

# $x$ MPS, the Extended MPS Format for Non-Linear Programs\*

Erlendur S. Thorsteinsson†

esth@cmu.edu

20th December 1999

## Abstract

We present a file format for Non-Linear Programs (NLP), the extended MPS format ( $x$ MPS). This format directly extends the standard MPS file format for Linear and Mixed Integer Programs to include NLPs and permits a standardized way of transferring benchmark problems. The format spares the modeler the tedious task of calculating derivative information with minimal extra work required by the optimizer and thus increases efficiency.

## 1 Introduction

The main criteria for a file format for Non-Linear Programs (NLP) is that it (a) be simple to use, (b) be compatible with some popular format used for linear programs, (c) extend in a simple way to memory representation, and (d) allow for automatic (analytical) differentiation. This would ensure, e.g., that (a) benchmarks problems could be easily transferred between systems, (b) existing Linear Programming (LP) readers would only have to be slightly modified to read the new NLP format, (c) problems could be transferred in a fast manner from the modeler to the optimizer, and (d) optimizers could easily be provided with exact derivatives at every point in the solution space without the derivative functions being explicitly included in the model.

The MPS file format is a widely accepted standard for expressing LP problems, recognized by many LP solvers and modeling languages. In this paper we are going to describe a file format for NLPs, called  $x$ MPS, which extends MPS. More information on MPS can be found in [3] and how different software packages implement MPS in [1, 2, 4].

## 2 The Extended MPS ( $x$ MPS) File Format

The MPS format is a *column oriented* format, defining the problem by columns (variables), rather than by rows (constraints) as problems are usually expressed in mathematical notation. Note that historically it included many restrictions, e.g., requiring input fields to occupy fixed columnar positions and limiting names to 8 characters or less. We choose, however, to extend a relaxed version of the MPS format where, e.g., input fields are separated by whitespaces (names can then not contain spaces as they can in the original MPS format) and names can be more than 8 characters.

An  $x$ MPS data file, like an MPS data file, consists of records of two types, *indicator* records and *data* records. Indicator records are only a single word and separate the individual sections of the  $x$ MPS file. The data records contain the information that describes the LP problem.

---

\*Carnegie Mellon University Mathematical Sciences Department Research Report No. 99-224

†Graduate School of Industrial Administration; Carnegie Mellon University; Pittsburgh, PA 15213; U.S.A.

## 2.1 Indicator Records

There are ten types of indicator records, of which NAME, ROWS, COLUMNS and ENDATA are required. Each keyword must begin in the first column of the data file and the records must appear in the data file in the following order:

**NAME** indicator record. Contains the problem name immediately following the indicator keyword.

**OBJSENSE** indicator record. The corresponding data records indicate the sense of the optimization, minimization or maximization. If not present then minimization will be assumed.

**ROWS** indicator record. The corresponding data records provide the name and sense of each row (constraint) of the matrix and the name of the objective function.

**COLUMNS** indicator record. The corresponding data records contain the name of each column (variable) of the matrix, the name of the row in which a coefficient in that column appears, and the value of the coefficient. Up to two coefficients may be defined in each column record. As many records for each column are included as necessary. The coefficients for the objective function are defined in a similar manner.

**NONLINEAR** indicator record. The corresponding data records contain the non-linear part of a row. A complete row is the sum of the linear part specified by the COLUMNS section and the non-linear part specified by this section.

**RHS** indicator record. The corresponding data records are the names and right-hand side values of the rows in the matrix. If not present then a default right-hand side of 0 will be assumed.

**RANGES** indicator record. The corresponding data records specify rows that are restricted to lie in the interval between two values.

**BOUNDS** indicator record. The corresponding data records contain the name and bounds of a column in the matrix. If not present then default bounds of 0 to  $+\infty$  will be assumed.

**INITIAL** indicator record. The corresponding data records are the initial values for each variable. If not present then a default initial value of 0 will be assumed.

**ENDATA** indicator record. The end of the data.

All variables and constraints must be declared in the ROWS and COLUMNS sections before being referenced in subsequent sections.

## 2.2 Data Records

Each data record contains up to six fields. Generally the fields contain the following information, but not all fields are used within each section of the data file:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
Indicator	Name	Name	Value	Name	Value

The fields are separated by *white space* characters, spaces and/or tabs. The first column of the data file may not be used for the data records, so the first field must begin in column two of the data file or beyond. No other restrictions are placed on the columnar positions of the fields of the data records. Each field should contain no more than 256 characters (any extra characters could be truncated).

Any ISO-8859-1 character (characters 32-126 and 161-255) is legal as input, but names may not contain any blanks. Values can be defined with decimal or exponential notation, using a period “.” as a decimal separator and “e” or “E” as an exponent separator.

Comments can be inserted into the data file by placing an asterix “\*” in column 1, then that line will be treated as a comment and ignored. Moreover, a dollar sign “\$” as the first character of any field indicates that the information from that point to the end of the line consists of a comment.

**OBJSENSE section** The data record in the OBJSENSE section specifies the sense of the optimization. The data record has the following form:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
	Sense				

where the sense is as follows:

MIN    Minimization,  
 MAX    Maximization.

**ROWS section** Data records in the ROWS section specify the name to be assigned to each row (constraint) of the model, as well as the type of constraint represented by the row. The name of the objective function is also defined in this section, even though it is not a constraint in the model. The data records have the following form:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
Type code	Row name				

where the type of row is as follows:

N    No constraint (usually the objective function),  
 G    Greater than or equal to,  
 L    Less than or equal to,  
 E    Equality.

**COLUMNS section** Data records in the COLUMNS section specify the names to be assigned to each column (variable) in the model and define, in column order, the coefficients of the model. The use of special markers in the COLUMNS section allows the modeler to define sets of integer variables. In addition, data records in this section define the coefficients of the objective function. The data records have the following form:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
	Column name	1 <sup>st</sup> row name	Coef. in 1 <sup>st</sup> row	2 <sup>nd</sup> row name	Coef. in 2 <sup>nd</sup> row

where fields 5 and 6 are optional. If they are used, they contain another row/coefficient pair for that column, as in fields 3 and 4, respectively. It is assumed that a coefficient that is not specified as described above is zero. Multiple records may be required to completely specify a column.

**NONLINEAR** The non-linear part of each row is represented by a list of lines where each line is of the type:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
	Row name	Line name	Operator	Argument 1	Argument 2

For each row the lines form a stack machine which expresses the non-linear part. The lines for a particular row should be contiguous, i.e., the NONLINEAR section is divided into subsection based on the row names. The line names within each subsection should be distinct and different from the column and row names. The last line name in each subsection should be RES, indicating the result of the stack machine.

The operator is in the set of known keywords for operators, see App. A. The arguments are either constants, column names drawn from the COLUMN section or line names of lines above the current line in the current subsection.

Variables that are only referred to in nonlinear equations (i.e., do not occur linearly in any constraint) still need to be declared in the COLUMNS section with only the second field populated. This allows them to be referred to in the NONLINEAR section.

Evaluating each stack machine is simple: We start with an empty stack and execute each line of the stack machine in order. At every line we apply the operator to its arguments and put the result on the stack and label it with the line name. The result of the evaluation is the element at the top of the stack when then line marked RES has been executed.

**RHS section** Data records in the RHS section specify the names and values of the right-hand side constraint vectors. The data records have the following form:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
	RHS name	1 <sup>st</sup> row name	RHS for 1 <sup>st</sup> row	2 <sup>nd</sup> row name	RHS for 2 <sup>nd</sup> row

where fields 5 and 6 are optional. If they are used, they contain another row/RHS pair, as in fields 3 and 4, respectively.

**RANGES section** Data records in the RANGES section specify the RHS range values to be applied to the constraints. The data records have the following form:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
	RHS name	1 <sup>st</sup> row name	Range for 1 <sup>st</sup> row	2 <sup>nd</sup> row name	Range for 2 <sup>nd</sup> row

where fields 5 and 6 are optional. If they are used, they contain another row/RHS range pair, as in fields 3 and 4, respectively.

The effects of specifying a RHS range depends on the sense of the specified row and the sign of the range:

Row type	Range sign	Effective lower RHS	Effective upper RHS
G	+ or -	rhs	rhs+ range
L	+ or -	rhs- range	rhs
E	+	rhs	rhs+range
E	-	rhs+range	rhs

**BOUNDS section** Data records in the BOUNDS section specify the names and values of the bound vectors, which impose limits on the values the columns (variables) may assume. The data records have the following form:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
Type code	Bounds name	1 <sup>st</sup> column name	Bound for 1 <sup>st</sup> row		

where the type of bound is as follows:

- LO Lower bound,
- UP Upper bound,
- FX Fixed variable,
- FR Free variable,
- MI Minus infinity,
- PL Plus infinity,
- BV Binary variable,
- LI Integer lower bound,
- UI Integer upper bound.

**INITIAL** Data records in the INITIAL section specify the values of the initial solution values. The data records have the following form:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
	Init. name	1 <sup>st</sup> col. name	1 <sup>st</sup> col. value	2 <sup>nd</sup> col. name	2 <sup>nd</sup> col. value

where fields 5 and 6 are optional. If they are used, they contain another column/value pair, as in fields 3 and 4, respectively.

### 2.3 An Example

To illustrate the *x*MPS format let us consider the following optimization problem:

$$\begin{aligned}
 \min \quad & \sin(x_1) + x_1x_2 + 2x_2 && (obj) \\
 \text{s.t.} \quad & x_1 + x_2 \leq 4, && (g_1) \\
 & 4 \ln(x_1x_2) + x_1 \geq 1. && (g_2)
 \end{aligned} \tag{1}$$

Looking at the linear part of (1) we notice that  $x_1$  appears in  $(g_1)$  and  $(g_2)$ , and  $x_2$  appears in  $(obj)$  and  $(g_1)$ . In addition,  $(obj)$  and  $(g_2)$  have some non-linear elements to them. Rewriting and regrouping we can thus state (1) in the following manner:

$$\begin{aligned}
 \min \quad & \left\{ \sin(x_1) + x_1x_2 \right\} + \begin{bmatrix} 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
 \text{s.t.} \quad & \left\{ \begin{array}{c} 0 \\ 4 \ln(x_1x_2) \end{array} \right\} + \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 4 \\ 1 \end{bmatrix}
 \end{aligned}$$

Suppose we are starting at  $x^0 = (x_1, x_2) = (1, 1)$ . This problem would be represented in *x*MPS as follows:

NAME	demo.xmps				
OBJSENSE	MIN				
ROWS					
N	obj				
L	g1				
G	g2				
COLUMNS					
x1	g1	1		g2	1
x2	obj	2		g1	1
NONLINEAR					
obj	v1	SIN	x1		
obj	v2	MULT	x1	x2	
obj	RES	ADD	v1	v2	
g2	v1	MULT	x1	x2	
g2	v2	LOG	v1		
g2	RES	MULT	4	v2	
RHS					
rhs	g1	4		g2	1
INITIAL					
init	x1	1		x2	1
ENDATA					

The ROWS section starts by declaring the rows,  $(obj)$ ,  $(g_1)$  and  $(g_2)$ . The COLUMNS section then declares the columns  $x_1$  and  $x_2$  and specifies in which rows each one occurs and the coefficients by which it occurs. We note, e.g., that according to this description the linear part of row  $(obj)$ , the objective function, is  $2x_2$ .

The NONLINEAR section then describes the non-linear part of each row. For example, the stack machine for  $(obj)$  would be evaluated in the following manner: We would first put  $\sin(x_1)$  on to the stack and label it  $v_1$ . We would then put  $x_1 \times x_2$  on to the stack and label it  $v_2$ . According to the last line in the  $(obj)$  subsection, the result should be  $v_1 + v_2$ , which is  $\sin(x_1) + x_1x_2$ .

Each row of the problem is then the sum of the linear and non-linear parts. Therefore, row  $(obj)$  is  $\sin(x_1) + x_1x_2 + 2x_2$ .

## 2.4 Integer Variables

Instead of using the bound types BV, LI and UI to specify integer variables, they can be identified in the COLUMNS section using special markers. A range of integer variables is enclosed by marker records that are inserted at the beginning and end of the range. Multiple markers are allowed. The marker records have the following form:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
	Marker name	'MARKER'	'INTORG' or 'INTEND'		

Note that the keywords in fields 3 and 4 include the quotation marks.

## 2.5 Special Ordered Sets

Three different types of special ordered sets are recognized:

- Type 1, where at most one variable in the set may take a nonzero value.
- Type 2, where at most two variables in the set may take nonzero values. If two variables are nonzero, they must be adjacent in the set.
- Type 3, where at most one variable in the set may take a nonzero value, and there was at least one equality row with an RHS value of 1.0 in which all the variables in the set had a coefficient of 1.0. In this case, it is possible to say that all the variables are 0–1 variables.

Special Ordered Sets are identified in the COLUMNS section using special markers. A range of variables is enclosed by marker records that are inserted at the beginning and end of the range. Multiple markers are allowed. The marker records have the following form:

For SOS types 1 or 2:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
S1 or S2	Marker name	'MARKER'	'SOSORG' or 'SOSEND'		

Note that the keywords in fields 3 and 4 include the quotation marks.

For SOS type 3:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
S3	Name of an equality row	'MARKER'	'SOSORG' or 'SOSEND'		

Note that the keywords in fields 3 and 4 include the quotation marks.

## 3 Conclusion

We have described an extension (*x*MPS) to the widely used linear and mixed integer programming MPS file format for non-linear programming.

*x*MPS has been implemented for the commercial modeling system MPL, which is available from Maximal Software at [www.maximal-usa.com](http://www.maximal-usa.com), using the NLP solvers CONOPT and LSGRG2.

## 4 Acknowledgments

The authors would like to thank Arne Drud for providing the initial draft of the nonlinear file format.

## References

- [1] Dash Optimization, Inc. *XPRESS-MP: User Manuals*, 2000. Version 12.11.
- [2] IBM. *The IBM Optimization Solutions and Library (OSL) for Multi-platforms Version 2*. [ism.boulder.ibm.com/es/oslv2/features/welcome.htm](http://ism.boulder.ibm.com/es/oslv2/features/welcome.htm).
- [3] IBM. *Mathematical Programming System Extended/370 (MPSX/370) Program Reference Manual*.
- [4] ILOG. *ILOG CPLEX Reference Manual*, 2001. Version 7.0.

## A Non-linear operators\*

Operator ( $O$ )	Keyword	$\partial O/\partial L$	$\partial O/\partial R$
	NONE		
$L + R$	ADD	1	1
$L - R$	SUB	1	-1
$L \times R$	MULT	$R$	$L$
$L/R$	DIV	$1/R$	$-L/R^2$
$-L$	NEG	-1	
$L + R$	SUM	1	1
$L^2$	SQR	$2L$	
$L^R$	POW	$L^{R-1}$	$\ln L \times L^R$
$\sqrt{L}$	SQRT	$1/(2\sqrt{L})$	
$L\%R$	MOD		
$e^L$	EXP	$e^L$	
$\ln L$	LOG	$1/L$	
$\log L$	LOG10	$1/(L \ln 10)$	
$\sin L$	SIN	$\cos L$	
$\cos L$	COS	$-\sin L$	
$\tan L$	TAN	$1/\cos^2 L$	
$\arcsin L$	ASIN	$1/(\sqrt{1-L^2})$	
$\arccos L$	ACOS	$-1/(\sqrt{1-L^2})$	
$\arctan L$	ATAN	$1/(1+L^2)$	
$\arctan(L/R)$	ATAN2	$1/(R(1+(L/R)^2))$	$-R/(L^2+R^2)$
$\sinh L$	SINH	$\cosh L$	
$\cosh L$	COSH	$\sinh L$	
$\tanh L$	TANH	$1/(\cosh^2 L)$	
$\operatorname{arcsinh} L$	ASINH	$1/\sqrt{1+L^2}$	
$\operatorname{arccosh} L$	ACOSH	$-1/\sqrt{1+L^2}$	
$\operatorname{arctanh} L$	ATANH	$1/(1-L^2)$	
$\operatorname{sign} L$	SIGN		
$ L $	ABS		
$\lceil L \rceil$	CEIL		
$\lfloor L \rfloor$	FLOOR		
$\operatorname{round} L$	ROUND		
$\operatorname{trunc} L$	TRUNC		

Table 1: Non-linear operators.

---

\*Please visit [www.maximal-usa.com/xmps](http://www.maximal-usa.com/xmps) for a current version of supported operators. Suggestions for additional operators should be submitted to [xmps@maximal-usa.com](mailto:xmps@maximal-usa.com).