Procedia Computer Science

Volume 80, 2016, Pages 834–844

ICCS 2016. The International Conference on Computational Science

Scalability of direct solver for non-stationary Cahn-Hilliard simulations with linearized time integration scheme

M. Woźniak[1], M. Smołka[1], A. Cortes[2], M. Paszyński[1], and R. Schaefer[1]

[1] AGH University of Science and Technology, Krakow, Poland
[2] King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

Abstract

We study the features of a new mixed integration scheme dedicated to solving the non-stationary variational problems. The scheme is composed of the FEM approximation with respect to the space variable coupled with a 3-leveled time integration scheme with a linearized right-hand side operator. It was applied in solving the Cahn-Hilliard parabolic equation with a nonlinear, fourth-order elliptic part. The second order of the approximation along the time variable was proven. Moreover, the good scalability of the software based on this scheme was confirmed during simulations. We verify the proposed time integration scheme by monitoring the Ginzburg-Landau free energy. The numerical simulations are performed by using a parallel multi-frontal direct solver executed over STAMPEDE Linux cluster. Its scalability was compared to the results of the three direct solvers, including MUMPS, SuperLU and PaSTiX.

Keywords: isogeometric analysis, Cahn-Hilliard equations, non-stationary problems, multi-frontal parallel direct solver, parallel efficiency and speedup

1 Introduction

In this paper we analyze the numerical properties of linearized time integration scheme concerning non-stationary Cahn-Hilliard equations proposed in [15]. We show that scheme is of the second order. We apply the scheme for numerical solution of Cahn-Hilliard equations [10, 11], but this time we split the fourth order Cahn-Hilliard equations into a system of two second order PDEs, following [12].

In the space domain the solution of the split Cahn-Hilliard equations is approximated by means of the isogeometric finite element method, utilizing B-spline basis functions [9]. The linearized version of the Crank-Nicolson integration scheme, that is unconditionally stable, is utilized for the time integration.

Our time integration scheme has been implemented within PETIGA toolkit [7], a part of the PETSc library [4, 5, 6]. The linearized system of equations is solved in every time step by using a parallel multi-frontal direct solver, executed over a STAMPEDE Linux cluster. For the numerical solution we test three multi-frontal direct solvers, including MUMPS [1, 2, 3],

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.373

SuperLU [17, 14] and PaSTiX [13]. We test the scalability of these solvers on the STAMPEDE linux cluster. We extend the results of sequential tests [8] performed for the simple Laplace problem. For efficient parallelization of the integration process, we refer to [16].

The structure of the paper is the following. We start with introduction of the strong formulation in Secton 2.1. It is followed by the two weak formulations in Sections 2.2 and 2.3, the first one L^2 in time, the second one C^1 in time. Later, we introduce the linearized time integration scheme in Section 2.4. We proof its second order in Section 3. We conclude the paper with numerical results discussed in Section 4. Finally, we perform numerical simulations with parallel multi-frontal direct solver executed over STAMPEDE Linux cluster. We test the scalability of the three direct solvers, including MUMPS, SuperLU and PaSTiX in Section 4.1.

2 Weak formulations of the Cahn-Hilliard equations

2.1 Strong formulation

This section presents the derivation of the weak form of Cahn-Hilliard equations based on [10]. We consider the following Cauchy problem: Find $u \in C^1(0,T;C^4(\Omega))$ such that

$$u_t = \nabla \cdot (B(u)\nabla\left(-\gamma\Delta u + \Psi'(u)\right)) \text{ on } [0,T]\times\Omega \text{ and } u(0,x) = u_0(x) \text{ on } \Omega, \tag{1}$$

where Ω is an open subset of $\mathbb{R}^n$, $n = 2, 3$ with smooth boundary and $\gamma > 0$ is a constant. u is usually interpreted as the difference of the two fluid phase concentrations, hence $u \in [-1, 1]$. In such a case $B(u) \geq 0$ is the diffusional mobility, whereas $\Psi(u)$ is the homogeneous free energy. A crucial notion in the theory of Cahn-Hilliard equations is the Ginzburg-Landau free energy

$$\mathcal{E}(u) = \int_\Omega \left(\frac{\gamma}{2}|\nabla u|^2 + \Psi(u)\right) dx. \tag{2}$$

As for the boundary conditions for (1), we choose the following ones:

$$\mathbf{n}\cdot(B(u)\nabla\left(-\gamma\Delta u + \Psi'(u)\right)) = 0, \quad \frac{\partial u}{\partial \mathbf{n}} = 0 \quad \text{on } [0,T]\times\partial\Omega, \tag{3}$$

where $\mathbf{n}$ is the outer normal to $\partial\Omega$.

As in paper [10], we select the following functions B and Ψ:

$$\begin{aligned} B(u) &= 1-u^2, \\ \Psi(u) &= \frac{\theta}{2}\left((1+u)\log(1+u) + (1-u)\log(1-u)\right) + 1 - u^2, \end{aligned} \tag{4}$$

where $\theta = 1.5$. Similarly to [12] we split (1) into two second-order equations

$$\begin{cases} u_t = \nabla\cdot(B(u)\nabla v) \\ v = \Psi'(u) - \gamma\Delta u. \end{cases} \tag{5}$$

In this case the boundary conditions have the following form

$$B(u)\frac{\partial v}{\partial \mathbf{n}} = 0, \quad \frac{\partial u}{\partial \mathbf{n}} = 0. \tag{6}$$

2.2 Weak formulation L^2 in time

The following weak formulation is presented in [10] together with the respective existence theorem. We seek $u \in L^2(0,T;H^2(\Omega)) \cap L^\infty(0,T;H^1(\Omega)) \cap C([0,T];L^2(\Omega))$ such that $u_t \in L^2(0,T;(H^1(\Omega))')$ and $B(u)\nabla\left(-\gamma\Delta u + \Psi'(u)\right) \in L^2(\Omega_T;\mathbb{R}^n)$, that satisfies (1) in the following sense

$$\int_0^T \langle u_t(t), \zeta(t)\rangle dt = -\int_{\Omega_T} \left[\gamma\Delta u \nabla\cdot(B(u)\nabla\zeta) + (B\Psi'')(u)\nabla u\cdot\nabla\zeta\right] dxdt, \tag{7}$$

for arbitrary $\zeta \in L^2(0,T;H^1(\Omega))$ such that $\nabla\zeta \in L^2(0,T;H^1(\Omega;\mathbb{R}^n)) \cap L^\infty(\Omega_T;\mathbb{R}^n)$ and $\nabla\zeta\cdot\mathbf{n} = 0$.

2.3 Weak formulation C^1 in time

Following [11, 10] we introduce the space differential operator $A : H^2(\Omega) \to (H^2(\Omega))'$, such that

$$\langle A(u), w\rangle = \int_\Omega \left[\gamma\Delta u\nabla\cdot(B(u)\nabla w) + (B\Psi'')(u)\nabla u\cdot\nabla w\right] dx. \tag{8}$$

For technical purposes, we introduce the simple dual operator $\tau : H^1(\Omega) \to (H^1(\Omega))'$ such that

$$\langle \tau(u), w\rangle = \int_\Omega u\cdot w\, dx, \quad \forall w \in H^1(\Omega). \tag{9}$$

The time-derivative operator $\cdot_t : C^1(0,T;H^1(\Omega)) \to C(0,T;(H^1(\Omega))')$ is defined in the following way

$$\langle u_t(t), w\rangle = \left\langle \tau\left(\frac{\partial u}{\partial t}(t)\right), w\right\rangle, \quad \forall w \in H^1(\Omega),\ \forall t \in [0,T]. \tag{10}$$

Then we look for $u \in C^1(0,T;H^2(\Omega))$ such that

$$\langle u_t(t), w\rangle + \langle A(u(t)), w\rangle = 0 \ \ \forall w \in H^1(\Omega), \ \ \forall t \in [0,T] \ \text{ and } \ u(0,x) = u_0(x) \text{ a.e. on } \Omega. \tag{11}$$

For the numerical convenience and similarly to [12, 10] we can split equation (11) into the following system:

$$\begin{cases} \langle u_t, w\rangle & = -\int_\Omega B(u)\nabla u\cdot\nabla w\ dx \\ \int_\Omega v\ z\ dx & = \int_\Omega \Psi'(u)\ z\ dx + \gamma\int_\Omega \nabla u\cdot\nabla z\ dx, \end{cases} \tag{12}$$

with arbitrary $w, z \in H^1(\Omega)$.

2.4 Finite difference schemes for semi-continuous variational equation

We introduce the network $\{t_0 = 0, t_1, \ldots, t_K = T\} \subset [0,T]$. Let $U = H^2(\Omega)$. Three finite difference schemes will be considered. The first is the forward Euler scheme:

$$\left\langle \tau\left(\frac{u^{i+1}-u^i}{t_{i+1}-t_i}\right), w\right\rangle + < A(u^i), w >= 0, \quad \forall w \in U, \quad i = 1,\ldots,K, \tag{13}$$

where $u^0 \in U$ is the initial condition, and $u^i = u(t_i)$. The second scheme is the Crank-Nicolson method:

$$\left\langle \tau\left(\frac{u^{i+1}-u^i}{t_{i+1}-t_i}\right), w\right\rangle + \frac{1}{2} < (A(u^{i+1}) + A(u^i)), w >= 0, \quad \forall w \in U, \quad i = 1,\ldots,K. \tag{14}$$

Finally, if $A \in C^1(U)$ we can use a linearized three-level scheme

$$\left\langle \tau \left(\frac{u^{i+2} - u^i}{t_{i+2} - t_i} \right), w \right\rangle + \left\langle \left(\frac{1}{2} DA|_{u^{i+1}} (u^{i+2} + u^i - 2u^{i+1}) + A(u^{i+1}) \right), w \right\rangle = 0, \quad \forall w \in U, \;\; i = 1, \ldots, K. \tag{15}$$

3 Order of the linearized 3-leveled schema

Consider the following autonomous ODE

$$y' = f(y(t)). \tag{16}$$

Let us assume that we know the solution of (16) on interval $[t_0 - h, t_0]$. Take $r > t_0$. If f is C^2, then we can write the Taylor's formula for f in the following form

$$f(w + \Delta w) = f(w) + f'(w)\Delta w + R(w, \Delta w), \tag{17}$$

$$R(w, \Delta w) = \left(\int_0^1 (1 - s) f''(w + s\Delta w) ds \right) (\Delta w)^2. \tag{18}$$

Applying (17)-(18) and assuming R is small enough we can write down a linearized version of (16) for an initial condition $y(t_0)$ and for $r > t_0$

$$y'(r + \xi) \approx f(y(r)) + f'(y(r)) \left(y(r + \xi) - y(r) \right). \tag{19}$$

For (19) we write down the 2-leveled Crank-Nicolson scheme subsequently linearizing the right-hand side f in central points of subintervals

$$\frac{y(t + 2h) - y(t)}{2h} \approx \frac{1}{2} \Big(f(y(t+h)) + f'(y(t+h)) \big(y(t) - y(t+h) \big) + f(y(t+h)) + f'(y(t+h)) \big(y(t+2h) - y(t+h) \big) \Big)$$

$$= \frac{1}{2} \Big(2f(y(t+h)) + f'(y(t+h)) \big(y(t) + y(t+2h) - 2y(t+h) \big) \Big). \tag{20}$$

Hence

$$y^{i+2} = y^i + h \Big(2f(y^{i+1}) + f'(y^{i+1}) \big(y^i + y^{i+2} - 2y^{i+1} \big) \Big). \tag{21}$$

Theorem 1. *If f is C^4 then scheme (21) has order 2.*

Proof. We assume that y is computed exactly at t_i and t_{i+1}. Denote by $r_i(h)$ the approximation error in time step $t_{i+2} = t_i + 2h$. Then

$$r_i(h) = y(t_i + 2h) - y^{i+2}$$

$$= y(t_i + 2h) - y^i - h \Big(2f(y^{i+1}) + f'(y^{i+1}) \big(y^i + y^{i+2} - 2y^{i+1} \big) \Big)$$

$$= y(t_i + 2h) - y(t_i) - h \Big(2f(y(t_{i+1})) + f'(y(t_{i+1})) \big(y(t_i) + y(t_{i+2}) - 2y(t_{i+1}) \big) \Big)$$

$$= y(t_i + 2h) - y(t_i) +$$

$$- h \Big(2f(y(t_i + h)) + f'(y(t_i + h)) \big(y(t_i) - y(t_i + h) \big) + f'(y(t_i + h)) \big(y(t_i + 2h) - y(t_i + h) \big) \Big).$$

From (17) and (16) it follows that

$$r_i(h) = y(t_i + 2h) - y(t_i) - hy'(t_i) + hR_1(h) - hy'(i_i + 2h) + hR_2(h),$$

where

$$R_1(h) = R\big(y(t_i + h), y(t_i) - y(t_i + h)\big), \tag{22}$$

and

$$R_2(h) = R\big(y(t_i + h), y(t_i + 2h) - y(t_i + h)\big). \tag{23}$$

To conclude the proof we need to show that r_i vanishes at 0 along with its first and second derivative. But it is easy to see that

$$r_i(0) = 0.$$

Next

$$r_i'(h) = 2y'(t_i+2h)-y'(t_i)-y'(t_i+2h)-2hy''(t_i+2h)+h(R_1'(h)+R_2'(h))+R_1(h)+R_2(h), \tag{24}$$

and

$$r_i''(h) = 4y''(t_i + 2h) - 2y''(t_i + 2h) - 2y''(t_i + 2h) - 4hy'''(t_i + 2h)+ \\ + h(R_1''(h) + R_2''(h)) + 2(R_1'(h) + R_2'(h)). \tag{25}$$

But because R has the form

$$R(h) = a(h)h^2,$$

from (22) and (23) it follows that

$$R_j(0) = R_j'(0) = R_j''(0) = 0.$$

Therefore, from (24), (25) we have

$$r_h'(0) = r_h''(0) = 0.$$

□

It is easy to see that Theorem 1 can be used to study the order of scheme (15) applied to the Cahn-Hilliard equation in the form (11). This equation can be rewritten in form (16) by replacing $y(t)$, $y'(t)$ and $f(y(t))$ with $u(t)$, $u_t(t)$ and $-A(u(t))$, respectively. The state space of the new equation is now $(H^2(\Omega))'$. Hence, it follows that scheme (15) applied to equation (11) has order 2 if B has class C^4 and Ψ has class C^5, which obviously holds in the sequel. Of course, if a spatial approximation of dimension n is applied, equation (11) is reduced to the system of n ordinary differential equations of first order with respect to the time variable t, so $y, y' : [0, T] \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^n$.

4 Numerical results

We performed the numerical simulations of our modified time integration scheme with PETIGA toolkit installed over STAMPEDE [18] Linux cluster from Texas Advanced Computing Center. We have executed our tests up to 256 nodes, 1 core per node, 32 GB of RAM per node.

We solved the Cahn-Hilliard equation over two dimensional domain with the parameters setup identical to those in [15]:

$\Omega = (0,1)^2$, $B(u) = 1 - u^2$, $\Psi'(u) = \frac{1}{2\theta} log \frac{u}{1-u} + 1 - 2u$, where $\theta = 1.5$ is a dimensionless number which represents the ratio between the critical temperature Ξ_c (the temperature at which the two phases attain the same composition) and the absolute temperature Ξ, $\gamma = \frac{1}{N^2}$ where N is the mesh size in one direction with periodic boundary conditions and $u_0(x) = 2\bar{c} - 1 + r$, where $\bar{c} = 0.63$ is the average concentration, and r is the scalar random perturbation with the uniform distribution $\mathcal{U}(\Omega)$.

In our simulations we utilize three different multi-frontal direct solvers for the solution of the linear system at every time step. We performed our tests over 2D grids with 256 times 256 elements. We used different polynomial orders of approximation, from $p = 1, 2, ..., 8$, with C^0 global continuity, possible to apply since we have split the Cahn-Hilliard equations into the system of second order PDE.

The snapshots from the Cahn-Hilliard simulations with our time integration scheme are presented in Figures 1-2.

The simulation stability has been controlled by monitoring the Ginzburg-Landau free energy, that has been constantly decreasing during the simulation.

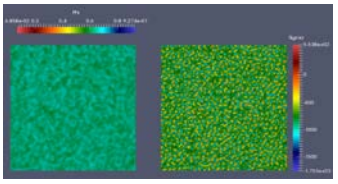
4.1 Comparison of the scalability of the parallel multi-frontal direct solvers

We have performed the comparison of the scalability of the three multi-frontal direct solvers executed over a distributed memory Linux cluster. We have utilized one processor per node, up to 256 nodes on the STAMPEDE Linux cluster. We have tested the three solvers, MUMPS [1, 2, 3], SuperLU [17, 14] and PaSTiX [13], available through PETSC interface [4, 5, 6]. We have executed the tests for the number of processors increasing from 1 till 256, and for the polynomial B-spline orders of approximations varying from $p = 1, ..., 8$ with global C^{p-1} solution. The results of the comparison are presented in Figures 3-5.

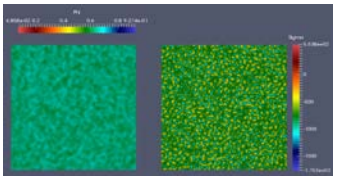
It is the open problem what we really gain by using higher order B-splines, since we utilize the split systems. It may be a matter of our future experiments. The scalability results will be identical that one performed for the Laplace equations, except here we have a system of two equations, so there are two unknowns per mesh node.

5 Conclusions

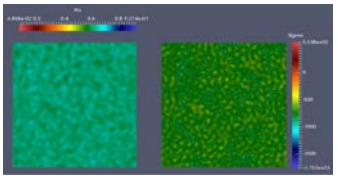
We have studied a new mixed integration scheme dedicated to solving the non-stationary variational problems, composed of the FEM approximation with respect to the space variable coupled with a The second order of the approximation along the time variable has been proven for the case of Cahn-Hilliard equations. We have tested the numerical scalability of the multi-frontal direct solvers applied at each time step of the time integration scheme under consideration. We have performed parallel simulations with three direct solvers, including MUMPS, SuperLU and PaSTiX, available through PETSc toolkit behind the PETIGA framework. From the above experiments we may conclude that increasing the degree of approximation polynomials p improves the parallel scalability of the multi-frontal direct solver. For the C^0 continuity of the solution, the computational cost of the multi-frontal solver is of the order of $\mathcal{O}(N^3/2p^3 + Np^6)$. The second term is related to the static condensation that can be fully parallelized. The computations related to the first term are harder to parallelize. With increasing p, this static condensation cost is growing, and the parallelization is becoming more efficient.



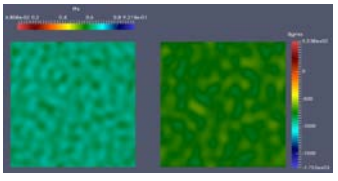
(a) Snapshot 1



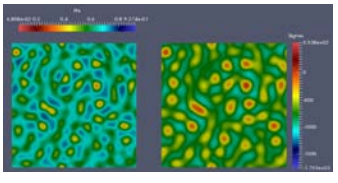
(b) Snapshot 2



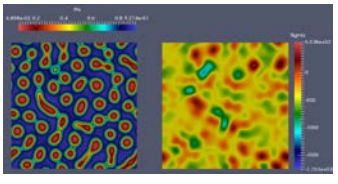
(c) Snapshot 3



(d) Snapshot 4

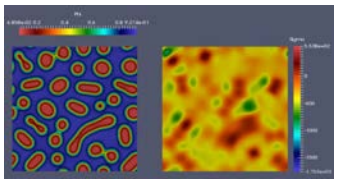


(e) Snapshot 5

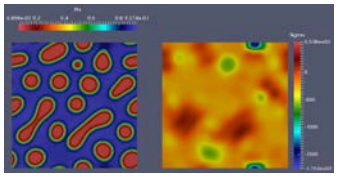


(f) Snapshot 6

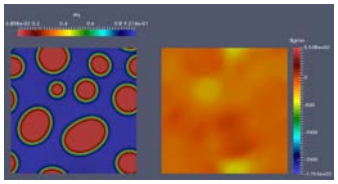
Figure 1: Snapshots from the split Cahn-Hilliard simulation solution by using the linearized scheme.



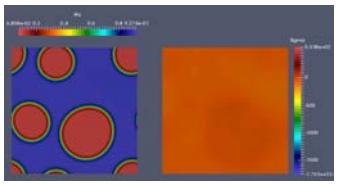
(a) Snapshot 1



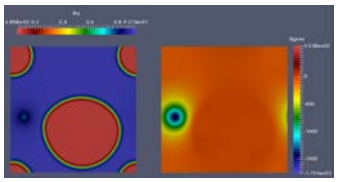
(b) Snapshot 2



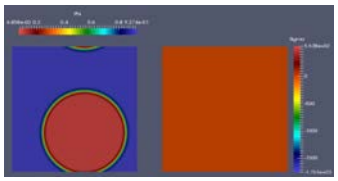
(c) Snapshot 3



(d) Snapshot 4



(e) Snapshot 5



(f) Snapshot 6

Figure 2: Further snapshots from the split Cahn-Hilliard simulation solution by using the linearized scheme.

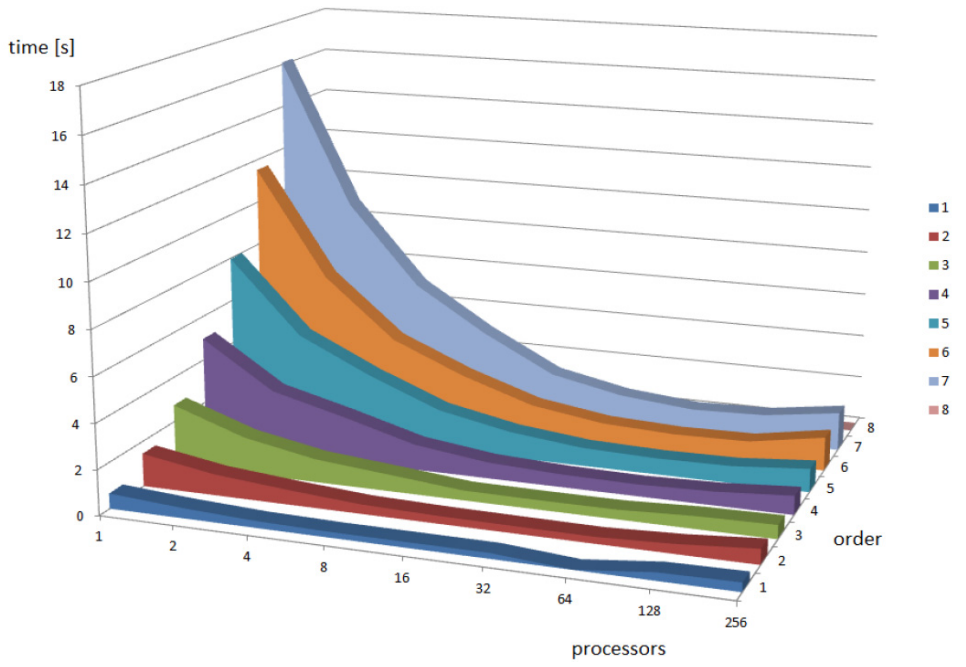


Figure 3: MUMPS solver. Comparison of scalability for C^{p-1} for different number of processors and different polynomial order of approximation, for problem size $256 * 256$ elements

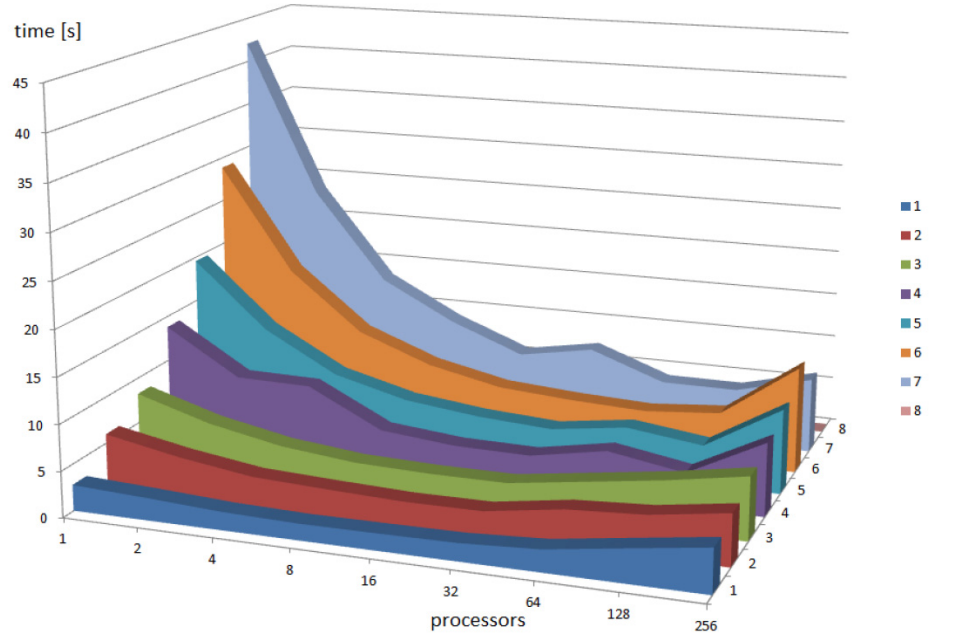


Figure 4: SuperLU solver. Comparison of scalability for C^{p-1} for different number of processors and different polynomial order of approximation, for problem size $256 * 256$ elements

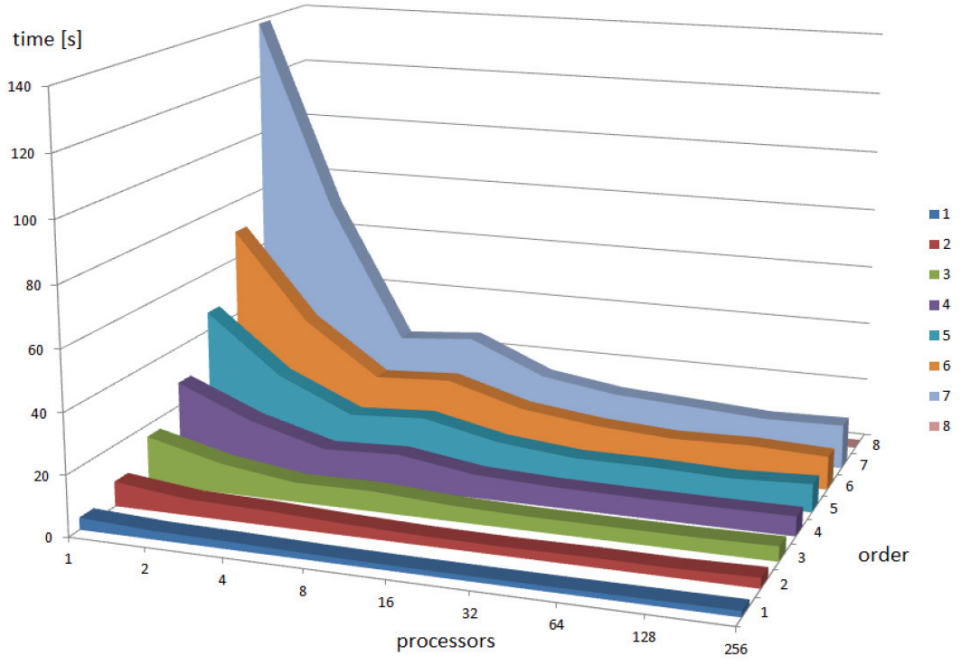


Figure 5: PaSTiX solver. Comparison of scalability for C^{p-1} for different number of processors and different polynomial order of approximation, for problem size $256 * 256$ elements

Acknowledgments

The work of MW, MS, MP, RS presented in this paper concerning the development of Cahn-Hilliard scheme has been supported by National Science Centre, Poland grant no. DEC-2012/ 07/ B/ST6/ 01229. The visit of AC and his work concerning the PETIGA solver interface has been supported by National Science Centre, Poland grant no. DEC-2012/06/M/ST1/00363.

Bibliography

References

[1] P. R. Amestoy , I. S. Duff, *M*ultifrontal parallel distributed symmetric and unsymmetric solvers, Computer Methods in Applied Mechanics and Engineering, 184 (2000) 501-520.

[2] P. R. Amestoy, I. S. Duff, J. Koster, J.Y. L'Excellent, *A* fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM Journal of Matrix Analysis and Applications, 1(23) (2001) 15-41.

[3] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, *H*ybrid scheduling for the parallel solution of linear systems, Computer Methods in Applied Mechanics and Engineering, 2(32) (2001) 136-156.

[4] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. Curfman McInnes, K. Rupp, B. F. Smith, H. Zhang, *PETSc* Web Page, *http://www.mcs.anl.gov/petsc* (2014)

[5] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. Curfman McInnes, K. Rupp, B. F. Smith, H. Zhang, *PETSc User Manual*, Argonne National Laboratory ANL-95/11 - Revision 3.4 (2013)

[6] S. Balay, W. D. Gropp, L. Curfman McInnes, B. F. Smith, *Efficient Management of Parallelism in Object Oriented Numerical Software Libraries*, Modern Software Tools in Scientific Computing, Editors E. Arge, A. M. Bruaset and H. P. Langtangen (1997) Birkhä user Press.

[7] N. Collier, L. Dalcin, V. M. Calo., *PetIGA: High-performance isogeometric analysis*, arxiv, (1305.4452), (2013) http://arxiv.org/abs/1305.4452

[8] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, V. Calo, The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers, Computer Methods in Applied Mechanics and Engineering (213-216) (2012) 353-361.

[9] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Unification of CAD and FEA* John Wiley and Sons, (2009)

[10] C. M. Elliott, H. Garcke, *On the Cahn-Hilliard equation with degenerate mobility* SIAM Journal of Mathematical Analysis, 27 (1996) 404-423.

[11] H. Gomes, V. M. Calo, Y. Bazileves, T.J.R. Hughes, *Isogeometric analysis of the Cahn-Hilliard phase-field model*, Computer Methods in Applied Mechanics and Engineering, 197 (2008) 4333-4352.

[12] H. Gomes, T.J.R. Hughes, *Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models*, Journal of Computational Physics, 230 (2011) 5310-5327.

[13] P. Hnon, P. Ramet, J. Roman, *P*aStiX: A High-Performance Parallel Direct Solver for Sparse Symmetric Definite Systems, Parallel Computing, 28(2) (2002) 301-321.

[14] X.S. Li, J.W. Demmel, J.R. Gilbert, iL. Grigori, M. Shao, I. Yamazaki, *S*uperLU Users' Guide, Lawrence Berkeley National Laboratory, LBNL-44289 *http://crd.lbl.gov/ xiaoye/SuperLU/* (1999).

[15] R. Schaefer, M. Smolka, L. Dalcin, M. Paszynski, A new time integration scheme for Cahn-Hilliard equations, Procedia Computer Science, 51 (2015) 1003-1012.

[16] M. Woźniak, Fast GPU integration algorithm for isogeometric finite element method solvers using task dependency graphs, Journal of Computational Science, (2015) in press. DOI: 10.1016/j.jocs.2015.02.007

[17] Xiaoye S. Li, *A*n Overview of SuperLU: Algorithms, Implementation, and User Interface, TOMS Transactions on Mathematical Software, 31(3) (2005) 302-325.

[18] https://portal.tacc.utexas.edu/user-guides/stampede