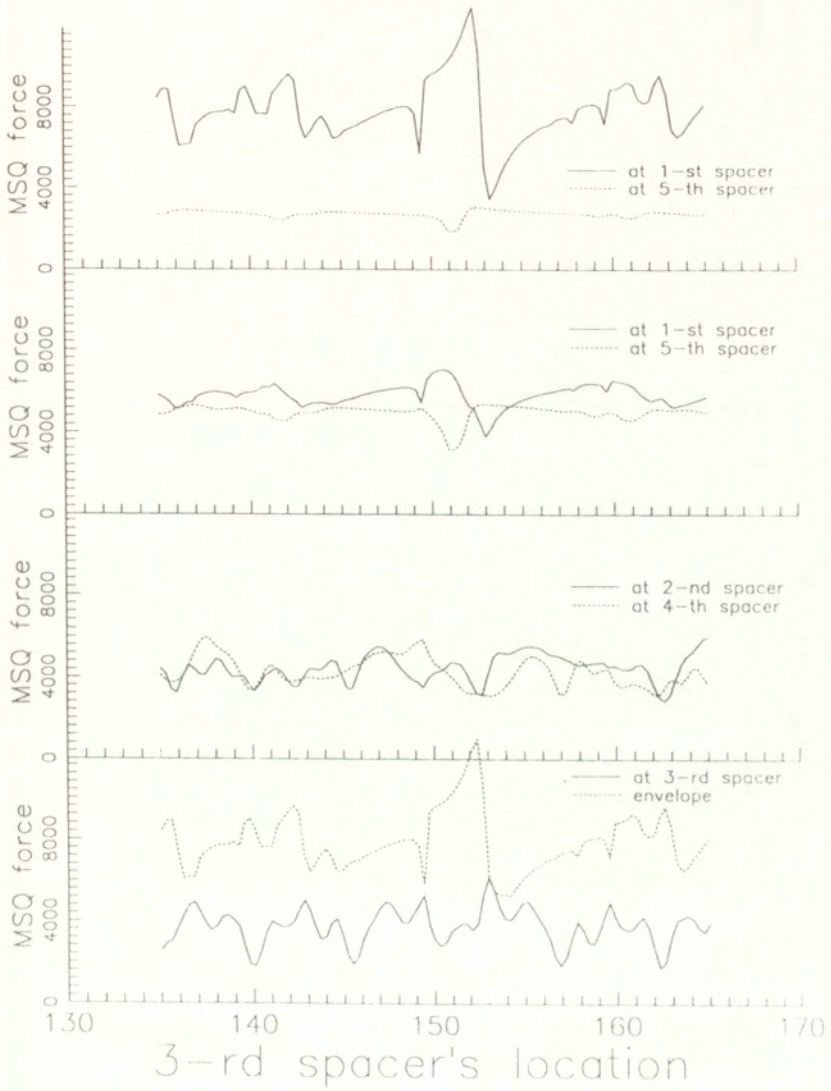

Fig 44

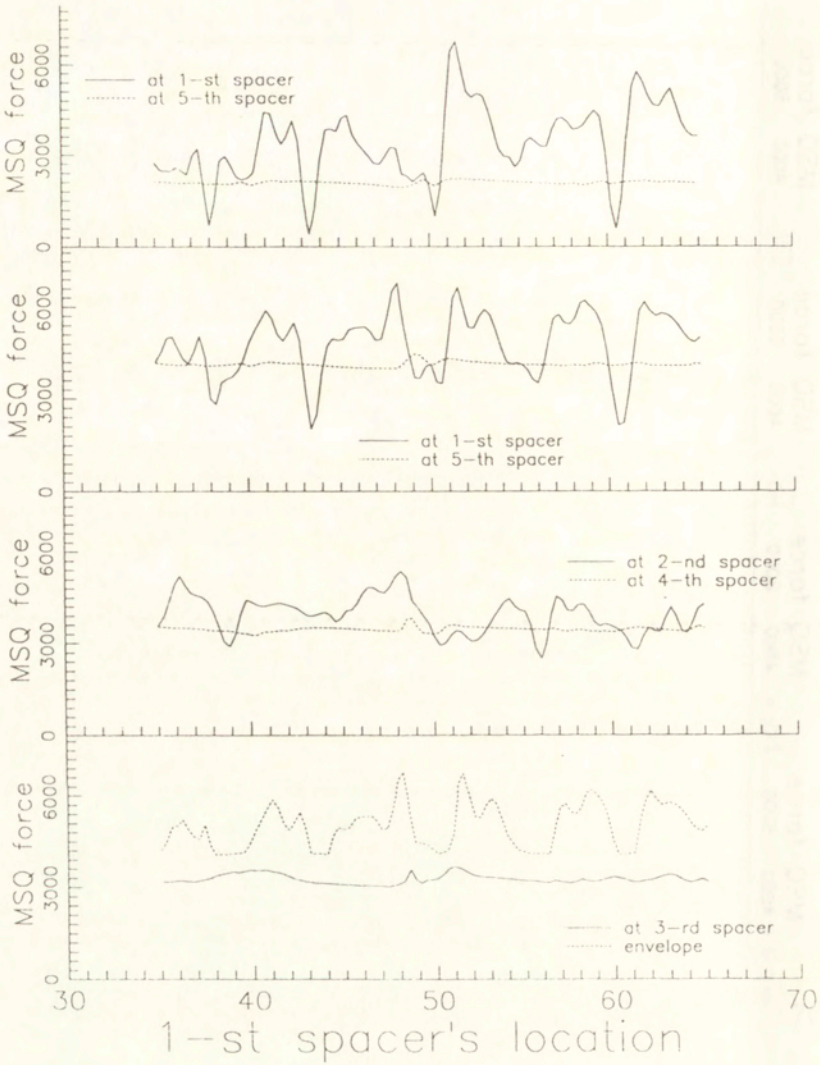

Fig. 45

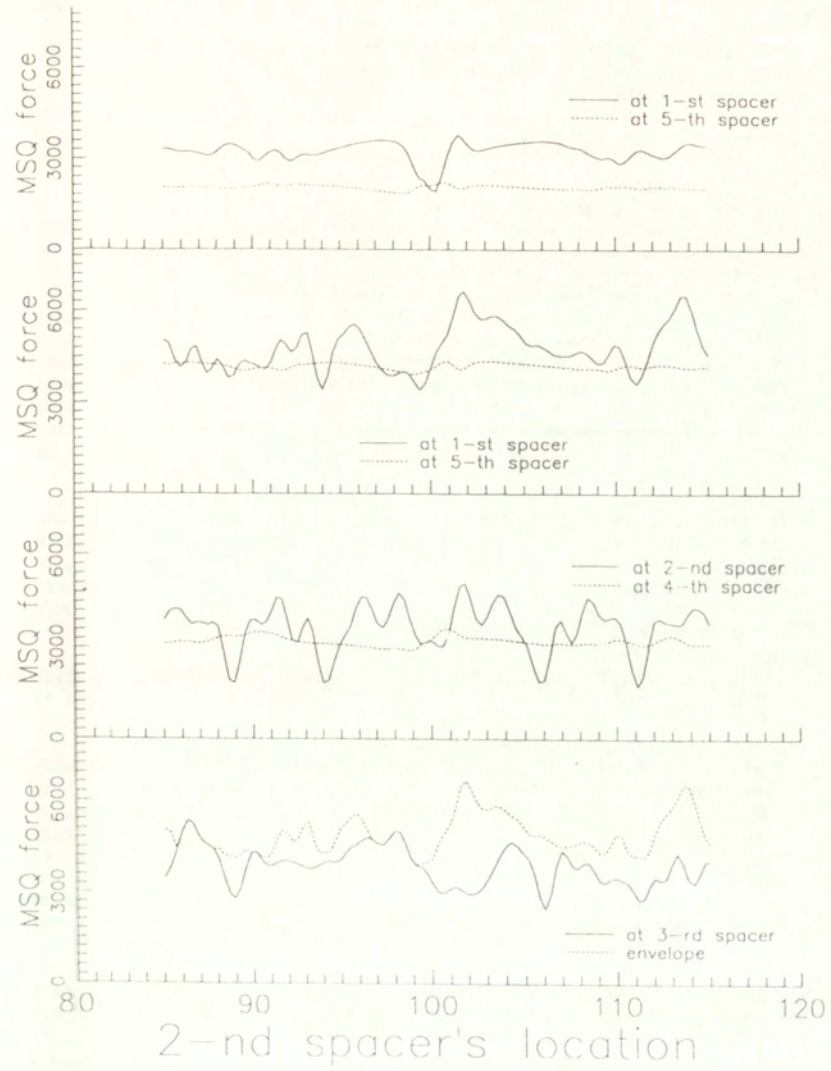

Fig 46

Appendix 1
Listing of the FORTRAN code SPA0.FOR implementing the 0-order-spectral-method

```
C
C     OPTIMIZATION OF PLACEMENT OF SPACERS IN A VERTICAL
C     BUNDLE OF TWO CONDUCTORS VIBRATING UNDER RANDOM
C     WIND-INDUCED LOADING.
C
C     MOST IMPORTANT PARAMETERS:
C
C     N - number of spacers
C     M - number of modes taken into account
C     OTENSION - tensile force in a conductor (in [N])
C     OLENGTH  - cable length (in [m])
C     OSPEGRAV - cable specific gravity (in [N/m])
C     ODSTR    - structural damping coefficient (in [N*s/m])
C     ODAMP    - spacer damping constant (in [N*s/m])
C     OSTIF    - spacer stiffness constant (in [N/m])
C     OINTENS  - intensity of external loading (in [N])
C     OALF     - loading temporal correlation length (in [s])
C     OLAM     - loading spatial correlation length (in [m])
C     ODEL     - loading spatial wave-length (in [m])
C     ONU      - loading temporal wave-length (in [s])
C
C       PROGRAM SPACERS
      IMPLICIT DOUBLE PRECISION(A-H,O-X,Z)
      IMPLICIT COMPLEX(Y)
      DIMENSION SUMM(5)
      DIMENSION EIGVAL(500),EIGVEC(5,500)
      DIMENSION XL(5),D(5)
      DIMENSION YDELK(500),YRP(500),YRM(500)
      COMMON /EIG/EIGVAL,EIGVEC
      COMMON /PHYS/OTENSION,OLENGTH,OSTIF,ODAMP,ODSTR,OSPEGRAV,OINTENS,O
     1ALF,OLAM,ODEL,ONU
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /MSQ/SUMO,SUMF,SUMM
      COMMON /PREC/PREC1,PREC2,IPREC3,P01,P99
      COMMON /OPT/RMS
      COMMON /KINJ/YDELK,YRP,YRM
      COMMON /TRANS/FORTR,FSQTR,DTRANS
      COMMON /HALF/TWO,TWOH
      COMMON /COMPL/YI,YOMPP,YOMMP,YOMPM,YOMMM,YJ
C
      OPEN(1,FILE='MSQPOS',STATUS='UNKNOWN')
      OPEN(2,FILE='EIGVAL',STATUS='UNKNOWN')
      OPEN(3,FILE='EIGVEC',STATUS='UNKNOWN')
      OPEN(4,FILE='MSQOPT',STATUS='UNKNOWN')
      OPEN(7,FILE='LASTRUN',STATUS='UNKNOWN')
C
C
      CALL DIMLESS
```

```
C
C
    1 WRITE(*,10)
      READ(*,*,ERR=13)IANS
      IF ((IANS.LT.1).OR.(IANS.GT.4))GO TO 13
      GO TO (2,3,4,5),IANS
C
C
    2 CALL DATA
      CALL DIMLESS
      GO TO 1
C
C
    3 CALL OPTIM
      GO TO 1
C
C
    4 CALL READPOS
      CALL SPECTRUM
      CALL PRINSPEC
D     STOP
      CALL MSQVALUE
      CALL PRINTMSQ
      GO TO 1
C
C
    5 CONTINUE
      WRITE(*,11)
      READ(*,*,ERR=13)IANS
      IF(IANS.NE.0)GO TO 1
      STOP
C
C
   13 WRITE(*,12)
      GO TO 1
C
C
   10 FORMAT(/////////'   PROGRAM      S P A C E R S    TO YOUR DISPOSAL'
     1,/////////,' CHOOSE THE TASK:',/,' 1-CHANGE/SET PARAMETERS',/,' 2-
     2OPTIMIZATION OF SPACERS PLACEMENT',/,' 3-MSQ VALUE OF BENDING STRE
     3SSES AT PRESCRIBED NODAL POINTS',/,' 4-QUIT',/////,' TYPE-IN YOUR
     4CHOICE :  '$)
   11 FORMAT(///' DO YOU REALLY WANT TO QUIT? (HIT 0 TO CONFIRM)      ')
   12 FORMAT(//' INCORRECT ANSWER, TRY AGAIN!')
      END
C
C
      SUBROUTINE SPECTRUM
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION SUMM(5)
      DIMENSION EIGVEC(5,500),EIGVAL(500)
      DIMENSION XL(5),D(5)
      COMMON /EIG/EIGVAL,EIGVEC
```

```
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /MSQ/SUMO,SUMF,SUMM
      COMMON /PREC/PREC1,PREC2,IPREC3,P01,P99
      PRECOLD=PREC1
      WRITE(*,107)
      DQC=.1DO*PI
      PRE=1.D-10
      DQ=DQC
      NEIG=0
C      OPEN(99,FILE='deter')
      Q=.1DO*PI
      CALL DETERM(DETP)
      Q=.2DO*PI
      CALL DETERM(DET)
    1 S=(DET-DETP)/DQ
      DQ=DQC*P01*(ZJ-P99/(ZJ+DET*DET))/(ZJ-P99/(ZJ+S*(S-DSIGN(S,DET))))
      Q=Q+DQ
      DETP=DET
      CALL DETERM(DET)
C      WRITE(99,*)Q,DET
      IF(DET*DETP.GT.ZO)GO TO 1
      IF(DET*DET.LE.PREC1)GO TO 4
      QR=Q
      QL=Q-DQ
    3 Q=(QL+QR)/2.
      IF((Q.EQ.QR).OR.(Q.EQ.QL))GO TO 7
      CALL DETERM(DETL)
      IF(DETL*DET.LT.ZO)THEN
                          QL=Q
                          GO TO 3
                        ENDIF
      IF(DETL*DETL.LE.PREC1)GO TO 4
      QR=Q
      DET=DETL
      GO TO 3
    4 NEIG=NEIG+1
C
      S=DSIN(Q)**2
      do 41 II=1,N
   41 S=S+D(II)**2
      IF(S*S.GT.PRE)THEN
C
      CALL EIGEN
      EIGVAL(NEIG)=Q
      DO 5 I=1,N
    5 EIGVEC(I,NEIG)=D(I)
C
      ELSE
C
      EIGVAL(NEIG)=-Q
      DO 6 I=1,N
```

```
    6 EIGVEC(I,NEIG)=ZO
C
      ENDIF
C
      IF (NEIG.EQ.M) GO TO 999
      DETP=DETL
      Q=Q+DQ
      CALL DETERM(DET)
      GO TO 1
    7 PREC1=DMIN1(DETL*DETL*10.D0,.01D0)
      WRITE(*,207)PREC1
      GO TO 4
  999 PREC1=PRECOLD
      RETURN
  107 FORMAT(/,' CALCULATING THE SPECTRUM')
  207 FORMAT(/////////////' THE PRESCRIBED PRECISION CANNOT BE ACHIEVED!'
     1,////,'  THE BEST POSSIBLE PRECISION IS  ',D20.12,//////////)
      END
C
C
      SUBROUTINE DETERM(DET)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION XL(5),D(5)
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      S=DSIN(Q)
      TKAPQ=TKAP*Q
      QI=Q*XL(1)
      SI=DSIN(QI)
      BI=DSIN(Q-QI)
      SUM=BI*SI
      D(1)=SI
      IF(N.EQ.1)GO TO 2
      CI=DCOS(QI)
      ROBA=SI*SI
      ROBB=CI*SI
      DO 1 I=2,N
      QI=Q*XL(I)
      SI=DSIN(QI)
      CI=DCOS(QI)
      BI=DSIN(Q-QI)
      DI=SI+(CI*ROBA-SI*ROBB)/TKAPQ
      ROBA=SI*DI+ROBA
      ROBB=CI*DI+ROBB
      SUM=BI*DI+SUM
    1 D(I)=DI
    2 DET=SUM-TKAPQ*S
      RETURN
      END
C
C
      SUBROUTINE EIGEN
```

```
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION EIGVEC(5,500),EIGVAL(500)
      DIMENSION XL(5),D(5)
      COMMON /EIG/EIGVAL,EIGVEC
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      TWO=2.DO
      C=TWO*DCOS(Q)
      QLI=Q*XL(1)
      QMQLI=Q-QLI
      SAI=DSIN(QLI)
      SBI=DSIN(QMQLI)
      CAI=DCOS(QLI)
      CBI=DCOS(QMQLI)
      DI=D(1)
      DISAI=DI*SAI
      DISBI=DI*SBI
      SAICAI=SAI*CAI
      SBICBI=SBI*CBI
      SUMA=DISBI*DISBI*(QLI-SAICAI)+DISAI*DISAI*(QMQLI-SBICBI)
      SUMB=ZO
      SUMC=ZO
      IF(N.EQ.1)GO TO 2
      ROBA=DISBI*(QLI-SAICAI)
      ROBB=DISAI*QLI
      ROBC=DISAI
      ROBD=DISAI*(SBI*CAI-CBI*SAI)
      DO 1 I=2,N
      QLI=Q*XL(I)
      QMQLI=Q-QLI
      SAI=DSIN(QLI)
      SBI=DSIN(QMQLI)
      CAI=DCOS(QLI)
      CBI=DCOS(QMQLI)
      DI=D(I)
      DISAI=DI*SAI
      DISBI=DI*SBI
      SAICAI=SAI*CAI
      SBICBI=SBI*CBI
      SUMA=DISBI*DISBI*(QLI-SAICAI)+DISAI*DISAI*(QMQLI-SBICBI)+SUMA
      SC=SBI*CAI-CBI*SAI
      SUMB=SUMB+DISBI*(TWO*ROBA+C*(ROBB-QLI*ROBC)-SC*ROBC+ROBD)
      SUMC=SUMC+DISAI*(QMQ LI-SBICBI)*ROBC
      ROBA=DISBI*(QLI-SAICAI)+ROBA
      ROBB=DISAI*QLI+ROBB
      ROBC=DISAI+ROBC
    1 ROBD=DISAI*SC+ROBD
    2 CNOR=SUMA+SUMB+TWO*SUMC
      CNOR=DSQRT(TWO*Q/CNOR)
      DO 3 I=1,N
    3 D(I)=D(I)*CNOR
      RETURN
```

```
      END
C
C
      SUBROUTINE OPTIM
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION XLOPT(5),SUMM(5),SUMMOPT(5)
      DIMENSION EIGVEC(5,500),EIGVAL(500)
      DIMENSION XL(5),D(5)
      COMMON /EIG/EIGVAL,EIGVEC
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /MSQ/SUMO,SUMF,SUMM
      COMMON /PREC/PREC1,PREC2,IPREC3,P01,P99
      COMMON /OPT/RMS
      COMMON /TRANS/FORTR,FSQTR,DTRANS
      COMMON /PHYS/OTENSION,OLENGTH,OSTIF,ODAMP,ODSTR,OSPEGRAV,OINTENS,O
     1ALF,OLAM,ODEL,ONU
    1 WRITE(*,111)
  111 FORMAT(//////,' THIS IS THE TABULATING - OPTIMIZING UNIT',////,
     1' I WILL DISPLAY THE RESULTS ON THE SCREEN. IF YOU WANT ME TO WRIT
     2E',/,' ADDITIONALLY THE MSQ-VALUES OF THE FORCES TO A FILE THEN HI
     3T 1.',/,' OTHERWISE HIT 0 (for screen output only)    ')
      READ(*,*,ERR=1)IANS
      IF ((IANS.NE.0).AND.(IANS.NE.1)) GO TO 1
    3 FORMAT(23(E14.6,1X))
      IF(N.GT.2)GO TO 300
      IF(N.EQ.2)GO TO 200
      DD=PREC2/DBLE(IPREC3)
      XL1=XL(1)-PREC2
      XL(1)=XL1
      CALL SPECTRUM
      CALL MSQVALUE
      XLOPT(1)=XL(1)
      SUMOOPT=SUMO
      SUMFOPT=SUMF
      SUMMOPT(1)=SUMM(1)
      RMS=DMAX1(SUMO,SUMM(1),SUMF)
      RMSM=RMS
      CALL SCREEN
      IF(IANS.EQ.1)THEN
                  WRITE(4,3)XL(1)*OLENGTH,RMS*FSQTR,SUMO*FSQTR,SUMM(1)*
     1FSQTR,SUMF*FSQTR
      WRITE(4,3)(ZJ-XL(1))*OLENGTH,RMS*FSQTR,SUMF*FSQTR,SUMM(1)*FSQTR,
     1SUMO*FSQTR
                     ENDIF
      DO 5 I=1,IPREC3
      XL1=XL1+DD
      XL(1)=XL1
      CALL SPECTRUM
      CALL MSQVALUE
      RMS=DMAX1(SUMO,SUMF,SUMM(1))
      CALL SCREEN
```

```
      IF(IANS.EQ.1)THEN
                  WRITE(4,3)XL(1)*OLENGTH,RMS*FSQTR,SUMO*FSQTR,SUMM(1)*
     1FSQTR,SUMF*FSQTR
      WRITE(4,3)(ZJ-XL(1))*OLENGTH,RMS*FSQTR,SUMF*FSQTR,SUMM(1)*FSQTR,
     1SUMO*FSQTR
                   ENDIF
      IF (RMS.GE.RMSM) GO TO 5
      RMSM=RMS
      XLOPT(1)=XL(1)
      SUMOOPT=SUMO
      SUMFOPT=SUMF
      SUMMOPT(1)=SUMM(1)
    5 CONTINUE
      GO TO 998
  200 DD=PREC2/DBLE(IPREC3)
      IF(IPREC3.GE.50)THEN
      XL1=XL(1)
      ELSE
      XL1=XL(1)-PREC2
      ENDIF
      XL2=XL(2)-PREC2
      XL(1)=XL1
      XL(2)=XL2
      CALL SPECTRUM
      CALL MSQVALUE
      RMS=DMAX1(SUMO,SUMM(1),SUMM(2),SUMF)
      CALL SCREEN
      XLOPT(1)=XL(1)
      XLOPT(2)=XL(2)
      SUMOOPT=SUMO
      SUMFOPT=SUMF
      SUMMOPT(1)=SUMM(1)
      SUMMOPT(2)=SUMM(2)
      RMSM=RMS
      IF(IANS.EQ.1)WRITE(4,3)XL(1)*OLENGTH,XL(2)*OLENGTH,RMS*FSQTR,SUMO*
     1FSQTR,SUMM(1)*FSQTR,SUMM(2)*FSQTR,SUMF*FSQTR
      IGOR=IPREC3*2
      DO 2 J=1,IGOR
      XL(2)=XL(2)+DD
      CALL SPECTRUM
      CALL MSQVALUE
      RMS=DMAX1(SUMO,SUMM(1),SUMM(2),SUMF)
      CALL SCREEN
      IF(IANS.EQ.1)WRITE(4,3)XL(1)*OLENGTH,XL(2)*OLENGTH,RMS*FSQTR,SUMO*
     1FSQTR,SUMM(1)*FSQTR,SUMM(2)*FSQTR,SUMF*FSQTR
      IF(RMS.GE.RMSM) GO TO 2
      XLOPT(2)=XL(2)
      SUMOOPT=SUMO
      SUMFOPT=SUMF
      SUMMOPT(1)=SUMM(1)
      SUMMOPT(2)=SUMM(2)
      RMSM=RMS
    2 CONTINUE
```

```
      IF(IPREC3.GE.50)go to 998
      XL2=XL2-DD
      IGORA=IGOR+1
      DO 10 I=1,IGOR
      XL(1)=XL(1)+DD
      XL(2)=XL2
      DO 9 J=1,IGORA
      XL(2)=XL(2)+DD
      CALL SPECTRUM
      CALL MSQVALUE
      RMS=DMAX1(SUMO,SUMM(1),SUMM(2),SUMF)
      CALL SCREEN
      IF(IANS.EQ.1)WRITE(4,3)XL(1)*OLENGTH,XL(2)*OLENGTH,RMS*FSQTR,SUMO*
     1FSQTR,SUMM(1)*FSQTR,SUMM(2)*FSQTR,SUMF*FSQTR
      IF(RMS.GE.RMSM) GO TO 9
      XLOPT(1)=XL(1)
      XLOPT(2)=XL(2)
      SUMOOPT=SUMO
      SUMFOPT=SUMF
      SUMMOPT(1)=SUMM(1)
      SUMMOPT(2)=SUMM(2)
      RMSM=RMS
    9 CONTINUE
   10 CONTINUE
      GO TO 998
  300 IF (IPREC3.GE.20)THEN
  301           PRINT *,' ENTER THE NUMBER OF SPACER TO VARY'
                READ(*,*,ERR=301)I
                IF ((I.LT.1).OR.(I.GT.N))THEN
      PRINT *,' THE NUMBER MUST BE BETWEEN 1 AND ',N
                GO TO 301
                ENDIF
      XLI=XL(I)
      XL(I)=XLI-PREC2
      ENDIF
      DD=PREC2*1.5D0/DBLE(IPREC3)
      CALL SPECTRUM
      CALL MSQVALUE
      CALL SCREEN
      RMS=DMAX1(SUMO,SUMF)
      DO 6 IR=1,N
      RMS=DMAX1(RMS,SUMM(IR))
      XLOPT(IR)=XL(IR)
    6 SUMMOPT(IR)=SUMM(IR)
      SUMOOPT=SUMO
      SUMFOPT=SUMF
      RMSM=RMS
      IF (IPREC3.GE.20)GO TO 999
      IF(IANS.EQ.1)WRITE(4,3)XL(1)*OLENGTH,XL(2)*OLENGTH,RMS*FSQTR,SUMO*
     1FSQTR,SUMM(1)*FSQTR,SUMM(2)*FSQTR,SUMF*FSQTR
      DO 17 K=1,5
      DD=DD/1.5D0
      DO 16 I=1,N
```

```
      XLI=XLOPT(I)
      DDD=ZO
      DO 15 J=1,IPREC3
      DDD=DDD+DD
      XL(I)=XLI-DDD
      CALL SPECTRUM
      CALL MSQVALUE
      RMS=DMAX1(SUMO,SUMF)
      DO 12 L=1,N
   12 RMS=DMAX1(RMS,SUMM(L))
      CALL SCREEN
      IF (IANS.EQ.1)WRITE(4,3)(XL(IR)*OLENGTH,IR=1,N),RMS*FSQTR,SUMO*FSQ
     1TR,(SUMM(IR)*FSQTR,IR=1,N),SUMF*FSQTR
      IF(RMS.GE.RMSM) GO TO 14
      RMSM=RMS
      DO 8 L=1,N
    8 SUMMOPT(L)=SUMM(L)
      SUMOOPT=SUMO
      SUMFOPT=SUMF
      XLOPT(I)=XL(I)
   14 XL(I)=XLI+DDD
      CALL SPECTRUM
      CALL MSQVALUE
      RMS=DMAX1(SUMO,SUMF)
      DO 11 L=1,N
   11 RMS=DMAX1(RMS,SUMM(L))
      CALL SCREEN
      IF (IANS.EQ.1)WRITE(4,3)(XL(IR)*OLENGTH,IR=1,N),RMS*FSQTR,SUMO*FSQ
     1TR,(SUMM(IR)*FSQTR,IR=1,N),SUMF*FSQTR
      IF(RMS.GE.RMSM) GO TO 15
      RMSM=RMS
      DO 7 L=1,N
    7 SUMMOPT(L)=SUMM(L)
      SUMOOPT=SUMO
      SUMFOPT=SUMF
      XLOPT(I)=XL(I)
   15 CONTINUE
   16 XL(I)=XLOPT(I)
   17 CONTINUE
  998 SUMO=SUMOOPT
      SUMF=SUMFOPT
      DO 51 I=1,N
      XL(I)=XLOPT(I)
   51 SUMM(I)=SUMMOPT(I)
      RMS=DMAX1(SUMO,SUMF)
      DO 612 L=1,N
  612 RMS=DMAX1(RMS,SUMM(L))
      WRITE(*,52)
   52 FORMAT(////' THE OPTIMAL POSITIONS AND RELATED MSQ-FORCES WERE FOU
     1ND AS FOLLOWS:')
      CALL SCREEN
      RETURN
  999 IF (IANS.EQ.1)WRITE(4,3)XL(I)*OLENGTH,SUMO*FSQTR,(SUMM(IR)*FSQTR,
```

```
     1IR=1,N),SUMF*FSQTR,RMS*FSQTR
      DD=DD/1.5D0
      IRKA=IPREC3*2
      DO 800 J=1,IRKA
      XL(I)=XL(I)+DD
      CALL SPECTRUM
      CALL MSQVALUE
      RMS=DMAX1(SUM0,SUMF)
      DO 811 L=1,N
  811 RMS=DMAX1(RMS,SUMM(L))
      CALL SCREEN
      IF (IANS.EQ.1)WRITE(4,3)XL(I)*OLENGTH,SUM0*FSQTR,(SUMM(IR)*FSQTR,
     1IR=1,N),SUMF*FSQTR,RMS*FSQTR
      IF(RMS.GE.RMSM) GO TO 800
      RMSM=RMS
      SUMMOPT(I)=SUMM(I)
      SUMOOPT=SUM0
      SUMFOPT=SUMF
      XLOPT(I)=XL(I)
  800 CONTINUE
      XL(I)=XLI
      GO TO 998
      END
C
C
      SUBROUTINE MSQVALUE
      IMPLICIT DOUBLE PRECISION(A-H,O-X,Z)
      IMPLICIT COMPLEX(Y)
      DIMENSION SUMM(5),SPJ(5),SPN(5),SJ(5),SN(5)
      DIMENSION EIGVAL(500),EIGVEC(5,500)
      DIMENSION XL(5),D(5)
      DIMENSION V0(500),VF(500),VK(5,500)
      DIMENSION YDELK(500),YRP(500),YRM(500)
      DIMENSION ZJ(4),ZN(4),PJ(4),PN(4)
      COMMON /EIG/EIGVAL,EIGVEC
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/Z0,ONE,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /MSQ/SUM0,SUMF,SUMM
      COMMON /KINJ/YDELK,YRP,YRM
      COMMON /HALF/TWO,TWOH
      COMMON /COMPL/YI,YOMPP,YOMMP,YOMPM,YOMMM,YJ
      WRITE(*,70)(XL(I),I=1,N)
   70 FORMAT(/,' APPROACHING CALCULATION OF MSQ-STRAINS AT SPACERS'' CLA
     1MPS',/,' FOR SPACERS LOCATED AT FOLLOWING POSITIONS:',/,15(10X,F10
     2.6,/))
      Y0=CMPLX(Z0,Z0)
      NPJ=N+1
      NMJ=N-1
      SUM0=Z0
      SUMF=Z0
      DO 1 IR=1,N
    1 SUMM(IR)=Z0
```

```
      DO 62 K=1,M
      QK=EIGVAL(K)
      ROK=DAMP*SCALPR(K,K)/2.D0+DSTR
      YDELK(K)=CSQRT(CMPLX(QK*QK-ROK*ROK))
      YRP(K)=CMPLX(ZO,ROK)+YDELK(K)
      YRM(K)=CMPLX(ZO,ROK)-YDELK(K)
      IF(YRM(K).EQ.YO)YRM(K)=CMPLX(ZO,.5dO*QK*QK/ROK)
      IF(QK.GT.ZO)GO TO 60
      VO(K)=TWO
      VF(K)=TWO*DNINT(DCOS(QK))
      DO 59 IR=1,N
   59 VK(IR,K)=ZO
      GO TO 62
   60 VOK=ZO
      VFK=ZO
      DO 61 IR=1,N
      A=QK*XL(IR)
      B=EIGVEC(IR,K)
      VK(IR,K)=QK*DSIN(QK)*B
      VOK=VOK+DSIN(QK-A)*B
   61 VFK=VFK-DSIN(A)*B
      VO(K)=VOK*QK
      VF(K)=VFK*QK
   62 CONTINUE
C
C
C
      DO 888 J=1,M
      CALL CONSPAT(J,QJ,DELMQJ,DELPQJ,ZJO,ZJN,ZJ,PJO,PJN,PJ,PPJ,PMJ,ZPJ,
     1ZMJ,SJ,SPJ,NMJ,NPJ)
      DO 888 NR=1,J
      CALL CONSPAT(NR,QN,DELMQN,DELPQN,ZNO,ZNN,ZN,PNO,PNN,PN,PPN,PMN,ZPN
     1,ZMN,SN,SPN,NMJ,NPJ)
      UJN=ZO
      A=XL(N)
      B=ONE
      EX=EXP(ALFM*(B-A))
      ROBA=ZJN*ZNN*DIAG(EX,QJ,DELMQJ,DELPQJ,PJN,PPJ,PMJ,ZPJ,ZMJ,
     1QN,DELMQN,DELPQN,PNN,PPN,PMN,ZPN,ZMN,A,B)
      UJN=UJN+ROBA
      A=ZO
      B=XL(1)
      EX=EXP(ALFM*B)
      ROBB=ZJO*ZNO*DIAG(EX,QJ,DELMQJ,DELPQJ,PJO,PPJ,PMJ,ZPJ,ZMJ,
     1QN,DELMQN,DELPQN,PNO,PPN,PMN,ZPN,ZMN,A,B)
      UJN=UJN+ROBB
      IF(N.LT.2)GO TO 31
      DO 84 JR=1,NMJ
      A=B
      B=XL(JR+1)
      EX=EXP(ALFM*(B-A))
      UJN=UJN+ZJ(JR)*ZN(JR)*DIAG(EX,QJ,DELMQJ,DELPQJ,PJ(JR),PPJ,PMJ,ZPJ
     1,ZMJ,QN,DELMQN,DELPQN,PN(JR),PPN,PMN,ZPN,ZMN,A,B)
```

```
84 CONTINUE
31 A=XL(1)
   BRCJ=SPN(1)*SMICS(ZO,A,ZPN,PPN,ZMN,PMN,QN)
   BRSJ=SPN(1)*SMISS(ZO,A,ZPN,PPN,ZMN,PMN,QN)
   BRCN=SPJ(1)*SMICS(ZO,A,ZPJ,PPJ,ZMJ,PMJ,QJ)
   BRSN=SPJ(1)*SMISS(ZO,A,ZPJ,PPJ,ZMJ,PMJ,QJ)
   B=A
   IF(N.LT.2)GO TO 52
   B=XL(2)
   UJN=UJN+SJ(1)*(
  1BICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
  2BRSJ)+SN(1)*(BICSP(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISSP(A,B,ZPN,PPN
  3,ZMN,PMN,QN)*BRSN)
   UJN=UJN+SPJ(2)*(BICS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISS(A,B,ZPJ,PPJ
  1,ZMJ,PMJ,QJ)*BRSJ)+SPN(2)*(BICS(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISS(
  2A,B,ZPN,PPN,ZMN,PMN,QN)*BRSN)
   DO 50 IR=2,N
   IRR=IR-1
   A=XL(IRR)
   B=XL(IR)
   BRCJ=BRCJ+
  1SN(IRR)*SMICSP(A,B,ZPN,PPN,ZMN,PMN,QN)
  2+SPN(IR)*SMICS(A,B,ZPN,PPN,ZMN,PMN,QN)
   BRSJ=BRSJ+
  1SN(IRR)*SMISSP(A,B,ZPN,PPN,ZMN,PMN,QN)
  2+SPN(IR)*SMISS(A,B,ZPN,PPN,ZMN,PMN,QN)
   BRCN=BRCN+
  1SJ(IRR)*SMICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
  2+SPJ(IR)*SMICS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
   BRSN=BRSN+
  1SJ(IRR)*SMISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
  2+SPJ(IR)*SMISS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
   IF(IR.EQ.N)GO TO 50
   A=B
   B=XL(IR+1)
   UJN=UJN+SJ(IR)*(
  1BICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
  2BRSJ)+SPJ(IR+1)*(
  3BICS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
  4BRSJ)+SN(IR)*(
  5BICSP(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISSP(A,B,ZPN,PPN,ZMN,PMN,QN)*
  6BRSN)+SPN(IR+1)*(
  7BICS(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISS(A,B,ZPN,PPN,ZMN,PMN,QN)*
  8BRSN)
50 CONTINUE
52 A=B
   B=1.DO
   UJN=UJN+SJ(N)*(
  1BICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
  2BRSJ)+SN(N)*(
  3BICSP(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISSP(A,B,ZPN,PPN,ZMN,PMN,QN)*
  4BRSN)
   UJN=UJN/2.dO
```

```
      FUJ=FUJN(J,NR)
      ROB=FUJ*UJN
      IF(J.EQ.NR)ROB=ROB/2.d0
      SUMO=SUMO+VO(J)*VO(NR)*ROB
      SUMF=SUMF+VF(J)*VF(NR)*ROB
      DO 65 IR=1,N
   65 SUMM(IR)=SUMM(IR)+VK(IR,J)*VK(IR,NR)*ROB
  888 CONTINUE
      RETURN
      END
C
C
      COMPLEX FUNCTION YSOM(Y)
      IMPLICIT DOUBLE PRECISION(A-H,O-X,Z)
      IMPLICIT COMPLEX(Y)
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /COMPL/YI,YOMPP,YOMMP,YOMPM,YOMMM,YJ
      YSOM=DALF*(Y*Y+A2N2)/(Y-YOMPP)/(Y-YOMMP)/(Y-YOMPM)/(Y-YOMMM)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION FUJN(I,J)
      IMPLICIT DOUBLE PRECISION(A-H,O-X,Z)
      IMPLICIT COMPLEX(Y)
      DIMENSION YDELK(500),YRP(500),YRM(500)
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /KINJ/YDELK,YRP,YRM
      COMMON /COMPL/YI,YOMPP,YOMMP,YOMPM,YOMMM,YJ
      Y1=CONJG(YRP(I))
      Y2=CONJG(YRM(I))
      Y3=YRP(J)
      Y4=YRM(J)
      FUJN=DBLE(YJ/(YOMPP-Y1)/(YOMPP-Y2)/(YOMPP-Y3)/(YOMPP-Y4)+
     1YJ/(YOMMP-Y1)/(YOMMP-Y2)/(YOMMP-Y3)/(YOMMP-Y4)+YI/YDELK(J)*(
     2YSOM(Y3)/(Y3-Y1)/(Y3-Y2)-YSOM(Y4)/(Y4-Y1)/(Y4-Y2)))
      RETURN
      END
C
C
      SUBROUTINE CONSPAT(J,QJ,DELMQJ,DELPQJ,ZJO,ZJN,ZJ,PJO,PJN,PJ,PPJ,PM
     1J,ZPJ,ZMJ,SJ,SPJ,NMJ,NPJ)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION SPJ(5),SJ(5),ZJ(4),PJ(4)
      DIMENSION EIGVAL(500),EIGVEC(5,500),XL(5),D(5)
      COMMON /EIG/EIGVAL,EIGVEC
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ONE,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /HALF/TWO,TWOH
      QJ=EIGVAL(J)
      JFLAG=1
      IF(QJ.LE.ZO)THEN
                  JFLAG=0
```

```
                    QJ=-QJ
                    ENDIF
      SINJ=DSIN(QJ)
      COSJ=DCOS(QJ)
      DELMQJ=DEL-QJ
      DELPQJ=DEL+QJ
      ZMJ=DSQRT(ALFSQ+DELMQJ*DELMQJ)
      ZPJ=DSQRT(ALFSQ+DELPQJ*DELPQJ)
      PMJ=DATAN2(ALF,DELMQJ)
      PPJ=DATAN2(ALF,DELPQJ)
      IF(JFLAG.EQ.1)GO TO 73
      TWCOSJ=-DNINT(COSJ)*TWOH
      IF(N.EQ.1)GO TO 72
      DO 75 JR=1,NMJ
      JRR=JR+1
      ZJ(JR)=TWO
      PJ(JR)=ZO
      SJ(JR)=TWCOSJ
   75 SPJ(JRR)=TWOH
   72 SJ(N)=TWCOSJ*2.d0
      SPJ(1)=TWO
      ZJO=TWO
      ZJN=TWO
      PJO=ZO
      PJN=ZO
      RETURN
   73 ROB1=ZO
      ROB2=ZO
      DO 10 JR=1,N
      ROB1=ROB1+DSIN(QJ*XL(JR))*EIGVEC(JR,J)
      JRR=NPJ-JR
      ROB2=ROB2+DSIN(QJ-QJ*XL(JRR))*EIGVEC(JRR,J)
      SJ(JR)=ROB1
   10 SPJ(JRR)=ROB2
   74 ZJO=SPJ(1)
      ZJN=-SJ(N)
      PJO=ZO
      PJN=-QJ
      IF(N.LT.2) RETURN
      DO 80 JR=1,NMJ
      SJIMJ=SJ(JR)
   80 CALL XI(SJIMJ*SINJ,SPJ(JR+1)-SJIMJ*COSJ,PJ(JR),ZJ(JR))
      RETURN
      END
C
C
      SUBROUTINE XI(A,B,C,D)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      D=DSQRT(A*A+B*B)
      IF(D.EQ.ZO)THEN
                C=ZO
                        D=ZJ
```

```
                RETURN
                ENDIF
      C=DATAN2(A,B)
      RETURN
      END
C
C
      DOUBLE PRECISION FUNCTION SCALPR(I,K)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION EIGVAL(500),EIGVEC(5,500)
      COMMON /EIG/EIGVAL,EIGVEC
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      ROB=ZO
      DO 1 J=1,N
    1 ROB=ROB+EIGVEC(J,I)*EIGVEC(J,K)
      SCALPR=ROB
      RETURN
      END
C
C
      DOUBLE PRECISION FUNCTION CO(A,B,X,Y)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      IF(X.EQ.0.)THEN
                  CO=(A-B)*DSIN(Y)
                  RETURN
                ELSE
      CO=(DCOS(B*X+Y)-DCOS(A*X+Y))/X
      RETURN
      ENDIF
      END
C
C
      DOUBLE PRECISION FUNCTION BICSP(X1,X2,ZP,PP,ZM,PM,Q)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /PAR/DAMP,QU,N,M,ALF,DEL,DNU,DLAM,DSTR
      BICSP=
     1(DCOS(X2*(DEL+Q)-PP-Q)/ZP-DCOS(X2*(DEL-Q)-PM+Q)/ZM)/EXP(ALF*X2)-
     2(DCOS(X1*(DEL+Q)-PP-Q)/ZP-DCOS(X1*(DEL-Q)-PM+Q)/ZM)/EXP(ALF*X1)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION BISSP(X1,X2,ZP,PP,ZM,PM,Q)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /PAR/DAMP,QU,N,M,ALF,DEL,DNU,DLAM,DSTR
      BISSP=
     1(DSIN(X2*(DEL+Q)-PP-Q)/ZP-DSIN(X2*(DEL-Q)-PM+Q)/ZM)/EXP(ALF*X2)-
     2(DSIN(X1*(DEL+Q)-PP-Q)/ZP-DSIN(X1*(DEL-Q)-PM+Q)/ZM)/EXP(ALF*X1)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION BICS(X1,X2,ZP,PP,ZM,PM,Q)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
```

```
      COMMON /PAR/DAMP,QU,N,M,ALF,DEL,DNU,DLAM,DSTR
      BICS=
     1(DCOS(X2*(DEL-Q)-PM)/ZM-DCOS(X2*(DEL+Q)-PP)/ZP)/EXP(ALF*X2)-
     2(DCOS(X1*(DEL-Q)-PM)/ZM-DCOS(X1*(DEL+Q)-PP)/ZP)/EXP(ALF*X1)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION BISS(X1,X2,ZP,PP,ZM,PM,Q)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /PAR/DAMP,QU,N,M,ALF,DEL,DNU,DLAM,DSTR
      BISS=
     1(DSIN(X2*(DEL-Q)-PM)/ZM-DSIN(X2*(DEL+Q)-PP)/ZP)/EXP(ALF*X2)-
     2(DSIN(X1*(DEL-Q)-PM)/ZM-DSIN(X1*(DEL+Q)-PP)/ZP)/EXP(ALF*X1)
      RETURN
      END
C
C
      DOUBLE PRECISION FUNCTION SMICSP(X1,X2,ZP,PP,ZM,PM,Q)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /PAR/DAMP,QU,N,M,ALF,DEL,DNU,DLAM,DSTR
      SMICSP=
     1EXP(ALF*X2)*(DCOS(X2*(DEL+Q)+PP-Q)/ZP-DCOS(X2*(DEL-Q)+PM+Q)/ZM)-
     2EXP(ALF*X1)*(DCOS(X1*(DEL+Q)+PP-Q)/ZP-DCOS(X1*(DEL-Q)+PM+Q)/ZM)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION SMISSP(X1,X2,ZP,PP,ZM,PM,Q)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /PAR/DAMP,QU,N,M,ALF,DEL,DNU,DLAM,DSTR
      SMISSP=
     1EXP(ALF*X2)*(DSIN(X2*(DEL+Q)+PP-Q)/ZP-DSIN(X2*(DEL-Q)+PM+Q)/ZM)-
     2EXP(ALF*X1)*(DSIN(X1*(DEL+Q)+PP-Q)/ZP-DSIN(X1*(DEL-Q)+PM+Q)/ZM)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION SMICS(X1,X2,ZP,PP,ZM,PM,Q)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /PAR/DAMP,QU,N,M,ALF,DEL,DNU,DLAM,DSTR
      SMICS=
     1EXP(ALF*X2)*(DCOS(X2*(DEL-Q)+PM)/ZM-DCOS(X2*(DEL+Q)+PP)/ZP)-
     2EXP(ALF*X1)*(DCOS(X1*(DEL-Q)+PM)/ZM-DCOS(X1*(DEL+Q)+PP)/ZP)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION SMISS(X1,X2,ZP,PP,ZM,PM,Q)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /PAR/DAMP,QU,N,M,ALF,DEL,DNU,DLAM,DSTR
      SMISS=
     1EXP(ALF*X2)*(DSIN(X2*(DEL-Q)+PM)/ZM-DSIN(X2*(DEL+Q)+PP)/ZP)-
     2EXP(ALF*X1)*(DSIN(X1*(DEL-Q)+PM)/ZM-DSIN(X1*(DEL+Q)+PP)/ZP)
      RETURN
      END
C
```

```
C
      DOUBLE PRECISION FUNCTION DIAG(EX,QJ,DELMQJ,DELPQJ,PJ,PPJ,PMJ,ZPJ
     1,ZMJ,QN,DELMQN,DELPQN,PN,PPN,PMN,ZPN,ZMN,A,B)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      ROBA=
     1(CO(A,B,QJ-QN,PJ-PMJ-PN)-CO(A,B,DELPQN-DELMQJ,PJ-PMJ+PN))/ZMJ-
     2(CO(A,B,DELMQN-DELPQJ,-PPJ-PJ-PN)-CO(A,B,QN-QJ,PN-PPJ-PJ))/ZPJ
      ROBB=DCOS(A*DELPQJ+PJ+PPJ)/ZPJ-DCOS(A*DELMQJ-PJ+PMJ)/ZMJ
      ROBC=(EX*DCOS(B*DELPQN+PN-PPN)-DCOS(A*DELPQN+PN-PPN))/ZPN
      ROBD=(EX*DCOS(B*DELMQN-PN-PMN)-DCOS(A*DELMQN-PN-PMN))/ZMN
      ROBE=DSIN(DELMQJ*A-PJ+PMJ)/ZMJ-DSIN(DELPQJ*A+PJ+PPJ)/ZPJ
      ROBF=(EX*DSIN(DELMQN*B-PN-PMN)-DSIN(DELMQN*A-PN-PMN))/ZMN
      ROBG=(EX*DSIN(DELPQN*B+PN-PPN)-DSIN(DELPQN*A+PN-PPN))/ZPN
      DIA=ROBA-ROBB*(ROBC-ROBD)-ROBE*(ROBF-ROBG)
      ROBA=
     1(CO(A,B,QN-QJ,PN-PMN-PJ)-CO(A,B,DELPQJ-DELMQN,PN-PMN+PJ))/ZMN-
     2(CO(A,B,DELMQJ-DELPQN,-PPN-PN-PJ)-CO(A,B,QJ-QN,PJ-PPN-PN))/ZPN
      ROBB=DCOS(A*DELPQN+PN+PPN)/ZPN-DCOS(A*DELMQN-PN+PMN)/ZMN
      ROBC=(EX*DCOS(B*DELPQJ+PJ-PPJ)-DCOS(A*DELPQJ+PJ-PPJ))/ZPJ
      ROBD=(EX*DCOS(B*DELMQJ-PJ-PMJ)-DCOS(A*DELMQJ-PJ-PMJ))/ZMJ
      ROBE=DSIN(DELMQN*A-PN+PMN)/ZMN-DSIN(DELPQN*A+PN+PPN)/ZPN
      ROBF=(EX*DSIN(DELMQJ*B-PJ-PMJ)-DSIN(DELMQJ*A-PJ-PMJ))/ZMJ
      ROBG=(EX*DSIN(DELPQJ*B+PJ-PPJ)-DSIN(DELPQJ*A+PJ-PPJ))/ZPJ
      DIAG=DIA+ROBA-ROBB*(ROBC-ROBD)-ROBE*(ROBF-ROBG)
      RETURN
      END
C
C
      SUBROUTINE PRINSPEC
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION EIGVAL(500),EIGVEC(5,500)
      COMMON /EIG/EIGVAL,EIGVEC
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
   19 WRITE(*,20)
   20 FORMAT(/' DO YOU WANT TO WRITE THE SPECTRUM TO A FILE?',/,
     1' HIT 1 FOR yes OR 0 FOR no    ')
      READ(*,*,ERR=21)IANS
      IF((IANS.NE.0).AND.(IANS.NE.1))GO TO 21
      IF (IANS.EQ.0)GO TO 24
      GO TO 23
   21 WRITE(*,22)
   22 FORMAT(/' WRONG ANSWER, TRY AGAIN')
      GO TO 19
   23 WRITE(2,25)(I,EIGVAL(I),I=1,M)
      DO 27 I=1,M
   27 WRITE(3,26)I,(EIGVEC(J,I),J=1,N)
   24 CONTINUE
      RETURN
   25 FORMAT(' EIGENVALUE No ',I3,5X,E15.6)
   26 FORMAT(/,' EIGENFORM No. ',I3,/,4(4(E15.7,5X),/))
      END
C
```

```
C
      SUBROUTINE PRINTMSQ
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION SUMM(5)
      DIMENSION XL(5),D(5)
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /MSQ/SUMO,SUMF,SUMM
      COMMON /TRANS/FORTR,FSQTR,DTRANS
      COMMON /PHYS/OTENSION,OLENGTH,OSTIF,ODAMP,ODSTR,OSPEGRAV,OINTENS,O
     1ALF,OLAM,ODEL,ONU
      WRITE(*,1)
    1 FORMAT(////' THE FOLLOWING MSQ-VALUES WERE OBTAINED:',/
     1/,1H ,50(1H-),/,'POSITION [m]  /(dimless)  I        MSQ-FORCE',/,
     2' ',50(1H-))
      WRITE(*,2)ZO,ZO,SUMO*FSQTR
    2 FORMAT(1X,F6.2,'     (',F6.4,1h),6X,1HI,5X,E15.7)
      DO 3 I=1,N
    3 WRITE(*,2)XL(I)*OLENGTH,XL(I),SUMM(I)*FSQTR
      WRITE(*,2)OLENGTH,ZJ,SUMF*FSQTR
   14 WRITE(*,15)
   15 FORMAT(1X,50(1H-),
     1/' DO YOU WANT TO WRITE THE MSQ-VALUES TO A FILE?',/,
     2' HIT 1 FOR yes OR 0 FOR no    ')
      READ(*,*,ERR=17)IANS
      IF ((IANS.NE.0).AND.(IANS.NE.1))GO TO 14
      IF (IANS.EQ.0) RETURN
      WRITE(1,*)ZO,ZO,SUMO*FSQTR
      DO 16 I=1,N
   16 WRITE(1,*)XL(I)*OLENGTH,XL(I),SUMM(I)*FSQTR
      WRITE(1,*)ZJ*OLENGTH,ZJ,SUMF*FSQTR
      RETURN
   17 WRITE(*,18)
   18 FORMAT(/' MISTAKEN ANSWER! TRY AGAIN')
      GO TO 14
      END
C
C
      SUBROUTINE DATA
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      CHARACTER*40 BL
      DIMENSION XL(5),D(5)
      COMMON /PHYS/OTENSION,OLENGTH,OSTIF,ODAMP,ODSTR,OSPEGRAV,OINTENS,O
     1ALF,OLAM,ODEL,ONU
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /PREC/PREC1,PREC2,IPREC3,PO1,P99
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /TRANS/FORTR,FSQTR,DTRANS
      BL='                                        '
      NOLD=N
  113 WRITE(*,100)N,M,OTENSION,ZJ,OLENGTH,ZJ,OSTIF,OSTIF*OLENGTH/OTENSIO
```

```
     1N,ODAMP,DAMP,ODSTR,DSTR,OSPEGRAV,ZJ,OINTENS,FSQTR/OINTENS/OTENSION
     2,OALF,ZJ/ALF,OLAM,ZJ/DLAM,ODEL,ZJ/DEL,ONU,ZJ/DNU,PREC1,PREC2,IPREC
     33
      READ(*,*,err=113)ians
      IF((IANS.LT.1).OR.(IANS.GT.19))THEN
                            WRITE(*,*)' WRONG ANSWER!!!!!!!!'
                            GO TO 113
                            ENDIF
     go to(11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29),ia
     1ns
   11 CALL MOD(N,'NO-OF-SPACERS                              '//BL)
      IF((N.LT.1).OR.(N.GT.5)) GO TO 111
      IF(NOLD.ne.N) THEN
                    WRITE(*,101)
                    NOLD=N
                    W=ZJ/DBLE(N+1)
                    DO 114 K=1,N
  114               XL(K)=DBLE(K)*W
                    GO TO 27
                    ENDIF
      GO TO 113
  101 FORMAT(/////////////,' THE NUMBER OF SPACERS HAS BEEN CHANGED;',//,
     1' thus, a placement for the new number of spacers had to be define
     2d.',//,' I placed the spacers uniformly (equididistantly);',//,
     3' now, YOU can choose (another) positions if you wish.',//)
  111 WRITE(*,112)
  112 FORMAT(' No-of-Spacers must be an integer 0<i<6')
      N=NOLD
      GO TO 11
   12 CALL MOD(M,'NUMBER-OF-MODES                            '//BL)
      GO TO 113
   13 CALL MODR(OTENSION,'TENSILE FORCE [N]'//BL)
      CALL DIMLESS
      GO TO 113
   14 CALL MODR(OLENGTH,'CONDUCTOR LENGTH [m]'//BL)
      CALL DIMLESS
      GO TO 113
   15 CALL MODR(OSTIF,'SPACER STIFNESS [N/m] '//BL)
      CALL DIMLESS
      GO TO 113
   16 CALL MODR(ODAMP,'SPACER DAMPING CONSTANT [Ns/m]'//BL)
      CALL DIMLESS
      GO TO 113
   17 CALL MODR(ODSTR,'STRUCTURAL DAMPING CONSTANT [Ns/m]'//BL)
      CALL DIMLESS
      GO TO 113
   18 CALL MODR(OSPEGRAV,'CONDUCTOR SPECIFIC GRAVITY [N/m]'//BL)
      CALL DIMLESS
      GO TO 113
   19 CALL MODR(OINTENS,'LOADING INTENSITY [N/m]'//BL)
      CALL DIMLESS
      GO TO 113
   20 CALL MODR(OALF,'LOADING CORRELATION LENGTH [m]'//BL)
```

```
      CALL DIMLESS
      GO TO 113
   21 CALL MODR(OLAM,'LOADING CORRELATION TIME [s]'//BL)
      CALL DIMLESS
      GO TO 113
   22 CALL MODR(ODEL,'LOADING MSQ-WAVE-LENGTH [m]'//BL)
      CALL DIMLESS
      GO TO 113
   23 CALL MODR(ONU,'LOADING MSQ-WAVE-PERIOD [s]'//BL)
      CALL DIMLESS
      GO TO 113
   24 CALL MODR(PREC1,'MIN-OF-CHAR.-DETERMINANT            '//BL)
      GO TO 113
   25 CALL MODR(PREC2,'MAX-DEPARTURE-FROM-UNIFORM-PLACEMENT'//BL)
      GO TO 113
   26 CALL MOD(IPREC3,'NUMBER OF SAMPLING POINTS PER SPACER'//BL)
      GO TO 113
   27 CALL READPOS
      GO TO 113
   28 WRITE(7,115)N,M,OTENSION,ZJ,OLENGTH,ZJ,OSTIF,OSTIF*OLENGTH/OTENSIO
     1N,ODAMP,DAMP,ODSTR,DSTR,OSPEGRAV,ZJ,OINTENS,FSQTR/OINTENS/OTENSION
     2,OALF,ZJ/ALF,OLAM,ZJ/DLAM,ODEL,ZJ/DEL,ONU,ZJ/DNU,PREC1,PREC2,IPREC
     33
   29 RETURN
  100 FORMAT(////' CHOOSE THE ITEM YOU WANT TO CHANGE (displayed are the
     1 actual values;',/' the values in braces are the dimensionless equ
     2ivalents)'//
     2' 1: NO-OF-SPACERS                          ',i2,/,
     3' 2: NUMBER-OF-MODES                       ',i3,/,
     4' 3: TENSILE FORCE [N]                     ',d9.3,2h (,d12.6,1h),/,
     5' 4: CONDUCTOR LENGTH [m]                  ',d9.3,2h (,d12.6,1h),/,
     6' 5: SPACER STIFNESS [N/m]                 ',d9.3,2h (,d12.6,1h),/,
     7' 6: SPACER DAMPING CONSTANT [Ns/m]        ',d9.3,2h (,d12.6,1h),/,
     8' 7: STRUCTURAL DAMPING CONSTANT [Ns/m]    ',d9.3,2h (,d12.6,1h),/,
     9' 8: CONDUCTOR SPECIFIC GRAVITY [n/m]      ',d9.3,2h (,d12.6,1h),/,
     a' 9: LOADING INTENSITY [N/m]               ',d9.3,2h (,d12.6,1h),/,
     F' 10:LOADING CORRELATION LENGTH [m]        ',d9.3,2h (,d12.6,1h),/,
     G' 11:LOADING CORRELATION TIME [s]          ',d9.3,2h (,d12.6,1h),/,
     H' 12:LOADING MSQ-WAVE-LENGTH [m]           ',d9.3,2h (,d12.6,1h),/,
     I' 13:LOADING MSQ-WAVE-PERIOD [s]           ',d9.3,2h (,d12.6,1h),/,
     b' 14:MIN-OF-CHAR.-DETERMINANT              ',d12.6,/,
     c' 15:MAX-DEPARTURE-FROM-UNIFORM-PLACEMENT  ',d12.6,/,
     d' 16:NUMBER OF SAMPLING POINTS PER SPACER    ',I3,/
     e' 17:SPACERS'' COORDINATES',/,' 18:WRITE PARAMETERS TO A FILE',/,
     k' 19:BACK TO MAIN MENU',/,'
     J                your choice:   ',$)
  115 FORMAT(' THE PARAMETERS USED IN THE LAST RUN    (displayed are the
     1 actual values;',/' the values in braces are the dimensionless equ
     2ivalents)'//
     2' 1: NO-OF-SPACERS                          ',i2,/,
     3' 2: NUMBER-OF-MODES                       ',i3,/,
     4' 3: TENSILE FORCE [N]                     ',d9.3,2h (,d12.6,1h),/,
     5' 4: CONDUCTOR LENGTH [m]                  ',d9.3,2h (,d12.6,1h),/,
```

```
     6' 5: SPACER STIFNESS [N/m]                ',d9.3,2h (,d12.6,1h),/,
     7' 6: SPACER DAMPING CONSTANT [Ns/m]       ',d9.3,2h (,d12.6,1h),/,
     8' 7: STRUCTURAL DAMPING CONSTANT [Ns/m]   ',d9.3,2h (,d12.6,1h),/,
     9' 8: CONDUCTOR SPECIFIC GRAVITY [N/m]     ',d9.3,2h (,d12.6,1h),/,
     a' 9: LOADING INTENSITY [N/m]              ',d9.3,2h (,d12.6,1h),/,
     F' 10:LOADING CORRELATION LENGTH [m]       ',d9.3,2h (,d12.6,1h),/,
     G' 11:LOADING CORRELATION TIME [s]         ',d9.3,2h (,d12.6,1h),/,
     H' 12:LOADING MSQ-WAVE-LENGTH [m]          ',d9.3,2h (,d12.6,1h),/,
     I' 13:LOADING MSQ-WAVE-PERIOD [s]          ',d9.3,2h (,d12.6,1h),/,
     b' 14:MIN-OF-CHAR.-DETERMINANT             ',d12.6,/,
     c' 15:MAX-DEPARTURE-FROM-UNIFORM-PLACEMENT ',d12.6,/,
     d' 16:NUMBER OF SAMPLING POINTS PER SPACER   ',I3)
      END
C
C
      SUBROUTINE READPOS
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION XL(5),D(5)
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      WRITE(*,18)
      DO 1 I=1,N
    1 WRITE(*,8)I,XL(I)
      WRITE(*,7)
      READ(*,*,ERR=113)IANS
      IF (IANS.LE.0)GO TO 4
      RETURN
    4 WRITE(*,10)N,N
      XMIN=ZO
      DO 15 I=1,N
    5 READ(*,*,ERR=13)XL(I)
      IF ((XL(I).LE.XMIN).OR.(XL(I).GE.1.DO))GO TO 14
      GO TO 15
   13 WRITE(*,12)
      GO TO 5
   14 WRITE(*,11)I,XMIN
      GO TO 5
   15 XMIN=XL(I)
   16 WRITE(*,9)
      DO 17 I=1,N
   17 WRITE(*,8)I,XL(I)
      WRITE(*,7)
      READ(*,*,ERR=113)IANS
      IF (IANS.LE.0)GO TO 4
      RETURN
  113 WRITE(*,19)
      GO TO 16
    7 FORMAT(/,' IF THE VALUES ARE O.K. TYPE-IN ANY POSITIVE INTEGER',/,
     1' IF THE VALUES ARE WRONG TYPE-IN 0 OR ANY NEGATIVE INTEGER    ')
    8 FORMAT(' SPACER  ',I2,' AT  ',F12.10)
    9 FORMAT(/,' THE POSITIONS CHOSEN ARE AS FOLLOWS:'/)
   10 FORMAT(/,' INTRODUCE THE POSITIONS FOR ',I2,' SPACERS',/,
     1' THAT IS, TYPE-IN THE REAL NUMBERS Xi S.T. 0<X1<X2<...<X',I2,'<1.
```

```
     2O',/,' WHERE Xi IS THE RATIO OF THE i-TH SPACER''S DISTANCE FROM T
     3HE SELECTED',/,' CONDUCTOR''S END AND THE CONDUCTOR''S LENGTH',/)
  11 FORMAT(/,' WRONG VALUE!',/,' X',I2,' MUST BE A REAL NUMBER GREATER
     1 THEN',F8.5,/,' AND LESS THEN 1.0',/,' TYPE-IN A PROPER VALUE   ')
  12 FORMAT(/,' MISTAKE! I WAIT FOR REAL INPUT DATA. TRY AGAIN')
  19 FORMAT(/,' MISTAKE! I WAIT FOR AN INTEGER')
  18 FORMAT(/,' THE ACTUAL POSITIONS OF SPACERS ARE AS FOLLOWS:'/)
     END
C
     SUBROUTINE DIMLESS
     IMPLICIT DOUBLE PRECISION(A-H,O-X,Z)
     IMPLICIT COMPLEX(Y)
     COMMON /PHYS/OTENSION,OLENGTH,OSTIF,ODAMP,ODSTR,OSPEGRAV,OINTENS,O
    1ALF,OLAM,ODEL,ONU
     COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
     COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
     COMMON /TRANS/T,FSQTR,D
     COMMON /COMPL/YI,YOMPP,YOMMP,YOMPM,YOMMM,YJ
     TKAP=OTENSION/OSTIF/OLENGTH
     T=OLENGTH*DSQRT(OSPEGRAV/OTENSION)
     FSQTR=OLENGTH*OLENGTH*OINTENS*OINTENS
     D=DSQRT(OSPEGRAV*OTENSION)
     DAMP=ODAMP/D
     DSTR=ODSTR/D
     ALF=OLENGTH/OALF
     DEL=OLENGTH/ODEL
     DLAM=T/OLAM
     DNU=T/ONU
     ALFM=-ALF
     XLAMM=-DLAM
     ALFSQ=ALF*ALF
     DALF=ALF*2.DO
     A2N2=ALF*ALF+DNU*DNU
     YOMPP=CMPLX(DNU,ALF)
     YOMMP=CMPLX(-DNU,ALF)
     YOMPM=CMPLX(DNU,-ALF)
     YOMMM=CMPLX(-DNU,-ALF)
     RETURN
     END
C
C
     SUBROUTINE SCREEN
     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
     DIMENSION SUMM(5)
     DIMENSION XL(5),D(5)
     COMMON /L/XL,D
     COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
     COMMON /MSQ/SUMO,SUMF,SUMM
     COMMON /OPT/RMS
     COMMON /TRANS/FORTR,FSQTR,DTRANS
     ZO=0.
     ZJ=1.
     WRITE(*,1)
```

```
    1 FORMAT(////' THE FOLLOWING MSQ-VALUES WERE OBTAINED:',/
     1/,1H ,50(1H-),/,'        POSITION          I        MSQ-FORCE',/,
     2' ',50(1H-))
      WRITE(*,2)ZO,SUMO*FSQTR
    2 FORMAT(5X,F12.10,9X,1HI,5X,E15.7)
      DO 3 I=1,N
    3 WRITE(*,2)XL(I),SUMM(I)*FSQTR
      WRITE(*,2)ZJ,SUMF*FSQTR
      WRITE(*,4)RMS*FSQTR
    4 FORMAT(1X,50(1H-),
     1/,' THE MAXIMAL MSQ-FORCE UNDER THIS PLACEMENT:  ',E15.7)
C     PAUSE
      RETURN
      END
C
C     MODIFIES THE VALUE OF AN INTEGER PARAMETER
C
C
      SUBROUTINE MOD(I,T)
      CHARACTER*40 ANSW,T,B
      CHARACTER*1 Y,N,A,YS,NS
      Y='Y'
      YS='y'
      N='N'
      NS='n'
      B='
      T=T//B
      WRITE(*,1)T,I,B
   10 READ (*,4)ANSW
      A=ANSW
      IF((A.EQ.N).OR.(A.EQ.NS))RETURN
   13 CONTINUE
      IF((A.EQ.Y).OR.(A.EQ.YS))THEN
                    WRITE(*,2)
                    READ (*,*,ERR=11)I
                    RETURN
                ELSE
                    WRITE(*,3)
                    GO TO 10
                ENDIF
   11 WRITE(*,'('' INVALID DATA; PARAMETER '',A40,/,'' IS INTEGER-TYPE;
     1 TRY AGAIN'')')T
      GO TO 13
     1 FORMAT(' ACTUAL VALUE OF PARAMETER "',A40,'"',/,' IS: ',I11,A8/
     1' DO YOU WISH TO CHANGE IT? (Y/N)   ')
     2 FORMAT(' NEW  VALUE  OF  PARAMETER: ',/)
     3 FORMAT(' INVALID ANSWER (ENTER Y=YES OR N=NO)  ')
     4 FORMAT(A40)
      END
C
C     MODIFIES THE VALUE OF A DOUBLE PRECISION PARAMETER
C
C
```

```
      SUBROUTINE MODR(Z,T)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      CHARACTER*40 ANSW,T,B
      CHARACTER*1 Y,N,A,YS,NS
      Y='Y'
      YS='y'
      N='N'
      NS='n'
      B='
      T=T//B
      WRITE(*,1)T,Z,B
   10 READ (*,4)ANSW
      A=ANSW
      IF((A.EQ.N).OR.(A.EQ.NS))RETURN
   13 CONTINUE
      IF((A.EQ.Y).OR.(A.EQ.YS))THEN
                     WRITE(*,2)
                     READ (*,*,ERR=11)Z
                     RETURN
                ELSE
                     WRITE(*,3)
                     GO TO 10
                ENDIF
   11 WRITE(*,'('' INVALID DATA;'',/,'' PARAMETER '',A40,'' IS DP - TYPE
     1;'',/,'' TRY AGAIN'')')T
      GO TO 13
    1 FORMAT(' ACTUAL VALUE OF PARAMETER "',A40,'",/,' IS: ',D20.12,A8/
     1' DO YOU WISH TO CHANGE IT? (Y/N)   ')
    2 FORMAT(' NEW  VALUE  OF  PARAMETER: ',/)
    3 FORMAT(' INVALID ANSWER (ENTER Y=YES OR N=NO)  ')
    4 FORMAT(A40)
      END
C
      BLOCK DATA
      IMPLICIT DOUBLE PRECISION(A-H,O-X,Z)
      IMPLICIT COMPLEX(Y)
      DIMENSION XL(5),D(5)
      COMMON /L/XL,D
      COMMON /PHYS/OTENSION,OLENGTH,OSTIF,ODAMP,ODSTR,OSPEGRAV,OINTENS,O
     1ALF,OLAM,ODEL,ONU
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /PREC/PREC1,PREC2,IPREC3,P01,P99
      COMMON /HALF/TWO,TWOH
      COMMON /COMPL/YI,YOMPP,YOMMP,YOMPM,YOMMM,YJ
      DATA TWO,TWOH/1.4142135623730951d0,.70710678118654757d0/
      DATA ZO,ZJ,PI,N,M/0.D0,1.D0,3.14159265358979323850,1,1/
      DATA OTENSION,OLENGTH,OSTIF,ODAMP,ODSTR,OSPEGRAV,OINTENS,OALF,OLAM
     1,ODEL,ONU/3.d4,3.d2,5.d4,1.d3,1.d-2,2.d1,1.d0,3.d1,.3d0,3.d0,.2d0/
      DATA XL(1)/.5D0/
      DATA YI,YJ/(0.,1.),(1.,0.)/
      DATA PREC1,PREC2,IPREC3,P01,P99/.1D-19,.4D0,50,.01D0,.99D0/
      END
```

Appendix 2
Listing of some segments of the FORTRAN code SPA2.FOR

The SPA2.FOR code implements the 2-nd-order-spectral-method. The overall structure of the program is that of the code SPA0.FOR. Below, only those of the segments of the SPA2.FOR program are listed, which are different from the respective segments of the SPA0.FOR program.

```
C
      SUBROUTINE MSQVALUE
      IMPLICIT DOUBLE PRECISION(A-H,O-X,Z)
      IMPLICIT COMPLEX(y)
      DIMENSION SUMM(5),SPJ(5),SPN(5),SJ(5),SN(5)
      DIMENSION EIGVAL(50),EIGVEC(5,50)
      DIMENSION XL(5),D(5)
      DIMENSION SMI(5)
      DIMENSION VO(50),VF(50),VK(5,50)
      DIMENSION YDELK(50),YRP(50),YRM(50)
      DIMENSION ZJ(4),ZN(4),PJ(4),PN(4)
      DIMENSION RIJ(50,50)
      COMMON /EIG/EIGVAL,EIGVEC
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ONE,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /MSQ/SUMO,SUMF,SUMM
      COMMON /KINJ/YDELK,YRP,YRM
      COMMON /HALF/TWO,TWOH
      COMMON /COMPL/YI,YOMPP,YOMMP,YOMPM,YOMMM,YJ
      WRITE(*,70)(XL(I),I=1,N)
   70 FORMAT(/,' APPROACHING CALCULATION OF MSQ-STRAINS AT SPACERS'' CLA
     1MPS',/,' FOR SPACERS LOCATED AT FOLLOWING POSITIONS:',/,15(10X,F10
     2.6,/))
      NPJ=N+1
      NMJ=N-1
      SUMO=ZO
      SUMF=ZO
      DO 1 IR=1,N
    1 SUMM(IR)=ZO
      DO 62 K=1,M
      QK=EIGVAL(K)
      ROK=DAMP*SCALPR(K,K)/2.DO+DSTR
      YDELK(K)=CSQRT(CMPLX(QK*QK-ROK*ROK))
      YRP(K)=CMPLX(ZO,ROK)+YDELK(K)
      YRM(K)=CMPLX(ZO,ROK)-YDELK(K)
      IF(YRM(K).EQ.YO)YRM(K)=CMPLX(ZO,.5DO*QK*QK/ROK)
      IF(QK.GT.ZO)GO TO 60
      VO(K)=TWO
      VF(K)=TWO*DNINT(DCOS(QK))
      DO 59 IR=1,N
   59 VK(IR,K)=ZO
      GO TO 62
   60 VOK=ZO
      VFK=ZO
      DO 61 IR=1,N
      A=QK*XL(IR)
```

```
      B=EIGVEC(IR,K)
      VK(IR,K)=QK*DSIN(QK)*B
      VOK=VOK+DSIN(QK-A)*B
   61 VFK=VFK-DSIN(A)*B
      VO(K)=VOK*QK
      VF(K)=VFK*QK
   62 CONTINUE
C
C
C
      DO 888 J=1,M
      CALL CONSPAT(J,QJ,DELMQJ,DELPQJ,ZJO,ZJN,ZJ,PJO,PJN,PJ,PPJ,PMJ,ZPJ,
     1ZMJ,SJ,SPJ,NMJ,NPJ)
      DO 888 NR=1,J
      CALL CONSPAT(NR,QN,DELMQN,DELPQN,ZNO,ZNN,ZN,PNO,PNN,PN,PPN,PMN,ZPN
     1,ZMN,SN,SPN,NMJ,NPJ)
      UJN=ZO
      A=XL(N)
      B=ONE
      EX=EXP(ALFM*(B-A))
      ROBA=ZJN*ZNN*DIAG(EX,QJ,DELMQJ,DELPQJ,PJN,PPJ,PMJ,ZPJ,ZMJ,
     1QN,DELMQN,DELPQN,PNN,PPN,PMN,ZPN,ZMN,A,B)
      UJN=UJN+ROBA
      A=ZO
      B=XL(1)
      EX=EXP(ALFM*B)
      ROBB=ZJO*ZNO*DIAG(EX,QJ,DELMQJ,DELPQJ,PJO,PPJ,PMJ,ZPJ,ZMJ,
     1QN,DELMQN,DELPQN,PNO,PPN,PMN,ZPN,ZMN,A,B)
      UJN=UJN+ROBB
      IF(N.LT.2)GO TO 31
      DO 84 JR=1,NMJ
      A=B
      B=XL(JR+1)
      EX=EXP(ALFM*(B-A))
      UJN=UJN+ZJ(JR)*ZN(JR)*DIAG(EX,QJ,DELMQJ,DELPQJ,PJ(JR),PPJ,PMJ,ZPJ
     1,ZMJ,QN,DELMQN,DELPQN,PN(JR),PPN,PMN,ZPN,ZMN,A,B)
   84 CONTINUE
   31 A=XL(1)
      BRCJ=SPN(1)*SMICS(ZO,A,ZPN,PPN,ZMN,PMN,QN)
      BRSJ=SPN(1)*SMISS(ZO,A,ZPN,PPN,ZMN,PMN,QN)
      BRCN=SPJ(1)*SMICS(ZO,A,ZPJ,PPJ,ZMJ,PMJ,QJ)
      BRSN=SPJ(1)*SMISS(ZO,A,ZPJ,PPJ,ZMJ,PMJ,QJ)
      B=A
      IF(N.LT.2)GO TO 52
      B=XL(2)
      UJN=UJN+SJ(1)*(
     1BICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
     2BRSJ)+SN(1)*(BICSP(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISSP(A,B,ZPN,PPN
     3,ZMN,PMN,QN)*BRSN)
      UJN=UJN+SPJ(2)*(BICS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISS(A,B,ZPJ,PPJ
     1,ZMJ,PMJ,QJ)*BRSJ)+SPN(2)*(BICS(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISS(
     2A,B,ZPN,PPN,ZMN,PMN,QN)*BRSN)
      DO 50 IR=2,N
```

```
      IRR=IR-1
      A=XL(IRR)
      B=XL(IR)
      BRCJ=BRCJ+
     1SN(IRR)*SMICSP(A,B,ZPN,PPN,ZMN,PMN,QN)
     2+SPN(IR)*SMICS(A,B,ZPN,PPN,ZMN,PMN,QN)
      BRSJ=BRSJ+
     1SN(IRR)*SMISSP(A,B,ZPN,PPN,ZMN,PMN,QN)
     2+SPN(IR)*SMISS(A,B,ZPN,PPN,ZMN,PMN,QN)
      BRCN=BRCN+
     1SJ(IRR)*SMICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
     2+SPJ(IR)*SMICS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
      BRSN=BRSN+
     1SJ(IRR)*SMISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
     2+SPJ(IR)*SMISS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
      IF(IR.EQ.N)GO TO 50
      A=B
      B=XL(IR+1)
      UJN=UJN+SJ(IR)*(
     1BICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
     2BRSJ)+SPJ(IR+1)*(
     3BICS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
     4BRSJ)+SN(IR)*(
     5BICSP(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISSP(A,B,ZPN,PPN,ZMN,PMN,QN)*
     6BRSN)+SPN(IR+1)*(
     7BICS(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISS(A,B,ZPN,PPN,ZMN,PMN,QN)*
     8BRSN)
   50 CONTINUE
   52 A=B
      B=1.D0
      UJN=UJN+SJ(N)*(
     1BICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
     2BRSJ)+SN(N)*(
     3BICSP(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISSP(A,B,ZPN,PPN,ZMN,PMN,QN)*
     4BRSN)
      UJN=UJN/2.d0
      RIJ(J,NR)=UJN
      IF(NR.LT.J)RIJ(NR,J)=SCALPR(J,NR)*DAMP
      ROB=FUJN(J,NR)*UJN
D      PRINT *,J,NR,'rij=',UJN,' fjn=',FUJN(J,NR),' rob=',ROB
      IF(J.EQ.NR)ROB=ROB/2.d0
      SUMO=SUMO+VO(J)*VO(NR)*ROB
      SUMF=SUMF+VF(J)*VF(NR)*ROB
      DO 65 IR=1,N
   65 SUMM(IR)=SUMM(IR)+VK(IR,J)*VK(IR,NR)*ROB
  888 CONTINUE
C
C      PRINT *,RIJ(1,1),RIJ(2,1),RIJ(2,2)
C      PRINT *,RIJ(1,2)
C
C
      IF(M.LE.1)RETURN
      DO 999 I=1,M
```

```
      IF (EIGVAL(I).LT.Z0)GO TO 999
      SOI=Z0
      SFI=Z0
      DO 101 IR=1,N
  101 SMI(IR)=Z0
      DO 997 L=1,I
      IF(EIGVAL(L).LE.Z0)GO TO 997
D      SUM=Z0
      SUMA=Z0
      SUMB=Z0
      SUMC=Z0
      DO 995 J=1,M
      SUM1=Z0
      SUM2=Z0
      SUM3=Z0
      IF(I.GE.J)THEN
                  RIJA=RIJ(I,J)
                  IF(I.GT.J)THEN
                  QIJA=RIJ(J,I)
                  ELSE
                  QIJA=Z0
                  ENDIF
              ELSE
                  RIJA=RIJ(J,I)
                  QIJA=RIJ(I,J)
              ENDIF
      DO 994 K=1,M
      CALL FUKINJ(I,J,K,L,S1,S2,S3)
      IF(J.LE.K)THEN
                  RJKA=RIJ(K,J)
                  IF(J.LT.K)THEN
                  QJKA=RIJ(J,K)
                  ELSE
                  QJKA=Z0
                  ENDIF
              ELSE
                  QJKA=RIJ(K,J)
                  RJKA=RIJ(J,K)
              ENDIF
      IF(L.LE.K)THEN
                  RKLA=RIJ(K,L)
                  IF(L.LT.K)THEN
                  QKLA=RIJ(L,K)
                  ELSE
                  QKLA=Z0
                  ENDIF
              ELSE
                  QKLA=RIJ(K,L)
                  RKLA=RIJ(L,K)
              ENDIF
      SUM3=SUM3+S3*QJKA*QKLA
D      IF(I.EQ.J)GO TO 994
      SUM2=SUM2+S2*QJKA*RKLA
```

```
      SUM1=SUM1+S1*RJKA*QKLA
  994 CONTINUE
D      SUM=SUM-SUM1*RIJA+SUM2*QIJA
      SUMA=SUMA+SUM1*QIJA
      SUMB=SUMB+SUM2*QIJA
      SUMC=SUMC+SUM3*RIJA
      SUM=SUMA-SUMB-SUMC
  995 CONTINUE
D      PRINT *,I,L,SUMA,SUMB,SUMC,SUMB+SUMC
D      IF(I.EQ.L)SUM=SUM/2.D0
      SOI=SOI+SUM*V0(L)
      SFI=SFI+SUM*VF(L)
      DO 996 IR=1,N
  996 SMI(IR)=SMI(IR)+SUM*VK(IR,L)
  997 CONTINUE
      SUMO=SUMO+SOI*V0(I)
      SUMF=SUMF+SFI*VF(I)
      DO 998 IR=1,N
  998 SUMM(IR)=SUMM(IR)+SMI(IR)*VK(IR,I)
  999 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE FUKINJ(I,J,K,L,S1,S2,S3)
      IMPLICIT DOUBLE PRECISION(A-H,O-X,Z)
      IMPLICIT COMPLEX(Y)
      DIMENSION YDELK(50),YRP(50),YRM(50)
      COMMON /CONST/Z0,ZJ,PI,TKAP,ALFM,DALF,A2N2,XLAMM,ALFSQ
      COMMON /KINJ/YDELK,YRP,YRM
      COMMON /COMPL/YI,YP,YM,YPM,YMM,YJ
      IF(I.EQ.J)THEN
C
C      EITHER:
               IF((J.EQ.K).OR.(K.EQ.L))GO TO 999
C      OR:
C                 ONLY  S3  EXISTS:
C
      S1=Z0
      S2=Z0
      YIP=CONJG(YRP(I))
      YIM=CONJG(YRM(I))
      YJP=YRP(J)
      YJM=YRM(J)
      YKP=YRP(K)
      YKM=YRM(K)
      YLP=YRP(L)
      YLM=YRM(L)
C      PRINT *,YP,YM,YIP,YIM,YJP,YJM,
      Y1=YPM
      Y2=YMM
      YDI=YDELK(I)
```

```
      S3=DBLE(
     1YJ/(Y1-YIP)/(Y1-YIM)/(Y1-YLP)/(Y1-YLM)/(Y1-YJP)/(Y1-YJM)
     2/(Y1-YKP)/(Y1-YKM)*Y1*Y1+
     3YJ/(Y2-YIP)/(Y2-YIM)/(Y2-YLP)/(Y2-YLM)/(Y2-YJP)/(Y2-YJM)
     4/(Y2-YKP)/(Y2-YKM)*Y2*Y2-YI*
     5(YSOM(YIP)/(YIP-YJP)/(YIP-YJM)/(YIP-YLP)/(YIP-YLM)/
     6(YIP-YKP)/(YIP-YKM)*YIP*YIP
     7-YSOM(YIM)/(YIM-YJP)/(YIM-YJM)/(YIM-YLP)/(YIM-YLM)/
     8(YIM-YKP)/(YIM-YKM)*YIM*YIM)/YDI)
      RETURN
                ENDIF
C
C
      IF(J.EQ.K)THEN
C
C           EITHER NONE OF SI EXISTS:
                                      IF(K.EQ.L)GO TO 999
C           OR ONLY S1 EXISTS:
C
      YIP=CONJG(YRP(I))
      YIM=CONJG(YRM(I))
      YJPC=CONJG(YRP(J))
      YJMC=CONJG(YRM(J))
      YKP=YRP(K)
      YKM=YRM(K)
      YLP=YRP(L)
      YLM=YRM(L)
      S1=DBLE(
     1YJ/(YP-YIP)/(YP-YIM)/(YP-YLP)/(YP-YLM)/(YP-YJPC)/(YP-YJMC)
     2/(YP-YKP)/(YP-YKM)*YP*YP+
     3YJ/(YM-YIP)/(YM-YIM)/(YM-YLP)/(YM-YLM)/(YM-YJPC)/(YM-YJMC)
     4/(YM-YKP)/(YM-YKM)*YM*YM+YI*(
     5(YSOM(YKP)/(YKP-YIP)/(YKP-YIM)/(YKP-YLP)/(YKP-YLM)/
     6(YKP-YJPC)/(YKP-YJMC)*YKP*YKP
     7-YSOM(YKM)/(YKM-YIP)/(YKM-YIM)/(YKM-YLP)/(YKM-YLM)/
     8(YKM-YJPC)/(YKM-YJMC)*YKM*YKM)/YDELK(K)+
     9(YSOM(YLP)/(YLP-YIP)/(YLP-YIM)/(YLP-YJPC)/(YLP-YJMC)/
     a(YLP-YKP)/(YLP-YKM)*YLP*YLP
     b-YSOM(YLM)/(YLM-YIP)/(YLM-YIM)/(YLM-YJPC)/(YLM-YJMC)/
     c(YLM-YKP)*(YLM-YKM)*YLM*YLM)/YDELK(L)))
      S2=Z0
      S3=Z0
      RETURN
                ENDIF
C
C
      IF(K.EQ.L)THEN
      S1=Z0
      S3=Z0
C
C                          ONLY S2 EISTS IN THIS CASE
C
      YIP=CONJG(YRP(I))
```

```
      YIM=CONJG(YRM(I))
      YJPC=CONJG(YRP(J))
      YJMC=CONJG(YRM(J))
      YKPC=CONJG(YRP(K))
      YKMC=CONJG(YRM(K))
      YLP=YRP(L)
      YLM=YRM(L)
      S2=DBLE(
     1YJ/(YP-YIP)/(YP-YIM)/(YP-YLP)/(YP-YLM)/(YP-YJPC)/(YP-YJMC)
     2/(YP-YKPC)/(YP-YKMC)*YP*YP+
     3YJ/(YM-YIP)/(YM-YIM)/(YM-YLP)/(YM-YLM)/(YM-YJPC)/(YM-YJMC)
     4/(YM-YKPC)/(YM-YKMC)*YM*YM+
     5YI/YDELK(L)*(
     6YSOM(YLP)/(YLP-YIP)/(YLP-YIM)/(YLP-YJPC)/(YLP-YJMC)/(YLP-YKPC)
     7/(YLP-YKMC)*YLP*YLP-
     8YSOM(YLM)/(YLM-YIP)/(YLM-YIM)/(YLM-YJPC)/(YLM-YJMC)/(YLM-YKPC)
     9/(YLM-YKMC)*YLM*YLM))
      RETURN
           ENDIF
C
C
C          T H E          G E N E R I C          C A S E
C
C     ALL S1,S2,S3 EXIST IN THIS GENERIC CASE (I ne J, J ne K, K ne L)
C
      YIP=CONJG(YRP(I))
      YIM=CONJG(YRM(I))
      YJP=YRP(J)
      YJM=YRM(J)
      YJPC=CONJG(YJP)
      YJMC=CONJG(YJM)
      YKP=YRP(K)
      YKM=YRM(K)
      YKPC=CONJG(YKP)
      YKMC=CONJG(YKM)
      YLP=YRP(L)
      YLM=YRM(L)
      Y1=YJ/(YP-YIP)/(YP-YIM)/(YP-YLP)/(YP-YLM)*YP*YP
      Y2=YJ/(YM-YIP)/(YM-YIM)/(YM-YLP)/(YM-YLM)*YM*YM
      Y7=YSOM(YLP)/(YLP-YIP)/(YLP-YIM)*YLP*YLP
      Y8=YSOM(YLM)/(YLM-YIP)/(YLM-YIM)*YLM*YLM
      YDK=YDELK(K)
      YDL=YDELK(L)
      YPJC=(YP-YJPC)*(YP-YJMC)
      YPK=(YP-YKP)*(YP-YKM)
      YMJC=(YM-YJPC)*(YM-YJMC)
      YMK=(YM-YKP)*(YM-YKM)
      YLPJ=(YLP-YJPC)*(YLP-YJMC)
      YLMJ=(YLM-YJPC)*(YLM-YJMC)
      YLPK=(YLP-YKP)*(YLP-YKM)
      YLMK=(YLM-YKP)*(YLM-YKM)
      S1=DBLE(
     aY1/YPJC/YPK+Y2/YMJC/YMK+YI*((YSOM(YKP)/(YKP-YIP)/(YKP-YIM)/
```

```
     b(YKP-YLP)/(YKP-YLM)/(YKP-YJPC)/(YKP-YJMC)*YKP*YKP-
     1YSOM(YKM)/(YKM-YIP)/(YKM-YIM)/(YKM-YLP)/(YKM-YLM)*YKM*YKM
     2/(YKM-YJPC)/(YKM-YJMC))/YDK+(Y7/YLPJ/YLPK-Y8/YLMJ/YLMK)/YDL))
      S2=DBLE(Y1/YPJC/(YP-YKPC)/(YP-YKMC)+Y2/YMJC/(YM-YKPC)/(YM-YKMC)
     1+YI*(Y7/YLPJ/(YLP-YKPC)/(YLP-YKMC)-Y8/YLMJ/(YLM-YKPC)/(YLM-YKMC))
     2/YDL)
      Y1=YPM
      Y2=YMM
      YDI=YDELK(I)
      IF(DBLE(YDI).EQ.Z0)YDI=-YDI
      S3=DBLE(
     1YJ/(Y1-YIP)/(Y1-YIM)/(Y1-YLP)/(Y1-YLM)/(Y1-YJP)/(Y1-YJM)
     2/(Y1-YKP)/(Y1-YKM)*Y1*Y1+
     3YJ/(Y2-YIP)/(Y2-YIM)/(Y2-YLP)/(Y2-YLM)/(Y2-YJP)/(Y2-YJM)
     4/(Y2-YKP)/(Y2-YKM)*Y2*Y2-YI*
     5(YSOM(YIP)/(YIP-YJP)/(YIP-YJM)/(YIP-YLP)/(YIP-YLM)/
     6(YIP-YKP)/(YIP-YKM)*YIP*YIP
     7-YSOM(YIM)/(YIM-YJP)/(YIM-YJM)/(YIM-YLP)/(YIM-YLM)/
     8(YIM-YKP)/(YIM-YKM)*YIM*YIM)/YDI)
      RETURN
C
C
C                NONE OF  S1,S2,S3  EXISTS IN THIS CASE
C
  999                S1=Z0
                     S2=Z0
                     S3=Z0
                     RETURN
      END
```

Appendix 3
Listing of some segments of the FORTRAN code SPA1.FOR

The SPA1.FOR code implements the 1-st-order-Green-function-approach. The overall structure of the program is that of the code SPA0.FOR. Below, only those of the segments of the SPA1.FOR program are listed, which are different from the respective segments of the SPA0.FOR program.

```
C
      SUBROUTINE MSQVALUE
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION SUMM(9),SPJ(9),SPN(9),SJ(9),SN(9)
      DIMENSION EIGVAL(20),EIGVEC(9,20)
      DIMENSION XL(9),D(9)
      DIMENSION SUMMJN(9),SUMJNK(9)
      DIMENSION VO(20),VF(20)
      DIMENSION AIJ(20,20),BIJ(20,20),BETI(20),DI(20)
      DIMENSION XA(20,20),XB(20,20),ZKI(20,20),PKI(20,20)
      DIMENSION ZMM(20),ZMP(20),ZPM(20),ZPP(20)
      DIMENSION PMM(20),PMP(20),PPM(20),PPP(20)
      DIMENSION ZJ(8),ZN(8),PJ(8),PN(8)
      COMMON /EIG/EIGVAL,EIGVEC
      COMMON /L/XL,D
      COMMON /PAR/DAMP,Q,N,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ONE,PI,TKAP,ALFM,XLAMM,ALFSQ
      COMMON /MSQ/SUMO,SUMF,SUMM
      COMMON /PREC/PREC1,PREC2,IPREC3
      COMMON /KINJ/DI,ZMM,ZMP,ZPM,ZPP,AIJ,BIJ,ZKI,PKI,XA,XB,BETI,
     1PMM,PMP,PPM,PPP
      COMMON /HALF/TWO,TWOH
      WRITE(*,70)(XL(I),I=1,N)
   70 FORMAT(/,' APPROACHING CALCULATION OF MSQ-STRAINS AT SPACERS'' CLA
     1MPS',/,' FOR SPACERS LOCATED AT FOLLOWING POSITIONS:',/,15(10X,F10
     2.6,/))
      NPJ=N+1
      NMJ=N-1
      SUMO=ZO
      SUMF=ZO
      DO 1 IR=1,N
    1 SUMM(IR)=ZO
      CALL CONSTANT
      DO 62 K=1,M
      QK=EIGVAL(K)
      IF(QK.GT.ZO)GO TO 60
      VO(K)=TWO
      VF(K)=TWO*DNINT(DCOS(QK))
      GO TO 62
   60 VOK=ZO
      VFK=ZO
      DO 61 IR=1,N
      A=QK*XL(IR)
      B=EIGVEC(IR,K)
      VOK=VOK+DSIN(QK-A)*B
   61 VFK=VFK-DSIN(A)*B
      VO(K)=VOK*QK
```

```
      VF(K)=VFK*QK
   62 CONTINUE
C
C
C
      DO 999 J=1,M
      IF(DI(J).LE.ZO)GO TO 999
      CALL CONSPAT(J,QJ,DELMQJ,DELPQJ,ZJO,ZJN,ZJ,PJO,PJN,PJ,PPJ,PMJ,ZPJ,
     1ZMJ,SJ,SPJ,NMJ,NPJ)
      DO 998 NR=1,M
      IF(DI(NR).LE.ZO)GO TO 998
      CALL CONSPAT(NR,QN,DELMQN,DELPQN,ZNO,ZNN,ZN,PNO,PNN,PN,PPN,PMN,ZPN
     1,ZMN,SN,SPN,NMJ,NPJ)
      UJN=ZO
      A=XL(N)
      B=ONE
      EX=EXP(ALFM*(B-A))
      ROBA=ZJN*ZNN*DIAG(EX,QJ,DELMQJ,DELPQJ,PJN,PPJ,PMJ,ZPJ,ZMJ,
     1QN,DELMQN,DELPQN,PNN,PPN,PMN,ZPN,ZMN,A,B)
      UJN=UJN+ROBA
      A=ZO
      B=XL(1)
      EX=EXP(ALFM*B)
      ROBB=ZJO*ZNO*DIAG(EX,QJ,DELMQJ,DELPQJ,PJO,PPJ,PMJ,ZPJ,ZMJ,
     1QN,DELMQN,DELPQN,PNO,PPN,PMN,ZPN,ZMN,A,B)
      UJN=UJN+ROBB
      IF(N.LT.2)GO TO 31
      DO 84 JR=1,NMJ
      A=B
      B=XL(JR)
      EX=EXP(ALFM*(B-A))
      UJN=UJN+ZJ(JR)*ZN(JR)*DIAG(EX,QJ,DELMQJ,DELPQJ,PJ(JR),PPJ,PMJ,ZPJ
     1,ZMJ,QN,DELMQN,DELPQN,PN(JR),PPN,PMN,ZPN,ZMN,A,B)
   84 CONTINUE
   31 A=XL(1)
      BRCJ=SPN(1)*SMICS(ZO,A,ZPN,PPN,ZMN,PMN,QN)
      BRSJ=SPN(1)*SMISS(ZO,A,ZPN,PPN,ZMN,PMN,QN)
      BRCN=SPJ(1)*SMICS(ZO,A,ZPJ,PPJ,ZMJ,PMJ,QJ)
      BRSN=SPJ(1)*SMISS(ZO,A,ZPJ,PPJ,ZMJ,PMJ,QJ)
      B=A
      IF(N.LT.2)GO TO 52
      B=XL(2)
      UJN=UJN+SJ(1)*(
     1BICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
     2BRSJ)+SN(1)*(BICSP(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISSP(A,B,ZPN,PPN
     3,ZMN,PMN,QN)*BRSN)
      UJN=UJN+SPJ(2)*(BICS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISS(A,B,ZPJ,PPJ
     1,ZMJ,PMJ,QJ)*BRSJ)+SPN(2)*(BICS(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISS(
     2A,B,ZPN,PPN,ZMN,PMN,QN)*BRSN)
      DO 50 IR=2,N
      IRR=IR-1
      A=XL(IRR)
      B=XL(IR)
```

```
      BRCJ=BRCJ+
     1SN(IRR)*SMICSP(A,B,ZPN,PPN,ZMN,PMN,QN)
     2+SPN(IR)*SMICS(A,B,ZPN,PPN,ZMN,PMN,QN)
      BRSJ=BRSJ+
     1SN(IRR)*SMISSP(A,B,ZPN,PPN,ZMN,PMN,QN)
     2+SPN(IR)*SMISS(A,B,ZPN,PPN,ZMN,PMN,QN)
      BRCN=BRCN+
     1SJ(IRR)*SMICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
     2+SPJ(IR)*SMICS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
      BRSN=BRSN+
     1SJ(IRR)*SMISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
     2+SPJ(IR)*SMISS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)
      IF(IR.EQ.N)GO TO 50
      A=B
      B=XL(IR+1)
      UJN=UJN+SJ(IR)*(
     1BICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
     2BRSJ)+SPJ(IR+1)*(
     3BICS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISS(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
     4BRSJ)+SN(IR)*(
     5BICSP(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISSP(A,B,ZPN,PPN,ZMN,PMN,QN)*
     6BRSN)+SPN(IR+1)*(
     7BICS(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISS(A,B,ZPN,PPN,ZMN,PMN,QN)*
     8BRSN)
   50 CONTINUE
   52 A=B
      B=1.D0
      UJN=UJN+SJ(N)*(
     1BICSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*BRCJ+BISSP(A,B,ZPJ,PPJ,ZMJ,PMJ,QJ)*
     2BRSJ)+SN(N)*(
     3BICSP(A,B,ZPN,PPN,ZMN,PMN,QN)*BRCN+BISSP(A,B,ZPN,PPN,ZMN,PMN,QN)*
     4BRSN)
      UJN=UJN/2.d0
C      IF(J.EQ.NR)UJN=UJN/2.d0
      SUMOJN=Z0
      SUMFJN=Z0
      DO 51 IR=1,N
   51 SUMMJN(IR)=Z0
C
C
      DO 997 K=1,M
      IF(DI(K).LE.Z0)GO TO 997
      QK=DABS(EIGVAL(K))
      QKSINQKL=DSIN(QK)*QK
      VOK=VO(K)
      VFK=VF(K)
      SJNK=Z0
      SFJNK=Z0
      DO 996 IR=1,N
  996 SUMJNK(IR)=Z0
      DO 995 I=1,M
      IF(DI(I).LE.Z0)GO TO 995
      QI=DABS(EIGVAL(I))
```

```
      QISINQIL=QI*DSIN(QI)
      UKINJ=FUKINJ(K,I,NR,J)
C      IF(K.NE.I)UKINJ=UKINJ+FUKINJ(I,K,NR,J)
C      write(9,74)k,i,nr,j,ukinj
   74 format(4(2x,i1),3x,d15.8)
      SJNK=SJNK+VO(I)*UKINJ
      SFJNK=SFJNK+VF(I)*UKINJ
      DO 65 IR=1,N
   65 SUMJNK(IR)=SUMJNK(IR)+UKINJ*QISINQIL*EIGVEC(IR,I)
  995 CONTINUE
      SUMOJN=SUMOJN+VOK*SJNK
      SUMFJN=SUMFJN+VFK*SFJNK
      DO 66 IR=1,N
   66 SUMMJN(IR)=SUMMJN(IR)+SUMJNK(IR)*QKSINQKL*EIGVEC(IR,K)
  997 CONTINUE
      SUMO=SUMO+SUMOJN*UJN
      SUMF=SUMF+SUMFJN*UJN
      DO 67 IR=1,N
   67 SUMM(IR)=SUMM(IR)+SUMMJN(IR)*UJN
  998 CONTINUE
  999 CONTINUE
      RETURN
      END
C
C
      DOUBLE PRECISION FUNCTION FUKINJ(K,I,N,J)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION AIJ(20,20),BIJ(20,20),BETI(20),DI(20)
      DIMENSION XA(20,20),XB(20,20),ZKI(20,20),PKI(20,20)
      DIMENSION ZMM(20),ZMP(20),ZPM(20),ZPP(20)
      DIMENSION PMM(20),PMP(20),PPM(20),PPP(20)
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,XLAMM,ALFSQ
      COMMON /KINJ/DI,ZMM,ZMP,ZPM,ZPP,AIJ,BIJ,ZKI,PKI,XA,XB,BETI,
     1PMM,PMP,PPM,PPP
      ROB=ZO
      RM=XA(I,J)-XA(K,N)
      RP=XA(I,J)+XA(K,N)
      AA=AIJ(K,N)*AIJ(I,J)
      ROB=ROB+AA*((
     1DCOS(RM+PPM(I)-PPP(K))/ZPP(K)-DCOS(RP+PPM(I)+PMP(K))/ZMP(K))/ZPM(I
     2)+(DCOS(RM+PMM(I)-PMP(K))/ZMP(K)-DCOS(RP+PMM(I)+PPP(K))/ZPP(K))/ZM
     3M(I))
      IF (K.LT.I)THEN
                 PKIM=-PKI(I,K)
                 PKIP=PKI(K,I)
                 ZKIM=ZKI(I,K)
                 ZKIP=ZKI(K,I)
                 ELSE
                 PKIM=PKI(K,I)
                 PKIP=PKI(I,K)
                 ZKIM=ZKI(K,I)
                 ZKIP=ZKI(I,K)
                 ENDIF
```

```
      IF (K.EQ.I)THEN
                 PKIM=Z0
                 ZKIM=2.*BETI(I)
                 ENDIF
      RM=RM-PKIM
      RP=RP+PKIP
      ROB=ROB+AA*(
     1(DCOS(RM+PPP(I))/ZKIM-DCOS(RP+PPP(I))/ZKIP)/ZPP(I)+
     2(DCOS(RM+PMP(I))/ZKIM-DCOS(RP+PMP(I))/ZKIP)/ZMP(I)-
     3(DCOS(RM+PPM(I))/ZKIM-DCOS(RP+PPM(I))/ZKIP)/ZPM(I)-
     4(DCOS(RM+PMM(I))/ZKIM-DCOS(RP+PMM(I))/ZKIP)/ZMM(I))
      IF (K.EQ.N) GO TO 30
      RM=XA(I,J)-XB(K,N)
      RP=XA(I,J)+XB(K,N)
      AA=BIJ(K,N)*AIJ(I,J)
      ROB=ROB+AA*((
     1DCOS(RM+PPM(I)-PPP(N))/ZPP(N)-DCOS(RP+PPM(I)+PMP(N))/ZMP(N))/ZPM(I
     2)+(DCOS(RM+PMM(I)-PMP(N))/ZMP(N)-DCOS(RP+PMM(I)+PPP(N))/ZPP(N))/ZM
     3M(I))
      IF (N.LT.I)THEN
                 PKIM=-PKI(I,N)
                 PKIP=PKI(N,I)
                 ZKIM=ZKI(I,N)
                 ZKIP=ZKI(N,I)
                 ELSE
                 PKIM=PKI(N,I)
                 PKIP=PKI(I,N)
                 ZKIM=ZKI(N,I)
                 ZKIP=ZKI(I,N)
                 ENDIF
      IF (N.EQ.I)THEN
                 PKIM=Z0
                 ZKIM=2.*BETI(I)
                 ENDIF
      RM=RM-PKIM
      RP=RP+PKIP
      ROB=ROB+AA*(
     1(DCOS(RM+PPP(I))/ZKIM-DCOS(RP+PPP(I))/ZKIP)/ZPP(I)+
     2(DCOS(RM+PMP(I))/ZKIM-DCOS(RP+PMP(I))/ZKIP)/ZMP(I)-
     3(DCOS(RM+PPM(I))/ZKIM-DCOS(RP+PPM(I))/ZKIP)/ZPM(I)-
     4(DCOS(RM+PMM(I))/ZKIM-DCOS(RP+PMM(I))/ZKIP)/ZMM(I))
   30 IF(I.EQ.J)GO TO 999
      RM=XB(I,J)-XA(K,N)
      RP=XB(I,J)+XA(K,N)
      AA=AIJ(K,N)*BIJ(I,J)
      ROB=ROB+AA*((
     1DCOS(RM+PPM(J)-PPP(K))/ZPP(K)-DCOS(RP+PPM(J)+PMP(K))/ZMP(K))/ZPM(J
     2)+(DCOS(RM+PMM(J)-PMP(K))/ZMP(K)-DCOS(RP+PMM(J)+PPP(K))/ZPP(K))/ZM
     3M(J))
      IF (K.LT.J)THEN
                 PKIM=-PKI(J,K)
                 PKIP=PKI(K,J)
                 ZKIM=ZKI(J,K)
```

```
                ZKIP=ZKI(K,J)
                ELSE
                PKIM=PKI(K,J)
                PKIP=PKI(J,K)
                ZKIM=ZKI(K,J)
                ZKIP=ZKI(J,K)
                ENDIF
      IF (K.EQ.J)THEN
                PKIM=ZO
                ZKIM=2.*BETI(J)
                ENDIF
      RM=RM-PKIM
      RP=RP+PKIP
      ROB=ROB+AA*(
     1(DCOS(RM+PPP(J))/ZKIM-DCOS(RP+PPP(J))/ZKIP)/ZPP(J)+
     2(DCOS(RM+PMP(J))/ZKIM-DCOS(RP+PMP(J))/ZKIP)/ZMP(J)-
     3(DCOS(RM+PPM(J))/ZKIM-DCOS(RP+PPM(J))/ZKIP)/ZPM(J)-
     4(DCOS(RM+PMM(J))/ZKIM-DCOS(RP+PMM(J))/ZKIP)/ZMM(J))
      IF(K.EQ.N)GO TO 999
      RM=XB(I,J)-XB(K,N)
      RP=XB(I,J)+XB(K,N)
      AA=BIJ(K,N)*BIJ(I,J)
      ROB=ROB+AA*((
     1DCOS(RM+PPM(J)-PPP(N))/ZPP(N)-DCOS(RP+PPM(J)+PMP(N))/ZMP(N))/ZPM(J
     2)+(DCOS(RM+PMM(J)-PMP(N))/ZMP(N)-DCOS(RP+PMM(J)+PPP(N))/ZPP(N))/ZM
     3M(J))
      IF (N.LT.J)THEN
                PKIM=-PKI(J,N)
                PKIP=PKI(N,J)
                ZKIM=ZKI(J,N)
                ZKIP=ZKI(N,J)
                ELSE
                PKIM=PKI(N,J)
                PKIP=PKI(J,N)
                ZKIM=ZKI(N,J)
                ZKIP=ZKI(J,N)
                ENDIF
      IF (N.EQ.J)THEN
                PKIM=ZO
                ZKIM=2.*BETI(J)
                ENDIF
      RM=RM-PKIM
      RP=RP+PKIP
      ROB=ROB+AA*(
     1(DCOS(RM+PPP(J))/ZKIM-DCOS(RP+PPP(J))/ZKIP)/ZPP(J)+
     2(DCOS(RM+PMP(J))/ZKIM-DCOS(RP+PMP(J))/ZKIP)/ZMP(J)-
     3(DCOS(RM+PPM(J))/ZKIM-DCOS(RP+PPM(J))/ZKIP)/ZPM(J)-
     4(DCOS(RM+PMM(J))/ZKIM-DCOS(RP+PMM(J))/ZKIP)/ZMM(J))
  999 FUKINJ=ROB
      RETURN
      END
C
C
```

```
      SUBROUTINE CONSTANT
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION EIGVAL(20),EIGVEC(9,20)
      DIMENSION AIJ(20,20),BIJ(20,20),BETI(20),DI(20)
      DIMENSION XA(20,20),XB(20,20),ZKI(20,20),PKI(20,20)
      DIMENSION ZMM(20),ZMP(20),ZPM(20),ZPP(20)
      DIMENSION PMM(20),PMP(20),PPM(20),PPP(20)
      COMMON /EIG/EIGVAL,EIGVEC
      COMMON /PAR/DAMP,QU,NOS,M,ALF,DEL,DNU,DLAM,DSTR
      COMMON /CONST/ZO,ZJ,PI,TKAP,ALFM,XLAMM,ALFSQ
      COMMON /KINJ/DI,ZMM,ZMP,ZPM,ZPP,AIJ,BIJ,ZKI,PKI,XA,XB,BETI,
     1PMM,PMP,PPM,PPP
      DO 1 I=1,M
      B=DAMP*SCALPR(I,I)/2.+DSTR
      D=EIGVAL(I)**2-B*B
      IF(D.LE.ZO)THEN
                 DI(I)=ZO
                 WRITE(*,*)' MODE ',I,' IS DAMPED-OUT'
                 GO TO 1
                 ENDIF
      D=DSQRT(D)
      DI(I)=D
      CALL XI(D-DNU,B-DLAM,PMM(I),ZMM(I))
      CALL XI(D-DNU,B+DLAM,PMP(I),ZMP(I))
      CALL XI(D+DNU,B-DLAM,PPM(I),ZPM(I))
      CALL XI(D+DNU,B+DLAM,PPP(I),ZPP(I))
    1 BETI(I)=B
      D=DI(1)
      IF(D.LE.ZO)GO TO 99
      D1=ZJ/D
      AIJ(1,1)=D1
      BIJ(1,1)=ZO
      XA(1,1)=ZO
      XB(1,1)=ZO
      DD=D*D
      B=BETI(1)
      CALL XI(D+D,B+B,PKI(1,1),ZKI(1,1))
      IF(M.LE.1)RETURN
      DO 2 N=2,M
      DN=DI(N)
      IF(DN.LE.ZO)GO TO 2
      SCP=SCALPR(1,N)
      BN=BETI(N)
      BMBN=B-BN
      DNBKMBN=2.DO*BMBN*DN
      DN=DN*DN
      ROB1=BMBN**2+DD
      ROB2=DSQRT(ROB1)
      ROB3=ROB1-DN
      ROB4=DSQRT((ROB1+DN)**2-4.*DD*DN)
      AIJ(1,N)=-ROB2/ROB4*D1*SCP*DAMP
      BIJ(1,N)=DAMP*SCP/ROB4
      XA(1,N)=DATAN2(D*ROB3,BMBN*(ROB1+DN))
```

```
      XB(1,N)=DATAN2(ROB3,DNBKMBN)
    2 CONTINUE
   99 IF(M.LE.1)RETURN
      DO 999 K=2,M
      D=DI(K)
      IF(D.LE.ZO)GO TO 999
      D1=ZJ/D
      AIJ(K,K)=D1
      BIJ(K,K)=ZO
      XA(K,K)=ZO
      XB(K,K)=ZO
      DD=D*D
      B=BETI(K)
      CALL XI(D+D,B+B,PKI(K,K),ZKI(K,K))
      KM1=K-1
      DO 3 N=1,KM1
      DN=DI(N)
      IF(DN.LE.ZO)GO TO 3
      SCP=SCALPR(K,N)
      BN=BETI(N)
      BMBN=B-BN
      DNBKMBN=2.DO*BMBN*DN
      DNN=DN*DN
      ROB1=BMBN**2+DD
      ROB2=DSQRT(ROB1)
      ROB3=ROB1-DNN
      ROB4=DSQRT((ROB1+DNN)**2-4.*DD*DNN)
      AIJ(K,N)=-ROB2/ROB4*D1*SCP*DAMP
      BIJ(K,N)=DAMP*SCP/ROB4
      XA(K,N)=DATAN2(D*ROB3,BMBN*(ROB1+DNN))
      XB(K,N)=DATAN2(ROB3,DNBKMBN)
      CALL XI(D-DN,B+BN,PKI(K,N),ZKI(K,N))
      CALL XI(D+DN,B+BN,PKI(N,K),ZKI(N,K))
    3 CONTINUE
      IF(K.EQ.M)GO TO 999
      KP1=K+1
      DO 4 N=KP1,M
      DN=DI(N)
      IF(DN.LE.ZO)GO TO 4
      SCP=SCALPR(K,N)
      BN=BETI(N)
      BMBN=B-BN
      DNBKMBN=2.DO*BMBN*DN
      DN=DN*DN
      ROB1=BMBN**2+DD
      ROB2=DSQRT(ROB1)
      ROB3=ROB1-DN
      ROB4=DSQRT((ROB1+DN)**2-4.*DD*DN)
      AIJ(K,N)=-ROB2/ROB4*D1*SCP*DAMP
      BIJ(K,N)=DAMP*SCP/ROB4
      XA(K,N)=DATAN2(D*ROB3,BMBN*(ROB1+DN))
      XB(K,N)=DATAN2(ROB3,DNBKMBN)
    4 CONTINUE
```

```
  999 CONTINUE
      RETURN
      END
C
```