Massively Parallel DUNE Implementation of Tensor-Product Multigrid Solvers in Geophysical Modelling

Andreas Dedner[†], **Eike Hermann Müller**[⋆], Robert Scheichl[⋆]

(thanks to all DUNE developers, esp. Oliver Sander for help with parallel UG)

[†]University of Warwick / [⋆]University of Bath

DUNE User Meeting, Aachen 25th/26th Sep 2013

UNIVERSITY OF BATH

PDEs in “flat” geometries in geophysical modelling

Geophysical modelling in “flat” geometries

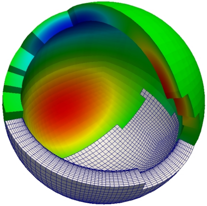
$$R_{\text{earth}} \sim 10^4 km \sim \textbf{Width} \gg \textbf{Height} \sim 10^2 km (\sim H_{\text{atmos}}, H_{\text{ocean}})$$

Numerical Weather- and Climate- Prediction (NWP)

Solve **elliptic PDE** for global pressure correction with $\gtrsim 10^{10}$ dof at every model time step (and solve it as fast as possible!)

PDEs with similar structure arise in

- Global ocean modelling [Marshall et al (1997), Gerritsen (2006)]
- Subsurface flow simulations
- Oil- and gas reservoir modelling [Lacroix et al (2003)] (supercoarsening multigrid)



⇒ algorithmically optimal & massively parallel solvers

1. Tensor product multigrid (TPMG)
   - Elliptic operator and grid structure
   - TPMG idea
   - DUNE implementation
2. Numerical results
   - Case Study I: Idealized flow
   - Case Study II: Aquaplanet
   - Massively parallel scaling on HECToR
3. Conclusion and Outlook

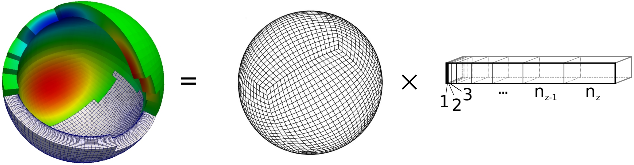
Model equation and grid structure

Elliptic operator for pressure correction π' (see [Wood et al. (2013)])

$$-\omega^2\left\{\alpha^{(r)}(\partial_r\pi') + \frac{\partial_r\left(\alpha^{(rr)}(\partial_r\pi')\right) + \nabla_{2d}\cdot\left(\alpha^{(hh)}(\nabla_{2d}\pi')\right)}{r^2}\right\} + \beta\pi'$$

Atmospheric state $\{\boldsymbol{u},\theta,\pi,\rho\} \Rightarrow$ profiles α, β

Tensor product grid

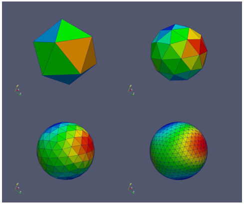


- Semi-structured horizontal grid: cubed sphere, icosahedral,...
- Regular (graded) 1d vertical grid, $n_z = \mathcal{O}(100)$
- $R_{\text{earth}}/H \approx 100 \Rightarrow$ **Grid-aligned anisotropy** $\left(\frac{\Delta x}{\Delta z}\right)^2 \gg 1$

Tensor product multigrid

Grid-aligned anisotropy ⇒ **Tensor product multigrid (TPMG)**

- Geometric multigrid
- Horizontal coarsening only
- Vertical line relaxation (tridiagonal solve)



Optimal for **factorising profiles**

[Börm & Hiptmair Num. Alg., 26:200-1 (1999)]

$$\alpha(r,\hat{\mathbf{r}}) = \alpha_r(r)\cdot\alpha_h(\hat{\mathbf{r}})$$

Atmospheric profiles **factorise approximately**

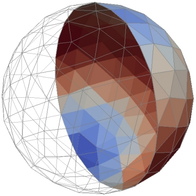
$$\alpha(r,\hat{\mathbf{r}}) \approx \alpha_r(r)\cdot\alpha_h(\hat{\mathbf{r}}) \equiv \alpha^{\otimes}(r,\hat{\mathbf{r}})$$

1. Still very efficient as **standalone solver**
   (demonstrated for full operational equation in Met Office Unified Model)
2. Use factorising $\alpha^{\otimes}(r,\hat{\mathbf{r}})$ in **preconditioner**
   Faster matrix construction as $\mathcal{O}(n_{\text{horiz}} + n_z) \ll \mathcal{O}(n_{\text{horiz}}\cdot n_z)$ storage

DUNE Implementation

Matrix-“free” DUNE implementation

- Bespoke geometric multigrid and iterative solvers (DUNE-grid)
- Parallel 2d host grid $\subset \mathbb{R}^3$: ALUGrid$\langle 2,3\rangle$, UGGrid$\langle 3,3\rangle$



Data structures:

- *3d Fields:* Store vector of length n_z on each horizontal grid cell
- *Matrix:* Construct on-the-fly from profiles $\alpha^{(hh)}, \alpha^{(rr)}, \alpha^{(r)}, \beta$
  1. Non-factorising case:
     - 3d fields of size $\mathcal{O}(n_{\text{horiz}} \cdot n_z)$
  2. Factorising case:
     - Horizontal components = 2d fields of size $\mathcal{O}(n_{\text{horiz}})$
     - Vertical component = 1d field of size $\mathcal{O}(n_z)$

Indirect addressing in horizontal only

⇒ “Hidden” by work in vertical as $n_z \gg 1$ [MacDonald et al. (2011)]

Case Study I: Idealized flow

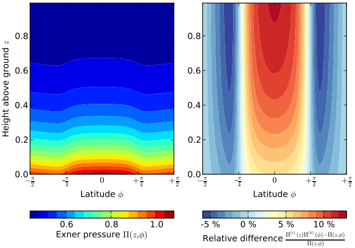
Idealised balanced atmospheric flow field

Only non-factorising ingredient to α, β: Exner pressure π

$$\pi(r,\hat{\mathbf{r}}) = \frac{\epsilon + E_r(r)E_h(\hat{\mathbf{r}})}{1+\epsilon}$$

$$\pi^{\otimes}(r,\hat{\mathbf{r}}) \equiv \frac{\epsilon + E_r(r)}{1+\epsilon}\cdot E_h(\hat{\mathbf{r}})$$

$$\alpha = \alpha^{\otimes}, \beta = \beta^{\otimes} \text{ for } \epsilon = 0$$



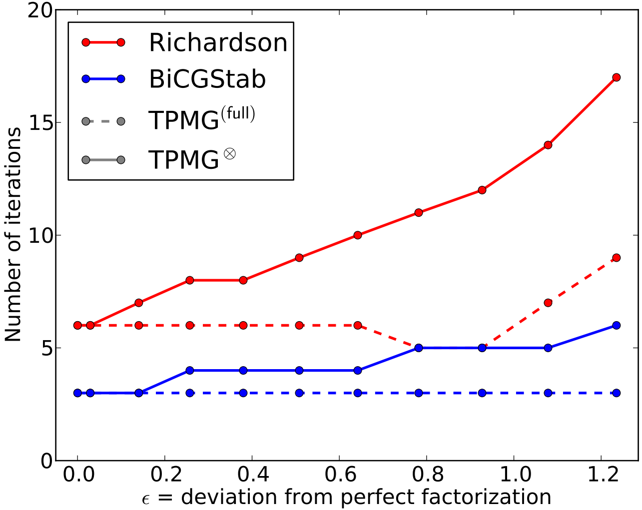
Preconditioners

$\mathrm{SOR}_{\rho=1}(2,2)$ V-cycle of TPMG

1. Full non-factorizing profiles ($\mathrm{TPMG}^{(\mathrm{full})}$)
2. Approximately factorized profiles ($\mathrm{TPMG}^{\otimes}$)

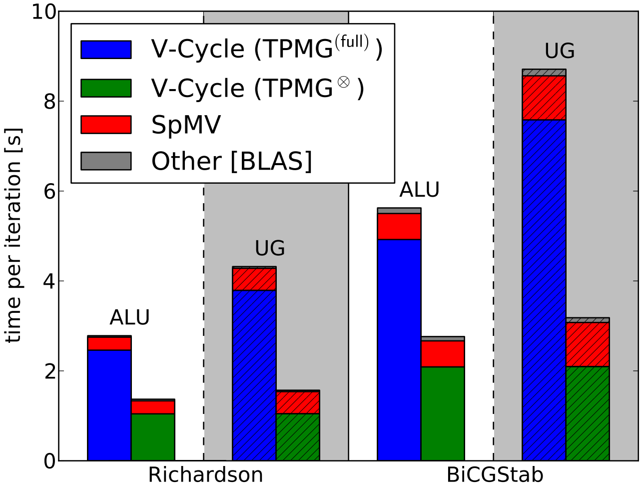
Results: Number of iterations

Number of iterations (relative residual reduction $= 10^{-5}$)



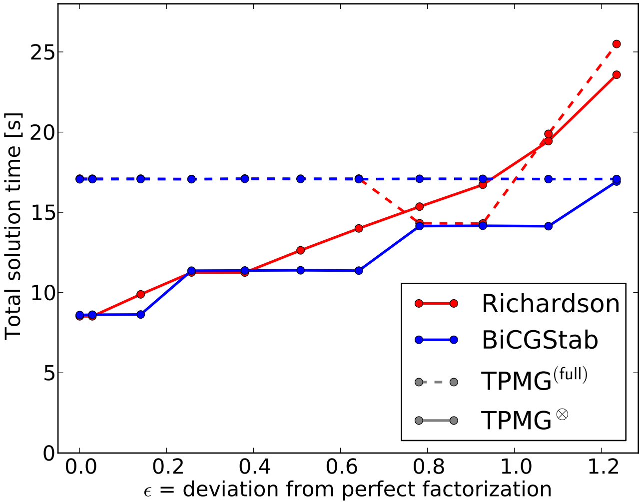
Results: Time per iteration

Time per iteration (sequential) for ALUGrid and UGGrid

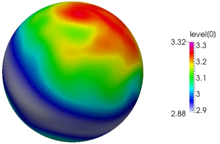


Results: Total solution time

Total sequential solution time for $2.6 \cdot 10^6$ dof, $n_z = 128$, ALUGrid



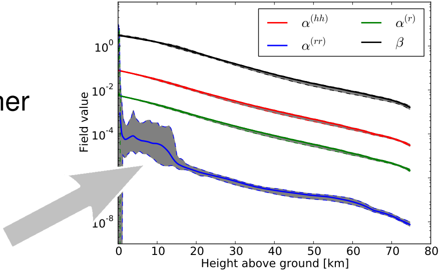
Case Study II: Aquaplanet



Realistic atmospheric testcase
Profiles from "aquaplanet" run with full Unified Model

Preconditioners

- TPMG$^{(\text{full})}$: still excellent preconditioner
- TPMG$^{\otimes}$: Near-surface convection
  - $\alpha^{(rr)}$ hard to factorize
  - Solver diverges



- TPMG$^{[\otimes]}$:
  - factorize $\alpha^{(hh)}$, $\alpha^{(r)}$ and β only
  - Performance loss of $\approx$ 15% in t_{iter} relative to TPMG$^{\otimes}$.

Results

Sequential performance, $2.6 \cdot 10^6$ dof, ALUGrid

Time per iteration [s]

Solver	$\text{TPMG}^{(\text{full})}$	$\text{TPMG}^{[\otimes]}$	$\text{TPMG}^{\otimes}$
Richardson	2.85	1.58	1.37
BiCGStab	5.70	3.19	2.76

Number of iterations and total solution time [s]

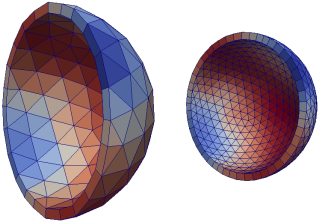
	# iterations		total time	
Solver	$\text{TPMG}^{(\text{full})}$	$\text{TPMG}^{[\otimes]}$	$\text{TPMG}^{(\text{full})}$	$\text{TPMG}^{[\otimes]}$
Richardson	5	7	14.58	11.37
BiCGStab	3	3	17.43	**9.91**

Parallel grids

Parallel grid implementations

Requirements: parallel 2d grid $\subset \mathbb{R}^3$ for unit sphere

- ALUGrid$\langle 2, 3\rangle$
  - Does not scale!
- UGGrid$\langle 3, 3\rangle$
  - 2d grid = thin shell
  - Horizontal refinement only
- Other ideas?



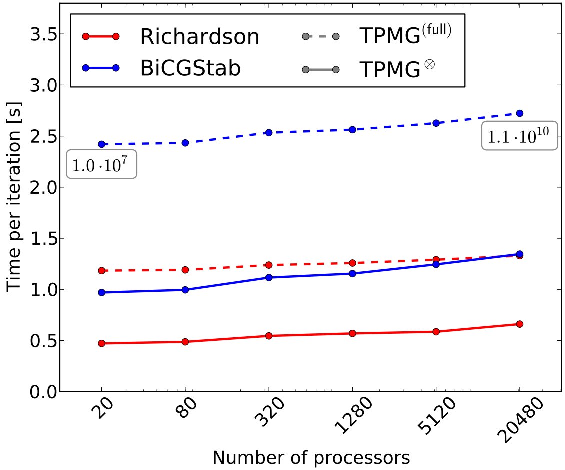
UGGrid originally not designed for more than $\sim$ 100 processors
$\Rightarrow$ Fix bugs & plug several memory leaks [Oliver Sander]

Parallel partitioning

- Read macrogrid with n_{proc} elements from `.dgf`
- Loadbalance $\Rightarrow$ one element / processor
- Refine (in horizontal only)

Massively parallel weak scaling

Weak scaling on HECToR: Case study I (balanced flow), UGGrid



Summary and outlook

Summary

- Geophysical modelling: **Elliptic PDEs in “flat” domains**
- **Tensor product grids**: 2d (semi-stuctured) $\otimes$ 1d
- **Tensor-product multigrid** solver [Börm and Hiptmair (1999)]
- **Matrix-free DUNE implementation** based on ALUGrid$\langle 2,3\rangle$ and UGGrid$\langle 3,3\rangle$
- **Approximate factorisation** in preconditioner
- Massively **parallel implementation** with UGGrid$\langle 3,3\rangle$

Outlook

- Extend proof in to non-factorising case
- Compare to DUNE-ISTL AMG
- Replace iteration over 2d grid by connectivity lookup tables
- More testing: impact of orography?
- Improve performance, parallel scaling $\Rightarrow$ UG, ALUGrid?
- (Mimetic) FEM [Cotter and Thuburn (2013)]: $P_0 \mapsto$ higher order DG