A review of the sparse grid combination technique for the solution of partial differential equations

Markus Hegland

Monash Workshop on Numerical PDEs, 15-19 February 2016

1. Introduction

2. Sparse grid combination technique

3. New sparse grids

4. Optimisation problems

Section 1

Introduction

Preview

PDEs

- simple examples:

$u_t = c^T \nabla u_x$	advection equation
$u_t = \Delta u$	heat equation
$\Delta u = f$	Poisson equation

- other examples include Navier-Stokes equations, Maxwell equations, Boltzmann equations, Kolmogorov equations, Schrodinger equations and many others
- they may contain integrals, nonlinearities and discontinuities
- we consider initial value problems, boundary value problems and eigenvalue problems
- a challenge is the curse of dimensionality

Numerical solvers

- we assume that we are **given** a numerical method which computes a k-sequence $\{u(\gamma) \mid \gamma \in \mathbb{N}^k\}$ which converges to the exact solution
- the index γ controls the accuracy, for example, for finite differences $2^{-\gamma_i} = h_i$ may denote the grid size for the discretisation in the i-th spatial dimension
- we will call γ the *numerical parameter*

Required properties of the numerical solvers

- the approximation $u(\gamma) \in V_\gamma$ where V_γ is a finite dimensional approximation space, e.g., a linear space of piecewise polynomials, wavelets, trigonometric functions etc
- the best approximations of the exact solution in V_γ converge to the exact solution
- in many cases the numerical method is quasi-optimal, i.e.,

$$\|u(\gamma) - u\| \leq c \min_{v \in V_\gamma} \|v - u\|$$

  for some constant c
- the spaces V_γ form a lattice and

$$V_\beta \subset V_\gamma \quad \text{iff} \quad \beta \leq \gamma$$

The sparse grid combination technique

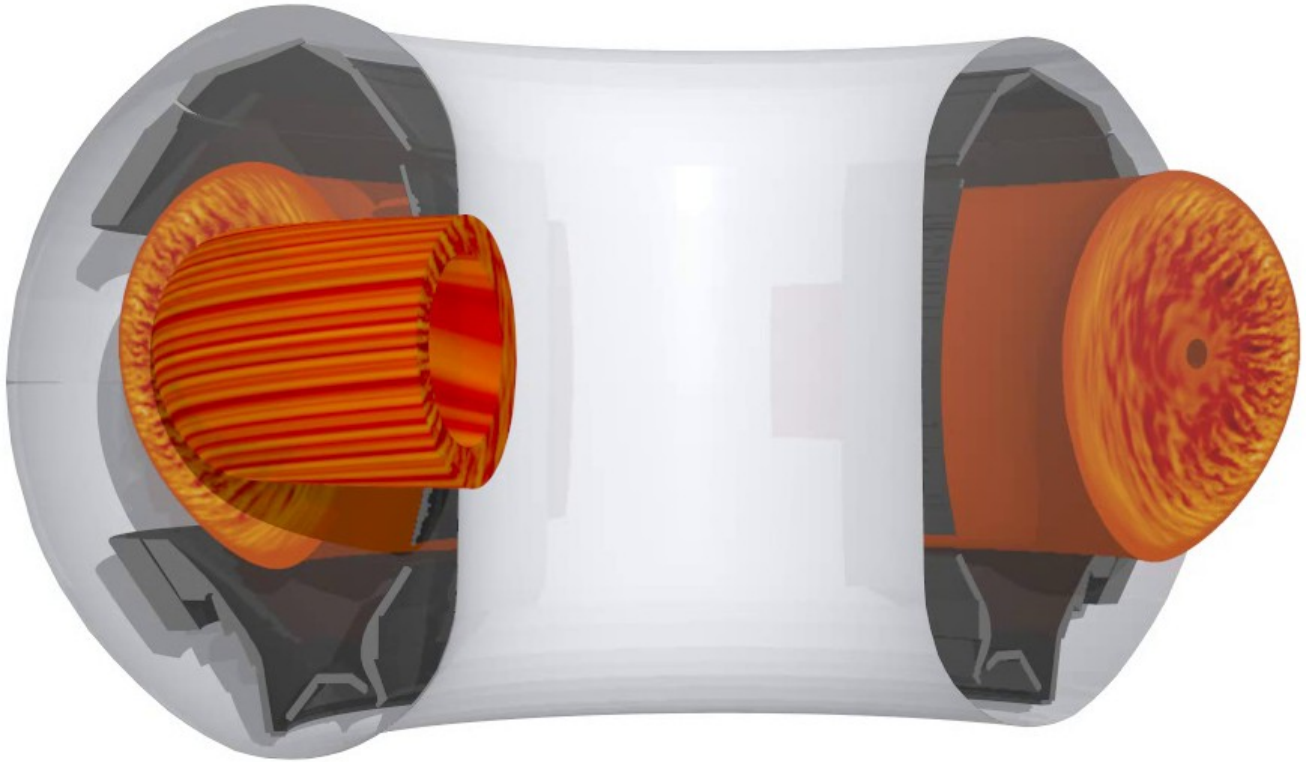
- the sparse grid combination technique is an extrapolation method which uses the the approximations $u(\gamma)$ as components of an approximation of the form

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

- the original combination technique uses a fixed set $I \subset \mathbb{N}^k$ and combination coefficients c_γ
- new methods have recently been developed where both I and c_γ are chosen to adapt to particular properties of the problem

Application

Fusion power and plasma physics



- Fusion could provide energy for millions of years[1]
- Challenge: energy loss through turbulent energy transport
- Numerical plasma simulations crucial to design and exploration (image: energy density from simulation)

[1] Wikipedia: Fusion power

Vlasov equations: a nonlinear system of intro-differential equations $u_t = F(u)$ with 6 state space variables which models microturbulent plasmas

$$\frac{\partial u_s}{\partial t} + \vec{v} \cdot \frac{\partial u_s}{\partial \vec{x}} + \frac{q_s}{m_s}(\vec{E} + \vec{v} \times \vec{B}) \cdot \frac{\partial u_s}{\partial \vec{v}} = 0, \quad s = 1, \ldots, S$$

where

- $u_s(\vec{x}, \vec{v}; t)$ plasma state space density of species (electrons and ions) s with charge q_s and mass m_s
- $\vec{E}$ and $\vec{B}$ electromagnetic fields, generated by plasma and external charges and current densities, solution of Maxwell equations

$$\rho = \sum_s q_s \int u_s \, dv \qquad \vec{j} = \vec{j}_0 + \sum_s q_s \int \vec{v} \, u_s \, dv$$

Numerical solver: GENE – Gyrokinetic Electromagnetic Numerical Experiment[2] (genecode.org)

- gyrokinetic (physical) approximation: rotation around magnetic field lines
- curved spatial coordinates aligned with magnetic field lines
  - 4th order stencils and spectral approximation in space
- velocity coordinates: component parallel to field lines and magnetic moment around field lines
  - 4th order stencil for derivative wrt parallel component, only integrals required for magnetic moment
- method of lines gives system of ODEs

$$\frac{du}{dt} = Lu + N(u)$$

  - 4th order Runge Kutta ODE solver
- eigenvalue problem with Jacobi-Davidson using SLEPc
- partition state space for parallelism (strong scalability)

[2] F. Jenko, 2000

Example of numerical parameters for GENE

- the Vlasov equations have a six-dimensional state space which is reduced to a five-dimensional space in GENE
- the curse of dimension limits the size of computationally feasible grids

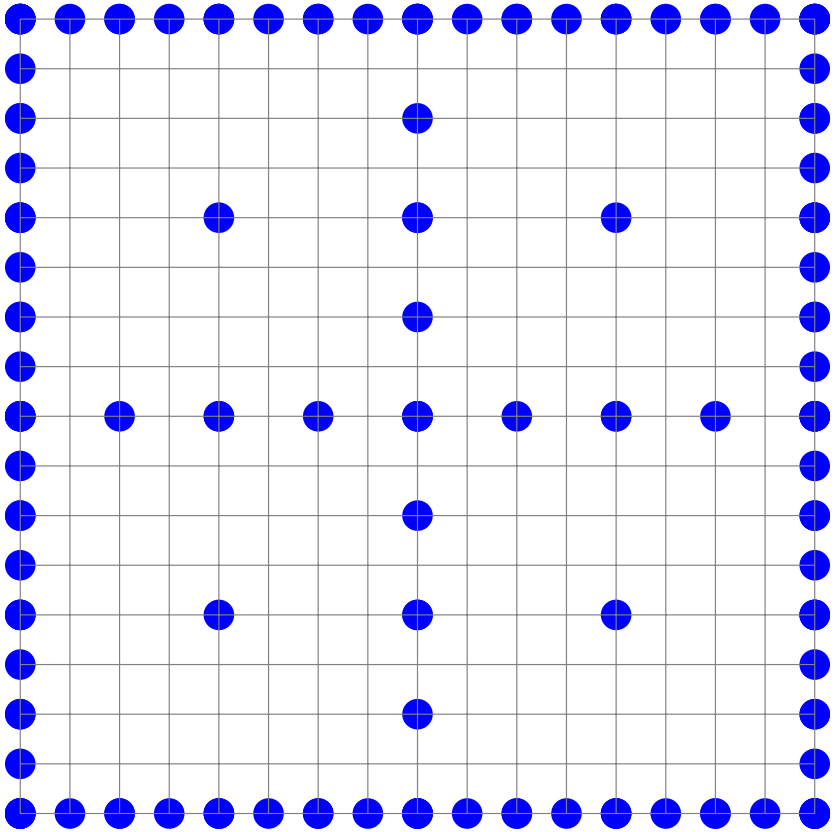
approach chosen here

- fix the numerical discretisation in x, y, z
- choose grids in velocity space[3] with grid size $2^{-\gamma_1}$ for parallel velocity and $2^{-\gamma_2}$ for angular momentum
- thus $\gamma \in \mathbb{N}^2$ in this case for the corresponding numerical approximation $u(\gamma)$

[3] note that a simple velocity-space approximation is used in the lattice Boltzmann method for CFD

Sparse grids

A typical sparse grid



- one can show that one only needs the function values at a small number of points to recover a smooth function
- number of grid points: $n(\log_2 n)^{d-1}$ where d is dimension and n is the number of grid points in one dimension

leads to substantial savings in computations

Zenger 1991, related idea: Smolyak 1963 (hyperbolic cross)

Nodal basis with hat functions in 1D

- hat function

$$\phi(x) = 1 - |x| \quad \text{for } x \in [-1, 1] \text{ and zero else}$$

- (Lagrangian) nodal basis using hat functions for fixed n:

$$\phi_j(x) = \phi(2^n x - j), \quad j = 0, \ldots, 2^n$$

- interpolant of function $u(x)$ with $h = 2^{-n}$

$$u_n(x) = \sum_{j=0}^{2^n} u(jh)\, \phi_j(x)$$

- $O(h^2)$ approximation if $u \in C^2$, similar finite element approximation
- 2 dimensions with $j = (j_1, j_2)$ and $x = (x_1, x_2)$

$$\phi_j(x) = \phi_{j_1}(2^{n_1} x_1 - j_1)\, \phi_{j_2}(2^{n_2} x_2 - j_2)$$

Hierarchical basis functions

- hierarchical (Schauder) basis in 1D for continuous functions

$$\phi_{j,n}(x) = \phi(2^n x - j), \quad n = 0, 1, \ldots, \quad j = 1, 3, \ldots, 2^n - 1$$

- representation

$$u(x) = \sum_{n,j} c_{n,j} \phi_{j,n}(x)$$

where $h_n = 2^{-n}$ and

$$c_{n,j} = u(jh_n) - \frac{1}{2}(u((j-1)h_n) + u((j+1)h_n)) = O(h_n^2)$$

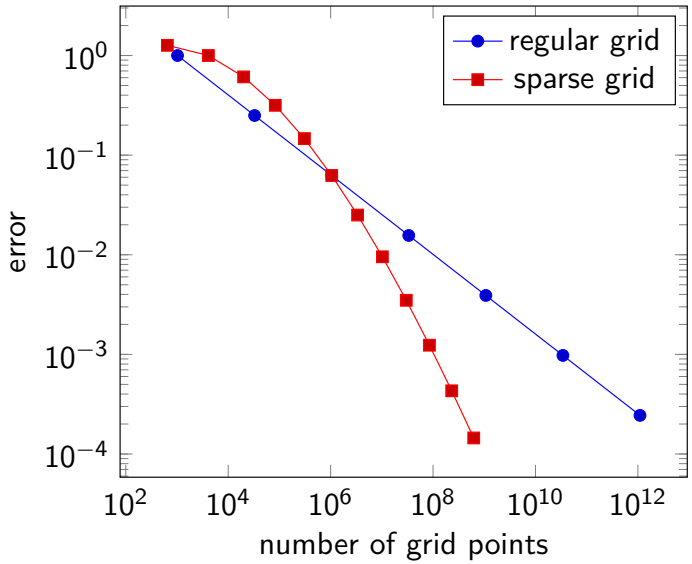
- $\phi_{j,n}$ for $n = 1, \ldots, k$ are an alternative basis of V_k
- in multiple dimensions product basis
- nodal basis preferred in FEM as matrix sparser

Sparse grid interpolant

$$u_S(x) = \sum_{n_1+n_2 \le n} \sum_{j} c_{n,j} \phi_{j_1,n_1}(x_1)\, \phi_{j_2,n_2}(x_2)$$

- coefficients $c_{j,n}$ are given by product difference formulas
- approximation $O(n^{d-1}\, h_n^2)$, here $d = 2$
- FEM with sparse grids: have to use hierarchical basis, rel. dense stiffness matrix
- also: has to redesign the whole code

Approximation with sparse grids for five dimensions



- approximation errors decrease with increasing number of grid points
- idea works well up to 10 dimensions
- can be extended

	number of points	L_2 error
regular grids	m^d	m^{-2}
sparse grids	$m\,(\log_2 m)^{d-1}$	$m^{-2}\,(\log_2 h)^{d-1}$

m is number of points in each dimension

Disadvantages of sparse grids

- requires a total redevelopment of the codes for hierarchical basis functions as most current codes use nodal basis
- the stiffness matrices etc turn out to be relatively dense
- the sparse grid approximation error bound is asymptotic in the size of the grids

Section 2

Sparse grid combination technique

A hierarchical decomposition

Decomposition of grid function $u(\gamma) \in V_\gamma$ into $w(\alpha) \in V_\alpha$

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

3 decompositions

- given one $u(\gamma)$ one defines $w(\alpha) = w(\alpha; \gamma)$ by
  - hierarchical decomposition based on interpolation
  - orthogonal decomposition where $(w(\alpha), w(\beta)) = \delta_{\alpha,\beta}$, for some scalar product, use projections
- given all $u(\alpha)$ for $\alpha \leq \gamma$ one has $w(\alpha)$ independent of γ
  - decomposition based obtained by solving equations for $w(\alpha)$ $\rightarrow$ next slides

motivation for the decomposition

- in some cases
$$\|w(\alpha)\| \leq K4^{-|\alpha|_1}$$
where $|\alpha|_1 = \alpha_1 + \cdots + \alpha_d$
- the values of $4^{-|\alpha|_1}$ arranged in our lattice 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 $w(0,0)$
  - not computing it creates error in sixth digit

why the error bound holds

- if $k = 1$ the piecewise linear interpolants $u(\gamma)$ lead to a $w(\gamma)$ which is called the *hierarchical surplus*
- it can be written as an integral over the second derivative
- this hierarchical surplus is of the order $4^{-\gamma}$ if the exact $u \in C^2$
- in higher dimensions one gets the same bounds on the surplus if $u \in C^{2,2,\ldots,2}$

error splitting

$$u(\gamma) = z_1 4^{-\gamma_1} + z_2 4^{-\gamma_2} + z_3(\gamma) 4^{-\gamma_1-\gamma_2}$$

- w in 2D

$$w(\gamma_1, \gamma_2) = u(\gamma_1, \gamma_2) - u(\gamma_1 - 1, \gamma_2) - u(\gamma_1, \gamma_2 - 1) + u(\gamma_1 - 1, \gamma_2 - 1)$$

- consequently

$$w(\gamma) = Z(\gamma) 4^{-\gamma_1-\gamma_2}$$

where $Z(\gamma$ can be written in terms of z_1, z_2 and $z_3(\gamma)$

decomposition of $u(\gamma)$ for $\gamma \leq (2,2)$

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

- multiply matrix of coefficients with vector of functions

$$\begin{bmatrix} u(2,2) \\ u(1,2) \\ u(2,1) \\ u(0,2) \\ u(1,1) \\ u(2,0) \\ u(0,1) \\ u(1,0) \\ u(0,0) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & & 1 & 1 & & 1 & 1 & 1 \\ & & 1 & & 1 & 1 & 1 & 1 & 1 \\ & & & 1 & & & 1 & & 1 \\ & & & & 1 & & 1 & 1 & 1 \\ & & & & & 1 & & 1 & 1 \\ & & & & & & 1 & & 1 \\ & & & & & & & 1 & 1 \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} w(2,2) \\ w(1,2) \\ w(2,1) \\ w(0,2) \\ w(1,1) \\ w(2,0) \\ w(0,1) \\ w(1,0) \\ w(0,0) \end{bmatrix}$$

computing the "surplus" $w(\alpha)$ from the $u(\gamma)$

- can determine the inverse of the previous matrix using the *inclusion-exclusion* principle

$$\begin{bmatrix} w(2,2) \\ w(1,2) \\ w(2,1) \\ w(0,2) \\ w(1,1) \\ w(2,0) \\ w(0,1) \\ w(1,0) \\ w(0,0) \end{bmatrix} = \begin{bmatrix} +1 & -1 & -1 & & +1 & & & & \\ & +1 & & -1 & -1 & & +1 & & \\ & & +1 & & -1 & -1 & & +1 & \\ & & & +1 & & & -1 & & \\ & & & & +1 & & -1 & -1 & +1 \\ & & & & & +1 & & -1 & \\ & & & & & & +1 & & -1 \\ & & & & & & & +1 & -1 \\ & & & & & & & & +1 \end{bmatrix} \begin{bmatrix} u(2,2) \\ u(1,2) \\ u(2,1) \\ u(0,2) \\ u(1,1) \\ u(2,0) \\ u(0,1) \\ u(1,0) \\ u(0,0) \end{bmatrix}$$

- row and column sums of matrix mostly equal to zero because
  - $\sum_{\alpha \leq \gamma} w(\alpha) = u(\gamma)$
  - if $u(\alpha) = u(0,0)$ for all $\alpha \leq \gamma$ then all $w(\alpha) = 0$ for $\alpha > (0,0)$

The sparse grid combination technique defined with the hierarchical decomposition

the sparse grid approximation of $u(\gamma)$ for $\gamma = (n, n, \ldots)$

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

- inserting the values from the previous equation gives the combination formula[4]

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

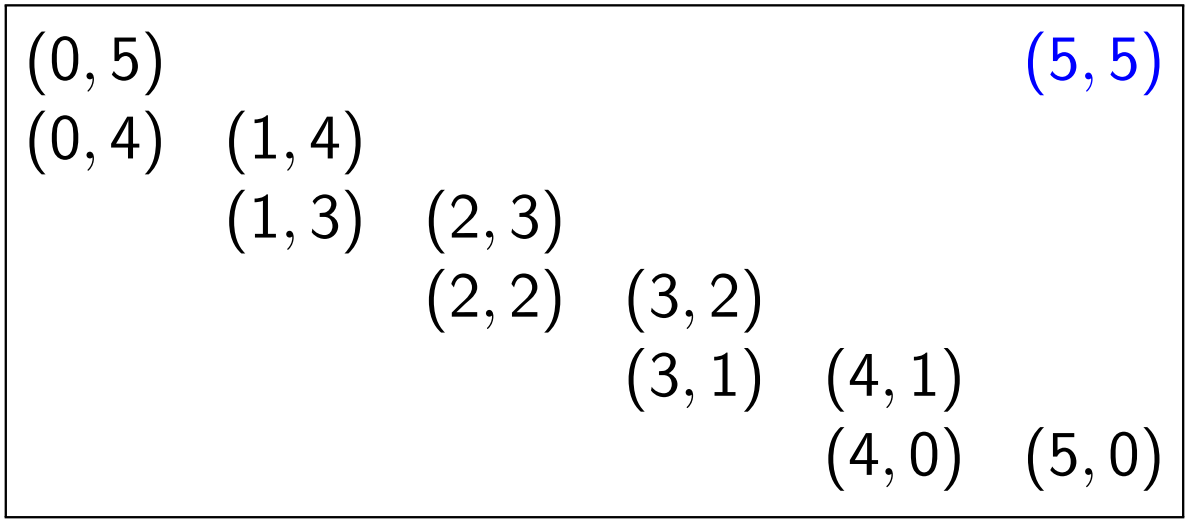
- in the case of $n = 2$ and $d = 2$ this is

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

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

grids used for sparse grid approximation

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



- omitting the grids above the diagonal contributes to the error
- the effects of the grids below the subdiagonal cancel exactly
- in d dimensions $d - 1$ subdiagonals are required

GENE decomposition of average particle density squared

The observable or quantity of interest f_s

- particle density of particle s

$$\rho_s(x, \gamma) = \int u(x, v; \gamma)\, dv$$

- average

$$f_s(\gamma) = \kappa \int \rho_s(x, \gamma)^2\, dx$$

- hierarchical decomposition of f_s

$$\sum_{\alpha \leq \gamma} q_s(\alpha) = f_s(\gamma)$$

The hierarchical decomposition of the quantity of interest

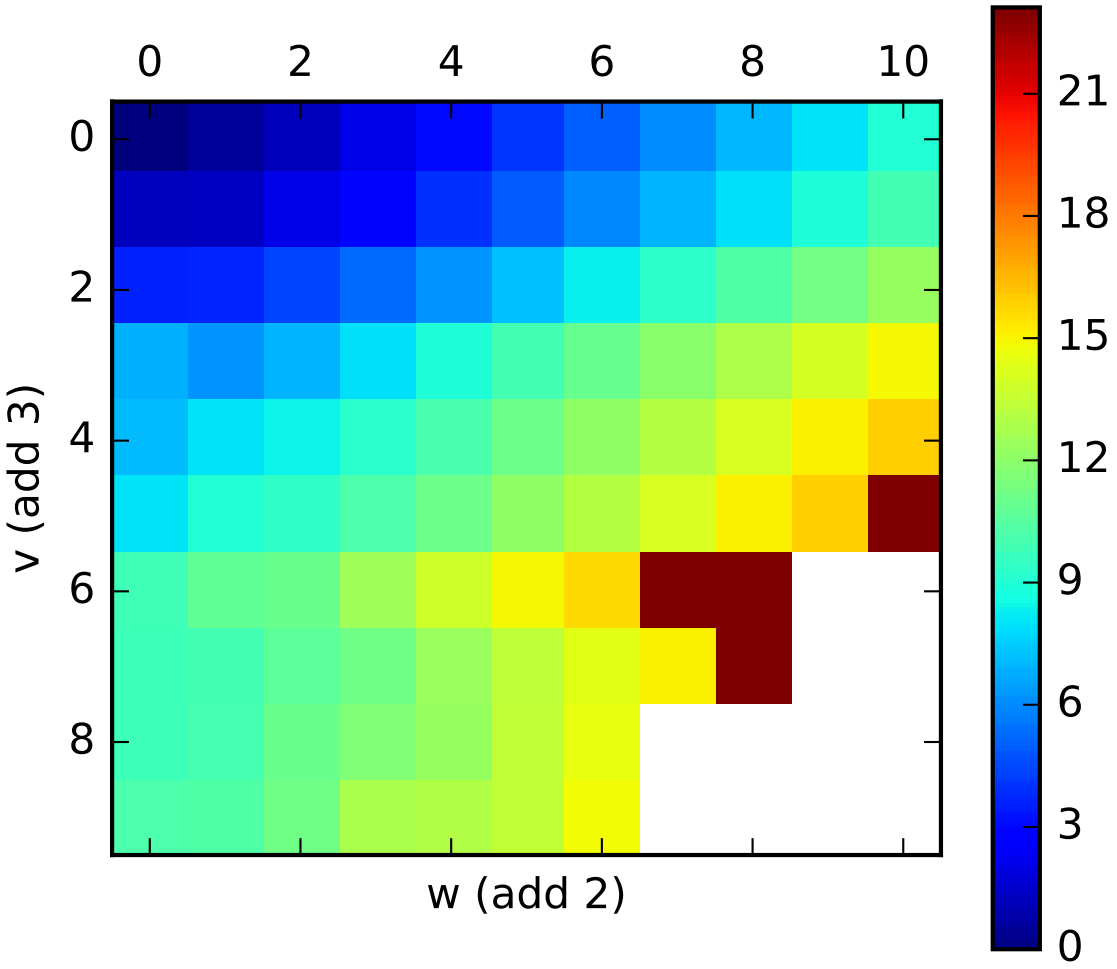


Figure 1:unequal approximation orders in the velocities

Section 3

New sparse grids

Choosing different sets *I*

The general form of the combination technique

- based on the hierarchical decomposition

$$u_C = \sum_{\alpha \in I} w(\alpha)$$

  - the set I is chosen to be a down-set, i.e., if $\beta \leq \alpha$ for some $\alpha \in I$ then $\beta \in I$

- the resulting combination formula

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

  - the coefficients are uniquely determined from the hierarchical decomposition
  - some of the coefficients c_γ may be zero

Why choose other sets I?

- Narrow grid cells (relating to γ with small and large components) often do not satisfy a general bound on $\|w(\alpha)\| \leq C4^{-|\alpha|}$
- Bounds of the type $\|w(\alpha)\| \leq \phi(|\alpha|)$ may just not hold at all
- Some of the components may be too costly to compute
- The computation of some of the components has failed and the corresponding components are thus not available (and the recomputation is too costly)

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[5]: $I = {\downarrow}\{\alpha \mid |\alpha| \leq n + d - 1,\ \alpha \geq \beta\}$
- 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[6]:
$$I = \bigcup_{k=1}^{d} \{\alpha \mid \alpha \leq n_0 1 + n_k e_k\}$$
- ANOVA:
$$I = \{\alpha \mid |\mathrm{supp}\,\alpha| \leq k\}$$

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

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

Examples

- ordinary sparse grid

$$\begin{array}{cccc} u & & & \\ u & u & & \\ w & u & u & \\ w & w & u & u \end{array}$$

- truncated sparse grid

$$\begin{array}{ccc} w & u & \\ w & u & u \\ w & w & w \end{array}$$

- with fault

$$\begin{array}{cccc} u & & & \\ w & 0 & & \\ u & w & u & \\ w & w & u & u \end{array}$$

- 2 scale

$$\begin{array}{cccc} w & u & & \\ w & w & & \\ w & u & w & u \\ w & w & w & w \end{array}$$

Robust and fault tolerant combination technique

What to do when some components $u(\gamma)$ are missing

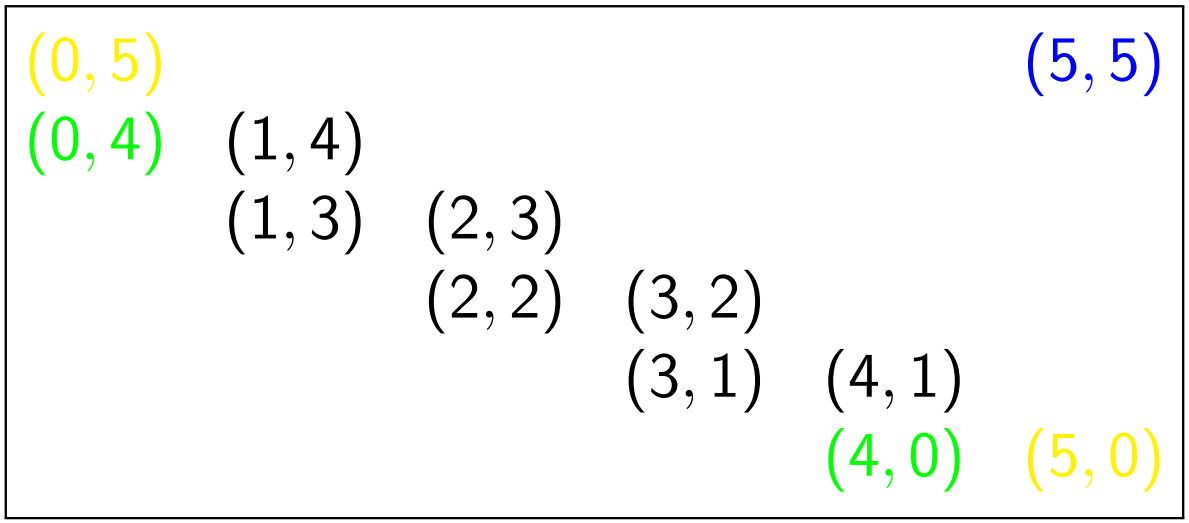
- large errors (predicted or detected) in $u(\gamma)$ because of limitations of physical model, numerical approximation or faults
  - error detection may use the $w(\alpha)$
- components $u(\gamma)$ have not been computed due to hardware issues
- some components $w(\alpha)$ are predicted to be very small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

approach

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

example 1: truncating extreme components

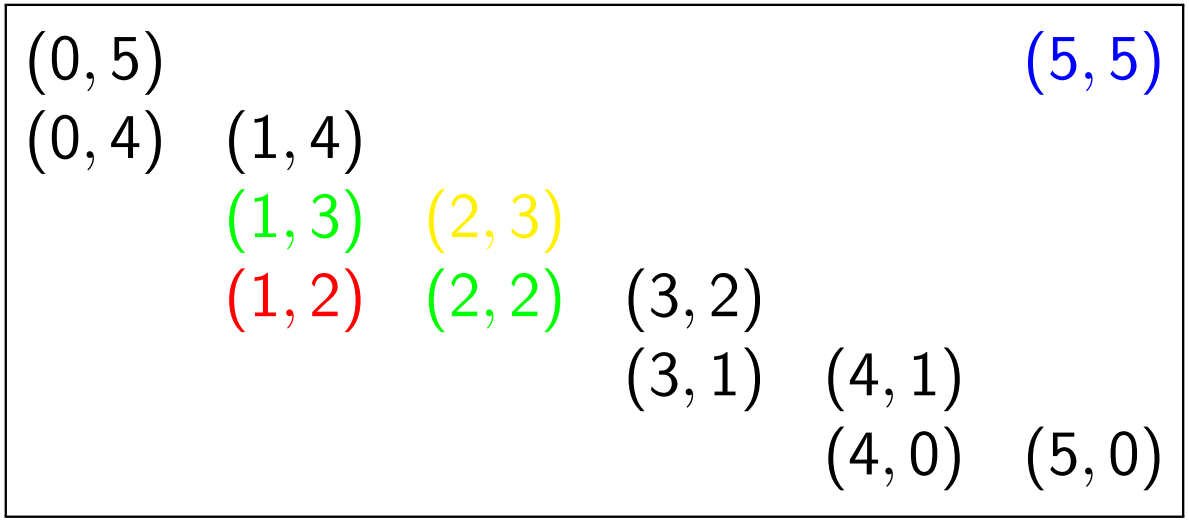
- case of $n = 5$ and $d = 2$ (target grid: blue)



- error increased by $w(0, 5) + w(5, 0)$ (yellow)
- components $u(0, 4)$ and $u(4, 0)$ are not used (green)
- overall, this truncation typically results in an actual decrease of the error

example 2: when a top level component is missing

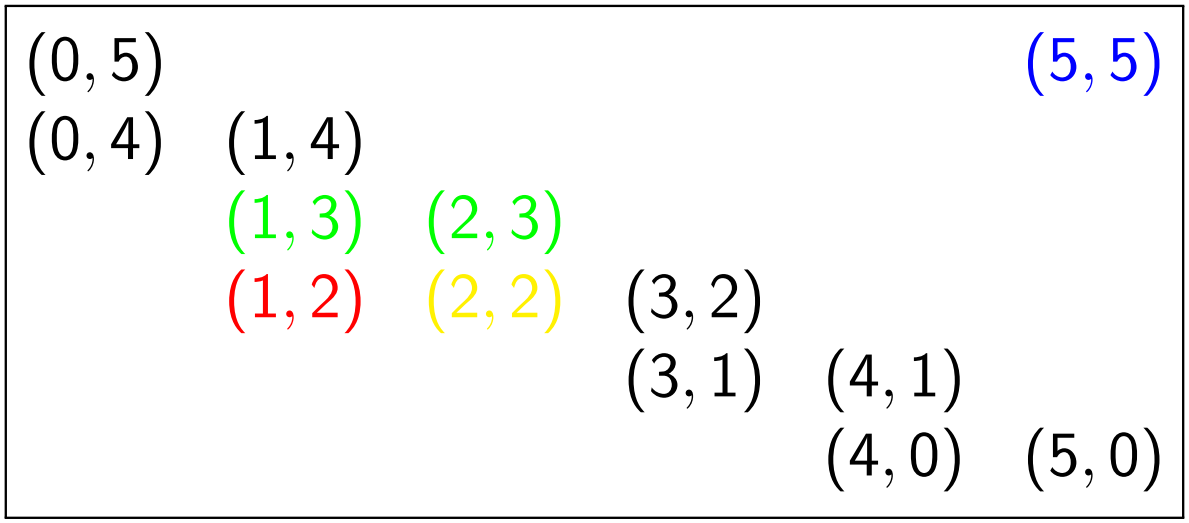
- case of $n = 5$ and $d = 2$ (target grid: blue)



- 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)

example 3: when a component at a lower level is missing

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



- 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

an adaptive approach

Dimension adaptive combination technique

$$u_I = \sum_{\alpha \in I} w(\alpha)$$

where

$$I = \{\alpha \leq \nu \mid \|w(\alpha)\| \geq \epsilon\} \text{ and } \nu = (n, \ldots, n)$$

- distance to accurate numerical solution $u(\gamma)$:

$$\|u(\gamma) - u_I\| \leq (n+1)^d \epsilon$$

- error bound

$$\|u - u_I\| \leq K 4^{-n} + (n+1)^d \, \epsilon$$

- proof: triangle inequality

Better error bounds

- if for some $0 < q < 1$ one has

$$\sum_{\alpha} \|w(\alpha)\|^q = \rho < \infty$$

  then the error of the adaptive method is bounded by

$$\|u(\gamma) - u_I\| \leq \sum_{\alpha \notin I} \|w(\alpha)\| \leq \rho\, \epsilon^{1-q}$$

- note that this error bound does not depend on the dimension

limitations

Limitation of combination approximations

- The combination technique is basically an extrapolation method and errors in the components may be amplified substantially, especially in higher dimensional problems (see PhD thesis J. Garcke)
- In fact, the combination technique may produce approximations which are worse than all the components $u(\gamma)$

Section 4

Optimisation problems

Opticom

Ritz method for optimisation

- convex optimisation problem

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

- Ritz method

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

- Ritz method optimal in terms of the Bregman distance

$$D(v, u) := J(v) - J(u) - \langle \nabla J(u), v - u \rangle$$

  i.e.

$$D(v(\gamma), u) \leq D(v, u), \quad \text{for all } v \in V_\gamma$$

- Ritz method quasi-optimal if one has

$$C_1 \|v - u\| \leq D(v, u)^q \leq C_2 \|v - u\|$$

Opticom – a sparse grid combination approximation based on Ritz

- Opticom

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

- quasi optimal for appropriate norm

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

- thus this combination is (up to constant) at least as good as component approximations $u(\gamma)$
- can make this fault tolerant: set constraint $c_\alpha = 0$ for missing (or underperforming) $u(\alpha)$

Combination approximation error

The error of a general combination approximation

- the error of a general combination approximation of the sparse grid solution is

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

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

- for the sparse grid solution the value in the bracket is zero
- we consider the case where certain coefficients c_γ are zero as the corresponding component $u(\gamma)$ is not available or suitable

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 $J(\mathbf{c})$:

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

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

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

  this is a piecewise linear optimisation problem with constraints

- the form of the objective function is motivated by the bounds

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

- approximation error is then bounded by

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

eigenvalue problems

Least squares method for eigenvalue problem $Au = \lambda u$

- assume $(s, u) = 1$ for some $s \in H$
- determine $u_n \in V_n$ satisfying $(s, u_n) = 1$ and $\lambda_n \in \mathbb{C}$ with maximal real part such that

$$\|Au_n - \lambda_n u_n\| \leq \|Av - \mu v\|$$

  for all $v \in V_n$ with $(s, v) = 1$ and $\mu \in \mathbb{C}$
- it follows that[7]

$$\|Au_n - \lambda_n u_n\| \leq \|(A - \lambda I)P_n u\| \leq \|A - \lambda I\| \, \|(I - P_n)u\|$$

- error bounds from Bauer-Fike theorem (1960) if $A = V^{-1}\Lambda V$

$$|\lambda - \lambda_n| \leq \kappa(V) \frac{\|Au_n - \lambda_n u_n\|}{\|u_n\|}$$

[7] e.g. Y.Saad 1983

Using Opticom for eigenvalue problems

$$\Phi(v, \mu) = \|Av - \mu v\|$$

- standard combination technique for eigenvalues[8]

$$\lambda_C = \sum_{|\gamma| \leq n} c_\gamma \lambda(\gamma)$$

- Opticom approximation for eigenvalue problem[9]

$$(u_O, \lambda_O) = \text{argmin}_{c_\gamma, \mu} \left\| (A - \mu I) \sum_{\gamma \in I} c_\gamma u(\gamma) \right\|$$

where $\sum_{\gamma \in I} c_\gamma = 1$

[8] Garcke '01, Kowitz '15

[9] Kowitz and H. '15

Computing the approximation[10]

- an augmented problem

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

- elimination of c gives

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

- solve equation for eigenvalue $\beta(\lambda) = 0$ with Newton
- application to the combination technique with $G = [u(\gamma)]_{\gamma \in I}$

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

- thus we get the computational method:
  1. solve the component problems to get $u(\gamma)$
  2. reduction operation to compute the matrix $K(\lambda)$
  3. solve optimisation problem for c_γ and λ and combine

[10] Osborne 1964, Kowitz 2015

A computational 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 $u(\gamma)$

References

- F. Bauer and C. Fike in Numerische Mathematik 2, 1960
- M. Osborne, in The Computer Journal 7, 1964
- Y. Saad, in Lecture Notes in Mathematics 973, Springer 1983
- F. Jenko et al, in Phys. Plasmas 7, 2000
- M.H., in Proc. of the 5th Int. Conf. on Curves and Surfaces, Nashboro Press 2003
- M.H., J. Garcke, V. Challis, in Linear Algebra and its Applications 420, 2007
- C. Kowitz, D. Pfluger, F. Jenko and M. H. in Sparse Grids and Applications, Springer 2013
- C. Kowitz, PhD thesis, TU Munich 2016