ENGINEERING SOFTWARE

Proceedings of the 1st International Conference
held at Southampton University, September 1979

Edited by
R.A. Adey, *Computational Mechanics*

PENTECH PRESS
London: Plymouth

First published 1979 by
Pentech Press Limited,
Estover Road, Plymouth
Devon

ISBN 0 7273 0501 8

Printed in Great Britain by
Redwood Burn Limited
Trowbridge & Esher

ENGINEERING SOFTWARE

Preface

This book contains an edited version of the papers presented at the First International Conference and Exhibition on Engineering Software which was held at Southampton University in September 1979.

The Conference was devoted to the development and application of engineering software with applications in Civil, Mechanical, Structural, Electrical, Electronic and most other branches of engineering. The aim was to provide a forum for the presentation and discussion of recent advances in engineering software and as such this book comprises a 'state of the art' collection of papers in this important field.

The impact of computers on the engineering community has created a new science and industry for the development and production of engineering software. The extensive use of computer packages for stress analysis, computer-design of electronic systems, simulation of traffic and water management, etc., provide but a few examples of the increased use of engineering software and new fields are continuously being developed.

The book is divided into sections covering major application areas and some sections on general techniques. The first session is devoted to certainly the most widely used analysis method of engineering, the finite element method. Papers in this section cover finite element systems designed to tackle specialised problems such as non-linear analysis and more unusually the development of finite element systems for micro-computers. One of the largest sections is devoted to engineering software in structures and stress analysis. Applications vary from off-shore structures, shear walls to the strength of fabrics. Further sections cover fluid mechanics and water resources, electrical and electronic engineering, civil engineering and mechanical engineering.

A series of papers on the important subjects of documentation and standards and software techniques are included in the Software Techniques section and a further multidisciplinary section covers Computer Aided Design Techniques.

The Editor

CONTENTS

SESSION I FINITE ELEMENT SYSTEMS

SESSION II STRUCTURES AND STRESS ANALYSIS

SESSION VI ELECTRICAL AND ELECTRONIC ENGINEERING

SESSION VII CIVIL ENGINEERING

SESSION VIII MECHANICAL ENGINEERING

SESSION I

FINITE ELEMENT SYSTEMS

A GENERAL PURPOSE TWO-DIMENSIONAL MESH GENERATOR

Schoofs A.J.G., van Beukering L.H.Th.M., Sluiter M.L.C.

Department of Mechanical Engineering
Eindhoven University of Technology, The Netherlands.

INTRODUCTION

Composing and checking input for finite element programs is very labour intensive; this is particular true for the division of the area to be studied into elements. In the past many programs, called mesh generators, have been developed in order to automate this job. A survey of these programs is given by Buell W.R. and Bush B.E. (1973). This paper deals with a mesh generator for two-dimensional areas; the principal characteristics of this mesh generator, named TRIQUAMESH, are:
1. a user oriented input language with debugging aid is provided; the user will only have to supply simple composable input data.
2. both single and multiple coherent two-dimensional areas with a complex geometry can be divided into triangular and/or quadrilateral elements.
3. easy specification of the magnitude of the elements.
4. substructuring facilities have been incorporated.
5. the shape of the generated elements is optimised.
6. the mesh generator has some possibilities to reduce the bandwidth of the assembled structural matrices.
7. the output of the mesh generator can be used directly as a part of the input for three finite element programs, including ASKA and MARC.

After introduction of some basic concepts, the method used in TRIQUAMESH will be dealt with. Afterwards the use and possibilities of TRIQUAMESH will be illustrated by means of some examples.

BASIC CONCEPTS

The area to be divided G can be divided into ns subareas G_1, ... G_{ns} in order to specify element and material properties and to define the substructuring of the area G. It is demanded that subareas are simple coherent; n-fold coherent areas can be made simple coherent by making at least n-1 cuts. The contour of an area G is C, subarea G_s has C_s as contour and the overall contour of all subareas is C^* (fig. 1).

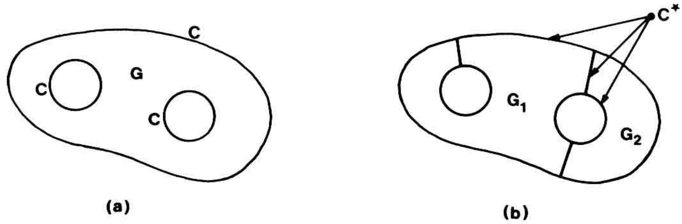

Figure 1. An arbitrary two-dimensional area (a) and a division thereof in simple coherent subareas (b).

In G a number of so-called basis points is fixed, by means of numbers and coordinates. These points will supply a basis for the geometrical description of contour C^* and for the determination of other user wishes, for example the desired element size. With two or more basis points oriented elementary curves can be defined, for example straight lines, arcs etc. (fig. 2a). A contour part (identified by a number) is a non-branched coupling of elementary curves and has an orientation. The geometric description of the contour part consists of the coupling of descriptions of the constituent elementary curves (fig. 2b).

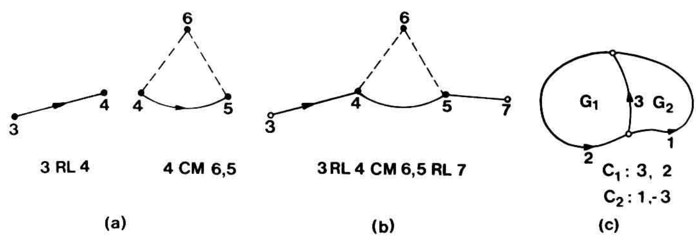

Figure 2. Description of elementary curves (a), a contour part (b) and subcontours (c).

Subcontour C_s consists of the coupling in a closed curve of one or more contour parts. For reasons of univocality, C_s is described by denoting the numbers of the anti-clockwise sequence of joining contour parts. If a contour part is met in a direction opposite to its orientation, the number of this contour part is denoted negatively (fig. 2c). A substructure is defined by one or more subareas. All substructures together form one structure: the area to be divided G. The concepts mentioned are hierarchically ordered. The basis points define elementary curves. These define contourparts etc.:

Basispoints → elementary curves → contourparts →
subareas → substructures → structure.
During the division of area G into elements, use is made of the roughness function g2(x,y). It is postulated that in G, g2 is directly proportional to the desired magnitude of the element sides; the length ℓ of any element side between the points (x_1, y_1) and (x_2, y_2) therefore shall have to be "the best fit possible":

$$\ell = \tfrac{1}{2}\{g2(x_1, y_1) + g2(x_2, y_2)\}.RI \quad (1)$$

The proportional constant RI has the dimension of length and will be called the standard element side. Equation 1 is the definition of the roughness concept.

DESCRIPTION OF THE METHOD

Globaly TRIQUAMESH has been developed as follows:
1. Checking and manipulation of input data.
2. Generation of nodal points on the total contour C^*.
3. Division of subareas into elements.
4. Post-processing, such as: optimization of element shape, bandwidth reduction, transformation to elements with more nodal points and output.

The aspects mentioned above will be described subsequently.

Checking and manipulation of input data

An user-oriented input language has been developed so that the input can be interpreted simply and elaborate tests for errors in the input data are possible. The program expects the input to be delivered by means of punchcards or by means of a file to be found on a disk-unit. From this input, arrays are determined which will serve as input parameters for the next steps. During processing, the input is also checked for syntax and semantic errors. Possible error messages are for instance:

```
1 TRIAX3 2,5)
        *       >>>> LEFT PARENTHESIS EXPECTED
          *     >>>> UNKNOWN SUBAREA
```

Generation of contour points

The user will have to supply values for the roughness function g2(x,y) for each of the basis points in area G. Doing so, using a chosen roughness behaviour along the contour, the roughness on the total contour C^* is fixed. Starting from this, and together with the user given standard element side RI, the nodal points to be generated on C^* will be determined. Because of the assembly of a subcontour out of contour parts, which in turn consist of elementary curves, it will only be necessary to explain the generation of nodal points on an elementary curve. Consider an elementary curve K with length ℓ and curvilinear coordinate s, $0 \leq s \leq \ell$; for reasons of simplicity a non essential simplification to a straight line is made. On curve K a number of basis points are denoted, numbered locally with j, j = 0,1,...,m, which divide the curve in m pieces, and, by means of the user given values gk_j, a piecewise harmonic roughness function is defined. (fig. 3):

$$gk(s) = \frac{gk_j+gk_{j+1}}{2} + \frac{gk_j-gk_{j+1}}{2} \cdot \cos \frac{s-s_j}{s_{j+1}-s_j} \cdot \pi \; ; \; j = 0..m \quad (2)$$

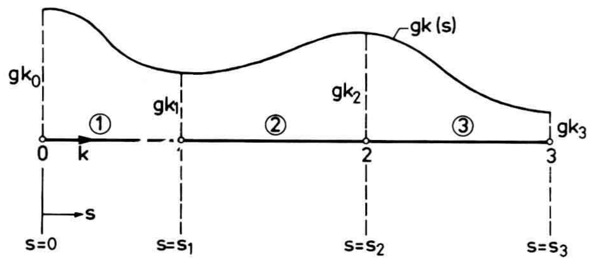

Figure 3. Harmonic roughness function on an elementary curve.

Starting form this function gk(s) nodal points are generated on curve K, this is done in two steps: first the determination of the number of nodal points and subsequently the computation of the correct location. Suppose that n-1 nodal points will have to be generated (and therefore n element sides) on the elementary curve (fig. 4).

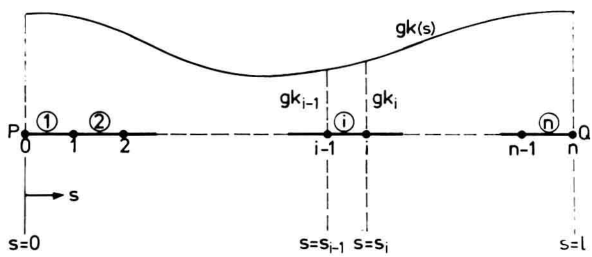

Figure 4. Nodal points and element sides on the curve PQ.

From equation 1 for element side i the following can be derived:

$$\frac{s_i-s_{i-1}}{gk_i+gk_{i-1}} = \frac{s_{i+1}-s_i}{gk_{i+1}-gk_i} \cdot (i = 1,2,...,n-1) \quad (3)$$

This can be met "as good as possible" for all element sides, by computing n as follows:

$$n = \frac{1}{RI} \cdot \int_0^{\ell} \frac{\ell}{gk(s)} \, ds \quad (4)$$

After which n is rounded off to an integer in a suitable way. As soon as the curvilinear coordinates $s_1 \ldots s_{n-1}$ are known, one can easily determine the coordinates in the overall two-dimensional system.

Generating of elements in a subarea

A contour is defined by sequentially connected nodal points on this contour. The connection is made by straight lines (the element sides). Every subarea will have to be divided either in triangles or quadrilaterals, depending upon the user given element type. The nodal points on contour C_s of subarea G_s are numbered locally 1 ... ncp (fig. 5).

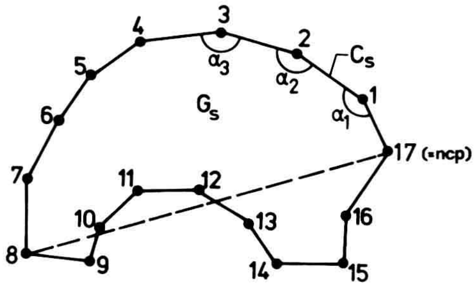

Figure 5. Local numbering of contourpoints on C_s.

Subarea G_s is concave whenever one of the angles enclosed by the contour $\alpha_i > \pi$. Let i and j, where $i \neq j$, be nodal points on C_s such that the interconnecting line between i and j lies completely within G_s. Whilst dividing G_s into elements such lines are frequently used, and an instantanuous check will have to be made to see whether this line is actually within G_s; for instance the connection between points 8 and 17 in fig. 5 is not acceptable. These checks for concave areas are quite complicated and therefor a concave subarea is split into two or more convex partial areas, after which these areas are divided into elements.

Splitting a concave subarea into convex partial areas A nodal point is called concave if $\alpha_i > \pi$, (fig. 5). The splitting is done by the following steps:

1. Take a concave nodal point on the contour; call this point P. If no such point exists, the area will be convex.
2. Determine the accumulation V_1 of nodal points on the contour which are visible from P.
3. Determine out of V_1 that nodal point Q in such a way that, based on given criteria, PQ is the best splitting line.
4. Determine the accumulation V_2 of nodal points on the contour which are visible from Q.
5. Define on PQ a roughness function, based on the roughness values of the nodal points in V_1 and V_2.

6. Generate, using that roughness function, nodal points on PQ.
7. Define two new areas separated by line PQ.
8. Continue with step 1 for both areas.

Explanation Step 1. Point P is chosen to be the most concave point on the contour or to be the middle point of a series of almost equally concave points. Steps 2 and 4. The determination of visible points is illustrated with an example (fig. 6a). Consider a continuous contour; β_k is defined as being the angle between line P-K and the tangent of the contour in P; Fig. 6b shows β_k as function of curvilinear coordinate s.

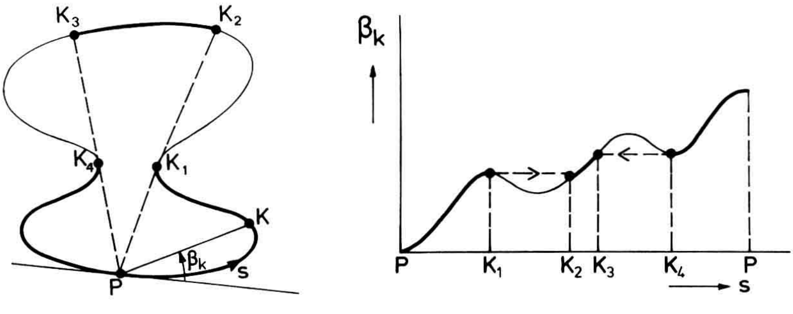

Figure 6. Determination of from P visible points.

Points of interest in the determination of visible points are those concave points L of the contour where β_L is a local extremum; not visible will be:
1. Points K with $s_K > s_L$ and $\beta_K < \beta_L$ if β_L is a local maximum; for example points between K_1 and K_2.
2. Points K with $s_K < s_L$ and $\beta_K > \beta_L$ if β_L is a local minimum; for example points between K_3 and K_4.

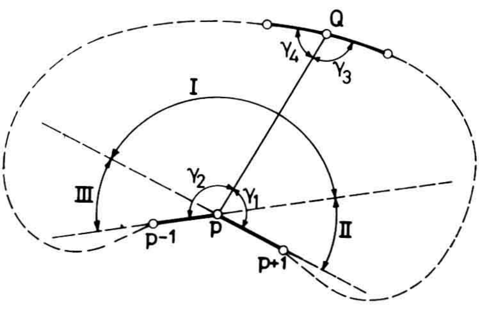

Figure 7. Search for a suitable splitting-line from point P.

Step 3. Consider fig. 7: The points P^{-1} and P^{+1} are the neighbouring points of P. The lines PP^{-1} and PP^{+1} divide the visible area in sectors I, II and III. At first the most suitable point