Improving parallelism for the NEMO ocean model

Hind Shouli

Inhoud

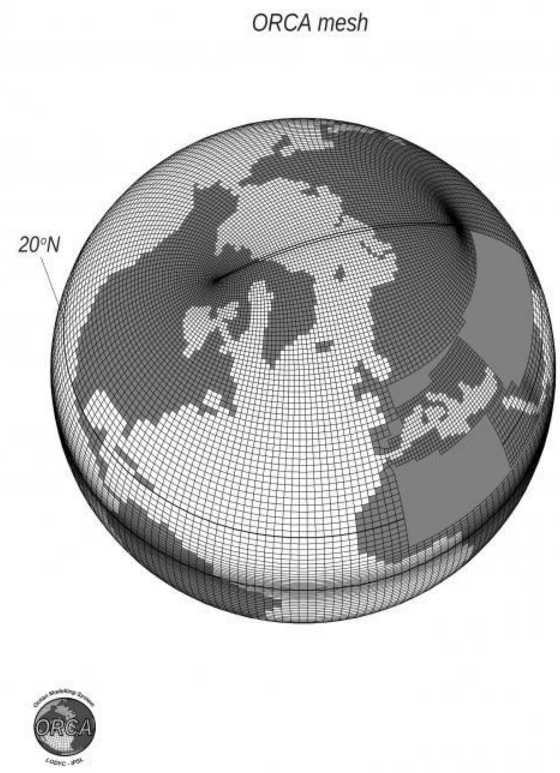
- Inleiding
- Probleem
- Numerieke methoden
- Testresultaten
- Conclusie

Inleiding

- SARA (Amsterdam) biedt onderzoekers in Nederland ondersteuning bij onder andere high performance computing.
- Nemo: Europees platvorm voor oceanografie
- Oceaanmodel van Nemo wordt ook in Nederland gebruikt.

Wat is het probleem?

- Nemo oceaanmodel
- $\frac{\partial}{\partial x}\left(\frac{1}{H}\frac{\partial \psi_t}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{1}{H}\frac{\partial \psi_t}{\partial y}\right)=\frac{\partial \bar{M}_2}{\partial x}-\frac{\partial \bar{M}_1}{\partial y}$
- Eindige differenties, gestructureerd grid
- Huidige oplosmethode: Conjugate Gradient
- Preconditionering: Diagonaalschaling



Doel en werkwijze

Doel:

- Sneller tot een oplossing komen.
- Algoritme geschikt voor parallel rekenen.

Werkwijze:

- In literatuur zoeken naar algoritmen voor snelle convergentie en parallel rekenen.
- De meest geschikte algoritmen testen op ‘testproblemen’.

Wat gaan we oplossen?

- Het lineaire stelsel: $Ax = b$
  - A discretizatiematrix
  - x onbekende
  - b rechterlid
- Matrix A is Symmetrisch, Positief Definiet (SPD) en sparse.
- Iteratieve Krylov methoden
  - Startoplossing u0.
  - Via iteratiestappen dichterbij de oplossing komen.
  - Stopcriterium.

Conjugate Gradient solver

- Meest toegepaste methode voor sparse SPD lineaire stelsels.
- Gebaseerd op een orthogonale projectie techniek op de Krylov deelruimte.
- Deze methode convergeert, maar de convergentie kan versneld worden met behulp van Preconditionering

Preconditionering

Waarom?

- Convergentie versnellen

Hoe?

- Het originele stelsel transformeren naar een stelsel die dezelfde oplossing heeft maar makkelijker op te lossen is.

- Kies preconditioner M zodanig dat:
    - M op A lijkt.
    - Het stelsel $Mx = b$ eenvoudig op te lossen is.

Preconditionering

- Transformeer het originele stelsel naar:

$$M-1Ax = M-1b$$

- Om de symmetrie van *A* te behouden, wordt de Cholesky ontbinding M = LLt gebruikt voor de transformatie. De transformatie is dan equivalent aan:

$$L-1AL-T \text{ etc.}$$

- Huidige preconditioner: $M = \text{diag}(A)$.

Incomplete Cholesky CG

- Cholesky ontbinding van *A* zonder fill-in.
- M = LdL

- Deze preconditioner geeft goede resultaten, maar is niet parallelliseerbaar.

Parallel rekenen

Parallel rekenen kan op twee manieren mogelijk worden gemaakt:

- Van bestaande algoritmen waar mogelijk parallelle versie maken.
- Nieuwe algoritmen ontwikkelen speciaal voor parallel rekenen

Domein decompositie

- Leunt op het principe van ‘verdeel-en heers’.
- Voordelen:
  - Makkelijkere subproblemen door eenvoudige geometrische vormen
  - Het oorspronkelijke fysieke probleem kan natuurlijk onderverdeeld worden
  - Maakt parallel rekenen mogelijk

Parallelle preconditioners

- Parallelle preconditioners hebben vaak slechte convergentie-eigenschappen.
- Doel is het vinden van een parallelle preconditioner die snelle convergentie levert.

Incomplete Poisson Preconditioner

- Speciaal voor de Poisson vergelijking.
- Ook hier geldt dat 'Incomplete' duidt op het weglaten van fill-in elementen.
- Deze preconditioner kan rij voor rij opgebouwd worden.
- Degree of parallellism = N.
- IP kan toegepast worden in het algoritme van PCG

Deflatie

- Reduceert het aantal iteraties van de PCG algoritme door de eigenwaarden van de preconditioned matrix aan te pakken.
- Bij deze algoritme kiest met een aantal ‘deflatievectoren’. Dit komt overeen met het verdelen van het domein in subdomeinen.
- Ook de Deflatie PCG is parallelliseerbaar.

Testproblemen

Op de eenheidsvierkant:

- 1: Poisson vergelijking met homogene Dirichlet randvoorwaarden
- 2: Homogene Nemo vergelijking met H = constant
- 3: Homogenen Nemo vergelijking met variabele H:
  - H = 100 op de rand en H = 5000 in het midden
  - H neemt in stapjes af

Numerieke Algoritmen

Op de testproblemen hebben we toegepast:

- Conjugate Gradient
- Incomplete Cholesky CG
- Incomplete Poisson (PCG)
- Deflated PCG, $M = D$
- Deflated PCG, $M = IP$
  - Domein opgedeeld in <u>strookjes</u> en <u>vierkantjes</u>

Resultaten Testprobleem 1 Poisson vergelijking

- Vergelijking: $\begin{cases} -\Delta u = -\dfrac{\partial^2 u}{\partial x^2} - \dfrac{\partial^2 u}{\partial y^2} = 1 \text{ on } (0,1) \times (0,1), \\ u = 0 \quad \text{on the boudary} \end{cases}$

Dimension	CG	ICCG(0)	IP	DPCG(D)	DPCG(IPP)
9x9	5	4	5	6	
100x100	25	9	14	10	7
900x900	64	20	34	11	8

- DPCG geeft de beste resultaten als het domein in vierkantjes is verdeeld. 100x100: 25 en 900x900: 225 vierkantjes.

Resultaten Testprobleem 2 Homogene Nemo vergelijking

- Vergelijking: $$\begin{cases} \frac{\partial}{\partial x}\left(\frac{1}{H}\frac{\partial \psi_t}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{H}\frac{\partial \psi_t}{\partial y}\right) = 0 & \text{on } (0,1) \times (0,1) \\ \frac{\partial \psi}{\partial t} = 0 & \text{on the boundary} \end{cases}$$
- H = 5250, constant.

Dimension	CG	ICCG(0)	IP	DPCG(D)	DPCG(IPP)
9x9	5	3	4		
100x100	19	7	10	7	5
900x900	57	18	31	11	8

Resultaten Testprobleem 3 Homogene Nemo vergelijking

- Geval 1: H = 100 op de buitenrand met breedte 0.3 In het midden H = 5000.

Dimension	CG	ICCG(0)	IP	DPCG(D)	DPCG(IPP)
9x9	7	5	7		
100x100	56	17	33	12	23
900x900	201	55	108	18	43

- DPCG(D) geeft hier beter resultaat dan DPCG(IPP).

- DPCG(IPP) is ondanks snellere convergentie niet altijd de meest geschikte methode omdat deze duurder is als DPCG(D).

Resultaten Testprobleem 3 Homogene Nemo vergelijking

- Geval 2: H varieert in stappen. Buiten strip van breedte 0.1 H=150, tweede strip H= 500, derde strip H= 2000, H = 3500 en in het midden H = 5000.

Dimension	CG	ICCG(0)	IP	DPCG(D)	DPCG(IPP)
9x9	6	3	6		
100x100	46	11	33	10	25
900x900	180	26	100	15	44

- Verschil met geval 1 niet groot.
- Wederom: DPCG(D) geeft verreweg het beste resultaat.

Een meer realistisch probleem

- Eenheidsvierkant verdeeld in 60x60 onbekenden.
- dim(A) = 3600 x 3600.
- H = variabel, geval 2.

Dimension	CG	ICCG(0)	IP	DPCG(D)	DPCG(IPP)
3600x3600	370	49	201	15	53

Conclusie

- Deflatie vermindert het aantal iteraties aanzienlijk.
- De beste resultaten als het domein in kleine vierkantjes wordt verdeeld.
- Diag(A) als preconditioner geeft voor variabele H het beste resultaat.

Vragen?


Bedankt!
© Disney/Pixar