УДК 004.942

O. Shcherbakov (M.Sc.)
Donetsk national technical university
shcherbakov@cs.donntu.edu.ua

DEVELOPMENT OF NEW VECTOR OPERATIONS IN PROMOT

The order of adding new functionality to ProMoT and peculiarities of the implementation of symbolic differentiation in ProMoT are considered in this article. The way of differentiation for new operators is proposed. The implementation of new vector operations is done.

Keywords: ProMoT, DIANA, vector operations, catalytic fixed bed reactor

Introduction

ProMoT [1] (Process Modeling Tool) is a software tool for describing complex technical and biological systems. Models are simulated and studied in DIANA [2]. ProMoT has support of vector operations, like summations and products of vector elements, but has only binary minimum and maximum operations. Vector form of these operations is necessary for describing some industry models.

One example is a catalytic fixed bed reactor (CFBR) with a wall cooling, see Figure 1. The wall temperature is controlled by a PI-controller

$$T_{\mathrm{wall}} = K\left\{(T_{\mathrm{SP}} - T') + \frac{1}{T_I}\int_0^t (T_{\mathrm{SP}} - T')\,\mathrm{d}\,t\right\} + T_{\mathrm{wall,SP}}, \quad (1)$$

where T_{SP} is the set-point temperature and T' is the measured temperature.

For PI-controller description in MDL (Model Description Language), the above expression should be reformulated as follows:

$$e_1 = T_{\mathrm{SP}} - T',$$

$$\frac{\mathrm{d}\,e_2}{\mathrm{d}\,t} = e_1, \quad (2)$$

$$e_2 = \int_0^t T_{\mathrm{SP}} - T'\mathrm{d}\,t.$$

Resulting in the MDL-equation

© *Shcherbakov O., 2012*

$$T_{\text{wall}} = K\left(e_1 + \frac{e_2}{T_I}\right) + T_{\text{wall,SP}} \tag{3}$$

that determines the wall temperature. T_{wall} regulates the reaction intensity to keep the maximum (measured as T′) temperature in the reactor not higher the set point temperature T_{SP}. With previous realization T′ is determined as

$$T' = T(z = 0.5, t) \tag{4}$$

that controls temperature only at the point z = 0.5. Problem of controlling the maximum temperature in the whole reactor can be solved with an introduction of new vector operations in ProMoT

$$T' = \underset{z}{\text{vmax}}\left(T(z,t)\right). \tag{5}$$

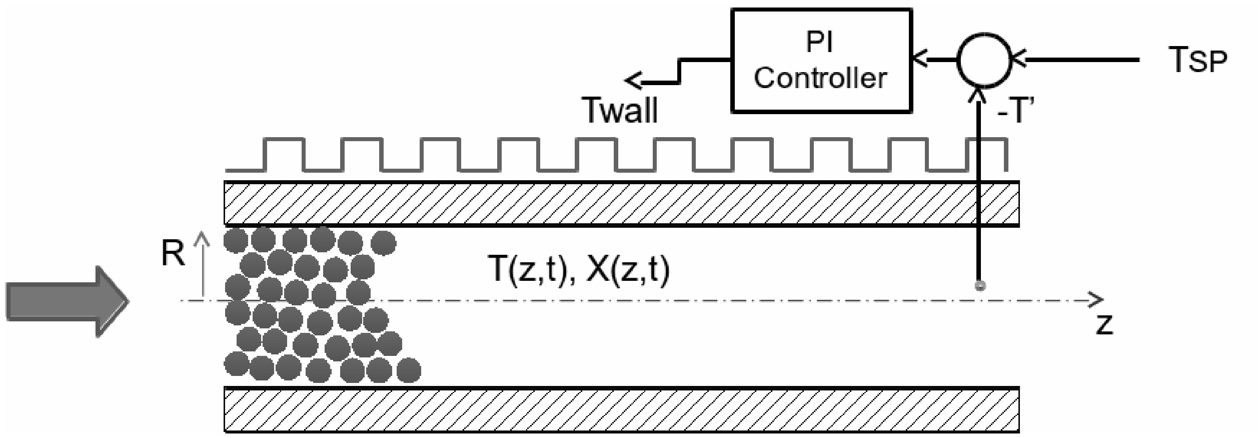


Figure 1: Simple CFBR model

ProMoT is written with use of different technologies and languages. The kernel is written in Common LISP [3], GUI — in Java. Maxima is used for simplification of expressions and symbolic differentiation. Models for DIANA are in C++, there are also available export functions to Matlab and DIVA (Fortran code).

Adding the new functionality was performed in the following steps:

- extending the ProMoT parser,
- writing template model implementation for DIANA in C++,
- extending the ProMoT-kernel to add vmin/vmax C++ code for residuals,
- extending ProMoT with Jacobian code generation for vmin/vmax,
- embedding new functionality into ProMoT/DIVA/DIANA environment.

Extending the ProMoT parser

Code-Generator is a component of ProMoT which was originally developed by Institut für Systemdynamik und Regelungstechnik (ISR) Universität Stuttgart. It is used in ProMoT as a parser for arithmetic expressions. Existing max/min operators are binary which accept two arguments and return the maximum or minimum value, respectively. Typical vector operators in ProMoT, like :sum and :prod, are reducers. They accept 4 arguments: index variable, lower and upper boundaries and expression. New vector operators, which are called *:vmax* and *:vmin*, should have the same arguments. The syntax, classes definitions and appropriate methods were copied from *:sum* and *:prod* operators as these reducer operators are similar.

Writing template model implementation for DIANA in C++

New C++-reducer code was a simple function for finding max in the array. Adapting existing code for this task was straightforward.

Extending the ProMoT-kernel

New classes and methods for implementing new operations were added to ProMoT-kernel. Existing methods were extended with generation of previously prepared C++ code. Simulation of a test model was done correctly but with a huge number of time-points when using generated symbolic Jacobian.

Extending ProMoT with Jacobian code generation for vmin/vmax

Symbolic jacobians in ProMoT are generated by Maxima system. Expression under reducer is translated to Maxima expression and then simplified and differentiated. The resulting Maxima-expression is translated to C++ code.

Sum and vmax function differentiation have similar implementations, for example for a scalar function R $f:\mathbb{R}\to\mathbb{R}$ and indexed variable x_i for $i = a,\ldots,b$:

$$\frac{\partial \sum_{i=a}^{b} f(x_i)}{\partial x_k} = f'(x_k), \tag{6}$$

$$\frac{\partial \mathrm{vmax}_{i=a}^{b} f(x_i)}{\partial x_k} = f'(x_k)\delta_{k,i_{\max}}, \tag{7}$$

Where δ_{ij} is the Kronecker delta, and i_{max} is the index of the maximum element in the array $f(x_i)$ for $i = a,\ldots,b$.

Calculation of indices of max/min elements was implemented in the code for residual generation. For making correspondence between these indices and the Jacobian-calculations the next has been done:

- vmin/vmax classes were extended with unique identifiers;
- translation to Maxima was rewritten for vmax/vmin reducers;
- caching of identical reducers was disabled for vmin/vmax reducers, as $\delta k, i_{max}$ in (7) may depend on reducers environment.

Generation of symbolic Jacobians was extended for vmin/vmax operators.

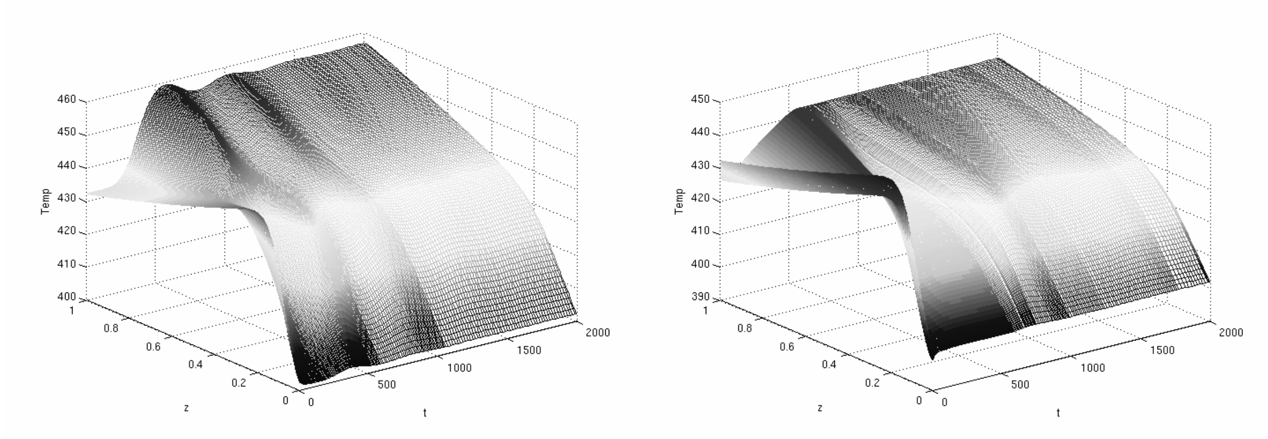
Embedding new functionality into ProMoT/DIVA/DIANA environment

New functionality was implemented for DIANA only. Export functions to other systems (DIVA, Matlab) were extended with an exception generating code for models with unsupported vmin/vmax functions.

Code-generator is also used as a standalone piece of software and is used in other projects, so all code changes were rewritten for applying only for build as a part of ProMoT.

Comparison of old and new model

Simulation results for old and new model are shown in Figure 2. PI-controller with new :vmax operator keeps maximum temperature not higher as given (T_{SP} = 442 K) in the reactor domain.



a) old model (4)

b) new model (5) with :vmax/:vmin operators

Figure 2: Simulation results

Conclusions

New vector operations are developed and new functions :vmax/:vmin are implemented in ProMoT. Functionality of dependent software is kept correct and expected. This work was done as a part of cooperation agreement between DonNTU and Max Planck Institute (MPI) for Dynamics of Complex Technical Systems Magdeburg during the scientific work of author in MPI in 2011-2012

References

1. Ginkel M., Kremling A., Nutsch T., Rehner R., and Gilles E.D. Modular modeling of cellular systems with ProMoT/Diva. Bioinformatics, 19, 2003, pages 1169–1176.
2. Krasnyk M., Bondareva K., Milokhov O., Teplinskiy K., Ginkel M., and Kienle A. The ProMoT/Diana simulation environment. In W. Marquardt and C. Pantelides, editors, 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering, Garmisch-Partenkirchen, Germany, 2006, pages 445–450.
3. Graham P. ANSI Common Lisp. Prentice-Hall, 1996.
4. Mirschel S., Steinmetz K., Rempel M., Ginkel M. and Gilles E. D. ProMoT: Modular Modeling for Systems Biology. Bioinformatics, 25:5, 2009, pages 687-689
5. Mangold M., Angeles-Palacios O., Ginkel M., Waschler R., Kienle A. and Gilles E. D. Computer Aided Modeling of Chemical and Biological Systems - Methods, Tools, and Applications. Industrial & Engineering Chemistry Research, 44(8), 2005, pages 2579-2591.

Надійшла до редакції 22.09.2012 р.
Рецензент: канд.техн.наук, проф. Анопрієнко О.Я.

А.С. Щербаков
Донецкий национальный технический университет

Разработка новых векторных операций в ProMoT. Рассмотрен порядок добавления новой функциональности в систему ProMoT. Рассмотрены особенности реализации символьного дифференцирования в ProMoT. Предложен способ дифференцирования для новых операторов. Выполнена имплементация новых векторных операций.
Ключевые слова: ProMoT, DIANA, векторные операции, catalytic fixed bed reactor

О.С. Щербаков
Донецький національний технічний університет

Розробка нових векторних операцій у ProMoT. Розглянуто порядок додавання нового фунціоналу до системи ProMoT. Розглянуті особливості реалізації символьного диференціювання у ProMoT. Запропоновано спосіб диференціювання для нових операторів. Виконано імлементацію нових векторних операцій.
Ключові слова: ProMoT, DIANA, векторні операції, catalytic fixed bed reactor