Mykhailo FEDOROV

West Pomeranian University of Technology, Szczecin

doi:10.15199/48.2015.03.05

Parallel computing techniques on enhancement of thermal-hydraulic analysis of fluid flow networked systems

Abstract. *The considerable computation time of a practical application of sequential algorithms for simulating thermal and flow distribution in fluid flow networked systems (FFNS's) is the motivating factor to study their parallel implementation. The mathematical model formulated and studied in the paper requires the solution of a set of nonlinear equations, which are solved by the Newton-Raphson method. An object-oriented solver automatically formulates the equations for networks of an arbitrary topology. The hydraulic model that is chosen as a benchmark consists of nodal flows and loop equations. A general decomposition algorithm for analysis of flow and temperature distribution in a FFNS is presented, and results of speedup of its parallel implementation are demonstrated.*

Streszczenie. *Zaproponowano model do symulacji równoległej zadania analizy statycznego przepływu cieczy w sieciach przepływowych. Model sprowadza się do rozwiązania układów równan nieliniowych metodą Newtona-Raphsona. Przedstawiono algorytm dekompozycyjny do analizy rozdysponowania przepływu i temperatur w sieci przepływowej oraz wyniki przyspieszenia jego implementacji równoległej.**(Techniki obliczeniowe rownoległe usprawniające analizę termiczną oraz hydrauliczną sieci przepływowych)***

Keywords: thermal system, steady state flow, parallel simulation, OpenMP
Słowa kluczowe: system cieplny, przepływ statyczny, symulacja rownoległa, OpenMP

Introduction

The domain of application of a fluid flow networked systems (FFNS) analysis is very wide, e.g. airplane hydraulic, fuel or environmental control systems, district heating systems, air-conditioning systems of buildings, data centers, trains or ships, water or gas distribution systems, and so on. An FFNS is a network of interacting components. Its task is to convert the magnitudes of parameters fixed at certain boundary system points into the magnitudes prescribed at the other ones. Such parameters are called boundary conditions. The conversion is a result of mutual transformations of different forms of energy and matter. The transformation is performed in the FFNS components, with each type defining its unique transformation process. Thus, the FFNS type is characterized by the totality of such processes. The moving forces of the transformations are finite temperature and pressure differences. To answer the question how such transformations are realized by the system requires solving analysis problem. To do this, we have additionally to fix magnitudes of geometrical parameters (e.g. for a tube they are tube diameter, length, roughness, wall thickness, etc.) and boundary conditions (temperatures, pressures). The result of the analysis is a vector of the thermodynamic parameters (pressures, temperatures, enthalpies, etc.) and flow rates. Practical simulations of aircraft environmental control systems [1] demonstrate the dependence of flow on temperature. It follows from the fact that a Reynolds number depends on dynamic viscosity being a function of temperature, in turn. For fluids, change of temperature on 20 K changes the dynamic viscosity almost twice, which can easily be seen from the water property tables [2]. The above general description concerns any thermal system type, including air conditioning systems of data centers. It is well-known how important it is to keep data center, servers or computer room at an optimum temperature for the best data center performance and reliability. For example, if the humidity drops below some relative humidity in a data center, static electrical charges can build-up causing sparks that damage servers and IT equipment. Data centers evolve. They are flexible architectures that may change, even drastically, their powers and emitting heat. That is, a lifecycle of servers does not match the lifecycle of air-conditioning systems of data centers. For the cost savings sake the components of a data center are integrated into a building management system that complicates essentially their design, analysis and control because its each component should be designed or analyzed as an integral part of the totality of interacting components. Being a criterion parameter Reynolds' number makes impact on the values of resistances and heat exchange intensity. The impact of a Reynolds number change due to temperature is more essential for laminar or smooth pipe flows [3]. If a pipe network is controllable, then any kind of flow (laminar, turbulent) can take place in its various parts. To account the temperature impact, the joint thermal and hydraulic simulation should be performed. Individually, the analysis methods for the simulation of flow distribution and temperatures have been addressed in literature. For the flow steady state analysis the most common methods are those in which independent variables are expressed in terms of the link or chord flows, of the loop flow correction or the nodal heads. In [4], a lack of stability of the nodal method [4, 5, 6] is reported. In [7, 8] the numerical superiority of the flow method over the nodal head method has been proved. The comprehensive analysis, history and examples of the use of these methods are available in [1], [9, 10, 11, 12, 13, 14]. In [15] the value of a full-set equation approach is demonstrated which lends itself to the technique of introducing additional equations to describe modified or added network characteristics meeting specified conditions. Being a more general formulation the full-set approach is also employed in this study. The thermal model is a particular form of the first thermodynamic law. The matrix formulation of thermal model is studied, in detail, in [16, 17] and, later, in [18]. The general methodology of the thermal-hydraulic simulation is addressed in [19].

FFNS is a product of design, i.e., on the one hand, its structure should be defined – a graph of components types and the sequence of the transformation processes in FFNS components. On the other hand, each component characteristics should be specified, defining qualitatively intensity of a transformation process in each FFNS component. The latter is provided by design of a component, which is often reduced to its sizing. Normally, FFNS is designed for a typical mode of its functioning. The simulation models should be used to determine that state parameters are still within allowable ranges of FFNS operation. Otherwise, the state parameters must be adjusted. Control devices are important components of any FFNS. They provide design values of temperatures, flows and pressures in certain parts of a system when FFNS boundary conditions change. One of the basic physical mechanisms of control is throttling, meaning the change of valve resistances. It changes the amount of working fluids

with different temperatures mixed in the temperature control devices in that way redistributing temperatures in the whole FFNS as well as flows and pressures. In engineering, control devices should maintain the operational state parameters and flow rates close to the ones obtained at the design stage. Valve resistances calculated for various control devices and boundary conditions may characterize steady state temperature, pressure, humidity or flow rate controllability of FFNS. This may support decision-making at FFNS design in specifying composition and location of FFNS control devices.

The detail analysis and design of such thermal systems become possible to perform if they are computer-aided. However, it is equally important to design the FFNS performing the desired task in emergency or changing environmental conditions. The range of these conditions constitutes the FFNS envelope within which the FFNS is considered to be statically controllable. Hence, the envelope specification accompanies FFNS design. It is impossible to do without the model which is capable of analyzing different FFNS configurations invoked by the changed valve positions of various types of control devices. Thus, the robust method for analysis of FFNS static controllability must result in a solution at any FFNS control configuration.

The use of it for more and more large networks is becoming more and more time-consuming. It is the motivating factor to study a parallel implementation of the sequential methodology [19]. The OpenMP standard [20] is chosen as a tool for paralleling, because its use is very straightforward.

Mathematical model

To formulate a FFNS model we use the graph theory as a framework to relate process models described by fundamental laws of physics. FFNS graph, $G = (V, E)$ is a number set of nodes $V \in \mathbb{N}$ and edges $E \subseteq V \times V \in \mathbb{N}$, with $n_t = |V|$, $e = |E|$ being the number of FFNS graph nodes and edges, respectively. Besides, $V = V_i \cup V_b$ such that $V_i = \{i \in \mathbb{N} : 1 \leq i \leq n_i\}$, $V_b = \{i \in \mathbb{N} : n_i < i \leq n, \mathrm{d_G}(i) = 1\}$, where $n_i = |V_i|$, $n_b = |V_b|$, $n = n_i + n_b$, and $\mathrm{d_G}$ stands for a graph node degree. Nodes from V_b are called boundary nodes, where some process parameters are given values. Edges may also be specified by some process parameter, variable or fixed (e.g. by nodal flow demands). We specify a FFNS graph with an incidence matrix, $\mathbf{A}_{n_i \times e}$.

We consider the stationary thermal and flow simulation of incompressible fluid in a pipe network. We do not explicitly take into account pumps in the network, but they could easily be incorporated, as well as the given discharges from the interior nodes. Three basic conservation physical principles are necessary to formulate the model: continuity, momentum and energy

$$\mathbf{A}\begin{pmatrix} \dot{\mathbf{m}} \\ \dot{\mathbf{M}} \end{pmatrix} = \left(\tilde{\mathbf{A}}_{n\times e_m}|\mathbf{0}\right)\dot{\mathbf{m}} + \left(\mathbf{0}|\bar{\mathbf{A}}_{n\times e_M}\right)\dot{\mathbf{M}} = \mathbf{A_m}\dot{\mathbf{m}} + \mathbf{A_M}\dot{\mathbf{M}} = \mathbf{0}\,, \tag{1}$$

$$\mathbf{A}\mathbf{D_m}\mathbf{T} = \mathbf{0}\,, \tag{2}$$

$$\mathbf{A}^T\mathbf{P} + \mathbf{D_r}\begin{pmatrix} \dot{\mathbf{m}} \\ \dot{\mathbf{M}} \end{pmatrix} = \mathbf{0}\,, \tag{3}$$

$$\mathbf{B}_{n_T\times e}\mathbf{T} = \mathbf{0}\,, \tag{4}$$

where $\mathbf{A} = \mathbf{A_m} + \mathbf{A_M}$; $\dot{\mathbf{m}}_{e_m\times 1}$, $\dot{\mathbf{M}}_{e_M\times 1}$, $\mathbf{T}_{e\times 1}$, $\mathbf{P}_{n\times 1}$, $\mathbf{K}_{e\times 1}$ are vectors of flow rate variables, fixed flow rates, temperatures, pressures, and resistances, respectively; $e_m = e - e_M$; $\mathbf{D_m} = \mathrm{diag}(\dot{\mathbf{m}}, \dot{\mathbf{M}})$, $\mathbf{D_r} = \mathrm{diag}(\mathbf{K}\,|\dot{\mathbf{m}}|)$.

The first principle (Eq.(1)) determines nodal equations implying that at each node the sum of flows in pipes incident to the node is zero. The second principle (Eq.(2)) expresses pressure differences between two pipe sections through flow rates and duct resistances K_k. The latter are defined differently for different types of ducts

$$K_k = L\lambda/2\rho D_h A_c^2\ , \tag{5}$$

or heat exchanger channels

$$K_k = \frac{1}{2\rho A_c^2}\left[\left(1-\sigma^2+K_c\right) + \lambda\frac{A_s}{A_c} - \left(1-\sigma^2-K_l\right)\right]_k\ , \tag{6}$$

where D_h- hydraulic diameter, A_c- cross-sectional area, ρ- density, λ - Darcy-Weisbach friction factor, L - duct length, σ - ratio of the free flow area to the frontal area on one side of the exchanger, K_c, K_l - entrance and exit loss coefficients of a heat exchanger channel, A_s - heat transfer surface on one side of a heat exchanger. We use the following formulas for friction factors (see Eq.(5)) in this study [21]

$$\lambda = f\left(Re, \Delta, D_h\right)\ , \tag{7}$$

where Δ is an inner surface duct roughness, Re is the Reynolds number

$$Re = \dot{m}D_h/\mu A_c\ , \tag{8}$$

where μ is a dynamic viscosity. The third principle yields an equation Eq.(2) and elemental equations which forms depend on the type of an element. In the paper we differentiate two elemental models that appear from the energy principle. Again, they are a tube and a heat exchanger channel. The tube model is defined as follows

$$T_{k,i_1} - e^{\left(\frac{UA_s}{C\dot{m}}L\right)_k}T_{k,i_2} + \left[e^{\left(\frac{UA_s}{C\dot{m}}L\right)_k} - 1\right]T_{k,w} = 0\ , \quad k = \overline{1,e}\ , i_1 \neq i_2\ , \tag{9}$$

where T_{k,i_1}, T_{k,i_2}, are temperatures at pipe ends incident to nodes i_1 and i_2, $T_{k,w}$ is a boundary temperature (e.g. a tube wall temperature); U is a overall heat transfer coefficient, C is an average heat capacity, A_s is an inner heat transfer area. The overall heat transfer coefficient is obtained from the following formula (see e.g. [2])

$$UA_s = \left[\frac{1}{hA_s} + \frac{D_{h,1}\ln\frac{D_{h,2}}{D_{h,1}}}{k_w A_s}\right]^{-1}\ , \tag{10}$$

which is the integral of the differential equation of heat conduction through a cylindrical wall with the given temperature on the tube outer surface as the boundary condition. In (Eq.(10)), $D_{h,1} \equiv D_h$, but $D_{h,2}$ is an outer tube diameter, h is a heat transfer coefficient, k_w is a thermal conductivity. In turn, equation (Eq.(9)) is a form of the analytical solution of the differential equation of convection heat transfer between incompressible fluid flow in the tube and the inner surface of a tube cylindrical wall (see e.g. [2]). Equation (2) means nodal energy balance. The heat transfer coefficient h in Eq.(9) is obtained from the definition of the Nusselt number

$$Nu = hD_h/k\ , \tag{11}$$

which is determined as follows [22]

$$Nu = 0.0021 Re^{0.8} Pr^{0.43} (Pr/Pr_w)^{0.14} \ , \quad Re \geq 10^4 \ , \tag{12}$$

$$Nu = (Nu_l^{6.267}/Nu_t^{5.267}) Re^{0.68 \ln Nu_t/Nu_l} \ , \quad 2300 \leq Re < 10^4 \ , \tag{13}$$

$$Nu = 1.55 (PrReD_h/L)^{1.33} (\mu/\mu_w)^{0.14} \ , \quad Re < 2300 \ , \tag{14}$$

for turbulent, transitional and laminar flows, respectively. Into Eqs.(12)-(14) the Prandtl number quantity enters. It is defined as follows

$$Pr = C\mu/k \ , \tag{15}$$

In Eq.(13) Nu_l is computed from Eq.(14) at $Re = 2300$, and Nu_t is computed from Eq.(12) at $Re = 10^4$. All working fluid properties (C, k, μ) are assumed to be functions of mean temperatures (T_m) for every pipe network element. Hence, all the dimensionless parameters (Re, Nu, Pr) are also functions of temperature.

The heat exchanger model is characterized, in turn, by the following relationships [25]

$$\epsilon = \frac{C_h \left(T_{h,1} - T_{h,2}\right)}{C_{min} \left(T_{h,1} - T_{c,1}\right)} = \frac{C_c \left(T_{c,2} - T_{c,1}\right)}{C_{min} \left(T_{h,1} - T_{c,1}\right)}, \tag{16}$$

where

$$\epsilon = \frac{1}{C \cdot N_{tu}} \sum_{i=0}^{\infty} \left(1 - e^{-N_{tu}} \sum_{j=0}^{i} \frac{N_{tu}^j}{j!}\right) \left[1 - e^{C \cdot N_{tu}} \sum_{j=0}^{i} \frac{(C \cdot N_{tu})^j}{j!}\right], \tag{17}$$

$$C = \frac{C_{min}}{C_{max}}, \tag{18}$$

$$C_{min} = min(C_c, C_h), \ C_{max} = min(C_c, C_h), \quad C_c = c_{p,c} \dot{m}_c, \ C_h = c_{p,h} \dot{m}_h, \tag{19}$$

$$N_{tu} = \frac{UA}{C_{min}}, \tag{20}$$

$$\frac{1}{UA} = \frac{1}{h_h \eta_{0h} A_h} + \frac{a}{k_w A_w} + \frac{1}{h_c \eta_{0c} A_c},$$

$$Pr = \frac{c_p \mu}{k}, \ \alpha = c_p \frac{\dot{m}}{A_c} St, \tag{21}$$

$$St = St(Re) Pr^{-\frac{2}{3}}, \tag{22}$$

$$\eta_0 = 1 - \frac{A_f}{A} \left[1 - \frac{\tanh 0.5mb}{0.5b}\right], \tag{23}$$

$$m = \sqrt{\frac{2h}{k_f \delta_f}}, \ D_h = \frac{4A_c}{A} L, Re = \frac{4\dot{m}}{\mu} \frac{L}{A}, \tag{24}$$

where $A = A_w + A_f$, A - heat transfer surface on one side of the exchanger, A_w - separation plate area, A_f - fin area, η_0 - overall fin efficiency, b - separation plate distance, St - the Stanton number, ϵ - heat exchanger effectiveness, δ_w - separation plate thickness, δ - fin thickness, 1 - inlet, 2 - outlet, c - cold, h - hot. For the purpose of study we assume that all the heat exchangers are one-pass compact cross-flow with both flows unmixed.

From the temperature model Eqs.(2), (4), (9)-(24) analysis it follows that it doesn't have full rank. Consequently, some additional equations must be automatically formed and added to the model [17, 19]. To do this, the flow rate distribution must be known. Assuming that it holds, we find out outflow pipes for each node. After mixing each outflow pipe has the same temperature.

Encoding of matrix $\mathbf{B}_{n_T \times e}$ is specific to the FFNS model in question. The peculiarity of the isothermal model of FFNS is a possibility to treat a temperature variable as an edge variable instead of being a node variable as it takes place in the general case. To encode the isothermal model of FFNS we introduce a set $V_T = \{k \in V : \Theta_k > 1\}$, where $\Theta_k = \mathrm{od_G}(k)$ denotes the out-degree of k. We also define a set $\Pi \supseteq \Pi_k = \Theta_k(1) \times (\Theta_k - \Theta_k(1))$ such that $\Pi = \bigcap_k \Pi_k = \oslash$, $k \in V_T$. Notice that the content of Π can be determined from $\mathbf{A}$. Then, n_T and $\mathbf{B}_{n_T \times e}$ in Eq.(4) are defined as follows, respectively $n_T = \sum_{k=1}^{|V_T|} |\Pi_k|$

$$B_{l,*} = \begin{cases} B_{l,i} = 1, & B_{l,j} = -1, \ \exists (i,j) \in \Pi_l \\ B_{l,k} = 0, & \forall k \notin (i,j), l = \overline{1, n_T}. \end{cases} \tag{25}$$

Having added Eqs.(4) and (25) to Eqs.(2), (4), (9)-(24) a FFNS model becomes the determined problem. With the model at hand, we can go over to answer the question how many times faster we can solve such a model in parallel than sequentially.

Results and discussion

With this end in view, a number of pipe networks configurations followed from the benchmark network (see Fig. 1) have been solved sequentially and in parallel. The configurations of pipe networks with the pipe number less than 66 are not depicted. Adding incrementally blocks of 66 pipes the maximum pipe number networks that has been solved has 528 pipes. Numerical experiments have been performed

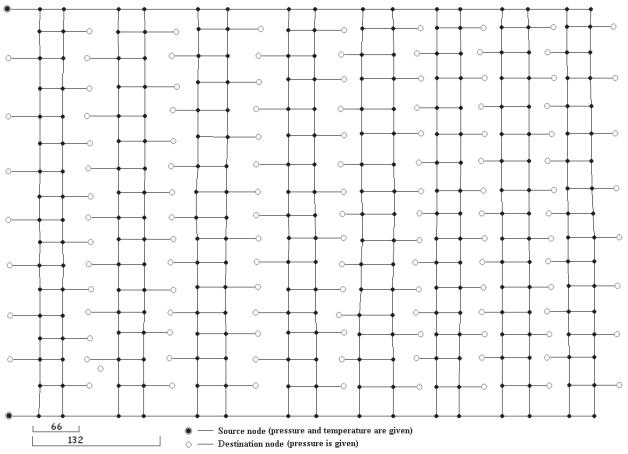

Fig. 1. The benchmark pipe network.

on the computer with 2×Quad Core Processors (Intel Xeon E5405 Quad Core Processor) under the Windows Server 2003 operation system. A pipe network solver is developed and compiled in the Visual Studio Team 2008 environment. The parallel implementation of the sequential algorithm is coded with the OpenMP standard, which substantially simplifies studies on parallelization, because programming with this standard is very straightforward [20]. Experience shows [1, 11, 19, 23] that a general solution algorithm of the nonlinear model described in section 2 can be decomposed into several stages (see the pseudo-code below): hydraulic, thermal and working fluid property ones.

```
function ṁ = hydraulic_model() {
\\ the fine-grained parallelization is implemented
\\ for all loops in the below functions
    while(δK > ε1) {
    \\ resistance fixed-point iterations
        while(δṁ > ε) {
        \\ Newton iterations for flow model
        \\ analysis
            form_hydlaulic_model_SLE(in K, in ṁ,
                out J, out F);
            \\ The below function solves Eq.(25)
            ṁ^new =solve_SLE_in_parallel(in J,
                    in F);
            ṁ^new =line_search(in ṁ^new);
            δṁ =estimate_relative_error(in ṁ^new,
            in ṁ);
            ṁ = ṁ^new ;
        }
        #pragma omp for \\ dependence-free loop
        for(i=0;i<number_of_pipes;i++)
        \\ evaluates Eq.(5) or Eq.(6)
            K^new = compute_resistances(in ṁ);
        δK = estimate_relative_error(in K^new,
                in K);
    }
}
function thermal-hydraulic_model() {
\\ a mean temperature is given
    while(δT_m > ε2) {
    \\ mean temperature fixed-point iterations
        \\ the first stage
        ṁ = hydraulic_model();
        \\ the second stage.
        \\ B is a right part of Eqs.(4), (2), (9)
        \\ or/and (16), (25)
        form_temperature_model_SLE(in C, in ṁ,
            out Φ, out B);
        T =solve_SLE_in_parallel(in Φ, in B);
        \\ the third stage
        #pragma omp for \\ dependence-free loops
        for(i=0;i<number_of_pipes;i++) {
            compute_mean_temperatures(in T,
                out T_m^new) ;
            compute_fluid_properties(in T_m^new);
            compute_model_parameters(in T_m^new);
        }
        δT_m = estimate_relative_error(in T_m^new,
            in T_m);
    }
}
```

Stage 1 operates at fixed mean temperatures. The hydraulic model computes flow rates at constant resistances with Newton's method [26], followed by parallel computation of new resistances being functions of flow rates. As a component of Newton's method, the parallel dense LU solver is implemented that uses Crout-like reduction with row pivoting. At each Newton iteration, the LU solver solves the following set of linear equations (SLE)

$$J(\dot{m}_i)\Delta\dot{m}_i = -F(\dot{m}_i) \ , \tag{26}$$

obtained from linearizing Eqs.(1), (3) and transforming Eq.(3) to the loop form, for the full-step flow correction vector, $\Delta\dot{m}_i$. To assure decreasing $\|F(\dot{m})\|_2^2/2$, a line search technique [26] is employed to correct $\Delta\dot{m}_i$, if necessary. Stage 2 operates both at fixed flow rates and mean temperatures (see the above pseudo-code).

In this case, Eqs.(2), (4), (9), (16), Eq.(25) constitute an SLE with constant coefficients Eqs.(10)-(24). Stage 3 computes average temperatures and fluid properties. Each iterative algorithm of the general one solves for one type of parameter vector and requires about 6-9 iterations to converge with accuracy $\epsilon \leq 10^{-9}$, $\epsilon \leq 10^{-6}$, $\epsilon_2 \leq 10^{-3}$ for flow rates, resistances and mean temperatures, respectively. One of the basic activities of the algorithm is to solve SLE. The SLEs are unsymmetrical and sparse that requires implementation of the solution algorithm for this general case of SLE. Numerical experiments demonstrate that the speedup of solving SLE benchmarks [27] of size of order 1000 by the LU factorization method doesn't exceed 2, for the given hardware (which configuration can be essential to achieving high parallel performance [28]). Therefore, we can expect

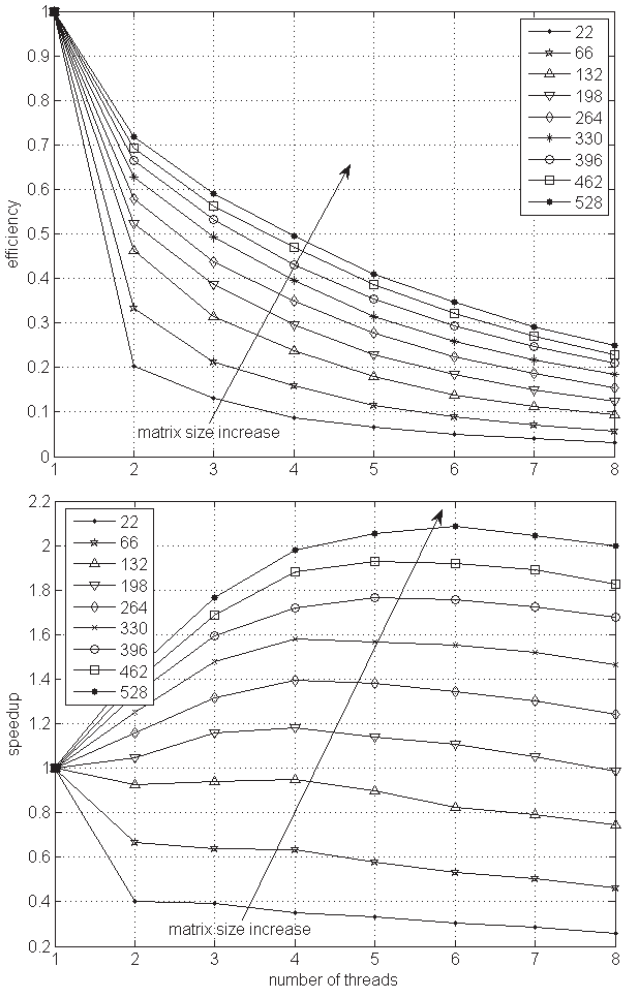

Fig. 2. The speedup and efficiency.

that the speedup of the whole algorithm will be of the similar order. Newton's method is largely sequential in nature. It consists of the sequence of subtasks such as formulation of equations, symbolic factorization, solving SLE, computing gradients and norms, etc., with numerous conditional constructs, reduction variables and a set of loops having small number of operations. Paralleling each task individually we realize fine-grained parallelization that requires the current task to be synchronized before running the next one that, in turn, might cause essential synchronization overheads. Despite the fact, numerical experiments verify (Fig. 2) scalability of fine-grained parallelism. Indeed, we can observe stable,

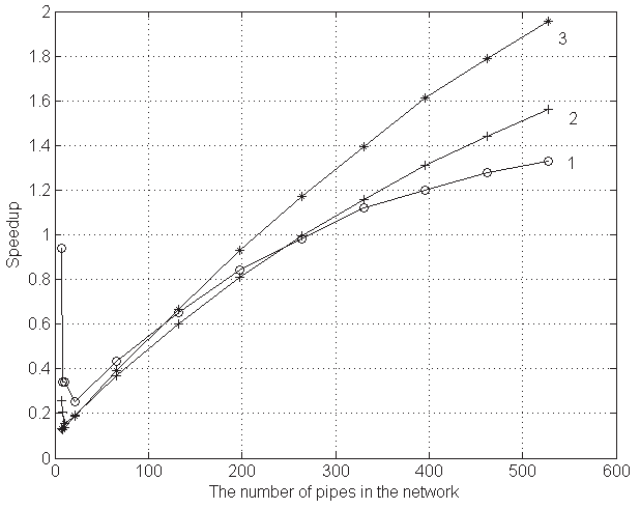

Fig. 3. Speedup of models corresponding different algorithm stages.

though not intensive, growth in speedup and efficiency starting even with relatively small pipe network sizes (about 150 pipes).

The next numerical experiments that are made give insight in the speedup effects of applying fixed-point iterations (the model decomposition) to the algorithm versus pipe network sizes and types of the models being solved, and that may be individually commercially useful. The aim is to verify growth in speedup with increase in element models complexity. Curve 1 in Fig. 3 demonstrates the speedup of parallelization of the flow model analysis, where resistances K are kept fixed (see the pseudo code). We can note, that speedup is observed for networks sizes being greater than 250 pipes (see Fig. 3). Experiments with the hydraulic model (curve 2) and the complete model (curve 3) demonstrate that speedup substantially increases. In comparison to the flow model the speedup observed is for minimal synchronization overheads incurred with the parallel implementation of the relaxation method, which correct resistances (stage 1), and average temperatures and working fluid properties (stage 3). Speedup will increase even greater if elements model (e.g. heat exchanger) of a pipe network require more computations for evaluating resistances and mean temperatures. Indeed, simulations have shown that the speedup for the heat exchanger network shown in Fig. 4 was a little greater than 1, whereas speedup for the equivalent pipe network with the same edge number is still less than 1. It sustains the assumption made above (due to more complex procedure of computation of a heat exchanger effectiveness with Eqs. (17)-(24) than with Eqs. (10)-(14) in case of calculation of a heat transfer coefficient of a pipe).

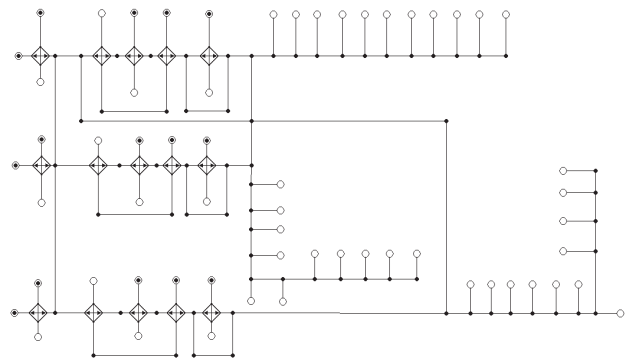

Fig. 4. Heat exchanger network

Conclusion

In the paper, the parallel implementation of the steady-state thermal-hydraulic analysis in OpenMP is presented. The mathematical model studied contains all typical tasks inherent to such an application domain. To demonstrate the influence of each task on the final speedup numerical experiments have been carried out for different sets of subtasks and sizes of pipe networks on computer with 2×Quad Core Processors (Intel Xeon E5405 Quad Core Processor) under the Windows Server 2003 operation system. The object-oriented code of the thermal-hydraulic analysis solver, which proved to be robust and scalable, has been implemented and compiled in Visual Studio Team 2008 environment. The paper demonstrates that the fine-grained parallel implementation in OpenMP of the decomposition algorithm considered results in speedup of order 2 for the network of 528 pipes, and the potential in speedup increase exists for larger sizes of FFNS's. It has been achieved by parallelization of the algorithm for solving SLE, being a dominant functionality of the flow model analysis and temperature submodels. Meanwhile, two fixed-point iteration procedures that compose these submodels into the whole are the sources of earned speedup value as compared to the hydraulic model, where solving SLE is a dominant functionality. For larger networks speedup will increase mainly thanks to greater number of its elements, which parameters are corrected by fixed-point iterations, and which, in turn, do not introduce additional data dependences and, as result, communication; while efficiency of the SLE parallelization will grow slowly. The results also demonstrate that further increase in speedup can be achieved by implementing more advanced symbolic matrix manipulations and decomposition methods resulting in minimal communications, provided that they altogether decrease the computation time of the sequential implementations of these algorithms.

The model presented can be easily extended to simulate pipe networks with more complex models of element (e.g. with heat exchangers) or working fluid properties (e.g. with account for phase transitions); doing so the speedup will always grow.

REFERENCES

[1] Kondrashenko, V., Vinnichuk, S., Fedorov M.: Simulation of Gas and Liquid Distributing Systems, Naukova Dumka, Kiev, 1990 (in Russian)
[2] Massoud M.: Engineering Thermofluids, Springer-Verlag, Berlin Heidelberg New York, 2005
[3] Idelchik I.E.: Handbook of Hydraulic Resistances, Machynostrojenije, Moscow, 1992 (in Russian)
[4] Donachie R.P.: Digital Program for Water Network Analysis, J. Hydraulics Div., vol.100, HY3, ASCE, pp. 393–403, 1974
[5] Chandrashekar M.: Extended Set of Components in Pipe Networks, J. Hydraulic Div., vol. 106, HY1, ASCE pp 133–149, 1980
[6] Nogueira A.C.: Steady-State Fluid Network Analysis, J. Hydraulic Eng., vol.119, 3, pp. 431–436, 1993
[7] Nielsen B.N.: Methods for Analyzing Pipe Networks, J. Hydraulic Eng., vol.115, 2, ASCE, pp. 139–157, 1989
[8] Altman T., Boulos P.F.: Convergence of Newton Method in Nonlinear Network Analysis, Mathl. Comput. Modelling, vol.21, 4, Elsevier, pp. 35–41, 1995
[9] Chandrashekar M.: Extended Set of Components in Pipe Networks, J. Hydraulic Div., vol. 106, HY1, pp. 133–149, 1980
[10] Evdokimov, A.G.: Optimal Problems of Engineering Networks, Wyzsha Shkola, Charkov, 1976 (in Russian)
[11] Merenkov A.P., Chasilev V.J.: Theory of Hydraulic Circuits, Nauka, Moscow, 1985 (in Russian)
[12] Larock B. E., Jeppson R. W., Watters G. Z.: Hydraulics of Pipeline Systems, CRC Press, Boca Raton London New York Washington, D.C., 2000
[13] Osiadacz A.J.: Steady-state Simulation of Gas Networks, Fluid systems Sp., Warszawa, 2001 (in Polish)

[14] Sennova E.V., Sidler V.G.: Mathematical Simulation and Optimization of Evolving Heat Supply Systems, Nauka, Novosibisk, 1987 (in Russian).
[15] Boulos P.F, Wood D.J.: Explicit Calculation of Pipe Network Parameters, J. Hydraulic Eng., vol. 116, ASCE, pp. 1329–1344, 1990a
[16] Fedorov M.: Automation of Mathematical Model Construction for the Analysis of Heat Regimes of Heat Exchanger Systems by the Method of Gauss Convolution, Electronic modeling, vol.22, 6, (2000), pp. 19–25 (in Russian) Engineering Simulation, vol.18, pp. 727–734, 2001 (English translation)
[17] Fedorov M.: Steady-State Simulation of Heat Exchanger Networks, Electronic modeling, vol.24, 1, pp. 101–111, 2002 (in Russian)
[18] Filho L.O.F., Queiroz E.M., Costa A.L.H.: A Matrix Approach for Steady-State Simulation of Heat Exchanger Networks, Applied Thermal Eng 27, pp. 2385–2393, 2007
[19] Fedorov M.: Thermal Modes Simulation of Heat Exchanger Networks Having Turbo-machines, Proceedings of the Int. Conf. On Marine Technology IV, WIT Press, Southampton, pp. 309–314, 2001
[20] http://openmp.org. OpenMP Standard.
[21] Cheng N.: Formulas for friction factor in transitional regimes, Journal of Hydraulic Engineering, 134(9), pp 1357–1362, 2008
[22] Kanevets G.E.: Heat Exchangers and Heat Exchanger Systems, Naukowa Dumka, Kiev, 1982 (in Russian)
[23] Bialecki R.A., Kruczek T.: Frictional Diathermal Flow of Steam in a Pipeline, Chem. Eng. Sc., vol.51, 19, pp. 4369–4378, 1996
[24]
[25] Kays W. M., London A. L.: Compact Heat Exchangers, 3rd ed., McGraw-Hill, New York, 1984
[26] Dennis J., Schnabel R.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations, SIAM, Philadelphia, 1996
[27] Rice J.R.: Matrix computations and mathematical software, McGraw-Hill, New York, 1996
[28] Čiegis Raim., Čiegis Rem., Meilūnas M., Jankevičiūtė G., Starikovičius V.: Parallel mumerical algorithm for optimization of electrical cables, Mathematical Modelling and Analysis, vol.13, 4, pp. 471–482, 2008

***Authors**: Ph.D. Mykhailo Fedorov, Department of Software Engineering, Faculty of Computer Science, West Pomeranian University of Technology, ul. Zolnierska 49, 71-210 Szczecin, Poland, email: mfedorov@wi.zut.edu.pl*