Transferring a symbolic polynomial expression from Mathematica to Matlab

A. Bret

ETSI Industriales, Universidad Castilla-La Mancha, 13071 Ciudad Real, Spain

Abstract

A Mathematica Notebook is presented which allows for the transfer or any kind of polynomial expression to Matlab. The output is formatted in such a way that Matlab routines such as “Root” can be readily implemented. Once the Notebook has been executed, only one copy-paste operation in necessary.

1 Introduction

Transferring a symbolic Mathematica expression to Matlab is a recurring issue one for anyone using these two programs. On the one hand, the Mathematica software is perfect to work out analytical calculations and derive involved symbolic formulas. On the other hand, Matlab offers a quasi-infinite and very flexible environment for graphical, numerical or statistical treatment. The perspective of working in both environment to solve a given problem is thus very attractive. As long as the equations involved remain simple, an expression derived in Mathematica can be straightforward reproduced in Matlab. Problems arise for long and intricate formulas which cannot be easily copy-pasted. The transfer of a polynomial poses an additional problem: if one is willing to take advantage of the Matlab polynomial routines, the very definition of the function needs to be adapted because Matlab defines a polynomial from its coefficients.

The Mathematica Notebook presented in this paper offers a simple solution to this problem. Given any polynomial expression derive in Mathematica, the Notebook generates a text which can be directly copy-pasted into Matlab.

Email address: antoineclaude.bret@uclm.es (A. Bret).
Consider a polynomial of the form,

\[ P(x) = \sum_{i=0}^{n} a_i(\alpha_1 \ldots \alpha_k)x^i, \] (1)

where the coefficients \(a_1 \ldots a_n\) are themselves polynomial expressions of some parameters \(\alpha_1 \ldots \alpha_k\). While both Mathematica and Matlab can deal with this kind of expressions, Matlab requires a different format if functions such as \texttt{roots} are to be exploited. More specifically, the polynomial \(P\) defined by Eq. (1) will be expressed in Matlab under the form of a 1D matrix,

\[ p = [a_n(\alpha_1 \ldots \alpha_k) \ a_{n-1}(\alpha_1 \ldots \alpha_k) \ \ldots \ a_0(\alpha_1 \ldots \alpha_k)] \]

The Notebook which is about to be described generates the format needed by Matlab from the Mathematica expression. The starting point is the definition of the polynomial,

\[ \text{In[1]} := P = a_n(\alpha_1 \ldots \alpha_k)x^n + a_{n-1}(\alpha_1 \ldots \alpha_k)x^{n-1} + \ldots + a_0(\alpha_1 \ldots \alpha_k); \]

Note that the polynomial does not need to be specifically formatted like in Eq. (1). The Notebook can perfectly deal with an expression where the coefficients of the \(x^i\)'s are not factorized. If the polynomial expression accounts for Greek characters (dealt with by Mathematica), they will have to be replaced by some equivalents for proper Matlab treatment. For example, if the Mathematica polynomial accounts for the parameters \(\beta, \gamma\) and \(\omega\), the next line of the Notebook is,

\[ \text{In[2]} := \text{PClear} = P/. \{\beta \rightarrow \text{beta}, \gamma \rightarrow \text{gamma}, \omega \rightarrow \text{omega}\}; \]

We then need the degree of the Polynomial, given by

\[ \text{In[3]} := \text{Deg} = \text{Exponent[PClear, x]}; \]

We now have Mathematica extracting the \(a_i\)'s with,

\[ \text{In[4]} := \text{Coefs} = \text{Table[InputForm[FullSimplify[Coefficient[PClear, x, k]]], \{k, 0, \text{Deg}, 1\}];} \]

Here, the \(k^{th}\) coefficient is first gathered through the \texttt{Coefficient[PClear, x, k]} command. The resulting expression is simplified with \texttt{FullSimplify}. Note that a simple \texttt{Simplify} can be inserted here if the argument is too involved for \texttt{FullSimplify} to execute successfully.
The \textbf{InputForm} command is a key element of the transfer to \textit{Matlab}. \textit{Mathematica} can deal with symbolic fractions or power, but not \textit{Matlab}. For example, the \textit{Mathematica} expression,

\[
\frac{a^2 b^3}{c^4(t - u)}
\]  \hspace{1cm} (2)

must be cast under the form

\[
a^2 b^3/(c^4(t-u)),
\]

for \textit{Matlab} to interpret it. The \textbf{InputForm} command is responsible for the \textit{Matlab} formatting of the polynomial coefficients. Finally, the \textbf{Table} command ensures that the operation is implemented for all the $a_i$’s, and the result is stored in the \textit{Mathematica} array \textbf{Coefs}.

We finally have \textit{Mathematica} displaying the content of the array \textbf{Coefs} under a form which can be copy-pasted to \textit{Matlab},

\begin{verbatim}
In[2]:=Do[Print[“P(”, Deg + 2 - n, “)=”, Coefs[[n]], “;”], {n, Deg + 1}];
\end{verbatim}

The command above thus prints all the coefficients of the polynomial from P(1), coefficient of $x^n$ to P(deg+1), coefficient of $x^0$. The output can then be copy-pasted to a “.m” \textit{Matlab} file. Note that it is here necessary to “Copy As Plain Text” from \textit{Mathematica}, as illustrated in Figure 1.
3 Conclusion - An example

Two files have been attached to this article, as an illustration of the kind of transfer allowed by this technique. The Mathematica Notebook “Bridge to MatMab.nb” introduces a imposing polynomial which arises as the dispersion equation of a quite involved relativistic magnetized beam plasma system [1]. It is a very involved polynomial in terms of the parameters $x, Zx, Zz, \alpha, \gamma_b, \beta, \Omega_B$. The degree of the polynomial in terms of $x$ is 14. The expression results from the determinant of a dielectric tensor, and the coefficients of $x$ are not gathered at all. The execution takes only a few minutes, which can be reduced even further by replacing the FullSimplify by the faster Simplify command in the coefficients calculation.

After copying the output as shown on Figure [1], the result has been directly pasted in the Matlab file “poly.m” and can be readily exploited in this Software.

4 Acknowledgements

This work has been achieved under projects ENE2009-09276 of the Spanish Ministerio de Educación y Ciencia and PAI-05-045 of the Consejería de Educación y Ciencia de la Junta de Comunidades de Castilla-La Mancha.

References

[1] A. Bret. The Astrophysical Journal, 699:993, 2009.