14. Computing integrals and sums [Fernandez2025, Sec. 5]

Quadratures for multivariate integrals (revisiting TCI.2)

Consider $I = \int d^{\mathcal{N}} x \; f(\vec{x}) \quad , \quad \vec{x} = (x_1, \, .\,., x_{\mathcal{N}}) \in D = D_1 \times \cdots \times D_{\mathcal{N}}$ (1)

1D integrals can be computed using discrete grid of points $\{x_\ell(\sigma_\ell)\}$ and quadrature weights, $\{w_\ell(\sigma_\ell)\}$ (2)

$\sigma_\ell = 1, \dots, d$

$$\int d x_\ell \, f(x_\ell) \simeq \sum_{\sigma_\ell = 1}^{d} w_\ell(\sigma_\ell) \, f(x_\ell(\sigma_\ell)) \tag{3}$$

(e.g. Gauss-Kronrod or Gauss-Legendre weights)

TCI unfolding of f yields factorized representation: $f(\vec{x}(\vec{\sigma})) = F_{\vec{\sigma}} \overset{\text{TCI}}{\simeq} \tilde{F}_{\vec{\sigma}} = \prod_{\ell=1}^{\mathcal{N}} M_\ell^{\sigma_\ell}$ (4)

Integral factorizes too:

$$\int d^{\mathcal{N}} x \; f(\vec{x}) \overset{(3)}{\simeq} \underbrace{\sum_{\vec{\sigma}}}_{d^{\mathcal{N}} \text{ grid points}} \left(\prod_{\ell=1}^{\mathcal{N}} w_\ell(\sigma_\ell) \right) f(\vec{x}(\vec{\sigma})) \overset{(4)}{\simeq} \prod_{\ell=1}^{\mathcal{N}} \underbrace{\sum_{\sigma_\ell=1}^{d} w_\ell(\sigma_\ell) \, M_\ell^{\sigma_\ell}} \tag{5}$$

represents χ^2 1d integrals, viewing M_ℓ as $\chi \times \chi$ matrix

Alternative: use 'weighted unfolding': $\left(\prod_{\ell=1}^{\mathcal{N}} w_\ell(\sigma_\ell) \right) f(\vec{x}(\vec{\sigma})) = F_{\vec{\sigma}} \overset{\text{TCI}}{\simeq} \tilde{F}_{\vec{\sigma}} = \prod_{\ell=1}^{\mathcal{N}} M_\ell^{\sigma_\ell}$ (6)

Then:

$$\int d^{\mathcal{N}} x \; f(\vec{x}) \overset{(3)}{\simeq} \sum_{\vec{\sigma}} \left(\prod_{\ell=1}^{\mathcal{N}} w_\ell(\sigma_\ell) \right) f(\vec{x}(\vec{\sigma})) \overset{(4)}{\simeq} \prod_{\ell=1}^{\mathcal{N}} \sum_{\sigma_\ell=1}^{d_\ell} M_\ell^{\sigma_\ell} \tag{7}$$

Weighted unfolding may achieving higher accuracy for given χ, since error estimation during TCI construction includes information about weights.
Weighted unfolding is typically combined with 'environment error' unfolding scheme (see TCI.9.10) designed for computing integrals.

First example of efficiency of integration via TCI unfolding, see TCI.2.

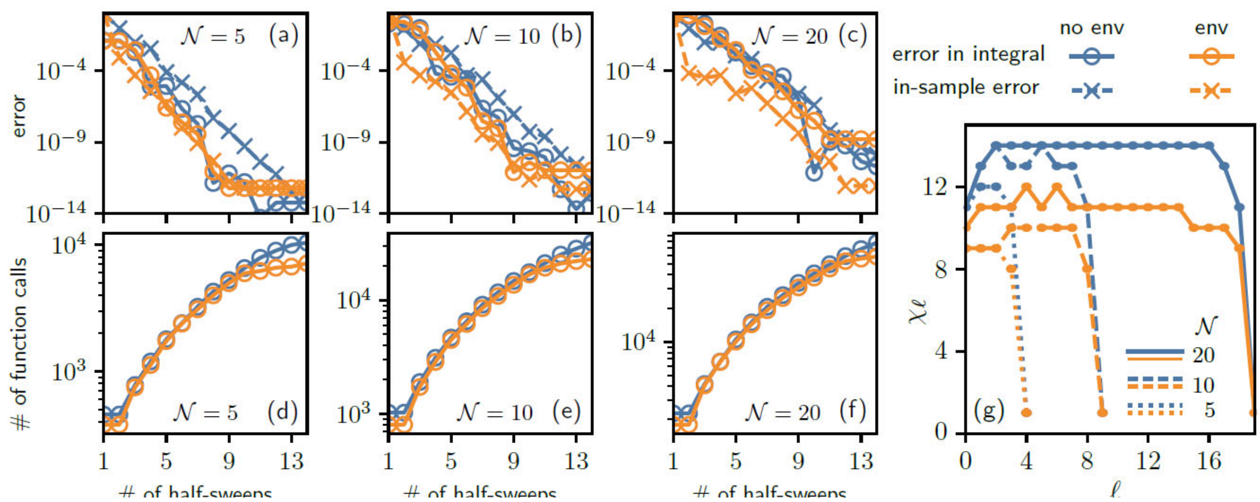
Second example: [Fernandez2025, Sec. 5.2]

$$I^{(\mathcal{N})} = \int_{[0,1]^{\mathcal{N}}} dx_1 \cdots dx_{\mathcal{N}} f(\mathbf{x}), \qquad f(\mathbf{x}) = \frac{2^{\mathcal{N}}}{1 + 2\sum_{\ell=1}^{\mathcal{N}} x_\ell} \tag{8}$$

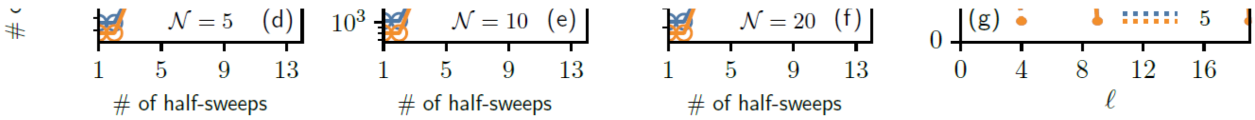
(9)

Integral is known exactly, e.g. $I^{(5)} = [-65205 \log(3) - 6250 \log(5) + 24010 \log(7) + 14641 \log(11)]/24$

TCI computation of integral using 15 Gauss-Kronrod grid (i.e. $d_\ell = 15$), i.e. # of grid points = $(15)^{\mathcal{N}}$



(10)



relative 'error in integral' (circles): $|1 - I^{(\mathcal{N})} / \tilde{I}^{(\mathcal{N})}|$ (11)

relative 'in sample error' (crosses) : $\max |1 - F_{\vec\sigma} / \tilde{F}_{\vec\sigma}|$ (12)

over all sampled pivots

blue: no environment mode
orange: environment mode
[cf. (TCI.9.10)]

Error decreases rapidly with # of half-sweeps!

function calls remains modest , $< 10^5$, even for $\mathcal{N} = 20$.

<u>Computation of sums: partition function of 1D Ising chain</u>

2^L configurations

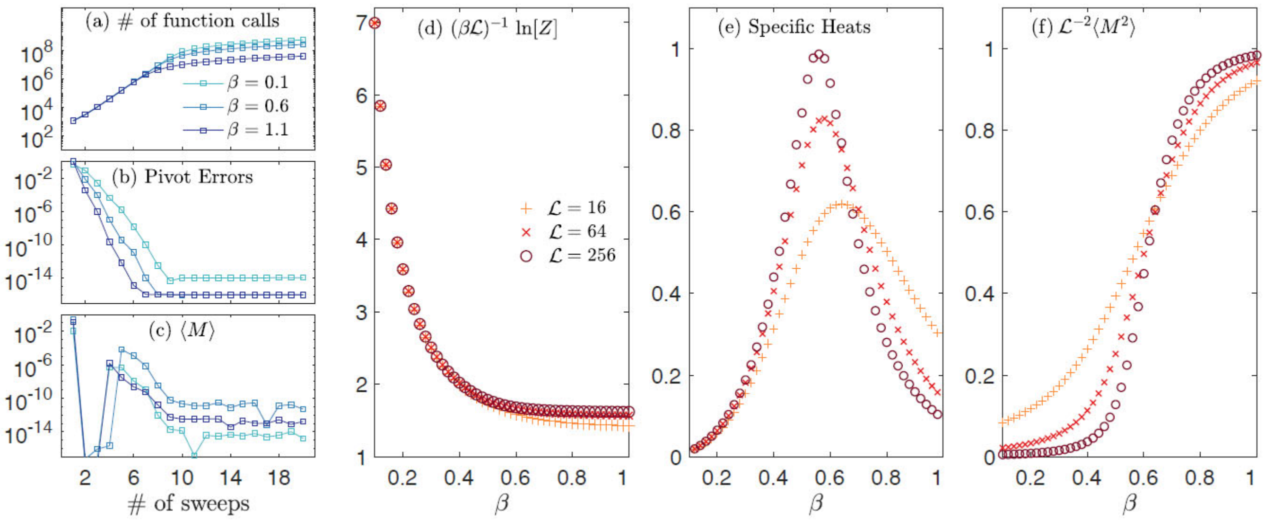
Partition function: $Z = \sum_{\vec\sigma} W_{\vec\sigma}$, Boltzmann weight: $W_{\vec\sigma} = e^{-\beta E_{\vec\sigma}}$ $\quad \sigma_\ell \in \{+1, -1\}$ (13)

1D Ising model: energy of configuration $\vec\sigma$: $E_{\vec\sigma} = -\sum_{\ell<\ell'} J_{\ell\ell'} \sigma_\ell \sigma_{\ell'}$ long-ranged coupling: $J_{\ell\ell'} = |\ell - \ell'|^{-2}$ (14)

TCI-factorize Boltzmann weight: $Z = \sum_{\vec\sigma} W_{\vec\sigma} \overset{\text{TCI}}{\simeq} \sum_{\vec\sigma} \tilde{W}_{\vec\sigma} = \sum_{\vec\sigma} \prod_{\ell=1}^{L} M_\ell^{\sigma_\ell} = \prod_{\ell=1}^{L} \sum_{\sigma_\ell} M_\ell^{\sigma_\ell}$ (15)

Free energy: $F = (\beta L)^{-1} \ln Z$, specific heat: $C = \beta^2 \frac{\partial \ln Z}{\partial \beta^2}$ (16)

Magnetization: $M = \sum_{\vec\sigma} W_{\vec\sigma} \left(\sum_{\ell=1}^{L} \sigma_\ell / L \right)$ (17)



Number of function calls (a) grows exponentially with # sweeps until pivot error (b) approaches convergence.

15. Quantics representations of functions   [Fernandez2025, Sec. 6]   TCI.15

Many functions have very sharp structures, or large domains of definition, or both.
Corresponding discretization grids must have high density of points, or a large domain, or both.
In short: we need grids with exponentially many grid points.

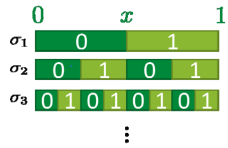
Solution: $\boxed{\text{'quantics' representation: use binary representation of each variable!}}$

Function of 1 variable $f : [0,1) \to \mathbb{C}\,, \quad x \mapsto f(x)$ (1)

Define uniform grid: $x = m/M\,, \quad m = 0, \dots, M-1\,,$ with $M = 2^R$ (2)

Binary form of grid index using R bits: $m = (\sigma_1 \cdots \sigma_R)_2 = \sum_{r=1}^{R} \sigma_r 2^{R-r}\,, \quad \sigma_r \in \{0,1\}$ (3)

e.g. $(011010)_2$ describes structures at scale 2^{-r}



Discretized variable: $\vec{\sigma} = (\sigma_1, \dots, \sigma_R)\,, \quad x(\vec{\sigma}) = x(m(\vec{\sigma}))$ (4)

Discretized function:
'quantics representation' of f $F_{\vec{\sigma}} = f(x(\vec{\sigma})) =$ 

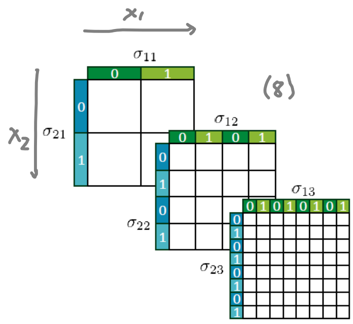
$\mathcal{L} = R$ indices,
each of dimension $d = 2$ (5)

Function of $\mathcal{N}$ variables $f : [0,1)^{\mathcal{N}} \to \mathbb{C}\,, \quad \vec{x} = (x_1, \dots, x_n, \dots, x_{\mathcal{N}}) \mapsto f(\vec{x})\,, \quad n = 1, \dots, \mathcal{N}$ (6)

Define uniform grid: $x_n = m_n / M\,, \quad m_n = 0, \dots, M-1\,,$ with $M = 2^R$ (7)

Binary form of grid index
using R bits per variable: $m_n = (\sigma_{n1}, \dots, \sigma_{nR})_2 = \sum_{r=1}^{R} \sigma_{nr}\, 2^{R-r}\,, \quad \sigma_{nr} \in \{0,1\}$ (8)

describes structures of
x_n at scale 2^{-r}



There are different possibilities for ordering the indices:

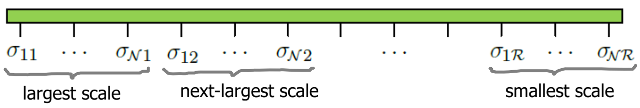
1. 'Interleaved quantics representation': group all bits describing the scale together:

$$\vec{\sigma} = (\underbrace{\sigma_{11}, \dots, \sigma_{\mathcal{N}1}}_{\text{largest scale: } 2^{-1}}, \underbrace{\sigma_{12}, \dots, \sigma_{\mathcal{N}2}}_{\text{next-largest scale: } 2^{-2}}, \dots, \underbrace{\sigma_{1R}, \dots, \sigma_{\mathcal{N}R}}_{\text{smallest scale: } 2^{-R}}) \tag{9}$$

indices are ordered by scale!   this is called 'scale separation'

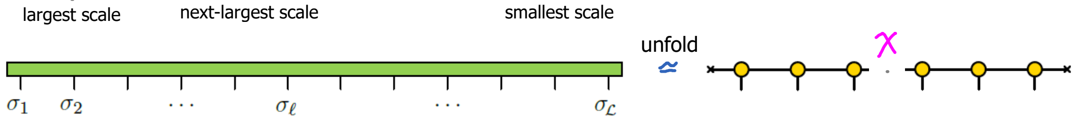
relabel indices:

$$\vec{\sigma} = (\sigma_1, \dots, \sigma_\ell, \dots, \sigma_{\mathcal{L}}) \quad \text{with} \quad \mathcal{L} = \mathcal{N}R\,, \quad \sigma_{\ell(n,r)} = \sigma_{nr}\,, \quad \ell = n + (r-1)\mathcal{N} \tag{10}$$

$f(\vec{x}(\vec{\sigma})) =$ 

(11)

(12)

$=$ 

$= \widetilde{F}_{\vec{\sigma}}$

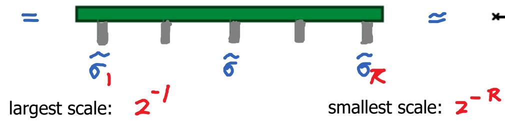
Once $F_{\vec{\sigma}}$ has been defined, it can be unfolded. The resulting $\widetilde{F}_{\vec{\sigma}}$ is called a 'quantics tensor train' (QTT).

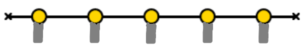
If variables at same scale are strongly 'entangled', but less so for variables at different scales, $\widehat{F}_{\vec{\sigma}}$ will have fairly low bond dimension, i.e. χ is strongly compressible. This turns out to be the case for many physical applications.

2. '<u>Fused quantics representation</u>': 'fuse' all bits for scale 2^{-r} into a single variable:

$$\tilde\sigma_r = (\sigma_{Nr}\cdots\sigma_{2r}\sigma_{1r})_2 = \sum_{n=1}^{N} 2^{n-1}\sigma_{nr} \in \{0,\dots,2^N-1\}, \qquad \tilde{\boldsymbol\sigma} = (\tilde\sigma_1,\dots,\tilde\sigma_R) \tag{13}$$

$$F_{\tilde{\boldsymbol\sigma}} = f(\vec x(\tilde{\boldsymbol\sigma})) = \quad \text{[diagram]} \quad \simeq \quad \text{[diagram]} \quad = \tilde F_{\tilde{\boldsymbol\sigma}} \tag{14}$$





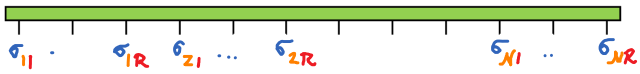
$\mathcal{L} = R, \quad d = 2^N$

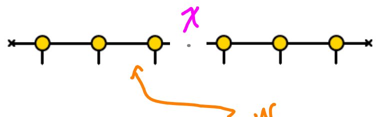
Again: if variables at different scales are not strongly entangled, $F_{\tilde{\boldsymbol\sigma}}$ will be strongly compressible.

3. Group together all bits addressing a given variable x_n, as done in the 'natural' tensor representation.

$$\vec\sigma = (\underbrace{\sigma_{11},\dots,\sigma_{1R}}_{x_1}, \underbrace{\sigma_{21},\dots,\sigma_{2R}}_{x_2}, \dots, \underbrace{\sigma_{N1},\dots,\sigma_{NR}}_{x_N}) \overset{\text{relabel}}{=:} (\sigma_1,\dots,\sigma_\ell,\dots,\sigma_{\mathcal L}), \qquad \mathcal L = NR \tag{15}$$

$$F_{\vec\sigma} = f(\vec x(\vec\sigma)) = \quad \text{[diagram]} \quad \overset{\text{unfold}}{\simeq} \quad \text{[diagram]} \quad = \tilde F_{\vec\sigma} \tag{16}$$





This is suitable if different variables are not strongly entangled, e.g. if function factorizes: $f(\vec x) = \prod_{n=1}^{N} f_n(x_n)$

Some simple analytic functions are approximated well as QTT with .

<u>1D examples</u>:

Example: pure exponential factorizes completely, yielding $F_{\vec\sigma}$ with $\chi = 1$:

$$f(x) = e^{\lambda x} \overset{(3)}{=} \exp\Big[\lambda \sum_{r=1}^{R} \sigma_r 2^{-r}\Big] = \prod_{r=1}^{R} \exp\big[\lambda\, \sigma_r 2^{-r}\big] \tag{17}$$

Example: sin and cosine yield $F_{\vec\sigma}$ with $\chi = 2$, since they can be expressed as sums of two exponentials:

$$\sin(x) = \tfrac{1}{2i}\left(e^{ix} - e^{-ix}\right), \qquad \cos x = \tfrac{1}{2}\left(e^{ix} + e^{-ix}\right) \tag{18}$$

Dirac delta function has $\chi = 1$. $\quad \delta(x - x') = \prod_{r=1}^{R} \delta_{\sigma_r, \sigma'_r}$ (all bits of x must equal all bits of x') (19)

Heavy-side step function $\Theta(x)$ has $\chi = 2$. (Show it!)

By contrast, random noise is incompressible.

Generally, if function has low quantics rank, sites representing different scales are not strongly 'entangled'.

The quantics representation makes this notion precise: cut bond, compute entanglement entropy between left and right parts of chain (as though it represented a quantum state) .

<u>2D example</u>: Kronecker symbol $f(m_1, m_2) = \delta_{m_1, m_2}$ with binary representation (8) for m_n (20)

Its matrix representation, a $2^R \times 2^R$ unit matrix, is incompressible by SVD (all singular values are 1).

Quantics representation: $f(m_1, m_2) = \prod_{r=1}^{\mathcal{R}} \delta_{\sigma_{1r}, \sigma_{2r}}$ = rank-1 MPS by fusing σ_{1r} and σ_{2r} (21)

<u>Multi-dimensional functions</u>: $f(\vec{x})$, $\vec{x} \in \mathbb{R}^{\mathcal{N}}$

Pure exponential has $\chi = 1$: $f(\vec{x}) = e^{\vec{\lambda}\cdot\vec{x}} = \prod_{n=1}^{\mathcal{N}} e^{\lambda_n x_n} = \prod_{n=1}^{\mathcal{N}} \prod_{r=1}^{\mathcal{R}} e^{\lambda_n \sigma_{nr} 2^{-r}}$ (22)

Dirac delta has $\chi = 1$ $\delta(\vec{x} - \vec{x}') = \prod_{n=1}^{\mathcal{N}} \prod_{r=1}^{\mathcal{R}} \delta_{\sigma_{nr}, \sigma_{n'r'}}$ (all bits of $\vec{x}$ must equal all bits of $\vec{x}'$) (23)

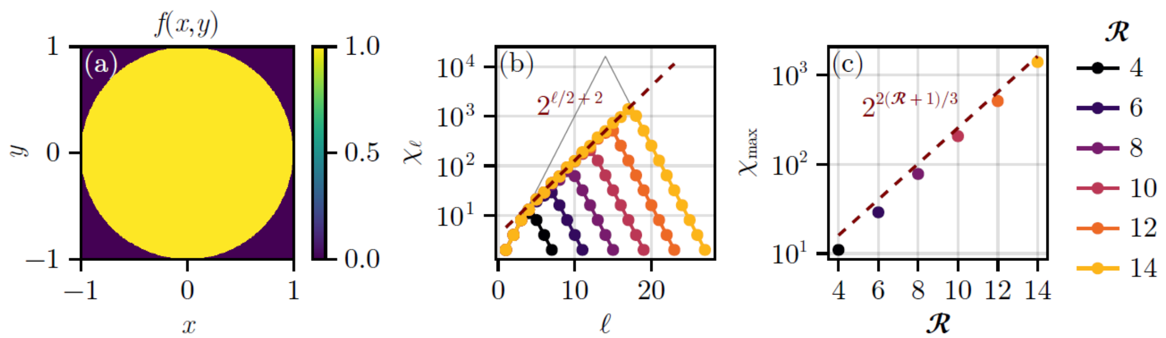
Step function with argument linear in $\vec{x}$, $\Theta(\vec{v}\cdot\vec{x} - \vec{b})$ has $\lambda = 2$ (24)

In all these examples, bond dimension is small due to 'separability of length scales'.

Example where bond dimension is not small: $f(\vec{x}) = \Theta(1 - \vec{x}^2) = \begin{cases} 1 & \text{inside unit sphere} \\ 0 & \text{outside unit sphere} \end{cases}$ (25)

Because surface of sphere is curve, χ_{max} depends on resolution with which curvature is resolved, i.e. on $\mathcal{R}$

$\mathcal{N} = 2$.



(26)

(b) $\chi_\ell \sim 2^{\ell/2}$ increases exponentially, because it additional pair of bits σ_{1r}, σ_{2r} doubles # of points close to circle, which are those containing new information

(c) $\chi_{max} \sim 2^{2(\mathcal{R}+1)/3}$ depends on $\mathcal{R}$, independent of specified tolerance, because step is abrupt. For broadened step, χ_{max} decreases significantly (not shown).

<u>Integrals</u>: are approximated as Riemann sums, then factorized over quantics bits:

$$\int_{[0,1]^{\mathcal{N}}} d^{\mathcal{N}}\mathbf{x} f(\mathbf{x}) \approx \frac{1}{2^{\mathcal{L}}} \sum_{\boldsymbol{\sigma}} f(\mathbf{x}(\boldsymbol{\sigma})) = \frac{1}{2^{\mathcal{L}}} \sum_{\boldsymbol{\sigma}} F_{\boldsymbol{\sigma}} \approx \frac{1}{2^{\mathcal{L}}} \sum_{\boldsymbol{\sigma}} \widetilde{F}_{\boldsymbol{\sigma}} = \frac{1}{2^{\mathcal{L}}} \prod_{\ell=1}^{\mathcal{L}} \Big[\sum_{\sigma_\ell=1}^{2} M_\ell^{\sigma_\ell} \Big] \quad (27)$$

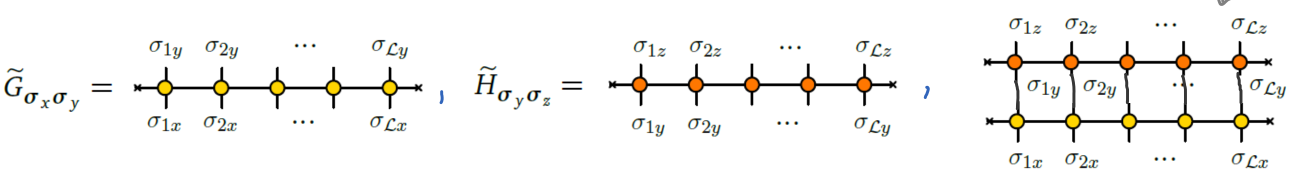
integration volume element, with $\mathcal{L} = \mathcal{N}\mathcal{R}$

# of discretization points $\sim 2^{\mathcal{L}}$ $\Rightarrow$ discretization error of integral $\sim 2^{-\mathcal{L}}$ ← exponential in $\mathcal{L}$!!

cost of TCI factorization $\mathcal{O}(\chi^2 d \mathcal{L})$ ← linear in $\mathcal{L}$!!

<u>'Matrix products'</u>: $f(\mathbf{x}, \mathbf{z}) = \int_D d^{\mathcal{N}}\mathbf{y}\, g(\mathbf{x}, \mathbf{y}) h(\mathbf{y}, \mathbf{z}) \approx 2^{-\mathcal{L}} \sum_{\boldsymbol{\sigma}_y} \widetilde{G}_{\boldsymbol{\sigma}_x \boldsymbol{\sigma}_y} \widetilde{H}_{\boldsymbol{\sigma}_y \boldsymbol{\sigma}_z} = \widetilde{F}_{\boldsymbol{\sigma}_x \boldsymbol{\sigma}_y}$ (28)

Use quantics for each variable: $\mathbf{x} = \mathbf{x}(\boldsymbol{\sigma}_x)$, $\boldsymbol{\sigma}_x = (\sigma_{1x}, \ldots, \sigma_{\mathcal{L}x})$, $\mathcal{L} = \mathcal{N}\mathcal{R}$ (29)



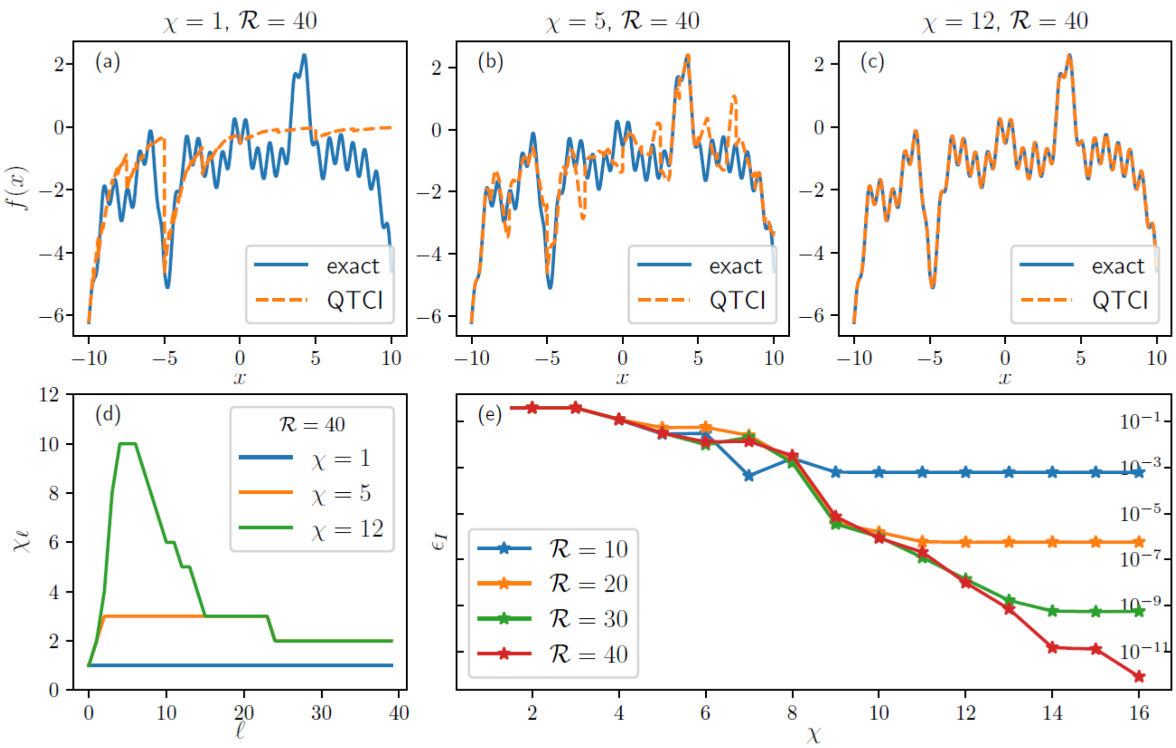
(30)

16. Numerical examples of QTCI

TCI.16

1D oscillating function: $f(x) = \mathrm{sinc}(x) + 3e^{-0.3(x-4)^2}\,\mathrm{sinc}(x-4) - \cos(4x)^2 - 2\,\mathrm{sinc}(x+10)e^{-0.6(x+9)}$ (1)

$\mathrm{sinc}(x) = \sin x / x$ $\qquad + 4\cos(2x)e^{-|x+5|} + \dfrac{6}{x-11} + \sqrt{(|x|)}\arctan(x/15)\,,\qquad x \in [-10, 10]$

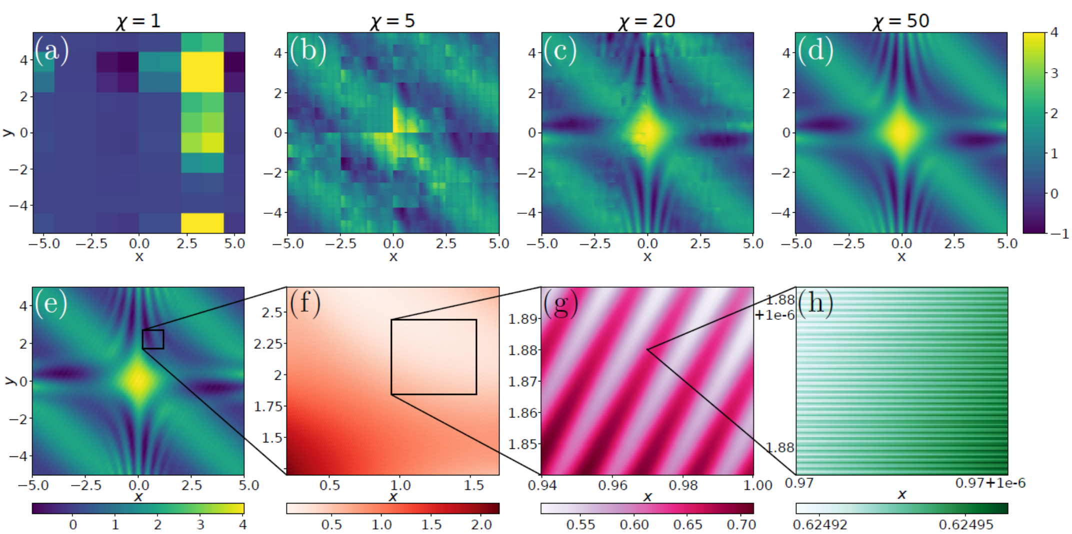


(e) Relative error $\epsilon_I = |\widetilde{I}/I - 1|$ of the integral $I = \int_{-10}^{10} \mathrm{d}x\, f(x)$ converges rapidly with increasing χ .

2D oscillating function (interleaved representation)

$$f(x,y) = 1 + e^{-0.4(x^2+y^2)} + \sin(xy)\,e^{-x^2} + \cos(3xy)\,e^{-y^2} + \cos(x+y)$$
$$+ 0.05\cos\left[10^2\cdot(2x-4y)\right] + 5\cdot 10^{-4}\cos\left[10^3\cdot(-2x+7y)\right] + 10^{-5}\cos\left(2\cdot 10^8 x\right) \qquad (2)$$

This function has structure (oscillations) on different scales. A QTT with $\mathcal{R} = 40$, $\chi = 50$ resolves them all!
At $\chi = 110$ the QTT becomes numerically exact (within machine precision).



QTCI representation reaches machine precision at $\chi = 110$

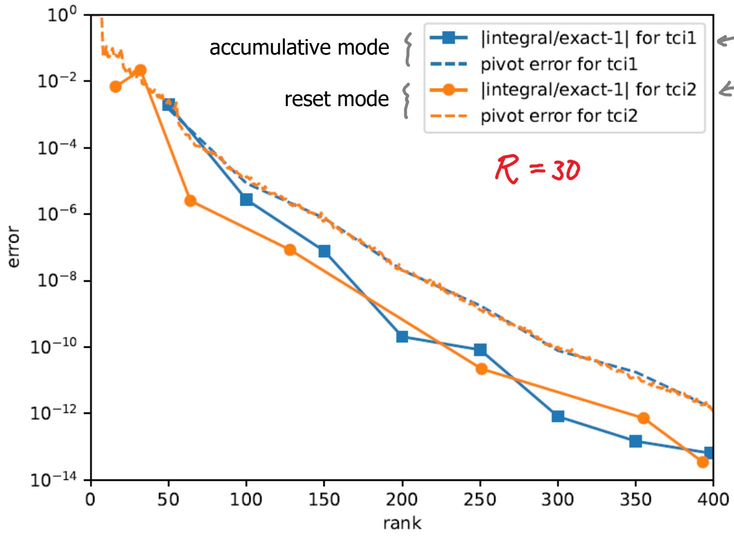
requires only 10^5 numbers
~ 1 MB of RAM

10^{19} orders difference!

naïve regular grid would require $\sim 10^{13}$ TB of Ram

3D integral: $I = \int_{\mathbb{R}^3} d^3\mathbf{x}\, e^{-\sqrt{x^2+y^2+z^2}} = 8\pi$

$[-40, 40]^3$



$\varepsilon_I = |\tilde{I}/I - 1|$

17. Quantics Fourier transform (QFT)

TCI.17

Remarkable fact: when using quantics representations of functions, the Fourier transform (FT) operator, represented as an MPO, has remarkably low rank ($\chi \approx 11$ for machine precision in 1D). Thus, Therefore, taking FT of functions having low-rank quantics TTs can be done exponentially faster(!) than with fast Fourier transform (FFT).

Goal: compute 1D FT: $\hat{f}(k) = \int dx\, f(x) e^{-ik\cdot x}$ (1)

Discretize: $f_m = f(x(m)) \in \mathbb{C}$ on uniform 1D grid, $x(m)$ (2)

Discrete FT (DFT): $\hat{f}_k = \sum_{m=0}^{M-1} T_{km} f_m$, $T_{km} = M^{-1/2} e^{-i2\pi k\cdot m/M}$ (3)

(or 'quantum FT')

quantics grid: $x = m/M$, with $M = 2^R$, $m = \sum_{r=1}^{R} \sigma_r 2^{R-r} = 0, \dots, M-1$, $\sigma_r \in \{0,1\}$ (4)

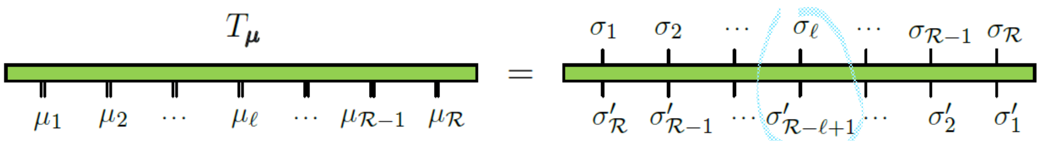
[see (TCI.15.1-5)]

Quantics grid has exponentially many grid points, so naïve computation of DFT sum in (3) is exponentially expensive.

QTCI strategy: find QTT representing T , then compute $\hat{f} = Tf$ by contracting TTs for T and f .

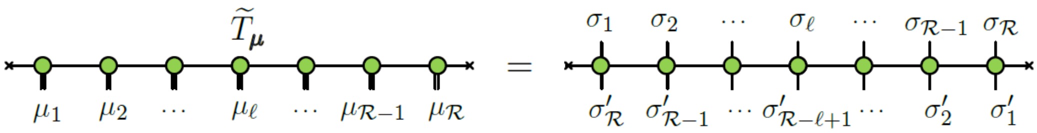
quantics representations: $m(\vec{\sigma}) = (\sigma_1 \cdots \sigma_R)_2 = \sum_{\ell=1}^{R} \sigma_\ell 2^{R-\ell}$, $k(\vec{\sigma}') = (\sigma'_1 \cdots \sigma'_R)_2 \sum_{\ell'=1}^{R} \sigma'_{\ell'} 2^{R-\ell'}$ (5)

FT operator: $T_{\vec{\mu}} = T_{\vec{\sigma}'\vec{\sigma}} = T_{k(\vec{\sigma}')\,m(\vec{\sigma})} = M^{-1/2} \exp\left[-i2\pi \sum_{\ell'\ell} 2^{R-\ell'-\ell} \sigma'_{\ell'} \sigma_\ell\right]$ (6)

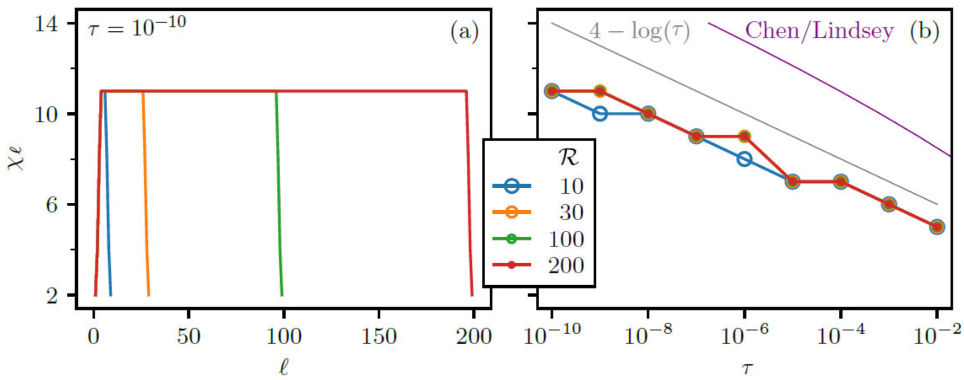
$=$  (7)

fused index: $\vec{\mu} = (\mu_1, \dots, \mu_R)$, $\mu_\ell = (\underbrace{\sigma'_{R-\ell+1}}_{2^{\ell-1}}, \underbrace{\sigma_\ell}_{2^{R-\ell}})$ $\leftarrow$ important: bits for $k(\vec{\sigma}')$ and $m(\vec{\sigma})$ are arranged in 'scale-reversed' order, to respect Fourier repricocity (8)

describes scale

Unfolded: $\widetilde{T}_{\vec{\mu}} =$  (9)

Scale-reversed ordering (8) ensures that $\widetilde{T}_{\vec{\mu}}$ has low rank:



$\chi = 11$ suffices for machine precision! (10)

$|T_\mu - \widetilde{T}_\mu|_\infty / |T_\mu|_\infty < 10^{-10}$

irrespective of R !

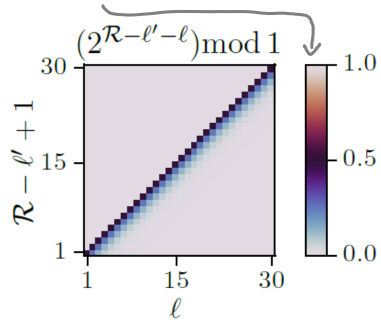
[Shinaoka2023, Chen2023]

Conclusion: for 1D function with rank χ' , the DFT can be obtained in $\mathcal{O}(\chi^2 \chi'^2 R) = \mathcal{O}(\chi^2 \chi'^2 \ln M)$ (11)

operations, exponentially faster than FFT, which needs $\mathcal{O}(M \ln M)$ operations.

Intuitive reason for low rank: for scale-reversed index order,
phase factor in DFT has simple structure.
Thus, it involves only 'short-range entanglement'.

$\exp\left[-i 2\pi \cdot 2^{\mathcal{R}-\ell'-\ell} \sigma_{\ell'} \sigma_\ell\right]$



Example: Quantics FT can be used to solve partial differential equation [Fernandez2025, Sec. 6.4]

1D heat equation: $\partial_t u(x,t) = \partial_x^2 u(x,t)$

Discretize position variable: $u_m(t) = u(x(m), t)$

In momentum space, solution is known analytically:

$$u_k^{\mathrm{FT}}(t) = g_k(t) u_k^{\mathrm{FT}}(0), \qquad g_k(t) = \exp\left[-(2/\delta)^2 \sin^2(\pi k/M) t\right]$$

$t=0$

Strategy: $u_m(0) \xrightarrow{\text{QTCI}} \widetilde{U}_{\boldsymbol{\sigma}}(0), \qquad g_k(t) \xrightarrow{\text{QTCI}} \widetilde{G}_{\boldsymbol{\sigma}'}(t), \qquad T_{km} \xrightarrow{\text{QTCI}} \widetilde{T}_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}$

$$\widetilde{U}_{\boldsymbol{\sigma}}(0) \xrightarrow{\times \widetilde{T}_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}} \widetilde{U}^{\mathrm{FT}}_{\boldsymbol{\sigma}'}(0) \xrightarrow{\times \widetilde{G}_{\boldsymbol{\sigma}'}(t)} \widetilde{U}^{\mathrm{FT}}_{\boldsymbol{\sigma}'}(t) \xrightarrow{\times \widetilde{T}^{-1}_{\boldsymbol{\sigma}\boldsymbol{\sigma}'}} \widetilde{U}_{\boldsymbol{\sigma}}(t).$$

Nasty initial condition: $u(x,0) = \frac{1}{100}\left[1 + \cos(120x)\sin(180x)\right] + \theta\left(x - \frac{7}{2}\right)\left[1 - \theta\left(x - \frac{13}{2}\right)\right]$