A Numerical Solution of the Constrained Energy Problem

Steff Helsen

(joint work with Marc Van Barel)

ICCAM, 29/07/04

Outline

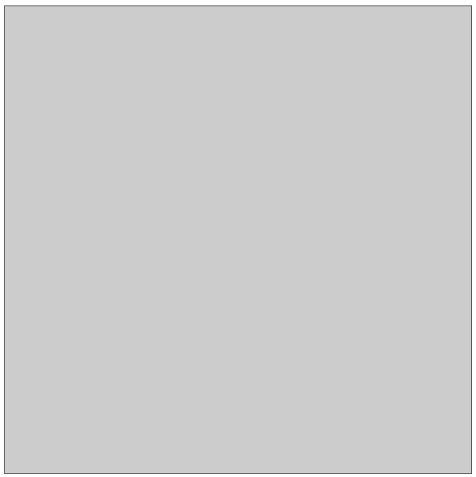
- Motivation
- Potential theory
- Connection with the motivation
- The algorithm
  - Main idea
  - Discretization
  - Refinement
- Todo

Motivation

How are eigenvalues computed?

Motivation

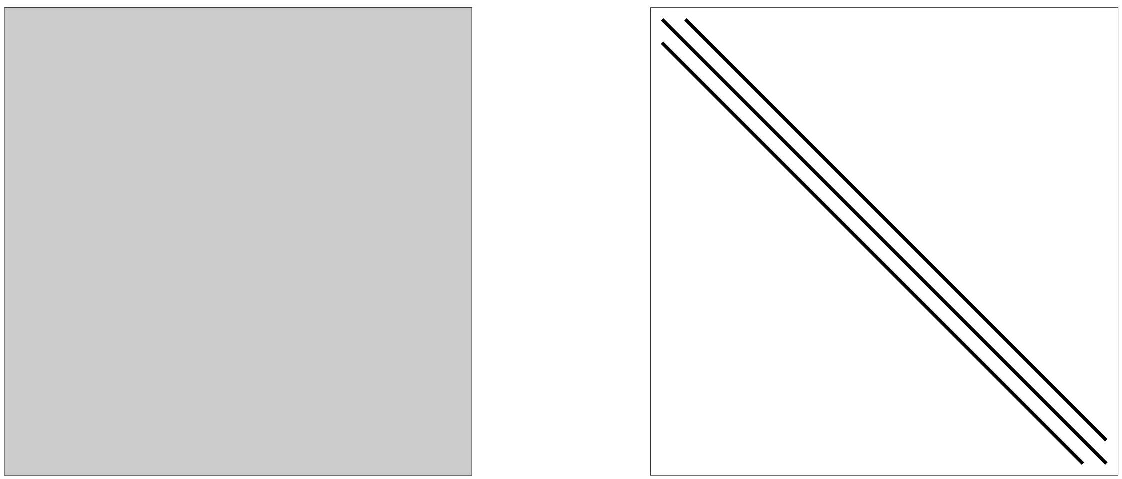
How are eigenvalues computed?



Large Hermitian matrix (dimension m)

Motivation

How are eigenvalues computed?

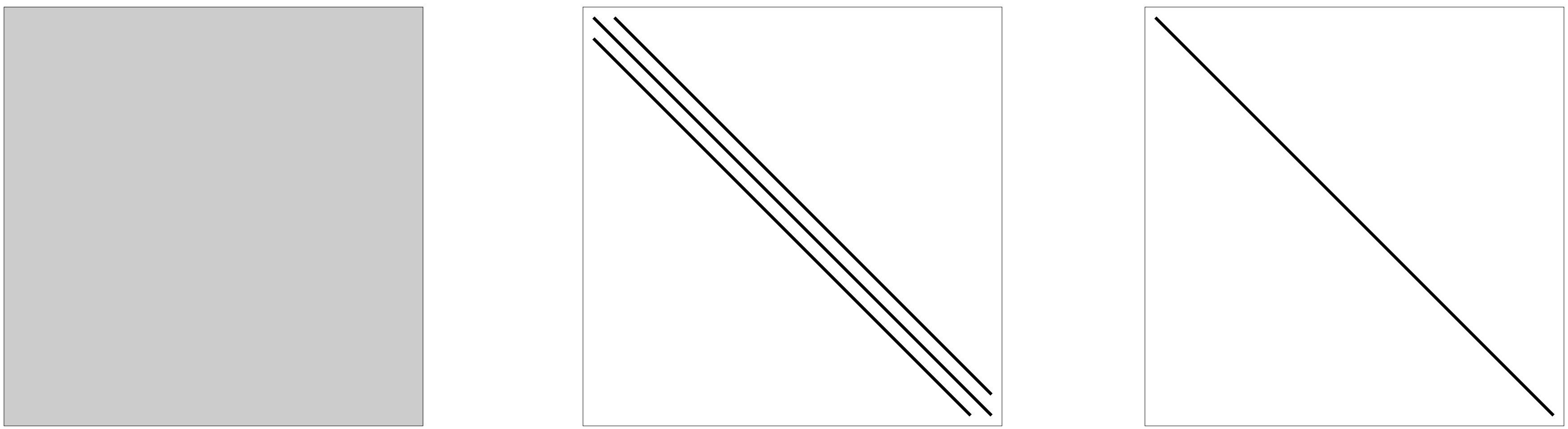


tridiagonalize: finite process

Householder: $\frac{4}{3}m^3$ *flops*

Motivation

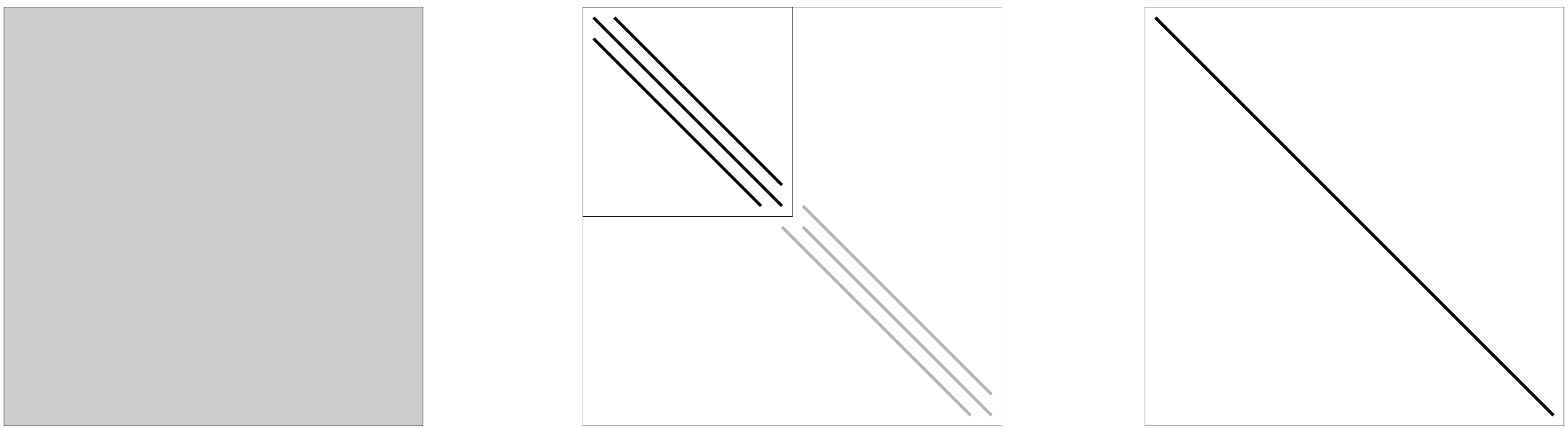
How are eigenvalues computed?



diagonalize: iterative process
QR

Motivation

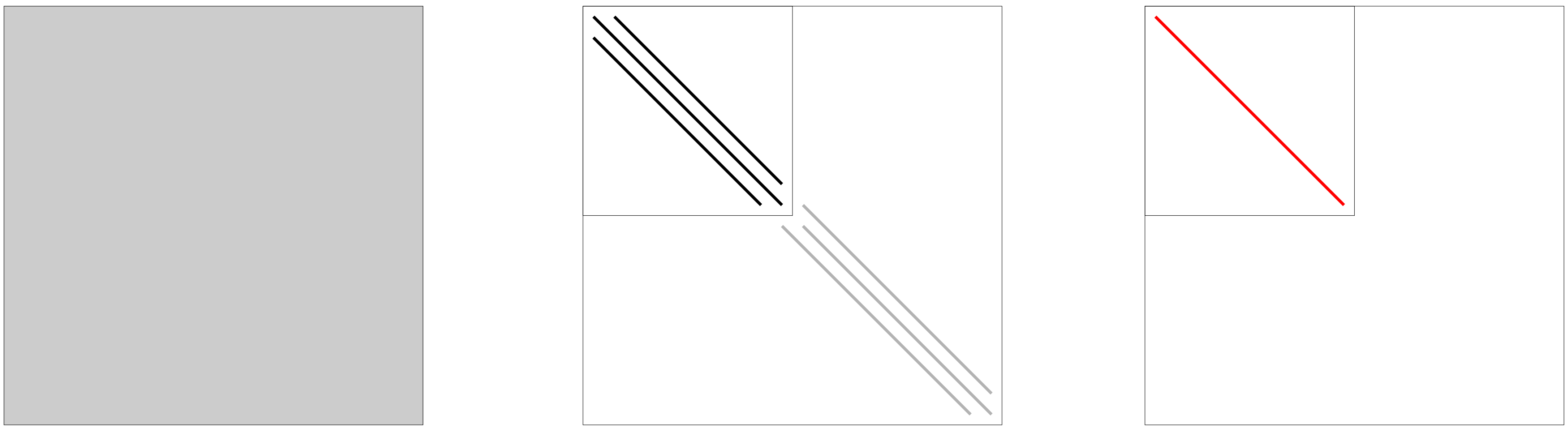
How are eigenvalues computed?



submatrix (dimension n)

Motivation

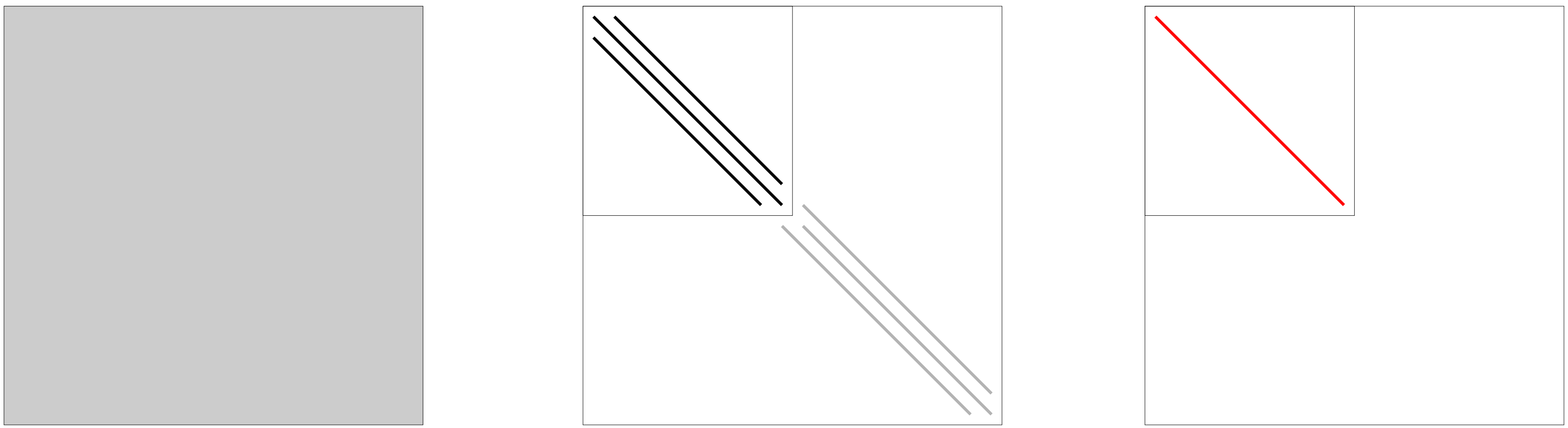
How are eigenvalues computed?



Ritz values

Motivation

How are eigenvalues computed?



Lanczos method

Motivation

Connection eigenvalues and Ritz values?

- eigenvalue distribution σ
- $t = \dfrac{n}{m}$
- Ritz value distribution μ_t ?

Motivation

Connection eigenvalues and Ritz values?

- eigenvalue distribution σ
- $t = \dfrac{n}{m}$
- Ritz value distribution μ_t ?
  - depends only on σ and t (!)
  - $0 \leqslant t\mu_t \leqslant \sigma$
  - ...

Potential theory

μ : measure with compact support on $\mathbb{C}$

the logarithmic potential of μ :

$$U^{\mu}(z) := \int \log \frac{1}{|y-z|} \, \mathrm{d}\mu(y)$$

the logarithmic energy of μ :

$$I(\mu) := \iint \log \frac{1}{|y-z|} \, \mathrm{d}\mu(y) \, \mathrm{d}\mu(z)$$

Potential theory

Energy Problem:

Minimize $I(\mu)$ among all Borel probability measures μ on K.

K: Compact set in $\mathbb{C}$

Potential theory

Energy Problem:

Minimize $I(\mu)$ among all Borel probability measures μ on K.

$\rightarrow \mu_K$ *(equilibrium measure)*

Potential theory

Energy Problem:

Minimize $I(\mu)$ among all Borel probability measures μ on K.

$\rightarrow \mu_K$ *(equilibrium measure)*

property:

- U^{μ_K} is constant on K and smaller everywhere else.

Potential theory

Energy Problem:

Minimize $I(\mu)$ among all Borel probability measures μ on K.

$\rightarrow \mu_K$ *(equilibrium measure)*

Constrained Energy Problem:

Minimize $I(\mu)$ among all Borel probability measures μ that satisfy $0 \leqslant t\mu \leqslant \sigma$.

σ : Borel probability measure with compact support $K \subset \mathbb{C}$
$t \in (0, 1)$

Potential theory

Energy Problem:

Minimize $I(\mu)$ among all Borel probability measures μ on K.

$\rightarrow \mu_K$ *(equilibrium measure)*

Constrained Energy Problem:

Minimize $I(\mu)$ among all Borel probability measures μ that satisfy $0 \leqslant t\mu \leqslant \sigma$.

$\rightarrow \mu_t$

Potential theory

Energy Problem:

Minimize $I(\mu)$ among all Borel probability measures μ on K.

$\rightarrow \mu_K$ *(equilibrium measure)*

Constrained Energy Problem:

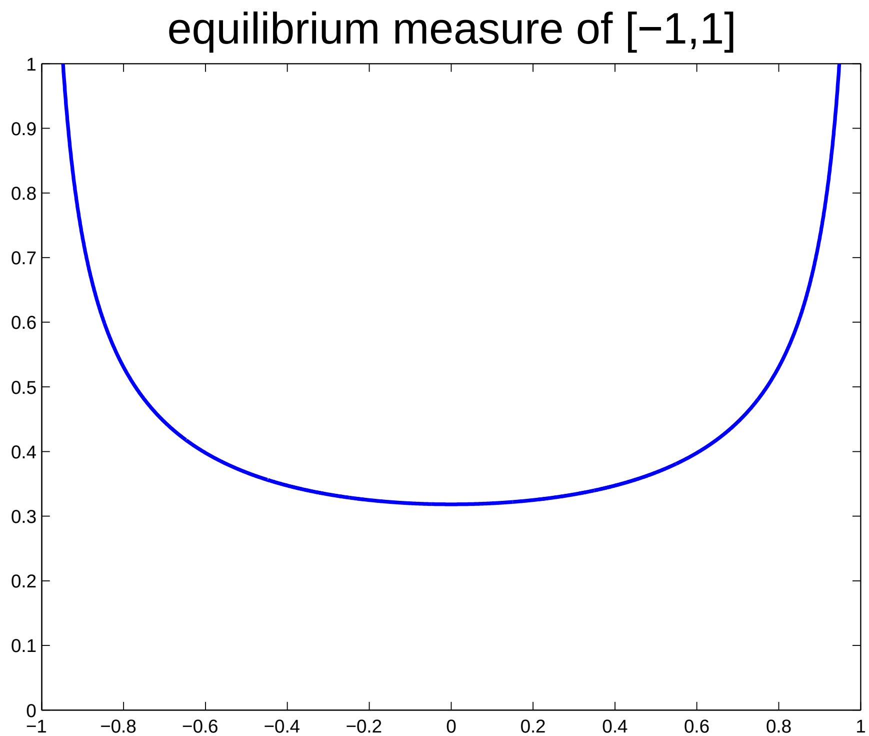
Minimize $I(\mu)$ among all Borel probability measures μ that satisfy $0 \leqslant t\mu \leqslant \sigma$.

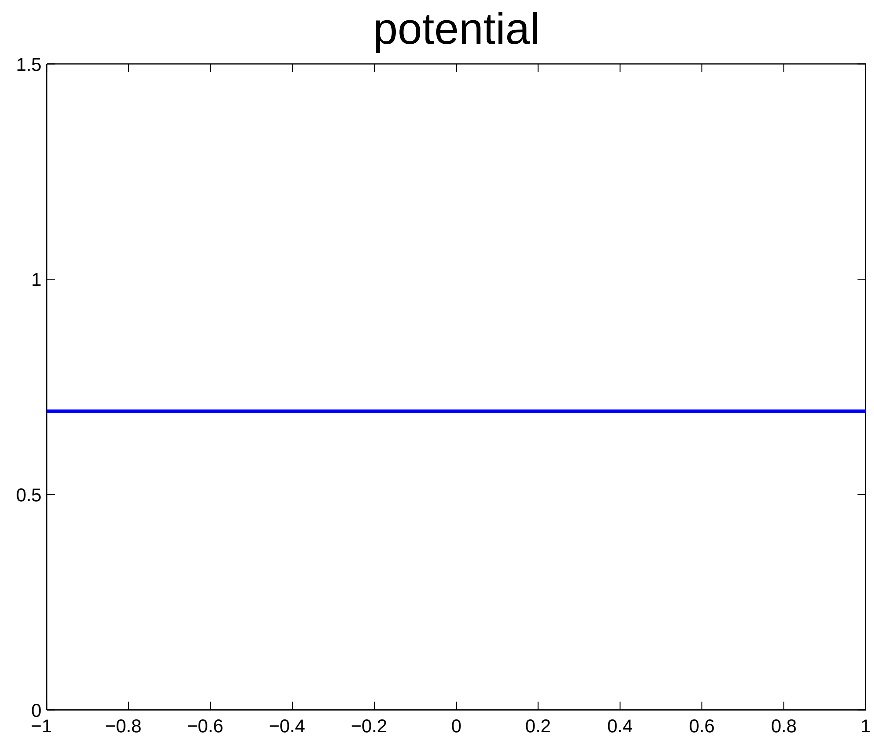
$\rightarrow \mu_t$

properties:

- if $t\mu_K \leqslant \sigma$, then $\mu_t = \mu_K$.
- U^{μ_t} is constant (F_t) on $\mathrm{supp}(\sigma - t\mu_t)$, and smaller everywhere else.

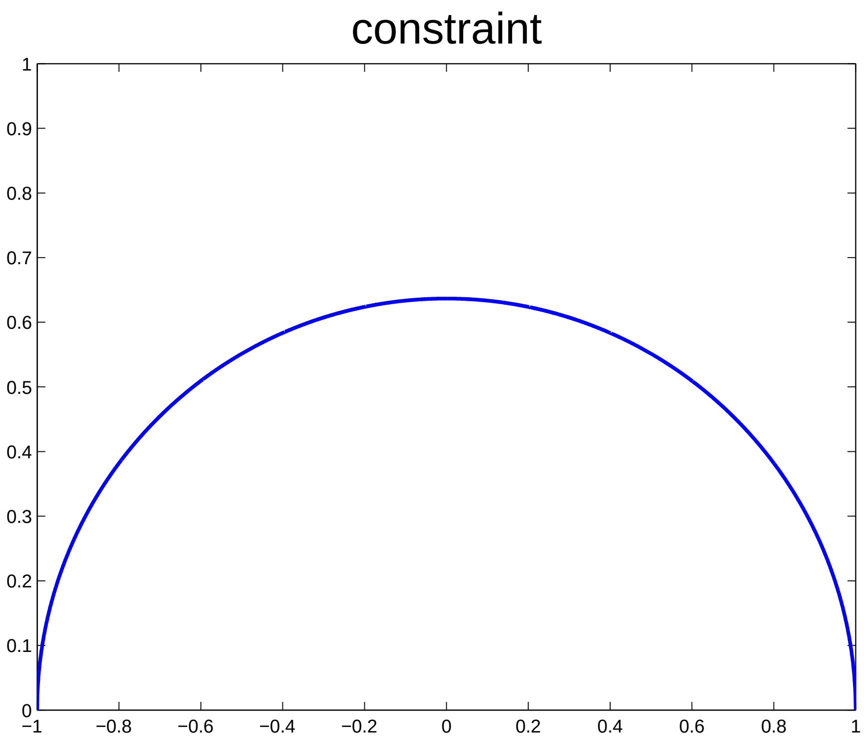
Potential theory

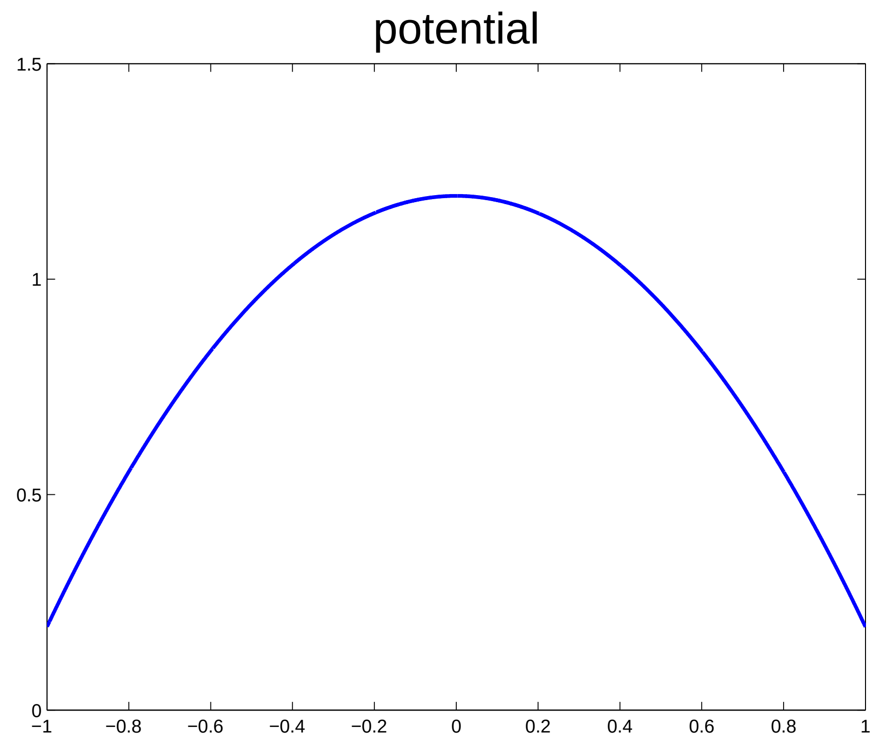




$$\frac{1}{\pi\sqrt{1-x^2}}$$

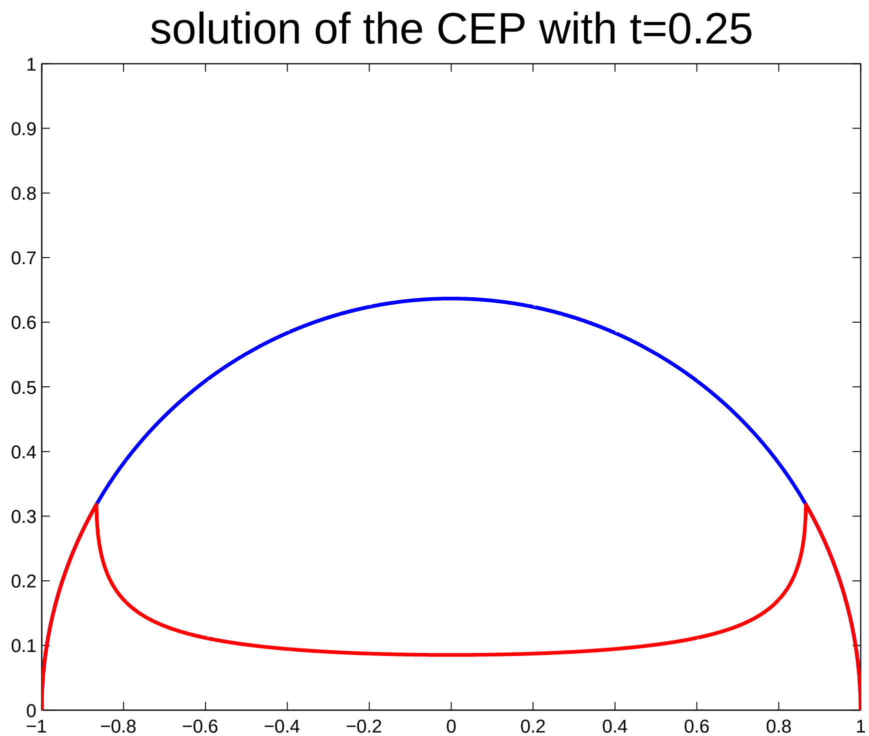
Potential theory

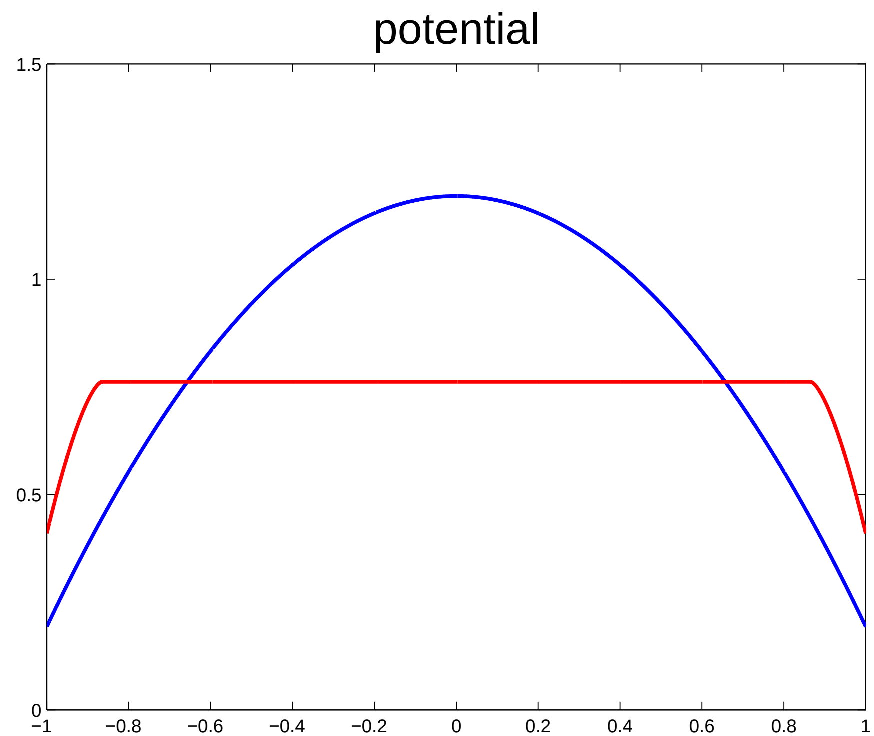




$$\frac{2\sqrt{1-x^2}}{\pi}$$

Potential theory





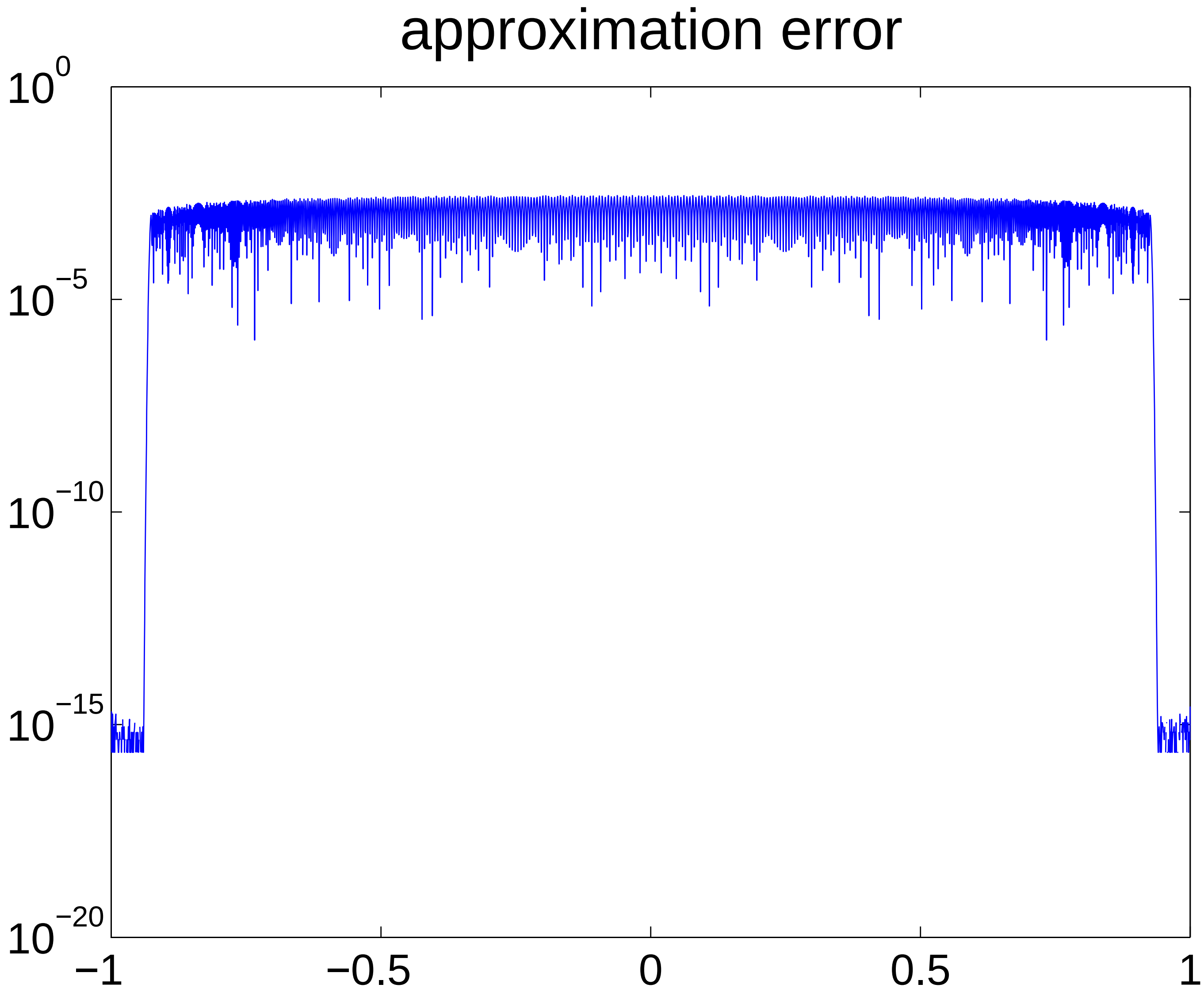
Connection with the motivation

Which eigenvalues are approximated, and the quality of the approximation, can be obtained from the Constrained Energy Problem:

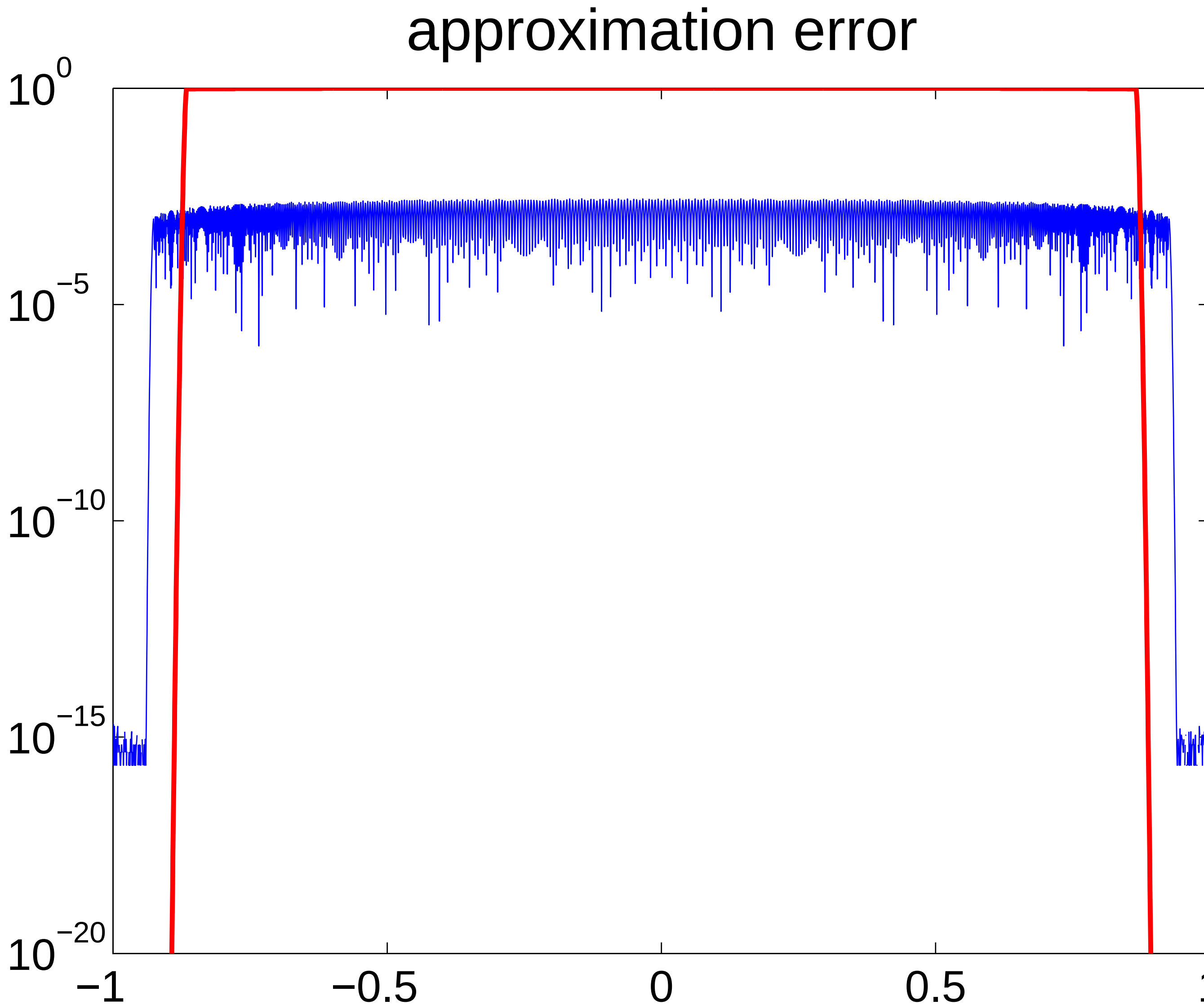
- In the region where $t\mu_t = \sigma$, eigenvalues are well approximated.
- The distance from an eigenvalue λ to the nearest Ritz value θ is given by

$$\exp\big(2n(U^{\mu_t}(\lambda) - F_t)\big).$$

Connection with the motivation



Connection with the motivation



The algorithm: Main idea

Property 1
The only probability measure μ that satisfies $0 \leqslant t\mu \leqslant \sigma$ and whose potential U^{μ} is constant on $\mathrm{supp}(\sigma - t\mu)$ and smaller everywhere else, is μ_t.

The algorithm: Main idea

Property 1

The only probability measure μ that satisfies $0 \leqslant t\mu \leqslant \sigma$ and whose potential U^{μ} is constant on $\mathrm{supp}(\sigma - t\mu)$ and smaller everywhere else, is μ_t.

Property 2

Suppose μ is a probability measure whose potential U^{μ} is constant on $\mathrm{supp}(\sigma - t\mu)$, then $\mathrm{supp}(\sigma - t\mu_t)$ is a subset of $\mathrm{supp}(\sigma - t\mu)^{+}$.

The algorithm: Main idea

Property 1

The only probability measure μ that satisfies $0 \leqslant t\mu \leqslant \sigma$ and whose potential U^{μ} is constant on $\operatorname{supp}(\sigma - t\mu)$ and smaller everywhere else, is μ_t.

Property 2

Suppose μ is a probability measure whose potential U^{μ} is constant on $\operatorname{supp}(\sigma - t\mu)$, then $\operatorname{supp}(\sigma - t\mu_t)$ is a subset of $\operatorname{supp}(\sigma - t\mu)^{+}$.

Corollary

Suppose μ is a probability measure whose potential U^{μ} is constant on $\operatorname{supp}(\sigma - t\mu)$, then on the region where $t\mu \geqslant \sigma$, $t\mu_t = \sigma$.

The algorithm: Main idea

Algorithm

input: σ, t

$I := \operatorname{supp}(\sigma)$
$J := \emptyset$
while (not converged)

$$\mu|_J := \frac{1}{t}\sigma|_J$$

solve $\begin{cases} U^{\mu|_I} = C - U^{\mu|_J} \\ \|\mu|_I\| = 1 - \|\mu|_J\| \end{cases}$

$$I := \{\text{“}t\mu < \sigma\text{”}\}$$

$$J := \{\text{“}t\mu \geqslant \sigma\text{”}\}$$

$\mu_t := \mu$

demo 1

The algorithm: Main idea

Remark:

The only probability measure μ that satisfies $0 \leqslant t\mu \leqslant \sigma$ and whose potential U^{μ} is constant on $\mathbf{supp}(\sigma - t\mu)$ and smaller everywhere else, is μ_t.

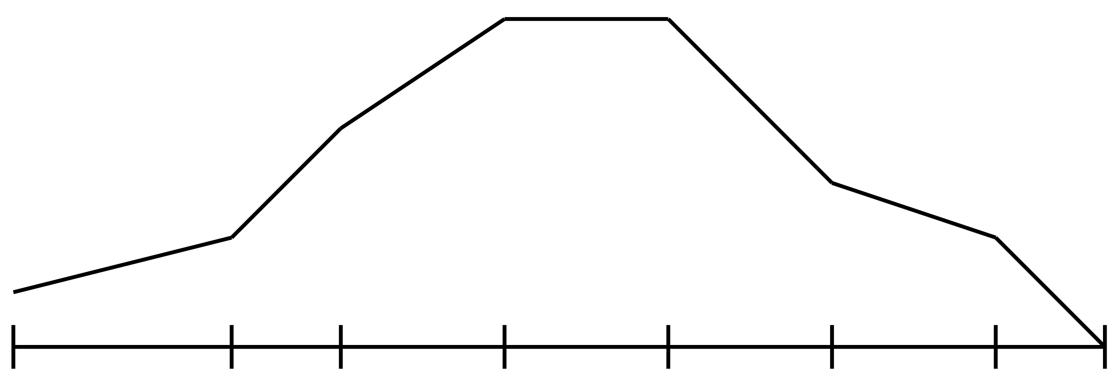
solve $\begin{cases} U^{\mu|_I} = C - U^{\mu|_J} \\ \|\mu|_I\| = 1 - \|\mu|_J\| \end{cases}$

We do not ask the potential to be smaller everywhere else, but one can prove that it is satisfied in every step of the algorithm.

The algorithm: Discretization

We discretize the support of σ : $x_0, x_1, \ldots, x_N$ and we suppose the density of μ is piecewise linear on each of the subintervals.

$$\mathrm{d}\mu(x) = (a_j x + b_j)\mathrm{d}x \qquad \text{for } x \in [x_{j-1}, x_j].$$

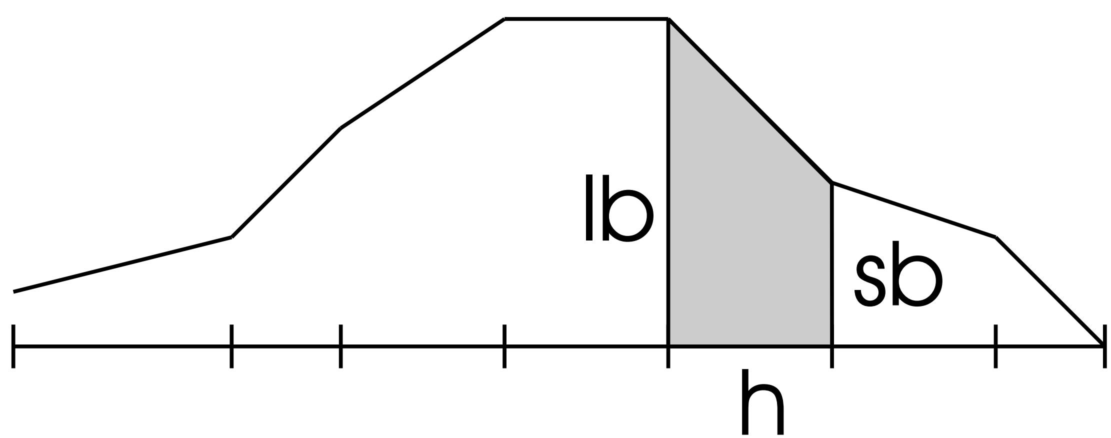


The algorithm: Discretization

We discretize the support of σ : $x_0, x_1, \ldots, x_N$ and we suppose the density of μ is piecewise linear on each of the subintervals.

$$\mathrm{d}\mu(x) = (a_j x + b_j)\mathrm{d}x \qquad \text{for } x \in [x_{j-1}, x_j].$$

The area of a trapezoid is $\frac{1}{2}(\text{lb} + \text{sb}\,)\times$ h, so the total mass of μ is $\sum_{j=1}^{N}(\mu_{j-1} + \mu_j)(x_j - x_{j-1})/2$. With this, we can create a rowvector $\overrightarrow{m}$ such that $\overrightarrow{m} \cdot \overrightarrow{\mu} = \|\mu\|$.



Algorithm: Discretization

$$U^{\mu}(y) = \int \log \frac{1}{|x-y|} \ \mathrm{d}\mu(x)$$

$$= \sum_j \int_{x_{j-1}}^{x_j} \log \frac{1}{|x-y|} (a_j x + b_j) \ \mathrm{d}x$$

Algorithm: Discretization

$$U^{\mu}(y) = \int \log \frac{1}{|x-y|} \, \mathrm{d}\mu(x)$$
$$= \sum_j \int_{x_{j-1}}^{x_j} \log \frac{1}{|x-y|} (a_j x + b_j) \, \mathrm{d}x$$

The primitive function of $x \mapsto \log \frac{1}{|x-y|}$ is

$$f(x,y) := \begin{cases} (x-y)(\log|x-y|-1) & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases}$$

Algorithm: Discretization

$$U^\mu(y) = \int \log \frac{1}{|x-y|}\ \mathrm{d}\mu(x)$$
$$= \sum_j \int_{x_{j-1}}^{x_j} \log \frac{1}{|x-y|}(a_j x + b_j)\ \mathrm{d}x$$

The primitive function of $x \mapsto x \log \frac{1}{|x-y|}$ is

$$g(x,y) := \begin{cases} \frac{1}{2}\log|x-y|(x^2-y^2) + \frac{1}{4}(x+y)^2 & \text{if } x \neq y \\ y^2 & \text{if } x = y \end{cases}$$

Algorithm: Discretization

$$
\begin{aligned}
U^{\mu}(y) &= \int \log \frac{1}{|x-y|} \ \mathrm{d}\mu(x) \\
&= \sum_j \int_{x_{j-1}}^{x_j} \log \frac{1}{|x-y|}(a_j x + b_j) \ \mathrm{d}x \\
&= \sum a_j \big(g(x_j, y) - g(x_{j-1}, y)\big) + b_j \big(f(x_j, y) - f(x_{j-1}, y)\big)
\end{aligned}
$$

Algorithm: Discretization

$$
\begin{aligned}
U^{\mu}(y) &= \int \log \frac{1}{|x-y|} \; \mathrm{d}\mu(x) \\
&= \sum_j \int_{x_{j-1}}^{x_j} \log \frac{1}{|x-y|} (a_j x + b_j) \; \mathrm{d}x \\
&= \sum a_j \big(g(x_j, y) - g(x_{j-1}, y) \big) + b_j \big(f(x_j, y) - f(x_{j-1}, y) \big)
\end{aligned}
$$

$$
\begin{cases} \mu_{j-1} = a_j x_{j-1} + b_j \\ \mu_j = a_j x_j + b_j \end{cases} \Rightarrow \begin{cases} a_j = \dfrac{\mu_j - \mu_{j-1}}{x_j - x_{j-1}} \\ b_j = \mu_j - a_j x_j = \dfrac{x_j \mu_{j-1} - x_{j-1} \mu_j}{x_j - x_{j-1}} \end{cases}
$$

Algorithm: Discretization

$$U^{\mu}(y) = \int \log \frac{1}{|x-y|} \, \mathrm{d}\mu(x)$$
$$= \sum_j \int_{x_{j-1}}^{x_j} \log \frac{1}{|x-y|} (a_j x + b_j) \, \mathrm{d}x$$
$$= \sum a_j \big(g(x_j, y) - g(x_{j-1}, y) \big) + b_j \big(f(x_j, y) - f(x_{j-1}, y) \big)$$

$$\big(U^{\mu}(x_j) \big)_j = P \overrightarrow{\mu}$$

demo 2

&

demo 3

Todo

- code optimization
- stability
- more tests
- multiple intervals
- comparing with other algorithms
- other applications
- ...

Thank you
for your attention!

(The End)