Algorithms of Scientific Computing

Hierarchical Methods and Sparse Grids

Tobias Neckel, Dirk Pflüger

Technische Universität München

Summer Term 2010

Part II

Cost and Accuracy

So far. . .

- Hierarchical and non-hierarchical one-dimensional quadrature
- Aim: dealing with high-dimensional functions
- Quadrature as an example: well-studied, relatively simple

- On the way to high dimensionalities we have to consider whether effort (measured in function evaluations, computations, . . .) is well-invested?

$\Rightarrow$ Consider ratio of cost vs. accuracy

ϵ-Complexity

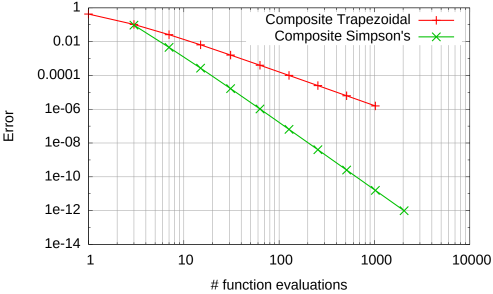
- Numerical methods: usually approximate solution with error ϵ
  - Error can be due to discretization, rounding, truncation, ...
- To measure cost W: count operations
- Relate cost W to error ϵ
  - How many operations $W(\epsilon)$ required to obtain error of at most ϵ?
- To this end: assumptions about solution again (differentiability, bounds for derivatives, ...)
  - Often don't hold in real-world settings
  - But good indication to compare different methods

ϵ-Complexity

- Numerical methods: usually approximate solution with error ϵ
  - Error can be due to discretization, rounding, truncation, ...
- To measure cost W: count operations
- Relate cost W to error ϵ
  - How many operations $W(\epsilon)$ required to obtain error of at most ϵ?
- To this end: assumptions about solution again (differentiability, bounds for derivatives, ...)
  - Often don't hold in real-world settings
  - But good indication to compare different methods
- Composite trapezoidal rule with n subintervals:
  - n+1 evaluations
  - Error $\mathcal{O}(n^{-2})$ (sufficiently smooth)
  - ϵ-complexity $W(\epsilon) = \mathcal{O}(\sqrt{1/\epsilon})$ (unit: number of evaluations)
- Composite Simpson's rule correspondingly $W(\epsilon) = \mathcal{O}(\sqrt[4]{1/\epsilon})$

CT and CS: Example

- Cost-error diagram for $F_1 := \int_0^\pi \sin(x)\,dx$:
  $|CT - F_1|$ and $|CS - F_1|$
- No function evaluations on the boundary



- ϵ-complexities $\mathcal{O}(\sqrt{1/\epsilon})$ and $\mathcal{O}(\sqrt[4]{1/\epsilon})$

$\rightsquigarrow$ Different gradients of the curves (asymptotically for large n; double-logarithmic scale)

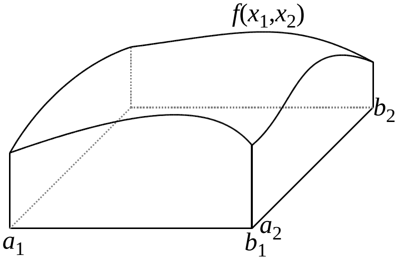
Multi-Dimensional Quadrature

- Now on to multi-dimensional functions:

$$\text{Area of integration } \Omega := \prod_{k=1}^{d} [a_k, b_k], \text{ function } f : \Omega \to \mathbb{R}$$

- Compute approximation for

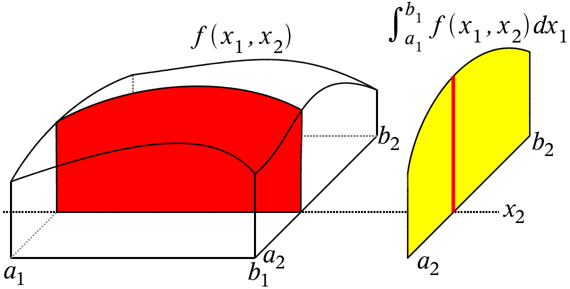
$$F_d(f, \Omega) := \int_\Omega f(x_1, \ldots, x_d)\, d\vec{x}.$$



Decomposition into One-Dimensional Integrals

- Decompose *d*-dimensional integral into sequence of one-dimensional ones (cf. Fubini's Theorem)

$$F_d(f,\Omega) = \int_{a_d}^{b_d} \cdots \int_{a_2}^{b_2} \left(\int_{a_1}^{b_1} f(x_1,\ldots,x_d)\, dx_1 \right) dx_2 \ldots dx_d.$$



Decomposition: Implementation

- Consider this decomposition using the function F_1 (one-dimensional integration), and functions G_k:

$$G_0(x_1, x_2, x_3, \ldots, x_d) \quad := \quad f(x_1, x_2, x_3, \ldots, x_d)$$

Decomposition: Implementation

- Consider this decomposition using the function F_1 (one-dimensional integration), and functions G_k:

$$
\begin{aligned}
G_0(x_1, x_2, x_3, \ldots, x_d) \quad &:= \quad f(x_1, x_2, x_3, \ldots, x_d) \\
G_1(x_2, x_3, \ldots, x_d) \quad &:= \quad F_1(G_0(\bullet, x_2, x_3, \ldots, x_d), a_1, b_1)
\end{aligned}
$$

Decomposition: Implementation

- Consider this decomposition using the function F_1 (one-dimensional integration), and functions G_k:

$$
\begin{aligned}
G_0(x_1, x_2, x_3, \ldots, x_d) &:= f(x_1, x_2, x_3, \ldots, x_d) \\
G_1(x_2, x_3, \ldots, x_d) &:= F_1(G_0(\bullet, x_2, x_3, \ldots, x_d), a_1, b_1) \\
G_2(x_3, \ldots, x_d) &:= F_1(G_1(\bullet, x_3, \ldots, x_d), a_2, b_2)
\end{aligned}
$$

Decomposition: Implementation

- Consider this decomposition using the function F_1 (one-dimensional integration), and functions G_k:

$$
\begin{aligned}
G_0(x_1, x_2, x_3, \ldots, x_d) &:= f(x_1, x_2, x_3, \ldots, x_d) \\
G_1(x_2, x_3, \ldots, x_d) &:= F_1(G_0(\bullet, x_2, x_3, \ldots, x_d), a_1, b_1) \\
G_2(x_3, \ldots, x_d) &:= F_1(G_1(\bullet, x_3, \ldots, x_d), a_2, b_2) \\
&\vdots \\
G_d() &:= F_1(G_{d-1}(\bullet), a_d, b_d)
\end{aligned}
$$

- G_k integrates over $x_1, \ldots, x_k$; remaining variables free

Decomposition: Implementation

- Consider this decomposition using the function F_1 (one-dimensional integration), and functions G_k:

$$
\begin{aligned}
G_0(x_1, x_2, x_3, \ldots, x_d) &:= f(x_1, x_2, x_3, \ldots, x_d) \\
G_1(x_2, x_3, \ldots, x_d) &:= F_1(G_0(\bullet, x_2, x_3, \ldots, x_d), a_1, b_1) \\
G_2(x_3, \ldots, x_d) &:= F_1(G_1(\bullet, x_3, \ldots, x_d), a_2, b_2) \\
&\vdots \\
G_d() &:= F_1(G_{d-1}(\bullet), a_d, b_d)
\end{aligned}
$$

- G_k integrates over $x_1, \ldots, x_k$; remaining variables free

Numerical quadrature

- Just replace F_1 by a quadrature formula, e.g. CT, CS

Cost and Accuracy

Cost

- Uniform grid with n subintervals for 1d quadrature
- d dimensions: Cartesian product of 1d grids
- Indices

$$(i_1, \ldots, i_d) \in \{0, 1, 2, \ldots, n\}^d$$

with corresponding grid points

$$(x_1, \ldots, x_d) \text{ with } x_k = a_k + i_k \frac{b_k - a_k}{n}$$

- Total cost:
  - $(n+1)^d$ (with grid points on domain's boundary $\partial\Omega$)
  - $(n-1)^d$ (if f is zero on $\partial\Omega$)

Cost and Accuracy (2)

Accuracy

- Still $\mathcal{O}(n^{-2})$ for CT, $\mathcal{O}(n^{-4})$ for CS
- Remark: starting with G_2, the current function values are erroneous by $\mathcal{O}(n^{-2})$ and $\mathcal{O}(n^{-4})$ resp.; this does not alter the overall accuracy

$\Rightarrow$ Thus everything is fine. . . ?

Multidimensional Quadrature: Example

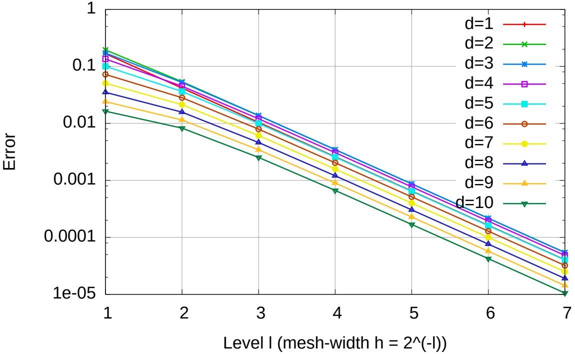
- Integration of

$$f(x_1, \ldots, x_d) := \prod_{k=1}^{d} 4x_k(1 - x_k)$$

on $\Omega = [0, 1]^d$ with the composite Trapezoidal rule

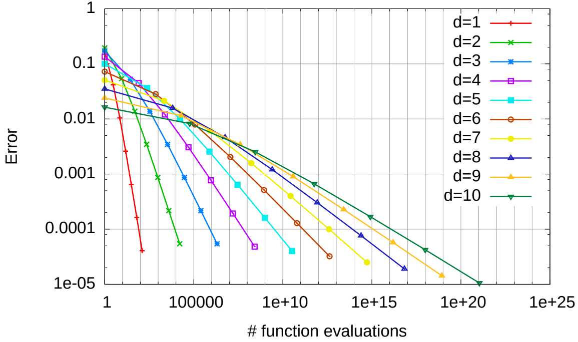
Multidimensional Quadrature: Example

- Integration of
$$f(x_1, \ldots, x_d) := \prod_{k=1}^{d} 4x_k(1 - x_k)$$
on $\Omega = [0, 1]^d$ with the composite Trapezoidal rule
- Error:



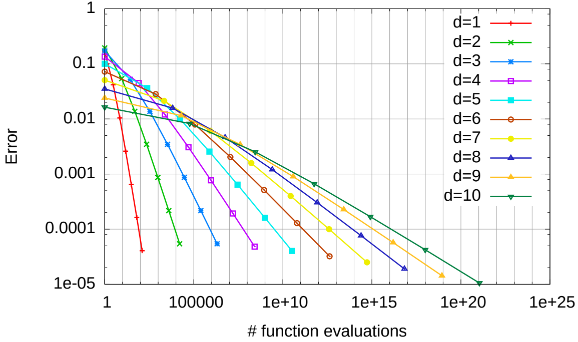
Multidimensional Quadrature: Example (2)

- Having ϵ-complexity in mind:
  - Use cost (number of function evaluations) as abscissa



Multidimensional Quadrature: Example (2)

- Having ϵ-complexity in mind:
  - Use cost (number of function evaluations) as abscissa



- Does not look that good any more. . .

Multidimensional Quadrature: Example (3)

10^{21}

- Large number…

Multidimensional Quadrature: Example (3)

10^{21}

- Large number...
- 1 ZByte (Zeta) = 1.000.000.000 TByte = 1.000.000.000.000.000.000.000 Byte to store grid (one Byte per grid point)

Multidimensional Quadrature: Example (3)

10^{21}

- Large number...
- 1 ZByte (Zeta) = 1.000.000.000 TByte = 1.000.000.000.000.000.000.000 Byte to store grid (one Byte per grid point)
- Compare national super computer HLRBII (Altix) @ LRZ:
    - Peak performance: 62.3 TFlop/s
    - Memory: 39 TByte
- It would take 6 months to compute quadrature, assuming that one integration operation can be performed in one clock cycle...

Curse of Dimensionality

ϵ-complexity

- CT: $\mathcal{O}(\epsilon^{-\frac{d}{2}})$, CS: $\mathcal{O}(\epsilon^{-\frac{d}{4}})$

Curse of Dimensionality

ϵ-complexity

- CT: $\mathcal{O}(\epsilon^{-\frac{d}{2}})$, CS: $\mathcal{O}(\epsilon^{-\frac{d}{4}})$

Curse of dimensionality

- Exponential dependency on dimensionality d
- Higher-dimensional problems infeasible to tackle ($d = 10$ is still moderate...)
- Property of the problem – or just of the algorithm?
- It's the algorithm $\Rightarrow$ hierarchical methods can mitigate the curse of dimensionality to some extent

Monte-Carlo Integration

- To motivate the search for better methods for numerical quadrature:
  - Consider Monte-Carlo method: simple approach, simple to implement

Monte-Carlo Integration

- To motivate the search for better methods for numerical quadrature:
  - Consider Monte-Carlo method: simple approach, simple to implement

Approach

- Be X a random variable, uniformly distributed on Ω
- Then it holds for the expectation value

$$E(f(X)) = \int_\Omega \frac{f(x)}{\text{Vol}(\Omega)}\,dx = \frac{1}{\text{Vol}(\Omega)} F_d(f,\Omega)$$

Monte-Carlo Integration

- To motivate the search for better methods for numerical quadrature:
  - Consider Monte-Carlo method: simple approach, simple to implement

Approach

- Be X a random variable, uniformly distributed on Ω
- Then it holds for the expectation value

$$E(f(X)) = \int_\Omega \frac{f(x)}{\mathrm{Vol}(\Omega)}\,dx = \frac{1}{\mathrm{Vol}(\Omega)} F_d(f,\Omega)$$

- On the other hand: if x_k are realizations of X we obtain

$$\lim_{M\to\infty} \frac{1}{M}\sum_{k=1}^{M} f(x_k) = E(f(X))$$

with probability 1: strong law of large numbers

Monte-Carlo Integration (2)

- Simple to implement
- Cost completely independent of d (counting function evaluations)

Monte-Carlo Integration (2)

- Simple to implement
- Cost completely independent of d (counting function evaluations)
- Accuracy?
    - Estimate stochastically: compute standard deviation (use additivity of variances)

$$\sqrt{\operatorname{Var}\left(\frac{1}{M}\sum_{k=1}^{M} f(x_k)\right)} = \sqrt{\frac{1}{M^2}\sum_{k=1}^{M}\operatorname{Var}(f)} = \sqrt{\frac{\operatorname{Var}(f)}{M}}$$

    - Independent of d, too
    - Dependencies of d only in $\operatorname{Var}(f)$ and $\operatorname{Vol}(\Omega)$ possible; does not affect exponent of M

Monte-Carlo Integration (2)

- Simple to implement
- Cost completely independent of d (counting function evaluations)
- Accuracy?
  - Estimate stochastically: compute standard deviation (use additivity of variances)

$$\sqrt{\mathrm{Var}\left(\frac{1}{M}\sum_{k=1}^{M} f(x_k)\right)} = \sqrt{\frac{1}{M^2}\sum_{k=1}^{M}\mathrm{Var}(f)} = \sqrt{\frac{\mathrm{Var}(f)}{M}}$$

  - Independent of d, too
  - Dependencies of d only in $\mathrm{Var}(f)$ and $\mathrm{Vol}(\Omega)$ possible; does not affect exponent of M
- Thus (stochastically) ϵ-complexity of $\mathcal{O}(\epsilon^{-2})$
  - Very slow convergence
  - Independence of d: very helpful tackling high-dimensional problems!

What next?

- We know, that the curse of dimensionality can be overcome
- Search for alternative (better?) methods
  - . . . which can be used for more applications than integration, for example