Algorithms of Scientific Computing
Hierarchical Methods and Sparse Grids

Michael Bader
Technische Universität München

Summer Term 2013
Part IV

Archimedes, $d$-Dimensional
Current State

One-dimensional quadrature

- One-dimensional functions $f$, interval $[a, b]$
- Compute approximation $F_1(f, a, b)$ of area:

\[ F_1(f, a, b) \approx \int_a^b f(x) \, dx \]

- Notation for approximation of exact integral value in the following: $F_d(.)$, with $d$ as the dimension
- One-dimensional quadrature rules:
  - Composite trapezoidal rule
  - Composite Simpson’s rule
  - Archimedes’ quadrature
Multi-Dimensional Quadrature

Consider multi-dimensional setting

$$F_d(f, \Omega) \approx \int_{\Omega} f(x_1, \ldots, x_d) \, d\vec{x}, \quad \Omega := \prod_{k=1}^{d} [a_k, b_k]$$
First Attempt

- remember theorem of Fubini:

\[ F_d(f, \Omega) = \int_{a_1}^{b_1} \ldots \int_{a_d}^{b_d} f(x_1, \ldots, x_d) \, dx_1 \ldots dx_d \]

- Use full-grid approach as before:

\[
G_0(x_1, x_2, x_3, \ldots, x_d) := f(x_1, x_2, x_3, \ldots, x_d)
\]

\[
G_1(x_2, x_3, \ldots, x_d) := F_1(G_0(\bullet, x_2, x_3, \ldots, x_d), a_1, b_1)
\]

\[
G_2(x_3, \ldots, x_d) := F_1(G_1(\bullet, x_3, \ldots, x_d), a_2, b_2)
\]

\[
\vdots
\]

\[
G_d() := F_1(G_{d-1}(\bullet), a_d, b_d)
\]

- We now consider the effect of Archimedes’ quadrature as one-dimensional quadrature method for \( F_1 \)
First Attempt: Employing Archimedes

- \( d \) nested loops \((x_1, x_2, \ldots)\)
- Summation of weighted function values
- No real advantages apart from adaptivity (which is not very useful this way)

Interplay of hierarchization and summation (integration)

- Consider setting with \( d = 2 \)
- First, compute integrals in \( x_1 \)-direction: \( F_1(G_0(\bullet, x_2), a_1, b_1) \)
  - Involves hierarchization in \( x_1 \)-direction
  - But no impact on \( G_1(x_2) \)
- \( G_1(x_2) \): no hierarchical values, thus all \( G_1(x_2) \) of same order
- After summation (integration) in \( x_1 \)-direction:
  - Hierarchization in \( x_2 \)-direction
  - Finally summation in \( x_2 \)-direction
Consider computing \( G_1(x_2) \)

- We are only interested in hierarchical surplus
- Hierarchical surplus typically much smaller than function value
  \[ \Rightarrow \] Could be computed with much less grid points in \( x_1 \)-direction

We change the order of “integration in \( x_1 \)-direction” and “hierarchization in \( x_2 \)-direction”

- Write hierarchical area elements of quadrature in \( x_2 \)-direction (trapezoid, segments, triangles) as function of \( x_1 \)
- Integrate those in \( x_1 \)-direction

Now interplay of dimensions for integration much more complicated

\[ \ldots \] but this will lead to much more efficient method
Example, 2d

Consider

\[ f(x_1, x_2) := \left( x_1 + \frac{1}{2} \right) \left( x_1 - \frac{3}{2} \right) \left( x_2 + \frac{1}{2} \right) \left( x_2 - \frac{3}{2} \right) \]

on \( \Omega = [0, 1] \times [0, 2] \)
Trapezoidal Volume and Remainder Segment

First step of the hierarchical decomposition

\[ F_2(f, \Omega) = F_1(T_2, a_1, b_1) + S_2(f, \Omega) \]

“Green function” → linear interpolation of values at \( a_2, b_2 \):

\[ f(x_1, a_2)(b_2 - x_2) + f(x_1, b_2)(x_2 - a_2) \]

\[ \frac{b_2 - a_2}{b_2 - a_2} \]

for any \( x_1 \)
Trapezoidal Volume and Remainder Segment (2)

Decompose volume into

- trapezoidal (for constant $x_1$) cross-section with area

$$T_2(x_1) := \frac{b_2 - a_2}{2} \left( f(x_1, a_2) + f(x_1, b_2) \right),$$

→ will be integrated in $x_1$-direction using quadrature rule $F_1$

- and remainder segment

$$S_2(f, \Omega) := F_2(f, \Omega) - F_1(T_2, a_1, b_1)$$

$$= \int_{a_2}^{b_2} \int_{a_1}^{b_1} \left( f(x_1, x_2) - \frac{f(x_1, a_2)(b_2 - x_2) + f(x_1, b_2)(x_2 - a_2)}{b_2 - a_2} \right) \, dx_1 \, dx_2$$

Note: $T_2$ is the integral over the linear interpolation (“green function”)
Triangular Volumes and Remainder Segments

Second step of the hierarchical decomposition

\[ S_2(f, \Omega) = F_1(D_2, a_1, b_1) + S_2(f, \ldots) + S_2(f, \ldots) \]

again: hierarchization in \( x_2 \)-direction; integrate in \( x_1 \)-direction
Decompose remainder segment \( S_2(f, \Omega) \) into

- triangular (for constant \( x_1 \)) cross-section with area

\[
D_2(x_1) := \frac{b_2 - a_2}{2} \left( f \left( x_1, \frac{a_2 + b_2}{2} \right) - \frac{f(x_1, a_2) + f(x_1, b_2)}{2} \right)
\]

- and two remainder segments

\[
S_2(f, [a_1, b_1] \times [a_2, b_2]) = F_1(D_2, a_1, b_1) + S_2(f, [a_1, b_1] \times \left[ a_2, \frac{a_2 + b_2}{2} \right] ) + S_2(f, [a_1, b_1] \times \left[ \frac{a_2 + b_2}{2}, b_2 \right] )
\]
Recursive decomposition

- Repeat last step for both remainder segments
- Decompose each into triangular sub-volume and two remainder segments
- Example for one of the two segments and sum of trapezoidal and first three triangular sub-volumes:
Recursive Structure of Function Calls

- Nested recursive structure of function calls
- For higher-dimensional problems: one more level \((F_d \text{ and } S_d)\) for each additional dimension

- Consider number of function evaluations for grid point inside of \(\Omega\)
  - Straightforward: \(3^d\) evaluations to compute surplus
  - All but one have already been computed!
Subvolumes

- \( F_1 \): the subvolumes (hierarchized in \( x_2 \)-direction) are decomposed (in \( x_1 \)-direction) into trapezoid and many triangles
- Integrand itself is area (one slice trapezoidal/triangular subareas)
- Subvolumes which are added in quadrature are pagodas (neglecting trapezoidals)
  - Height of pagodas: \( d \)-dimensional hierarchical surplus
  - Volume of pagodas: \( 2^{-d} \) times size of support times surplus (more in next part)
- Taking stopping criterion depending on surplus (\( d \) criteria: one in \( S_i \) each)
  - Find those grid points for which function evaluation is worthwhile
  - In general much less than naive implementation
- Extend from composite trapezoidal rule to Simpsons’ as in one-dimensional setting