Parallel computing for interface problems

Li Luo

Department of Mathematics
The Hong Kong University of Science and Technology, Hong Kong

Jan 5, 2017

Rapid development of supercomputers

Tianhe2
Top 1 of 2013.06

- Peak: 55 PF
- HPL: 33.8 PF
- Cores: 3.12 M
- Arch: CPU-MIC

Sunway TaihuLight
Top 1 of 2016.06

- Peak: 125 PF
- HPL: 93 PF
- Cores: 10.6 M
- Arch: Sunway 26010 chip
- Gordon Bell Prize: 2016.11 – Weather Patterns

The coupled Cahn-Hilliard Navier-Stokes equations:

$$\frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla)\,\phi = \mathcal{L}_d \Delta \mu, \qquad \text{in} \quad \Omega, \qquad (0.1)$$

$$\mu = -\epsilon \Delta \phi - \frac{\phi}{\epsilon} + \frac{\phi^3}{\epsilon}, \qquad \text{in} \quad \Omega, \qquad (0.2)$$

$$Re\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) = -\nabla p + \nabla \cdot (\eta D(\mathbf{u})) + \mathcal{B}\mu \nabla \phi + \mathbf{g}_{ext}, \qquad \text{in} \quad \Omega, \qquad (0.3)$$

$$\nabla \cdot \mathbf{u} = 0, \qquad \text{in} \quad \Omega. \qquad (0.4)$$

$$\rho = \frac{1+\phi}{2} + \lambda_\rho \frac{1-\phi}{2}, \qquad \eta = \frac{1+\phi}{2} + \lambda_\eta \frac{1-\phi}{2},$$

The generalized Navier boundary condition:

$$\frac{\partial \phi}{\partial t} + u_\tau \partial_\tau \phi = -\mathcal{V}_s L(\phi), \qquad \text{on} \quad \Gamma_w, \qquad (0.5)$$

$$(\mathcal{L}_s l_s)^{-1} u_\tau = \mathcal{B} L(\phi) \partial_\tau \phi / \eta - \mathbf{n} \cdot D(\mathbf{u}) \cdot \boldsymbol{\tau}, \qquad \text{on} \quad \Gamma_w, \qquad (0.6)$$

$$u_n := \mathbf{u} \cdot \mathbf{n} = 0, \qquad \partial_n \mu = 0, \qquad \text{on} \quad \Gamma_w. \qquad (0.7)$$

Cahn-Hilliard system:

- 2×2 block element stiffness matrix for ϕ_h^{n+1} and μ_h^{n+1}: non-symmetric

$$\begin{pmatrix} K_{\phi\phi}^n & K_{\phi\mu}^n \\ K_{\mu\phi}^n & K_{\mu\mu}^n \end{pmatrix} \text{ and } \begin{pmatrix} F_\phi^n \\ F_\mu^n \end{pmatrix} \tag{0.8}$$

- If treat the nonlinear term explicitly: constant coefficients
- The unknowns are ordered node by node so that the nonzeros are placed as close as possible in the matrix:

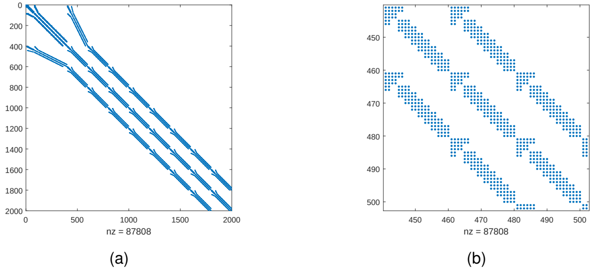


Figure: (a) Matrix pattern of the Cahn-Hilliard system obtained using the linear decoupled scheme on a structured mesh. (b) An enlarged view.

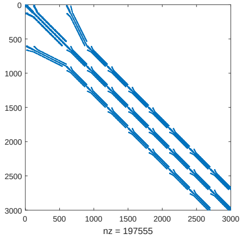
Velocity system of Navier-Stokes equations:

- In complex domains, one should combine all the components of

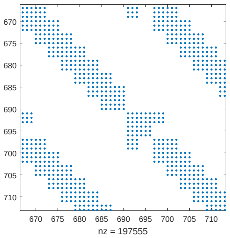
$$\mathbf{u} = (u_x, u_y, u_z), \qquad \mathbf{n} = (n_x, n_y, n_z), \qquad \boldsymbol{\tau} = (\tau_x, \tau_y, \tau_z)$$

together into calculation, leading to 3×3 block element stiffness matrices.

$$\begin{pmatrix} K^n_{u_x u_x} & K^n_{u_x u_y} & K^n_{u_x u_z} \\ K^n_{u_y u_x} & K^n_{u_y u_y} & K^n_{u_y u_z} \\ K^n_{u_z u_x} & K^n_{u_z u_y} & K^n_{u_z u_z} \end{pmatrix} \text{ and } \begin{pmatrix} F^n_{u_x} \\ F^n_{u_y} \\ F^n_{u_z} \end{pmatrix} \tag{0.9}$$



(a)



(b)

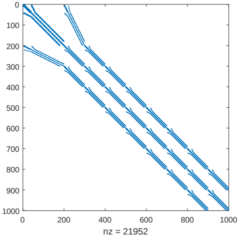
Pressure system of Navier-Stokes equations:

$$\left(\nabla(p_h^{n+1} - p_h^n), \nabla q_h\right) = -\frac{\bar{\rho}}{\delta t} Re(\nabla \cdot \mathbf{u}_h^{n+1}, q_h). \qquad (0.10)$$

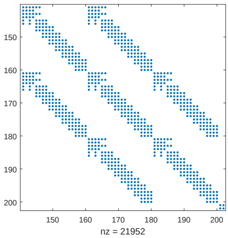
The resulting element stiffness matrix is symmetric positive definite.

$$K_p^n(i,j) = (\nabla\chi_i, \nabla\chi_j),$$

$$F_p^n(i) = -\frac{\bar{\rho}}{\delta t} Re(\nabla \cdot \mathbf{u}_h^{n+1}, \chi_i) + (\nabla p_h^n, \nabla\chi_i).$$



(c)



(d)

Three linear systems for $(\phi_h^{n+1}, \mu_h^{n+1})$, $\mathbf{u}_h^{n+1}$ and p_h^{n+1} respectively, i.e.

$$A_h x = b_h \tag{0.11}$$

Features of A_h:

- A_h is very large for 3D problems
- A_h is ill-conditioned, $\kappa(A_h) = \frac{\max_i \lambda_i}{\min_i \lambda_i}$
- A_h is sparse

Comparison:

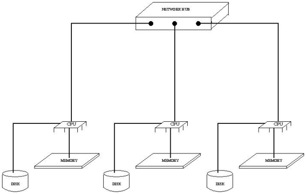
- Direct methods: Gaussian elimination or its variation
  - Exact solution can be obtained if without rounding error
  - Great cost on memory
- Fast algorithms: FFT and cyclic reduction methods
  - Not suitable for general matrices and distributed memory system
- Iterative methods:
  - Lower memory requirement and generally fewer arithmetic operations
  - Easy to implement in parallel

Preconditioned Conjugate Gradient Method

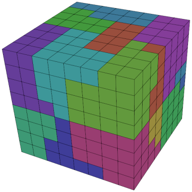
$$
\begin{aligned}
&r_h^0 = b_h - A_h x_h^0 \\
&M_h z_h^0 = r_h^0 \\
&p_h^0 = z_h^0 \\
&\text{For } k = 0, 1, \ldots, N \\
&\quad \alpha_k = \frac{(z_h^k, r_h^k)}{(p_h^k, A_h p_h^k)} \\
&\quad x_h^{k+1} = x_h^k + \alpha_k p_h^k \\
&\quad r_h^{k+1} = r_h^k - \alpha_k A_h p_h^k \\
&\quad M_h z_h^{k+1} = r_h^{k+1} \\
&\quad \beta_{k+1} = \frac{(z_h^{k+1}, r_h^{k+1})}{(z_h^k, r_h^k)} \\
&\quad p_h^{k+1} = z_h^{k+1} + \beta_{k+1} p_h^k
\end{aligned}
$$

- Memory: Store four vectors $(x_h; z_h; p_h; r_h)$ and possibly a sparse matrix
- $\kappa(M_h^{-1} A_h) \ll \kappa(A_h)$, or M_h is spectrally close to A_h
- $M_h^{-1} x_h$ has to be easy to compute, easy to parallelize

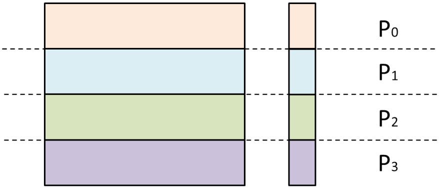
Distributive Computing based on MPI



Distributed Memory

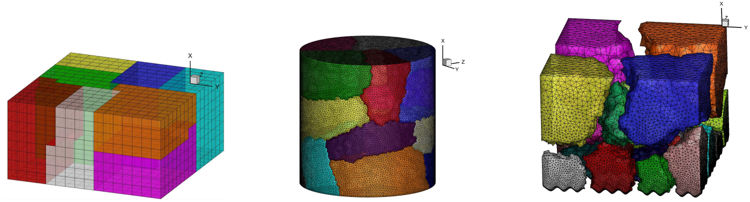


Domain Decomposition



Subdivision of matrix and vector

- Interface problem requires a very fine mesh to capture the interface, especially in 3D as $\epsilon \to 0$.
- A partition of the domain $\Omega_h = \Omega_{h,1} \cup \cdots \cup \Omega_{h,np}$ where $\Omega_{h,i} \cap \Omega_{h,j} = \emptyset$ for all $i \neq j$.
- Meshes are partitioned using Metis on a relatively coarse level and are refined sufficiently for computation.



(e) (f) (g)

Figure: (a) A sample partition of a structured mesh into 8 subdomains, (b) a partition of an unstructured mesh into 16 subdomains, (c) a sample partition into 24 subdomains.

Right preconditioning of the linear system

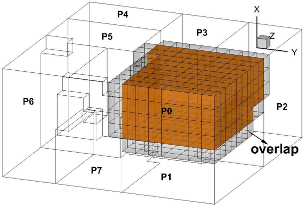
$$A_h M_h^{-1} y_h = b_h, \text{ with } x_h = M_h^{-1} y_h, \tag{0.12}$$

A geometrical restrict additive Schwarz (RAS) [Cai1999] preconditioned GMRES method is employed to solve the implicit systems of (ϕ, μ) and **u**.

$$b_{h,i}^{\delta} = R_{h,i}^{\delta} b_h = (I \quad 0) \begin{pmatrix} b_{h,i}^{\delta} \\ b \backslash b_{h,i}^{\delta} \end{pmatrix},$$

$$M_h^{-1} = \sum_{i=1}^{np} (R_{h,i}^{0})^T (A_{h,i})^{-1} R_{h,i}^{\delta},$$

$$A_{h,i} = R_{h,i}^{\delta} A_h (R_{h,i}^{\delta})^T.$$



Subdomain solver: Incomplete LU (ILU) factorization

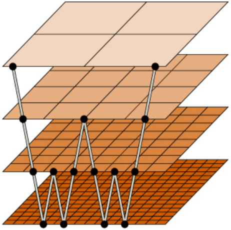
$$A_{h,i} = L_{h,i} U_{h,i} + P_{h,i},$$

where $P_{h,i}$ satisfies some criteria such as preserving certain sparsity patterns. Details about the ILU(p) factorization can be found in [Saad2003].

An algebraic multigrid (AMG) preconditioned CG method is used to solve the pressure Poisson system.
— BoomerAMG from Hypre library
— Smoothed Aggregation in PETSc library

AMG: $x_l = AMG_l(x_l, d_l)$:

0. If on the coarsest level, then:
   Solve $C_l x_l = d_l$ by Gaussian elimination, else:
1. Apply μ steps of smoothing to $C_l x = d_l$
2. Coarse grid correction:
   (a). Set $d_{l+1} = (I^l_{l+1})^T (d_l - C x_l)$ and $x_{l+1} = 0$
   (b). Solve the coarse problem $B_{l+1} x_{l+1} = d_{l+1}$
   by γ applications of $x_{l+1} = AMG_{l+1}(x_{l+1}, d_{l+1})$
   (c). Then correct the solution on the level l
   by $x_l \leftarrow x_l + I^l_{l+1} x_{l+1}$
3. Apply μ steps of smoothing to $C_l x = d_l$.



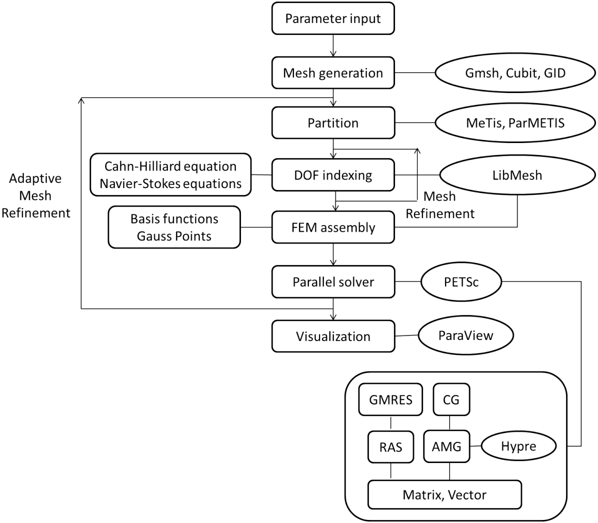
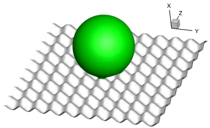


Figure: The overall solution procedure.

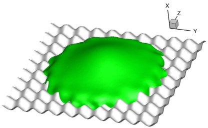
A droplet spreading on rough surface with unstructured mesh of 212,434,560 elements and 31,711,677 vertices. The scalability tests are performed on the Tianhe 2 supercomputer.

Table: The scalability test for the proposed solution algorithm. The average number of GMRES(CG) iterations, compute time per time step, and speedup for solving the Cahn-Hilliard system, the velocity system, and the pressure system.

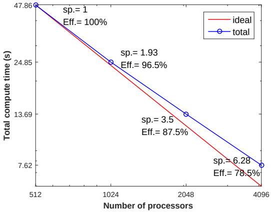
#unknowns	Cahn-Hilliard system 63,423,354			velocity system 95,135,031			pressure system 31,711,677		
np	GMRES	time	sp.	GMRES	time	sp.	CG	time	sp.
512	8.1	12.68	1	9.3	23.27	1	30	11.88	1
1,024	7.4	6.48	1.96	9.8	11.88	1.96	30.4	6.46	1.84
2,048	8.4	3.45	3.68	9.6	6.65	3.50	32.3	3.56	3.34
4,096	8.3	1.87	6.78	10.1	3.71	6.27	31.1	2.0	5.94



(a)



(b)

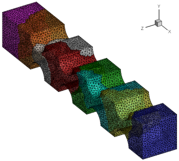


(c)

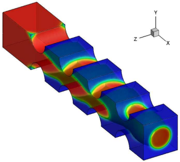
The two-phase bumpy channel flow with unstructured mesh of 301,412,352 elements and 51,270,353 vertices. The scalability tests are performed on the Tianhe 2 supercomputer.

Table: The scalability test for the two-phase bumpy channel flow case. The average number of GMRES (CG) iterations, compute time per time step, and speed up for solving the Cahn-Hilliard system, the velocity system, and the pressure system. “-" means the case fails to converge.

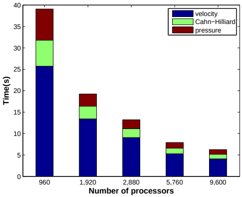
		Cahn-Hilliard system #unknowns=102,540,706			velocity system #unknowns=153,811,059			pressure system #unknowns=51,270,353			
np	subsolve	GMRES	time	sp.	GMRES	time	sp.	sweep	CG	time	sp.
1,920	ILU(2)	71.6	4.74	1	-	-	-	1	24.7	2.85	1
1,920	ILU(3)	20.5	2.96	1	28.1	13.43	1	2	21.2	3.41	1
1,920	ILU(4)	15.9	3.37	1	18.6	18.72	1	3	20.6	4.20	1
5,760	ILU(2)	79.6	2.72	1.74	-	-	-	1	24.7	1.28	2.23
5,760	ILU(3)	21.2	1.31	2.26	28.3	5.32	2.52	2	19.7	1.60	2.13
5,760	ILU(4)	17.1	1.46	2.31	18.9	7.19	2.60	3	20	1.78	2.36
9,600	ILU(2)	77.7	1.58	3	-	-	-	1	25.6	1.1	2.59
9,600	ILU(3)	22.2	1.07	2.76	33.4	4.09	3.86	2	21.3	1.47	2.32
9,600	ILU(4)	17.9	1.15	2.93	24.6	5.37	3.49	3	20.6	1.61	2.61



(d)



(e)

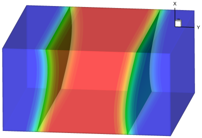


(f)

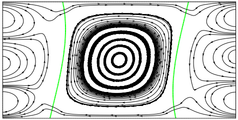
The two-phase Couette flow with structured mesh of 9,584,640 elements and 8,520,321 vertices. The scalability tests are performed on the Sunway TaihuLight supercomputer.

Table: The average number of iterations, compute time per time step, and speed up for solving the Cahn-Hilliard-velocity system

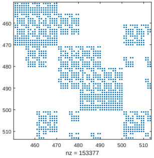
#unknowns	Cahn-Hilliard-velocity system 42,601,605					
subsolve	ILU(2)			rPBGS(2)		
np	GMRES	time	sp.	GMRES	time	sp.
512	33.3	103.95	-	114.2	65.48	-
1,024	34.5	55.79	1.86	121.3	26.09	2.51
2,048	35.9	30.05	3.46	129.2	14.00	4.68
4,096	38.9	17.42	5.97	137	8.14	8.04



(g)



(h)



(i)

Thank You