Nonlinear optimization via summation and integration

(MINLP: A Space Odyssey)

Matthias Köppe

UC Davis, Mathematics

based on joint works with
V. Baldoni, N. Berline, J. A. De Loera,
R. Hemmecke, M. Vergne, R. Weismantel

Nonlinear optimization via summation and integration

(MINLP: A Space Odyssey)

Matthias Köppe

UC Davis, Mathematics

based on joint works with
V. Baldoni, N. Berline, J. A. De Loera,
R. Hemmecke, M. Vergne, R. Weismantel

Outline

Outline

The main idea

Optimization (maximization) is just the limit case of power-of-p integration (summation)

$$\lim_{p\to\infty} ||f||_p = ||f||_\infty$$

Classic idea, variants re-appear in the optimization literature

- A.I. Barvinok, *Exponential integrals and sums over convex polyhedra* (Russian), Funktsional. Anal. i Prilozhen. 26 (1992).
- J.B. Lasserre, *Generating functions and duality for integer programs*, Discrete Optim. 1 (2004).

The main idea

Optimization (maximization) is just the limit case of power-of-p integration (summation)

$$\lim_{p \to \infty} ||f||_p = ||f||_\infty$$

Classic idea, variants re-appear in the optimization literature

- A.I. Barvinok, *Exponential integrals and sums over convex polyhedra* (Russian), Funktsional. Anal. i Prilozhen. 26 (1992).
- J.B. Lasserre, *Generating functions and duality for integer programs*, Discrete Optim. 1 (2004).

The main idea

Optimization (maximization) is just the limit case of power-of-p integration (summation)

$$\lim_{p\to\infty} ||f||_p = ||f||_\infty$$

Classic idea, variants re-appear in the optimization literature

- A.I. Barvinok, *Exponential integrals and sums over convex polyhedra* (Russian), Funktsional. Anal. i Prilozhen. 26 (1992).
- J.B. Lasserre, *Generating functions and duality for integer programs*, Discrete Optim. 1 (2004).

The main idea, the discrete case

Consider the (discrete) optimization problem $\max\{\, f(\mathbf{x}) : \mathbf{x} \in S \,\}$ where f is non-negative on S and the feasible region S is finite.

Approximation properties of ℓ_p norms

$$S = \{\mathbf{x}^1, \dots, \mathbf{x}^N\} \qquad \text{and} \qquad \mathbf{f} = \begin{pmatrix} f(\mathbf{x}^1) \\ \vdots \\ f(\mathbf{x}^N) \end{pmatrix} \in \mathbf{R}^N.$$

Then

$$\lim_{p\to\infty} \|\mathbf{f}\|_p = \lim_{p\to\infty} \big(f(\mathbf{x}^1)^p + \dots + f(\mathbf{x}^N)^p\big)^{1/p}$$
$$= \max\{f(\mathbf{x}^1), \dots, f(\mathbf{x}^N)\} = \|\mathbf{f}\|_\infty$$

More precisely, for arbitrary $\mathbf{f}$:

$$N^{-1/k}\, \|\mathbf{f}\|_p \le \|\mathbf{f}\|_\infty \le \|\mathbf{f}\|_p$$

The main idea, the discrete case

Consider the (discrete) optimization problem $\max\{\, f(\mathbf{x}) : \mathbf{x} \in S \,\}$ where f is non-negative on S and the feasible region S is finite.

Approximation properties of ℓ_p norms

$$S = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\} \qquad \text{and} \qquad \mathbf{f} = \begin{pmatrix} f(\mathbf{x}^1) \\ \vdots \\ f(\mathbf{x}^N) \end{pmatrix} \in \mathbf{R}^N.$$

Then

$$\lim_{p\to\infty} \|\mathbf{f}\|_p = \lim_{p\to\infty} \left(f(\mathbf{x}^1)^p + \cdots + f(\mathbf{x}^N)^p \right)^{1/p}$$
$$= \max\{f(\mathbf{x}^1), \ldots, f(\mathbf{x}^N)\} = \|\mathbf{f}\|_\infty$$

More precisely, for arbitrary $\mathbf{f}$:

$$N^{-1/k} \|\mathbf{f}\|_p \le \|\mathbf{f}\|_\infty \le \|\mathbf{f}\|_p$$

The main idea, the discrete case

Consider the (discrete) optimization problem $\max\{ f(\mathbf{x}) : \mathbf{x} \in S \}$ where f is non-negative on S and the feasible region S is finite.

Approximation properties of ℓ_p norms

$$S = \{\mathbf{x}^1, \dots, \mathbf{x}^N\} \qquad \text{and} \qquad \mathbf{f} = \begin{pmatrix} f(\mathbf{x}^1) \\ \vdots \\ f(\mathbf{x}^N) \end{pmatrix} \in \mathbf{R}^N.$$

Then

$$\lim_{p\to\infty} \|\mathbf{f}\|_p = \lim_{p\to\infty} \left(f(\mathbf{x}^1)^p + \dots + f(\mathbf{x}^N)^p \right)^{1/p}$$
$$= \max\{f(\mathbf{x}^1), \dots, f(\mathbf{x}^N)\} = \|\mathbf{f}\|_\infty$$

More precisely, for arbitrary $\mathbf{f}$:

$$N^{-1/k} \|\mathbf{f}\|_p \le \|\mathbf{f}\|_\infty \le \|\mathbf{f}\|_p$$

The main idea, the discrete case

Consider the (discrete) optimization problem $\max\{\, f(\mathbf{x}) : \mathbf{x} \in S \,\}$ where f is non-negative on S and the feasible region S is finite.

Approximation properties of ℓ_p norms

$$S = \{\mathbf{x}^1, \dots, \mathbf{x}^N\} \qquad \text{and} \qquad \mathbf{f} = \begin{pmatrix} f(\mathbf{x}^1) \\ \vdots \\ f(\mathbf{x}^N) \end{pmatrix} \in \mathbf{R}^N.$$

Then

$$\lim_{p\to\infty} \|\mathbf{f}\|_p = \lim_{p\to\infty} \left(f(\mathbf{x}^1)^p + \dots + f(\mathbf{x}^N)^p \right)^{1/p}$$
$$= \max\{f(\mathbf{x}^1), \dots, f(\mathbf{x}^N)\} = \|\mathbf{f}\|_\infty$$

More precisely, for arbitrary $\mathbf{f}$:

$$N^{-1/k} \|\mathbf{f}\|_p \le \|\mathbf{f}\|_\infty \le \|\mathbf{f}\|_p$$

The main idea, the discrete case

Consider the (discrete) optimization problem $\max\{\, f(\mathbf{x}) : \mathbf{x} \in S \,\}$ where f is non-negative on S and the feasible region S is finite.

Approximation properties of ℓ_p norms

$$S = \{\mathbf{x}^1, \dots, \mathbf{x}^N\} \qquad \text{and} \qquad \mathbf{f} = \begin{pmatrix} f(\mathbf{x}^1) \\ \vdots \\ f(\mathbf{x}^N) \end{pmatrix} \in \mathbf{R}^N.$$

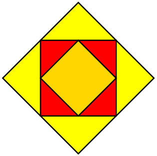
Then

$$\lim_{p\to\infty} \|\mathbf{f}\|_p = \lim_{p\to\infty} \left(f(\mathbf{x}^1)^p + \dots + f(\mathbf{x}^N)^p \right)^{1/p}$$
$$= \max\{f(\mathbf{x}^1), \dots, f(\mathbf{x}^N)\} = \|\mathbf{f}\|_\infty$$

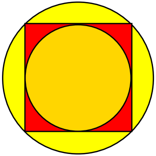
More precisely, for arbitrary $\mathbf{f}$:

$$N^{-1/k} \|\mathbf{f}\|_p \le \|\mathbf{f}\|_\infty \le \|\mathbf{f}\|_p$$

$k = 1$



$k = 2$



The continuous case

Consider the (continuous) optimization problem $\max\{ f(\mathbf{x}) : \mathbf{x} \in S \}$ where the feasible region $S \subseteq \mathbf{R}^n$ is compact and f is continuous and non-negative on S.

Then f is L_p-integrable for $1 \leq p \leq \infty$, and

$$\lim_{p\to\infty} ||f||_p = \lim_{p\to\infty} \Big(\int_S f^p(\mathbf{x})\,\mathrm{d}m(\mathbf{x})\Big)^{1/p} = \operatorname{vrai}\max\{ f(\mathbf{x}) : \mathbf{x} \in S \} = ||f||_\infty$$

If S is well-behaved and the measure $\mathrm{d}m$ is chosen right, $\operatorname{vrai}\max = \max$.

Hard estimates (for finding the right p to give an approximation quality ϵ) come from Hölder's inequality; need some global Lipschitz constant or similar

The continuous case

Consider the (continuous) optimization problem $\max\{ f(\mathbf{x}) : \mathbf{x} \in S \}$ where the feasible region $S \subseteq \mathbf{R}^n$ is compact and f is continuous and non-negative on S.

Then f is L_p-integrable for $1 \leq p \leq \infty$, and

$$\lim_{p\to\infty} ||f||_p = \lim_{p\to\infty} \Big(\int_S f^p(\mathbf{x}) \, \mathrm{d}m(\mathbf{x}) \Big)^{1/p} = \operatorname{vrai\,max}\{ f(\mathbf{x}) : \mathbf{x} \in S \} = ||f||_\infty$$

If S is well-behaved and the measure $\mathrm{d}m$ is chosen right, $\operatorname{vrai\,max} = \max$.

Hard estimates (for finding the right p to give an approximation quality ϵ) come from Hölder's inequality; need some global Lipschitz constant or similar

The continuous case

Consider the (continuous) optimization problem $\max\{ f(\mathbf{x}) : \mathbf{x} \in S \}$ where the feasible region $S \subseteq \mathbf{R}^n$ is compact and f is continuous and non-negative on S.

Then f is L_p-integrable for $1 \leq p \leq \infty$, and

$$\lim_{p\to\infty} \|f\|_p = \lim_{p\to\infty} \Big(\int_S f^p(\mathbf{x})\,\mathrm{d}m(\mathbf{x}) \Big)^{1/p} = \operatorname{vrai\,max}\{ f(\mathbf{x}) : \mathbf{x} \in S \} = \|f\|_\infty$$

If S is well-behaved and the measure $\mathrm{d}m$ is chosen right, $\operatorname{vrai\,max} = \max$.

Hard estimates (for finding the right p to give an approximation quality ϵ) come from Hölder's inequality; need some global Lipschitz constant or similar

The continuous case

Consider the (continuous) optimization problem $\max\{ f(\mathbf{x}) : \mathbf{x} \in S \}$ where the feasible region $S \subseteq \mathbf{R}^n$ is compact and f is continuous and non-negative on S.

Then f is L_p-integrable for $1 \leq p \leq \infty$, and

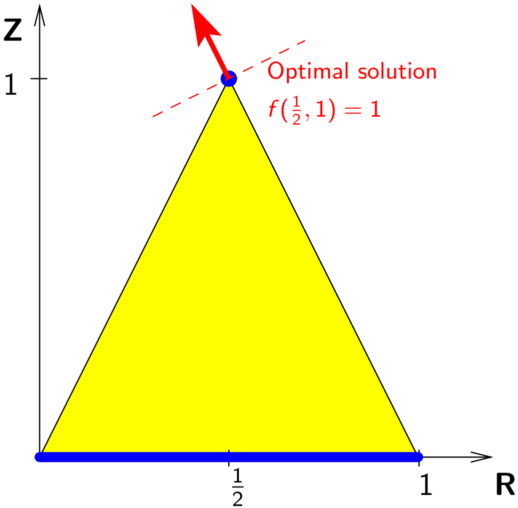
$$\lim_{p\to\infty} ||f||_p = \lim_{p\to\infty} \Big(\int_S f^p(\mathbf{x}) \, \mathrm{d}m(\mathbf{x}) \Big)^{1/p} = \operatorname{vrai\,max}\{ f(\mathbf{x}) : \mathbf{x} \in S \} = ||f||_\infty$$

If S is well-behaved and the measure $\mathrm{d}m$ is chosen right, $\operatorname{vrai\,max} = \max$.

Hard estimates (for finding the right p to give an approximation quality ϵ) come from Hölder's inequality; need some global Lipschitz constant or similar

The mixed-integer case

Here the measure needs to be chosen carefully



Can this really work?!

Let us study the "easiest" case first, integration vs. continuous optimization.

Yes, it works!

In quite general settings, we can solve continuous optimization problems efficiently to arbitrary precision if we have access to an efficient exact integration method.

Does this really work?!

Yes, but it works "too well".

Now hardness of optimization implies that integration is hard.

Can this really work?!

Let us study the "easiest" case first, integration vs. continuous optimization.

Yes, it works!

In quite general settings, we can solve continuous optimization problems efficiently to arbitrary precision if we have access to an efficient exact integration method.

Does this really work?!

Yes, but it works "too well".

Now hardness of optimization implies that integration is hard.

Can this really work?!

Let us study the "easiest" case first, integration vs. continuous optimization.

Yes, it works!

In quite general settings, we can solve continuous optimization problems efficiently to arbitrary precision if we have access to an efficient exact integration method.

Does this really work?!

Yes, but it works "too well".

Now hardness of optimization implies that integration is hard.

Can this really work?!

Let us study the "easiest" case first, integration vs. continuous optimization.

Yes, it works!

In quite general settings, we can solve continuous optimization problems efficiently to arbitrary precision if we have access to an efficient exact integration method.

Does this really work?!

Yes, but it works "too well".

Now hardness of optimization implies that integration is hard.

Outline

A complexity argument against this method

Theorem (Motzkin–Straus)

Let G be a graph with n vertices and clique number $\omega(G)$. Consider the quadratic form

$$Q_G(\mathbf{x}) = \frac{1}{2} \sum_{(i,j)\in E(G)} x_i x_j.$$

Then

$$\|Q_G(\mathbf{x})\|_\infty = \max_{\mathbf{x}\in\Delta} Q_G(\mathbf{x}) = \frac{1}{2}\left(1 - \frac{1}{\omega(G)}\right),$$

where Δ is the standard simplex in $\mathbf{R}^n$.

A complexity argument against this method

Lemma

For $\epsilon > 0$ we have

$$(\|Q_G\|_\infty - \epsilon)(\frac{\epsilon}{4})^{(n-1)/p} \leq \|Q_G\|_p \leq \|Q_G\|_\infty .$$

Lemma

Let G be a graph with n vertices and clique number $\omega(G)$. Let $Q_G(\mathbf{x})$ be the Motzkin–Straus quadratic form. Then for $p \geq 4(e-1)n^3 \ln(32n^2)$, the clique number $\omega(G)$ is equal to $\lceil \frac{1}{1-2\|Q_G\|_p} \rceil$.

Corollary ("I'm sorry Dave, I'm afraid I can't do that.")

The problem of computing the integral of a polynomial over a polytope (even when restricted to powers of quadratic forms over simplices) is NP-hard.

A pragmatic argument against this method

In a numerical method to evaluate the integral $\int_P f^p(\mathbf{x})\,\mathrm{d}\mathbf{x}$,
I need to evaluate $f(\mathbf{x})$ on many points of a mesh anyway,
so why not just pick the maximum?

The End

The End ?!

Integration in one dimension

So why can we compute the (one-dimensional) integral

$$\int_a^b f(x)\,\mathrm{d}x$$

efficiently (and for arbitrary degrees)?

The reason is that there exists an antiderivative $F(x)$ with $F'(x) = f(x)$, and

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

(We can compute the antiderivative symbolically.)

So we have a local formula for the integral– we only need to evaluate the antiderivative on two points, the end points of the interval $[a, b]$.

So there's hope for the optimization-by-integration method if we find formulas like this.

Integration in one dimension

So why can we compute the (one-dimensional) integral

$$\int_a^b f(x)\,\mathrm{d}x$$

efficiently (and for arbitrary degrees)?

The reason is that there exists an antiderivative $F(x)$ with $F'(x) = f(x)$, and

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

(We can compute the antiderivative symbolically.)

So we have a local formula for the integral– we only need to evaluate the antiderivative on two points, the end points of the interval $[a, b]$.

So there's hope for the optimization-by-integration method if we find formulas like this.

Integration in one dimension

So why can we compute the (one-dimensional) integral

$$\int_a^b f(x)\,\mathrm{d}x$$

efficiently (and for arbitrary degrees)?

The reason is that there exists an antiderivative $F(x)$ with $F'(x) = f(x)$, and

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

(We can compute the antiderivative symbolically.)

So we have a local formula for the integral– we only need to evaluate the antiderivative on two points, the end points of the interval $[a, b]$.

So there's hope for the optimization-by-integration method if we find formulas like this.

Integration in one dimension

So why can we compute the (one-dimensional) integral

$$\int_a^b f(x)\,\mathrm{d}x$$

efficiently (and for arbitrary degrees)?

The reason is that there exists an antiderivative $F(x)$ with $F'(x) = f(x)$, and

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

(We can compute the antiderivative symbolically.)

So we have a local formula for the integral– we only need to evaluate the antiderivative on two points, the end points of the interval $[a, b]$.

So there's hope for the optimization-by-integration method if we find formulas like this.

Integration in one dimension

So why can we compute the (one-dimensional) integral

$$\int_a^b f(x)\,\mathrm{d}x$$

efficiently (and for arbitrary degrees)?

The reason is that there exists an antiderivative $F(x)$ with $F'(x) = f(x)$, and

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

(We can compute the antiderivative symbolically.)

So we have a local formula for the integral– we only need to evaluate the antiderivative on two points, the end points of the interval $[a, b]$.

So there's hope for the optimization-by-integration method if we find formulas like this.

Integration in one dimension

So why can we compute the (one-dimensional) integral

$$\int_a^b f(x)\,\mathrm{d}x$$

efficiently (and for arbitrary degrees)?

The reason is that there exists an antiderivative $F(x)$ with $F'(x) = f(x)$, and

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

(We can compute the antiderivative symbolically.)

So we have a local formula for the integral– we only need to evaluate the antiderivative on two points, the end points of the interval $[a, b]$.

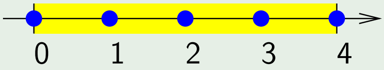
So there's hope for the optimization-by-integration method if we find formulas like this.

Outline

The integer case

A. Barvinok, Exponential integrals and sums over convex polyhedra (in Russian), Funktsional. Anal. i Prilozhen. **26** (1992), 64–66.

Generating functions

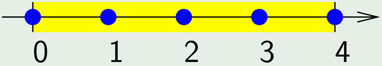


$$g_P(z) = z^0 + z^1 + z^2 + z^3 + z^4$$

The integer case

A. Barvinok, Exponential integrals and sums over convex polyhedra (in Russian), Funktsional. Anal. i Prilozhen. **26** (1992), 64–66.

Generating functions



$$g_P(z) = z^0 + z^1 + z^2 + z^3 + z^4$$
$$= \frac{1 - z^5}{1 - z} \qquad \text{for } z \neq 1$$

The integer case

The Euler differential operator

$$g_P(z) = z^0 + z^1 + z^2 + z^3 + z^4$$

Apply differential operator:

$$\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 2z^2 + 3z^3 + 4z^4$$

Apply differential operator again:

$$\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right)\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 4z^2 + 9z^3 + 16z^4$$

The integer case

The Euler differential operator

$$g_P(z) = z^0 + z^1 + z^2 + z^3 + z^4$$

Apply differential operator:

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 2z^2 + 3z^3 + 4z^4$$

Apply differential operator again:

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 4z^2 + 9z^3 + 16z^4$$

The integer case

The Euler differential operator

$$g_P(z) = z^0 + z^1 + z^2 + z^3 + z^4 = \frac{1}{1-z} - \frac{z^5}{1-z}$$

Apply differential operator:

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 2z^2 + 3z^3 + 4z^4 = \frac{1}{(1-z)^2} - \frac{-4z^5 + 5z^4}{(1-z)^2}$$

Apply differential operator again:

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 4z^2 + 9z^3 + 16z^4 = \frac{z+z^2}{(1-z)^3} - \frac{25z^5 - 39z^6 + 16z^7}{(1-z)^3}$$

The integer case

Using the Euler differential operator

$$\sum_{x=a}^{b} f(x) z^x = \sum_{x=a}^{b} f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) z^x = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \sum_{x=a}^{b} z^x = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \left(\frac{z^a - z^{b+1}}{1-z}\right)$$

Now evaluate at $z = 1$:

$$\sum_{x=a}^{b} f(x) = \sum_{x=a}^{b} f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) z^x \bigg|_{z=1} = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \sum_{x=a}^{b} z^x \bigg|_{z=1} = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \left(\frac{z^a - z^{b+1}}{1-z}\right)\bigg|_{z=1}$$

$$= f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \frac{z^a}{1-z}\bigg|_{z=1} + f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \frac{z^b}{1-z^{-1}}\bigg|_{z=1}$$

The integer case

Using the Euler differential operator

$$\sum_{x=a}^{b} f(x) z^x = \sum_{x=a}^{b} f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) z^x = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \sum_{x=a}^{b} z^x = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \left(\frac{z^a - z^{b+1}}{1-z} \right)$$

Now evaluate at $z = 1$:

$$\sum_{x=a}^{b} f(x) = \sum_{x=a}^{b} f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) z^x \Big|_{z=1} = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \sum_{x=a}^{b} z^x \Big|_{z=1} = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \left(\frac{z^a - z^{b+1}}{1-z} \right) \Big|_{z=1}$$

$$= f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \frac{z^a}{1-z} \Big|_{z=1} + f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \frac{z^b}{1-z^{-1}} \Big|_{z=1}$$

The integer case

Using the Euler differential operator

$$\sum_{x=a}^{b} f(x) z^x = \sum_{x=a}^{b} f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) z^x = f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \sum_{x=a}^{b} z^x = f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \left(\frac{z^a - z^{b+1}}{1 - z}\right)$$

Now evaluate at $z = 1$:

$$\sum_{x=a}^{b} f(x) = \sum_{x=a}^{b} f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) z^x \bigg|_{z=1} = f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \sum_{x=a}^{b} z^x \bigg|_{z=1} = f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \left(\frac{z^a - z^{b+1}}{1 - z}\right) \bigg|_{z=1}$$

$$= f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \frac{z^a}{1 - z} \bigg|_{z=1} + f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \frac{z^b}{1 - z^{-1}} \bigg|_{z=1}$$

The integer case

Using the Euler differential operator

$$\sum_{x=a}^{b} f(x) z^x = \sum_{x=a}^{b} f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) z^x = f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \sum_{x=a}^{b} z^x = f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \left(\frac{z^a - z^{b+1}}{1 - z} \right)$$

Now evaluate at $z = 1$:

$$\sum_{x=a}^{b} f(x) = \sum_{x=a}^{b} f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) z^x \bigg|_{z=1} = f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \sum_{x=a}^{b} z^x \bigg|_{z=1} = f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \left(\frac{z^a - z^{b+1}}{1 - z} \right) \bigg|_{z=1}$$

$$= f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \frac{z^a}{1 - z} \bigg|_{z=1} + f\left(z \frac{\mathrm{d}}{\mathrm{d}z}\right) \frac{z^b}{1 - z^{-1}} \bigg|_{z=1}$$

Using the Euler differential operator

$$\sum_{x=a}^{b} f(x) z^x = \sum_{x=a}^{b} f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) z^x = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \sum_{x=a}^{b} z^x = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \left(\frac{z^a - z^{b+1}}{1 - z}\right)$$

Now evaluate at $z = 1$:

$$\sum_{x=a}^{b} f(x) = \sum_{x=a}^{b} f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) z^x \Big|_{z=1} = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \sum_{x=a}^{b} z^x \Big|_{z=1} = f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \left(\frac{z^a - z^{b+1}}{1 - z}\right)\Big|_{z=}$$

$$= f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \frac{z^a}{1 - z}\Big|_{z=1} + f\Big(z \frac{\mathrm{d}}{\mathrm{d}z}\Big) \frac{z^b}{1 - z^{-1}}\Big|_{z=1}$$

The integer case

Using the Euler differential operator

$$\sum_{x=a}^{b} f(x) z^x = \sum_{x=a}^{b} f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) z^x = f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) \sum_{x=a}^{b} z^x = f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big)\left(\frac{z^a - z^{b+1}}{1-z}\right)$$

Now evaluate at $z = 1$:

$$\sum_{x=a}^{b} f(x) = \sum_{x=a}^{b} f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) z^x \Big|_{z=1} = f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) \sum_{x=a}^{b} z^x \Big|_{z=1} = f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big)\left(\frac{z^a - z^{b+1}}{1-z}\right)\Big|_{z=}$$

$$= f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big)\frac{z^a}{1-z}\Big|_{z=1} + f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big)\frac{z^b}{1-z^{-1}}\Big|_{z=1}$$

The integer case

Using the Euler differential operator

$$\sum_{x=a}^{b} f(x) z^x = \sum_{x=a}^{b} f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) z^x = f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) \sum_{x=a}^{b} z^x = f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) \left(\frac{z^a - z^{b+1}}{1-z}\right)$$

Now evaluate at $z = 1$:

$$\sum_{x=a}^{b} f(x) = \sum_{x=a}^{b} f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) z^x\Big|_{z=1} = f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) \sum_{x=a}^{b} z^x\Big|_{z=1} = f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) \left(\frac{z^a - z^{b+1}}{1-z}\right)\Big|_{z=}$$

$$= f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) \frac{z^a}{1-z}\Big|_{z=1} + f\Big(z\frac{\mathrm{d}}{\mathrm{d}z}\Big) \frac{z^b}{1-z^{-1}}\Big|_{z=1}$$

The integer case

Alexander I. Barvinok, Mathematics of Operations Research **19** (1994), 769–779.

Theorem (Barvinok, 1994)

Let the dimension d be fixed. There is a polynomial-time algorithm for computing a representation of the generating function

$$g_P(z_1, \dots, z_d) = \sum_{(\alpha_1,\dots,\alpha_d) \in P \cap \mathbf{Z}^d} z_1^{\alpha_1} \cdots z_d^{\alpha_d} = \sum_{\alpha \in P \cap \mathbf{Z}^d} \mathbf{z}^{\alpha}$$

of the integer points $P \cap \mathbf{Z}^d$ of a polyhedron $P \subset \mathbf{R}^d$ (given by rational inequalities) in the form of a rational function,

$$g_P(\mathbf{z}) = \sum_{i \in I} \epsilon_i \frac{\mathbf{z}^{\mathbf{a}_i}}{\prod_{j=1}^{d} (1 - \mathbf{z}^{\mathbf{b}_{ij}})}.$$

The integer case

Alexander I. Barvinok, Mathematics of Operations Research **19** (1994), 769–779.

Theorem (Barvinok, 1994)

Let the dimension d be fixed. There is a polynomial-time algorithm for computing a representation of the generating function

$$g_P(z_1, \dots, z_d) = \sum_{(\alpha_1,\dots,\alpha_d)\in P\cap \mathbf{Z}^d} z_1^{\alpha_1}\cdots z_d^{\alpha_d} = \sum_{\boldsymbol\alpha\in P\cap \mathbf{Z}^d} \mathbf{z}^{\boldsymbol\alpha}$$

of the integer points $P \cap \mathbf{Z}^d$ of a polyhedron $P \subset \mathbf{R}^d$ (given by rational inequalities) in the form of a rational function,

$$g_P(\mathbf{z}) = \sum_{i\in I} \epsilon_i \frac{\mathbf{z}^{\mathbf{a}_i}}{\prod_{j=1}^d (1 - \mathbf{z}^{\mathbf{b}_{ij}})}.$$

The integer case

Alexander I. Barvinok, Mathematics of Operations Research **19** (1994), 769–779.

Theorem (Barvinok, 1994)

Let the dimension d be fixed. There is a polynomial-time algorithm for computing a representation of the generating function

$$g_P(z_1,\dots,z_d) = \sum_{(\alpha_1,\dots,\alpha_d)\in P\cap \mathbf{Z}^d} z_1^{\alpha_1}\cdots z_d^{\alpha_d} = \sum_{\boldsymbol{\alpha}\in P\cap\mathbf{Z}^d} \mathbf{z}^{\boldsymbol{\alpha}}$$

of the integer points $P\cap\mathbf{Z}^d$ of a polyhedron $P\subset\mathbf{R}^d$ (given by rational inequalities) in the form of a rational function,

$$g_P(\mathbf{z}) = \sum_{i\in I} \epsilon_i \frac{\mathbf{z}^{\mathbf{a}_i}}{\prod_{j=1}^d (1-\mathbf{z}^{\mathbf{b}_{ij}})}.$$

Differential operators on generating functions

Lemma

An polynomial function $f(x_1, \ldots, x_d) = \sum_{\boldsymbol{\beta}} c_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$ *can be converted to a differential operator*

$$D_f = f\left(z_1 \frac{\partial}{\partial z_1}, \ldots, z_d \frac{\partial}{\partial z_d}\right) = \sum_{\boldsymbol{\beta}} c_{\boldsymbol{\beta}} \left(z_1 \frac{\partial}{\partial z_1}\right)^{\beta_1} \ldots \left(z_d \frac{\partial}{\partial z_d}\right)^{\beta_d}$$

which maps $g(\mathbf{z}) = \sum_{\boldsymbol{\alpha} \in S} \mathbf{z}^{\boldsymbol{\alpha}} \quad \longmapsto \quad (D_f g)(\mathbf{z}) = \sum_{\boldsymbol{\alpha} \in S} f(\boldsymbol{\alpha}) \mathbf{z}^{\boldsymbol{\alpha}}.$

Theorem (De Loera, Hemmecke, K., Weismantel, 2004 / Barvinok, 2004)

Let $g_P(\mathbf{z})$ *be the Barvinok representation of the generating function of the lattice points of* P*. Let* f *be a polynomial in* $\mathbf{Z}[x_1, \ldots, x_d]$ *of maximum total degree* D*.*

We can compute, in time polynomial in D *and the size of the input data, a Barvinok rational function representation* $g_{P,f}(\mathbf{z})$ *for the function* $\sum_{\boldsymbol{\alpha} \in P \cap \mathbf{Z}^d} f(\boldsymbol{\alpha}) \mathbf{z}^{\boldsymbol{\alpha}}$.

Differential operators on generating functions

Lemma

An polynomial function $f(x_1,\dots,x_d) = \sum_{\boldsymbol\beta} c_{\boldsymbol\beta}\mathbf{x}^{\boldsymbol\beta}$ *can be converted to a differential operator*

$$D_f = f\left(z_1\frac{\partial}{\partial z_1},\dots,z_d\frac{\partial}{\partial z_d}\right) = \sum_{\boldsymbol\beta} c_{\boldsymbol\beta}\left(z_1\frac{\partial}{\partial z_1}\right)^{\beta_1}\dots\left(z_d\frac{\partial}{\partial z_d}\right)^{\beta_d}$$

which maps $g(\mathbf{z}) = \sum_{\boldsymbol\alpha\in S}\mathbf{z}^{\boldsymbol\alpha} \quad\longmapsto\quad (D_f g)(\mathbf{z}) = \sum_{\boldsymbol\alpha\in S} f(\boldsymbol\alpha)\mathbf{z}^{\boldsymbol\alpha}$.

Theorem (De Loera, Hemmecke, K., Weismantel, 2004 / Barvinok, 2004)

Let $g_P(\mathbf{z})$ *be the Barvinok representation of the generating function of the lattice points of* P*. Let* f *be a polynomial in* $\mathbf{Z}[x_1,\dots,x_d]$ *of maximum total degree* D*.*

We can compute, in time polynomial in D *and the size of the input data, a Barvinok rational function representation* $g_{P,f}(\mathbf{z})$ *for the function* $\sum_{\boldsymbol\alpha\in P\cap\mathbf{Z}^d} f(\boldsymbol\alpha)\mathbf{z}^{\boldsymbol\alpha}$.

FPTAS for mixed-integer polynomial optimization

We handle the continuous variables by discretization.

Theorem (Fully polynomial-time approximation schemes)

Let the dimension $n = n_1 + n_2$ be fixed. Let an optimization problem

$$\max\{\, f(\mathbf{x},\mathbf{z}) : (\mathbf{x},\mathbf{z}) \in P,\ \mathbf{x} \in \mathbf{R}^{n_1},\ \mathbf{z} \in \mathbf{Z}^{n_2} \,\}$$

of a polynomial function f [with unary-encoded exponents] over the mixed-integer points of a polytope P and an error bound ϵ [in unary encoding] be given.

(a) *There exists an FPTAS for all polynomial functions $f(\mathbf{x},\mathbf{z})$ that are non-negative on the feasible region. That is, there exists a polynomial-time algorithm that, computes a feasible solution $(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) \in P \cap \left(\mathbf{R}^{n_1} \times \mathbf{Z}^{n_2}\right)$ with*

$$\left| f(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) - f(\mathbf{x}_{\max},\mathbf{z}_{\max}) \right| \leq \epsilon f(\mathbf{x}_{\max},\mathbf{z}_{\max}).$$

(b) *There exists a polynomial-time algorithm that, given the above data, computes a feasible solution $(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) \in P \cap \left(\mathbf{R}^{n_1} \times \mathbf{Z}^{n_2}\right)$ with*

$$\left| f(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) - f(\mathbf{x}_{\max},\mathbf{z}_{\max}) \right| \leq \epsilon \left| f(\mathbf{x}_{\max},\mathbf{z}_{\max}) - f(\mathbf{x}_{\min},\mathbf{z}_{\min}) \right|.$$

FPTAS for mixed-integer polynomial optimization

We handle the continuous variables by discretization.

Theorem (Fully polynomial-time approximation schemes)

Let the dimension $n = n_1 + n_2$ be fixed. Let an optimization problem

$$\max\{\, f(\mathbf{x},\mathbf{z}) : (\mathbf{x},\mathbf{z}) \in P,\ \mathbf{x} \in \mathbf{R}^{n_1},\ \mathbf{z} \in \mathbf{Z}^{n_2}\,\}$$

of a polynomial function f [with unary-encoded exponents] over the mixed-integer points of a polytope P and an error bound ϵ [in unary encoding] be given.

(a) *There exists an FPTAS for all polynomial functions $f(\mathbf{x},\mathbf{z})$ that are non-negative on the feasible region. That is, there exists a polynomial-time algorithm that, computes a feasible solution $(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) \in P \cap \left(\mathbf{R}^{n_1} \times \mathbf{Z}^{n_2}\right)$ with*

$$\left|f(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) - f(\mathbf{x}_{\max},\mathbf{z}_{\max})\right| \le \epsilon f(\mathbf{x}_{\max},\mathbf{z}_{\max}).$$

(b) *There exists a polynomial-time algorithm that, given the above data, computes a feasible solution $(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) \in P \cap \left(\mathbf{R}^{n_1} \times \mathbf{Z}^{n_2}\right)$ with*

$$\left|f(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) - f(\mathbf{x}_{\max},\mathbf{z}_{\max})\right| \le \epsilon\left|f(\mathbf{x}_{\max},\mathbf{z}_{\max}) - f(\mathbf{x}_{\min},\mathbf{z}_{\min})\right|.$$

FPTAS for mixed-integer polynomial optimization

We handle the continuous variables by discretization.

Theorem (Fully polynomial-time approximation schemes)

Let the dimension $n = n_1 + n_2$ be fixed. Let an optimization problem

$$\max\{\, f(\mathbf{x},\mathbf{z}) : (\mathbf{x},\mathbf{z}) \in P,\ \mathbf{x} \in \mathbf{R}^{n_1},\ \mathbf{z} \in \mathbf{Z}^{n_2} \,\}$$

of a polynomial function f [with unary-encoded exponents] over the mixed-integer points of a polytope P and an error bound ϵ [in unary encoding] be given.

(a) *There exists an FPTAS for all polynomial functions $f(\mathbf{x},\mathbf{z})$ that are non-negative on the feasible region. That is, there exists a polynomial-time algorithm that, computes a feasible solution $(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) \in P \cap \left(\mathbf{R}^{n_1} \times \mathbf{Z}^{n_2}\right)$ with*

$$\left|f(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) - f(\mathbf{x}_{\max},\mathbf{z}_{\max})\right| \leq \epsilon f(\mathbf{x}_{\max},\mathbf{z}_{\max}).$$

(b) *There exists a polynomial-time algorithm that, given the above data, computes a feasible solution $(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) \in P \cap \left(\mathbf{R}^{n_1} \times \mathbf{Z}^{n_2}\right)$ with*

$$\left|f(\mathbf{x}_\epsilon,\mathbf{z}_\epsilon) - f(\mathbf{x}_{\max},\mathbf{z}_{\max})\right| \leq \epsilon \left|f(\mathbf{x}_{\max},\mathbf{z}_{\max}) - f(\mathbf{x}_{\min},\mathbf{z}_{\min})\right|.$$

Implementation

Barvinok's algorithm for computing

$$g_P(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} \mathbf{z}^{\mathbf{x}}$$

implemented in:

- LattE (De Loera et al., 2002–2003),
- `barvinok` (Verdoolaege, 2005–),
- LattE macchiato (K., 2006–).

It is the state-of-the-art method for counting integer points in general polytopes.

Symbolic method using the Euler operator:

$$g_{P,f}(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} f(\mathbf{x}) \mathbf{z}^{\mathbf{x}} = D_f \, g_P(\mathbf{z})$$

creates huge symbolic expressions (Maple), very slow

Implementation

Barvinok's algorithm for computing

$$g_P(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} \mathbf{z}^{\mathbf{x}}$$

implemented in:

- LattE (De Loera et al., 2002–2003),
- barvinok (Verdoolaege, 2005–),
- LattE macchiato (K., 2006–).

It is the state-of-the-art method for counting integer points in general polytopes.

Symbolic method using the Euler operator:

$$g_{P,f}(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} f(\mathbf{x})\mathbf{z}^{\mathbf{x}} = D_f\, g_P(\mathbf{z})$$

creates huge symbolic expressions (Maple), very slow

Implementation

Barvinok's algorithm for computing

$$g_P(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} \mathbf{z}^{\mathbf{x}}$$

implemented in:

- LattE (De Loera et al., 2002–2003),
- `barvinok` (Verdoolaege, 2005–),
- LattE macchiato (K., 2006–).

It is the state-of-the-art method for counting integer points in general polytopes.

Symbolic method using the Euler operator:

$$g_{P,f}(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} f(\mathbf{x}) \mathbf{z}^{\mathbf{x}} = D_f g_P(\mathbf{z})$$

creates huge symbolic expressions (Maple), very slow

Implementation

Barvinok's algorithm for computing

$$g_P(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} \mathbf{z}^{\mathbf{x}}$$

implemented in:

- LattE (De Loera et al., 2002–2003),
- `barvinok` (Verdoolaege, 2005–),
- LattE macchiato (K., 2006–).

It is the state-of-the-art method for counting integer points in general polytopes.

Symbolic method using the Euler operator:

$$g_{P,f}(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} f(\mathbf{x}) \mathbf{z}^{\mathbf{x}} = D_f g_P(\mathbf{z})$$

creates huge symbolic expressions (Maple), very slow

Image source: Wikipedia

Implementation

Barvinok's algorithm for computing

$$g_P(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} \mathbf{z}^{\mathbf{x}}$$

implemented in:

- LattE (De Loera et al., 2002–2003),
- `barvinok` (Verdoolaege, 2005–),
- LattE macchiato (K., 2006–).

It is the state-of-the-art method for counting integer points in general polytopes.

Symbolic method using the Euler operator:

$$g_{P,f}(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} f(\mathbf{x})\mathbf{z}^{\mathbf{x}} = D_f g_P(\mathbf{z})$$

creates huge symbolic expressions (Maple), very slow

Image source: Wikipedia

Implementation

Barvinok's algorithm for computing

$$g_P(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} \mathbf{z}^{\mathbf{x}}$$

implemented in:

- LattE (De Loera et al., 2002–2003),
- `barvinok` (Verdoolaege, 2005–),
- LattE macchiato (K., 2006–).

It is the state-of-the-art method for counting integer points in general polytopes.

Symbolic method using the Euler operator:

$$g_{P,f}(\mathbf{z}) = \sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} f(\mathbf{x}) \mathbf{z}^{\mathbf{x}} = D_f \, g_P(\mathbf{z})$$

creates huge symbolic expressions (Maple), very slow

Image source: Wikipedia

The Barvinok–Berline–Vergne algorithm

N. Berline and M. Vergne, Contemporary Mathematics **452** (2008), 15–33.

Berline–Vergne's local Euler–Maclaurin formula

$$\sum_{\mathbf{x}\in P\cap\mathbf{Z}^d} f(\mathbf{x}) = \sum_{F \text{ face of } P} \int_F D(P,F) f(\mathbf{x})\,\mathrm{d}m_F(\mathbf{x})$$

where

- $D(P,f)$ is a differential operator with constant coefficients (of infinite order)
- $\mathrm{d}m_F$ is the integral Lebesgue measure on $\langle F\rangle$

So now we have two reasons to study efficient integration over polytopes

- It appears directly in non-linear continuous optimization
- It appears, via Euler–Maclaurin formulas, also in non-linear integer optimization

The Barvinok–Berline–Vergne algorithm

N. Berline and M. Vergne, Contemporary Mathematics **452** (2008), 15–33.

Berline–Vergne's local Euler–Maclaurin formula

$$\sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} f(\mathbf{x}) = \sum_{F \text{ face of } P} \int_F D(P, F) f(\mathbf{x}) \, \mathrm{d}m_F(\mathbf{x})$$

where

- $D(P, f)$ is a differential operator with constant coefficients (of infinite order)
- $\mathrm{d}m_F$ is the integral Lebesgue measure on $\langle F \rangle$

So now we have two reasons to study efficient integration over polytopes

- It appears directly in non-linear continuous optimization
- It appears, via Euler–Maclaurin formulas, also in non-linear integer optimization

The Barvinok–Berline–Vergne algorithm

N. Berline and M. Vergne, Contemporary Mathematics **452** (2008), 15–33.

Berline–Vergne's local Euler–Maclaurin formula

$$\sum_{\mathbf{x}\in P\cap\mathbf{Z}^d} f(\mathbf{x}) = \sum_{F \text{ face of } P} \int_F D(P,F) f(\mathbf{x})\, \mathrm{d}m_F(\mathbf{x})$$

where

- $D(P,f)$ is a differential operator with constant coefficients (of infinite order)
- $\mathrm{d}m_F$ is the integral Lebesgue measure on $\langle F\rangle$

So now we have two reasons to study efficient integration over polytopes

- It appears directly in non-linear continuous optimization
- It appears, via Euler–Maclaurin formulas, also in non-linear integer optimization

The Barvinok–Berline–Vergne algorithm

N. Berline and M. Vergne, Contemporary Mathematics **452** (2008), 15–33.

Berline–Vergne's local Euler–Maclaurin formula

$$\sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} f(\mathbf{x}) = \sum_{F \text{ face of } P} \int_F D(P,F) f(\mathbf{x}) \, \mathrm{d}m_F(\mathbf{x})$$

where

- $D(P, f)$ is a differential operator with constant coefficients (of infinite order)
- $\mathrm{d}m_F$ is the integral Lebesgue measure on $\langle F \rangle$

So now we have two reasons to study efficient integration over polytopes

- It appears directly in non-linear continuous optimization
- It appears, via Euler–Maclaurin formulas, also in non-linear integer optimization

The Barvinok–Berline–Vergne algorithm

N. Berline and M. Vergne, Contemporary Mathematics **452** (2008), 15–33.

Berline–Vergne's local Euler–Maclaurin formula

$$\sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} f(\mathbf{x}) = \sum_{F \text{ face of } P} \int_F D(P,F) f(\mathbf{x}) \, \mathrm{d}m_F(\mathbf{x})$$

where

- $D(P,f)$ is a differential operator with constant coefficients (of infinite order)
- $\mathrm{d}m_F$ is the integral Lebesgue measure on $\langle F \rangle$

So now we have two reasons to study efficient integration over polytopes

- It appears directly in non-linear continuous optimization
- It appears, via Euler–Maclaurin formulas, also in non-linear integer optimization

The Barvinok–Berline–Vergne algorithm

N. Berline and M. Vergne, Contemporary Mathematics **452** (2008), 15–33.

Berline–Vergne's local Euler–Maclaurin formula

$$\sum_{\mathbf{x} \in P \cap \mathbf{Z}^d} f(\mathbf{x}) = \sum_{F \text{ face of } P} \int_F D(P, F) f(\mathbf{x}) \, \mathrm{d}m_F(\mathbf{x})$$

where

- $D(P, f)$ is a differential operator with constant coefficients (of infinite order)
- $\mathrm{d}m_F$ is the integral Lebesgue measure on $\langle F \rangle$

So now we have two reasons to study efficient integration over polytopes

- It appears directly in non-linear continuous optimization
- It appears, via Euler–Maclaurin formulas, also in non-linear integer optimization

Outline

Integration in one dimension (again)

(Don't read this slide – you've seen it before.)
So why can we compute the (one-dimensional) integral

$$\int_a^b f(x)\,\mathrm{d}x$$

efficiently (and for arbitrary degrees)?
The reason is that there exists an antiderivative $F(x)$ with $F'(x) = f(x)$, and

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

(We can compute the antiderivative symbolically.)
So we have a local formula for the integral– we only need to evaluate the antiderivative on two points, the end points of the interval $[a, b]$.
So there's hope for the optimization-by-integration method if we find formulas like this.

Integration in one dimension (again)

(Don't read this slide – you've seen it before.)
So why can we compute the (one-dimensional) integral

$$\int_a^b f(x)\,\mathrm{d}x$$

efficiently (and for arbitrary degrees)?
The reason is that there exists an antiderivative $F(x)$ with $F'(x) = f(x)$, and

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

(We can compute the antiderivative symbolically.)
So we have a local formula for the integral– we only need to evaluate the antiderivative on two points, the end points of the interval $[a, b]$.
So there's hope for the optimization-by-integration method if we find formulas like this.

Integration in one dimension (again)

(Don't read this slide – you've seen it before.)
So why can we compute the (one-dimensional) integral

$$\int_a^b f(x)\,\mathrm{d}x$$

efficiently (and for arbitrary degrees)?
The reason is that there exists an antiderivative $F(x)$ with $F'(x) = f(x)$, and

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

(We can compute the antiderivative symbolically.)
So we have a local formula for the integral– we only need to evaluate the antiderivative on two points, the end points of the interval $[a, b]$.
So there's hope for the optimization-by-integration method if we find formulas like this.

Integration over a box

Now let $\square^d = [a_1, b_1] \times \cdots \times [a_d, b_d]$ be a box.
Let $f(\mathbf{x}) = \mathbf{x}^{\mathbf{m}} = x_1^{m_1} \ldots x_d^{m_d}$ be a monomial function.
Then

$$\begin{aligned}\int_{\square^d} f(\mathbf{x})\,\mathrm{d}\mathbf{x} &= \int_{\square^d} x_1^{m_1} \ldots x_d^{m_d}\,\mathrm{d}\mathbf{x} \\ &= \int_{a_1}^{b_1} \ldots \int_{a_d}^{b_d} x_1^{m_1} \ldots x_d^{m_d}\,\mathrm{d}x_d \ldots \mathrm{d}x_1 \\ &= \left(\int_{a_1}^{b_1} x_1^{m_1}\,\mathrm{d}x_1\right) \ldots \left(\int_{a_d}^{b_d} x_d^{m_d}\,\mathrm{d}x_d\right)\end{aligned}$$

(one-dimensional integrals)
This works only because $\mathbf{x}^m$ is factorable and $\square^d$ is a direct product of segments.

Integration over a box

Now let $\square^d = [a_1, b_1] \times \cdots \times [a_d, b_d]$ be a box.
Let $f(\mathbf{x}) = \mathbf{x}^{\mathbf{m}} = x_1^{m_1} \dots x_d^{m_d}$ be a monomial function.
Then

$$\begin{aligned}\int_{\square^d} f(\mathbf{x})\,\mathrm{d}\mathbf{x} &= \int_{\square^d} x_1^{m_1} \dots x_d^{m_d}\,\mathrm{d}\mathbf{x} \\ &= \int_{a_1}^{b_1} \dots \int_{a_d}^{b_d} x_1^{m_1} \dots x_d^{m_d}\,\mathrm{d}x_d \dots \mathrm{d}x_1 \\ &= \left(\int_{a_1}^{b_1} x_1^{m_1}\,\mathrm{d}x_1\right) \dots \left(\int_{a_d}^{b_d} x_d^{m_d}\,\mathrm{d}x_d\right)\end{aligned}$$

(one-dimensional integrals)
This works only because $\mathbf{x}^m$ is factorable and $\square^d$ is a direct product of segments.

Integration over a box

Now let $\square^d = [a_1, b_1] \times \cdots \times [a_d, b_d]$ be a box.
Let $f(\mathbf{x}) = \mathbf{x}^{\mathbf{m}} = x_1^{m_1} \dots x_d^{m_d}$ be a monomial function.
Then

$$\begin{aligned} \int_{\square^d} f(\mathbf{x}) \, \mathrm{d}\mathbf{x} &= \int_{\square^d} x_1^{m_1} \dots x_d^{m_d} \, \mathrm{d}\mathbf{x} \\ &= \int_{a_1}^{b_1} \dots \int_{a_d}^{b_d} x_1^{m_1} \dots x_d^{m_d} \, \mathrm{d}x_d \dots \mathrm{d}x_1 \\ &= \left(\int_{a_1}^{b_1} x_1^{m_1} \, \mathrm{d}x_1 \right) \dots \left(\int_{a_d}^{b_d} x_d^{m_d} \, \mathrm{d}x_d \right) \end{aligned}$$

(one-dimensional integrals)

This works only because $\mathbf{x}^m$ is factorable and $\square^d$ is a direct product of segments.

Integration over a box

Now let $\square^d = [a_1, b_1] \times \cdots \times [a_d, b_d]$ be a box.
Let $f(\mathbf{x}) = \mathbf{x}^{\mathbf{m}} = x_1^{m_1} \dots x_d^{m_d}$ be a monomial function.
Then

$$\begin{aligned}\int_{\square^d} f(\mathbf{x})\,\mathrm{d}\mathbf{x} &= \int_{\square^d} x_1^{m_1} \dots x_d^{m_d}\,\mathrm{d}\mathbf{x} \\ &= \int_{a_1}^{b_1} \dots \int_{a_d}^{b_d} x_1^{m_1} \dots x_d^{m_d}\,\mathrm{d}x_d \dots \mathrm{d}x_1 \\ &= \left(\int_{a_1}^{b_1} x_1^{m_1}\,\mathrm{d}x_1\right) \dots \left(\int_{a_d}^{b_d} x_d^{m_d}\,\mathrm{d}x_d\right)\end{aligned}$$

(one-dimensional integrals)
This works only because $\mathbf{x}^m$ is factorable and $\square^d$ is a direct product of segments.

Integration over a polytope

Let $P = \{ A\mathbf{x} \leq \mathbf{b} \} \subseteq \mathbf{R}^d$ be a full-dimensional polytope. Compute:

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{a_1}^{b_1} \int_{a_2(x_1)}^{b_2(x_1)} \int_{a_3(x_1,x_2)}^{b_3(x_1,x_2)} \dots \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{x})\,\mathrm{d}x_d \dots \mathrm{d}x_1.$$

Again, it's just iterated one-dimensional integration.
But how do we handle the parametric limits of integration?

Integration over a polytope

Let $P = \{ A\mathbf{x} \leq \mathbf{b} \} \subseteq \mathbf{R}^d$ be a full-dimensional polytope. Compute:

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{a_1}^{b_1} \int_{a_2(x_1)}^{b_2(x_1)} \int_{a_3(x_1,x_2)}^{b_3(x_1,x_2)} \dots \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{x})\,\mathrm{d}x_d \dots \mathrm{d}x_1.$$

Again, it's just iterated one-dimensional integration.

But how do we handle the parametric limits of integration?

Integration over a polytope

Let $P = \{ A\mathbf{x} \leq \mathbf{b} \} \subseteq \mathbf{R}^d$ be a full-dimensional polytope. Compute:

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{a_1}^{b_1} \int_{a_2(x_1)}^{b_2(x_1)} \int_{a_3(x_1,x_2)}^{b_3(x_1,x_2)} \dots \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{x})\,\mathrm{d}x_d \dots \mathrm{d}x_1.$$

Again, it's just iterated one-dimensional integration.

But how do we handle the parametric limits of integration?

Integration over a polytope

Let $P = \{ A\mathbf{x} \leq \mathbf{b} \} \subseteq \mathbf{R}^d$ be a full-dimensional polytope. Compute:

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{a_1}^{b_1} \int_{a_2(x_1)}^{b_2(x_1)} \int_{a_3(x_1,x_2)}^{b_3(x_1,x_2)} \dots \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{x})\,\mathrm{d}x_d \dots \mathrm{d}x_1.$$

Again, it's just iterated one-dimensional integration.
But how do we handle the parametric limits of integration?

Integration over a polytope by elimination

M. Schechter, American Mathematical Monthly **105** (1998), 246–251.

We use Fourier–Motzkin elimination of the d-th variable:

$$P = \left\{ \begin{pmatrix} C^0 & \mathbf{0} \\ C^+ & \mathbf{c}^+ \\ C^- & \mathbf{c}^- \end{pmatrix} \mathbf{x} \le \begin{pmatrix} \mathbf{d}^0 \\ \mathbf{d}^+ \\ \mathbf{d}^- \end{pmatrix} \right\}, \qquad P_d := \operatorname{proj}_{x_1,\dots,x_{d-1}} P = \{\, C^0 \mathbf{y} \le \mathbf{d}^0 \,\}$$

We can write:

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{P_d} \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{y}, x_d)\,\mathrm{d}x_d\,\mathrm{d}\mathbf{y}.$$

Here

$$b_d(x_1,\dots,x_{d-1}) = \min_k \frac{1}{c_k^+}\big(d_k^+ - C_k^+(x_1,\dots,x_{d-1})\big)$$

$$a_d(x_1,\dots,x_{d-1}) = \max_k \frac{1}{c_k^-}\big(d_k^- - C_k^-(x_1,\dots,x_{d-1})\big)$$

(Piecewise linear functions.)

Integration over a polytope by elimination

M. Schechter, American Mathematical Monthly **105** (1998), 246–251.

We use Fourier–Motzkin elimination of the d-th variable:

$$P = \left\{ \begin{pmatrix} C^0 & \mathbf{0} \\ C^+ & \mathbf{c}^+ \\ C^- & \mathbf{c}^- \end{pmatrix} \mathbf{x} \leq \begin{pmatrix} \mathbf{d}^0 \\ \mathbf{d}^+ \\ \mathbf{d}^- \end{pmatrix} \right\}, \qquad P_d := \operatorname{proj}_{x_1,\dots,x_{d-1}} P = \{\, C^0\mathbf{y} \leq \mathbf{d}^0 \,\}$$

We can write:

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{P_d} \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{y}, x_d)\,\mathrm{d}x_d\,\mathrm{d}\mathbf{y}.$$

Here

$$b_d(x_1,\dots,x_{d-1}) = \min_k \frac{1}{c_k^+}\left(d_k^+ - C_k^+(x_1,\dots,x_{d-1})\right)$$

$$a_d(x_1,\dots,x_{d-1}) = \max_k \frac{1}{c_k^-}\left(d_k^- - C_k^-(x_1,\dots,x_{d-1})\right)$$

(Piecewise linear functions.)

Integration over a polytope by elimination

M. Schechter, American Mathematical Monthly **105** (1998), 246–251.

We use Fourier–Motzkin elimination of the d-th variable:

$$P = \left\{ \begin{pmatrix} C^0 & \mathbf{0} \\ C^+ & \mathbf{c}^+ \\ C^- & \mathbf{c}^- \end{pmatrix} \mathbf{x} \le \begin{pmatrix} \mathbf{d}^0 \\ \mathbf{d}^+ \\ \mathbf{d}^- \end{pmatrix} \right\}, \qquad P_d := \operatorname{proj}_{x_1,\dots,x_{d-1}} P = \{ C^0 \mathbf{y} \le \mathbf{d}^0 \}$$

We can write:

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{P_d} \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{y}, x_d)\,\mathrm{d}x_d\,\mathrm{d}\mathbf{y}.$$

Here

$$b_d(x_1,\dots,x_{d-1}) = \min_k \frac{1}{c_k^+}\left(d_k^+ - C_k^+(x_1,\dots,x_{d-1})\right)$$

$$a_d(x_1,\dots,x_{d-1}) = \max_k \frac{1}{c_k^-}\left(d_k^- - C_k^-(x_1,\dots,x_{d-1})\right)$$

(Piecewise linear functions.)

Integration over a polytope by elimination

M. Schechter, American Mathematical Monthly **105** (1998), 246–251.

We use Fourier–Motzkin elimination of the d-th variable:

$$P = \left\{ \begin{pmatrix} C^0 & \mathbf{0} \\ C^+ & \mathbf{c}^+ \\ C^- & \mathbf{c}^- \end{pmatrix} \mathbf{x} \le \begin{pmatrix} \mathbf{d}^0 \\ \mathbf{d}^+ \\ \mathbf{d}^- \end{pmatrix} \right\}, \qquad P_d := \operatorname{proj}_{x_1,\dots,x_{d-1}} P = \{\, C^0\mathbf{y} \le \mathbf{d}^0 \,\}$$

We can write:

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{P_d} \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{y}, x_d)\,\mathrm{d}x_d\,\mathrm{d}\mathbf{y}.$$

Here

$$b_d(x_1,\dots,x_{d-1}) = \min_k \frac{1}{c_k^+}\left(d_k^+ - C_k^+(x_1,\dots,x_{d-1})\right)$$

$$a_d(x_1,\dots,x_{d-1}) = \max_k \frac{1}{c_k^-}\left(d_k^- - C_k^-(x_1,\dots,x_{d-1})\right)$$

(Piecewise linear functions.)

Integration over a polytope by elimination

M. Schechter, American Mathematical Monthly **105** (1998), 246–251.

Compute the polyhedral pieces $P_{d,i}$ where the functions a_d, b_d are affine-linear.

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \sum_i \int_{P_{d,i}} \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{y}, x_d)\,\mathrm{d}x_d\,\mathrm{d}\mathbf{y}.$$

Then iterate for the remaining variables.

If the dimension d is allowed to grow, iterated Fourier–Motzkin elimination takes exponential time. . . even if restricted to simplices!

Integration over a polytope by elimination

M. Schechter, American Mathematical Monthly **105** (1998), 246–251.

Compute the polyhedral pieces $P_{d,i}$ where the functions a_d, b_d are affine-linear.

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \sum_i \int_{P_{d,i}} \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{y}, x_d)\,\mathrm{d}x_d\,\mathrm{d}\mathbf{y}.$$

Then iterate for the remaining variables.

If the dimension d is allowed to grow, iterated Fourier–Motzkin elimination takes exponential time…even if restricted to simplices!

Integration over a polytope by elimination

M. Schechter, American Mathematical Monthly **105** (1998), 246–251.

Compute the polyhedral pieces $P_{d,i}$ where the functions a_d, b_d are affine-linear.

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \sum_i \int_{P_{d,i}} \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{y}, x_d)\,\mathrm{d}x_d\,\mathrm{d}\mathbf{y}.$$

Then iterate for the remaining variables.

If the dimension d is allowed to grow, iterated Fourier–Motzkin elimination takes exponential time. . . even if restricted to simplices!

Integration over a polytope by elimination

M. Schechter, American Mathematical Monthly **105** (1998), 246–251.

Compute the polyhedral pieces $P_{d,i}$ where the functions a_d, b_d are affine-linear.

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \sum_i \int_{P_{d,i}} \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{y}, x_d)\,\mathrm{d}x_d\,\mathrm{d}\mathbf{y}.$$

Then iterate for the remaining variables.

If the dimension d is allowed to grow, iterated Fourier–Motzkin elimination takes exponential time. . . even if restricted to simplices!

Integration over a polytope by elimination

M. Schechter, American Mathematical Monthly **105** (1998), 246–251.

Compute the polyhedral pieces $P_{d,i}$ where the functions a_d, b_d are affine-linear.

$$\int_P f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \sum_i \int_{P_{d,i}} \int_{a_d(x_1,\dots,x_{d-1})}^{b_d(x_1,\dots,x_{d-1})} f(\mathbf{y}, x_d)\,\mathrm{d}x_d\,\mathrm{d}\mathbf{y}.$$

Then iterate for the remaining variables.

If the dimension d is allowed to grow, iterated Fourier–Motzkin elimination takes exponential time. . . even if restricted to simplices!

Lasserre–Avrachenkov's "multidim. version of $\int_a^b x^p \,\mathrm{d}x$"

J. B. Lasserre and K. Avrachenkov, American Mathematical Monthly **108** (2001), 151–154

Let $\mathbf{s}_1, \ldots, \mathbf{s}_{d+1}$ be the vertices of a d-dimensional simplex Δ.

Theorem (Lasserre–Avrachenkov, 2001)

For H a symmetric multilinear form defined on $(\mathbf{R}^d)^M$:

$$\int_\Delta H(\mathbf{x}, \mathbf{x}, \ldots, \mathbf{x}) d\mathbf{x} = \frac{\mathrm{vol}(\Delta)}{\binom{M+d}{M}} \sum_{1 \le i_1 \le i_2 \le \cdots i_M \le d+1} H(\mathbf{s}_{i_1}, \mathbf{s}_{i_2}, \ldots, \mathbf{s}_{i_M}).$$

Polarization formula: $H_f(\mathbf{x}_1, \ldots, \mathbf{x}_M) = \frac{1}{2^M M!} \sum_{\epsilon \in \{\pm 1\}^M} \epsilon_1 \epsilon_2 \cdots \epsilon_M f\left(\sum_{i=1}^M \epsilon_i \mathbf{x}_i\right)$,

Corollary (for a homogeneous polynomial f of degree M in d variables)

$$\int_\Delta f(\mathbf{y}) \,\mathrm{d}\mathbf{y} = \frac{\mathrm{vol}(\Delta)}{2^M M! \binom{M+d}{M}} \sum_{1 \le i_1 \le i_2 \le \cdots \le i_M \le d+1} \sum_{\epsilon \in \{\pm 1\}^M} \epsilon_1 \epsilon_2 \cdots \epsilon_M f\left(\sum_{k=1}^M \epsilon_k \mathbf{s}_{i_k}\right).$$

Lasserre–Avrachenkov's formula

Table: Integration of a random monomial of prescribed degree using polarization

	Degree M											
n	1	2	5	10	20	30	40	50	100	200	300	1000
2	**0**	**0**	**0**.1	**15**								
3	**0**	**0**	**0**.3	**67**								
4	**0**	**0**	**0**.7	**250**								
5	**0**	**0**	**1**.5									
8	**0**	**0**	**8**.1									
10	**0**	**0**.1	**20**									
15	**0**	**0**.2	**120**									
20	**0**.1	**0**.3	**450**									

Running time in CPU seconds with Maple 10 on Sun Fire V440 machines with UltraSPARC-IIIi processors running at 1.6 GHz. Time limit 600 seconds per example.

Outline

Brion's formula

M. Brion, Ann. Sci. École Norm. Sup. **21** (1988), 653–663.

Theorem (Brion)

Let Δ be the simplex that is the convex hull of $(d+1)$ affinely independent vertices $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{d+1}$ in $\mathbf{R}^n$.
Let ℓ be a linear form which is regular w.r.t. Δ, i.e.,

$$\langle \ell, \mathbf{s}_i \rangle \neq \langle \ell, \mathbf{s}_j \rangle \quad \text{for } i \neq j$$

Then:

$$\int_\Delta e^\ell \, \mathrm{d}m = d! \operatorname{vol}(\Delta, \mathrm{d}m) \sum_{i=1}^{d+1} \frac{e^{\langle \ell, \mathbf{s}_i \rangle}}{\prod_{j \neq i} \langle \ell, \mathbf{s}_i - \mathbf{s}_j \rangle}.$$

By expanding the exponential as a Taylor series:

Corollary

$$\int_\Delta \ell^M \, \mathrm{d}m = d! \operatorname{vol}(\Delta, \mathrm{d}m) \frac{M!}{(M+d)!} \Big(\sum_{i=1}^{d+1} \frac{\langle \ell, \mathbf{s}_i \rangle^{M+d}}{\prod_{j \neq i} \langle \ell, \mathbf{s}_i - \mathbf{s}_j \rangle} \Big).$$

Complexity situation

The problem of computing the integral of a polynomial over a polytope (even when restricted to powers of quadratic forms over simplices) is NP-hard.

The problem of computing the integral of a power of a (regular) linear form over a simplex is polynomial-time solvable.

Complexity situation

The problem of computing the integral of a polynomial over a polytope (even when restricted to powers of quadratic forms over simplices) is NP-hard.

The problem of computing the integral of a power of a (regular) linear form over a simplex is polynomial-time solvable.

Complexity situation

The problem of computing the integral of a polynomial over a polytope (even when restricted to powers of quadratic forms over simplices) is NP-hard.

The problem of computing the integral of a power of a (regular) linear form over a simplex is polynomial-time solvable.

But we want to be able to handle arbitrary polynomials!

Outline

A simple construction

Lemma

We can represent any monomial as a sum of powers of linear forms:

$$\begin{aligned}\mathbf{x}^{\mathbf{M}} &= x_1^{M_1} x_2^{M_2} \cdots x_n^{M_n} \\ &= \frac{1}{|\mathbf{M}|!} \sum_{0 \le p_i \le M_i} (-1)^{|\mathbf{M}|-(p_1+\cdots+p_n)} \binom{M_1}{p_1} \cdots \binom{M_n}{p_n} (p_1 x_1 + \cdots + p_n x_n)^{|\mathbf{M}|},\end{aligned}$$

where $|\mathbf{M}| = M_1 + \cdots + M_n \le M$.

These are at most 2^M summands.

Integration by decomposition into powers of linear forms

	Degree M											
n	1	2	5	10	20	30	40	50	100	200	300	1000
1	0	0	0	0	0	0	0	0	0	0	0.1	0.9
2	0	0	0	0.1	0.3	0.5	1.0	1.5	6.3	38	100	
3	0	0	0	0.2	1.4	3.5	9.3	16	180			
4	0	0	0.1	0.5	3.9	17	47	140				
5	0	0	0.1	0.8	12	54						
8	0	0.1	0.4	3.1	99							
10	0	0.1	0.6	7.5								
20	0.2	0.5	4	71								
30	0.5	1.4	12	260								
50	1.5	4.6	43									

The polynomial Waring problem

J. Alexander and A. Hirschowitz, J. Algebraic Geom. **4** (1995), 201–222.

How many powers of linear forms do we really need?

The polynomial Waring problem

What is the smallest integer $r(M, n)$ such that a generic homogeneous polynomial $f(x_1, \dots, x_n)$ of degree M in n variables is expressible as the sum of $r(M, n)$ M-th powers of linear forms?

Theorem (Alexander–Hirschowitz, 1995)

A generic homogeneous polynomial of degree M in n variables is expressible as the sum of

$$r(M, n) = \left\lceil \frac{\binom{n+M-1}{M}}{n} \right\rceil$$

M-th powers of linear forms, with the exception of the cases $r(3, 5) = 8$, $r(4, 3) = 6$, $r(4, 4) = 10$, $r(4, 5) = 15$, and $M = 2$, where $r(2, n) = n$.

(Much better bounds than those of our simple construction.)

The polynomial Waring problem

J. Alexander and A. Hirschowitz, J. Algebraic Geom. **4** (1995), 201–222.

How many powers of linear forms do we really need?

The polynomial Waring problem

What is the smallest integer $r(M, n)$ such that a generic homogeneous polynomial $f(x_1, \ldots, x_n)$ of degree M in n variables is expressible as the sum of $r(M, n)$ M-th powers of linear forms?

Theorem (Alexander–Hirschowitz, 1995)

A generic homogeneous polynomial of degree M in n variables is expressible as the sum of

$$r(M, n) = \left\lceil \frac{\binom{n+M-1}{M}}{n} \right\rceil$$

M-th powers of linear forms, with the exception of the cases $r(3, 5) = 8$, $r(4, 3) = 6$, $r(4, 4) = 10$, $r(4, 5) = 15$, and $M = 2$, where $r(2, n) = n$.

(Much better bounds than those of our simple construction.)

The polynomial Waring problem

J. Alexander and A. Hirschowitz, J. Algebraic Geom. **4** (1995), 201–222.

How many powers of linear forms do we really need?

The polynomial Waring problem

What is the smallest integer $r(M, n)$ such that a generic homogeneous polynomial $f(x_1, \ldots, x_n)$ of degree M in n variables is expressible as the sum of $r(M, n)$ M-th powers of linear forms?

Theorem (Alexander–Hirschowitz, 1995)

A generic homogeneous polynomial of degree M in n variables is expressible as the sum of

$$r(M, n) = \left\lceil \frac{\binom{n+M-1}{M}}{n} \right\rceil$$

M-th powers of linear forms, with the exception of the cases $r(3, 5) = 8$, $r(4, 3) = 6$, $r(4, 4) = 10$, $r(4, 5) = 15$, and $M = 2$, where $r(2, n) = n$.

(Much better bounds than those of our simple construction.)

The polynomial Waring problem

J. Alexander and A. Hirschowitz, J. Algebraic Geom. **4** (1995), 201–222.

How many powers of linear forms do we really need?

The polynomial Waring problem

What is the smallest integer $r(M,n)$ such that a generic homogeneous polynomial $f(x_1,\dots,x_n)$ of degree M in n variables is expressible as the sum of $r(M,n)$ M-th powers of linear forms?

Theorem (Alexander–Hirschowitz, 1995)

A generic homogeneous polynomial of degree M in n variables is expressible as the sum of

$$r(M,n) = \left\lceil \frac{\binom{n+M-1}{M}}{n} \right\rceil$$

M-th powers of linear forms, with the exception of the cases $r(3,5)=8$, $r(4,3)=6$, $r(4,4)=10$, $r(4,5)=15$, and $M=2$, where $r(2,n)=n$.

(Much better bounds than those of our simple construction.)

The polynomial Waring problem, constructive version

Problem

Given a homogeneous polynomial f of degree M in n variables, find a minimal representation as a sum of powers of linear forms.

(Easy if f happens to be the power of one linear form.)

The polynomial Waring problem, constructive version

Problem

Given a homogeneous polynomial f of degree M in n variables, find a minimal representation as a sum of powers of linear forms.

(Easy if f happens to be the power of one linear form.)

The polynomial Waring problem, constructive version

Problem

Given a homogeneous polynomial f of degree M in n variables, find a minimal representation as a sum of powers of linear forms.

(Easy if f happens to be the power of one linear form.)

Outline

Algorithmic generalization of Brion's formula

We generalize from the power of one linear form to a polynomial in a fixed number of linear forms.

Theorem

Fix D. There exists a polynomial-time algorithm for computing the integral

$$\int_{\Delta} f(\langle \ell_1, \mathbf{x}\rangle, \dots, \langle \ell_D, \mathbf{x}\rangle)\,\mathrm{d}m(\mathbf{x})$$

where

- $\Delta \subseteq \mathbf{R}^n$ *is the simplex with given vertices* $\mathbf{s}_1, \dots, \mathbf{s}_{d+1}$,
- $f \in \mathbf{Q}[X_1, \dots, X_D]$ *is a polynomial of d variables*
- $\mathrm{d}m$ *is the integral Lebesgue measure of the rational affine subspace* $\langle \Delta \rangle$.

Algorithmic generalization of Brion's formula

We use the following generalization:

Theorem

Let $\ell_1, \ldots, \ell_D$ be D linear forms on $\mathbf{R}^n$. We have the following Taylor expansion:

$$\sum_{\mathbf{M} \in \mathbf{N}^D} t_1^{M_1} \ldots t_D^{M_D} \frac{(|\mathbf{M}| + d)!}{d! \operatorname{vol}(\Delta, \mathrm{d}m)} \int_\Delta \frac{\ell_1^{M_1} \cdots \ell_D^{M_D}}{M_1! \ldots M_D!} \, \mathrm{d}m = \frac{1}{\prod_{i=1}^{d+1} (1 - t_1 \langle \ell_1, \mathbf{s}_i \rangle - \cdots - t_D \langle \ell_D, \mathbf{s}_i \rangle)}.$$

The algorithm reads off the blue term as a Taylor coefficient of the right-hand side.

Algorithmic generalization of Brion's formula

We use the following generalization:

Theorem

Let $\ell_1, \dots, \ell_D$ be D linear forms on $\mathbf{R}^n$. We have the following Taylor expansion:

$$\sum_{\mathbf{M} \in \mathbf{N}^D} t_1^{M_1} \dots t_D^{M_D} \, \frac{(|\mathbf{M}| + d)!}{d! \operatorname{vol}(\Delta, \mathrm{d}m)} \int_\Delta \frac{\ell_1^{M_1} \cdots \ell_D^{M_D}}{M_1! \dots M_D!} \, \mathrm{d}m = \frac{1}{\prod_{i=1}^{d+1} (1 - t_1 \langle \ell_1, \mathbf{s}_i \rangle - \dots - t_D \langle \ell_D, \mathbf{s}_i \rangle)}.$$

The algorithm reads off the blue term as a Taylor coefficient of the right-hand side.

Questions

New algorithmic Waring-type questions

- What can I save by decomposing a polynomial into polynomials of D linear forms?
- How do I compute a decomposition?
  ... work in progress for two linear forms...
- What's the best trade-off between few linear forms and few summands?

The big question

Can we make this work well enough for solving optimization problems?!

Questions

New algorithmic Waring-type questions

- What can I save by decomposing a polynomial into polynomials of D linear forms?
- How do I compute a decomposition?

  . . . work in progress for two linear forms. . .
- What's the best trade-off between few linear forms and few summands?

The big question

Can we make this work well enough for solving optimization problems?!

Questions

New algorithmic Waring-type questions

- What can I save by decomposing a polynomial into polynomials of D linear forms?
- How do I compute a decomposition?
  ... work in progress for two linear forms...
- What's the best trade-off between few linear forms and few summands?

The big question

Can we make this work well enough for solving optimization problems?!

Questions

New algorithmic Waring-type questions

- What can I save by decomposing a polynomial into polynomials of D linear forms?
- How do I compute a decomposition?
  ... work in progress for two linear forms...
- What's the best trade-off between few linear forms and few summands?

The big question

Can we make this work well enough for solving optimization problems?!

Questions

New algorithmic Waring-type questions

- What can I save by decomposing a polynomial into polynomials of D linear forms?
- How do I compute a decomposition?
  . . . work in progress for two linear forms. . .
- What's the best trade-off between few linear forms and few summands?

The big question

Can we make this work well enough for solving optimization problems?!

References

Free software

Matthias Köppe. LattE macchiato, version 1.2-mk-0.9, an improved version of De Loera et al.'s LattE program for counting integer points in polyhedra with variants of Barvinok's algorithm, 2007.
http://www.math.ucdavis.edu/~mkoeppe/latte/

Papers

- Jesús A. De Loera, Raymond Hemmecke, Matthias Köppe, Robert Weismantel: Integer Polynomial Optimization in Fixed Dimension. Mathematics of Operations Research, **31** (2006), pp. 147–153.
- Velleda Baldoni, Nicole Berline, Jesús A. De Loera, Matthias Köppe, Michèle Vergne: How to Integrate a Polynomial over a Simplex. Preprint arXiv.org:0809.2083