ACCELERATION OF IMPLICIT SCHEMES USING REDUCED MODELS AND GRID COMPUTING

Yuri Vassilevski

Institute of Numerical Mathematics
Russian Academy of Sciences
Moscow, Russia

MSU, 5.7.2012

Instead of introduction

movie

Complexity and parallelization

- OpenMP computation of droplet

  1100 time steps, $h = 8^{-1} - 512^{-1}$, no parallel solvers

# cores Intel Xeon 2.6 Ghz	2	4	8
Time, hours	30.50	22.84	18.815

- MPI visualization of flooding

  Povray-MPI, resolution 4096×3072 of single frame

# cores Intel Xeon 2.6 Ghz	15	30	60	120
Time, seconds	367	195	107	62

- Parallel computing technology should correspond to algorithm
- I proceed to an algorithm which fits to GRID computing

Complexity and parallelization

- OpenMP computation of droplet

  1100 time steps, $h = 8^{-1} - 512^{-1}$, no parallel solvers

# cores Intel Xeon 2.6 Ghz	2	4	8
Time, hours	30.50	22.84	18.815

- MPI visualization of flooding

  Povray-MPI, resolution 4096×3072 of single frame

# cores Intel Xeon 2.6 Ghz	15	30	60	120
Time, seconds	367	195	107	62

- Parallel computing technology should correspond to algorithm
- I proceed to an algorithm which fits to GRID computing

Complexity and parallelization

- OpenMP computation of droplet

  1100 time steps, $h = 8^{-1} - 512^{-1}$, no parallel solvers

# cores Intel Xeon 2.6 Ghz	2	4	8
Time, hours	30.50	22.84	18.815

- MPI visualization of flooding

  Povray-MPI, resolution 4096×3072 of single frame

# cores Intel Xeon 2.6 Ghz	15	30	60	120
Time, seconds	367	195	107	62

- Parallel computing technology should correspond to algorithm
- I proceed to an algorithm which fits to GRID computing

Complexity and parallelization

- OpenMP computation of droplet

  1100 time steps, $h = 8^{-1} - 512^{-1}$, no parallel solvers

# cores Intel Xeon 2.6 Ghz	2	4	8
Time, hours	30.50	22.84	18.815

- MPI visualization of flooding

  Povray-MPI, resolution 4096×3072 of single frame

# cores Intel Xeon 2.6 Ghz	15	30	60	120
Time, seconds	367	195	107	62

- Parallel computing technology should correspond to algorithm
- I proceed to an algorithm which fits to GRID computing

Acceleration of implicit schemes using reduced models and grid computing

Joint work with

Damien Tromeur-Dervout

University of Lyon 1

Acceleration of implicit schemes using reduced models and grid computing

Joint work with

Damien Tromeur-Dervout

University of Lyon 1

D.Tromeur-Dervout, Yu.V., Journal of Computational Physics, 219, 2006

D.Tromeur-Dervout, Yu.V., Advances in Engineering Software, 38, 2007

Inexact Newton backtracking method for $F(u) = 0$

S.Eisenstat, H.Walker,'90s

- Newton: $F'(u_k)s_k = -F(u_k)$, $u_{k+1} = u_k + s_k$

- Inexact Newton: $\|F(u_k) + F'(u_k)s_k\| \leq \eta_k \|F(u_k)\|$
  $\eta_k = |\|F(u_k)\| - \|F(u_{k-1}) + F'(u_{k-1})s_{k-1}\||/\|F(u_{k-1})\|$

- Inexact Newton Backtracking:
  while $\|F(u_k + s_k)\| > [1 - 10^{-4}(1 - \eta_k)]\|F(u_k)\|$
  update $s_k \leftarrow \theta s_k$ and $\eta_k \leftarrow 1 - \theta(1 - \eta_k)$
  $u_{k+1} = u_k + s_k$

$F'(u_k)v = \frac{1}{\delta}[F(u_k + \delta v) - F(u_k)]$

NITSOL H.Walker,'98

Fully implicit schemes

Backward Euler: $\frac{u^i - u^{i-1}}{\Delta t} + A(u^i) = g^i$, or

$$F^i(u^i) = 0$$

For $i = 1, \ldots$:

- set $u_0^i = u^{i-1}$
- solve $F^i(u^i) = 0$ by INB

Idea: get better u_0^i by POD and Reduced Model

Related works

- P.Fischer '98, M.Clemens et al.'03 — reconstruction from $u^{i-m}, \ldots, u^{i-1}$
- M.Rathinam, L.Petzold '03 — review of POD and RM
- D.Tromeur-Dervout, Yu.V. '05,'06,'07 — INB+POD+RM
- R.Marcovinovic, J.Jansen '07 — POD-RM for linear schemes

Proper orthogonal decomposition

What vector $v \in R^N$ is the most close to $\{u^i\}_{i=1}^n$:

$$v = \arg\min_{v \in R^N} \sum_{i=1}^{n} \|u^i - P_v u^i\|^2?$$

Proper orthogonal decomposition

What vector $v \in R^N$ is the most close to $\{u^i\}_{i=1}^n$:

$$v = \arg \min_{v \in R^N} \sum_{i=1}^{n} \|u^i - P_v u^i\|^2 ?$$

Generate correlation matrix $R = XX^T$, $X = \{u^i\}$

$$Rw_j = \lambda_j w_j, \quad \lambda_1 \geq \cdots \geq \lambda_N \geq 0$$

$$v = w_1 \;\rightarrow\; \sum_{i=1}^{n} \|u^i - P_v u^i\|^2 = \sum_{j=2}^{N} \lambda_j$$

Proper orthogonal decomposition

What vector $v \in R^N$ is the most close to $\{u^i\}_{i=1}^n$:

$$v = \arg\min_{v \in R^N} \sum_{i=1}^{n} \|u^i - P_v u^i\|^2?$$

Generate correlation matrix $R = XX^T$, $X = \{u^i\}$

$$Rw_j = \lambda_j w_j, \quad \lambda_1 \geq \cdots \geq \lambda_N \geq 0$$

$$v = w_1 \ \rightarrow \ \sum_{i=1}^{n} \|u^i - P_v u^i\|^2 = \sum_{j=2}^{N} \lambda_j$$

or generate $n \times n$-matrix $r = X^T X$, $X = \{u^i\}$

$$r\hat{w}_j = \mu_j \hat{w}_j, \quad \mu_1 \geq \cdots \geq \hat{\mu}_n \geq 0$$

$$\lambda_j = \mu_j, \quad w_j = X\hat{w}_j, \quad j = 1, \ldots, n$$

Proper orthogonal decomposition

What m-dimensional subspace S is the most close to $\{u^i\}_{i=1}^n$?

$$S = \mathrm{span}\{w_j\}_{j=1}^m \;\to\; \sum_{i=1}^{n} \|u^i - P_S u^i\|^2 = \sum_{j=m+1}^{N} \lambda_j$$

How to use POD?

- Given $u^1, ..., u^n$ and small integer m
- Form $X = \{u^1, ..., u^n\}$, $R = XX^T$ (or $r = X^T X$)
- Find m largest e.-v. of $Rw_j = \lambda_j w_j$, $(w_i, w_j) = \delta_{ij}$
- Form Reduced Model basis $V_m = \{w_1, ..., w_m\}$
- Form Reduced Model problem $V_m^T F(V_m \hat{u}) = 0$

How to use POD?

- Given $u^1, ..., u^n$ and small integer m
- Form $X = \{u^1, ..., u^n\}$, $R = XX^T$ (or $r = X^T X$)
- Find m largest e.-v. of $Rw_j = \lambda_j w_j$, $(w_i, w_j) = \delta_{ij}$
- Form Reduced Model basis $V_m = \{w_1, ..., w_m\}$
- Form Reduced Model problem $V_m^T F(V_m \hat{u}) = 0$

problem $V_m^T F(V_m \hat{u}) = 0$ has low dimension!

How to use Reduced Model?

- Solve Reduced Model problem $\hat{F}(\hat{u}) = V_m^T F(V_m \hat{u}) = 0$ by INB
- Project the solution to R^N: $u_0 = V_m \hat{u}$
- Use u_0 as initial guess for $F(u) = 0$

How to use Reduced Model?

- Solve Reduced Model problem $\hat{F}(\hat{u}) = V_m^T F(V_m \hat{u}) = 0$ by INB
- Project the solution to R^N: $u_0 = V_m \hat{u}$
- Use u_0 as initial guess for $F(u) = 0$

INB for $\hat{F}(\hat{u}) = 0$:

- Evaluation of $\hat{F}$ requires $V_m \cdot$, $F(\cdot)$, $V_m^T \cdot$
- No preconditioning (small m)

Fully implicit schemes accelerated by POD and RM

Choose n, $\epsilon > 0$. For $i = 1, ...$

If $i \leq n$ solve $F^i(u^i) = 0$ with accuracy ϵ and $u^i_0 = u^{i-1}$

Fully implicit schemes accelerated by POD and RM

Choose n, $\epsilon > 0$. For $i = 1, \ldots$

If $i \le n$ solve $F^i(u^i) = 0$ with accuracy ϵ and $u_0^i = u^{i-1}$

Else

1. if($mod(i,n) = 1$): form $X = \{u^j\}_{j=i-n}^{i-1}$, $R = XX^T$,

   form RM basis $V_m = \{w_j\}_{j=1}^{m}$: $\sum\limits_{k=m+1}^{N} \lambda_k < \epsilon$
2. solve $V_m^T F^i(V_m \hat{u}^i) = 0$ with accuracy $\epsilon/10$ ←Reduced Model
3. set $u_0^i = V_m \hat{u}^i$ ←Initial Guess
4. solve $F^i(u^i) = 0$ with accuracy ϵ ←Original Model

step 1 produces the reduced model basis (seldom)

step 2 solves implicitly the reduced model without preconditioning (m is small)

step 3 gives better initial guess for the original problem to be solved at step 4

adaptive choice of m may be replaced by $N\lambda_{m+1} < \epsilon$, or even fixed m

Example: unsteady lid driven cavity (2d NS)

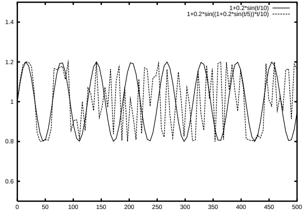
Streamfunction formulation

$$-\frac{\partial}{\partial t}(\Delta\psi) + \frac{1}{Re}\Delta^2\psi + (\psi_y(\Delta\psi)_x - \psi_x(\Delta\psi)_y) = 0$$

$$\psi = 0 \text{ on } \partial\Omega, \ \frac{\partial\psi}{\partial n} = \begin{cases} v(t) & \text{if } y = 1 \\ 0 & \text{if } 0 \le y < 1 \end{cases}$$

$\psi|_{t=0} = 0 \quad v(t) = 1 + 0.2\sin(t/10) \rightarrow$ quasi-periodic

$\psi|_{t=0} = 0 \quad v(t) = 1 + 0.2\sin([1 + 0.2\sin(t/5)]t/10) \rightarrow$ arrhythmic flow



GMRES preconditioned by $\frac{1}{Re}\Delta^2 \rightarrow$ independence of mesh size for $Re = 10^3$

Effect of POD-RM for quasi-periodic case

$h = 256^{-1} \to$ 65025 dof, $\|F^i(u_k^i)\| < 10^{-7}\|F^0(0)\|$
$\Delta t = 5 \to\sim 13$ time steps per period

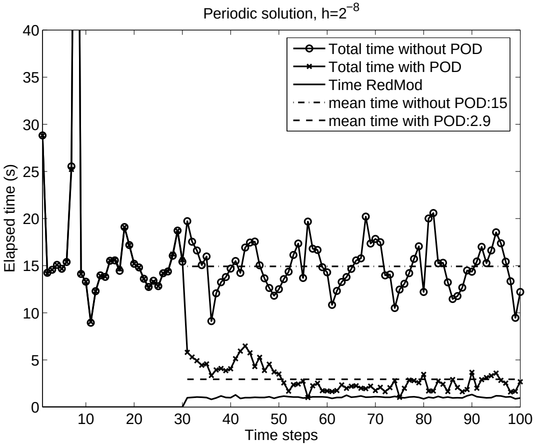
$n = 20 \to X = \{u^{20k-10} \dots u^{20k+9}\},\ k = 1, 2, \dots$
dimension of the reduced model $m = 10$.

$m = 10$ eigenvectors is more than enough: $\lambda_1 = 4.7 \cdot 10^3$, $\lambda_{10} = 5.6 \cdot 10^{-8}$

Arnoldi process with 50 multiplications by matrix $R \to$

$$\|Rw_1 - \lambda_1 w_1\|/\lambda_1 = 5 \cdot 10^{-16},\ \|Rw_{10} - \lambda_{10} w_{10}\|/\lambda_{10} = 8 \cdot 10^{-7}$$

Effect of POD-RM for quasi-periodic case

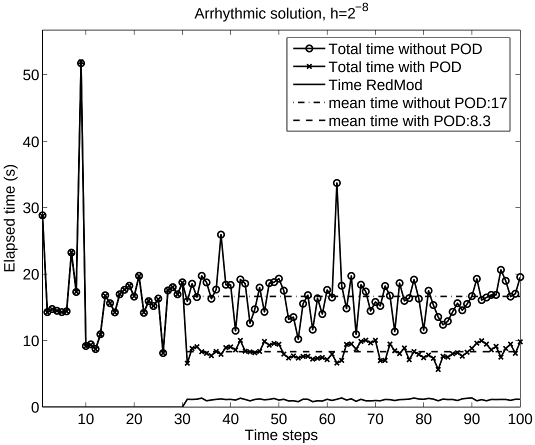


Effect of POD-RM for quasi-periodic case

i	10	20	30	32	52	72
	$u_0^i = u^{i-1}$			$u_0^i = V_m \hat{u}^i$		
$\Vert F(u_0^i)\Vert$	0.36	0.79	0.09	$22 \cdot 10^{-6}$	10^{-6}	$2.6 \cdot 10^{-6}$
n_{evF}	166	186	189	44+55	45+11	44+19
n_{precF}	160	180	183	0+51	0+ 9	0+16
CPU	13.4	15.3	16.1	1.2+4.2	1.1+1.1	1.1+1.2

average acceleration 5-6 times

Effect of POD-RM for arrhythmic case



Effect of POD-RM for arrhythmic case

i	10	20	30	32	52	72
	$u_0^i = u^{i-1}$			$u_0^i = V_m \hat{u}^i$		
$\lVert F(u_0^i)\rVert$	0.01	2.2	1.3	$5 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
n_{evF}	115	205	227	50+98	42+80	49+77
n_{precF}	110	198	221	0+93	0+75	0+72
CPU	9.2	16.7	19.5	1.3+7.7	1.0+6.4	1.1+5.9

average acceleration 2 times

GRID application

Architecture of application

Two types of remoted computing resources:

Solver $\longleftrightarrow$ **PODgenerator**

Features of GRID architecture and POD acceleration

- Slow communication network with high latency time between clusters of resources

Features of GRID architecture and POD acceleration

- Slow communication network with high latency time between clusters of resources
  - POD acceleration can be used whenever POD data are available

Features of GRID architecture and POD acceleration

- Slow communication network with high latency time between clusters of resources
  - POD acceleration can be used whenever POD data are available
  - no idling due to asynchronous non blocking communications

$$POD(generator) \leftrightarrow Solver$$

Features of GRID architecture and POD acceleration

- High probability of failure of some part of the computing resources

(if resources are not dedicated to single application and single user)

Features of GRID architecture and POD acceleration

- High probability of failure of some part of the computing resources

  (if resources are not dedicated to single application and single user)

  - POD generator can restart the computation on the solver resource

Features of GRID architecture and POD acceleration

- High probability of failure of some part of the computing resources

  (if resources are not dedicated to single application and single user)

  - POD generator can restart the computation on the solver resource

    It can spawn a new set of MPI processes on Solver resource

    It can provide last solution recovered from the reduced model (no backup)

Features of GRID architecture and POD acceleration

- POD resource may work with several tasks

Features of GRID architecture and POD acceleration

- POD resource may work with several tasks
  - Its task is waiting data from Solver and computing POD basis

Features of GRID architecture and POD acceleration

- POD resource may work with several tasks
  - Its task is waiting data from Solver and computing POD basis
  - Due to asynchronous non blocking communications it can
    - serve other Solvers
    - make post-processing (visualization, a posteriori error estimates)

Example: unsteady driven cavity (2d NS)

A Computer A = SGI Altix350

(2 Ithanium 1.3Ghz/3Mo, 1.3Gb/s network bandwidth)

B Computer B = Linux cluster

(AMD BiAthlon 1600+ MP, 100Mb/s network bandwidth)

N Network = latency 140 μs, maximum bandwidth $71 Mb/s$

Example: unsteady driven cavity (2d NS)

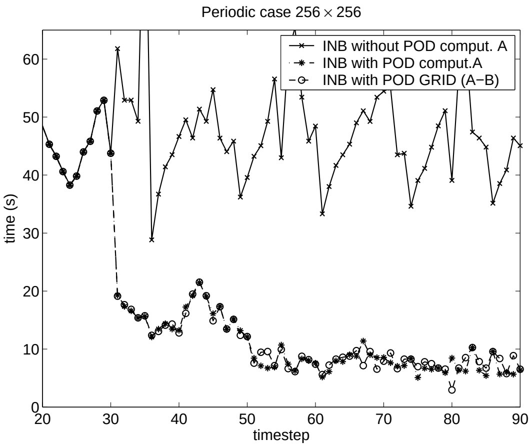
- A Computer A = SGI Altix350

  (2 Ithanium 1.3Ghz/3Mo, 1.3Gb/s network bandwidth)
- B Computer B = Linux cluster

  (AMD BiAthlon 1600+ MP, 100Mb/s network bandwidth)
- N Network = latency 140 μs, maximum bandwidth $71 Mb/s$

Solver resource = Computer A
POD resource = Computer B

Effect of POD-RM for quasi-periodic case



Conclusions

- Parallel computing technology should correlate to computational algorithm
- INB-POD-RM can accelerate implicit schemes in sequential and parallel computations
- INB-POD-RM can accelerate implicit schemes in GRID computing

Conclusions

- Parallel computing technology should correlate to computational algorithm
- INB-POD-RM can accelerate implicit schemes in sequential and parallel computations
- INB-POD-RM can accelerate implicit schemes in GRID computing

Conclusions

- Parallel computing technology should correlate to computational algorithm
- INB-POD-RM can accelerate implicit schemes in sequential and parallel computations
- INB-POD-RM can accelerate implicit schemes in GRID computing