NUMAQ v1.1

Contents Page

INTRODUCTION

NUMAQ is a program for HP 49g+ & HP 50g calculators.

Briefly, NUMAQ's goal is to calculate the amount of machines needed per workstation to produce a certain amount of products.

NUMAQ is developed in UserRPL and also uses an input form developed in SysRPL. Initially, I made the input form with the help of Steen Schmidts' InFormBuilder v1.81; then the form's code was modified and significantly improved by César Vásquez Alvarado.

Contents of the ZIP file (ENGLISH Folder):

NUMAQ.hp: HP directory with all variables needed to run the program. It's advisable to transfer this file to HOME by the means of an SD card.

NUMAQ_v1.1_Manual_ENG.PDF: This document.

NUMAQ_v1.1_Quick_Guide.txt: Brief version of this manual, to read on the calculator.

THEORETICAL FRAMEWORK

Suppose we have to produce “n” different products, each one of them must be processed by one or more of the “m” workstations needed to carry out the production. Suppose each workstation is composed by one or more machines of the same kind, with a given standard process time for each product.

For each workstation we determine an efficiency E, utilization U and scrap rate p.

The plant will have to work a gross time of T minutes along a given time horizon when demand of Di units for each product must be met. In the same time horizon, we’ll have R minutes of time losses because of rest periods, lunch, etc.

The problem is to calculate the number of machines to be placed on each workstation to meet demand at the planned time horizon.

INSTALLATION & NUMAQ DIRECTORY CONTENTS

To install the program, copy the file NUMAQ.hp to HOME, either using the USB cable or through an SD card, the latter being more practical.

Using an SD card, the installation procedure is as follows:

-By the means of a PC with an SD slot, open NUMAQ.ZIP and extract NUMAQ.hp to the SD card.

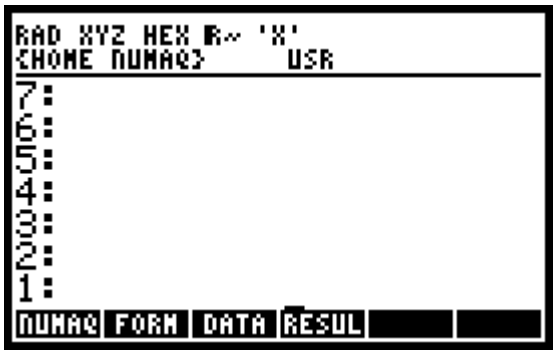
-Extract the SD card from the PC SD slot and put it in the calculators SD slot.

-Copy NUMAQ.hp from the SD card to HOME, using the Filer (Left Shift APPS).

For the remainder of the manual, it is assumed the user has FLAG 117 SET (Soft Menu) and FLAG 95 CLEARED (RPN mode).

Located in the HOME directory, by pressing the VAR key we see NUMAQ directory where all variables needed to execute the program are stored.

Inside the NUMAQ directory we find the following variables:



NUMAQ: variable where the main program is stored, by pressing F1 the program is run, opening the data input form in the first place.

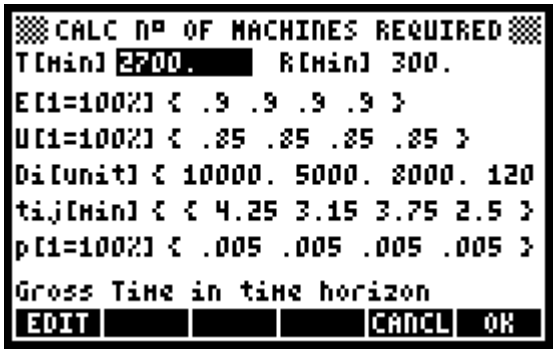
FORM: SysRPL program of the data input form, used by NUMAQ program. It's placed there just to be called by the NUMAQ program.

DATA: variable where data entered by the user (through the data input form) is stored in a list.

RESUL: directory where result variables are stored, after running the program

DATA ENTRY

Located in NUMAQ directory, pressing F1 (NUMAQ) shows the data input form, as follows:



The following data must be entered:

T [min] gross work time in minutes, for the given time horizon when demand must be met. For example, if the plant works the whole year in 2 x 8h shifts, 5 days a week, and demand must be met in a 1 year time horizon, T[min]=2*8*60*5*4*12=230400min (holydays should be discounted if any in the time horizon , example, if there were 15 holydays a year: T[min]=2*8*60*5*4*12 – 2*8*60*15=216000min). In this case, data to be entered is a number, for ex.: 230400.

R [min] represents time losses (rests, lunches, etc), also in minutes for the given time horizon when demand must be met. In last paragraph's example, if we have 30min per shift for lunch and 15min for rest, R[min]=2*(30+15)*5*4*12=21600min. Data to be entered is also a number, ex.: 21600.

E [1=100%] efficiency of machines in each workstation, for example amount of time machines worked, divided by amount of time machines should have worked. E=1 means 100% efficiency. Data to be entered in this case is a list of m numbers, where m is the number of workstations, for ex. if we have 5 workstations the efficiency list to be entered could be:

{ 0.9 0.85 0.95 0.75 0.8 }

U [1=100%] utilization of machines in each workstation. Utilization is the fraction of time that's decided the machine will be producing. Commonly for this model we suppose a utilization of 100% for all workstations, but in practice high utilization is assigned to expensive resources (ex: 95%) while lower utilization is assigned to cheaper resources. U=1 means a utilization of 100%. Data to be entered in this case is a list of m numbers, where m is the amount of workstations, for example if we have 5 workstations, the utilization list to be entered could be:

{ 0.7 0.8 0.95 0.75 0.6 }

Di [units] demand, i.e. amount of units of each product type to be produced during the planned time horizon. A list of n numbers should be entered, where n are the different classes (or types) of products being produced. Example, if 7 different product types are produced, the demand list could be:

{ 1500 2000 3400 2300 500 250 800 }

tij[min] are the production times, in minutes, for product i in workstation j. Ex.: product 3 requires a 1.5 process time in worstation 5, hence $t_{35} = 1.5$

Data to be entered has the form of a "list of lists" with the following format:

$$\left\{ \left\{t_{11}\ t_{21}....t_{n1}\right\}\ \left\{t_{12}\ t_{22}....t_{n2}\right\}\ \left\{t_{13}\ t_{23}....t_{n3}\right\}.....\left\{t_{1m}\ t_{2m}....t_{nm}\right\}\ \right\}$$

Where n are the different classes (or types) of products being produced and m is the amount of workstations. Each sublist contains the process times that one station employs to process a unit of each product type, for example $\{t_{13}\ t_{23}....t_{n3}\}$ is a list of process times used by workstation 3 to process one unit of each product type.

p [1=100%] scrap rate for workstations, ex: if workstation 3 produces 98 good units for every 100 processed units, we say it has a 2% scrap rate, i.e. $p_3 = 0.02$. A list of m numbers should be entered in this field, for example for a plant with 5 workstations could be:

{ 0.04 0.1 0.08 0.05 0.01 }

It is assumed the scrap rate is associated with the workstation, regardless of the type of product being processed.

When all data is correctly entered, by pressing F6 (OK), the program will save this data in N_MAQ_REQ variable in the hidden directory. It also saves this data in global variable DATA to be accessible to the user in case of need for any further calculation with this data.

After running the program for the first time, any subsequent execution of NUMAQ will display, in the input form, data that was entered in the previous execution of the program, so, should you made a mistake in the entry of any data or having to partially change data regarding the last run, this way is much more comfortable for the user.

Manual changes of the DATA variable by the user will not affect the next execution of NUMAQ, as the program in each execution calls the variable N_MAQ_REQ from the hidden directory to auto-complete the fields of the input form, and after the user makes the necessary changes to each field in the input form, pressing F6 (OK) saves the data entered to variable N_MAQ_REQ in the hidden directory and also saves it to the global variable DATA, and then perform calculations and shows results on screen.

DATA LIST FORMAT, DATA VARIABLE

After entering the data in the input form and pressing F6 (OK) the program stores data entered in the DATA variable.

The DATA variable is a “list of lists” with the following structure, according to previous definitions:

$$\left\{ T \left\{ E_1\ E_2 ... E_m \right\} R \left\{ U_1\ U_2 ... U_m \right\} \left\{ D_1\ D_2 ... D_n \right\} \left\{ \left\{ t_{11}\ t_{21} ... t_{n1} \right\} \left\{ t_{12}\ t_{22} ... t_{n2} \right\} ... \left\{ t_{1m}\ t_{2m} ... t_{nm} \right\} \right\} \left\{ p_1\ p_2 ... p_m \right\} \right\}$$

The user doesn’t need to directly interact with this variable to run the program, but as said earlier it can be useful if the user needs to do any other calculations involving this data.

RESULTS PRESENTATION, OUTPUT VARIABLES

NUMAQ gives the following results (output):

WL_j, Work Load of each workstation in minutes, during the considered time horizon. It uses the following equation to calculate it:

$$WL_j = \frac{\sum_{i=1}^{n} D_i \times t_{ij}}{1 - p_j}$$

AT_j, Available Time for production in minutes, for each workstation, during the considered time horizon. It uses the following equation to calculate it:

$$AT_j = (T - R) \times E_j \times U_j$$

n_j, the "machine fraction", a non necessarily integer quantity, representing the theoretical quantity of machines one must have in each of the j workstations to be able to satisfy demand. It uses the following equation to calculate it:

$$n_j = \frac{WL_j}{AT_j}$$

N_j, the integer quantity of machines one must have in each workstation to be able to satisfy demand, according the following criteria:

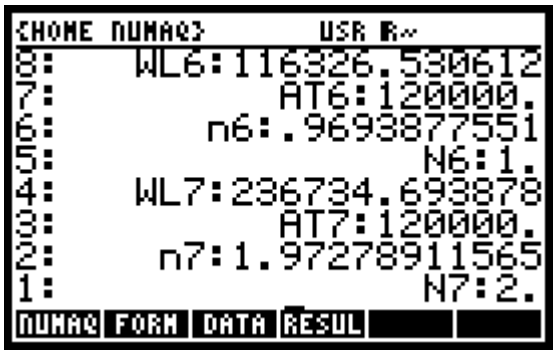
$$N_j = \left\{ x \,/\, x \in \mathbb{Z} \ \wedge \ 0 \le x - n_j < 1 \right\}$$

That is, if n_j is an integer then $N_j = n_j$, otherwise N_j is the integer immediately superior to n_j.

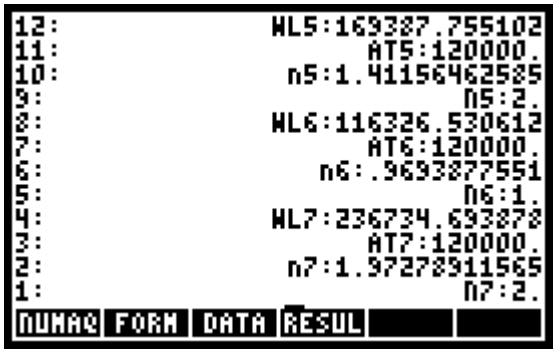
Results are presented individually at the stack and tagged, in the following order:

Level 4*m:	WL_1
Level 4*m-1:	AT_1
Level 4*m-2:	n_1
Level 4*m-3:	N_1
……	…….
Level 4*k+4:	WL_{m-k}
Level 4*k+3:	AT_{m-k}
Level 4*k+2:	n_{m-k}
Level 4*k+1:	N_{m-k}
……..	……
Level 8:	WL_{m-1}
Level 7:	AT_{m-1}
Level 6:	n_{m-1}
Level 5:	N_{m-1}
Level 4:	WL_m
Level 3:	AT_m
Level 2:	n_m
Level 1:	N_m

For the example we'll see further ahead, results are presented in the following way:

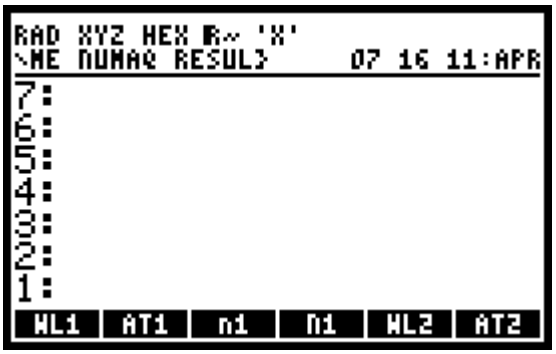


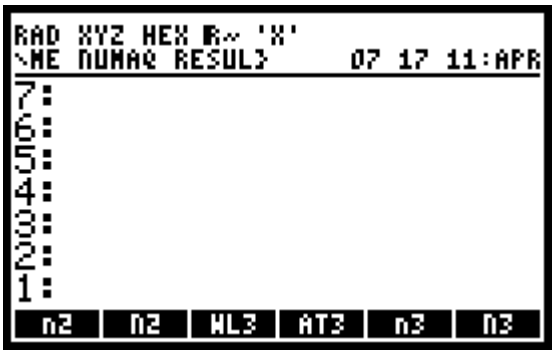
With a smaller stack font and removing the header, it's easier to see the order of results previously explained, in this case m=7:

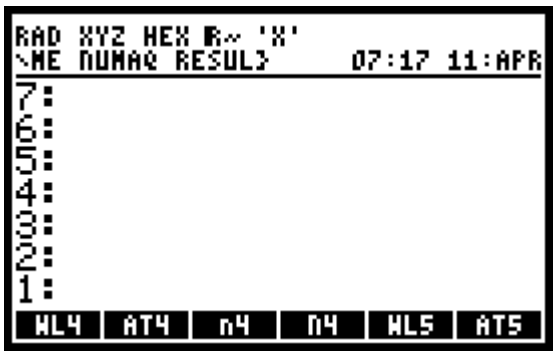


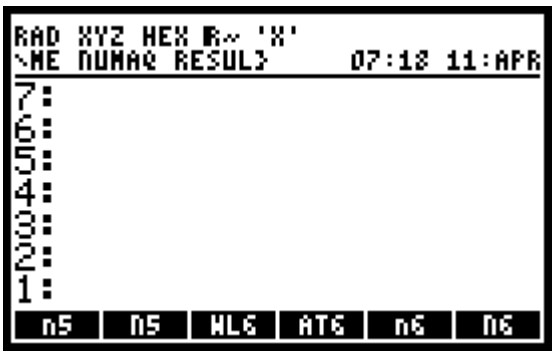
In page 14 you can see the full output for the given example.

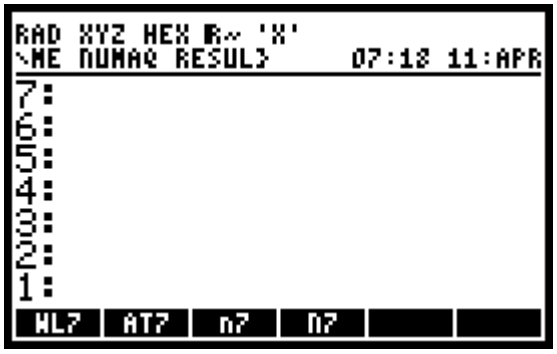
All the output data is stored as global variables in the RESUL directory, as seen in the following images:











It's important to notice that every time NUMAQ is ran, all global variables previously stored in the RESUL directory are overwritten with new values.

EXAMPLE

A factory produces nine products using seven workstations. Standard production times tij (in minutes) for each product in each workstation are given in the table below. Also annual demand for each product is given.

		Workstations							
		ST1	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**	**ST7**	**Demand**
Products	**P1**	2,5	0	0	0	1,7	0	0	**20000**
	P2	0	0	0	3,2	0	0	1,8	**30000**
	P3	0	0	1,5	0	0	2,1	0	**20000**
	P4	1,3	0	0	0	0	0	0	**50000**
	P5	0	2,1	1,5	0	0	0	0	**75000**
	P6	0	0	0	2,8	0	0	1,2	**35000**
	P7	0	0	0	0	0	0	1,7	**80000**
	P8	1,8	0	0	0	2,2	0	0	**60000**
	P9	0	0,9	2	0	0	1,5	0	**48000**

The factory works for a single 9h shift a day, 250 days a year. Workers have 45 minutes for lunch and a 15 minutes break in the afternoon.

There's a 2% scrap rate for all products in all workstations.

To make things simpler, efficiency and utilization are considered to be 100% for all workstations.

Calculate the quantity of machines needed in each workstation to be able to satisfy demand.

Resolution:

1) Calculate the gross work time for the year and time losses for the year due to lunches and breaks.

$$T = 9\frac{h}{day} \times 250 \text{ days} = 2250 \text{ h} \times 60 \frac{min}{h} = 135000 \text{ min}$$

$$R = (15+45)\frac{min}{día} \times 250 \text{ días} = 15000 \text{ min}$$

2) Calculate the Work Load (WL) for each workstation.

$$WL_j = \frac{\sum_{i=1}^{n} D_i \times t_{ij}}{1 - p_j}$$

$$WL_1 = \frac{\sum_{i=1}^{n} D_i \times t_{i1}}{1 - p_1} = \frac{20000 \times 2,5 + 30000 \times 0 + 20000 \times 0 + 50000 \times 1,3 + 75000 \times 0 + 35000 \times 0 + 80000 \times 0 + 60000 \times 1,8 + 48000 \times 0}{1 - 0,02}$$

$$WL_1 = 227551,02 \text{ min}$$

$$WL_2 = \frac{\sum_{i=1}^{n} D_i \times t_{i2}}{1 - p_2} = \frac{20000 \times 0 + 30000 \times 0 + 20000 \times 0 + 50000 \times 0 + 75000 \times 2,1 + 35000 \times 0 + 80000 \times 0 + 60000 \times 0 + 48000 \times 0,9}{1 - 0,02}$$

$$WL_2 = 204795,92 \text{ min}$$

$$WL_3 = \frac{\sum_{i=1}^{n} D_i \times t_{i3}}{1 - p_3} = \frac{20000 \times 0 + 30000 \times 0 + 20000 \times 1,5 + 50000 \times 0 + 75000 \times 1,5 + 35000 \times 0 + 80000 \times 0 + 60000 \times 0 + 48000 \times 2}{1 - 0,02}$$

$$WL_3 = 243367,35 \text{ min}$$

$$WL_4 = \frac{\sum_{i=1}^{n} D_i \times t_{i4}}{1 - p_4} = \frac{20000 \times 0 + 30000 \times 3,2 + 20000 \times 0 + 50000 \times 0 + 75000 \times 0 + 35000 \times 2,8 + 80000 \times 0 + 60000 \times 0 + 48000 \times 0}{1 - 0,02}$$

$$WL_4 = 197959,18 \text{ min}$$

$$WL_5 = \frac{\sum_{i=1}^{n} D_i \times t_{i5}}{1 - p_5} = \frac{20000 \times 1,7 + 30000 \times 0 + 20000 \times 0 + 50000 \times 0 + 75000 \times 0 + 35000 \times 0 + 80000 \times 0 + 60000 \times 2,2 + 48000 \times 0}{1 - 0,02}$$

$$WL_5 = 169387{,}76 \text{ min}$$

$$WL_6 = \frac{\sum_{i=1}^{n} D_i \times t_{i6}}{1 - p_6} = \frac{20000 \times 0 + 30000 \times 0 + 20000 \times 2{,}1 + 50000 \times 0 + 75000 \times 0 + 35000 \times 0 + 80000 \times 0 + 60000 \times 0 + 48000 \times 1{,}5}{1 - 0{,}02}$$

$$WL_6 = 116326{,}53 \text{ min}$$

$$WL_7 = \frac{\sum_{i=1}^{n} D_i \times t_{i7}}{1 - p_7} = \frac{20000 \times 0 + 30000 \times 1{,}8 + 20000 \times 0 + 50000 \times 0 + 75000 \times 0 + 35000 \times 1{,}2 + 80000 \times 1{,}7 + 60000 \times 0 + 48000 \times 0}{1 - 0{,}02}$$

$$WL_7 = 236734{,}69 \text{ min}$$

3) Calculate the available time AT for each workstation.

$$AT_j = (T - R) \times E_j \times U_j$$

In this example, as efficiency E and utilization U are equal to 100% for all workstations, then available time for work is identical for all workstations:

$$AT_j = (135000 - 15000) \times 1 \times 1 = 120000 \text{ min} \quad \forall j$$

4) Calculate the machine fraction n for each workstation, that is, a non integer representing the theoretical amount of machines every workstation should have to satisfy demand for all products. The number of machines N needed for each workstation is deduced from the machine fraction n, N being the immediately superior integer for each workstation with an n machine fraction.

$$n_j = \frac{WL_j}{AT_j} \qquad N_j = \left\{ x \,/\, x \in \mathbb{Z} \ \wedge \ 0 \le x - n_j < 1 \right\}$$

$$n_1 = \frac{WL_1}{AT_1} = \frac{227551{,}02\ \text{min}}{120000\ \text{min}} = 1{,}896 \Rightarrow N_1 = 2$$

$$n_2 = \frac{WL_2}{AT_2} = \frac{204795{,}92\ \text{min}}{120000\ \text{min}} = 1{,}707 \Rightarrow N_2 = 2$$

$$n_3 = \frac{WL_3}{AT_3} = \frac{243367{,}35\ \text{min}}{120000\ \text{min}} = 2{,}028 \Rightarrow N_3 = 3$$

$$n_4 = \frac{WL_4}{AT_4} = \frac{197959{,}18\ \text{min}}{120000\ \text{min}} = 1{,}650 \Rightarrow N_4 = 2$$

$$n_5 = \frac{WL_5}{AT_5} = \frac{169387{,}76\ \text{min}}{120000\ \text{min}} = 1{,}412 \Rightarrow N_5 = 2$$

$$n_6 = \frac{WL_6}{AT_6} = \frac{116326{,}53\ \text{min}}{120000\ \text{min}} = 0{,}969 \Rightarrow N_6 = 1$$

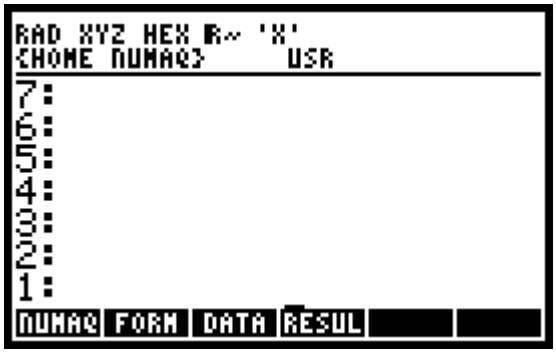
$$n_7 = \frac{WL_7}{AT_7} = \frac{236734{,}69\ \text{min}}{120000\ \text{min}} = 1{,}973 \Rightarrow N_7 = 2$$

The quantity of machines needed for each workstation is given in the following table:

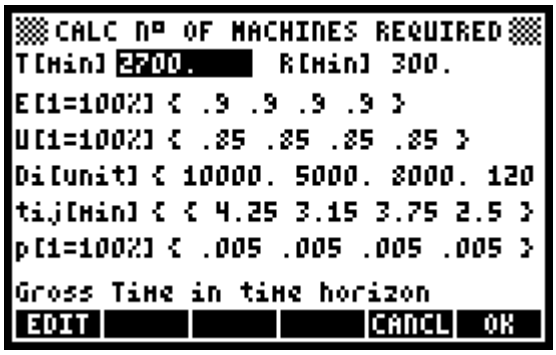
ST1	ST2	ST3	ST4	ST5	ST6	ST7
2	2	3	2	2	1	2

SOLVING PREVIOUS EXAMPLE USING NUMAQ

Start by going into NUMAQ directory, then we should see the following:



Pressing F1 (NUMAQ) the data input form will show. In case you run the program for the first time, some default data will appear.



We enter the following data (T and R must be calculated previously), pressing ENTER after typing the data corresponding to each field:

$T = 135000$

$R = 15000$

$E = \{ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \}$

$U = \{ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \}$

$D_i = \{ \ 20000 \ \ 30000 \ \ 20000 \ \ 50000 \ \ 75000 \ \ 35000 \ \ 80000 \ \ 60000 \ \ 48000 \ \}$

$t_{ij} = \{ \ \{ \ 2.5 \ \ 0 \ \ 0 \ \ 1.3 \ \ 0 \ \ 0 \ \ 0 \ \ 1.8 \ \ 0 \ \}$

$\{ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 2.1 \ \ 0 \ \ 0 \ \ 0 \ \ 0.9 \ \}$

$\{ \ 0 \ \ 0 \ \ 1.5 \ \ 0 \ \ 1.5 \ \ 0 \ \ 0 \ \ 0 \ \ 2 \ \}$

$\{ \ 0 \ \ 3.2 \ \ 0 \ \ 0 \ \ 0 \ \ 2.8 \ \ 0 \ \ 0 \ \ 0 \ \}$

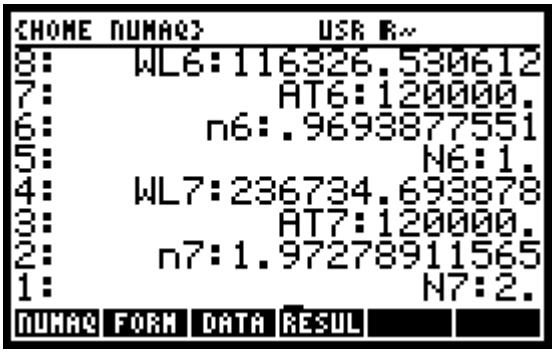
$\{ \ 1.7 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 2.2 \ \ 0 \ \}$

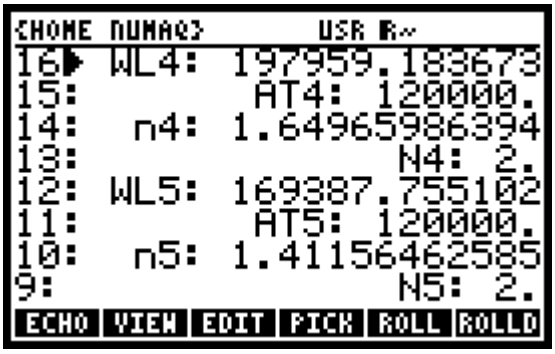
$\{ \ 0 \ \ 0 \ \ 2.1 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 1.5 \ \}$

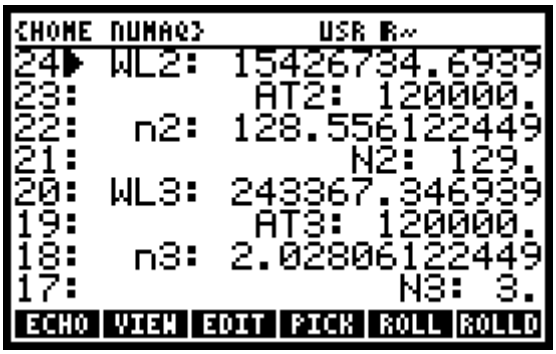
$\{ \ 0 \ \ 1.8 \ \ 0 \ \ 0 \ \ 0 \ \ 1.2 \ \ 1.7 \ \ 0 \ \ 0 \ \} \ \}$

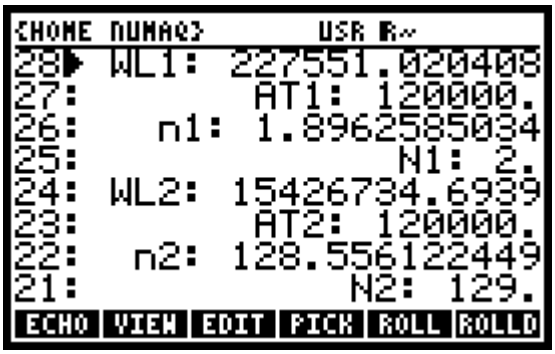
$p = \{ \ .02 \ \ .02 \ \ .02 \ \ .02 \ \ .02 \ \ .02 \ \ .02 \ \}$

Once all data is entered, press **F6 (OK)**. NUMAQ will run for a few seconds (aprox. 6 seconds for this example on an HP 50g) after which the following results will be shown at the stack:









AKNOWLEDGMENTS

César Vásquez Alvarado (CesarV), who notably improved the SysRPL code for the input data form. Visit **CesarV's site** (spanish) and **HonradosHP** forum (spanish).

Steen Schmidt, for his **InFormBuilder v1.81**

Roger Broncano Reyes, for his masterpiece **HPUserEdit 5**.

AdictosHP community (spanish).

comp.sys.hp48 community and its valuable archive.

Eric Rechlin for **HPCalc.org**.