MACROTEX : a LATEX code generator in MACSYMA

J.P. Chancelier, Agnès Sulem

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports Techniques

N° 93

MACROTEX : A LATEX CODE GENERATOR IN MACSYMA

Jean-Philippe CHANCELIER
Agnès SULEM

DECEMBRE 1987

MACROTEX : a LATEX code generator in MACSYMA
MACROTEX : Un generateur de code LATEX implanté en MACSYMA

Jean-Philippe Chancelier   Agnès Sulem
INRIA
Domaine de Voluceau
Rocquencourt- BP105
78153 Le Chesnay Cedex
France

December 8, 1987

Abstract

This paper deals with a LATEX code generator called MACROTEX , written in Common Lisp, devoted to the generation of LATEX code at Macsyma level. It is composed of a translator of Macsyma expressions into LATEX syntax and a generator of LATEX reports.

Résumé

Ce document présente un générateur de code LATEX appelé MACROTEX , écrit en Common Lisp, permettant de générer du code LATEX à partir de Macsyma, réunissant ainsi la puissance de calcul symbolique de Macsyma et la qualité d'édition de LATEX.
MACROTEX est composé d'un ensemble de fonctions de traduction de Macsyma en LATEX et de facilités pour générer des articles en LATEX.

Contents

1 Introduction

Macsyma is a computer algebra system suited to handle symbolic and numerical computations. LaTeX is a program designed to produce high-quality typesetting, especially for mathematical text. Our purpose is to gather Macsyma and LaTeX in order to use the computer algebra power of Macsyma and LaTeX quality of typesetting. A Macsyma user can thus perform symbolic manipulations and write LaTeX code without leaving Macsyma, and thus ignoring LaTeX syntax.
We thus have written a LaTeX Code Generator in Common Lisp that we have called "MACROTEX". It runs on a Symbolics Lisp Machine, release-7 in Common Lisp.

MACROTEX is composed of a set of elementary and macro-instructions devoted to the translations of mathematical expressions, using Macsyma expertise for the output (for example, divide long expressions into subexpressions) and to the automatic generation of a LaTeXreport, at Macsyma level.

2 Description of MACROTEX

2.1 The Macsyma-LaTeX translator

Let us describe the first part of MACROTEX devoted to the translator of Macsyma expressions into LaTeX Syntax:

- The basic function TEX(exp) takes as input a Macsyma mathematical expression, and optionally a label and outputs the expression in LaTeX syntax. The expression is numbered if there is a label and can be referenced elsewhere in the report (see the format directive Z).

- The function EXPTXT(exp) takes a mathematical Macsyma expression as argument and outputs it as a LaTeX in-text formula.

- The function FORMATT(string, args) behaves like the lisp function *format*, with the following additional directives : K, L, N, Y, Z.

  - ~K and ~L translates the arguments in LaTeX using respectively the functions TEX and EXPTXT.
  - ~N translates its argument in LaTeX syntax but not in a displaymath environment.
  - ~Y and ~Z are the directives to insert respectively a quotation or a reference.

- The function LISTEXPT takes a list of expressions as argument and optionally a position argument and a label. It outputs the system of mathematical expressions displayed one under the other in LaTeX syntax. The position argument specifies the alignment of the expressions (centered, flush right or flush left).

- The function DECOUPET is used when a mathematical expression is too long to fit on one line.

```
decoupet(arg1,[arg2,arg3,arg4],["+","-"])
        ==> arg1 = arg2
                 + arg3
                 - arg4
```

If the operators for each continuation line are all the same, the list of identical operators may be substituted by this operator.

- The macro-instruction DECOUPEM is provided for a user who does not know a priori how to cut a long mathematical expression. This function takes a Macsyma formula and a linel as arguments, and outputs the formula in LaTeX, cut into subformulae of size controlled by the value of the linel.

```
(C226) DECOUPEM(MATRIX([SUM(SUM(X[I,J],I,1,N1),J,1,N2), INTEGRATE(F(X),X,0,1)],
                  [DIFF(F(X),X[1],N),PRODUCT(X[I],I,1,N)]),LAB,60);
(C227) EXECUTE(%);
```

$$\begin{pmatrix} E1 & E2 \\ \frac{d^N F(X)}{dX^N} & \prod_{I=1}^{N} X_I \end{pmatrix}$$

$$E1 = \sum_{J=1}^{N2} \sum_{I=1}^{N1} X_{I\,J} \tag{1}$$

$$E2 = \int_0^1 F(X)\,dX$$

Like Macsyma display, the translator from Macsyma representation into LaTeX syntax is "data driven" implemented in order to be easily customized to the specific display needs of the user. A specific translation function is attached to the Macsyma operator via its property list. A Macsyma function TEX_TRANSLATION is provided in order to define specific translations for some Macsyma functions at Macsyma level.

2.2 The generator of LaTeX reports

The second part of MACROTeX is devoted to the generation of a LaTeX report at Macsyma level. A MACROTeX program is a Macsyma list compound of MACROTeX instructions. The translation into LaTeX is done by calling the function EXECUTE on it. The MACROTeX instructions are of two types, the elementary and the macro-instructions.

2.2.1 The elementary-instructions

A elementary-instruction is a Macsyma list the first element of which can be

1. the name of the corresponding LaTeX instruction with a "T" appended at its end: LABELT, REFT, CITET,...

   We list below a few of them with the LaTeX code they generate:

MACROTeX	LaTeX
[documentstylet,"article",11]	\documentstyle[11pt]{article}
[labelt, "Euler"]	\label{Euler}
[reft, "Newton"]	\ref{Newton}
[citet, "autl"]	\cite{autl}
[sectiont, "~N method","resolution"]	\section[resolution method]{resolution method}

Note that the instructions REFT and CITET can also be obtained by [formatt, "~Z", "Newton"] and [formatt, "~Y", "aut1"].

The user can customize the system to his needs and define his own instructions: For example by defining the following Macsyma function :

```
Mboxt(x):=formatt("\\mbox{~A}",x);
```

a new operator is added to the system to generate LaTeX code preventing line breaks in an expression.

2. one of the functions of the translator: TEX, EXPTXT, FORMATT, ...
3. one of the instructions which deal with global variables useful for the generation of the report:
   - the function CONTEXTT assigns values to the documentstyle parameters, and initializes the global variables "list_references" and "list_notation" to the empty lists. In these lists are stored the notation and the bibliography of the report.
   - the functions ADD_REFERENCET and ADD_NOTATIONT push a reference or a notation in "list_references" and "list_notation". ....

2.2.2 The macro-instructions

By macro-instruction we mean a command which will generate several elementary LaTeX commands. A LaTeX macro-instruction has the same syntax as an elementary-instruction but the translation is done in two steps: the macro-instruction is first expanded into a list of elementary-instructions which in turn are translated into LaTeX code. The macro-instructions are characterized by a "M" appended at the end of their names. For example, the macro BIBLIOM, when the global variable "list_references" is bound to [[label1,ref1],[label2,ref2]] expands into the Macsyma list :

```
[[beginbibliot,2],
 [bibitemt,[[label1,ref1],[label2,ref2]]],
 [endbibliot]]
```

Let us mention some macro-instructions which are used in the report generation:

- STYLEM makes the declaration of the documentstyle options,
- ABSTRACTM generates an abstract,
- NOTATIONM generates a section of notation, using the macsyma list "list_notations".
- ITEMIZEM and ENUMERATEM provide the itemize and enumerate list-making environments, ...

Let us breefly describe the macro REPORTM:
[REPORTM ,title, list-aut-addr, abstract, chapters] expands into

```
[[stylem, styleflag, sizeflag, heightflag, widthflag,
          topmarginflag, oddsidemarginflag, evensidemarginflag, marginparwidthflag],
title,
list-aut-addr,
[begindocumentt],
```

```
[maketitlet],
abstract,
[tableofcontentt],
[notationm],
chapters,
[bibliom],
[enddocumentt]]]
```

REPORTM first applies the macro STYLEM with the values of the documentstyle parameters assigned in the fuction CONTEXTT. It then outputs the title, the list of authors and adresses, the abstract, the table of contents (which can be omitted by changing a parameter in the function CONTEXTT), the section of the notations, the body of the report, and the bibliography. The report we get is complete and the LaTeX compiler can be called on it directly.

3 Example of a generated report

Let us now generate a whole article in MACROTEX :
The following Macsyma function STOCH_CONTROL(n1,n2,v,df,dr,cf,alea) generates a report on the Dynamic-programming method for solving stochastic control problems.
n1 is the dimension of the state variable, n2 is the dimension of the control variable, v is the optimal cost, dr and df are Macsyma-lists of length n1 (drift and diffusion terms) , cf is a mathematical expression (cost function) .

```
stoch_control(n1,n2,v,df,dr,cf,alea):=block([xx,uu,auv],
   CONTEXTT(),
   xx:makelist(\x[i],i,1,n1),
   uu:makelist(\u[i],i,1,n2),
   auv:funmake('A,uu).v,
ADD_REFERENCET(
 [Fleming,"FLEMING,W.H.-RISHEL,R.(1975).
   Deterministic and Stochastic Optimal Control. Springer verlag, New York."],
 [Quadrat,"QUADRAT,J.P.(1980).
   Existence de solution et algorithme de r\\'esolutions num\\'eriques de probl\\'emes
   stochastiques d\\'eg\\'en\\'er\\'es ou non. SIAM Journal of Control."]),
ADD_NOTATIONT(
 [FORMATT,"State variables: ~L ", xx],
 [FORMATT,"Control variables: ~L ",uu],
 [FORMATT,"Optimal cost: ~L", v]) ,
makelist(ADD_NOTATIONT(
 [FORMATT,"Derivative operator with respect to ~L: ~L.", xx[i], D[i]]), i, 1, n1),
execute(
[REPORTM ,
  [TITLET,  "A ~N control problem", alea],
  [AUTHORT,["Pandore", "Inria"]],
  [ABSTRACTM,
     [FORMATT, "We consider a ~N control problem. The optimal cost satisfies a Bellman equation, defined
              on $\\Omega = ~N$, which is solved numerically.", alea, domain(n1)]],
  [[SECTIONT, "Optimality conditions"],
     [FORMATT, "The optimal cost ~L satisfies the Bellman equation ~Y. ", v, Fleming],
```

```
        [TEX, minim(uu,auv+cf)-v = 0, eq],
        [FORMATT, "with"],
        [LISTEXPT, cons(auv=hamiltonian(df,dr,n1,v),dirichlet_homogene(n1,xx,v))],
      [SECTIONT, "Dynamic Programming method"],
        [FORMATT, "Our purpose is to solve the Bellman equation ~Z after discretization ~Y.", eq, Quadrat],
        [SUBSECTIONT, "Discretization:"],
          [FORMATT,"We denote by ~L the i-th space variable discretization step and we define:", \h[\i]],
          [LISTEXPT, translation(n1,xx,v)] ,
          [FORMATT, "We approximate"],
          [ITEMIZEM, [[FORMATT, "~L by ~L", D[\i].v , d1(\i,v)],
                      [FORMATT, "~L by ~L.", (D[\i]^2).v, d2(\i,v)]]],
          [FORMATT, "The discretized Bellman equation is obtained by replacing the operator $A$
                      by the operator ~L in equation ~Z with", A[\h], eq],
          [DECOUPEM, funmake(A[\h],uu).v = factorized_hamiltonian(df,dr,n1,v)],
        [SUBSECTIONT, "Probabilistic Interpretation of the discretized equation :"],
          [FORMATT, "The discretization of the Bellman equation can be interpreted as a control problem
                      of Markov chain. The transition matrix is given by:~K", matrice(df,dr,n1,xx,v)]]]))$

/*generation of the range of the state variables*/
domain(n):=apply(concat,cons("[0,1]",makelist("\\times[0,1]",i,1,n-1)))$
/*definition of the finite derivatives */
d1(i,v):=(S[i].v-S[i]^^(-1).v)/(2*\h[i])$
d2(i,v):=(S[i]^^(-1).v+S[i].v-2*v)/\h[i]^2$
/* generation of the limit conditions */
dirichlet_homogene(n,xx,v):= apply(append,map(lambda([x],
                      makelist(subst(x,xx[i], funmake(v,xx)=0),i,1,n)),[0,1]))$
/*definition of the translation operators */
translation(n,xx,v):=makelist(s[i](v)=funmake(v,subst(xx[i]+\h[i],xx[i],xx)),i,1,n)$
/*definition of the hamiltonian */
hamiltonian(aa,b,n,v):= sum(aa[i].(D[i]^2).v+b[i].D[i].v,i,1,n)$
/*generation of the discretized hamiltonian*/
discrete_hamiltonian(a,b,n,v):= expand(sum(a[i]*d2(i,v)+b[i]*d1(i,v),i,1,n))$
/*generation of the factorized and ordered discretized equation*/
factorized_hamiltonian(a,b,n,v):= apply("+",map(lambda([x],coeff(discrete_hamiltonian(a,b,n,v),x).x),
                              append([v],makelist(s[i].v,i,1,n),makelist(s[i]^^(-1).v,i,1,n))))$
/*generation of the matrix*/
matrice(a,b,n,xx,v):= apply(matrix, append(
        [["Initial-pt","Final-pt","Transition-probability"],[xx,xx,0]],
makelist(lign(discrete_hamiltonian(a,b,n,v),i,\h[i],s[i].v,n,xx,v),i,1,n),
        makelist(lign(discrete_hamiltonian(a,b,n,v),i,-\h[i],s[i]^^(-1).v,n,xx,v),i,1,n)))$
/*generation of the row of the matrix*/
lign(l,i,s1,s2,n,xx,v):=[xx, subst(xx[i]+s1,xx[i],xx), ratsimp(coeff(l,s2)/-coeff(l,v))]$
```

The call of the function STOCH_CONTROL with the arguments
stoch_control(2,2,v,[1,1],[u[1],u[2]],$u[1]^2 + u[2]^2$,"stochastic")
generates LaTeX code displayed in appendix A. The resulting article is provided in appendix B

4 Breef reference manual of MACROTEX

4.1 The Macsyma-LATEX translator

- TEX_EMACS(< *function* >,< *arg1* >,..,< *argn* >)
  applies < *function* > to< *args* > and outputs the result in a default emacs buffer.

- TEX(<*Macsyma-expression*>,{< *label* >,< *stream* >})
  translates a Macsyma expression in Latex code in Math mode.
  < *label* > is optional. If present, it is a Macsyma atom or string which will be used later to reference the equation (unless the value of < *label* > is false).
  < *stream* > is optional. If present it is the stream where the output is directed. If < *stream* > is bound to false there is no display and the function returns a string.

- LISTEXPT(<*Macsyma-list-of-expressions*>,{< *justification* >,< *label* >,< *stream* >})
  displays a system of Macsyma expressions.
  < *justification* > ::= "l" | "r" | "c"
  < *justification* > specifies the alignment of the expressions (flush left ,centered, flush right).
  < *label* > and < *stream* > (see function tex) .

- LISTEQUAT([< *MLists* >],{< *justification* >,< *label* >,< *stream* >})
  displays a list of equations.

  < *Mlists* >::= [< *left* >,< *right* >]|[< *left* >,*right*],< *Mlists* >
  < *left* >::=< *Macsyma-expression* >
  < *right* >::=< *Macsyma-expression* >

  the [< *left* >,< *right* >] lists describe the equations < *left* >=< *right* >.
  < *justification* > is a Macsyma string of the form "xyz" with x, y, z among the characters l ,c ,r .
  The equations are displayed one above the other and justified according to the string justification (which contains three characters, one for each part of the equation).
  < *label* > and < *stream* > (see function TEX).

- DECOUPET(< *arg1* >,< *arg2* >,< *operator* >,{< *label* >,< *stream* >})
  displays a "large" equation with the subparts provided by the user.

  < *arg1* >:=< *Macsyma-expression* > | < *Macsyma-list* >
  < *arg2* >:=< *Macsyma-expression* > | < *Macsyma-list* >
  < *operator* >:=< *Macsyma-string* > | < *List-of-strings* >

  if arg1 is a Macsyma expression, arg2 must be a list :

```
decoupet(arg1,[arg2,arg3,arg4],["op1","op2"])
        ==> arg1 = arg2
                op1 arg3
                op2 arg4
decoupet(arg1,[arg2,arg3,arg4],"op")
        <==>decoupe(arg1,[arg2,arg3,···],["op","op",...])
```

if arg1 is a list then arg2 must be an expression or an empty list ([])

```
decoupet([arg1,arg2,arg3],arg4,["op1","op2"])
          ==> arg1
                op1 arg2
                op2 arg3 = arg4
decoupet([arg1,arg2,arg3],[],["op1","op2"])
          ==> arg1
                op1 arg2
                op2 arg3
```

- EXPTXT(<*Macsyma-expression*>, < *stream* >)
  translates a Macsyma expression in LaTeX code as an in-text mathematical formula.
- DECOUPEM(< *Macsyma-expression*>, < *label* >, < *integer* >)
  is a macro function provided to cut automatically long expressions.
  < *label* > is optional (see function TEX),
  < *integer* > is an integer of default value 150.
  A Macsyma expression of "length" larger than < *integer* > will be cut.
  <*Macsyma-expression*> can be any Macsyma expression. If it is an equation, just the right hand side of it will be cut.
  This macro function does not print directly the generated LaTeX code but returns a Macsyma List containing LISTEXPT or LISTEQUAT. (see the report generator part for the execution of Macro instructions).
- FORMATT(<*Macsyma-string*>, < *arg1* >, ..., < *argn* >)
  is an extension of the *format* function in Macsyma environment.
  <*Macsyma-string*> is a string describing a format directive . The following directives where added :
  - ~K and ~L display their arguments using respectively the functions TEX and EXPTXT.
  - ~N translates its argument in LaTeX syntax but not in math mode.
  - ~Y and ~Z are the directives to insert respectively a quotation or a reference.
- TEX_TRANSLATION(<*function-name*>, < *string* >, {<*operator-list*>})
  defines a specific display in LaTeX code for the function <*function-name*>,
  < *string* > is a string describing a format specification which is to be applied on the arguments of the function <*function-name*>to provide the translation,

<*operator-list*>::=< *operator* > |[< *operator1* >, ..., < *operatorn* >]
< *operator* >::=<*Macsyma-function-name-or-lambda*>

<*operator-list*> is either a list of Macsyma functions to be applied on the arguments of <*function-name*> before the format execution, or a function name or a lambda function to be applied to all the arguments of <*function-name*>.(see the examples for more details) .

4.2 The generator of LaTeX reports

4.2.1 The elementary instructions

1. the instructions which deal with the global variables:

   - The function CONTEXTT sets the global variables list_references and list_notations to the empty lists, assigns the variable "planflag" to true in order to have the table of contents automatically generated in the report, kills the labels, and assigns values to the documentstyle parameters used by the macro STYLEM: "styleflag" is bound to article, "sizeflag" to 11, "heightflag" to 660, "widthflag" to 470, "topmarginflag" to -27, "oddsidemarginflag" and "evensidemarginflag" to 0, "marginparwidthflag" to 60.
   - The function RESET_BIBLIOMT sets list_references to the empty list.
   - The functions ADD_REFERENCET and ADD_NOTATIONT add a reference or a notation respectively to the Macsyma lists "list_references" and "list_notations".
     A reference has the following form: [key, source], where key and source are Macsyma strings.
     A notation is a MACROTEX instruction.

2. the instructions corresponding to LaTeX commands

   - the commands for the declaration of the documentstyle parameters.
     [DOCUMENTSTYLET,< *Mstring* >], [HEIGHTT, < *number* >] ...
   - LABELT, REFT, CITET are used as follow:

     [LABELT,< *Mstring* >]
     [REFT, < *Mstring* >]
     [CITET, < *LMstring* >]
     < *LMstring* > ::=< *Mstring* > | < *Mstring* > , < *LMstring* >
     < *Mstring* > ::= < *Macsyma-string*>

     The functions REFT and CITET provide respectively the same facilities as the ~Z and the ~Y directive in the function FORMATT.
   - [TITLET, *format-string, format-args*] generates a title command.
   - [SECTIONT, *format-string, format-args*] generates the LaTeX sectionning command and produces the tables of contents entries.
     SUBSECTIONT and SUBSUBSECTIONT are similar.
   - [AUTHORT,[[< *Mstring* >,< *Mstring* >],[< *Mstring* >,< *Mstring* >], ...]]
     generates the names and addresses of the authors. The first element of each list is the name of the author, and the second element is his address.

4.2.2 The macro instructions

- The macro [ABSTRACTM, < *body* >] generates an abstract.
  < *body* > is a Macsyma list of MACROTEX instructions.

- the macro [CITEM, [*key1,source1*], [*key2, source2*], ...] calls CITET on *key1, key2*, ... and pushes the references *source1, source2*, ... into "list_references".

- the macro BIBLIOM generates the bibliography using the global variable list_references which is build by the means of the macro CITEM or by the elementary instruction ADD_REFERENCET.
- the macros [ITEMIZEM, < *body* >] and [ENUMERATEM, < *body* >] provide the itemize and enumerate list-making environments. Their arguments are Macsyma-lists of MACROTEX instructions.
- The macro STYLEM provides the declaration of the documentstyle parameters.

  [STYLEM, article,11,660,470,-27,0,0,60] expands into

  [[DOCUMENTSTYLET, article, 11],
  [HEIGHTT,660],
  [WIDTHT,470],
  [TOPMARGINT,-27],
  [ODDSIDEMARGINT,0],
  [EVENSIDEMARGINT,0],
  [MARGINPARWIDTHT, 60]]

- The macro NOTATIONM generates a section of notations, using the global variable "list_notation". Each notation is begun with an "item" command. When this list is not empty, the macro [NOTATIONM] expands into

  [[SECTIONT, Notation],
  [ITEMIZEM, list_notation]]

- The macro REPORTM takes for arguments a title , a list of authors with their addresses, an abstract, and the body of the report which is the list of sections. To generate a report, we thus have to build a Macsyma list compound by elementary and macro-statements according to a specific grammar defined below. In the following "*e.i*" stands for elementary-instruction and "*m.i*" for macro-instruction.

[REPORTM,< *title* >, < *author* >, < *abstract* >,< *body* >]
< *title* > ::=[TITLET, < *Mstring* >, <*list-M*>]
< *author* >::=[AUTHORT,[[< *Mstring* >,< *Mstring* >],
[< *Mstring* >,< *Mstring* >],...]]
< *abstract* >::=[ABSTRACTM, < *body* >]
< *body* > ::= [< *e.i* >,< *body* >] | [< *m.i* >,< *body* >] | < *e.i* > | < *m.i* >
< *e.i* > ::= [< *keyT* >, <*list-M*>]
< *m.i* > ::= [< *keyM* >, <*list-M*>]
< *keyT* > ::= TEX| EXPT| FORMATT| LISTEXPT | LISTEQUAT| DECOUPET| AUTHORT| CITET| LABELT| REFT| ADD_NOTATIONT| ADD_REFERENCET| TITLET| SECTIONT| SUBSECTIONT | SUBSUBSECTIONT
< *keyM* > ::= DECOUPEM| ITEMIZEM| ENUMERATEM| CITEM

<*list-M*> ::= <*Macsyma-expression*> | <*Macsyma-expression*>,<*list-M*>
< *Mstring* > ::= <*Macsyma-string*>

RAPPORTM first applies the macro STYLEM with the values of the parameters assigned in the function CONTEXTT. It then outputs the title, the list of authors and adresses, the abstract, the table of contents (if the variable "planflag" is set to true), the section of the notations, the body of the report, and the bibliography. The notation section and the bibliography appear in the report if the lists "list_references" and "list_notations" are not empty.

The translation into LaTeX of a MACROTEX program is done by calling the function EXECUTE on it. The report is thus obtained by calling EXECUTE on the Macsyma list [RAPPORTM, args].

5 Session of examples of use of MACROTEX

We list here some examples of the MACROTEX functions. For some of them we show the generated LaTeX code in "verbatim mode" when this code cannot be inserted without confusion in our article. For some others the LaTeX code is not shown, we only see the result of its compilation (i.e the final LaTeX display).

5.1 Translations functions from Macsyma to LaTeX

```
(C9) execute([[print,"Example of a labelled  equation :"]
            [tex,sin(x)+x/y-integrate(f(x),x,0,1),label2]]);
```

Example of a labelled equation :

$$\frac{X}{Y} - \int_0^1 F(X)\,dX + \sin(X) \tag{2}$$

```
(D9)    DONE
```

```
(C10) execute([[print,"Example of equation with a label:"],
        [tex,sum(x(i),i,1,n2),"sumlab"],
        [formatt,"The equation is referenced by:~Z", "sumlab"]]);
```

Example of equation with a label:

$$\sum_{I=1}^{N2} X(I) \tag{3}$$

The equation is referenced by:(3)

```
(D10)    DONE
```

```
(C11) execute([[print,"Example of a in-text formula :"],
        [exptxt,sum(x(i),i,a,b)]]);
```

Example of a in-text formula : $\sum_{I=A}^{B} X(I)$

```
(D11)    DONE
```

```
(C12) execute([[formatt,"An example with format :~%
              We consider a ~N control problem. The dynamic of the system is ~%
              described  by  a  diffusion process defined on $\\Omega = ~N $ ~
              by ~K","stochastic",[0,1],\dx[\t]=g.\dt+s.\dw[\t]]]);
```

An example with format : We consider a stochastic control problem. The dynamic of the system is described by a diffusion process defined on $\Omega = [0,1]$ by

$$dX_t = S\,dW_t + GdT$$

```
(D12)    DONE
```

```
(C13) execute([[print,"Display of a system of centered expressions (default value)"],
        [listexpt,[sum(x(i)^n3,i,n1,n2),integrate(f(x),x,arg1,arg2)]]]);
```

Display of a system of centered expressions (default value)

$$\sum_{I=N1}^{N2} X(I)^{N3}$$

$$\int_{ARG1}^{ARG2} F(X)\,dX$$

```
(D13)    DONE
```

```
(C14) execute([[print,"Same thing but the system is numbered and flush left ."],
        [listexpt,[sum(x(i)^n3,i,n1,n2),integrate(f(x),x,arg1,arg2)],"1","explab"]]);
```

Same thing but the system is numbered and flush left .

$$\begin{array}{l} \sum_{I=N1}^{N2} X(I)^{N3} \\ \int_{ARG1}^{ARG2} F(X)\,dX \end{array} \qquad (4)$$

```
(D14)    DONE
```

```
(C15) execute([[formatt,"In the case of a system of equations, ~%~
                  each member can be centered. ~%~%~
                 First example in centered-centered-centered :"],
       [listequat,[[sum(x(i)^n3,i,n1,n2),sin(x)+log(x)],[a2+f(x),g(x)]],"ccc","label1"],
       [print," Second example in ''lcr'' with a label"],
       [listequat,[[a1,sum(x(i)^n3,i,n1,n2)+sin(x)],
              [sum(x(i)^n3,i,n1,n2),g(x)]],"lcr","label"]]);
```

In the case of a system of equations, each member can be centered.
First example in centered-centered-centered :

$$\sum_{I=N1}^{N2} X(I)^{N3} = \sin(X) + LOG(X)$$
$$F(X) + A2 = G(X) \tag{5}$$

Second example in "lcr" with a label

$$A1 = \sin(X) + \sum_{I=N1}^{N2} X(I)^{N3}$$
$$\sum_{I=N1}^{N2} X(I)^{N3} = G(X) \tag{6}$$

```
(D15)   DONE

(C16) execute([[formatt,"The case of an expression to be cut because too long to fit on one line -
              and the subparts are provided by the user."],
        [decoupet,f(x),[cos(y),1.234*integrate(g(x),x,0,1),sin(x)],["+","-"]],
        [print,"Second example "],
        [decoupet,[cos(y),1.234*integrate(g(x),x,0,1),sin(x)],f(x),["+","-"]],
        [print,"Third example "],
        [decoupet,[cos(y),1.234*integrate(g(x),x,0,1),sin(x)],[],["+","-"]]]);
```

The case of an expression to be cut because too long to fit on one line and the subparts are provided by the user.

$$\begin{aligned} F(X) &= \cos(Y) \\ &+ 1.234 \int_0^1 G(X)\,dX \\ &- \sin(X) \end{aligned}$$

Second example

$$\begin{aligned} &\cos(Y) \\ &+1.234 \int_0^1 G(X)\,dX \\ &-\sin(X) \qquad = F(X) \end{aligned}$$

Third example

$$\begin{aligned} &\cos(Y) \\ &+1.234 \int_0^1 G(X)\,dX \\ &-\sin(X) \end{aligned}$$

```
(D16)    DONE

(C17) /*application examples of macro-instructions
  which are expanded into a list of elementary instructions */
execute([[formatt,"The macro DECOUPEM  cuts a long equation into sub-equations. ~%~% ~
          First example :the r.h.s. is cut"],
         [decoupem,'exp=cos(y)+1.234*integrate(g(x),x,0,1)-sin(x)+(x^2-y[1]+c)/x+y[1]-c,
                 false,100]]);
```

The macro DECOUPEM cuts a long equation into sub-equations.

First example :the r.h.s. is cut

$$EXP = \cos(Y) + 1.234 \int_0^1 G(X)\,dX - \sin(X) + \frac{(X^2 + C - Y_1)}{X} - C + Y_1$$

```
(D17)    DONE

(C18) execute([[formatt,"Second example with another value of the linel argument:"],
         [decoupem,'exp=cos(y)+1.234*integrate(g(x),x,0,1)-sin(x)+(x^2-y[1]+c)/x+y[1]-c,
                 false,50]]);
```

Second example with another value of the linel argument:

$$EXP = E5 + E4 + E3 + E2 + E1 - C$$

$$E1 = \cos(Y)$$

$$E2 = 1.234 \int_0^1 G(X)\,dX$$

$$E3 = -\sin(X)$$

$$E4 = \frac{(X^2 + C - Y_1)}{X}$$

$$E5 = Y_1$$

```
(D22)    DONE

(C23) execute([[formatt,"Third example: the case of an equation with a large matrix"],
         [decoupem,ev(mat=matrix([a,d,f,g],[a,a,a,a],[a,a,a,a],[b,b,b,b]),
         a=sum(sin(x(i))*y(i),i,jk,lp),b=sin(x)+cos(x)),false,50]]);
```

Third example: the case of an equation with a large matrix

$$MAT = \begin{pmatrix} \sum_{I=JK}^{LP} Y(I)\sin(X(I)) & D & F & G \\ E6 & E6 & E6 & E6 \\ E6 & E6 & E6 & E6 \\ E7 & E7 & E7 & E7 \end{pmatrix}$$

$$E6 = \sum_{I=JK}^{LP} Y(I)\sin(X(I))$$

$$E7 = \sin(X) + \cos(X)$$

```
(D24)    DONE

(C25) execute([[formatt,"Cut of a long expression"],
          [decoupem,ev(matrix([a,b],[c,d]),c=sin(x)+cos(x),
  a=matrix([sin(x),cos(x)],[cos(x),sin(x)]),d=sum(y(i),i,1,n2)),false,40]]);
```

Cut of a long expression

$$\begin{pmatrix} E8 & B \\ E7 & E9 \end{pmatrix}$$

$$E8 = \begin{pmatrix} \sin(X) & \cos(X) \\ \cos(X) & \sin(X) \end{pmatrix}$$

$$E7 = \sin(X) + \cos(X)$$

$$E9 = \sum_{I=1}^{N2} Y(I)$$

```
(D26)    DONE

(C27) execute([[formatt,"Thiner cut of an expression: "],
          [decoupem,ev(matrix([a,b],[c,d]),c=sin(x)+cos(x),
  a=matrix([sin(x),cos(x)],[cos(x),sin(x)]),d=sum(y(i),i,1,n2)),false,20]]);
```

Thiner cut of an expression:

$$\begin{pmatrix} E8 & B \\ E7 & E9 \end{pmatrix}$$

$$E8 = \begin{pmatrix} E10 & E11 \\ E11 & E10 \end{pmatrix}$$

$$E10 = \sin(X)$$

$$E11 = \cos(X)$$

$$E7 = \sin(X) + \cos(X)$$

$$E9 = \sum_{I=1}^{N2} E12$$

$$E12 = Y(I)$$

```
(D29)   DONE
```

```
(C30) execute([[formatt,"How to create specific translations at Macsyma Level: ~%~% ~
                First example if the directive N is available: ~%~%"],
        [tex_translation,diff2,"~N^{(~N)}(~N)"],
  [formatt,"A new display for the derivative :~K",diff2(f,N,x)]]);
```

How to create specific translations at Macsyma Level:
First example if the directive N is available:
A new display for the derivative :

$$F^{(N)}(X)$$

```
(D30)   DONE
```

```
(C31) execute([[formatt,"The same example without the N directive. ~%~%~
                The function to apply is given for each argument of the function ~
                diff2. In this example [stex,stex,stex] can be replaced by stex ~
                because the same operator is applied to all the arguments of diff2."],
        [tex_translation,diff2,"~A^{(~A)}(~A)",[stex,stex,stex]],
  [tex,diff2(f,N,x)]]);
```

The same example without the N directive.

The function to apply is given for each argument of the function diff2. In this example [stex,stex,stex] can be replaced by stex because the same operator is applied to all the arguments of diff2.

$$F^{(N)}(X)$$

```
(D31)   DONE
```

```
(C32) execute([[sectiont,"The instructions to generate a report "]])$
\section[The instructions to generate a report ]{The instructions to generate a report }

(C33) /*application examples of elementary instructions */
execute([[documentstylet,"article", 11],
        [heightt,660],
        [widtht ,470],
        [titlet, "a ~N stochastic control problem", "discounted" ],
        [sectiont,"Notation"],
        [labelt, name],
        [reft, name],
        [citet,aut1,aut2]]);
\documentstyle[11pt]{article}
\textheight=660pt
\textwidth=470pt
\title{a discounted stochastic control problem}
\section[Notation]{Notation}
\label{NAME}
\ref{NAME}
\cite{AUT1,AUT2}
(D33)   DONE

(C34) execute([itemizem, [[[formatt, ''variables''],
                          [enumeratem,[[formatt,"a"],[formatt,"b"]]]],
                                     [formatt, ''constants'']]]);
```

- variables
    1. a
    2. b
- constants

```
(D34)   DONE

(C35) execute([stylem,"article",11, 660, 470, -27, 0, 0, 60])$

\documentstyle[11pt]{article}
\textheight=660pt
\textwidth=470pt
\topmargin=-27pt
\oddsidemargin=0pt
\evensidemargin=0pt
\marginparwidth=60pt

(C36) execute([titlet, "a ~N stochastic control problem", "discounted"])$
```

```
\title{a discounted stochastic control problem}

(C37) execute([authort ,["Pandore", "Inria"], ["Dupont", "Universite Paris"]])$

\author{     Pandore \\ Inria \and    Dupont \\ Universite Paris   }

(C38) execute([abstractm,[[formatt, "We consider a ~N control problem.~
                    The dynamic of the system is ~
                    described  by  a diffusion process defined on $ \\Omega = ~N.$",
                    "stochastic",[0,1]],
                      .     [print, "The optimal cost satisfies a Bellman equation. " ]]]) $

\begin{abstract}
We consider a stochastic control problem. The dynamic of the system is
described  by  a diffusion process defined on $ \Omega = \left[ 0,1 \right].$
The optimal cost satisfies a Bellman equation.
 \end{abstract}

(C39) execute([[add_notationt, [formatt,"State variables: ~L ", [x1,x2]],
                                [formatt,"Control variables: ~L ",[u1,u2]]],
        [notationm]])$

\section[Notation]{Notation}
\begin{itemize}
\item Variables d'\'etat:  $X1,X2 \; $ ,
\item Variables de commande:  $U1,U2 \; $ ,
\item Temps:  $T \; $ ,
\item State variables:  $\left[ X1,X2 \right] \; $
\item Control variables:  $\left[ U1,U2 \right] \; $
\end{itemize}

(C40) execute([[reset_bibliomt],
          [add_referencet,[aut1," The reference for the aut1 paper"],
                          [aut2," The reference for the aut2 paper"]],
          [formatt,"the article of ~N can be found in the bibliography under the number ~Y",
            aut1,aut1],
          [formatt,"the function citet provides the same facilities as the Y directive in ~
            the function formatt."],
          [citet,aut1,aut2],
          [bibliom]])$

the article of AUT1 can be found in the bibliography under the number \cite{AUT1}
the function citet provides the same facilities as the Y directive in the function formatt.
\cite{AUT1,AUT2}
 \begin{thebibliography}{3}
\bibitem{AUT1} The reference for the aut1 paper
\bibitem{AUT2} The reference for the aut2 paper
```

```
\end{thebibliography}
```

6 Conclusion

MACROTEX is one component of the system PANDORE which is an expert system in stochastic control[5]. Let us mention another component of PANDORE called MACROFORT which is similar to MACROTEX and which is devoted to the automatic generation of Fortran programs at Macsyma level [4].

References

[1] MACSYMA Reference Manual. Version 12 (1986). Symbolics, Inc.

[2] STEELE,Guy. COMMON LISP. The language. Digital Press.

[3] LAMPORT,Leslie. LATEX. A Document Preparation System. Addison-Wesley Publishing Company. v

[4] CHANCELIER,J.P., GOMEZ,C., QUADRAT, J.P. (October 1987) MACROFORT: a FORTRAN code generator in MACSYMA. Macsyma Newsletter, Symbolics Inc.

[5] CHANCELIER,J.P., GOMEZ,C., QUADRAT,J.P. AND SULEM,A. *Automatic Study in Stochastic Control.* Volume 10 of *IMA Volumes in Mathematics and its Applications*, Springer Verlag, 1987. Proceedings of the Workshop on Stochastic Differential Systems, Stochastic Control Theory and Applications, Minneapolis, Minnesota, June 86.

A Appendix : the Latex code generated

```
\documentstyle[11pt]{article}
\textheight=660pt
\textwidth=470pt
\topmargin=-27pt
\oddsidemargin=0pt
\evensidemargin=0pt
\marginparwidth=60pt

\title{A stochastic control problem}
\author{      Pandore \\ Inria    }
\begin{document}
\maketitle
\begin{abstract}
We consider a stochastic control problem. The optimal cost satisfies a Bellman
equation, defined on $\Omega = [0,1]\times[0,1]$, which is solved numerically.
 \end{abstract}

\tableofcontents

\section[Notation]{Notation}
\begin{itemize}
\item State variables:  $\left[ x_{1},x_{2} \right] \; $
\item Control variables:  $\left[ u_{1},u_{2} \right] \; $
\item Optimal cost:  $V \; $
\item Derivative operator with respect to  $x_{1} \; $ :  $D_{1} \; $ .
\item Derivative operator with respect to  $x_{2} \; $ :  $D_{2} \; $ .
\end{itemize}

\section[Optimality conditions]{Optimality conditions}
The optimal cost  $V \; $  satisfies the Bellman equation \cite{FLEMING} .
\begin{equation}
 \min_{u_{1},u_{2}} \left(  A(u_{1},u_{2})   V +u_{2}^{2}+u_{1}^{2}\right)- V  = 0
\label{EQ}
\end{equation}with
\[
\begin{array}{c}
    {\displaystyle  A(u_{1},u_{2})   V  = u_{2}   D_{2}   V +
        D_{2}^{2}   V + u_{1}   D_{1}   V + D_{1}^{2}   V }\\ \\
    {\displaystyle V(0,x_{2}) = 0}\\ \\
    {\displaystyle V(x_{1},0) = 0}\\ \\
    {\displaystyle V(1,x_{2}) = 0}\\ \\
    {\displaystyle V(x_{1},1) = 0}
 \end{array}
\]

\section[Dynamic Programming method]{Dynamic Programming method}
```

```
Our purpose is to solve the Bellman equation (\ref{EQ}) after discretization
\cite{QUADRAT} .
\subsection[Discretization:]{Discretization:}
We denote by  $h_{i} \; $  the i-th space variable discretization step and
we define:
\[
\begin{array}{c}
    {\displaystyle S_{1}( V ) = V(x_{1}+h_{1},x_{2})}\\ \\
    {\displaystyle S_{2}( V ) = V(x_{1},x_{2}+h_{2})}
 \end{array}
\]
We approximate\begin{itemize}
\item  $ D_{i}   V  \; $  by
  ${ {\left( S_{i}   V -  S_{i}^{-1}   V  \right)}  \over  2   h_{i} } \; $
\item  $ D_{i}^{2}   V  \; $  by
$ h_{i}^{-2}   {\left(- 2   V + S_{i}^{-1}   V + S_{i}   V \right)}   \; $ .
\end{itemize}
The discretized Bellman equation is obtained by replacing the operator $A$
 by the operator  $A_{h} \; $  in equation (\ref{EQ}) with
\[
\begin{array}{ccc}
    {\displaystyle  A_{h}( u_{1} , u_{2} )   V }&{\displaystyle =}&
     {\displaystyle E5+E4+E3+E2+E1}\\ \\
      {\displaystyle E1}&{\displaystyle =}&
       {\displaystyle  {\left({ 0.5   u_{2}  \over  h_{2} }+
       h_{2}^{-2}\right)}   S_{2}   V }\\ \\
      {\displaystyle E2}&{\displaystyle =}&
       {\displaystyle  {\left(h_{2}^{-2}-
        { 0.5   u_{2}  \over  h_{2} }\right)}   S_{2}^{-1}   V }\\ \\
      {\displaystyle E3}&{\displaystyle =}&
       {\displaystyle  {\left(- 2   h_{1}^{-2}
        - 2   h_{2}^{-2} \right)}   V }\\ \\
      {\displaystyle E4}&{\displaystyle =}&
       {\displaystyle  {\left({ 0.5   u_{1}  \over  h_{1} }+
       h_{1}^{-2}\right)}   S_{1}   V }\\ \\
      {\displaystyle E5}&{\displaystyle =}&
       {\displaystyle  {\left(h_{1}^{-2}-
       { 0.5   u_{1}  \over  h_{1} }\right)}   S_{1}^{-1}   V }
 \end{array}
\]

\subsection[Probabilistic Interpretation of the discretized equation :]
           {Probabilistic Interpretation of the discretized equation :}
The discretization of the Bellman equation can be interpreted as a control problem
of Markov chain. The transition matrix is given by:
\[
 \left( \begin{array}{ccc}
    {\displaystyle Initial-pt}&{\displaystyle Final-pt}&
```

```
    {\displaystyle Transition-probability}\\ \\
  {\displaystyle \left[ x_{1},x_{2} \right]}&
    {\displaystyle \left[ x_{1},x_{2} \right]}&{\displaystyle 0}\\ \\
  {\displaystyle \left[ x_{1},x_{2} \right]}&
    {\displaystyle \left[ x_{1}+h_{1},x_{2} \right]}&
    {\displaystyle { {\left( h_{1}  u_{1} +2\right)}  h_{2}^{2}
       \over  {\left( 4  h_{2}^{2} + 4  h_{1}^{2} \right)} }}\\ \\
  {\displaystyle \left[ x_{1},x_{2} \right]}&
    {\displaystyle \left[ x_{1},x_{2}+h_{2} \right]}&
    {\displaystyle { {\left( h_{1}^{2}  h_{2}  u_{2} + 2  h_{1}^{2} \right)}
       \over  {\left( 4  h_{2}^{2} + 4  h_{1}^{2} \right)} }}\\ \\
  {\displaystyle \left[ x_{1},x_{2} \right]}&
    {\displaystyle \left[ x_{1}- h_{1} ,x_{2} \right]}&
    {\displaystyle -{ {\left( h_{1}  u_{1} -2\right)}  h_{2}^{2}
       \over  {\left( 4  h_{2}^{2} + 4  h_{1}^{2} \right)} }}\\ \\
  {\displaystyle \left[ x_{1},x_{2} \right]}&
    {\displaystyle \left[ x_{1},x_{2}- h_{2}  \right]}&
    {\displaystyle -{ {\left( h_{1}^{2}  h_{2}  u_{2} - 2  h_{1}^{2} \right)}
       \over  {\left( 4  h_{2}^{2} + 4  h_{1}^{2} \right)} }}
 \end{array} \right)
\]
 \begin{thebibliography}{3}
\bibitem{FLEMING}FLEMING,W.H.-RISHEL,R.(1975).
   Deterministic and Stochastic Optimal Control. Springer verlag, New York.
\bibitem{QUADRAT}QUADRAT,J.P.(1980).
   Existence de solution et algorithme de r\'esolutions num\'eriques de probl\'emes
   stochastiques d\'eg\'en\'er\'es ou non. SIAM Journal of Control.
 \end{thebibliography}
\end{document}
```

B Appendix : The generated article

A stochastic control problem

Pandore
Inria

November 27, 1987

Abstract

We consider a stochastic control problem. The optimal cost satisfies a Bellman equation, defined on $\Omega = [0,1] \times [0,1]$, which is solved numerically.

Contents

1 Notation

- State variables: $[x_1, x_2]$
- Control variables: $[u_1, u_2]$
- Optimal cost: V
- Derivative operator with respect to x_1 : D_1 .
- Derivative operator with respect to x_2 : D_2 .

2 Optimality conditions

The optimal cost V satisfies the Bellman equation [1] .

$$\min_{u_1,u_2} \left(A(u_1,u_2)V + u_2^2 + u_1^2 \right) - V = 0 \tag{1}$$

with

$$A(u_1,u_2)V = u_2 D_2 V + D_2^2 V + u_1 D_1 V + D_1^2 V$$

$$V(0,x_2) = 0$$

$$V(x_1,0) = 0$$

$$V(1,x_2) = 0$$

$$V(x_1,1) = 0$$

3 Dynamic Programming method

Our purpose is to solve the Bellman equation (1) after discretization [2] .

3.1 Discretization:

We denote by h_i the i-th space variable discretization step and we define:

$$S_1(V) = V(x_1 + h_1, x_2)$$

$$S_2(V) = V(x_1, x_2 + h_2)$$

We approximate

- $D_i V$ by $\frac{(S_i V - S_i^{-1} V)}{2h_i}$
- $D_i^2 V$ by $h_i^{-2}\left(-2V + S_i^{-1} V + S_i V\right)$.

The discretized Bellman equation is obtained by replacing the operator A by the operator A_h in equation (1) with

$$\begin{aligned}
A_h(u_1,u_2)V &= E5 + E4 + E3 + E2 + E1 \\
E1 &= \left(\frac{0.5u_2}{h_2} + h_2^{-2}\right) S_2 V \\
E2 &= \left(h_2^{-2} - \frac{0.5u_2}{h_2}\right) S_2^{-1} V \\
E3 &= \left(-2h_1^{-2} - 2h_2^{-2}\right) V \\
E4 &= \left(\frac{0.5u_1}{h_1} + h_1^{-2}\right) S_1 V \\
E5 &= \left(h_1^{-2} - \frac{0.5u_1}{h_1}\right) S_1^{-1} V
\end{aligned}$$

3.2 Probabilistic Interpretation of the discretized equation :

The discretization of the Bellman equation can be interpreted as a control problem of Markov chain. The transition matrix is given by:

$$\begin{pmatrix} Initial-pt & Final-pt & Transition-probability \\ [x_1,x_2] & [x_1,x_2] & 0 \\ [x_1,x_2] & [x_1+h_1,x_2] & \frac{(h_1u_1+2)h_2^2}{(4h_2^2+4h_1^2)} \\ [x_1,x_2] & [x_1,x_2+h_2] & \frac{(h_1^2h_2u_2+2h_1^2)}{(4h_2^2+4h_1^2)} \\ [x_1,x_2] & [x_1-h_1,x_2] & -\frac{(h_1u_1-2)h_2^2}{(4h_2^2+4h_1^2)} \\ [x_1,x_2] & [x_1,x_2-h_2] & -\frac{(h_1^2h_2u_2-2h_1^2)}{(4h_2^2+4h_1^2)} \end{pmatrix}$$

References

[1] FLEMING,W.H.-RISHEL,R.(1975). Deterministic and Stochastic Optimal Control. Springer verlag, New York.

[2] QUADRAT,J.P.(1980). Existence de solution et algorithme de résolutions numériques de problèmes stochastiques dégénérés ou non. SIAM Journal of Control.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique