Automated parallelization of Crank-Nicolson scheme in feedback-nanoporous adsorption systems

Anatoliy Doroshenko
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
doroshenkoanatoliy2@gmail.com

Olena Yatsenko
Institute of Software Systems, NAS of Ukraine
Kyiv, Ukraine
oayat@ukr.net

Mykhaylo Petryk, Dmytro Mykhalyk
Ternopil Ivan Puluj National Technical University
Ternopil, Ukraine
mykhaylo_petryk@tntu.edu.ua
d.mykhalyk@gmail.com

***Abstract*—The results of the design and parallelization of a program implementing the Crank-Nicolson method for solving partial differential equations are given. The particular feature of our approach consists in using high-level algebra-algorithmic program specifications represented in a natural linguistic form. The developed software tools for constructing and synthesis of programs automatically translate the specifications into source code in a programming language.**

***Keywords*—algebra of algorithms, automated program design, finite difference method, parallel computing.**

Introduction

Parallel computing on multiprocessor systems is the main source providing highly-productive computation at solving complex scientific and technical problems. The problem of designing efficient programs for multiprocessor architectures can be successfully solved by specialization to subject domains and deep coverage of all stages of the software life cycle with the use of tools for automated designing and programming, beginning from writing primary specifications to generating executable code. First of all, the basis for such automation is the high-level formalization of designing parallel programs and automation of formal transformations of programs with the purpose of optimization of their performance. In the previous works [1–3], the formal facilities for developing parallel programs for multicore processors and graphics processing units were developed. They were based on Glushkov's system of algorithmic algebra (SAA) and term rewriting technique. Based on the developed theory and methodology, the integrated toolkit for designing and synthesis of programs (IDS) was developed.

In this paper, we apply our algebra-algorithmic methodology and tools for designing a parallel program implementing the Crank-Nicolson method [4] to be executed on a multicore processor. This method is applied in modeling adsorption and diffusion of several gases through a nanoporous solid and the instantaneous (out of equilibrium) distribution of the adsorbed phases [5, 6] is particularly important in many fields, such as gas separation, heterogeneous catalysis, purification of confined atmospheres, reduction of exhaust emissions contributing to global warming, etc.

Mathematical model of absorption mass transfer

The program implementing the Crank-Nicolson method is used in the software library for mathematical modeling of kinetic parameters of diffusional mass transfer of gas in zeolites with microporous structure [5, 6]. The model of one-component absorption mass transfer in the catalytic media of nanoporous structure particles is presented as the following system of differential equations:

$$\frac{\partial c}{\partial t} = D_{inter}\frac{\partial^2 c}{\partial z^2} - \theta_{intra}\left(\frac{\partial q}{\partial r}\right)|_{r=R}\ , \tag{1}$$

$$\frac{\partial q}{\partial t} = D_{intra}\left(\frac{\partial^2 q}{\partial r^2} + \frac{2}{r}\frac{\partial q}{\partial r}\right), \tag{2}$$

with initial and boundary conditions

$$\begin{gathered} c(t,z)|_{z=0} = 0,\ q(t,r,z)|_{t=0} = 0, \\ c(t,z)|_{z=l} = c_\infty,\ q(t,r,z)|_{r=R} = k\cdot c(t,z), \\ \frac{\partial c(t,z)}{\partial z}|_{z=0} = 0,\ \frac{\partial q(t,x,z)}{\partial r}|_{r=0} = 0. \end{gathered} \tag{3}$$

Equation (1) describes mass transfer in the interparticle space and contains, in the right part, the function of diffusion impact in nanoporous spherical particles on diffusion in the interparticle space. Equation (2) describes mass transfer in intraparticle space with current concentration $q(t,r,z)$ associated with a concentration in the interparticle space $c(t,z)$ with a boundary feedback-condition — equilibrium condition on the particle surface. The diffusion coefficients D_{inter} and D_{intra} characterize the rate of mass transfer processes in the interparticle space and in nanopores of intraparticle space, and the coefficient θ_{intra} describes the impact of the intraparticle transfer on interparticle one, R radius of the particle [5]. In [5] a high-performance accurate analytical solution of this model is constructed using Heaviside operational method. The

Crank-Nicolson method is used in modeling diffusion mass transfer for a homogeneous nanoporous bed. It is a finite difference method used for numerically solving partial differential equations [4]. It is based on the central difference in space, and the trapezoidal rule in time, giving second-order convergence in time.

Automated design of the parallel program implementing Crank-Nicolson method

Our approach to program design is based on SAA [1], the algebra which is applied for the formalized representation of algorithmic knowledge. The developed IDS toolkit [1, 2] is intended for the automated design of algorithm schemes and generation of programs in target programming languages (C, C++, Java). The advantage of using SAA schemes is that they are represented in a form close to natural language. The design process is represented by a tree of an algorithm with selecting SAA constructs from a list, which eliminates syntax errors. Consider the process of the parallelization of one of the subroutines of the Crank-Nicolson scheme. The sequential SAA scheme of the subroutine designed using the IDS toolkit is given below. It contains a loop by variable $k \in [1, ..., N]$, in which functions iterate_c(k) and iterate_q(k) compute k-th layer for c and q values, correspondingly.

```
SCHEME CRANK-NICOLSON SEQUENTIAL ==
“iterations”
==== FOR (k FROM 1 TO N)
        “iterate_c(k)”; “iterate_q(k)”
        END OF LOOP
END OF SCHEME
```

The parallelization of the scheme consists in dividing the segment $[1...N]$ into *NumThreads* sections to be processed in parallel. The SAA scheme of the parallelized algorithm is the following:

```
SCHEME CRANK-NICOLSON PARALLEL ====
“iterations”
===PARALLEL(i = 1,..., NumThreads) (
       “IterateThread(i)”);
       WAIT ‘Processing in all (NumThreads) threads
          is finished’;
“IterateThread(i)”
===“chunk := N / NumThreads”;
       “start := (i – 1) * chunk + 1”;
       “end := (i – 1) * chunk + chunk”;
       IF (i=NumThreads) THEN “end:= N” END IF;
       FOR (k FROM start TO end)
          “iterate_c(k)”; “iterate_q(k)”
        END OF LOOP;
        CP ‘Processing in thread (i) is finished’;
END OF SCHEME
```

IDS toolkit automatically generated Java code based on the constructed SAA schemes. The sequential and parallel programs were executed on Intel Core 2 Quad Q9300 processor, 4 cores, 2.5 GHz. The dependency of the execution time of the programs on the value of N is given in Fig. 1. Fig. 2 shows the values of multiprocessor speedup $Sp = T(1)/T(NumThreads)$, where $T(1)$, $T(NumThreads)$ is the execution time of the sequential and parallel program, respectively. The average value of Sp is 2.5, and the efficiency of processor usage is 64%, which is quite a good parallelization rate.

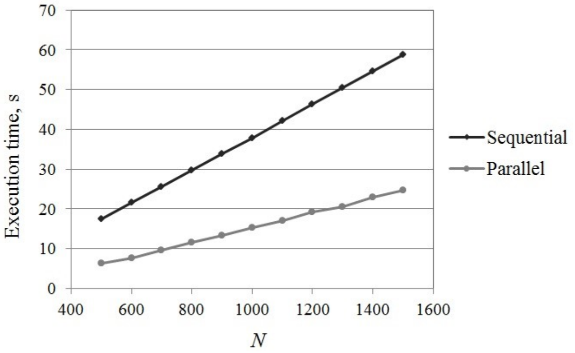

Fig. 1. The dependency of the execution time on N for sequential and parallel Crank-Nicolson scheme

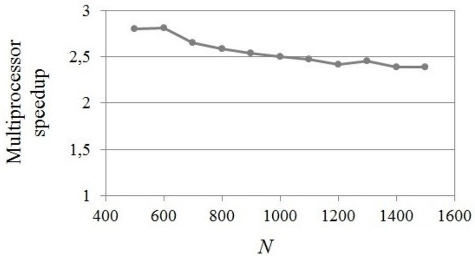

Fig. 2. The dependency of the multiprocessor speedup on N for parallel Crank-Nicolson scheme

Conclusions

The results of the design and parallelization of the Crank-Nicolson scheme for solving partial differential equations are given. The particular feature of our approach consists in using high-level algebra-algorithmic program specifications represented in a natural linguistic form. The developed software tools for constructing and synthesis of programs automatically translate the specifications into source code in a programming language.

References

[1] P. I. Andon, A. Yu. Doroshenko, K. A. Zhereb, and O. A. Yatsenko, Algebra-Algorithmic Models and Methods of Parallel Programming. Kyiv: Akademperiodyka, 2018.

[2] A. Doroshenko, K. Zhereb, and O. Yatsenko, “Developing and optimizing parallel programs with algebra-algorithmic and term rewriting tools,” Proc. Int. Conf. ICTERI 2013, Kherson, Ukraine (19–22 June 2013), Revised Selected Papers, vol. 412.

[3] A. Doroshenko, O. Beketov, M. Bondarenko, and O. Yatsenko, “Automated design of parallel programs for heterogeneous platforms using algebra-algorithmic tools,” Proc. Int. Conf. ICTERI 2019, Revised Selected Papers, vol. 1175, 2020.

[4] J. Crank and P. Nicolson, “A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type,” Proc. Camb. Phil. Soc., vol. 43, no. 1.

[5] M. Petryk, S. Leclerc, D. Canet, and J. Fraissard, “Modeling of gas transport in a microporous solid using a slice selection procedure: application to the diffusion of benzene in ZSM5,” Catalysis Today, vol. 139, no. 3, pp. 234–240, 2008.

[6] M. Petryk et al., “Competitive diffusion of gases in a zeolite bed: NMR and slice selection procedure, modelling and parameter identification,” The Journal of Physical Chemistry C. ACS (USA), vol. 119, no. 47, pp. 26519–26525, 2015.