1988-12

A computer program for the installation of blading in a turbomachinery rotor resulting in minimum unbalance

Yardimoglu, Ali Riza

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A Computer Program for the Installation
of Blading in a Turbomachinery Rotor
Resulting in Minimum Unbalance

by

Ali Riza Yardimoglu

December 1988

Thesis Advisor: Paul F. Pucci

~~Further distribution only as directed by the Naval Postgraduate School, Monterey, CA, Code 69Pc, or higher authority, to prevent premature dissemination of information in a development state.~~

~~DESTRUCTION NOTICE - Destroy by any method that will prevent disclosure of contents or reconstruction of the document.~~

THIS PAGE INTENTIONALLY LEFT BLANK

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:

3. DISTRIBUTION/AVAILABILITY OF REPORT: ~~Further distribution only as directed by the Naval Postgraduate School, Monterey, CA, Code 69Pc, or higher authority, to prevent premature dissemination of information in a development state.~~

4. PERFORMING ORGANIZATION REPORT NUMBER(S):

5. MONITORING ORGANIZATION REPORT NUMBER(S): DOWNGRADED APPROVED FOR PUBLIC RELEASE

6a. NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School

6b. OFFICE SYMBOL *(If applicable)*: Code 69

7a. NAME OF MONITORING ORGANIZATION:

6c. ADDRESS *(City, State, and ZIP Code)*: Monterey, California 93943-5000

7b. ADDRESS *(City, State, and ZIP Code)*: Monterey, California 93943-5000

8a. NAME OF FUNDING/SPONSORING ORGANIZATION:

8b. OFFICE SYMBOL *(If applicable)*:

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c. ADDRESS *(City, State, and ZIP Code)*:

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*: A Computer Program for the Installation of Blading in a Turbomachinery Rotor Resulting in Minimum Unbalance.

12. PERSONAL AUTHOR(S): YARDIMOGLU, Ali Riza

13a. TYPE OF REPORT: Master's Thesis

13b. TIME COVERED: FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*: 1988 December

15. PAGE COUNT: 75

16. SUPPLEMENTARY NOTATION: The view expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*: Rotor Blade Balancing, Centrifugal Force, FORTRAN 77

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The assembly of blades of different masses in a turbomachinery rotor may result in an unbalanced force acting on the rotor during rotation. There is a blade sequence for which a minimum imbalance exists. In this thesis, an algorithm was devised and a computer program was written to determine the sequence for minimum imbalance.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL: Professor Pucci

22b. TELEPHONE *(Include Area Code)*: (408) 646-2363

22c. OFFICE SYMBOL: 69Pc

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—606-24.

UNCLASSIFIED

A Computer Program for the Installation of Blading in a Turbomachinery Rotor Resulting in Minimum Unbalance

by

Ali Riza Yardimoglu
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1982

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1988

Author: [redacted]
Ali Riza Yardimoglu

Approved: [redacted]
Paul F. Pucci, Thesis Advisor

[redacted]
Anthony J. Healey, Chairman,
Department of Mechanical Engineering

[redacted]
Gordon E. Schacher, Dean of
Science and Engineering

ABSTRACT

The assembly of blades of different masses in a turbomachinery rotor may result in an unbalanced force acting on the rotor during rotation. There is a blade sequence for which a minimum imbalance exists. In this thesis, an algorithm was devised and a computer program was written to determine the sequence for minimum imbalance.

TABLE OF CONTENTS

TABLE OF SYMBOLS AND ABBREVIATIONS

Note: (166) refers to the maximum variable dimension (maximum blade number)

COMPUTER SYMBOL	TEXT VARIABLE	DESCRIPTION
A (166)	A	Angle of each blade position from horizontal axis of the rotor's center of gravity (in degree)(counter clockwise, after looking forward).
ANG	Ang	Occupied angle by each blade (in degree).
C, L		Counters used in program for first blade set up.
	F	1. Centrifugal unbalanced force with respect to rotation. 2. Static unbalanced force.
F (166)	Fc	The centrifugal force magnitude acting on each blade.
FX (166)	Fcx	X-component of the F(166)
FY (166)	Fcy	Y-component of the F (166)

FXN (166)	Fxn	X-component of the recorded minimum unbalanced centrifugal force acting on each blade
FYN (166)	Fyn	Y-component of the recorded minimum force acting on each blade
I,K		Counters used in program for marking the blade position
IS		Counter used in program for the stage number selection
IX		Section marking for blade swapping with respect to angle boundaries THH and THL, respectively
IY		Section marking for blade swapping with respect to angle boundaries TH and TL, respectively

INC		The range of option (OPT)
ISN		The parameter to choose the best blade to swap with respect to the average mass
J, JJ and JN		Counters used in program for iteration process
M (166)	M	Mass value of each blade
	m	Mass value of the rigid rotor
MAVG	Mavg	Average Mass value of the blades
	mv	Approximate variation range of average mass value in each data file
N	N	Total blade number
NETUNB	Netunb	The total unbalanced centrifugal force magnitude which is acting on rotor's center of gravity
NTUNB	Ntunb	The recorded minimum total unbalanced centrifugal force magnitude after all iteration process

OPT	Opt	Option number which chooses the best mass value to swap and refers to the range of searching with respect to the average mass value
P (166), PPOS	P, Ppos	Mass value of each blade after setting up or swapping them with each other
PN (166)	Pn	Same as P (166) but after all iteration process for finding the minimum total unbalanced centri-fugal force
POS	Pos	The position number of blades which are numbered counter clockwise starting from horizontal axis of the rotor's center of gravity
POSN	Posn	Same as POS but at the end of the option range for further usage in the program
POSWP	Poswp	Same as POS but at the opposite side (180 degrees) of the rotor (swapping with POS)
POSWPN	Poswpn	Same as POSWP but at the end of the option range for further usage in the program

R (166)	R	Radius from the center of gravity of each blade to the center of gravity of the rotor
RA (166)		Same as A (166) but in radian
RAD (166), RADPO		Same as R (166) but after setting up or swapping the blades
RADN (166)		Same as RAD (166) but after all iteration process for finding the minimum total unbalanced centrifugal force
RANG		Same as ANG but in radian
RATIO	Ratio	Ratio for the TOTFY to the TOTFX
RUN	Run	Desired number of iteration for program
S		Counter used in program for selection of compressor or turbine balance calculation
THETA	Theta	Direction of the total unbalanced centrifugal force (NETUNB) from the horizontal axis of rotor's center of gravity (in degree) (counter clockwise, aft looking forward)

THETAN	Thetan	Direction of the recorded minimum total unbalanced centrifugal force (NTUNB) from the horizontal axis of rotor's center of gravity (in degree) (counter clockwise, aft looking forward)
TH, TL	Th, Tl	The angle boundaries to choose the best mass value to swap with respect to the opposite side of the total unbalanced centrifugal force direction (high and low respectively) (in degrees)
THH, THL	Thh, Thl	The angle boundaries to choose the best mass value to swap with respect to total unbalanced centri- fugal force direction (high and low, respectively) (in degrees
THM	Thm	Mean angle of the angle boundaries THH and THL (in degree)
TM	Tm	Mean angle of the angle boundaries TH and TL (in degree)
THNET	Thnet	Same as THETA but in radian
THNETN	Thnetn	Same as THETAN but in radian

THR	Thr	Direction of the total unbalanced centrifugal force (NETUNB)
TOTFX	Totfx	X-component of NETUNB
TOTFXN	Totfxn	X-component of NTUNB
TOTFY	Totfy	Y-component of NETUNB
TOTFYN	Totfyn	Y-component of NTUNB
TOTM	Totm	The total mass value of blade masses
TSWAP	Tswap	The opposite angle (180°) to the THETA (othjer side of unbalanced force direction) (in degrees)
TSWBD, THBLD	Tswbd, Thbld	The angles of blade positions from horizontal axis for swapping with each other (in degrees)
UNB (22)		Variable definition for counting and storing NETUNB. (22 refers to index number for comparing the force magnitudes with each other in Alternate Swap Case in the program)
W	w	Angular velocity of the rotor at axis of rotation
Z	z	Variable definition used in the program to define of quadrant of THR

ACKNOWLEDGEMENT

My sincere appreciation goes to all the instructors who have contributed to the improvement of my educational experience at the Naval Postgraduate School. This thesis combines both the knowledge given by them and the presentation of what I have learned. I would also like to express my gratitude to Professor Paul F. Pucci for his patient help, suggestions, and constructive criticism.

I. INTRODUCTION

The current improvements in turbomachinery design has led to high speed and low weight components to obtain higher power densities [Ref. 1]. To achieve these higher speeds, the imbalance of the rotating components must be minimized [Ref. 2].

Many turbomachinery rotors, compressor or turbines, consist of a rotating element such as a disk or a drum into which blades are inserted at the periphery. Blades can be fastened to the rotors in several ways. One of the most common way is known as the "fir-tree" method [Ref. 3]. The roots of the blades were formed with a fir-tree profile which is shown in Figure 1.1. Furthermore, this spline form of blade root is effectively located in its position on the rotor radially and resists the blade's centrifugal force, The blades also could be secured by pinning each side.

Even though the rotor drum and disk is dynamically balanced, the insertion of blades with slightly different masses due to the manufacturing tolerances will cause to an unbalanced force.

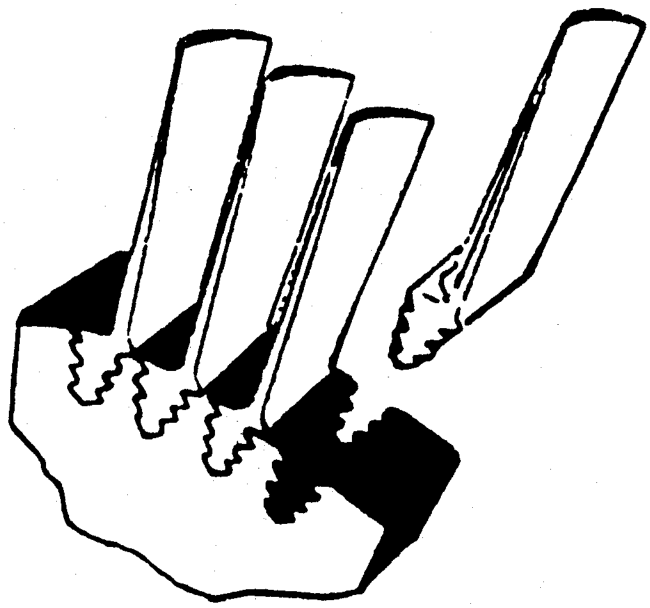

Figure 1.1 Fir-Tree Method of Attaching Blades

The objective of this thesis is to develop a method to determine the particular sequence of blade insertions which produces the minimum unbalance.

An algorithm for achieving the minimum unbalance was developed and a FORTRAN 77 computer program was written for use on an IBM type personal computer.

Chapter II summarizes the analytic background for the computer program, Chapter III describes the algorithm employed in the computer program (RBB), and Chapter IV presents the results of some sample runs. These results are compared with the results obtained from the computer program BLADEBAL.DEP which was developed by Frantz and Kennedy of the General Electric Company [Ref. 4].

The concluding remarks are in Chapter V which discusses the developments which are expected in the future and some recommendations.

II. ANALYTIC BACKGROUND

The rotor balancing procedure is a fundamental criteria in the proper and effective design of turbo-machinery. An unbalanced rotor, especially at high speed operations, will cause both the machine foundation vibration and unwanted support bearing reaction forces [Ref. 5, 6].

In the manufacture of the disk or drum, dimensional inaccuracies and manufacturing tolerances are the major causes of unbalance in this component. Normal rotor balancing techniques, such as material removal or added weights, will result in a balanced disk or drum.

The manufacturing tolerances of the blades yield slightly differing blade masses. When inserted into the disk or drum they will produce an unbalanced force whose magnitude and direction will depend on the sequence in which the blades have been inserted. An "optimum" sequence will yield the minimum unbalance. This unbalance could then be reduced by adding weights or removing material from the disk or drum. It is also possible that the unbalanced force vector of the blades be combined in such a manner as to reduce the required added weights or mass removal.

The analytic model developed in this thesis deals only with the contribution of the blades to the rotor assembly unbalance.

Each rotating blade mass has a centrifugal force acting radial outward. The vector sum of all blade centrifugal forces results in an unbalanced force in some particular direction, shown in Figure 2.1.

To seek a lower minimum unbalance, the blades must be rearranged. The rearrangement of the blade sequence is accomplished by the successive exchange or swapping of positions of two blades dictated by the logic of the analytic program. Repetitive trials are made and the blade sequence yielding the minimum unbalance is stored and displayed at the end of the trial runs.

In this thesis, only single, independent blades which can be moved to any position on the rotor are considered. In some assemblies one or more blades have fixed positions and cannot be moved. In others, a pair of adjacent (perhaps joined together) may have a fixed position.

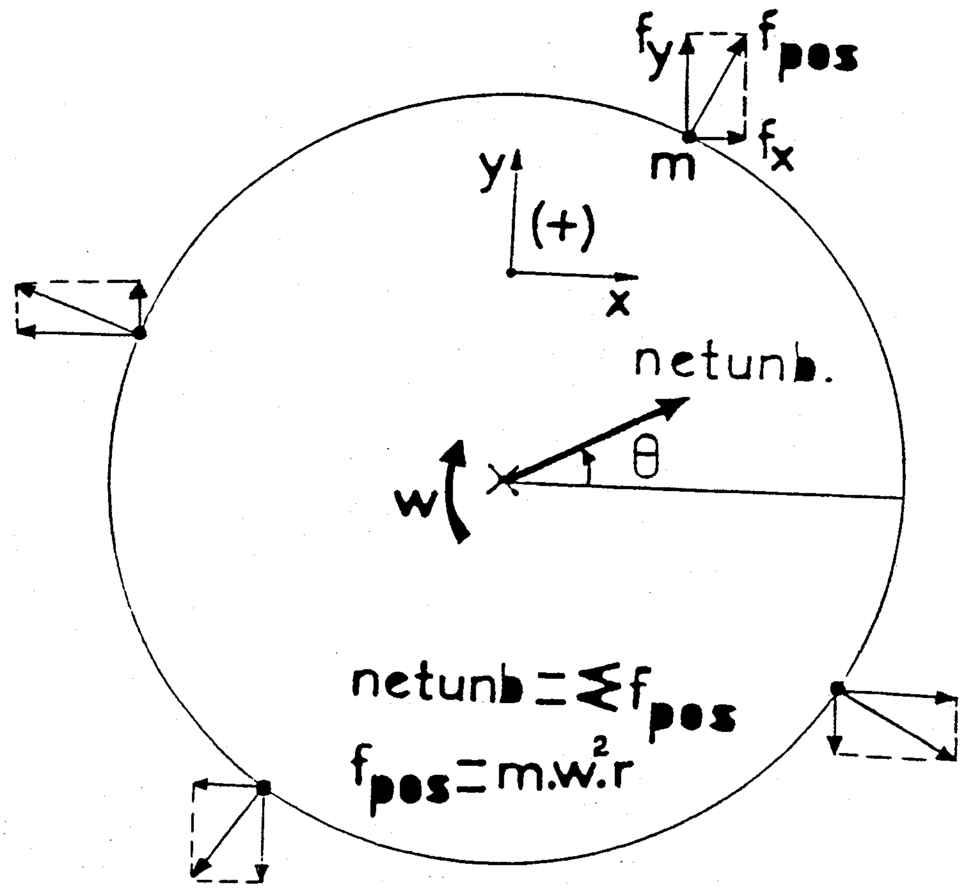

Figure 2.1 Directions and Magnitudes of Centrifugal Unbalanced Forces Acting on Rotor And Its Blades

III. COMPUTATIONAL APPROACH

A. THE FLOW OF PROGRAM RBB

Rotor Blade Balancing Program, RBB (Appendix A), is coded in FORTRAN 77 language and all calculations have been done in double precision. Program was compiled on an IBM 3033 computer by using both the FORTVS compiler and WATFOR77 compiler. The same program was also compiled on an IBM PC/XT and stored on a 5 1/4 inch flexible diskette. The entire algoritm process is shown in Figure 3.1.

The program assumes the turbomachine to be a gas turbine engine which has three subassemblies, a high pressure compressor, a high pressure turbine, and a low pressure turbine. Each blade row data is stored in a data file. To begin the program, the user first selects one of the above three subassemblies, and then a stage within it. At this point, the program calls the related data file by using an "open statement" and the execution process starts. The blades are numbered counter clockwise and their position angles are measured from a reference radius of the rotor (Figure 3.2). The program arranges the blades all around the rotor by using a heuristic approach. In this particular way, blades are

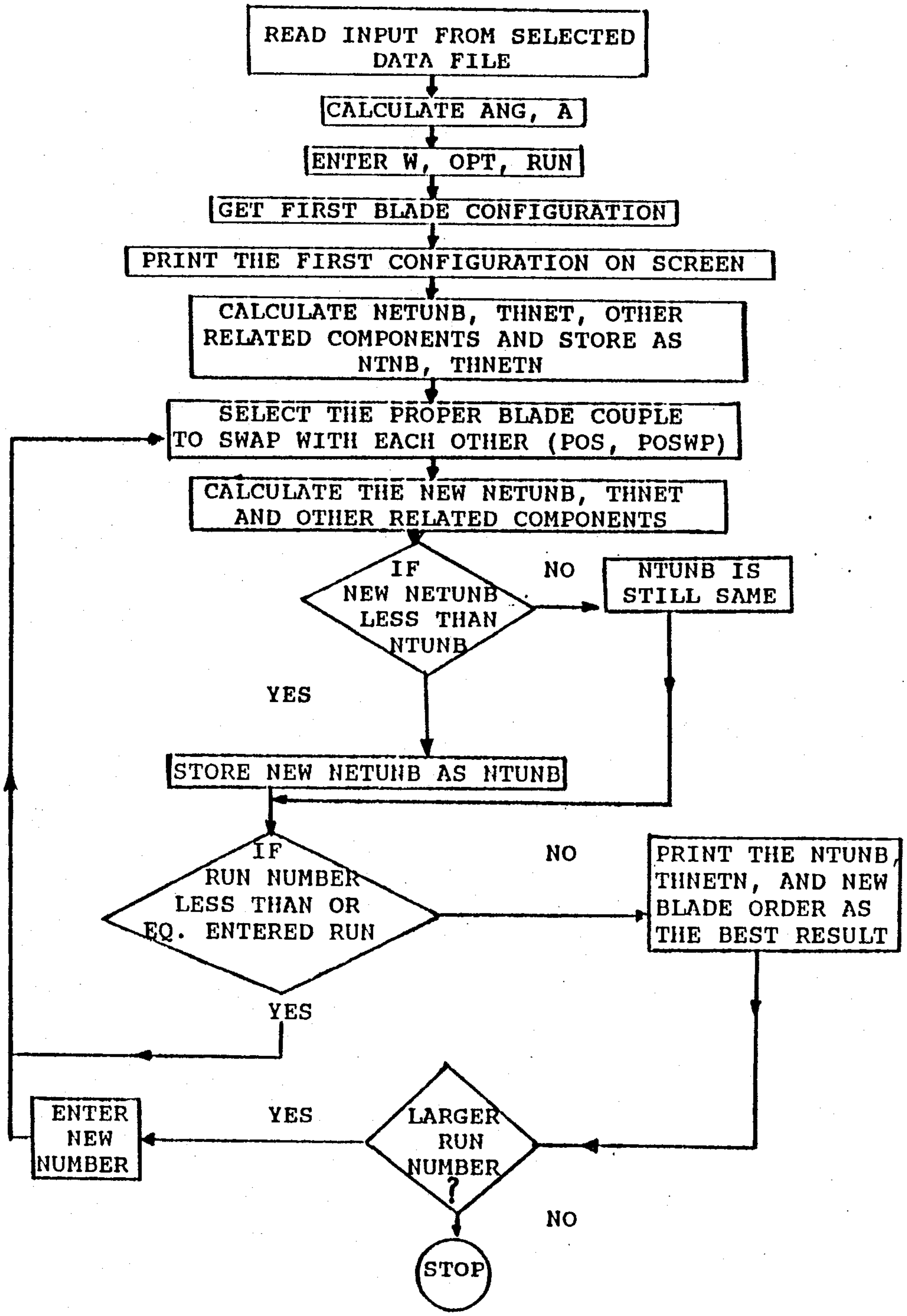

Figure 3.1 Flow Chart For RBB Computer Code

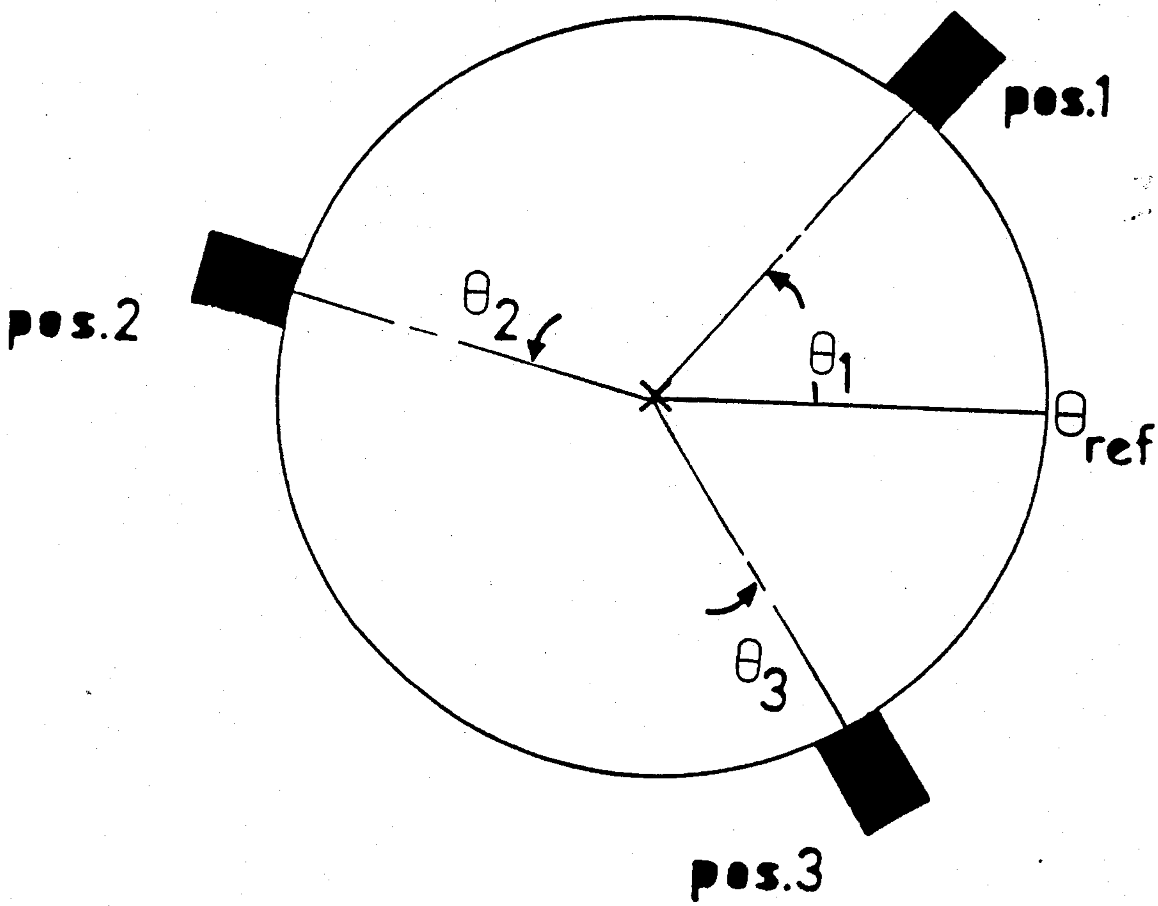

Figure 3.2 The Positions Of The Sample Blades On The Rotor

set up as the heaviest one to the position number 1, the second heaviest one opposite to it and continues third heaviest one to the position number 2, the fourth heaviest one opposite to it, until the end of the blades (Figure 3.3). After this first blade configuration, the position numbers and mass values of the blades are written to the computer screen by using the Subroutine TABLE.

As a further step, the user is asked to input rotor angular velocity, w, and option number, Opt, which will choose the best blade couple to swap with each other. Entering the rotor angular velocity as unity, makes the calculation simple and systematic. Option number is the number of blades searched from the location of the unbalanced force vector to obtain the heavier (or lighter) than average blade mass for swapping. Also, Subroutine SWAP1 and SWAP2 explain more about choosing the best blade couple to swap. The other question for the user is to input the desired number of iterations (Run) to execute the program.

After the blade swap, the program calculates the total centrifugal unbalanced force magnitude, and direction of this new configuration. This calculation is explained more in Subroutine UNBLNC.

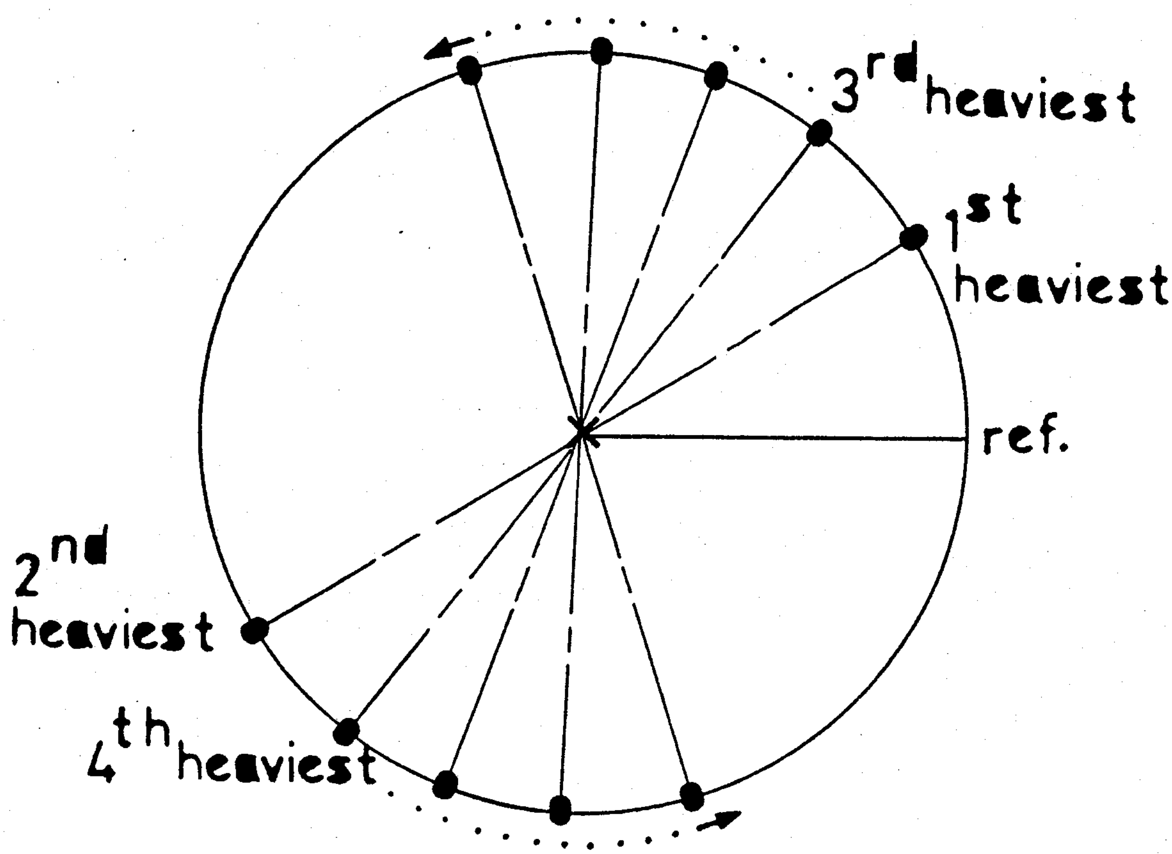

Figure 3.3 The Initial Set-Up Configuration

After each run, if the calculated total centrifugal unbalanced force magnitude, Netunb, is less than a previously calculated one, the program continues to find a proper blade couple to swap with each other in the given desired run number range.

The Subroutine POSTN, SWAP1, POSSWP, SWAP2, and Alternate swap case are used for proper blade couple selection.

In general, two different methods to stop the algorithm could be considered:

(a) Based on a minimum desired unbalanced force magnitude; or

(b) Based on a given number of trials.

The second method was chosen for the program. The program maintains in memory the configuration with the least unbalance force magnitude, Ntunb, and its direction, Thetan. The program also reports the trial number at which this related blade configuration was achieved by using Subroutine RESULT.

B. CONSTRUCTION OF INPUT DATA FILES

The desired data file is called by the program by using an OPEN statement. After the user is asked which component of turbomachinery is selected, the stage of related component's rotor also is asked. In the program, the high pressure compressor has 16 stages, the high pressure turbine has 2, and the low pressure turbine has 6 stages corresponding to the LM2500 Gas Turbine Engine. Although the computer program allows different blade center of gravity locations (accompanying the different blade masses), the examples selected in this thesis assumed the center of gravities of all blades in a given blade row to be at the same radius.

The first column of each data file presents mass values of each blade and the second column presents an average radius from the assumed average blade's center of gravity to the rotor axis. The units are grams and inches, respectively. A sample input data file is shown in Table 3.1 (High Pressure Compressor, Stage Number 2).

Table 3.1 Sample Input Data File For Program RBB

DATA FILE

HIGH PRESS. COMPR. STAGE NO : 2 BLADE NO : 26

MASSES	RADII
191.50	9.54500
191.27	9.54500
191.02	9.54500
190.90	9.54500
190.57	9.54500
190.34	9.54500
190.11	9.54500
189.87	9.54500
189.66	9.54500
189.42	9.54500
189.19	9.54500
188.96	9.54500
188.73	9.54500
188.27	9.54500
188.04	9.54500
187.81	9.54500
187.59	9.54500
187.34	9.54500
187.12	9.54500
186.91	9.54500
186.66	9.54500

In the place of actual blade data, a typical average mass and its range of variation for each blade row was assumed. A linear distribution of the variation was used to obtain individual blade masses. For example, the following was assumed for the second stage of the high pressure compressor:

(a) The average mass, Mavg, is 188.50 grams and the approximate range of variation, mv, is ± 3.0 grams. The second stage has 26 blades (N=26).

(b) The individual blade masses can be established by taking the heaviest blade to be Mavg+mv and the lightest blade to be Mavg-mv. In this example, they are 188.50 + 3.0 and 188.50 - 3.0, respectively.

(c) After the calculation of the mass of the heaviest blade, the masses of the blades are decreased by increments of 2mv/N from the heaviest to the lightest blade. The numeric value of this increment is approximately 0.23 in this example.

C. LOGIC OF PROGRAM ALGORITHM BY DESCRIPTION OF SUBROUTINES

Program RBB uses a heuristic approach at the beginning of its algorithm as explained before. Setting the blades as the heaviest and second heaviest opposite to it, third heaviest, and the fourth heaviest opposite to it, and so on, is considered as a sensible way for the first blade configuration (Figure 3.3).

The logic of the algorithm is to find the minimum magnitude of the total centrifugal unbalanced force magnitude. Therefore, the calculated force magnitude of each run is always compared to a previously calculated minimum unbalance and the minimum one is stored until another minimum is obtained.

Blade swapping is the main part of the program. The criterion used to select a blade for swapping is based on the assumption that the unbalanced force is directed toward the heavy side of the rotor. A heavy blade, defined as one whose mass is greater than or equal to the average mass of the blades, which is nearest to the unbalanced force vector is the one selected. This blade is swapped with a blade whose mass is less than the average at a location nearest to the direction opposite to the direction of the unbalanced force vector. Subroutine SWAP1 and SWAP2 explain more in detail about this comparison and selection of the best couple. Besides using Subroutine SWAP1 and SWAP2, Alternate swap case is also used. During the run, it was observed that particular blade swappings and the resultant unbalance were repeated. This was called "Program Lock-Up". Alternate Swap Case is used to break out of this situation.

1. Subroutine TABLE

The purpose of this subroutine is to list the blade mass values and angular positions from the reference radius. The blades are numbered counter clockwise starting from the reference radius by the angle occupied by each blade. The main program uses this subroutine after the first blade set-up configuration.

2. Subroutine UNBLNC

Subroutine UNBLNC calculates the centrifugal force magnitude F, and its X and Y components, Fx and Fy respectively, for each individual blade (Eq 3.1 and 3.2). Finally, the total centrifugal unbalanced force magnitude, Netunb, which is acting at the center of gravity of the rotor, and its components are calculated (Eq 3.3, 3.4, and 3.5). Furthermore, the direction of Netunb, named as Theta, is determined by taking the ratio of the Y component of Netunb, which is called Totfy, to its X component, Totfx (Eq. 3.6). Also, Figure 2.1 shows the general view of magnitudes and directions of the forces.

$$Fx = F.CosA \qquad \text{(Eq. 3.1)}$$

$$Fy = F.SinA \qquad \text{(Eq. 3.2)}$$

$$Totfx = \sum_{i=1}^{N} Fx \qquad \text{(Eq. 3.3)}$$

$$Totfy = \sum_{i=1}^{N} Fy \qquad \text{(Eq. 3.4)}$$

$$Netunb = \sqrt{(Totfx)^2 + (Totfy)^2} \qquad \text{(Eq. 3.5)}$$

$$Thr = \arctan\ (Totfy/Totfx) \qquad \text{(Eq. 3.6)}$$

It is necessary to identify the quadrant of the direction of the total centrifugal unbalanced force, Thr, on the trigonometric circle. The following trigonometric relationship is used in this subroutine to find the direction angle:

$$\tan\ (Thr) = \tan\ (Thr + 180^\circ) \qquad \text{(Eq. 3.7)}$$

The minimum unbalanced force magnitude is stored as Ntunb in the Subroutine UNBLNC. Also store in this subroutine is the direction of the unbalanced force and the blade sequence.

3. Subroutine RESULT

Subroutine RESULT writes the following results of the best blade configuration on the computer screen: The recorded minimum total centrifugal unbalanced force magnitude, Ntunb, its X and Y components, Totfxn, Totfyn respectively, and the direction angles, Thnet and Thetan, in radians and in degrees respectively. In addition, the position numbers of blades (counter clockwise), mass-radius values, Pn and Radn respectively, and the individual centrifugal force components, Fxn and Fyn, respectively, are included.

4. Subroutine POSTN

Subroutine POSTN is designed to find the position number, Pos, and identify the mass value, P(Pos), of the blade which is the closest one to the total centrifugal unbalanced force vector. In other words, the angle of the chosen blade, A, is the closest angle to the direction angle of the total centrifugal unbalanced force vector, Theta. Subroutine POSTN uses a sector searching technique to find the position number of this particular blade. In this technique, Thh and Thl, low one and high one, respectively, are started from the reference radius and are increased by the value of the occupied angle of each blade, Ang. To illustrate this technique, a simple example is analyzed as follows:

Assumption 1: Rotor has 10 blades and occupied angle by each blade is 36 degrees ($N = 10$, $ANG = 36^\circ$)

Assumption 2: The total centrifugal unbalanced force direction is 42 degrees ($Theta = 42^\circ$)

$$Ang = \frac{360}{N} \qquad \text{(Eq. 3.8)}$$

The angle boundaries are constructed as 0 and 36 degrees, low one and high one, respectively. By observing that, if Theta is not covered by these boundaries, the new low and high limits are chosen as 36 and 72 (36 + Ang) degrees, where the increment is the value of Ang which is calculated in equation 3.8 above. This particular example is also shown schematically in Figure 3.4. The mean value of the angle between the high and low limits, Thm, is then calculated.

$$Thm = 1/2\ (Thh + Thl) \qquad \text{(Eq. 3.9)}$$

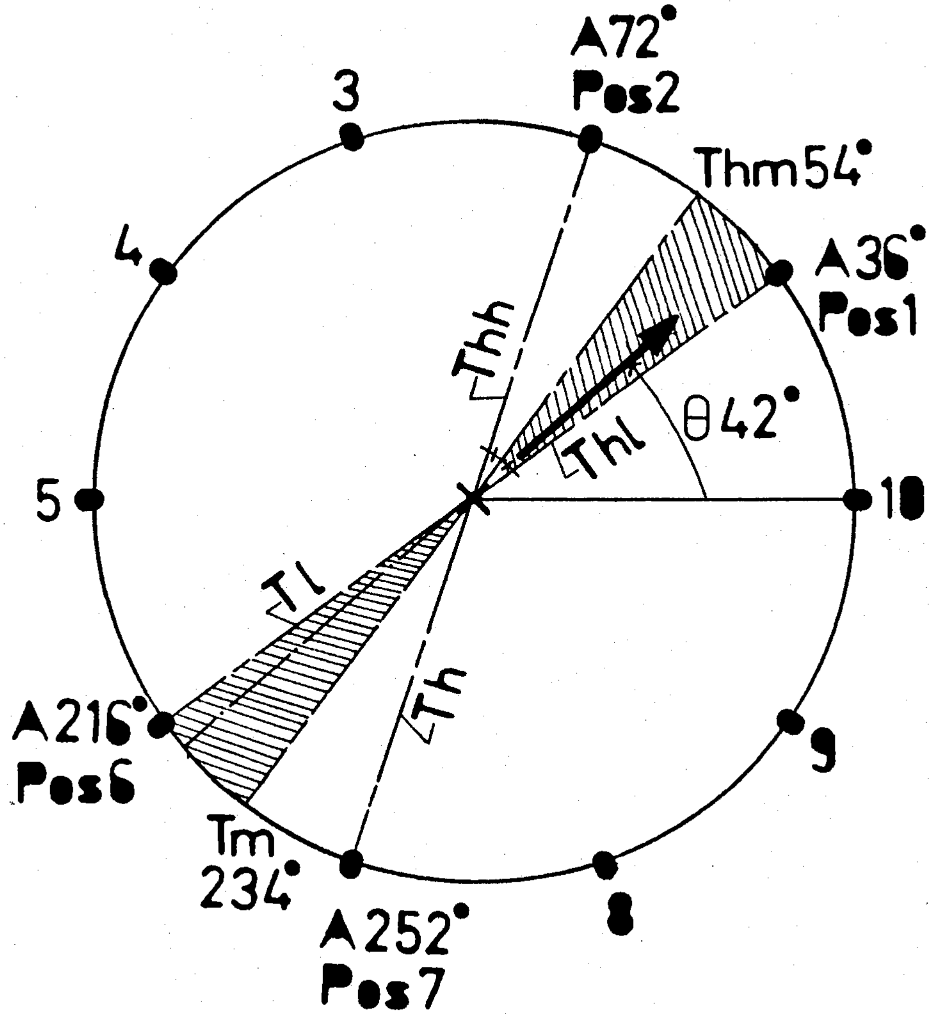

Figure 3.4 Example For Determining The Closest Blade Number To The Total Unbalanced Force

The purpose of the mean angle to identify where the total centrifugal unbalanced force direction, Theta, is. If theta is greater than Thm, the high boundary angle, Thh, is declared as a chosen blade angle, Thbld. Thbld represents the blade angle which is the closest one to the direction angle of the total centrifugal unbalanced force, Theta. The position number, Pos, is calculated by the ratio of chosen blade angle, Thbld, to the occupied angle of each blade, Ang. If Theta is less than or equal to Thm, the low boundary angle, Thl, is declared as Thbld.

$$\text{Pos} = \text{Thbld}/\text{Ang} \qquad \text{(Eq. 3.10)}$$

In the previous given example, it can be observed that Thl and Thh are 36 and 72 degrees, respectively. The blades are numbered 1, 2, 3, through 10 counter clockwise having 36, 72, 108, through 360 degree blade angles, respectively. The mean angle, Thm, is 54 degrees. Recalling that Theta is 42 degrees, it is observed that Thl is the declared chosen blade angle, Thbld (Figure 3.4). Therefore, the blade position number, Pos, is 1, the closest one to the total centrifugal unbalanced force.

5. Subroutine SWAP1

The determined blade which has the closest position number to the total centrifugal unbalanced force is compared with the average mass value, Mavg, in this subroutine and the program decides whether to swap it or not. If the mass value of this blade is greater than or equal to the average mass, Mavg, the program decides to swap this heavy blade with a light one which is on the other side of the rotor. It is always possible that when searching for a heavy blade to swap, that the blade nearest the unbalanced force vector is lighter than the average. The algorithm then checks the neighboring blades in the order closest to the unbalanced force vector. The number of neighboring blades checked is determined by the user in the specification of the option number, Opt.

Using the same example which is given in the explanation of the Subroutine POSTN, the following is concluded: If the chosen blade (position number 1) is less than the average mass, blade number 2 will be checked and then after that number 10, number 3, and number 9. This back and forth method will be repeated in the program until finding a proper blade to swap in the given option number range. If the blade still cannot be found which is heavier than or equal to average mass, the program chooses the heaviest one of all checked blades to swap.

6. Subroutine POSSWP

Subroutine POSSWP works as Subroutine POSTN does but on the opposite side of the rotor to locate the light blade to swap with the selected heavy blade. This subroutine is constructed to find the position number, Poswp, and identify the mass value, P(Poswp), of the blade. Tswap is the direction directly opposite of the total centrifugal unbalanced force direction angle, Theta.

$$Tswap = Theta + 180^{\circ} \qquad \text{(Eq. 3.11)}$$

As applied in Subroutine POSTN, it is used in a sector searching technique to find the position number, Poswp, in this subroutine. The low and high angle boundaries, Tl and Th, respectively are again started from the reference radius and increased by the occupied angle of each blade, Ang. Recalling the same example as above :

- The opposite angle to the total centrifugal unbalanced force direction angle, Tswap, is 222 degrees ($42^{\circ} + 180^{\circ}$).
- The angle boundaries are 216° and 252° as low and high, respectively. These results are shown in Figure 3.4.

The mean value, Tm, is given by:

$$Tm = 1/2\ (Tm + Tl) \qquad \text{(Eq. 3.12)}$$

After finding the mean angle, program identifies where opposite angle, Twsap, is. If Tswap is greater than or equal to Tm, the high boundary angle, Th, is declared as a opposite chosen blade angle, Tswbd. The position number of this blade, Poswp, is calculated by the ratio of Tswbd to the occupied angle of each blade, Ang.

$$Poswp = Tswbd/Ang \qquad \text{(Eq. 3.13)}$$

If Tswap is less than Tm, low boundary angle, Tl, is declared as Tswbd. In numerical given example, the mean angle is calculated as 234 degrees. Recalling that Tswap is 222 degrees, it is observed that Tl is the declared chosen blade angle, Tswbd, to swap. Therefore, this blade number, Poswp, is 6 and is swapped with blade number 1 which was already chosen in Subroutine POSTN.

7. Subroutine SWAP2

The blade number Poswp which is the opposite to the already chosen blade number Pos, in Subroutine SWAP1 has been compared with the average mass. If the mass value of this blade is less than or equal to the average mass, Mavg, the program determines to swap this light blade with the chosen blade in Subroutine SWAP1. As a result, the heavier blade which is the closet one to the total centrifugal unbalanced force and the lighter blade which is opposite to the heavier one are swapped for shifting the force magnitude from its unbalanced position.

If mass value in the blade number Poswp, which is the nominee one to swap, is greater than Mavg, the Subroutine tries to find another neighbor blade which is lighter than or equal to the average mass. The angle boundaries, Th and Tl, are used with respect to option number range for this searching.

Numerical example can be concluded as follows: Poswp is 6 and if it is heavier than Mavg, blade number 7 will be checked and then after that number 5, number 8, and number 4 (Figure 3.4). This back and forth method, as in Subroutine SWAP1, will be repeated until finding the proper blade in the given option number range. If the searching fails, the lightest one between all checked blades is chosen to swap.

8. Alternate Swap Case

If, during the iteration process, the selection of blades to swap are repeating a previous swap, the situation is called "program lock-up". To break out of this, the program arbitrarily begins the search at an angle 90 degrees beyond Theta, and then continues in the normal fashion.

D. CONSTRAINTS OF PROGRAM

The initial constraint of the program is the heuristic way to set the blades up as a first configuration. However, this order of the first blade configuration is assumed as a good starting point.

Another constraint appears is in Alternate Swap Case. The program checks the repeated blades for swapping after the first 20 runs. While running the program, it was observed that there was no repeated blade couple to swap in the early run numbers (approximately before 20). But, it is not an exact rule, just the common output of the program. If there is a repeated blade couple in the early runs, i.e., before 20 iterations, the Alternate Swap Section is not applied.

A third constraint is imposed by assuming all blade center of gravities are located at the same radial distance. But in fact, the center of gravities of the blades will be different from each other because of their different mass values. Although this difference is not very significant, it is a difference. Data files assume that these very small differences are negligible. The computer program is written, however, to include a different radius for each blade.

IV. RESULTS AND COMPARISON BETWEEN PROGRAM RBB AND BLADEBAL.DEP BY GENERAL ELECTRIC COMPANY

A. THE INTRODUCTION OF PROGRAM BLADEBAL.DEP

The GE BLADEBAL. DEP program may be used for two cases involving blade arrangement. First, it can be used to replace a row of (old) blades with new blades and to arrange the blades to preserve the old blade configuration unbalance in both magnitude and direction. Secondly, it can be used to insert new blades in an initial build-up to obtain a specified minimum unbalance.

B. LOGIC OF THE PROGRAM BLADEBAL.DEP AND COMPARISON OF THE PROGRAMS

The RBB program output is given in Appendix B. The main logic of program RBB is to compare the blade massses with the average mass value of the blades, Mavg, and to decide the blade couple to be swapped. The Bladebal.Dep program logic sets the initial configuration with the heaviest, then the lightest, then next heaviest, next lightest, etc., around the rotor. The calculated unbalanced force magnitude is checked with the old value or zero for an initial build-up. The check limit is 0.2 grams. If magnitude is greater than required value, swapping begins. The swapping procedure is an follows:

At the position of unbalanced vector, the program selects the heavy blade and swaps it with the light blade directly opposite. If a light blade already exists at unbalanced vector, advance one blade to a heavy one and computation continues. If this makes the rotor even worse, the program replaces the swapped pair in original positions and tries the neighbor blades. After that unbalanced force is again calculated and checked with the previous one until the required limit is reached.

Another feature of General Electric includes the possibility of fixed blades, therefore, if a fixed blade is selected to be swapped, it is left in position and a neighbor blade is tried. While executing the program, it is possible to be locked-in without changing the force magnitude. Randomly selected blades are then swapped to break-out of this situation.

After having the desired magnitude, all blades, except those fixed, are rotated to get the desired direction of the unbalanced force. This also can change the already calculated magnitude. By following the swap procedure, the program again works to get the desired magnitude and angle values.

A comparison of the two programs is illustrated by using them to obtain a minimum unbalance configuration for a particular set of blades. The second stage of the high pressure compressor comprising 26 blades was used.

Unit radius and unit rotational speed were used in the RBB program to compare results with the General Electric Bladebal.Dep program. The results of this comparison are shown in Table 4.1. As can be seen from these results, different configurations meeting the desired unbalance are obtained.

The RBB program required 1748 iterations to meet the minimum unbalance criterion.

Table 4.1. Comparison of Results

PROGRAM		RBB	BLADEBAL.DEP
Unbalanced Force (Based on r=1, w=1)		0.06315 GM	0.09 GM
Direction		284.2615 DEG.	101.53 DEG.
Configuration	1	187.81	190.90
	2	190.34	189.66
	3	189.87	185.74
	4	188.04	187.81
	5	186.43	185.97
	6	188.27	189.19
	7	190.90	186.10
	8	191.27	191.27
	9	186.10	190.34
	10	187.59	186.91
	11	191.02	190.11
	12	185.50	187.12
	13	189.66	187.34
	14	187.12	190.57
	15	185.97	191.50
	16	190.57	186.43

Table 4.1. Comparison of Results (Continued)

17	190.11	189.42
18	188.73	185.50
19	188.96	188.27
20	187.34	188.73
21	186.91	186.66
22	191.50	188.04
23	189.42	191.02
24	189.19	188.96
25	185.74	187.59
26	186.66	189.87

V. CONCLUSION AND RECOMMENDATIONS

The initial blade set-up configuration can be selected by any programmer. Also, the selected criterion for blades to be swapped, i.e., average mass, is optional. The two approaches discussed previously are but two of the many heuristic methods for the case of replacing a set of old blades with a specific unbalance magnitude and direction.

To determine the absolute minimum imbalance would involve trying every possibility one by one. The number of ways a certain number of blades can be arranged is the factorial of this certain number. That means for the 166 blades of the low pressure turbine first stage must be tried in 166 factorial configurations to find the real unique solution. But the time required to iterate that large number in the computer is not realistic. Therefore, the heuristic methods are chosen.

During the initial phases of this thesis, optimization techniques were investigated. One particular method was linear integer programming. However, no clear solution appeared evident, and hence the heuristic methods were employed. The technique presented here which is used to balance a single row of blades must be extended to multiple blade rows. In this latter case, the individual unbalanced forces must be combined to determine the dynamic unbalance of the assembled rotor.

The existing program, RBB, should be modified to include the user option of specifying a minimum acceptable unbalance magnitude and direction.

The existing program, RBB, must also be modified to include fixed blades or twin blades.

APPENDIX A

RBB PROGRAM LISTING

```
*     " RBB.FORTRAN "
*     ROTOR BLADE BALANCING PROGRAM - ALI R. YARDIMOGLU


*     VARIABLE  DECLERATIONS

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8  M(166),A(166),RA(166),R(166),MAVG,F(166),
     +FX(166),FY(166),NETUNB,P(166),RAD(166),UNB(22),
     +NTUNB,PN(166),FXN(166),FYN(166),RADN(166),PI,W
      INTEGER C,RUN,S,OPT,POS,POSA,POSB,POSWP,POSN,POSWPN,N
      NTUNB=100000000000.0D0



10    WRITE(*,3)
3     FORMAT(/,' GAS TURBINE ROTOR BALANCING PROGRAM BY BLADE ORDER')
      WRITE(*,23)
23    FORMAT(/,' SELECT AND ENTER :  1 FOR HPC , 2 FOR HPT , 3 FOR LPT')
      READ(*,*) S


* INPUT ERROR CORRECTION IF THERE IS

      IF(S.GT.3) GO TO 20
       IF(S.LT.1) GO TO 20
      WRITE(*,24)  S
24    FORMAT(/,' SELECTION  ',I1,' IS ON PROCESS')


* TO CHOOSE DATA FILE ACCORDING TO SELECTED NUMBER

      IF(S.EQ.1) GO TO 30
       IF(S.EQ.2) GO TO 40
        IF(S.EQ.3) GO TO 50
      GO TO 30
20    WRITE(*,28)
28    FORMAT(/,' SELECTION MUST BE EITHER 1,2 OR 3 ; ENTER AGAIN !...')
      GO TO 10
30    CONTINUE


* DATA FILES FOR HIGH PRESSURE COMPRESSOR

91    WRITE(*,33)
33    FORMAT(/,' WHICH STAGE DO YOU WANT TO CHOOSE IN HPC ?  1 THRU 16')
      READ(*,*) IS
      IF(IS.LT.1) GO TO 101
      IF(IS.GT.16) GO TO 101
102   CONTINUE
```

```
      IF(IS.EQ.1) GO TO 103
      IF(IS.EQ.2) GO TO 104
      IF(IS.EQ.3) GO TO 105
      IF(IS.EQ.4) GO TO 106
      IF(IS.EQ.5) GO TO 107
      IF(IS.EQ.6) GO TO 108
      IF(IS.EQ.7) GO TO 109
      IF(IS.EQ.8) GO TO 701
      IF(IS.EQ.9) GO TO 702
      IF(IS.EQ.10) GO TO 703
      IF(IS.EQ.11) GO TO 704
      IF(IS.EQ.12) GO TO 705
      IF(IS.EQ.13) GO TO 706
      IF(IS.EQ.14) GO TO 707
      IF(IS.EQ.15) GO TO 708
      IF(IS.EQ.16) GO TO 709
      GO TO 103

101   WRITE(*,509)
509   FORMAT(/,' STAGE NUMBER  MUST BE BETWEEN 1 AND 16  FOR HPC ')
      GO TO 91

103   OPEN(UNIT=15,FILE='HPCS1.DAT',STATUS='OLD')
         DO 60 I =1,36
         READ(15,*) M(I),R(I)
60       CONTINUE
      N = 36
      GO TO 90

104   OPEN(UNIT=15,FILE='HPCS2.DAT',STATUS='OLD')
         DO 70 I =1,26
         READ(15,*) M(I),R(I)
70       CONTINUE
      N=26
      GO TO 90

105   OPEN(UNIT=15,FILE='HPCS3.DAT',STATUS='OLD')
         DO 80 I =1,42
         READ(15,*) M(I),R(I)
80       CONTINUE
      N=42
      GO TO 90

106   OPEN(UNIT=15,FILE='HPCS4.DAT',STATUS='OLD')
         DO 73 I =1,45
         READ(15,*) M(I),R(I)
73       CONTINUE
      N=45
      GO TO 90

107   OPEN(UNIT=15,FILE='HPCS5.DAT',STATUS='OLD')
         DO 74 I =1,48
         READ(15,*) M(I),R(I)
74       CONTINUE
      N=48
      GO TO 90

108   OPEN(UNIT=15,FILE='HPCS6.DAT',STATUS='OLD')
         DO 76 I =1,54
```

```
        READ(15,*) M(I),R(I)
76      CONTINUE
     N=54
     GO TO 90

109  OPEN(UNIT=15,FILE='HPCS7.DAT',STATUS='OLD')
        DO 77 I =1,56
        READ(15,*) M(I),R(I)
77      CONTINUE
     N=56
     GO TO 90

701  OPEN(UNIT=15,FILE='HPCS8.DAT',STATUS='OLD')
        DO 78 I =1,64
        READ(15,*) M(I),R(I)
78      CONTINUE
     N=64
     GO TO 90

702  OPEN(UNIT=15,FILE='HPCS9.DAT',STATUS='OLD')
        DO 79 I =1,66
        READ(15,*) M(I),R(I)
79      CONTINUE
     N=66
     GO TO 90

703  OPEN(UNIT=15,FILE='HPCS10.DAT',STATUS='OLD')
        DO 81 I =1,66
        READ(15,*) M(I),R(I)
81      CONTINUE
     N=66
     GO TO 90

704  OPEN(UNIT=15,FILE='HPCS11.DAT',STATUS='OLD')
        DO 82 I =1,76
        READ(15,*) M(I),R(I)
82      CONTINUE
     N=76
     GO TO 90

705  OPEN(UNIT=15,FILE='HPCS12.DAT',STATUS='OLD')
        DO 83 I =1,76
        READ(15,*) M(I),R(I)
83      CONTINUE
     N=76
     GO TO 90

706  OPEN(UNIT=15,FILE='HPCS13.DAT',STATUS='OLD')
        DO 84 I =1,76
        READ(15,*) M(I),R(I)
84      CONTINUE
     N=76
     GO TO 90

707  OPEN(UNIT=15,FILE='HPCS14.DAT',STATUS='OLD')
        DO 6 I =1,76
        READ(15,*) M(I),R(I)
6       CONTINUE
     N=76
     GO TO 90
```

```
708     OPEN(UNIT=15,FILE='HPCS15.DAT',STATUS='OLD')
           DO 17 I =1,76
           READ(15,*) M(I),R(I)
17         CONTINUE
        N=76
        GO TO 90

709     OPEN(UNIT=15,FILE='HPCS16.DAT',STATUS='OLD')
           DO 88 I =1,76
           READ(15,*) M(I),R(I)
88         CONTINUE
        N=76

90      WRITE(*,219)
219     FORMAT(/,' DATA FILE')
        WRITE(*,205) IS,N
205     FORMAT(/,' HIGH PRESS. COMPR. STAGE NO :',I2,' BLADE NO :',I3)
        GO TO 689
40      CONTINUE


*       DATA FILES FOR HIGH PRESSURE TURBINE

527     WRITE(*,37)
37      FORMAT(/,' WHICH STAGE DO YOU WANT TO  CHOOSE IN HPT ?  1 THRU 2')
        READ(*,*) IS
        IF(IS.LT.1) GO TO 528
        IF(IS.GT.2) GO TO 528
529     CONTINUE
        IF(IS.EQ.1) GO TO 532
        IF(IS.EQ.2) GO TO 533
        GO TO 532
528     WRITE(*,549)
549     FORMAT(/,' STAGE NUMBER MUST BE BETWEEN 1 AND 2 FOR HPT')
        GO TO 527

532     OPEN(UNIT=15,FILE='HPTS1.DAT',STATUS='OLD')
           DO 534 I =1,88
           READ(15,*) M(I),R(I)
534        CONTINUE
        N=88
        GO TO 900

533     OPEN(UNIT=15,FILE='HPTS2.DAT',STATUS='OLD')
           DO 535 I =1,90
           READ(15,*) M(I),R(I)
535        CONTINUE
        N=90

900     WRITE(*,657) IS,N
657     FORMAT(/,' HIGH PRESS. TURBINE STAGE NO :',I2,2X,' BLADE NO :',I3)
        GO TO 689
50      CONTINUE


*       DATA FILES FOR LOW PRESSURE TURBINE

771     WRITE(*,753)
```

```
753   FORMAT(/,' WHICH STAGE DO YOU WANT TO CHOOSE IN LPT ? 1 THRU 6')
      READ(*,*) IS
      IF(IS.LT.1) GO TO 754
      IF(IS.GT.6) GO TO 754
755   CONTINUE

      IF(IS.EQ.1) GO TO 756
      IF(IS.EQ.2) GO TO 757
      IF(IS.EQ.3) GO TO 758
      IF(IS.EQ.4) GO TO 759
      IF(IS.EQ.5) GO TO 760
      IF(IS.EQ.6) GO TO 761
      GO TO 756

754   WRITE(*,770)
770   FORMAT(/,' STAGE NUMBER MUST BE BETWEEN 1 AND 6 FOR LPT')
      GO TO 771

756   OPEN(UNIT=15,FILE='LPTS1.DAT',STATUS='OLD')
         DO 781 I =1,166
         READ(15,*) M(I),R(I)
781      CONTINUE
      N=166
      GO TO 147

757   OPEN(UNIT=15,FILE='LPTS2.DAT',STATUS='OLD')
         DO 782 I =1,142
         READ(15,*) M(I),R(I)
782      CONTINUE
      N=142
      GO TO 147

758   OPEN(UNIT=15,FILE='LPTS3.DAT',STATUS='OLD')
         DO 783 I =1,126
         READ(15,*) M(I),R(I)
783      CONTINUE
      N=126
      GO TO 147

759   OPEN(UNIT=15,FILE='LPTS4.DAT',STATUS='OLD')
         DO 784 I =1,112
         READ(15,*) M(I),R(I)
784      CONTINUE
      N=112
      GO TO 147

760   OPEN(UNIT=15,FILE='LPTS5.DAT',STATUS='OLD')
         DO 786 I =1,90
         READ(15,*) M(I),R(I)
786      CONTINUE
      N=90
      GO TO 147

761   OPEN(UNIT=15,FILE='LPTS6.DAT',STATUS='OLD')
         DO 787 I =1,72
         READ(15,*) M(I),R(I)
787      CONTINUE
      N=72

147   WRITE(*,811) IS,N
```

```
811   FORMAT(/,' LOW PRESS. TURBINE STAGE NO :',I2,' BLADE NO :',I3)
689   CONTINUE


* OCCUPIED ANGLE BY EACH BLADE

      ANG=360.0/(N)


* TO DEFINE PI NUMBER AND TO EXPRESS IN RADIAN

      PI=4.*DATAN(1.D0)
      RANG=(ANG*PI)/180.

140   WRITE(*,150)
150   FORMAT(/,6X,'MASSES',7X,'RADII')


* CALCULATION OF AVERAGE MASS AND TOTAL MASS

      TOTM=0
      J=1
      DO 160 I = 1,N
          TOTM=M(I)+TOTM
           MAVG=TOTM/N
            A(I) = ANG * J
             RA(I) = (A(I) * PI)/180.
            WRITE(*,170) M(I),R(I)
170        FORMAT(/,3X,F9.2,5X,F8.5)
          IF (J.GT.N ) GO TO 165
      J=J+1
160   CONTINUE
165   WRITE(*,180) MAVG,TOTM
180   FORMAT(///,13X,'AVERAGE MASS,MAVG = ',F9.2,4X,'MASS TOTAL,TOTM =
     +',F9.2,/)


* ROTOR SPEED INPUT

      WRITE(*,182)
182   FORMAT(/,' ENTER THE ANGULAR ROTOR SPEED ( RADIAN/SECONDS )')
      READ(*,*) W
      WRITE(*,184) W
184   FORMAT(/,' W = ',F8.3,2X,' RAD/SEC')


* THE FIRST BLADE SET UP CONFIGURATION ( HEAVIEST-LIGHTEST AND SO ON )

44    CONTINUE
      L=1
      C=1
190   CONTINUE
      DO 200 I = C,N,N/2
          P(I)=M(L)
           RAD(I)=R(L)
```

```
         L=L+1
200    CONTINUE
       C=C+1
       IF(C.GT.N/2) GO TO 210
       GO TO 190
210    CONTINUE
       CALL TABLE (I,N,A,P,PN)


* DESIRED ITERATION NUMBER WHICH IS ENTERED BY USER

       WRITE(*,224)
224    FORMAT(/, ' ENTER THE OPTION NUMBER WHICH WILL CHOOSE THE BEST MAS
      1S',/,' TO SWAP. ( MINIMUM THE HALF OF BLADE NUMBER )')
       READ(*,*) OPT
       WRITE(*,268) OPT
268    FORMAT(/,' OPTION IS',I3,' FOR THIS STAGE ' )
274    WRITE(*,278)
278    FORMAT(/,' HOW MANY TIMES DO YOU WANT TO ITERATE THE PROGRAM ? ')
       READ(*,*) RUN
671    DO 220 J = 1,RUN
       JJ=J


*      TO USE SUBROUTINES FOR ALL CALCULATIONS

66     CALL UNBLNC (I,N,TOTFX,TOTFY,F,P,RAD,W,RA,FX,FY,NETUNB,RATIO,THR,
      1             Z,PN,THNET,THETA,JN,JJ,NTUNB,THETAN,THNETN,TOTFXN,
      2             TOTFYN,FXN,FYN,RADN,R,PI)
*
       IF(JJ.GT.20) GO TO 123
       UNB(JJ)=NETUNB
       GO TO 4

123      CONTINUE
         IF(JJ.EQ.21) GO TO 125
         UNB(22)=NETUNB
         DO 124 I=1,21
         UNB(I)=UNB(I+1)
124      CONTINUE
125      CONTINUE
         UNB(21)=NETUNB
       IF(UNB(21).EQ. UNB(19)) GO TO 777
       IF(UNB(21).EQ. UNB(17)) GO TO 777
       IF(UNB(21).EQ. UNB(15)) GO TO 777
       IF(UNB(21).EQ. UNB(13)) GO TO 777
       IF(UNB(21).EQ. UNB(11)) GO TO 777
       IF(UNB(21).EQ. UNB(9)) GO TO 777
       IF(UNB(21).EQ. UNB(7)) GO TO 777
       IF(UNB(21).EQ. UNB(5)) GO TO 777
       IF(UNB(21).EQ. UNB(3)) GO TO 777
       IF(UNB(21).EQ. UNB(1)) GO TO 777
       GO TO 4

777    CALL POSTN (THL,THH,THETA,ANG,THM,THBLD,IX,POS,P,N)
       CALL SWAP1 (P,POS,MAVG,IX,N,OPT,INC,ISN,POSN)
```

```
*     ALTERNATE SWAP SECTION TO AVOID BLADE LOCKING UP

      POS=POS+N/4
      IF(POS.GT.N) GO TO 881
      GO TO 53
881   POS=POS-N
53    CONTINUE
      IF(P(POS).GE.MAVG) GO TO 699
      ISN=1
      DO 700 I = 1,60

           INC=ISN*I
            POS=POS+INC
             ISN=(-1)*ISN
              IF(POS.GT.N) GO TO 882
                IF(POS.EQ.N) GO TO 52
               IF(POS.LE.0) GO TO 32
                GO TO 52
32             POS=POS+N
              GO TO 52
882          POS=POS-N
52          CONTINUE
           IF(P(POS).GE.MAVG) GO TO 699

700   CONTINUE
      GO TO 86
699   CONTINUE

      CALL POSSWP (TL,TH,TSWP,THETA,ANG,TM,TSWBD,IY,POSWP,P,N)
      CALL SWAP2  (P,POSWP,MAVG,IY,PPOS,POS,N,OPT,INC,ISN,POSWPN,RADPO,
     1             RAD,RADN)
      GO TO 66
4     CALL POSTN  (THL,THH,THETA,ANG,THM,THBLD,IX,POS,P,N)
      CALL SWAP1  (P,POS,MAVG,IX,N,OPT,INC,ISN,POSN)
      CALL POSSWP (TL,TH,TSWP,THETA,ANG,TM,TSWBD,IY,POSWP,P,N)
      CALL SWAP2(P,POSWP,MAVG,IY,PPOS,POS,N,OPT,INC,ISN,POSWPN,RADPO,
     1            RAD,RADN)

220   CONTINUE

      CALL RESULT(N,TOTFXN,TOTFYN,NTUNB,THETAN,THNETN,PN,FXN,FYN,RADN)
      WRITE(*,12)  JN
12    FORMAT(/,' RUN NUMBER FOR BEST BLADE CONFIGURATION :',I7)
      WRITE(*,29) RUN
29    FORMAT(/,' TOTAL RUN NUMBER FOR ITERATION PROCESS  :',I7)
      GO TO 87
86    WRITE(*,56)
56    FORMAT(/,3X,'ALT.SWAP IS NOT MUCH EFFICIENT')
      GO TO 878

87    WRITE(*,59)
59    FORMAT(/,' DO YOU WANT TO TRY LARGER ITERATION NUMBER? (1 OR 0)')
      READ(*,*) LIN
      IF(LIN.LT.0) GO TO 868
      IF(LIN.GT.1) GO TO 868
      IF(LIN.EQ.0) GO TO 878
      GO TO 643
868   WRITE(*,633)
633   FORMAT(/,' ENTER 1 OR 0 ONLY')
      GO TO 87
```

```
643   WRITE(*,631)
631   FORMAT(/,' ENTER NEW LARGER ITERATION NUMBER')
      READ(*,*) NLI
      IF(NLI.LT.RUN) GO TO 569
      RUN = NLI+0
      GO TO 671
569   WRITE(*,584) RUN
584   FORMAT(/,' ENTER BIGGER ITERATION NO. THAN OLD ONE','(',I7,')')
      GO TO 643
878   WRITE(*,892)
892   FORMAT(/,' BALANCING PROCEDURE IS OVER FOR THIS STAGE')
      WRITE(*,896)
896   FORMAT(/,' RECOMPILE PROGRAM FOR DIFFERENT STAGE NUMBERS')
      WRITE(*,*) '**********************************************'
      STOP
      END



************************************************************************

      SUBROUTINE TABLE(I,N,A,P,PN)

************************************************************************


*     PURPOSE : TO LIST THE BLADE MASS/POSITION CONFIGURATION

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER I,N
      REAL*8 A(166),P(166),PN(166)

      WRITE(*,294)
294   FORMAT(/,8X,'FIRST BLADE SET UP CONFIGURATION',/)
      WRITE(*,297)
297   FORMAT(/,6X,'POSITION',5X,'ANGLE (DEG.)',5X,'MASS',/)
      DO 20 I=1,N
        WRITE(*,30)I,A(I),P(I)
30      FORMAT(/,9X,I3,7X,F7.3,6X,F9.2)
20    CONTINUE
      RETURN
      END



************************************************************************

      SUBROUTINE UNBLNC(I,N,TOTFX,TOTFY,F,P,RAD,W,RA,FX,FY,NETUNB,RATIO,
     1                 THR,Z,PN,THNET,THETA,JN,JJ,NTUNB,THETAN,THNETN,
     2                 TOTFXN,TOTFYN,FXN,FYN,RADN,R,PI)

************************************************************************
```

```
*      PURPOSE : CALCULATION OF NET UNBALANCED FORCE CHARACTERISTICS

       IMPLICIT REAL*8 (A-H,O-Z)
       INTEGER I,N
       REAL*8  F(166),P(166),RAD(166),RA(166),FX(166),FY(166),NETUNB,
      +PN(166),NTUNB,FXN(166),FYN(166),RADN(166),R(166),W
       TOTFX=0
       TOTFY=0
       PI=4.D0*DATAN(1.D0)

       DO 987 I=1,N

         F(I)=P(I)*RAD(I)*(W**2)
          FX(I)=F(I)*DCOS(RA(I))
           FY(I)=F(I)*DSIN(RA(I))
            TOTFX=FX(I)+TOTFX
            TOTFY=FY(I)+TOTFY

* CALCULATION OF UNBALANCED FORCE MAGNITUDE AND ITS DIRECTION

           NETUNB=SQRT ( TOTFX**2 + TOTFY**2 )
          RATIO=TOTFY/(TOTFX+0.00000000000001D0)
         THR=DATAN(RATIO)

* QUADRANT DETERMINATION FOR UNBALANCED FORCE ANGLE

         IF(RATIO.GE.0) GO TO 30
          IF(TOTFY.GE.0) GO TO 40
          Z=2*PI
         GO TO 60
30       IF(TOTFY.GE.0) GO TO 50
40        Z=1*PI
           GO TO 60
50          Z=0
60         THNET=THR+Z
          THETA=THNET*180.0D0/PI
987     CONTINUE

        IF(NETUNB.LT.NTUNB) GO TO 500
        GO TO 510
500     CONTINUE
        NTUNB=NETUNB
        THETAN=THETA
        THNETN=THNET
        TOTFXN=TOTFX
        TOTFYN=TOTFY
        JN=JJ
        DO 501 K = 1,N
        PN(K)=P(K)
        RADN(K)=RAD(K)
        FXN(K)=FX(K)
        FYN(K)=FY(K)
501     CONTINUE
510     CONTINUE
        RETURN
        END
```

```
*************************************************************************

      SUBROUTINE RESULT(N,TOTFXN,TOTFYN,NTUNB,THETAN,THNETN,PN,FXN,
     1                  FYN,RADN)

*************************************************************************

*     PURPOSE : TO WRITE THE RESULTS OF CALCULATIONS OF THE BEST RUN

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 TOTFXN,TOTFYN,NTUNB,THETAN,THNETN,FXN(166),FYN(166),P(166),
     +PN(166),RADN(166),RAD(166)
      INTEGER N
      WRITE(*,17)
17    FORMAT(/,'  POS.',4X,'MASS',8X,'RADIUS',17X,'X-COMP',11X,'Y-COMP')
      DO 35 I=1,N

      WRITE(*,99)  I,PN(I),RADN(I),FXN(I),FYN(I)
99    FORMAT(/,2X,I3,2X,F9.2,5X,F8.5,10X,F14.4,3X,F14.4)

35    CONTINUE

      WRITE(*,90)
90    FORMAT(//,13X,'SUM OF X COMPONENT',7X,'SUM OF Y COMPONENT')
      WRITE(*,*) '            --------------------------------------'
      WRITE(*,100) TOTFXN,TOTFYN
100   FORMAT(7X,F18.8,7X,F18.8)
      WRITE(*,110)
110   FORMAT(//,3X,'NET UNBALANCED VECTOR',7X,'DIRECTION OF VECTOR',8X,'
     +DIRECTION OF VECTOR',/,8X,'MAGNITUDE',17X,'THETA (RADIAN)',13X,'TH
     +ETA (DEGREE)')
      WRITE (*,*) '      -------------------------------------------
     +----------------------'
      WRITE(*,120) NTUNB,THNETN,THETAN
120   FORMAT(6X,F9.4,18X,F9.4,20X,F9.4)
      RETURN
      END


*************************************************************************

      SUBROUTINE POSTN (THL,THH,THETA,ANG,THM,THBLD,IX,POS,P,N)

*************************************************************************

*     PURPOSE : TO IDENTIFY THE POSITION AND THE VALUE OF THE CLOSEST
*               MASS TO THE UNBALANCED VECTOR

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER IX,POS,N
      REAL*8 P(166)
```

```
* THE CONSTRUCTION OF THE ANGLE BOUNDARIES ( HIGH & LOW )

      THL=0.0
      THH=ANG+0
10    CONTINUE
        IF(THETA.LE.THH) GO TO 20
          THL=THL+ANG
            THH=THL+ANG
          GO TO 10
20      CONTINUE
      THM=(THH+THL)/2
      IF(THETA.GT.THM) GO TO 30

* SECTION MARKING WITH RESPECT TO THE MIDDLE ANGLE (THM)

      THBLD=THL+0
      IX=0
      GO TO 40
30    THBLD=THH+0
      IX=1
40    CONTINUE

* FINDING THE POSITION

      POS=THBLD/ANG
      IF(POS.EQ.0) GO TO 45
      GO TO 46
45    POS=POS+N
46    CONTINUE
      RETURN
      END




*************************************************************************

      SUBROUTINE SWAP1 (P,POS,MAVG,IX,N,OPT,INC,ISN,POSN)

*************************************************************************


*     PURPOSE : TO CHOSE THE PROPER MASS VALUE TO SWAP

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER IX,POSN,POS,OPT,INC,ISN,N
      REAL*8 P(166),MAVG

      IF(P(POS).GE.MAVG) GO TO 60
        IF(IX.EQ.0) GO TO 555
        ISN=-1
         GO TO 666
555      ISN=1
      CONTINUE

      DO 700 I=1,OPT
      INC=ISN*I
      POS=POS+INC
```

```
      IF(POS.GT.N) GO TO 888
      IF(POS.EQ.N) GO TO 50
      IF(POS.LE.0) GO TO 3
      GO TO 50
3     POS=POS+N
      GO TO 50
888   POS=POS-N
50    CONTINUE
      IF(P(POS).GE.MAVG) GO TO 60
      ISN=(-1)*ISN
700   CONTINUE
      GO TO 111

666   CONTINUE

      DO 800 I=1,OPT
      INC=ISN*I
      POS=POS+INC
      IF(POS.GT.N) GO TO 881
      IF(POS.EQ.N) GO TO 53
      IF(POS.LE.0) GO TO 33
      GO TO 53
33    POS=POS+N
      GO TO 53
881   POS=POS-N
53    CONTINUE
      IF(P(POS).GE.MAVG) GO TO 60
      ISN=(-1)*ISN
800   CONTINUE

111   POSN=POS
      P(POSN)=P(POS)
60    CONTINUE
      RETURN
      END



*************************************************************************

      SUBROUTINE POSSWP (TL,TH,TSWAP,THETA,ANG,TM,TSWBD,IY,POSWP,P,N)

*************************************************************************


*     PURPOSE : TO IDENTIFY THE POSITION AND VALUE OF THE MASS WHICH
*               IS BEING SWAPPED ACCORDING TO THE CLOSEST MASS TO THE
*               UNBALANCED VECTOR IN THE OTHER SIDE OF THE ROTOR

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER IY,POSWP,N
      REAL*8 P(166)

* THE CONSTRUCTION OF THE ANGLE BOUNDARIES ( HIGH & LOW )

      TL=0
      TH=ANG
```

```
10    CONTINUE
        IF(THETA.GE.180) GO TO 20
          TSWAP=THETA+180
           GO TO 30
20        CONTINUE
        TSWAP=THETA-180
30    CONTINUE
      IF(TSWAP.LE.TH) GO TO 40
        TL=TL+ANG
         TH=TL+ANG
        GO TO 10
40    CONTINUE
      TM=(TH+TL)/2
      IF(TSWAP.GE.TM) GO TO 50

* SECTION MARKING WITH RESPECT TO THE MIDDLE ANGLE (TM)

      TSWBD=TL+0
      IY=0
      GO TO 60
50    TSWBD=TH+0
      IY=1
60    CONTINUE

* FINDING THE POSITION IN THIS SIDE (180 DEGREE OPPOSITE)

      POSWP=TSWBD/ANG
      IF(POSWP.EQ.0) GO TO 45
      GO TO 46
45    POSWP=POSWP+N
46    CONTINUE
      RETURN
      END



**************************************************************************

      SUBROUTINE SWAP2 (P,POSWP,MAVG,IY,PPOS,POS,N,OPT,INC,ISN,POSWPN,
     1                 RADPO,RAD,RADN)

**************************************************************************


*     PURPOSE : TO CHOOSE THE PROPER MASS VALUE TO SWAP WITH THE
*               CHOSEN ONE IN SUBROUTINE "SWAP1"

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER IY,POS,POSWPN,POSWP,ISN,OPT,INC,N
      REAL*8 P(166),MAVG,RADPO,RAD(166),RADN(166)

* TO COMPARE MASS VALUE TO AVERAGE MASS

      IF(P(POSWP).LE.MAVG) GO TO 10
        IF(IY.EQ.0) GO TO 555
      ISN=-1
      GO TO 666
```

```
555     ISN=1
        CONTINUE

        DO 700 I=1,OPT
        INC=ISN*I
        POSWP=POSWP+INC
        IF(POSWP.GT.N) GO TO 888
        IF(POSWP.EQ.N) GO TO 50
        IF(POSWP.LE.0) GO TO 3
        GO TO 50
3       POSWP=POSWP+N
        GO TO 50
888     POSWP=POSWP-N
50      CONTINUE
        IF(P(POSWP).LE.MAVG) GO TO 60
        ISN=(-1)*ISN
700     CONTINUE
        GO TO 111

666     CONTINUE

        DO 800 I=1,OPT
        INC=ISN*I
        POSWP=POSWP+INC
        IF(POSWP.GT.N) GO TO 881
        IF(POSWP.EQ.N)  GO TO 53
        IF(POSWP.LE.0) GO TO 33
        GO TO 53
33      POSWP=POSWP+N
        GO TO 53
881     POSWP=POSWP-N
53      CONTINUE
        IF(P(POSWP).LE.MAVG) GO TO 60
        ISN=(-1)*ISN
800     CONTINUE

111     POSWPN=POSWP
        P(POSWPN)=P(POSWP)
60      CONTINUE

*       TO DEFINE THE MASSES AND RADII WHICH ARE SWAPPED EACH OTHER

        PPOS=P(POS)
        P(POS)=P(POSWP)
        P(POSWP)=PPOS

           RADPO=RAD(POS)
           RAD(POS)=RAD(POSWP)
           RAD(POSWP)=RADPO

        GO TO 90

* TO DEFINE THE MASSES WHICH ARE SWAPPED EACH OTHER

10      PPOS=P(POS)
        P(POS)=P(POSWP)
        P(POSWP)=PPOS

           RADPO=RAD(POS)
```

```
         RAD(POS)=RAD(POSWP)
         RAD(POSWP)=RADPO
      GO TO 120
90    CONTINUE
120   CONTINUE
      RETURN
      END
```

APPENDIX B

SAMPLE OUTPUT FROM PROGRAM RBB

GAS TURBINE ROTOR BALANCING PROGRAM BY BLADE ORDER

SELECT AND ENTER : 1 FOR HPC , 2 FOR HPT , 3 FOR LPT
1

SELECTION 1 IS ON PROCESS

WHICH STAGE DO YOU WANT TO CHOOSE IN HPC ? 1 THRU 16
2

DATA FILE

HIGH PRESS. COMPR. STAGE NO : 2 BLADE NO : 26

MASSES	RADII
191.50	9.54500
191.27	9.54500
191.02	9.54500
190.90	9.54500
190.57	9.54500
190.34	9.54500
190.11	9.54500
189.87	9.54500
189.66	9.54500
189.42	9.54500
189.19	9.54500
188.96	9.54500
188.73	9.54500
188.27	9.54500
188.04	9.54500
187.81	9.54500
187.59	9.54500
187.34	9.54500
187.12	9.54500
186.91	9.54500
186.66	9.54500

186.43	9.54500
186.10	9.54500
185.97	9.54500
185.74	9.54500
185.50	9.54500

AVERAGE MASS,MAVG = 188.50 MASS TOTAL,TOTM = 4901.02

ENTER THE ANGULAR ROTOR SPEED (RADIAN/SECONDS)
1

W = 1.000 RAD/SEC

FIRST BLADE SET UP CONFIGURATION

POSITION	ANGLE (DEG.)	MASS
1	13.846	191.50
2	27.692	191.02
3	41.538	190.57
4	55.385	190.11
5	69.231	189.66
6	83.077	189.19
7	96.923	188.73
8	110.769	188.04
9	124.615	187.59
10	138.462	187.12
11	152.308	186.66
12	166.154	186.10
13	180.000	185.74
14	193.846	191.27
15	207.692	190.90
16	221.538	190.34
17	235.385	189.87

18	249.231	189.42
19	263.077	188.96
20	276.923	188.27
21	290.769	187.81
22	304.615	187.34
23	318.462	186.91
24	332.308	186.43
25	346.154	185.97
26	360.000	185.50

ENTER THE OPTION NUMBER WHICH WILL CHOOSE THE BEST MASS
TO SWAP. (MINIMUM THE HALF OF BLADE NUMBER)
16

OPTION IS 16 FOR THIS STAGE

HOW MANY TIMES DO YOU WANT TO ITERATE THE PROGRAM ?
1748

POS.	MASS	RADIUS	X-COMP	Y-COMP
1	187.81	9.54500	1740.5554	429.0084
2	190.34	9.54500	1608.6923	844.3069
3	189.87	9.54500	1356.5329	1201.7833
4	188.04	9.54500	1019.5864	1477.1258
5	186.43	9.54500	631.0103	1663.8374
6	188.27	9.54500	216.6089	1783.9347
7	190.90	9.54500	-219.6348	1808.8550
8	191.27	9.54500	-647.3923	1707.0331
9	186.10	9.54500	-1009.0673	1461.8864
10	187.59	9.54500	-1340.2433	1187.3520
11	191.02	9.54500	-1614.4395	847.3232
12	185.50	9.54500	-1719.1472	423.7317
13	189.66	9.54500	-1810.3047	.0000
14	187.12	9.54500	-1734.1607	-427.4322
15	185.97	9.54500	-1571.7585	-824.9225
16	190.57	9.54500	-1361.5341	-1206.2139

17	190.11	9.54500	-1030.8103	-1493.3865
18	188.73	9.54500	-638.7951	-1684.3643
19	188.96	9.54500	-217.4028	-1790.4728
20	187.34	9.54500	215.5389	-1775.1226
21	186.91	9.54500	632.6350	-1668.1213
22	191.50	9.54500	1038.3471	-1504.3055
23	189.42	9.54500	1353.3178	-1198.9350
24	189.19	9.54500	1598.9729	-839.2057
25	185.74	9.54500	1721.3714	-424.2799
26	186.66	9.54500	1781.6697	.0000

SUM OF X COMPONENT	SUM OF Y COMPONENT
.14849642	-.58421279

NET UNBALANCED VECTOR MAGNITUDE	DIRECTION OF VECTOR THETA (RADIAN)	DIRECTION OF VECTOR THETA (DEGREE)
.6028	4.9613	284.2615

RUN NUMBER FOR BEST BLADE CONFIGURATION : 1748

TOTAL RUN NUMBER FOR ITERATION PROCESS : 1748

DO YOU WANT TO TRY LARGER ITERATION NUMBER? (1 OR 0)
0

BALANCING PROCEDURE IS OVER FOR THIS STAGE

RECOMPILE PROGRAM FOR DIFFERENT STAGE NUMBERS
****
Stop - Program terminated.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
2.	Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	1
3.	Professor Paul F. Pucci, Code 69Pc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	4
4.	Naval Engineering Curricular Officer, Code 34 Naval Postgraduate School Monterey, California 93943-5000	1
5.	Ali Riza Yardimoglu Harp Akad.Sitesi D-85 Zincirlikuyu Istanbul/Turkey	2
6.	Professor Dr. Gultekin Gunay Turkiye Bilimsel ve Teknik Arastirma Kurumu Muhendislik Arastirma Grubu Yurutme Komitesi Sekreteri Ataturk Bulvari 221 06100 Kavaklidere Ankara/Turkey	1
7.	Professor Dr. Kemal Kafali ITU Rektoru Istanbul/Turkey	1
8.	Orta Dogu Teknik Universitesi Makine Muh.Fakultesi Dekanligi Ankara/Turkey	1

LIST OF REFERENCES

1. Kakad, Y. P., Hari, Y., "The Application of Non-Linear Programming Technique to Balancing Flexible Rotors," 12th Annual Southeastern Symposium on System Theory, Virginia Beach, Virginia, May, 1980.

2. Vance, J. M., Rotor Dynamics of Turbomachinery, pp. 34-37, John Wiley and Sons, Inc., Canada, 1988.

3. Norbye, J. P., The Gas Turbine Engine, pp. 66-79, Chilton Book Company, Pennsylvania, 1975.

4. Frantz, R., Kennedy, B., BLADEBAL.DEP LM2500 Blade Balance Program, Version 2.5, General Electric Company, Cincinnati, Ohio, January 1985.

5. Rao, J. S., Rotordynamics, pp. 189-202, John Wiley and Sons, Inc., New York, 1983.

6. Rieger, N. F., Balancing of Rigid and Flexible Rotors, The Shock and Vibration Information Center, Naval Research Laboratory, pp. 1-12, Washington, D. C., 1986.

9. Istanbul Teknik Universitesi 1
Makine Muh.Fakultesi Dekanligi
Istanbul/Turkey

10. Dz. K. K. Personel Egitim Daire Bsk.ligi 3
Bakanliklar
Ankara/Turkey

11. Bogazici Universitesi 1
Makine Muh. Fakultesi Dekanligi
Istanbul/Turkey

12. Dz. H. O. Komutanligi 1
Tuzla
Istanbul/Turkey

13. Dz. K. K. Teknik Daire Bsk. Ligi 1
Bakanliklar
Ankara/Turkey

14. Hv. K. K. Teknik Daire Bsk.Ligi 1
Bakanliklar
Ankara/Turkey

15. Yildiz Universitesi 1
Makine Muh.Fakultesi Dekanligi
Istanbul/Turkey

16. Dz. H. Akademisi Komutanligi 1
Levent
Istanbul/Turkey