Recent developments in the theory and application of the sparse grid combination technique

Markus Hegland, Brendan Harding and Peter Strazdins, ANU
and Christoph Kowitz, TUM and Dirk Pflueger, Uni Stuttgart

SPPEXA workshop, 25. January 2016

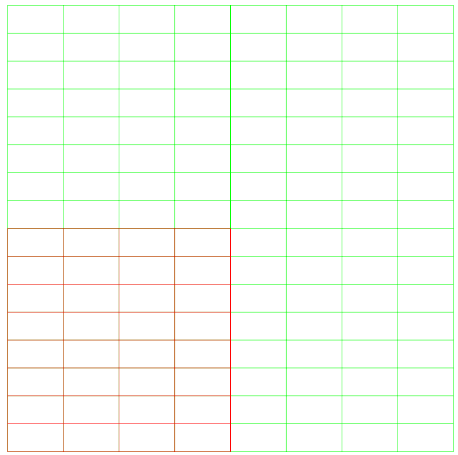
Section 1

Overview

Basic extrapolation approach

- use a highly scalable PDE solver
- run for a selection of numerical grids
- combine results to get higher accuracy and fault tolerance

spatial discretisation



- $n_1 \times n_2$ grid with $n_i = 2^{\gamma_i} + 1$, e.g. $\gamma = (3, 4)$
- parallelise over cells of size $m_1 \times m_2$, e.g. $m_{1,2} = 5, 9$
- numerical solution $u(\gamma)(x, t)$ obtained by interpolation

The PDE: Vlasov–Maxwell Equations

- Vlasov-Equation

$$\frac{\partial f_s}{\partial t} + \vec{v} \cdot \frac{\partial f_s}{\partial \vec{x}} + \frac{q_s}{m_s}(\vec{E} + \vec{v} \times \vec{B}) \cdot \frac{\partial f_s}{\partial \vec{v}} = 0$$

- Moments of the Distribution Function f

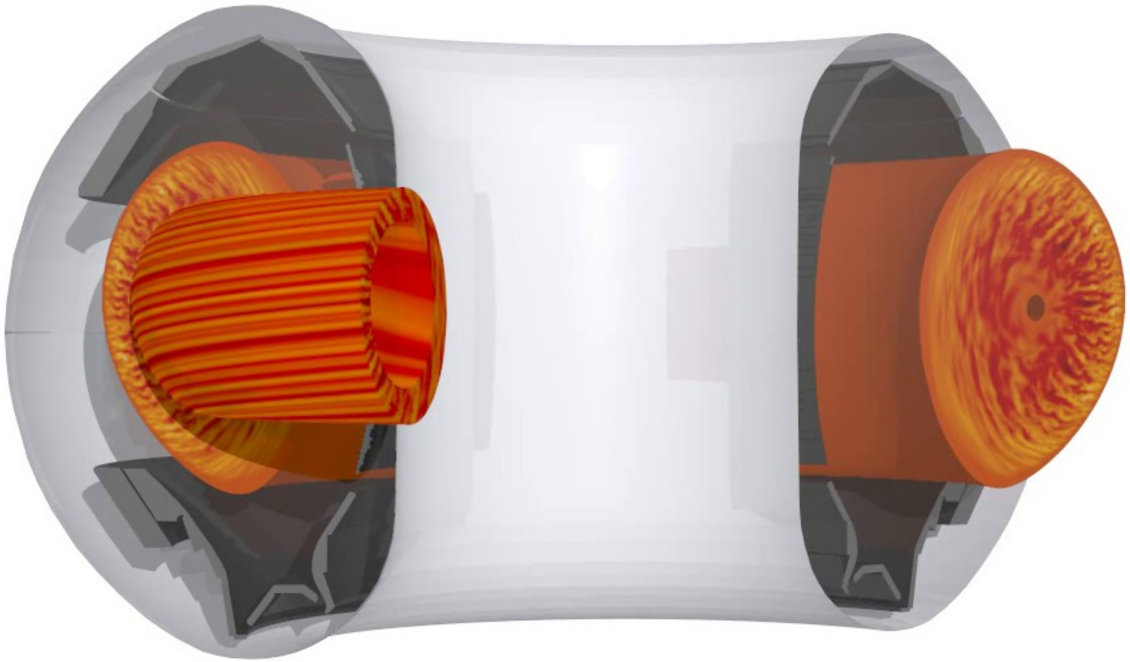
$$\rho(\vec{x}, t) = \sum_s q_s \int f_s(\vec{x}, \vec{v}, t)\, dv \quad \vec{j}(\vec{x}, t) = \sum_s q_s \int f_s(\vec{x}, \vec{v}, t)\vec{v} dv$$

- Maxwell Equations

$$-\frac{1}{c^2}\frac{\partial \vec{E}}{\partial t} + \nabla \times \vec{B} = \mu_0(\vec{j}_0 + \vec{j}) \qquad \nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0}$$

$$\frac{\partial \vec{B}}{\partial t} + \nabla \times \vec{E} = 0 \qquad \nabla \cdot \vec{B} = 0$$

GENE – Gyrokinetic Electromagnetic Numerical Experiment (genecode.org)



- open source plasma research code for investigation of microturbulence
- gyrokinetic approximation of the Vlasov equations
- nonlinear parametrisation of state space with 5 dimensions
- uses MPI and has been shown to be strongly scalable

Section 2

Sparse grid combination technique

Hierarchical decomposition

- let V_γ be the approximation space corresponding to $u(\gamma)$
- we assume $V_\alpha \subset V_\beta$ if $\alpha \leq \beta$
- hierarchical decomposition of $u(\gamma) \in V_\gamma$ into $w(\alpha) \in V_\alpha$:

  $$u(\gamma) = \sum_{\alpha \leq \gamma} w(\alpha)$$

  such $w(\alpha)$ exist and are unique, there is a formula ...
- if $u(\gamma) = I_\gamma u$ are the interpolants in V_γ on regular grids, the $w(\alpha)$ are the *hierarchical surplus* of u in V_α

Motivation for the decomposition

- in some cases

$$\|w(\alpha)\| \leq K4^{-|\alpha|}$$

  where $|\alpha| = \alpha_1 + \cdots + \alpha_d$
- the values of $4^{-|\alpha|}$ for $\alpha \leq (5,5)$ are

10^{-3}	$2.5 \cdot 10^{-4}$	$6 \cdot 10^{-5}$	$1.5 \cdot 10^{-5}$	$4 \cdot 10^{-6}$	10^{-6}
$4 \cdot 10^{-3}$	10^{-3}	$2.5 \cdot 10^{-4}$	$6 \cdot 10^{-5}$	$1.5 \cdot 10^{-5}$	$4 \cdot 10^{-6}$
0.016	$4 \cdot 10^{-3}$	10^{-3}	$2.5 \cdot 10^{-4}$	$6 \cdot 10^{-5}$	$1.5 \cdot 10^{-5}$
0.0625	0.016	$4 \cdot 10^{-3}$	10^{-3}	$2.5 \cdot 10^{-4}$	$6 \cdot 10^{-5}$
0.25	0.0625	0.016	$4 \cdot 10^{-3}$	10^{-3}	$2.5 \cdot 10^{-4}$
1	0.25	0.0625	0.016	$4 \cdot 10^{-3}$	10^{-3}

- typical complexity of computing $w(\alpha)$ is $O(2^{|\alpha|})$ – the smallest components are the most expensive:
  - $w(5,5)$ costs 1024 times more than $w(0,0)$
  - neglecting $w(5,5)$ produces error in sixth digit

Motivation for the decomposition

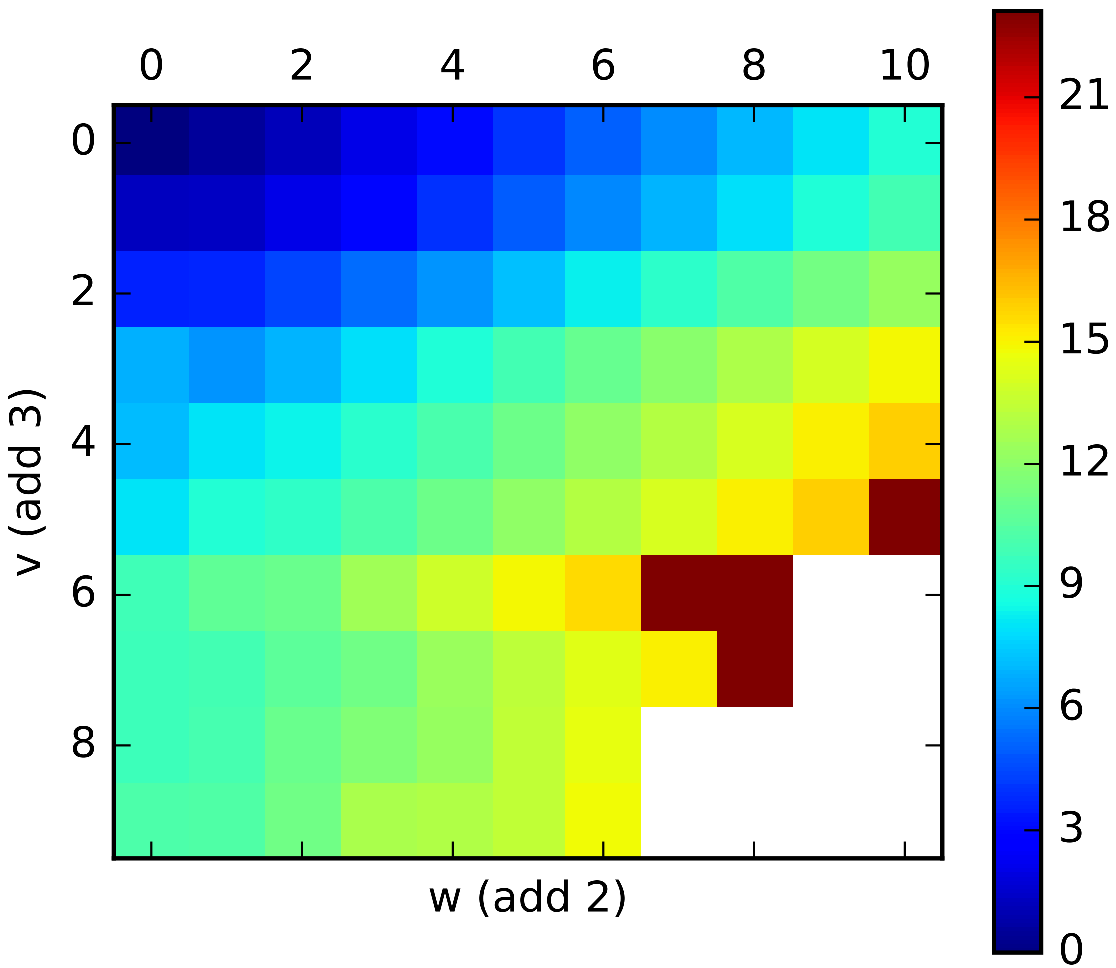


Figure 1:surplus of QoI

Combination formula

- sparse grid combination approximation of $u(\gamma)$ for $\gamma = (n, n, \dots, n)$

$$u_n^C = \sum_{|\alpha| \leq n} w(\alpha)$$

- combination formula[1]

$$u_n^C = \sum_{|\gamma| \leq n} c_\gamma \, u(\gamma)$$

- in the case of $n = 2$ and $d = 2$ this is

$$u_2^C = u(0,2) + u(1,1) + u(2,0) - u(0,1) - u(1,0)$$

[1] $|\alpha| = \alpha_1 + \cdots + \alpha_d$

Sketch of parallel performance

1. Compute all component solutions $u(\gamma)$
   - computation of every component is distributed over subcluster[2]
   - multiple $u(\gamma)$ are computed concurrently
   - lower bound for wall clock time $O(2^n/p)$
2. Compute the combination $\sum_\gamma c_\gamma \, u(\gamma)$
   - distributed addition of one component to the sum
   - parallel summation
   - lower bound for wall clock time $O((d-1)\log_2(n)\, 2^n/p)$

For PDE solvers, time for first step usually dominates

[2] $p=$ size of subcluster

Section 3

nonstandard sparse grids

When some components $u(\gamma)$ are missing

- large errors (predicted or detected) in $u(\gamma)$ because of limitations of physical model, numerical approximation or faults
  - error detection may use the $w(\alpha)$
- components $u(\gamma)$ have not been computed due to hardware issues
- some components $w(\alpha)$ are predicted to be very small and can be neglected

approach

- determine downset containing available grids
- compute some of the missing ones
- use combination coefficients for the particular downset

when a top level component is missing

- case of $n = 5$ and $d = 2$ (target grid: blue)

$(0,5)$					$(5,5)$
$(0,4)$	$(1,4)$				
	$(1,3)$	$(2,3)$			
	$(1,2)$	$(2,2)$	$(3,2)$		
			$(3,1)$	$(4,1)$	
				$(4,0)$	$(5,0)$

- the error increases by $w(2,3)$ if $u(2,3)$ is missing (yellow)
- we need to compute $u(1,2)$ in order to compute this approximation (red)
- the values $u(1,3)$ and $u(2,2)$ are now not needed any more (green)
- solution: compute all components 2 levels down from the top (this is cheap as the corresponding grids are small)

when a component at a lower level is missing

- case of $n = 5$ and $d = 2$ (target grid: blue)

$(0,5)$					$(5,5)$
$(0,4)$	$(1,4)$				
	$(1,3)$	$(2,3)$			
	$(1,2)$	$(2,2)$	$(3,2)$		
			$(3,1)$	$(4,1)$	
				$(4,0)$	$(5,0)$

- when $u(2,2)$ is missing (yellow) we again could compute $u(1,2)$ (red)
- in the combination one then does not use $u(1,3)$ and $u(2,3)$ (green) and the error increases by $w(2,3)$
- two possible solutions:
  - precompute the components two levels down from the top
  - duplicate computations of all components one level from the top

Families of sparse grids for the combination technique

$$u_I = \sum_{\gamma \in I} c_\gamma \, u(\gamma)$$

- classical SG: $I = \{\alpha \mid |\alpha| \leq n + d - 1\}$
- truncated SG[3]: $I = \downarrow \{\alpha \mid |\alpha| \leq n + d - 1, \ \alpha \geq \beta\}$
- partial SG if $\beta \geq 1$:
  $$I = \downarrow \{\alpha \mid |\alpha| \leq n + |\beta| - 1, \ \alpha \geq \beta\}$$
- SG if $u(\beta)$ removed and $|\beta| = n + d - 1$:
  $$I = \{\alpha \mid |\alpha| \leq n + d - 1, \ \alpha \neq \beta\}$$
- 2-scale SG[4]:
  $$I = \bigcup_{k=1}^{d} \{\alpha \mid \alpha \leq n_0 1 + n_k e_k\}$$
- ANOVA:
  $$I = \{\alpha \mid |\operatorname{supp} \alpha| \leq k\}$$

---

[3] $\downarrow I$ is smallest downset containing I

[4] special case: $n_0 = n_k = n$

Section 4

nonstandard coefficients

Opticom method – best possible combination coefficients

- convex optimisation problem

$$u = \operatorname{argmin}\{J(v) \mid v \in V\}$$

- Ritz method

$$u(\gamma) = \operatorname{argmin}\{J(v) \mid v \in V(\gamma)\}$$

- Opticom method utilises Ritz approach

$$u^O = \operatorname{argmin}_v\{J(v) \mid v = \sum_{\gamma \in I} c_\gamma u(\gamma),\ c_\gamma \in \mathbb{R}\}$$

- Ritz is quasi optimal for appropriate norm

$$\|u^O - u\| \leq C \left\| \sum_\gamma c_\gamma u(\gamma) - u \right\| \quad \text{for all } c_\gamma \in \mathbb{R}$$

Comparing a general combination with the standard one for classical sparse grid

$$
\begin{aligned}
u_n^C - u_n^{SG} &= \sum_{|\gamma|\le n} c_\gamma u(\gamma) - \sum_{|\alpha|\le n} w(\alpha) \\
&= \sum_{|\gamma|\le n} c_\gamma \sum_{\alpha\le\gamma} w(\alpha) - \sum_{|\alpha|\le n} w(\alpha) \\
&= \sum_{|\alpha|\le n} \left(\sum_{\gamma\in I(\alpha,n)} c_\gamma - 1 \right) w(\alpha)
\end{aligned}
$$

where $I(\alpha, n) = \{\gamma \mid |\gamma| \le n, \alpha \le \gamma\}$

- bracket is zero for standard sparse grid solution
- faults require to set certain c_γ to zero

An optimal apriori choice of the (remaining) c_γ

- for the case where $u(\beta)$ is not available or not acceptable one sets $c_\beta = 0$ and the other components are obtained minimising $J(\mathbf{c})$:

$$\mathbf{c}^{\text{best}} = \operatorname{argmin}\{J(\mathbf{c}) \mid c_\beta = 0\}$$

  where $\mathbf{c}$ is the vector with components c_γ for $|\gamma| \le n$ and

$$J(\mathbf{c}) = \sum_{|\alpha| \le n} 4^{-|\alpha|} \left| \sum_{\gamma \in I(\alpha, n)} c_\gamma - 1 \right|$$

  this is a piecewise linear optimisation problem with constraints
- the form of the objective function is motivated by the bounds

$$\|w(\alpha)\| \le 4^{-|\alpha|} K$$

- approximation error is then bounded by

$$\|u_n^C - u_n^{\text{SG}}\| \le K\, J(\mathbf{c})$$

Section 5

Application

Eigenvalue problems

- eigenvalue problem $Lu = \lambda u$ with constraint $\langle s, u \rangle = 1$
- let $u(\gamma)$ and $\lambda(\gamma)$ be approximations satisfying the constraint
- introduce the operator $G : \mathbb{R}^m \to V$ defined by

$$Gc = \sum_{\gamma \in I} c_\gamma \, u(\gamma)$$

- consider the quadratic optimisation problem

$$(c^C, \lambda^C) = \operatorname{argmin}_{(c,\lambda)} \|LGc - \lambda Gc\|$$

with constraint[5] $\sum_{\gamma \in I} c_\gamma = 1$

[5] follows from the constraint $\langle s, Gc \rangle = 1$

A solution from 1964

- Osborne considered the problem

$$\begin{bmatrix} K(\lambda) & t \\ s^* & 0 \end{bmatrix} \begin{bmatrix} c \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- this gives

$$\beta(\lambda) = -\frac{1}{\langle s^*, K(\lambda)^{-1} t \rangle}$$

- the problem $\beta(\lambda) = 1$ is then solved with Newton's method
- application to the combination technique with

$$K(\lambda) = (LG - \lambda G)^* (LG - \lambda G)$$

- thus we get the method:
  1. solve the component problems to get $u(\gamma)$
  2. reduction operation to compute the matrix $K(\lambda)$
  3. solve the optimisation problem to get the coefficients c_γ
  4. evaluate the combination $\sum_{\gamma \in I} c_\gamma u(\gamma)$

Summary

- consider general combination formulas of the form

$$u^C = \sum_{\gamma \in I} c_\gamma u(\gamma)$$

- theory based on hierarchical decompositions

$$u(\gamma) = \sum_{\alpha \leq \gamma} w(\alpha)$$

- lead to new algorithms
  - with extra degree of parallelism
  - avoids curse of dimension
  - provides a new level of fault tolerance
  - maintains scalability
  - reuses the original code to compute $u(\gamma)$